

IndraDrive

IndraMotion MLD (2G)
Bibliotheken ab MPx-18

Referenz
R911338915

Ausgabe 06



Titel	IndraDrive IndraMotion MLD (2G) Bibliotheken ab MPx-18
Art der Dokumentation	Referenz
Dokumentations-Type	DOK-INDRV*-MLD-SYSLIB3-RE06-DE-P
Interner Ablagevermerk	RS-86e472c00f785de60a6846a5017a43c8-6-de-DE-14
Zweck der Dokumentation	Diese Dokumentation beschreibt die von IndraMotion MLD zur Verfügung gestellten SPS-Funktionen/-Funktionsbausteine und Datentypen, sowie eine Auflistung der möglichen Fehlerreaktionen. Die Dokumentation ist für folgende Firmwaren gültig: • ab MPx-18 • ab PSB-19
Ausgabestand	Ausgabe 06, 2018-10 Änderungsverlauf siehe ""Ausgaben dieser Dokumentation" auf Seite 7"
Schutzvermerk	© Bosch Rexroth AG 2018 Alle Rechte vorbehalten, auch bzgl. jeder Verfügung, Verwertung, Reproduktion, Bearbeitung, Weitergabe sowie für den Fall von Schutzrechtsanmeldungen.
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.
Redaktion	DC-AE/EPI5 (SA), DC-AE/EME1 (ML), DC-AE/EFC2 (KA)

Inhaltsverzeichnis

	Seite
1	Über diese Dokumentation..... 7
2	Allgemeines..... 11
3	Bibliotheken..... 13
3.1	Unterstützte Bibliotheken..... 13
3.2	Abhängigkeiten zwischen Parametrierung der Achse, Funktionspaket und ausgewähltem Funktionsbaustein..... 15
3.3	Funktionsbausteine beim Firmware-Derivat PSB..... 15
4	Signallaufverhalten der am häufigsten verwendeten Ein- und Ausgänge..... 17
5	Fehlerbehandlung..... 25
5.1	Allgemeines..... 25
5.2	Fehler-Referenzlisten..... 26
6	Datentypen..... 33
7	Parameterstruktur..... 35
7.1	Aufbau..... 35
7.2	Sercos Attribut..... 35
8	IecSfc..... 37
9	MX_Base..... 39
9.1	Einleitung..... 39
9.2	Datentypen..... 39
9.3	Globale Variablenliste (Global Variables List)..... 40
10	MX_CanL2..... 43
10.1	Einleitung..... 43
10.2	MX_CANL2_FifoReceive..... 43
10.3	MX_CANL2_FifoSend..... 44
10.4	MX_CANL2_Init..... 44
10.5	MX_CANL2_SetLED..... 46
10.6	MX_fCANL2_MBoxReceive..... 47
10.7	MX_fCANL2_MBoxSend..... 48
10.8	MX_fCANL2_Status..... 49
10.9	MX_fCANL2_MBoxRecStatus..... 49
10.10	MX_fCANL2_MBoxSndStatus..... 50
10.11	MX_fCANL2_MBoxUpdate..... 50

Inhaltsverzeichnis

	Seite
10.12 MX_fCANtoHI.....	51
10.13 MX_fCANtoHs.....	52
10.14 MX_fHtoCANI.....	52
10.15 MX_fHtoCANS.....	53
 11 MX_CheckRtv.....	 55
 12 MX_Debug.....	 57
 13 MX_PLCOpen.....	 59
13.1 Einleitung.....	59
13.2 "Drive control" - Funktionen für Diagnose.....	59
13.2.1 MX_fCheckCmdRequest.....	59
13.2.2 MX_fGetDriveWarning.....	59
13.2.3 MX_fSetDriveError.....	60
13.2.4 MX_fSetDriveErrorText.....	60
13.2.5 MX_fSetDriveWarning.....	61
13.2.6 MX_fSetDriveWarningText.....	61
13.2.7 MX_SetCmdState.....	62
13.3 "Motion" - Funktionsbausteine/Funktionen zur Antriebsregelung.....	63
13.3.1 Allgemeine Eigenschaften.....	63
13.3.2 MB_ChangeCamData.....	63
13.3.3 MB_ChangeProfileSet	66
13.3.4 MB_ChangeProfileStep.....	69
13.3.5 MB_ClearAllError.....	71
13.3.6 MB_GearInPos.....	72
13.3.7 MB_Home.....	77
13.3.8 MB_MotionProfile.....	78
13.3.9 MB_Phasing.....	81
13.3.10 MB_PhasingSlave.....	83
13.3.11 MB_PreSetMode.....	85
13.3.12 MB_SetPositionControlMode.....	86
13.3.13 MB_Stop.....	88
13.3.14 MC_CamIn.....	90
13.3.15 MC_CamOut.....	93
13.3.16 MC_GearIn.....	94
13.3.17 MC_GearOut.....	96
13.3.18 MC_Jog.....	97
13.3.19 MC_MoveAbsolute.....	101
13.3.20 MC_MoveAdditive.....	104
13.3.21 MC_MoveRelative.....	106
13.3.22 MC_MoveVelocity.....	109
13.3.23 MC_Power.....	111
13.3.24 MC_ReadStatus.....	113
13.3.25 MC_Reset.....	114

Inhaltsverzeichnis

	Seite
13.3.26 MC_Stop.....	116
13.3.27 MC_TorqueControl.....	118
13.3.28 MX_MoveAbsolute.....	120
13.3.29 MX_MoveAdditive.....	123
13.3.30 MX_MoveRelative.....	126
13.3.31 MX_SetControl.....	129
13.3.32 MX_SetDeviceMode.....	130
13.3.33 MX_SetOpMode.....	132
13.4 "Parameter" - Funktionsbausteine/Funktionen für Parameter.....	135
13.4.1 MB_Command.....	135
13.4.2 MB_ReadListParameter.....	141
13.4.3 MB_ReadMaxRealValue.....	144
13.4.4 MB_ReadMaxValue.....	146
13.4.5 MB_ReadMinRealValue.....	147
13.4.6 MB_ReadMinValue.....	149
13.4.7 MB_ReadName.....	150
13.4.8 MB_ReadParameter.....	152
13.4.9 MB_ReadRealListParameter.....	154
13.4.10 MB_ReadRealParameter.....	156
13.4.11 MB_ReadSercosAttribute	158
13.4.12 MB_ReadSercosDataStatus.....	160
13.4.13 MB_ReadStringParameter.....	162
13.4.14 MB_ReadUnit.....	164
13.4.15 MB_WriteListParameter	166
13.4.16 MB_WriteParameter.....	168
13.4.17 MB_WriteRealListParameter	170
13.4.18 MB_WriteRealParameter.....	173
13.4.19 MB_WriteStringParameter.....	175
13.4.20 MX_Command.....	177
13.4.21 MX_fReadParamDINT.....	180
13.4.22 MX_fReadStringParam.....	181
13.4.23 MX_fSetParamLimits.....	181
13.4.24 MX_fSetParamName.....	182
13.4.25 MX_fSetParamUnit.....	183
13.4.26 MX_fWriteParamDINT.....	183
13.4.27 MX_fWriteStringParam.....	185
13.4.28 MX_ReadParamDINT.....	186
13.4.29 MX_WriteParamDINT.....	187
13.5 "Scaling" - Funktionen für Skalierung.....	188
13.5.1 MX_fDINT_AccTo_REAL.....	188
13.5.2 MX_fDINT_DistTo_REAL	189
13.5.3 MX_fDINT_VelTo_REAL	189
13.5.4 MX_fREAL_AccTo_DINT	190
13.5.5 MX_fREAL_DistTo_DINT	191
13.5.6 MX_fREAL_TorqueRampTo_DINT	191
13.5.7 MX_fREAL_TorqueTo_DINT	192

Inhaltsverzeichnis

	Seite
13.5.8 MX_fREAL_VelTo_DINT	192
13.6 "Tools" - allgemeine Funktionen.....	193
13.6.1 MX_fGetFreeTicks.....	193
13.6.2 MX_fGetHighResTime.....	193
13.6.3 MX_fHighResTimerTicks_to_us.....	194
13.6.4 MX_IECTaskGetLoad.....	194
13.6.5 MX_PositionLoopEventInfo.....	195
14 Basisbibliotheken.....	197
14.1 MX_CommonTypes.....	197
14.1.1 Einleitung.....	197
14.1.2 Globale Variablen.....	198
14.1.3 Sercos Attribute.....	198
14.1.4 Datatypes of logical addresses.....	198
14.1.5 Datatypes of PLCopen.....	200
14.1.6 Struktur "ERROR_STRUCT".....	205
14.1.7 Datentypen zur Wiedergabe des Datums und absoluter Zeitangaben bei IndraMotion-Bausteinen.....	205
14.1.8 Datentypen zur Unterstützung von Messtaster-Funktionen.....	207
14.1.9 Datentypen zur Unterstützung von Kurvenscheiben, Motion- und FlexProfilen.....	212
14.2 RIL_HMI_Uutilities.....	235
14.3 RIL_LoopControl.....	236
14.4 RIL_ModbusTCP.....	236
14.5 RIL_SercosIII.....	236
14.5.1 Allgemeines.....	236
14.6 RIL_SocketComm.....	236
14.7 RIL_Uutilities.....	236
14.8 RMB_SercosIII_Util.....	237
15 Systembibliotheken.....	239
15.1 Allgemeines.....	239
15.2 Standard.....	239
15.3 Util.....	239
15.4 SysDir.....	239
15.4.1 Allgemeines.....	239
15.4.2 Verwendungshinweise.....	239
15.5 SysFile.....	240
15.5.1 Allgemeines.....	240
15.5.2 Verwendungshinweise.....	240
15.6 SyslecTasks23.....	240
15.7 SysMem.....	241
15.8 SysCallback23.....	241
15.9 SysSem.....	241
15.10 SysSockets.....	241
15.11 SysStr23	242

Inhaltsverzeichnis

	Seite
15.11.1 Übersicht.....	242
15.11.2 SysStrCmp	242
15.11.3 SysStrCmpl	242
15.11.4 SysStrCmpN	243
15.11.5 SysStrCmpNI	243
15.11.6 SysStrCpy	243
15.11.7 SysStrLen	244
15.12 SysTimeRTC.library.....	244
15.12.1 Einleitung.....	244
15.12.2 SysTimeRtcGet.....	244
15.12.3 SysTimeRtcSet.....	244
15.12.4 SysTimeRtcHighResGet.....	245
15.12.5 SysTimeRtcHighResSet.....	245
15.13 IoDrvEtherNetIP.library.....	246
15.13.1 Lizenzierung und Betrieb.....	246
15.13.2 Performance und Zykluszeit der MLD.....	247
15.13.3 Ressourcenbedarf im Bootprojekt.....	247
15.13.4 Unterstützte Schnittstellen und Koexistenz mit Sercos.....	247
15.13.5 Anforderungen an das IP-Netzwerk.....	248
15.13.6 Einschränkungen des Funktionsumfangs der Bibliothek.....	249
15.13.7 Projektierung des EtherNet/IP-Scanners.....	249
15.13.8 Editoren für die Konfiguration von EtherNet/IP.....	252
15.13.9 Azyklische Kommunikation (Explicit Messaging Service).....	259
16 Technologie-Bibliotheken.....	263
16.1 Einleitung.....	263
16.2 MX_TechInterface.....	263
16.2.1 Übersicht.....	263
16.2.2 Anwendung des Achs-Interface im System IndraMotion MLD.....	263
16.3 MX_TechBase.....	266
16.4 MX_TechMotionBase.....	266
16.5 MX_TechMotion.....	266
16.6 MX_TechWinder.....	266
16.7 MX_TechCrosscutCrossseal	267
16.8 RMB_TechCam.....	267
17 Freie SPS-Bibliothek von "OSCAT".....	269
Index.....	271

1 Über diese Dokumentation

Ausgaben dieser Dokumentation

Ausgabe	Stand	Bemerkung
01	2013-01	Enthält Beschreibungen der Bibliotheken für das Zielsystem "IndraDrive MP18"
02	2013-11	Korrekturen und Ergänzungen
03	2014-07	- Bibliotheken "MX_TechInterface" eingefügt - Korrekturen und Ergänzungen
04	2015-10-09	Enthält Beschreibungen der Bibliotheken für das Zielsystem "IndraDrive MP20" - Bibliothek "SysTimeRTC" eingefügt - Technologie-Bibliotheken "MX_TechMotionBase", "MX_TechBase", "MX_TechMotion", "MX_TechWinder" eingefügt - Korrekturen und Ergänzungen - Beschreibung von "MX_Base" erweitert - Beschreibung von "MX_CommonTypes" erweitert
05	2016-08-11	- Ab MPx-20: Neue Bibliotheken "IoDrvEtherNetIP" "RMB_SercosIII_Util" - Technologie-Bibliothek "MX_TechCrosscutCrossseal" - Ab MPx-20: VmAxis1, RmAxis1, LinkAxis1 sind bei den Synchronisationsbausteinen auch bei MLD-S als Masterachse verwendbar
06	2018-10-31	- Fehlertabelle bei "MX_CANL2_Init" ergänzt - Schreibweise von "Sercos" angepasst - Datentyp von pResult in "SysTimeRtcGet" korrigiert in RTS_IEC_RESULT - Hinweise bei "MX_PLCOpen.library", "Parameter" aufgenommen: - Lange Laufzeit bei Schreib-/Lesezugriffen auf einige Parameter - Wenn Parameter durch ein Passwort schreibgeschützt sind, ist auch das Beschreiben von Parameterwerten über Funktionen oder Funktionsbausteine nicht möglich - Technologiebibliothek "RMB_TechCam" ergänzt

Tab. 1-1: Änderungsverlauf

Darstellungsmittel in dieser Dokumentation

Um Ihnen das Lesen in dieser Dokumentation zu erleichtern, erhalten Sie hier einen Überblick über die Darstellungsmittel und Schreibweisen immer wiederkehrender Begriffe.

Was?	Wie?	Zum Beispiel...
Pfade und Menüfolgen...	...sind in Fettschrift Schritt für Schritt gekennzeichnet dargestellt	Klicken Sie auf Fenster ▶ Sicht anzeigen ▶ Eigenschaften
Schaltflächen und Tastenbezeichnungen...	... sind in spitzen Klammern dargestellt	Klicken Sie auf <Hinzufügen> oder Drücken Sie <Strg>+<Alt>

Über diese Dokumentation

Was?	Wie?	Zum Beispiel...
Wichtige Sachverhalte die im Fließtext hervorgehoben werden sollen	Fettschrift	Bei entfernten Achsen wird mit "Active"=TRUE angezeigt, dass ...
Parameternamen, Diagnosenamen, Funktionsbezeichnungen	Anführungszeichen	"Axis2" bis "Axis10" entsprechen den entfernten Achsen. Die in der Liste "P-0-1801.x.10, CCD: Adressen projektierter Antriebe" vorhandenen ...

Tab. 1-2: Konventionen in der Schreibweise

Alle wichtigen Hinweise sind im Text besonders hervorgehoben. Sie sind jeweils mit einem Symbol versehen, das Ihnen sagt, um welche Art es sich handelt. Sie haben folgende Bedeutung:

⚠ GEFAHR

...

Bei der Missachtung eines Gefahrenhinweises **werden** Tod oder schwere Körperverletzung eintreten.

⚠ WARNUNG

...

Bei der Missachtung einer Warnung **können** Tod oder schwere Körperverletzung eintreten.

⚠ VORSICHT

...

Bei Missachtung **können** mittelschwere oder leichte Körperverletzung eintreten.

HINWEIS

...

Bei Nichtbeachten eines Sicherheitshinweises **können** Sachschäden eintreten.



Dieser Hinweiskasten gibt Ihnen wichtige Informationen auf die Sie achten sollten.



Mit dieser Darstellung werden nützliche Tipps und Tricks gekennzeichnet.

Ihr Feedback

Ihre Erfahrungen sind für uns ein wichtiger Bestandteil im Verbesserungsprozess für Produkt und Dokumentation.

Wenn Sie in dieser Dokumentation Fehler entdecken oder Änderungen wünschen, können Sie Ihre Rückmeldungen an folgende E-Mail-Adresse senden:

Dokusupport@boschrexroth.de

Für die Bearbeitung ihres Feedbacks geben Sie bitte folgendes an:

- Die Nummer, die unter "Interner Ablagevermerk" angegeben ist.
- Die Seitenzahl.

Dokumentation als Online-Hilfe

Diese Dokumentation ist auch in das Hilfesystem zu IndraDrive eingebunden.

Über diese Dokumentation

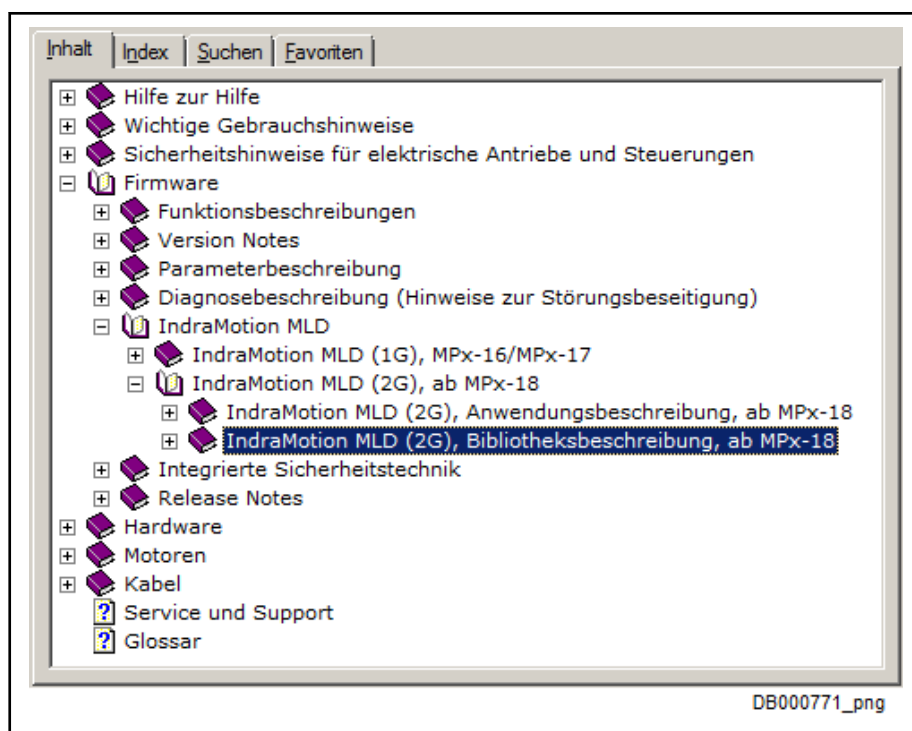


Abb. 1-1: Beschreibung der "Rexroth IndraDrive-Firmware-Bibliotheken" in dem Hilfesystem zu IndraDrive

Mitgeltende Dokumentationen

Titel	Dokumentationsart	Dokumentations-Type ¹⁾	Materialnummer
IndraDrive IndraMotion MLD ab MPx18	Anwendungsbeschreibung	DOK-INDRV*-MLD3-**VRS*-APRS-DE-P	R911338913
IndraWorks 14VRS IndraLogic 2G SPS-Programmiersystem	Anwendungsbeschreibung	DOK-IWORKS-IL2GPRO*V14-APxx-DE-P	R911343570
IndraWorks 14VRS Basis-Bibliotheken IndraLogic 2G	Bibliotheksbeschreibung	DOK-IL*2G*-BASLIB**V14-Llxx-DE-P	R911343918
IndraWorks 11VRS SPS-System-Bibliotheken IndraLogic 2G	Bibliothek	DOK-IL*2G*-SYSLIB**V11-Llxx-DE-P	R911336156
IndraMotion MLC 14VRS Technologie-Bibliotheken	Bibliothek	DOK-MLC***-TF*LIB**V14-Llxx-DE-P	R911341510
IndraWorks 14VRS Feldbus-Bibliotheken	Bibliothek	DOK-IWORKS-FB*LIB**V14-Llxx-DE-P	R911343574

¹⁾ In den Dokumentations-Typen ist "xx" ein Platzhalter für den aktuellen Ausgabestand der Dokumentation (Beispiel: AP01 bedeutet die erste Ausgabe einer Anwendungsbeschreibung)

Tab. 1-3: Dokumentationen – Übersicht

2 Allgemeines

Mit dem SPS-Programmiersystem IndraLogic 2G können SPS-Komponenten einer Steuerung vom Typ IndraLogic XLC / IndraMotion MLC, IndraMotion MLD oder IndraMotion MTX programmiert und in Betrieb genommen oder durch eine intern genutzte Visualisierung unterstützt werden. IndraLogic unterstützt die Verwaltung unterschiedlicher Bibliotheken.

Welche Hardware vorhanden sein muss, um die antriebsintegrierten Steuerungen "IndraMotion MLD" nutzen zu können, können Sie in der jeweiligen Anwendungsbeschreibung nachlesen:

Beschreibung	Titel	Dokumentations-Type ¹⁾	Materialnummer
Firmware FWA-INDRV-MP*02VRS bis FWA-INDRV-MP*08VRS	Rexroth IndraDrive Rexroth IndraMotion MLD	DOK-INDRV*-MLD-**VRS*- AWxx-DE-P	R911306071
Firmware FWA-INDRV-MP*17VRS	Rexroth IndraDrive Rexroth IndraMotion MLD ab MPx-17	DOK-INDRV*-MLD2-**VRS*- APxx-DE-P	R911334350
ab Firmware FWA-INDRV-MP*18VRS	IndraDrive IndraMotion MLD ab MPx-18	DOK-INDRV*-MLD3-**VRS*- APxx-DE-P	R911338913

- 1) In den Dokumentations-Typen ist "xx" ein Platzhalter für den aktuellen Ausgabestand der Dokumentation (Beispiel: AP01 bedeutet die erste Ausgabe einer Anwendungsbeschreibung)

Tab. 2-1: Dokumentationen – Übersicht

Außerdem muss eines der Funktions-Erweiterungspakete "IndraMotion MLD" freigeschaltet sein (siehe auch Firmware-Funktionsbeschreibung: "Funktionspaketfreischaltung").

Wenn ein Projekt mit einem IndraDrive-Zielsystem angelegt wird, werden einige Rexroth-Bibliotheken automatisch an das Projekt gebunden. Über den Bibliotheksverwalter ist es auch möglich Bibliotheken nachträglich zu einem Projekt hinzuzufügen, um dann deren Funktionen nutzen zu können.

Bibliotheken können sowohl Bausteine als auch Deklarationen und / oder Variablen enthalten. Mit der antriebsintegrierten SPS (Rexroth IndraMotion MLD) werden Basisbibliotheken geliefert, die wesentliche Bausteine und Definitionen/Variablen enthalten.

Je nach Typ der Bibliothek kann man bei den Bausteinen IEC-Funktionen / Funktionsbausteine erstellen oder auch Firmware-Bausteine definieren.

So genannte "Interne Bibliotheken" beinhalten den IEC-Code.



In der Dokumentation wird häufig nur von dem Antrieb/ Antriebsregelgerät oder der Achse gesprochen, da die Funktionen ursprünglich für Antriebsregelgeräte entwickelt wurden (Firmware MPx). In vielen Fällen sind die Funktionen auch in der Firmware PSB (IndraDrive ML als Versorgungsgeräte) nutzbar.

3 Bibliotheken

3.1 Unterstützte Bibliotheken

HINWEIS

Sachschäden durch das Einbinden von nicht unterstützten Bibliotheken.

Es dürfen nur Bibliotheken eingebunden werden, die hier aufgeführt sind.

Bibliothek	Beschreibung
IecSfc	Stellt normkonforme IEC-Schritte in Ablaufsprache (AS) [englisch: "Sequential Function Chart" (SFC)] zur Verfügung
IoDrvEtherNetIP	Es wird ein EtherNet/IP-Scanner zur Verfügung gestellt. Mit ihm ist es möglich andere EtherNet/IP-Geräte als Slaves einzubinden (z. B. Barcode-Leser, RFID, Durchflussmesser, Laser-Entfernungsmesser, I/O, HMI,...).
MX_Base (IndraDrive) / MY_Base (HydraulicDrive)	- Datentypen bzw. Strukturen - Zyklische Parameter als Direktvariable (systemglobale Variable) - Achsstrukturen [bei Mehrachssystem (MLD-Systemmodus)]
MX_CanL2 (nur IndraDrive)	Enthält Datentypen, Funktionen und Funktionsbausteine für CAN-Kommunikation auf Level 2
MX_CommonTypes	Enthält Datentypen und Strukturen, die zum größten Teil nur intern verwendet werden
MX_CheckRtv	Nur zur programminternen Verwendung!: Funktionen zum Prüfen bzw. Melden von Laufzeitfehlern
MX_Debug	Nur für interne Testzwecke (Labor)!
MX_PLCCopen	- Funktionen für Diagnose - Funktionsbausteine/Funktionen zur Antriebsregelung - Funktionsbausteine/Funktionen für Parameter - Funktionen für Skalierung - allgemeine Funktionen
MX_TechInterface (ab Firmware-Version 20)	Das Achs-Interface bündelt und erweitert PLCopen-Bewegungs-Funktionsbausteine und stellt eine einfach zu bedienende Schnittstelle für die Antriebsfunktionalität zur Verfügung. Weniger Code und leistungsfähigere Kommandos beschleunigen die Programmentwicklung von Applikationen. Das Achs-Interface enthält Steuersignale und Parameter für die verschiedenen Betriebsarten der Leitachse und Folgeachse sowie Einstellmöglichkeiten für angewählte Prozesswerte.
MX_TechBase (ab Firmware-Version 20)	Beinhaltet Basis-Funktionalitäten, wie Kontroll- und Steuerungsfunktionen, Funktionen, um Systeminformationen abzufragen und trigonometrische Funktionen.
MX_TechMotionBase (ab Firmware-Version 20)	Beinhaltet Funktionalitäten, wie Messtasterfunktion, Positionsüberwachung, Nockenschaltwerk und Funktionen zur Verwaltung des zyklischen Datenkanals.
MX_TechMotion (ab Firmware-Version 20)	Beinhaltet einen Funktionsbaustein, um die Zugkraft auf einen Sollwert zu regeln. Hierzu wird der Messwert einer Kraftmessdose ausgewertet.
MX_TechWinder (ab Firmware-Version 20)	Beinhaltet Funktionsbausteine, die Funktionen für das Auf- und Abwickeln von Stoffbahnen (Wicklern) bereitstellen.
MX_TechCrosscutCrossseal (ab Firmware-Version 20)	Beinhaltet Funktionsbausteine für Quersiegler- und Querschneider-Anwendungen.
RIL_Fieldbus	Stellt gemeinsame Datentypen (Feldbustypen) für die Rexroth-Feldbusbibliotheken bereit
RIL_HMI_Uilities	Enthält Funktionsbausteine, die die HMI-Geräte unterstützen
RIL_LoopControl	Stellt Grundelemente und Regler der Regelungstechnik zur Verfügung
RIL_ModbusTCP	Stellt Funktionsbausteine zur Verfügung, mit denen die Kommunikation zwischen Ethernet-Geräten, die das Modbus-Protokoll unterstützen, ermöglicht wird
RIL_SercosIII (nur IndraDrive)	Stellt Funktionen zur Verfügung, mit denen die Kommunikation zwischen der SPS-Programmierungsumgebung und Sercos III-Teilnehmern ermöglicht wird

Bibliotheken

Bibliothek	Beschreibung
RIL_SocketComm	Funktionen und Funktionsbausteine, um die TCP-/UDP-Kommunikation nutzen zu können
RIL_Uilities	Funktionen und Funktionsbausteine für Konvertierung und Beeinflussung verschiedener Datentypen
RMB_SercosIII_Util (ab Firmware-Version 20)	Funktionsbausteine für die Achsansteuerung über den Sercos Bus
RMB_TechCam (nur IndraDrive; ab Firmware-Version 20)	Funktionsbausteine zum Erstellen von Kurvenscheibentabellen
CmplecTask	Systembibliothek, die unter anderem Schnittstellenfunktionen enthält, um einen Watchdog abzuschalten (und anschließend wieder einzuschalten) (siehe MLD-Anwendungsbeschreibung: "Task-Überwachung (Watchdog)")
SysDir / SysDir23*1	Funktionen, um synchron auf ein Datei-Verzeichnissystem auf dem Zielsystem zuzugreifen
SysFile / SysLibFile23*1	Funktionen, um synchron auf Dateien zuzugreifen
SysFileAsync / SysFileAsync23*1	Funktionen, um asynchron auf Dateien zuzugreifen
SyslecTasks23	Unterstützt wird die Funktion "SysIECTaskGetConfig", mit der die Task-Konfiguration ausgelesen werden kann
SysMem / SysMem23*1	Funktionen zur Speicherverwaltung
NetVarUdp	Nur zur programminternen Verwendung!: Funktionen zur Verarbeitung von Netzwerkvariablen
Standard	Funktionen und Funktionsbausteine, die von der IEC 61131-3 als Standardbausteine für ein IEC-Programmiersystem gefordert werden
SysCallback23*1	Funktionen, um definierte Callback-Funktionen für Laufzeitergebnisse zu aktivieren
SysSem / SysSem23*1	Funktionen, um Semaphore für die Synchronisation von Tasks zu erzeugen und zu benutzen
SysSockets / SysSockets23*1	Funktionen, mit denen der Zugriff auf Sockets zur Kommunikation über TCP/IP und UDP unterstützt wird
SysStr23*1	Funktionen für den Umgang mit Strings
SysTime	Stellt Funktionen zur Verfügung, die das Auslesen bzw. Setzen der Echtzeituhr des lokalen Systems erlauben, sowie diverse Konvertierungen der Uhrzeitdaten ermöglichen. HINWEIS: Bitte überprüfen Sie zunächst, ob Ihre Anforderungen durch die Bibliothek "RIL_Uilities" erfüllt werden können.
SysTimeRTC	Stellt Funktionen zur Verfügung, mit denen die Systemzeit des Antriebs im IEC 61588-Format ausgelesen bzw. beschrieben werden kann ("S-0-1305.0.1, Systemzeit").
Util	Funktionsbausteine, die für BCD-Konvertierung, Bit/Byte-Funktionen, mathematische Hilfsfunktionen, als Regler, Signalgeneratoren, Funktionsmanipulatoren und zur Analogwertverarbeitung verwendet werden können
OSCAT	Freie SPS-Bibliothek "OSCAT" ("Open Source Community for Automation Technology"). Diese quelloffene Bibliothek beinhaltet eine Vielzahl nützlicher Funktionen in den Bereichen "Automatisierungstechnik" und "Gebäudeautomation".

***1** Bibliotheken mit der Endung "23" (wie z. B. "SysSockets23") dienen dazu, Projekte portieren zu können; für neue Projekte sollen die Bibliotheken ohne Endung "23" verwendet werden



"*.library" ist das Standardformat für Bibliotheken.

Bei den über IndraWorks ausgelieferten Bibliotheken ist der Code geschützt, sie besitzen das Format "*.compiled-library-ge33".

3.2 Abhängigkeiten zwischen Parametrierung der Achse, Funktionspaket und ausgewähltem Funktionsbaustein

Je nach Funktionspaket und Parametrierung der Achse sind nicht alle Funktionsbausteine lauffähig. In der nachfolgenden Tabelle sind die Abhängigkeiten zwischen Parametrierung der Achse, Funktionspaket und ausgewählter Betriebsart dargestellt.

P-0-2003 (1)	P-0-0045, Bit 14/15 (2)	P-0-0075 (3)	Positioniersatzbetrieb (MX_MoveRelative / MC_MoveRelative, MX_MoveAbsolute / MC_MoveAbsolute, MX_MoveAdditive / MC_MoveAdditive, MC_Jog)	Drehmoment-/ Kraftregelung (MC_TorqueControl)	Geschwindigkeits- Synchronisation (MC_GearIn)	Lage-Synchronisation (Winkel- Synchronisation, Elektronische Kurvenscheibe, MotionProfile: MC_CamIn, MB_GearInPos, MB_MotionProfile)
0x 0000 0001	-	-	-	-	-	-
0x 0000 0021	-	-	-	-	+	-
0x 0000 0002	0b00	-	+	+	-	-
	0b01	-	-	-	-	-
	0b10		+	-	-	-
	0b11	0	-	-	-	-
		>0	+	-	-	-
0x 0000 0022	0b00	-	+	+	+	+
	0b01	-	-	-	+	-
	0b10	-	+	-	+	+
	0b11	0	-	-	+	-
		>0	+	-	+	+

- Funktionsbaustein ist **nicht** lauffähig
 + Funktionsbaustein ist lauffähig
 1 P-0-2003, Auswahl der Funktionspakete
 2 P-0-0045, Stromregler-Steuerwort
 3 "P-0-0075, Gebertyp 2 (optionaler Geber)"; wird erst ab den folgenden Firmwaren ausgewertet: MPx06V14, MPx07V10, MPx17

Tab. 3-1: Abhängigkeiten zwischen Parametrierung der Achse, Funktionspaket und ausgewähltem Funktionsbaustein

3.3 Funktionsbausteine beim Firmware-Derivat PSB

Bei der Firmware PSB für Versorgungsgeräte sind alle Bibliotheken vorhanden. Jedoch können die Funktionsbausteine der Bibliothek "MX_PLCCopen", die speziell für Motion-Achsen vorhanden sind, nicht genutzt werden.

Beim Versuch, einen nicht unterstützten Funktionsbaustein zu kommandieren, wird der Error-Ausgang an diesem Funktionsbaustein gesetzt.

4 Signallaufverhalten der am häufigsten verwendeten Ein- und Ausgänge

Funktionsbausteine nach PLCopen verwenden Ein- und Ausgangssignale mit fest definierten Funktionen.

Eingangssignale "Execute" / "Enable"

Um die Bearbeitung eines Funktionsbausteins anzustoßen/ihn zu aktivieren, verfügt ein Funktionsbaustein entweder über einen Eingang "Execute", einen Eingang "ExecuteLock" oder einen Eingang "Enable". Alle drei Eingänge sind vom Typ BOOL.

Der Aktivierungseingang "Execute" wird an Funktionsbausteinen eingesetzt, die flankengesteuert arbeiten. Mit einer positiven Flanke von "Execute" werden die Eingangsvariablen erfasst und gleichzeitig eine einmalige Ausführung des Funktionsbausteins kommandiert. Der Eingang ist immer gekoppelt an die Ausgänge "Done" bzw. "InVelocity", "InTorque", "In_xxxx". Neue Eingangswerte werden erst über eine erneute positive Flanke von "Execute" wirksam. Eine noch nicht abgeschlossene und damit unterbrochene Kommandierung wird durch Setzen des Ausgangs "CommandAborted" angezeigt.



Bei flankengesteuerten Funktionsbausteinen mit "Execute"-Eingang ist Retriggerung möglich; d.h. erhält der Aktivierungseingang bei aktivem Funktionsbaustein eine erneute Flanke, dann wird der alte Auftrag verworfen, die Eingänge werden erneut übernommen und der neue Auftrag wird mit den neuen Werten fortgesetzt.

Der Aktivierungseingang "ExecuteLock" verhält sich wie der Eingang "Execute". Solange aber dieser Eingang TRUE ist, kann der Funktionsbaustein nicht unterbrochen werden. Ausnahme: Der Funktionsbaustein "MC_Stop" kann den Funktionsbaustein "MB_Stop" unterbrechen.

Der Aktivierungseingang "Enable" wird an Funktionsbausteinen eingesetzt, die zustandsgesteuert arbeiten. Mit der positiven Flanke von "Enable" werden die Eingangsvariablen übernommen. Neue Eingangswerte werden erst über eine erneute positive Flanke von "Enable" wirksam. Falls Eingangsvariablen zyklisch übertragen werden (z.B. bei Prozessregler-Bausteinen), ist dies explizit zu dokumentieren und am Funktionsbaustein-Eingang zu kennzeichnen.

Ausgangssignal "Active"

Bei entfernten Achsen wird mit "Active"=TRUE angezeigt, dass der Funktionsbaustein die Bearbeitungsanweisung angenommen hat (positive Flanke an "Execute", "ExecuteLock" bzw. "Enable").



Das "Active"-Signal kann zum Start des Funktionsbausteins verzögert sein, da evtl. nötige Vorverarbeitungen ausgenommen sind.

"Active" wird FALSE wenn "Done", "CommandAborted" oder "Error" TRUE werden. Bei zustandsgesteuerten Bausteinen wird "Active" mit löschen von "Enable" ebenfalls FALSE.



Bei **Parameter-Funktionsbausteinen** wird "Active" bei einer **Einzelachse** nicht gesetzt!

Ausgangssignal "Done"

Funktionsbausteine die einen administrativen Auftrag haben ("MC_Power", ...) oder Motion-Funktionsbausteine (wie "MC_MoveAbsolute",

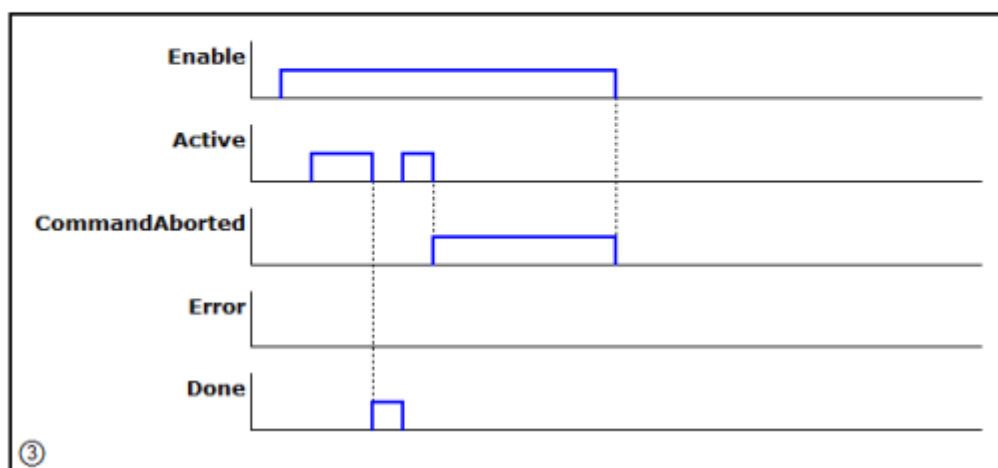
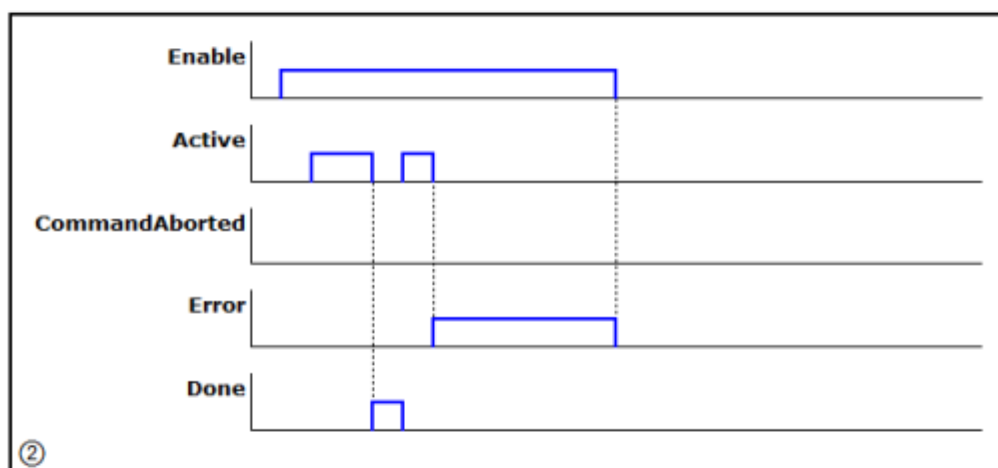
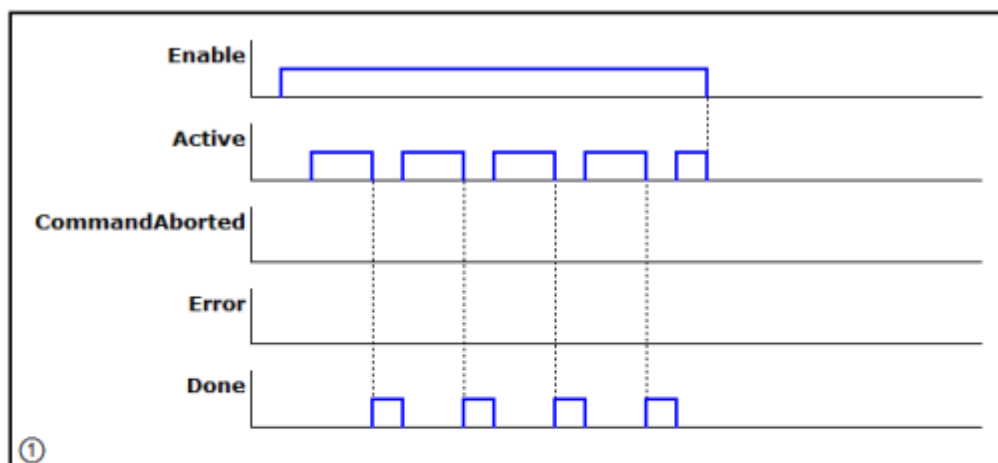
Signallaufverhalten der am häufigsten verwendeten Ein- und Ausgänge

"MC_MoveRelative", ...), die im Zustand (State) "Standstill" enden, zeigen über den "Done"-Ausgang die Beendigung ihres Auftrags an und sind nach ihrer Ausführung nicht mehr aktiv; mit anderen Worten: Dass ein Funktionsbaustein seinen Auftrag erfolgreich abgeschlossen hat und eventuell gelieferte Daten gültig sind, wird mit dem Ausgang "Done" signalisiert.

Bei flankengesteuerten Funktionsbausteinen bleibt "Done" für einen Zyklus TRUE, wenn zu dem Zeitpunkt zu dem "Done" auf TRUE gesetzt wurde, der Eingang "Execute"=FALSE war. Ist beim Setzen von "Done" der Eingang "Execute"=TRUE, so bleibt der "Done" solange TRUE bis "Execute" auf FALSE gesetzt wird.

Bei zustandsgesteuerten Funktionsbausteinen bleibt "Done", wenn "Enable"=TRUE ist, nach erfolgreicher Bearbeitung genau für einen Zyklus TRUE, bevor die Bearbeitung erneut beginnt.

Signallaufverhalten der am häufigsten verwendeten Ein- und Ausgänge

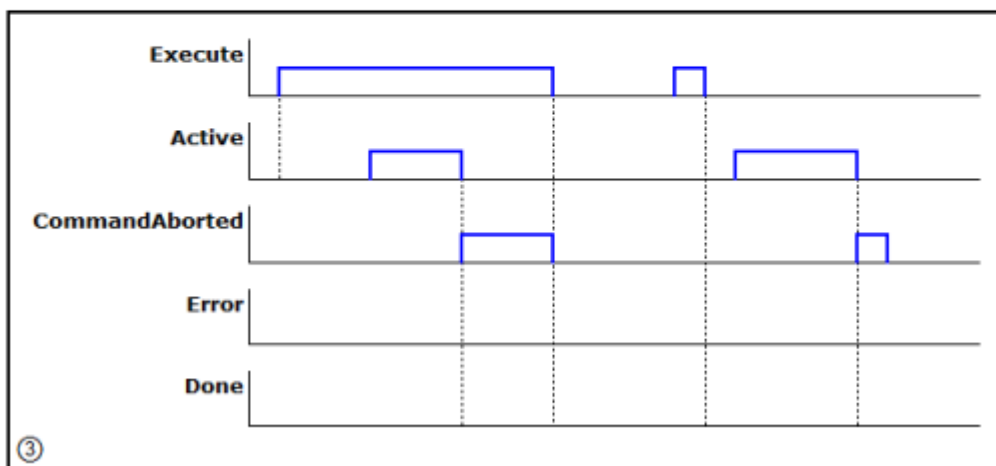
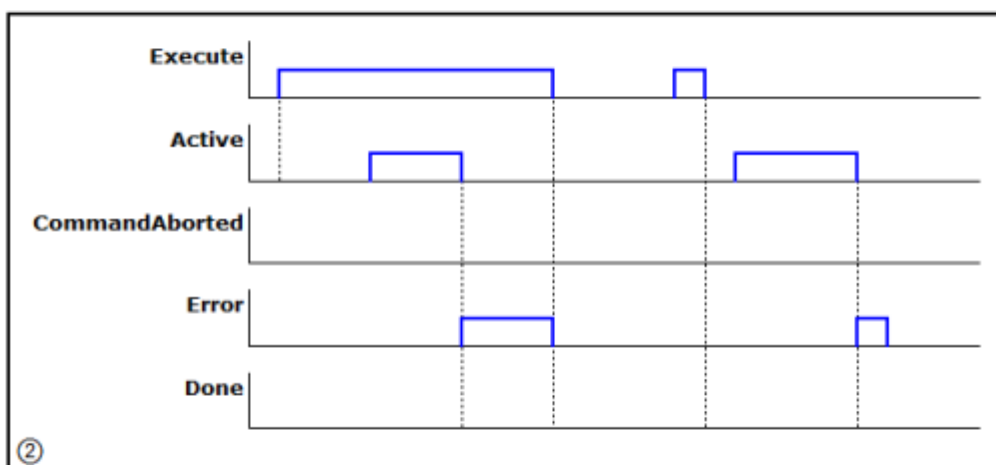
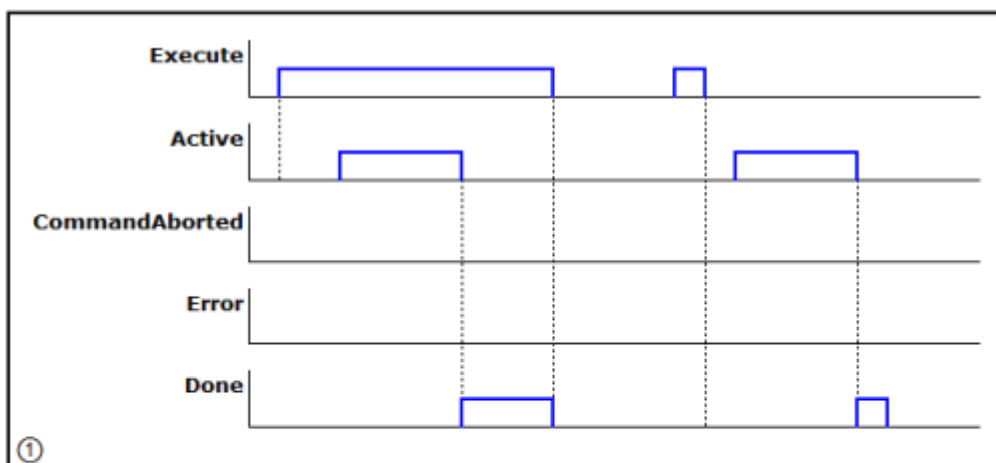


- ① Zustandsgesteuerte Bausteinbearbeitung erfolgreich abgeschlossen
- ② Zustandsgesteuerte Bausteinbearbeitung mit Fehler abgebrochen

Signallaufverhalten der am häufigsten verwendeten Ein- und Ausgänge

- ③ Zustandsgesteuerte Bausteinbearbeitung wurde unterbrochen
Abb. 4-1: Signal-Zeitverhalten von zustandsgesteuerten Funktionsbausteinen (mit Eingang "Enable"); bei Parameter-Funktionsbausteinen wird "Active" bei einer Einzelachse nicht gesetzt!

Signallaufverhalten der am häufigsten verwendeten Ein- und Ausgänge

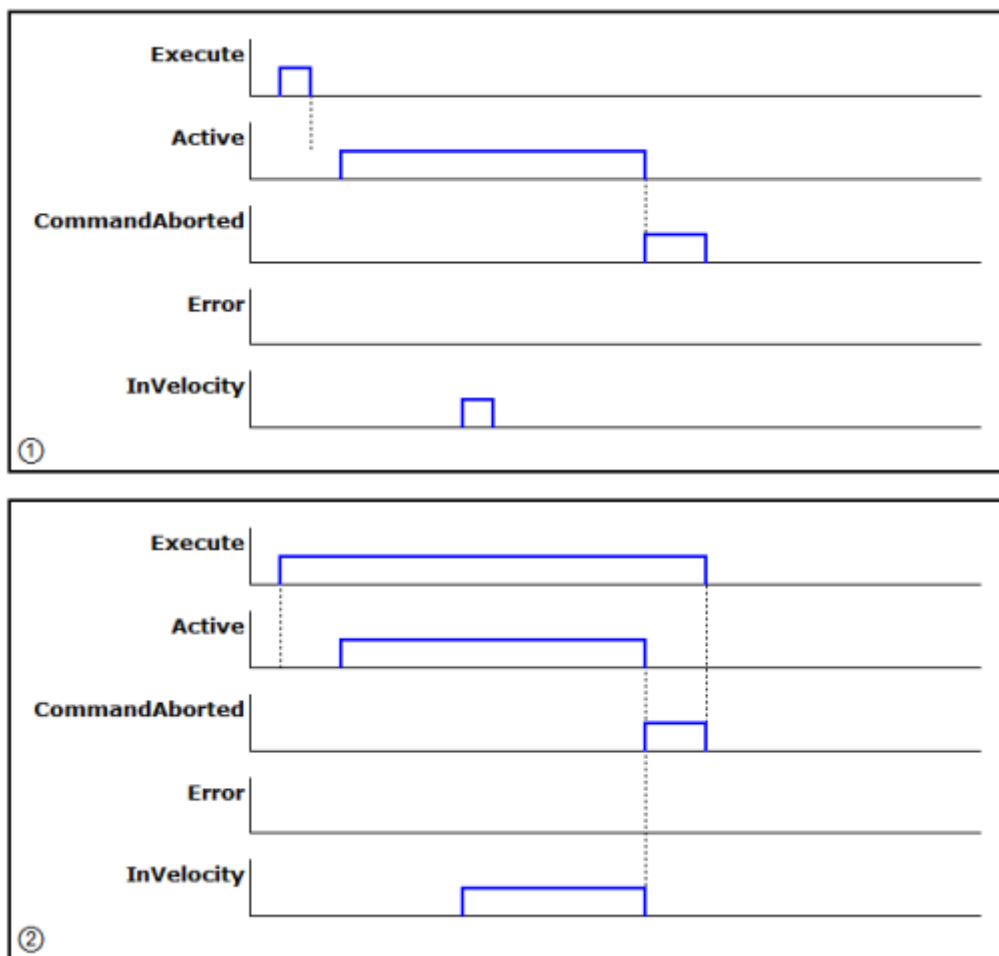


- ① Flankengesteuerte Bausteinbearbeitung erfolgreich abgeschlossen
- ② Flankengesteuerte Bausteinbearbeitung mit Fehler abgebrochen

Signallaufverhalten der am häufigsten verwendeten Ein- und Ausgänge

- ③ Flankengesteuerte Bausteinbearbeitung wurde unterbrochen
 Abb. 4-2: Signal-Zeitverhalten von flankengesteuerten Funktionsbausteinen (mit Eingang "Execute"); bei Parameter-Funktionsbausteinen wird "Active" bei einer Einzelachse nicht gesetzt!

Funktionsbausteine, die nach Erreichen ihres Auftrags ihre Bewegung- oder Kraft-Aktivität weiter fortsetzen (wie z.B. "MC_MoveVelocity", "MC_TorqueControl", "MB_MotionProfile") und nur durch den Aufruf eines "MC_Stop" oder eines anderen Funktionsbausteins abgelöst werden, signalisieren diesen Zustand indem sie "Active" bleiben; d.h. "Active" und z.B. "InVelocity" usw. sind gleichzeitig TRUE.



- ①+② Flankengesteuerte Bausteinbearbeitung wurde unterbrochen
 Abb. 4-3: Signal-Zeitverhalten von flankengesteuerten Funktionsbausteinen (mit Eingang "Execute"); "Active" in Kombination mit "InVelocity"

Ausgangssignal "CommandAborted"

Der Ausgang "CommandAborted" zeigt an, dass der Funktionsbaustein seine Aufgabe nicht beenden konnte, weil er durch einen anderen Motionbaustein unterbrochen wurde, oder die Achse aufgrund eines Fehlers abgeschaltet wurde.

Bei flankengesteuerten Funktionsbausteinen bleibt der Ausgang "CommandAborted" für einen Zyklus TRUE, wenn zu dem Zeitpunkt zu dem "CommandAborted" auf TRUE gesetzt wurde, der Eingang "Execute"=FALSE war. Ist beim Setzen von "CommandAborted" der Eingang "Execute"=TRUE,

Signallaufverhalten der am häufigsten verwendeten Ein- und Ausgänge

so bleibt "CommandAborted" solange TRUE bis "Execute" auf FALSE gesetzt wird.

Bei zustandsgesteuerten Funktionsbausteinen wird der Ausgang "CommandAborted" mit Zurücksetzen des "Enable"-Eingangs gelöscht.

5 Fehlerbehandlung

5.1 Allgemeines

Die Funktionsbausteine haben, je nach Zielsystem, zwei oder drei Ausgänge, die bei Auftreten eines Fehlers Informationen enthalten:

- "Error",
- "ErrorID" und
- "ErrorIdent"

Error Konnte die Bearbeitung eines Auftrages nicht erfolgreich beendet werden, wird dies durch den Ausgang "Error" signalisiert. Der "Error"-Ausgang wird mit den Eingängen "Execute"=FALSE oder "Enable"=FALSE zurückgesetzt; ein zusätzlicher Reset Eingang ist daher nicht notwendig.

Bei flankengesteuerten Funktionsbausteinen bleibt der Ausgang "Error" für einen Zyklus TRUE, wenn zu dem Zeitpunkt zu dem der Ausgang "Error" auf TRUE gesetzt wurde, der Eingang "Execute"=FALSE war. Ist beim Setzen von "Error" der Eingang "Execute"=TRUE, so bleibt der Ausgang "Error" solange TRUE, bis der Eingang "Execute" auf FALSE gesetzt wird.

Bei zustandsgesteuerten Funktionsbausteinen bleibt der Ausgang "Error" solange TRUE bis der Eingang "Enable" auf FALSE gesetzt wird.

ErrorID Der Ausgang "ErrorID" ist vom Typ ERROR_CODE. Liegt ein Fehler am Funktionsbaustein vor, kann an dem Ausgang "ErrorID" die Fehlerklassifizierung entnommen werden.



Die Ausgangswerte von "ErrorID" müssen bei den Zielsystemen unterschiedlich interpretiert werden!

Der Typ ERROR_CODE ist in der "CommonTypes"-Bibliothek des jeweiligen Zielsystems definiert.

ErrorIdent Der Ausgang "ErrorIdent" gibt die genaue Fehlerursache wieder.

"ErrorIdent" ist vom Typ ERROR_STRUCT. ERROR_STRUCT beinhaltet folgende Elemente:

- "Table",
- "Additional1" und
- "Additional2".



Die Struktur [ERROR_STRUCT](#) und deren Elemente ist in der Bibliothek "MX_CommonTypes" definiert.

5.2 Fehler-Referenzlisten



Beim Debuggen eines SPS-Programms können Sie in IndraWorks über **Debuggen** ► **Anzeigeformat** auswählen, ob binäre Datentypen (BYTE, WORD, DWORD) dezimal, hexadezimal oder binär dargestellt werden sollen. Aktiviert ist die Auswahl, vor der ein Haken erscheint.

MLD_TABLE An dem Ausgang "ErrorIdent" verweist das Element "Table" auf die Fehlertabelle "MLD_TABLE".



Wenn kein Fehler anliegt, hat das Element "Table" den Wert "NO_TABLE_USED".

"ErrorID"	"ErrorIdent", Element "Additional1" (hexadezimal)	"ErrorIdent", Element "Additional2"	Beschreibung
-	16#00000000	übergebene Achsnummer	Wenn kein Fehler anliegt, hat das Element "Table" den Wert "NO_TABLE_USED"
INPUT_INVALID_ERROR	16#00000001	übergebene Achsnummer	Angewählte Funktion für diese Achse nicht vorhanden
RESOURCE_ERROR	16#00000001	übergebene Achsnummer	Achsreferenz ungültig / undefinierte Achse. Angegebene Achse (auch Slaveachse) ist nicht vorhanden
ACCESS_ERROR	16#00000002	übergebene Achsnummer	Der verwendete Funktionsblock darf im System nur eine Instanz haben (z. B.: "MX_SynchronControl")
STATE_MACHINE_ERROR	16#00000003	übergebene Achsnummer	Antrieb befindet sich nicht im Betriebszustand "Ab" / Kommunikationsphase 4 - oder - Wenn die virtuelle Leitachse über einen "Motion"-Funktionsbaustein kommandiert werden soll: Die Leitachse ist nicht aktiviert. Aktivieren Sie die Leitachse, indem Sie Bit 0 in P-0-0917 (Steuerwort Leitachsgenerator) setzen oder / und P-0-0758 (Virtuelle Leitachse, Lageistwert) in P-0-0916 (Leitachsformatwandler Signalauswahl) eintragen
STATE_MACHINE_ERROR	16#00000004	übergebene Achsnummer	Antrieb befindet sich nicht im Betriebszustand "Ab"
ACCESS_ERROR	16#00000005	übergebene Achsnummer	MLD hat weder permanente (P-0-1367) noch temporäre Kontrolle über die angegebene (lokale) Achse
ACCESS_ERROR	16#00000006	übergebene Achsnummer	Motionkommando oder "MC_Stop": Antrieb ist momentanlos / nicht momentanbehaftet
STATE_MACHINE_ERROR	16#00000006	übergebene Achsnummer	"MC_Power": Antrieb ist nicht betriebsbereit. Antrieb wurde extern abgeschaltet
RESOURCE_ERROR	16#00000007	übergebene Achsnummer	Funktionsbaustein mit aktuellem Funktionspaket / aktueller Regelungsart nicht möglich (siehe Tabelle " Abhängigkeiten zwischen Parametrierung der Achse, Funktionspaket und ausgewähltem Funktionsbaustein "). - oder - Ein von einem Motion-Funktionsbaustein benötigter Parameter ist bereits zyklisch konfiguriert und daher schreibgeschützt.
INPUT_RANGE_ERROR	16#00000009	übergebene Achsnummer	Parametrierte Beschleunigung (Positionierbeschleunigung) ist "0" - oder - Die am Eingang "Acceleration" übergebene Beschleunigung ist "0"

Fehlerbehandlung

"ErrorID"	"ErrorIdent", Element "Additional1" (hexadezimal)	"ErrorIdent", Element "Additional2"	Beschreibung
INPUT_RANGE_ERROR	16#00000010	übergebene Achsnummer	Die am Eingang "Deceleration" übergebene Verzögerung ist "0"
STATE_MACHINE_ERROR	16#00000011	übergebene Achsnummer	Nach der Aktivierung mit "MB_PreSetMode" wurde ohne eine Motion-Kommandierung versucht mit "MC_Power" einzuschalten
STATE_MACHINE_ERROR	16#00000012	übergebene Achsnummer	Eine Instanz von "MB_Stop" versucht eine Achse zu beauftragen während eine Instanz von "MC_Stop" aktiv ist - oder - Die Achse befindet sich in "Stopping" und darf nicht kommandiert werden ("MB_Stop"/"MC_Stop" mit aktivem "Execute"-Eingang)
INPUT_RANGE_ERROR	16#00000013	übergebene Achsnummer	- Betriebsart Kurvenscheibe: Der Eingang "CamTableID" liegt nicht im gültigen Bereich (1 .. 8) - Betriebsart MotionProfile: - Gewählter Satz ungültig (Eingang "SetSelection" bzw. "SetNumber"). Gültiger Bereich: 0 .. 1 - oder - - Gewählter Schritt ungültig (Eingang "StepNumber"). Gültiger Bereich: 1 .. 8
STATE_MACHINE_ERROR	16#00000014	übergebene Achsnummer	MLD befindet sich in State "ErrorStop". Abhilfe: "MC_Reset" durchführen
STATE_MACHINE_ERROR	16#00000015	übergebene Achsnummer	Aktivierung des "MC_Power". Die Freigabe "DriveEnable" in Achssteuerwort (P-0-0116, Bit14) fehlt
ACCESS_ERROR	16#00000016	übergebene Achsnummer	MLD in Slaveachse hat permanente Kontrolle. Der MLD-M-Master kann den Slave daher nicht steuern
INPUT_RANGE_ERROR	16#00000017	übergebene Achsnummer	Parametrierte Beschleunigung ist ≤ 0
INPUT_RANGE_ERROR	16#00000018	übergebene Achsnummer	Parametrierte Verzögerung ist ≤ 0
INPUT_RANGE_ERROR	16#00000019	übergebene Achsnummer	Parametrierte Geschwindigkeit ist < 0
ACCESS_ERROR	16#00000020	Index der zweiten Task	Mehrere Tasks verwenden die selbe Instanz eines Funktionsbausteins, dies ist nicht erlaubt
STATE_MACHINE_ERROR	16#00000021	übergebene Achsnummer	Es wurde ein additives Motionkommando gestartet, das in diesem Achsstatus nicht erlaubt ist
ACCESS_ERROR	16#00000023	übergebene Achsnummer	Motionkommando konnte wegen Bufferüberlauf nicht verarbeitet werden
INPUT_INVALID_ERROR	16#00000024	übergebene Achsnummer	Gleiche Achsreferenz für Master- und Slaveachse. Von der Slaveachse wird gefordert sich selbst zu folgen; rekursive Leitachskombinationen werden nicht unterstützt.
OTHER_ERROR	16#00000199	übergebene Achsnummer	Interner Verarbeitungsfehler (undefinierter Zustand)
ACCESS_ERROR	16#00000200	Parameter-Identnummer	Interner Fehler: Undefiniertes Parameterdatenformat, Parameter kann nicht in FLOAT konvertiert werden
ACCESS_ERROR	16#00000201	Parameter-Identnummer	Mit diesem Funktionsbaustein kann kein ASCII-Parameter gelesen/geschrieben werden
ACCESS_ERROR	16#00000202	Parameter-Identnummer	Mit diesem Funktionsbaustein kann kein Listenparameter gelesen/geschrieben werden
INPUT_INVALID_ERROR	16#00000203	Parameter-Identnummer	Parameter-Identnummer gehört zu keinem Kommando
OTHER_ERROR	16#00000204	übergebene Achsnummer	Interner Verarbeitungsfehler (String ist zu kurz)

Fehlerbehandlung

"ErrorID"	"ErrorIdent", Element "Additional1" (hexadezimal)	"ErrorIdent", Element "Additional2"	Beschreibung
ACCESS_ERROR	16#00000205	Parameter- Identnummer	Die geforderte Anzahl an Bytes ist kleiner als die Istlänge der Liste
INPUT_INVALID_ERROR	16#00000206	übergebene Achsnnummer / übergebene Sercos Adresse	Ungültiger Pointer am Eingang des Funktionsbausteins [Adresse des angelegten Feldes liegt außerhalb des SPS-Bereichs oder liegt im Speicherbereich einer Funktion (globales Feld verwenden)]. Nur bei Bibliothek "RIL_SercosIII.lib": "Additional2" enthält die übergebene Sercos Adresse
ACCESS_ERROR	16#00000207	Parameter- Identnummer	Mit diesem Funktionsbaustein können nur ASCII-Parameter bearbeitet werden
CCD_ERROR	16#00000208	16#00000200	Zu viele gleichzeitige Lese-/Schreib-Aufträge auf entfernte Achsen
SYSTEM_ERROR	16#00000208	CCD- Fehlernummer	Interner CCD-Systemfehler; Fehlernummer siehe "Additional2"
SYSTEM_ERROR	16#00000209	CoSeMa- Fehlernummer	Interner CoSeMa-Systemfehler; Fehlernummer siehe "Additional2"
ACCESS_ERROR	16#00000210	99	Ausführung des Resetkommandos S-0-0099 fehlerhaft
ACCESS_ERROR	16#00000211	übergebene Achsnnummer	"MC_Reset" kann den Fehler nicht zurücksetzen (Timeout)
RESOURCE_ERROR	16#00000220	übergebene Achsnnummer	Kein externer Geber vorhanden
RESOURCE_ERROR	16#00000250	übergebene Achsnnummer	Führungskommunikation synchronisiert das Gerät
RESOURCE_ERROR	16#00000251	übergebene Achsnnummer	NC-Zykluszeit (S-0-0001) stimmt nicht mit der MLD-Taskzykluszeit überein
INPUT_RANGE_ERROR	16#00000252	übergebene Achsnnummer	Der Eingang "CmdDelay" ist größer als der Maximalwert
RESOURCE_ERROR	16#00000253	übergebene Achsnnummer	Den Echtzeitkanal darf nur eine MLD-Task benutzen
RESOURCE_ERROR	16#00000254	übergebene Achsnnummer	Zu viele Echtzeitvariablen verwendet. Es sind zur Zeit 4 RTcW_-Variablen und 4 RTcR_-Variablen erlaubt
CALCULATION_ERROR	16#00000255	übergebene Achsnnummer	Interner Fehler (Variablenoffset zu groß)
CALCULATION_ERROR	16#00000256	übergebene Achsnnummer	Interner Fehler (Variablenzugriff fehlerhaft)
RESOURCE_ERROR	16#00000257	übergebene Achsnnummer	Echtzeitkanal Istwerte aber keine Sollwerte definiert
ACCESS_ERROR	16#00000258	übergebene Achsnnummer	Synchrone Motiontask (CCD / FKM) vom Typ "extern ereignisgesteuert" unterstützt den lokalen Echtzeitkanal nicht
ACCESS_ERROR	16#00000259	übergebene Achsnnummer	Für die Task ist der lokale Echtzeitkanal nicht konfiguriert
ACCESS_ERROR	16#00000300	übergebene Achsnnummer	MLD hat bereits permanente Kontrolle. Eine zusätzliche Umschaltung auf temporäre Kontrolle ist nicht sinnvoll
ACCESS_ERROR	16#00000301	übergebene Achsnnummer	Temporäre Kontrolle ist mit virtuellem Slave nicht möglich
ACCESS_ERROR	16#00000302	übergebene Achsnnummer	Handbetrieb aktiv, keine temporäre Kontrolle möglich
STATE_MACHINE_ERROR	16#00000320	übergebene Achsnnummer	Führungskommunikation synchronisiert intern das Gerät / erlaubt den Phasenwechsel nicht (P-0-4086, Bit 0=0)

Fehlerbehandlung

"ErrorID"	"ErrorIdent", Element "Additional1" (hexadezimal)	"ErrorIdent", Element "Additional2"	Beschreibung
STATE_MACHINE_ERROR	16#00000321	übergebene Achsnummer	Antriebsinitialisierung ist noch nicht vollständig abgeschlossen (inkl. Erreichung der Zielphase) Bei Führungskommunikation mit Analog- bzw. Parallel-Interface muss die automatische Phasenumschaltung nach "P4" abgewartet werden. Tritt bei dieser ersten Phasenumschaltung ein Umschaltkommando-Fehler auf, muss 10 s gewartet werden; erst dann darf mit dem Funktionsbaustein "MX_SetDeviceMode" die Phase vorgegeben werden
ACCESS_ERROR	16#00000322	übergebene Achsnummer	Von anderer Stelle wurde ein Umschaltkommando gestartet bzw. gelöscht
DEVICE_ERROR	16#00000350	übergebene Achsnummer	Unbekannte Nebenbetriebsart ("NewOpMode">7 eingestellt)
RESOURCE_ERROR	16#00000355	übergebene Achsnummer	Die am Eingang "Master" angelegte Achse ist keine gültige Masterachse. Es werden nur Achsen als Masterachse unterstützt, die entweder in der Leitachs-Konfigurationsliste enthalten sind (P-0-1820.0.1) oder die in der Slaveachse lokal vorhanden sind (VmAxisInt, RmAxisInt und LinkAxisInt).
INPUT_RANGE_ERROR	16#00000370	falscher "CmdState"	Eingang "CmdState" ungültig (falscher enum-Wert)
INPUT_RANGE_ERROR	16#00000371	falscher "CmdState"	"CmdState" nicht erlaubt, z. B. "CmdState"="MX_CMD_NO_ACTIVE" während "P-0-1449, C4900 SPS-Kommando"="3"
INPUT_RANGE_ERROR	16#00000380	falsche Diagnosenummer	Eingang der Diagnosenummer ungültig (N<1 oder N>4)
ACCESS_ERROR	16#00000400	Parameter-Identnummer	Der angegebene Parameter kann mit diesem Funktionsbaustein nicht gelesen / beschrieben werden. Zum Beispiel Parameter vom Datentyp HEX oder BIN mit "MB_WriteRealParameter"
ACCESS_ERROR	16#00000401	Parameter-Identnummer	Es kann mit diesem Funktionsbaustein kein FLOAT-Parameter gelesen / beschrieben werden. SPS-Register (z.B. P-0-1370) können als FLOAT konfiguriert werden.
INPUT_RANGE_ERROR	16#00000402	übergebene Achsnummer / übergebene Sercos Adresse	Der angelegte Parameterwert passt nicht in das Datenformat des Parameters Nur bei Bibliothek "RIL_SercosIII.lib": "Additional2" enthält die übergebene Sercos Adresse
INPUT_INVALID_ERROR	16#00000403	übergebene Achsnummer	Bei dem Funktionsbaustein "MB_ReadListParameter" bzw. "MB_WriteListParameter" ist die an dem Eingang "ValueAdr" angelegte Adresse nicht WORD-gerecht (2Byte); d.h. die Adresse ist nicht durch 2 teilbar.
INPUT_INVALID_ERROR	16#00000404	übergebene Achsnummer	Bei dem Funktionsbaustein "MB_ReadListParameter" bzw. "MB_WriteListParameter" ist die an dem Eingang "ValueAdr" angelegte Adresse nicht DWORD-gerecht (4Byte); d.h. die Adresse ist nicht durch 4 teilbar.
ACCESS_ERROR	16#00000420	übergebene Achsnummer	Es wurde eine deaktivierte Achse angesprochen.
ACCESS_ERROR	16#00000422	übergebene Achsnummer	Der Servicekanal steht nicht mehr zur Verfügung (Phase < 2).
ACCESS_ERROR	16#00000423	übergebene Achsnummer	Doppelte Adressvergabe. Mindestens 2 Achsen haben die gleiche Sercos Adresse.
SYSTEM_ERROR	16#00000424	übergebene Achsnummer	Queue-Fehler. Es wurden zu viele Aufträge an eine Achse geschickt.

Fehlerbehandlung

"ErrorID"	"Errorident", Element "Additional1" (hexadezimal)	"Errorident", Element "Additional2"	Beschreibung
INPUT_RANGE_ERROR	16#00000500	übergebener Port	Es wurde ein Wert für "Port"<>"COM1" übergeben
ACCESS_ERROR	16#00000501	übergebener Port	COM1 ist bereits geöffnet

Tab. 5-1: Fehlertabelle MLD_TABLE

SERCOS_TABLE

Fehlercode		
hexadezimal	dezimal	Erläuterung
0x1001	4097	IDN nicht vorhanden
0x1009	4105	falscher Zugriff auf Element 1
0x2001	8193	Name nicht vorhanden
0x2002	8194	Name zu kurz übertragen
0x2003	8195	Name zu lang übertragen
0x2004	8196	Name nicht änderbar
0x2005	8197	Name zur Zeit schreibgeschützt
0x3002	12290	Attribut zu kurz übertragen
0x3003	12291	Attribut zu lang übertragen
0x3004	12292	Attribut nicht änderbar
0x3005	12293	Attribut zur Zeit schreibgeschützt
0x4001	16385	Einheit nicht vorhanden
0x4002	16386	Einheit zu kurz übertragen
0x4003	16387	Einheit zu lang übertragen
0x4004	16388	Einheit nicht änderbar
0x4005	16389	Einheit zur Zeit schreibgeschützt
0x5001	20481	minimaler Eingabewert nicht vorhanden
0x5002	20482	minimaler Eingabewert zu kurz übertragen
0x5003	20483	minimaler Eingabewert zu lang übertragen
0x5004	20484	minimaler Eingabewert nicht änderbar
0x5005	20485	minimaler Eingabewert zur Zeit schreibgeschützt
0x6001	24577	maximaler Eingabewert nicht vorhanden
0x6002	24578	maximaler Eingabewert zu kurz übertragen
0x6003	24579	maximaler Eingabewert zu lang übertragen
0x6004	24580	maximaler Eingabewert nicht änderbar
0x6005	24581	maximaler Eingabewert zur Zeit schreibgeschützt
0x7002	28674	Datum zu kurz übertragen
0x7003	28675	Datum zu lang übertragen

Fehlerbehandlung

Fehlercode		
hexadezimal	dezimal	Erläuterung
0x7004	28676	Datum nicht änderbar
0x7005	28677	Datum zur Zeit schreibgeschützt (Grund: Kommunikationsphase oder Modus)
0x7006	28678	Datum kleiner als min. Eingabewert
0x7007	28679	Datum größer als max. Eingabewert
0x7008	28680	Datum nicht korrekt (z. B. nicht unterstützte IDN, ungültige Bitnummer, ungültige Bitkombination, ungültige Listenlänge)
0x7009	28681	Datum passwortgeschützt
0x700A	28682	Datum zur Zeit schreibgeschützt, da es zyklisch konfiguriert ist (IDN ist im MDT bzw. AT konfiguriert, deshalb Schreiben über Servicekanal nicht erlaubt)
0x700B	28683	ungültiges Listenelement (IDN wird nicht unterstützt, Wert außerhalb der Eingabegrenzen)
0x700C	28684	Datum zurzeit schreibgeschützt aufgrund anderer Einstellungen (z. B. Parameter, Betriebsart, Antriebsfreigabe, Antrieb EIN usw.)
0x7010	28688	Kommando bereits aktiv
0x7011	28689	Kommando nicht unterbrechbar
0x7012	28690	Kommando zur Zeit nicht ausführbar (z. B. Kommando in dieser Phase nicht aktivierbar)
0x7013	28691	Kommando nicht ausführbar (ungültige oder falsche Parameter)

Tab. 5-2: Sercos Fehlercodes

INDRV_TABLE



"INDRV_TABLE" ist keine Tabelle im eigentlichen Sinn, sondern dient als Verweis auf die Dokumentation "Rexroth IndraDrive, Diagnosen"; dort gibt die Fehlernummer (Diagnosenummer) Aufschluss über Ursache und Abhilfe.

6 Datentypen

Datentyp	Bitgröße	Beschreibung
BOOL	mit Adressangabe (z. B. "AT %MX1.2"): 1	"BOOL" kann die Werte "TRUE" oder "FALSE" annehmen; ohne Adressangabe werden 8 Bit belegt
	ohne Adressangabe: 8	
BYTE	8	Byte-Wert 0 bis 255
WORD	16	Wort-Wert 0 bis 65535
DWORD	32	Doppelwort-Werte 0 bis $(2^{32})-1$
LWORD	64	Langwort-Werte 0 bis $(2^{64})-1$
INT	16	Ganzzahl (16 Bit) von -32768 bis +32767
DINT	32	Ganzzahl (32 Bit) von -2^{31} bis $(2^{31})-1$
SINT	8	Ganzzahl (8 Bit) von -128 bis +128
USINT	8	Ganzzahl (8 Bit) von 0 bis +255
UINT	16	Ganzzahl (16 Bit) von 0 bis +65535
UDINT	32	Ganzzahl (32 Bit) von 0 bis +4294967295
LINT	64	Ganzzahl (64 Bit) von -2^{63} bis $+2^{63}-1$
ULINT	64	Ganzzahl (64 Bit) von 0 bis $+2^{64}-1$
REAL	32	IEEE-Gleitpunktzahl
LREAL	64	IEEE-Gleitpunktzahl
TIME	32	IEC-Zeit in Intervallen von 1 ms
LTIME	64	Wie "TIME". Zusätzlich können "µs" (Mikrosekunden) und "ns" (Nanosekunden) verwendet werden
DATE	32	IEC-Datum in Intervallen von 1 Tag
DATE_AND_TIME, DT	32	Zeit in 1s-Raster: Jahr-Monat-Tag-Stunde:Minute:Sekunde
TIME_OF_DAY, TOD	32	Zeit in 1ms-Raster: Stunde (0-23), Min. (0-59), Sekunde (0-59), ms (0-999)
SYSTIME	64	Zeit in Millisekunden seit 01.01.1970 00:00:00

Tab. 6-1: Allgemeine Datentypen

7 Parameterstruktur

7.1 Aufbau

Parameter setzen sich aus den folgenden Elementen zusammen:

Element-Nummer	Beschreibung	Anforderung
1	Identnummer	notwendig
2	Name	optional
3	Attribut (siehe "Sercos Attribut")	notwendig
4	Einheit	optional
5	Minimalwert	optional
6	Maximalwert	optional
7	Datum	notwendig

Tab. 7-1: Parameteraufbau

7.2 Sercos Attribut

Die folgende Grafik / Tabelle gibt den Aufbau des Sercos Attributs wieder.

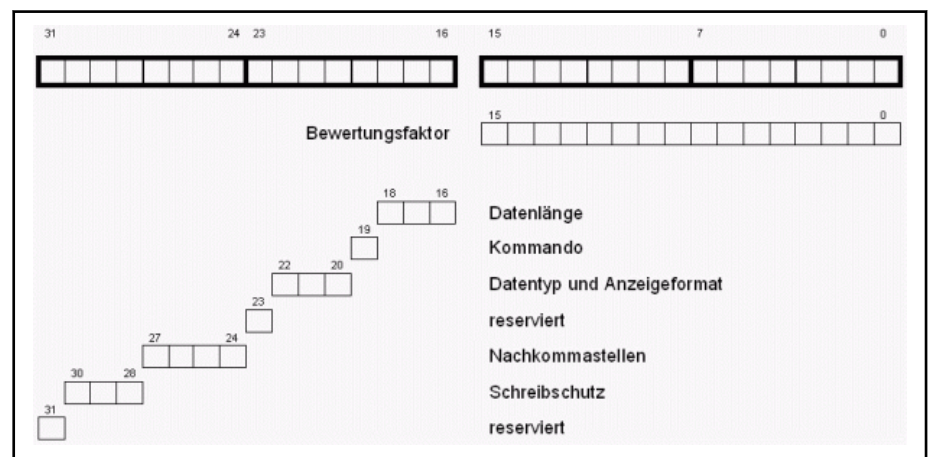


Abb. 7-1: Aufbau des Sercos Attributs

Binärstelle	Bezeichnung/Funktion
30	Schreibschutz des Parameter-Betriebsdatums im Betriebsmodus (Sercos Phase 4) 0: Betriebsdatum ist beschreibbar 1: Betriebsdatum ist schreibgeschützt
29	Schreibschutz des Parameter-Betriebsdatums (Sercos Phase 3) 0: Betriebsdatum ist beschreibbar 1: Betriebsdatum ist schreibgeschützt
28	Schreibschutz des Parameter-Betriebsdatums im Parametriermodus (Sercos Phase 2) 0: Betriebsdatum ist beschreibbar 1: Betriebsdatum ist schreibgeschützt

Parameterstruktur

Binärstelle	Bezeichnung/Funktion	
27...24	Anzahl der Nachkommastellen des Betriebsdatums 0000 (keine Nachkommastelle) bis 1111 (15 Nachkommastellen)	
23	reserviert	
22...20	Datentyp 000: Binärzahl 001: Vorzeichenlose ganze Zahl 010: Ganze Zahl 011: Vorzeichenlose ganze Zahl 100: Erweiterter Zeichensatz 101: Vorzeichenlose ganze Zahl 110: Gleitkommazahl 111: Vorzeichenlose ganze Zahl	Anzeigeformat binär Dezimalzahl ohne Vorzeichen Dezimalzahl mit Vorzeichen hexadezimal Text Parameter-Identnummerformat (Sercos Norm, 2 Byte) Dezimalzahl mit Vorzeichen Parameter-Identnummerformat (Sercos Norm, 4 Byte)
19	Funktion des Betriebsdatums 0: Betriebsdatum oder Parameter 1: Kommando	
18...16	Datenlänge des Betriebsdatums (Die Datenlänge des Betriebsdatums wird von der Steuerung für das ordnungsgemäße Beenden der Datenübertragung über den Servicekanal benötigt.) 000: reserviert 001: Betriebsdatum ist 2 Byte lang 010: Betriebsdatum ist 4 Byte lang 011: reserviert 100: Betriebsdatum hat eine variable Länge mit 1-Byte-Daten 101: Betriebsdatum hat eine variable Länge mit 2-Byte-Daten 110: Betriebsdatum hat eine variable Länge mit 4-Byte-Daten 111: reserviert	
15...0	Bewertungsfaktor Der Bewertungsfaktor ist eine vorzeichenlose ganze Zahl, die dazu dient, numerische Daten in ein Anzeigeformat umzuwandeln. Der Bewertungsfaktor ist auf den Wert von "1" gesetzt, wenn er für die Datenanzeige nicht verwendet wird.	

Tab. 7-2: Elemente des Sercos Attributs

8 IecSfc

Die Bibliothek "IecSfc.library" wird automatisch in SPS-Projekte eingebunden. Mit der Bibliothek wird die Funktion "SFCActionControl" bereitgestellt.

Um zu konfigurieren, in welcher Art eine Aktion, die einem IEC-Schritt zugewiesen ist, ausgeführt werden soll, können verschiedene Qualifizierer verwendet werden. Die Qualifizierer werden von der Funktion "SFCActionControl" verarbeitet.

Siehe auch Dokumentation "IndraLogic 2G, SPS-Programmiersystem", Kapitel "Qualifizierer für Aktionen in AS".

9 MX_Base

9.1 Einleitung

Mit der Bibliothek "MX_Base.library" wird für IndraMotion MLD folgendes zur Verfügung gestellt:

- Datentypen bzw. Strukturen
- Zyklische Parameter als Direktvariable (systemglobale Variable)
- Achsstrukturen [bei Mehrachssystem (MLD-Systemmodus)]

Die Struktur der Bibliothek "MX_Base.library" ist wie folgt:

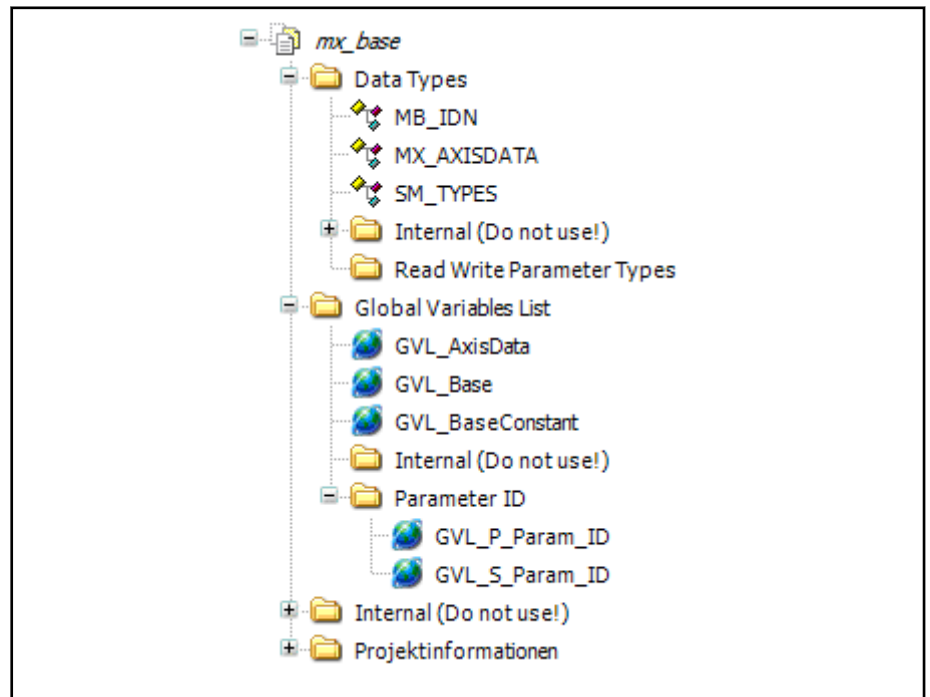


Abb. 9-1: Struktur der Bibliothek "MX_Base.library"

9.2 Datentypen

Struktur "Data Types"

- MB_IDN
Datentype, um Parameter adressieren zu können.
- MX_AXISDATA
Siehe IndraMotion MLD, Anwendungsbeschreibung: "Aufbau der AxisData"
- SM_TYPES
Siehe IndraMotion MLD, Anwendungsbeschreibung: "SM_TYPES"

MX_Base

9.3 Globale Variablenliste (Global Variables List)

GVL_AxisData

Name	Datentyp	Kommentar
AxisData	ARRAY [MB_MIN_AXIS_NUMBER..MB_MAX_AXIS_NUMBER] OF MX_AXISDATA	AxisData aller realen Achsen

GVL_Base Die globale Variablenliste "GVL_Base" enthält alle Achstypen, die in IndraMotion MLD deklariert werden können.

Name	Datentyp	Initialwert	Kommentar
NO_AXIS	AXIS_REF	STRUCT(CntrlNo := LOCAL_CNTRL, AxisNo := NO_OBJECT)	Definition für "keine Achse"
AXIS1		STRUCT(CntrlNo := LOCAL_CNTRL, AxisNo := AXIS_1)	Lokale Achse 1, genutzt als Achse, Slave oder Master (MLD-M)
AXIS2		STRUCT(CntrlNo := LOCAL_CNTRL, AxisNo := AXIS_2)	Entfernte Achse 2, genutzt als Achse, Slave oder Master (MLD-M)
...		...	...
AXIS10		STRUCT(CntrlNo := LOCAL_CNTRL, AxisNo := AXIS_10)	Entfernte Achse 10, genutzt als Achse, Slave oder Master (MLD-M)
VmAxis1		STRUCT(CntrlNo := LOCAL_CNTRL, AxisNo := VMA_1)	Virtuelle Achse 1, genutzt als Achse, Slave (intern) oder Master (MLD-M)
VmAxisInt		STRUCT(CntrlNo := LOCAL_CNTRL, AxisNo := VMA_INT)	Virtuelle Master-Achse, P-0-0761, genutzt als Master
RmAxis1		STRUCT(CntrlNo := LOCAL_CNTRL, AxisNo := RMA_1)	Reale Master-Achse 1, genutzt als Master (MLD-M)
RmAxisInt		STRUCT(CntrlNo := LOCAL_CNTRL, AxisNo := RMA_INT)	Reale Master-Achse, P-0-0052, genutzt als Master
LinkAxis1		STRUCT(CntrlNo := LOCAL_CNTRL, AxisNo := MA_LINK_1)	Externe Master-Achse 1, genutzt als Master (MLD-M)
LinkAxisInt		STRUCT(CntrlNo := LOCAL_CNTRL, AxisNo := MA_LINK_INT)	Externe Master-Achse, P-0-0053 oder P-0-0787 genutzt als Master
DV_Axis	ARRAY [AXIS_1..AXIS_1] OF MX_DirectVarAxis		Direktvariablen-Struktur für "alle" lokalen Achsen

GVL_BaseConstant Die globale Variablenliste "GVL_BaseConstant" enthält die systemabhängigen Konstanten in Abhängigkeit von der jeweiligen Hardware und der Firmware:

Name	Datentyp	Initialwert	Kommentar
IL_TARGET_TYPE	UINT	4	MLD
IL_TARGET_VERSION		20 ¹⁾	MLD*VRS
IL_SYSTEM_MLD	INT	1	System MLD
IL_TARGET_NAME	STRING	IndraMotion_MLD_E	IndraDrive
		IndraMotion_MLD_H	HydraulicDrive
MB_MIN_AXIS_NUMBER	INT	AXIS_1	Erste Achsnummer der MLD
MB_MAX_AXIS_NUMBER		VMA_1	Letzte Achsnummer der MLD (virtuelle Master-Achse)
MB_NUMBER_OF_AXIS		((MB_MAX_AXIS_NUMBER - MB_MIN_AXIS_NUMBER) + 1)	Maximale Anzahl der Achsen
MB_MAX_RECOMMENDED_AXIS			Empfohlene maximale Anzahl der Achsen
INT_MB_IDN_32BIT		1	Interne Information für "RIL_SercosIII.library"
MB_MIN_SERCOS_ADDRESS	INT	0	Erste Sercos Adresse (0==Master)
MB_MAX_SERCOS_ADDRESS		253	Letzte Sercos Adresse
MB_NUMBER_OF_SERCOS_DEVICES		((MB_MAX_SERCOS_ADDRESS - MB_MIN_SERCOS_ADDRESS) + 1)	Maximale Anzahl der Sercos Geräte

1) Der Wert hängt von der Firmware-Version ab; "20", wie hier gezeigt, steht für den Initialwert der Firmware-Version 20

Parameter ID

- **GVL_P_Param_ID**

Die globale Variablenliste "GVL_P_Param_ID" enthält Konstanten, um P-Parameter über die Identnummer adressieren zu können.

Beispiel: `FP_P_0_0001`, um den Parameter P-0-0001 (Schaltfrequenz der Leistungsstufe) adressieren zu können.

- **GVL_S_Param_ID**

Die globale Variablenliste "GVL_S_Param_ID" enthält Konstanten, um S-Parameter über die Identnummer adressieren zu können.

10 MX_CanL2

10.1 Einleitung

Die Bibliothek "MX_CanL2" enthält Funktionen, Funktionsbausteine und Datentypen für CAN-Kommunikation auf Level 2.

Funktionen und Funktionsbausteine

- MX_CANL2_FifoReceive
- MX_CANL2_FifoSend
- MX_CANL2_Init
- MX_CANL2_SetLED
- MX_fCANL2_MBoxReceive
- MX_fCANL2_MBoxSend
- MX_fCANL2_Status
- MX_fCANL2_MBoxRecStatus
- MX_fCANL2_MBoxSndStatus
- MX_fCANL2_MBoxUpdate
- MX_fCANtoHI
- MX_fCANtoHs
- MX_fHtoCANI
- MX_fHtoCANs

Datentypen

- MX_CANL2_LED_MODE
- MX_CANL2_MSG_INIT

10.2 MX_CANL2_FifoReceive

Mit Hilfe der Funktion "MX_CANL2_FifoReceive" kann ein CAN-Telegramm aus dem CAN-Fifo ausgelesen werden.



Abb. 10-1: Firmware-Funktion "MX_CANL2_FifoReceive"

Eingangs-Variablen	Datentyp	Beschreibung
Handle	DWORD	Handle von "MX_CANL2_Init"
CobId	POINTER TO UDINT	Pointer auf UDINT-Variable für COB-ID der Mailbox
BufLength	INT	Länge des Lesebuffers in Bytes
ReadData	POINTER TO BYTE	Adresse der Lesebuffers (typ. ARRAY[1..8] of BYTE)

Tab. 10-1: Eingangs-Variablen der Firmware-Funktion "MX_CANL2_FifoReceive"

MX_CanL2

Rückgabewert	Datentyp	Beschreibung
MX_CANL2_FifoReceive	DINT	Anzahl der gesendeten Bytes (0..8) - oder - Fehlernummer: -1: Ungültiger Handle -103: Länge "BufLength" zu klein -602: Kein neues Telegramm vorhanden

Tab. 10-2: Rückgabewert der Firmware-Funktion "MX_CANL2_FifoReceive"

10.3 MX_CANL2_FifoSend

Mit Hilfe der Funktion "MX_CANL2_FifoSend" kann ein CAN-Telegramm über den CAN-Fifo gesendet werden.



Abb. 10-2: Firmware-Funktion "MX_CANL2_FifoSend"

Eingangs-Variablen	Datentyp	Beschreibung
Handle	DWORD	Handle von "MX_CANL2_Init"
CobId	UDINT	COB-ID der Mailbox
Length	UDINT	Anzahl der zu sendenden Bytes
WriteData	POINTER TO BYTE	Adresse der Sendebuffers (typ. ARRAY[1..8] of BYTE)

Tab. 10-3: Eingangs-Variablen der Firmware-Funktion "MX_CANL2_FifoSend"

Rückgabewert	Datentyp	Beschreibung
MX_CANL2_FifoSend	DINT	Anzahl der gesendeten Bytes (0..8) - oder - Fehlernummer: -1: Ungültiger Handle -104: Ungültige Länge (Length) -601: Sende-Fifo ist voll

Tab. 10-4: Rückgabewert der Firmware-Funktion "MX_CANL2_FifoSend"

10.4 MX_CANL2_Init

Der Firmware-Funktionsbaustein "MX_CANL2_Init" dient zur Initialisierung und Telegrammkonfiguration einer CAN-Schnittstelle.

Die CAN-Schnittstelle wird damit einmalig initialisiert. Sie kann während der Buskommunikation nicht geändert werden, d. h. für eine Änderung muss die Init-Funktion erneut aufgerufen werden, wodurch der CAN-Controller deaktiviert und wieder aktiviert wird.

Zur Verfügung stehen 16 Sende- und 16 Empfangsmailboxen, die jeweils auf eine COB-ID eingestellt werden. Zusätzlich wird noch ein Sende-FIFO und ein Empfangs-FIFO für nicht konfigurierte Telegramme bereitgestellt. Es sind

MX_CanL2

momentan 2 Ports möglich. Port 1 adressiert die erste gefundene CAN-Karte, die nicht von der Führungskommunikation belegt ist, Port 2 die nächste CAN-Karte.

Beispiel: Beim Pitchumrichter ist der Port 1 auf dem Steckplatz X50.27 und Port 2 auf dem Steckplatz X50.32.

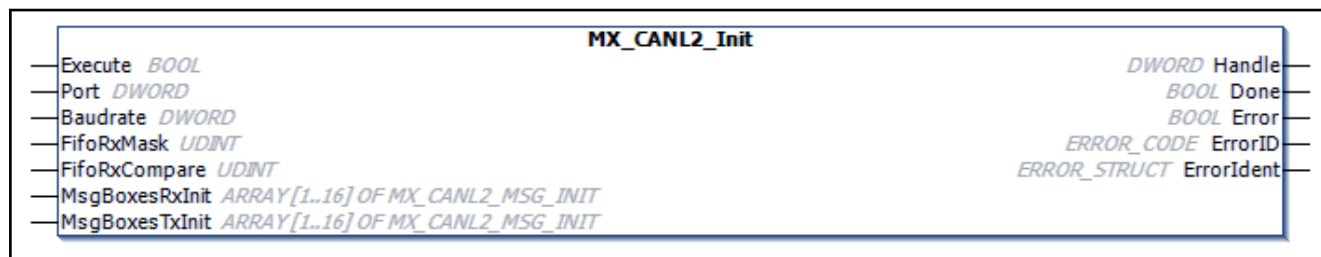


Abb. 10-3: Firmware-Funktionsbaustein "MX_CANL2_Init"

Eingangs-Variablen	Datentyp	Beschreibung
Execute	BOOL	siehe "Signallaufverhalten"
Port	DWORD	CAN-Port der Hardware 1: Erste von der FKM nicht genutzte CAN-Baugruppe 2: Zweite von der FKM nicht genutzte CAN-Baugruppe 3: Dritte von der FKM nicht genutzte CAN-Baugruppe ...
Baudrate	DWORD	Gewünschte CAN-Baudrate
FifoRxMask	UDINT	Maske für COB-ID-Vergleich* ¹
FifoRxCompare	UDINT	Wert für COB-ID-Vergleich* ¹

*¹ Beispiel: "Compare"=0000 1000 0011 0000,
 "Mask"=1111 1111 1111 0000 ⇒ Bereich zwischen 2096-2111
 (830h- 83Fh)

Tab. 10-5: Eingangs-Variablen des Firmware-Funktionsbausteins "MX_CANL2_Init"

Ausgangs-Variablen	Datentyp	Beschreibung
Handle	DWORD	Handle des Ports für weitere Zugriffe
Done	BOOL	Mit "Done"=TRUE zeigt der Funktionsbaustein an, dass die Bearbeitung der Arbeitsanweisung erfolgreich abgeschlossen ist
Error	BOOL	Mit "Error"=TRUE meldet der Funktionsbaustein, dass ein Fehler vorliegt
ErrorID	ERROR_CODE	"ErrorID" gibt eine grobe Fehlerbeschreibung wieder.
ErrorIdent	ERROR_STRUCT	"ErrorIdent" gibt über die Elemente: "Table", "Additional1" und "Additional2" die genaue Fehlerursache wieder.

Tab. 10-6: Ausgangs-Variablen des Firmware-Funktionsbausteins "MX_CANL2_Init"

"ErrorID"	"ErrorIdent", Element "Additional1"	"ErrorIdent", Element "Additional2"	Beschreibung
INPUT_INVALID_ERROR	-205	0	Der Zugriffsindex ist nicht definiert
INPUT_INVALID_ERROR	-10	0	Der Controller hat einen schwerwiegenden Fehler
INPUT_INVALID_ERROR	-101	0	Der Controller ist nicht im Konfigurationsmodus

MX_CanL2

"ErrorID"	"ErrorIdent", Element "Additional1"	"ErrorIdent", Element "Additional2"	Beschreibung
INPUT_INVALID_ERROR	-201	0	Eine COB-ID wird zweimal in der gleichen Richtung verwendet
INPUT_INVALID_ERROR	-202	0	Der Wert einer COB-ID ist ungültig

Tab. 10-7: *Fehlertabelle MLD_TABLE des Firmware-Funktionsbausteins "MX_CANL2_Init"*

Eingangs-/Ausgangs-Variable (VAR_IN_OUT)	Datentyp	Beschreibung
MsgBoxesRxInit	ARRAY[1..16] OF MX_CANL2_MSG_INIT	Satz von initialisierten Strukturen für den Empfang
MsgBoxesTxInit	ARRAY[1..16] OF MX_CANL2_MSG_INIT	Satz von initialisierten Strukturen für das Senden

Tab. 10-8: *Eingangs-/Ausgangs-Variablen des Firmware-Funktionsbausteins "MX_CANL2_Init"*

Datentyp "MX_CANL2_MSG_INIT" Der Datentyp "MX_CANL2_MSG_INIT" dient zur Initialisierung einer Mailbox der CAN-Schnittstelle.

Variablen	Datentyp	Beschreibung
udiCobId	UDINT	COB-ID der Mailbox
bInitialized	BOOL	TRUE: Mailbox-Parameter sind initialisiert worden
bNonOverwrite	BOOL	RX: TRUE: Weiterer Empfang wird blockiert bis Daten ausgelesen wurden TX: TRUE: Weiteres Senden wird blockiert bis Daten versendet wurden
bRtr	BOOL	Nur gültig für RX-Mailboxen FALSE: Sofort senden TRUE: Nur auf Anforderung senden
bEnable	BOOL	TRUE: Mailbox aktivieren

Tab. 10-9: *Variablen des Datentyps "MX_CANL2_MSG_INIT"*

10.5 MX_CANL2_SetLED

Die Funktion "MX_CANL2_SetLED" dient zur Steuerung der LED auf der CAN-Baugruppe.

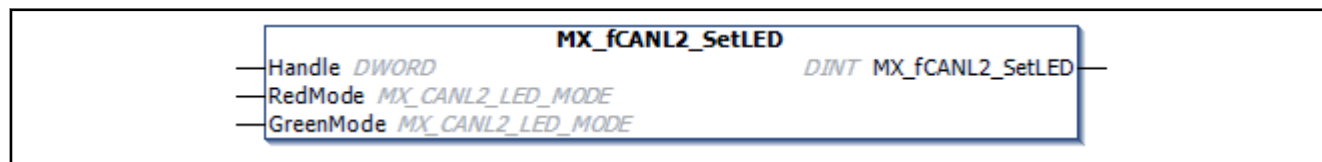


Abb. 10-4: *Firmware-Funktion "MX_CANL2_SetLED"*

Eingangs-Variablen	Datentyp	Beschreibung
Handle	DWORD	Handle von "MX_CANL2_Init"
RedMode	MX_CANL2_LED_MODE	Anzeigemodus der roten LED
GreenMode	MX_CANL2_LED_MODE	Anzeigemodus der grünen LED

Tab. 10-10: Eingangs-Variablen des Firmware-Funktionsbausteins "MX_CANL2_SetLED"

Rückgabewert	Datentyp	Beschreibung
MX_CANL2_SetLED	DINT	0: kein Fehler -1: Ungültiger Handle -301: Ungültiger LED-Modus

Tab. 10-11: Rückgabewert des Firmware-Funktionsbausteins "MX_CANL2_SetLED"

Datentyp "MX_CANL2_LED_MODE"

Der Datentyp "MX_CANL2_LED_MODE" dient zur Steuerung der LED auf der CAN-Baugruppe.

Werte	Beschreibung
LED_OFF	Aus
LED_FLICKERING	Flackern mit 10 Hz
LED_BLINKING	Blinken mit 2,5 Hz
LED_ON	An
LED_SINGLE_FLASH	Ein Puls mit 0,2 Sekunden und eine Sekunde Pause
LED_DOUBLE_FLASH	Zwei Pulse mit 0,2 Sekunden und eine Sekunde Pause
LED_TRIPLE_FLASH	Drei Pulse mit 0,2 Sekunden und eine Sekunde Pause
LED_QUAD_FLASH	Vier Pulse mit 0,2 Sekunden und eine Sekunde Pause

Tab. 10-12: Variablen des Datentyps "MX_CANL2_LED_MODE"

10.6 MX_fCANL2_MBoxReceive

Mit Hilfe der Funktion "MX_fCANL2_MBoxReceive" kann eine Mailbox aus dem CAN-Treiber ausgelesen werden.



Abb. 10-5: Firmware-Funktion "MX_fCANL2_MBoxReceive"

Eingangs-Variablen	Datentyp	Beschreibung
Handle	DWORD	Handle von "MX_CANL2_Init"
MsgBoxNum	INT	Nummer der Mailbox

MX_CanL2

Eingangs-Variablen	Datentyp	Beschreibung
Length	INT	Länge des Lesebuffers in Byte
ReadData	POINTER TO BYTE	Adresse der Lesebuffers (typ. ARRAY[1..8] of BYTE)

Tab. 10-13: Eingangs-Variablen der Firmware-Funktion "MX_fCANL2_MBoxReceive"

Rückgabewert	Datentyp	Beschreibung
MX_fCANL2_MBoxReceive	INT	Anzahl der gesendeten Bytes (0..8) - oder - Fehlernummer: -2 : Ungültiger Bereich des Zielspeichers -103: Der Zielpuffer ist zu klein -501: Ungültige Mailboxnummer -502: Mailbox ist nicht initialisiert -503: Mailbox ist nicht eingeschaltet -505: Kein neues Telegramm vorhanden

Tab. 10-14: Rückgabewert der Firmware-Funktion "MX_fCANL2_MBoxReceive"

10.7 MX_fCANL2_MBoxSend

Mit Hilfe der Funktion "MX_fCANL2_MBoxSend" kann ein CAN-Telegramm in eine Mailbox zum Senden eingestellt werden.

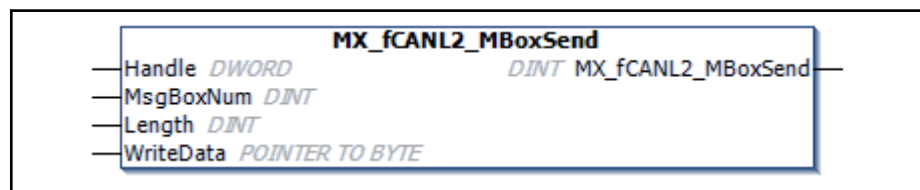


Abb. 10-6: Firmware-Funktion "MX_fCANL2_MBoxSend"

Eingangs-Variablen	Datentyp	Beschreibung
Handle	DWORD	Handle von "MX_CANL2_Init"
MsgBoxNum	INT	Nummer der Mailbox
Length	INT	Anzahl der zu sendenden Bytes
WriteData	POINTER TO BYTE	Adresse der Sendbuffers (typ. ARRAY[1..8] of BYTE)

Tab. 10-15: Eingangs-Variablen der Firmware-Funktion "MX_fCANL2_MBoxSend"

Rückgabewert	Datentyp	Beschreibung
MX_fCANL2_MBoxSend	INT	Anzahl der gesendeten Bytes (0..8) - oder - Fehlernummer: -1: Ungültiger Handle -501: Ungültige Mailboxnummer -502: Mailbox ist nicht initialisiert -503: Mailbox ist nicht eingeschaltet -504: Mailbox ist nicht frei (NonOverwrite-Flag eingeschaltet)

Tab. 10-16: Rückgabewert der Firmware-Funktion "MX_fCANL2_MBoxSend"

10.8 MX_fCANL2_Status

Mit Hilfe der Funktion "MX_fCANL2_Status" kann der Status des CAN-Treibers ausgelesen werden.

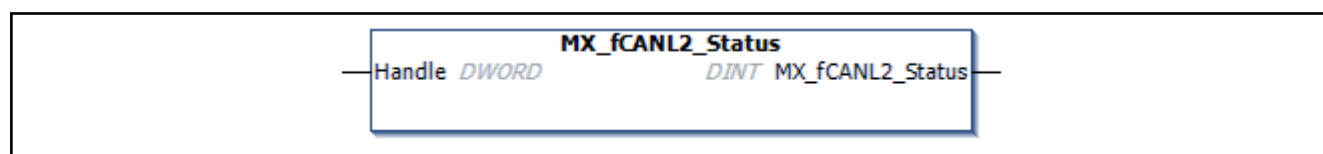


Abb. 10-7: Firmware-Funktion "MX_fCANL2_Status"

Eingangs-Variablen	Datentyp	Beschreibung
Handle	DWORD	Handle von "MX_CANL2_Init"

Tab. 10-17: Eingangs-Variablen der Firmware-Funktion "MX_fCANL2_Status"

Rückgabewert	Datentyp	Beschreibung
MX_fCANL2_Status	INT	0001: Status "Init" 0100: Status "Operational" 8001: Status "ErrorWarning" 8002: Status "Error" 8003: Status "BusOff" -1: Ungültiger Handle

Tab. 10-18: Rückgabewert der Firmware-Funktion "MX_fCANL2_Status"

10.9 MX_fCANL2_MBoxRecStatus

Mit Hilfe der Funktion "MX_fCANL2_MBoxRecStatus" kann der Status aller Empfangsmailboxen bitkodiert zurückgeliefert werden. Jede der 16 Mailboxen steuert ein Bit des Rückgabewerts an. Bei einem Fehler sind alle 32 Bits gesetzt.

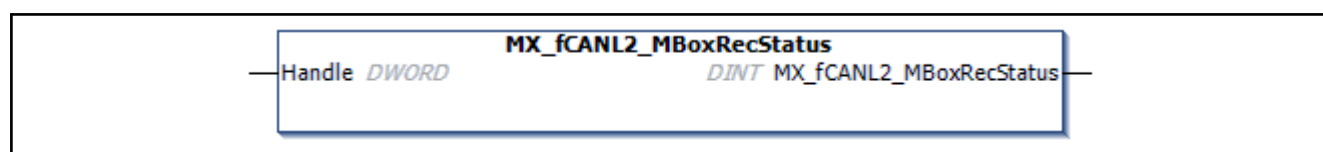


Abb. 10-8: Firmware-Funktion "MX_fCANL2_MBoxRecStatus"

MX_CanL2

Eingangs-Variablen	Datentyp	Beschreibung
Handle	DWORD	Handle von "MX_CANL2_Init"

Tab. 10-19: Eingangs-Variablen der Firmware-Funktion "MX_fCANL2_MBoxRecStatus"

Rückgabewert	Datentyp	Beschreibung
MX_fCANL2_MBoxRecStatus	DINT	Bits #0..#15 entsprechen den Status der Mailboxen 1..16. Statusbit 0: Mailbox ist leer Statusbit 1: Mailbox enthält Daten -1: Ungültiger Handle

Tab. 10-20: Rückgabewert der Firmware-Funktion "MX_fCANL2_MBoxRecStatus"

10.10 MX_fCANL2_MBoxSndStatus

Mit Hilfe der Funktion "MX_fCANL2_MBoxSndStatus" kann der Status aller Sendemailboxen bitkodiert zurückgeliefert werden. Jede der 16 Mailboxen steuert ein Bit des Rückgabewerts an. Bei einem Fehler sind alle 32 Bits gesetzt.



Abb. 10-9: Firmware-Funktion "MX_fCANL2_MBoxSndStatus"

Eingangs-Variablen	Datentyp	Beschreibung
Handle	DWORD	Handle von "MX_CANL2_Init"

Tab. 10-21: Eingangs-Variablen der Firmware-Funktion "MX_fCANL2_MBoxSndStatus"

Rückgabewert	Datentyp	Beschreibung
MX_fCANL2_MBoxSndStatus	DINT	Bits #0..#15 entsprechen den Status der Mailboxen 1..16. Statusbit 0: Mailbox ist leer / wurde gesendet Statusbit 1: Mailbox enthält noch Daten / wurde noch nicht gesendet -1: Ungültiger Handle

Tab. 10-22: Rückgabewert der Firmware-Funktion "MX_fCANL2_MBoxSndStatus"

10.11 MX_fCANL2_MBoxUpdate

Der Baustein wird benutzt,

- um in die Mailbox zu schreiben (Mailbox-Update)
- oder -
- um eine Remote-Anfrage zu senden (siehe Eingangs-Variable "Length").



Abb. 10-10: Firmware-Funktion "MX_fCANL2_MBoxUpdate"

Eingangs-Variablen	Datentyp	Beschreibung
Handle	DWORD	Handle von "MX_CANL2_Init"
MsgBoxNum	DINT	Nummer der Mailbox
Length	DINT	- Length > "0": Update der Sendemailbox im Controller; der Inhalt wird nicht aktiv versendet - Length = "0": Remote-Anfrage; die angefragte Mailbox sendet nach Empfang der Anfrage ihren Inhalt
WriteData	POINTER TO BYTE	Pointer auf Quell Speicher

Tab. 10-23: Eingangs-Variablen der Firmware-Funktion "MX_fCANL2_MBoxUpdate"

Rückgabewert	Datentyp	Beschreibung
MX_fCANL2_MBoxUpdate	DINT	Länge der geschriebenen Daten (0..8) - oder - Fehlernummer: -1: Schnittstellenfehler (ungültiger Handle) -501: Ungültige Mailboxnummer -502: Mailbox ist nicht initialisiert -503: Mailbox ist nicht eingeschaltet -504: Mailbox ist nicht frei (NonOverwrite-Flag eingeschaltet)

Tab. 10-24: Rückgabewert der Firmware-Funktion "MX_fCANL2_MBoxUpdate"

10.12 MX_fCANtoHI

Swap-Funktionen, um den Datenaustausch zwischen Host (MLD) und CAN zu vereinfachen. Sie kapseln die Endianes und dienen zur systemübergreifend kompatiblen Programmierung von Datenaustausch über CAN.

Die Funktion "MX_fCANtoHI" erhält einen 32-Bitwert in CAN-Bytereihenfolge und liefert einen 32-Bitwert in der Bytereihenfolge des Zielsystems (MLD) zurück.



Abb. 10-11: Firmware-Funktion "MX_fCANtoHI"

Eingangs-Variablen	Datentyp	Beschreibung
Can	DWORD	Datum aus CAN

Tab. 10-25: Eingangs-Variablen der Firmware-Funktion "MX_fCANtoHI"

MX_CanL2

Rückgabewert	Datentyp	Beschreibung
MX_fCANtoHI	DWORD	Datum für MLD

Tab. 10-26: Rückgabewert der Firmware-Funktion "MX_fCANtoHI"

10.13 MX_fCANtoHs

Swap-Funktionen, um den Datenaustausch zwischen Host (MLD) und CAN zu vereinfachen. Sie kapseln die Endianes und dienen zur systemübergreifend kompatiblen Programmierung von Datenaustausch über CAN.

Die Funktion "MX_fCANtoHs" erhält einen 16-Bitwert in CAN-Bytereihenfolge und liefert einen 16-Bitwert in der Bytereihenfolge des Zielsystems (MLD) zurück.



Abb. 10-12: Firmware-Funktion "MX_fCANtoHs"

Eingangs-Variablen	Datentyp	Beschreibung
Can	WORD	Datum aus CAN

Tab. 10-27: Eingangs-Variablen der Firmware-Funktion "MX_fCANtoHs"

Rückgabewert	Datentyp	Beschreibung
MX_fCANtoHs	WORD	Datum für MLD

Tab. 10-28: Rückgabewert der Firmware-Funktion "MX_fCANtoHs"

10.14 MX_fHtoCANI

Swap-Funktionen, um den Datenaustausch zwischen Host (MLD) und CAN zu vereinfachen. Sie kapseln die Endianes und dienen zur systemübergreifend kompatiblen Programmierung von Datenaustausch über CAN.

Die Funktion "MX_fHtoCANI" erhält einen 32-Bitwert in Host-Bytereihenfolge (MLD) und liefert einen 32-Bitwert in der Bytereihenfolge des Netzwerks CAN zurück.

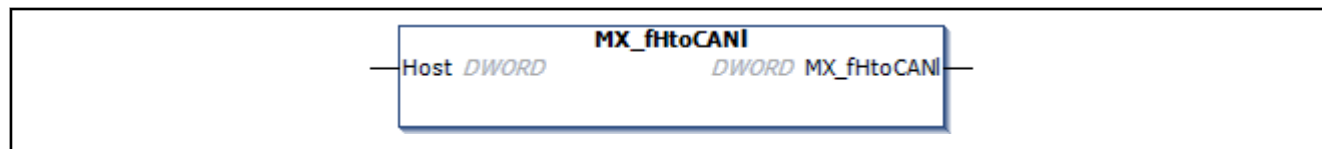


Abb. 10-13: Firmware-Funktion "MX_fHtoCANI"

Eingangs-Variablen	Datentyp	Beschreibung
Host	DWORD	Datum aus MLD

Tab. 10-29: Eingangs-Variablen der Firmware-Funktion "MX_fHtoCANI"

Rückgabewert	Datentyp	Beschreibung
MX_fHtoCANI	DWORD	Datum für CAN

Tab. 10-30: Rückgabewert der Firmware-Funktion "MX_fHtoCANI"

10.15 MX_fHtoCANs

Swap-Funktionen, um den Datenaustausch zwischen Host (MLD) und CAN zu vereinfachen. Sie kapseln die Endianes und dienen zur systemübergreifend kompatiblen Programmierung von Datenaustausch über CAN.

Die Funktion "MX_fHtoCANs" erhält einen 16-Bitwert in Host-Bytereihenfolge (MLD) und liefert einen 16-Bitwert in der Bytereihenfolge des Netzwerks CAN zurück.

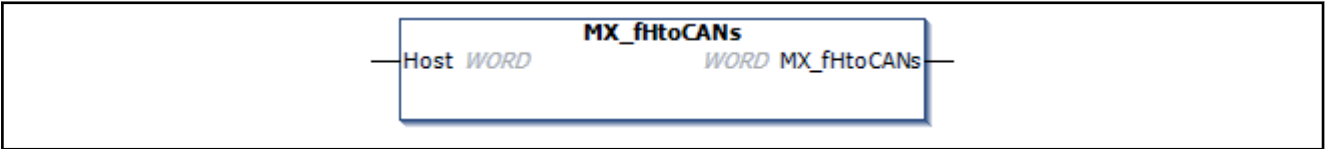


Abb. 10-14: Firmware-Funktion "MX_fHtoCANs"

Eingangs-Variablen	Datentyp	Beschreibung
Host	WORD	Datum aus MLD

Tab. 10-31: Eingangs-Variablen der Firmware-Funktion "MX_fHtoCANs"

Rückgabewert	Datentyp	Beschreibung
MX_fHtoCANs	WORD	Datum für CAN

Tab. 10-32: Rückgabewert der Firmware-Funktion "MX_fHtoCANs"

11 MX_CheckRtv



Die Funktionen der "MX_CheckRtv"-Bibliothek werden implizit vom SPS-Programm aufgerufen und dürfen nicht in ein Anwenderprogramm eingebaut werden!

In der Bibliothek "MX_CheckRtv" sind Funktionen zum Prüfen bzw. Melden von Laufzeitfehlern, wie z. B. Division durch Null, enthalten.

Wenn ein neues Projekt mit dem Zielsystem "IndraDrive" oder "HydraulicDrive" angelegt wird, wird die entsprechende Bibliothek automatisch eingebunden und ist sichtbar.

Wirkungsweise

Die Funktionen der "CheckRtv"-Bibliothek werden bei jeder entsprechenden Aktion (z. B. Division) aufgerufen und prüfen den Zugriff. Ist der Zugriff erlaubt, so wird das Programm ungehindert durchlaufen. Ist der Zugriff nicht erlaubt, wird ein Antriebsfehler (F6010) generiert und die SPS wird angehalten.

Siehe auch "Applikationshinweise" in der Anwendungsbeschreibung zu Rexroth IndraMotion MLD.

Funktionen

- **CheckBounds**
- **CheckDivReal**
- **CheckDivLReal**
- **CheckPointer**
- **CheckRangeSigned**
- **CheckRangeUnsigned**

12 MX_Debug

Die in der Bibliothek "MX_Debug" enthaltenen Funktionen werden für interne Testzwecke (Labor) benötigt.

WARNUNG

Sachschäden, Tod oder schwere Verletzung durch fehlerhafte Ansteuerung von Motoren bei Verwendung von Funktionen aus der Bibliothek "MX_Debug"!

Die Funktionen aus der Bibliothek "MX_Debug" werden für interne Testzwecke benötigt und dürfen vom Anwender nicht benutzt werden!

13 MX_PLCoPen

13.1 Einleitung

Mit der Bibliothek "MX_PLCoPen.library" werden folgende Funktionsbausteine/Funktionen zur Verfügung gestellt:

- Funktionen für Diagnose
- Funktionsbausteine/Funktionen zur Antriebsregelung
- Funktionsbausteine/Funktionen für Parameter
- Funktionen für Skalierung
- allgemeine Funktionen

13.2 "Drive control" - Funktionen für Diagnose

13.2.1 MX_fCheckCmdRequest

Die Firmware-Funktion "MX_fCheckCmdRequest" dient dazu, die Kommandovorgabe zurückzuliefern, die über den Antriebsparameter "P-0-1449, C4900 SPS-Kommando" vorgegeben wurde.

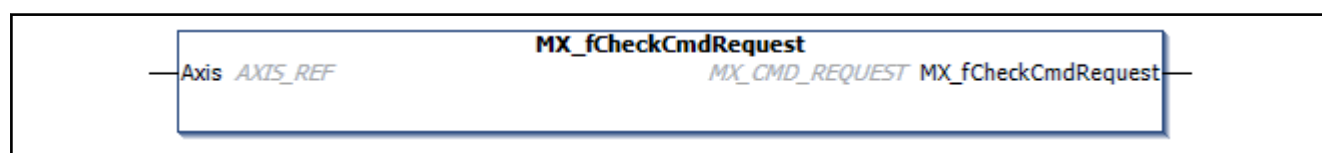


Abb. 13-1: Firmware-Funktion "MX_fCheckCmdRequest"

Eingangs-Variable	Datentyp	Beschreibung
Axis	AXIS_REF	Achsnnummer (hier kann nur die globale Variable "Axis1" übergeben werden)

Tab. 13-1: Eingangs-Variablen der Firmware-Funktion "MX_fCheckCmdRequest"

Rückgabewert	Datentyp	Beschreibung
MX_fCheckCmdRequest	MX_CMD_REQUEST	Die Funktion "MX_fCheckCmdRequest" kann folgenden Werte zurückgeben: • MX_CMD_REQ_ERROR: Es wurde eine ungültige Achsnnummer übergeben • MX_CMD_REQ_STOP: Kommando soll beendet werden • MX_CMD_REQ_BREAK: Kommando soll unterbrochen werden • MX_CMD_REQ_START: Kommando soll gestartet werden

Tab. 13-2: Rückgabewert der Firmware-Funktion "MX_fCheckCmdRequest"

13.2.2 MX_fGetDriveWarning

Mit der Firmware-Funktion "MX_fGetDriveWarning" lässt sich abfragen, ob eine bestimmte Warnungs-Diagnose im Antrieb aktiv ist.

In Abhängigkeit des Wertes am Eingang "WarningNo" (1 bis 7) werden die Warnungen E2011 ... E2017 geprüft.

MX_PLCoPen

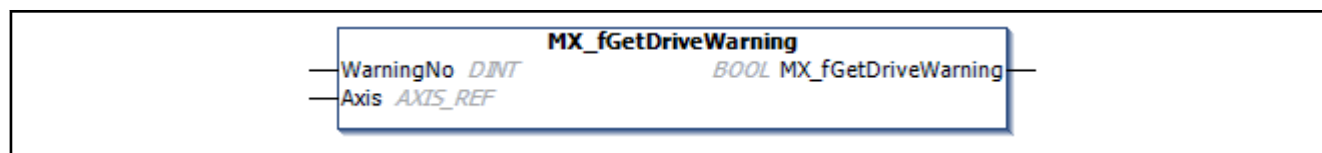


Abb. 13-2: Firmware-Funktion "MX_fGetDriveWarning"

Eingangs-Variable	Datentyp	Beschreibung
WarningNo	DINT	Ordnungsnummer der auszulesenden Warnung
Axis	AXIS_REF	Achsnummer (hier kann nur die globale Variable "Axis1" übergeben werden)

Tab. 13-3: Eingangs-Variablen der Firmware-Funktion "MX_fGetDriveWarning"

Rückgabewert	Datentyp	Beschreibung
MX_fGetDriveWarning	BOOL	Ist die zu prüfende Warnung im Antrieb aktiv, liefert die Funktion TRUE als Rückgabewert, andernfalls FALSE

Tab. 13-4: Rückgabewert der Firmware-Funktion "MX_fGetDriveWarning"

13.2.3 MX_fSetDriveError

Mit Hilfe der Firmware-Funktion "MX_fSetDriveError" kann im Antrieb eine Fehler-Diagnose ausgelöst werden.

In Abhängigkeit des Wertes am Eingang "ErrorNo" (1 bis 7) werden die Fehler F2011 ... F2017 im Antrieb ausgelöst.

Die Fehler-Diagnose kann durch Ausführen des Funktionsbausteins "MC_Reset" gelöscht werden.

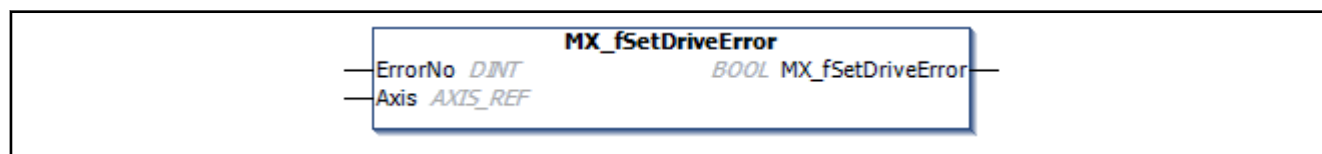


Abb. 13-3: Firmware-Funktion "MX_fSetDriveError"

Eingangs-Variable	Datentyp	Beschreibung
ErrorNo	DINT	Ordnungsnummer des auszulösenden Fehlers
Axis	AXIS_REF	Achsnummer (hier kann nur die globale Variable "Axis1" übergeben werden)

Tab. 13-5: Eingangs-Variablen der Firmware-Funktion "MX_fSetDriveError"

Rückgabewert	Datentyp	Beschreibung
MX_fSetDriveError	BOOL	Ist der Wert von "ErrorNo" im zugelassenen Bereich (1 bis 7), liefert die Funktion TRUE als Rückgabewert, andernfalls FALSE

Tab. 13-6: Rückgabewert der Firmware-Funktion "MX_fSetDriveError"

13.2.4 MX_fSetDriveErrorText

Mit der Firmware-Funktion "MX_fSetDriveErrorText" kann der Diagnose-Text der SPS-Fehler F2011 bis F2017 geändert werden.

Durch Aus- und Einschalten des Antriebs werden die SPS-Fehler wieder mit ihren Standard-Diagnosetexten (z. B. "SPS-Fehler Nr. 1") initialisiert.

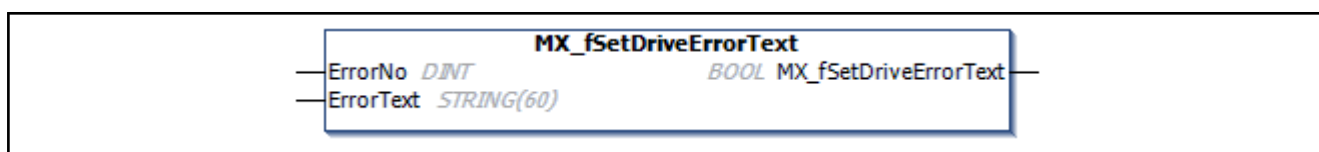


Abb. 13-4: Firmware-Funktion "MX_fSetDriveErrorText"

Eingangs-Variable	Datentyp	Beschreibung
ErrorNo	DINT	Ordnungsnummer des SPS-Fehlers, von dem der Diagnose-Text geändert werden soll: • "F2011 SPS-Fehler Nr. 1": "ErrorNo": "1" • "F2012 SPS-Fehler Nr. 2": "ErrorNo": "2" • ... • "F2017 SPS-Fehler Nr. 7": "ErrorNo": "7"
ErrorText	STRING(60)	Zu ändernder Diagnose-Text; maximale Länge: 60 Zeichen (inklusive Stringende "\0")

Tab. 13-7: Eingangs-Variable der Firmware-Funktion "MX_fSetDriveErrorText"

Rückgabewert	Datentyp	Beschreibung
MX_fSetDriveErrorText	BOOL	Wurde der Diagnose-Text erfolgreich geändert, liefert die Funktion TRUE als Rückgabewert, andernfalls FALSE

Tab. 13-8: Rückgabewert der Firmware-Funktion "MX_fSetDriveErrorText"

13.2.5 MX_fSetDriveWarning

Die Firmware-Funktion "MX_fSetDriveWarning" dient dazu, im Antrieb eine Warnungs-Diagnose zu setzen bzw. zu löschen.

In Abhängigkeit des Wertes am Eingang "WarningNo" (1 bis 7) können die Warnungen E2011 ... E2017 im Antrieb ausgelöst werden.

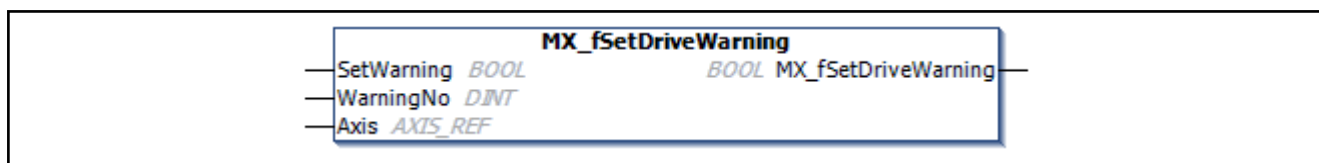


Abb. 13-5: Firmware-Funktion "MX_fSetDriveWarning"

Eingangs-Variable	Datentyp	Beschreibung
SetWarning	BOOL	TRUE: Warnung setzen, FALSE: Warnung löschen
WarningNo	DINT	Ordnungsnummer der auszulösenden Warnung
Axis	AXIS_REF	Achsnummer (hier kann nur die globale Variable "Axis1" übergeben werden)

Tab. 13-9: Eingangs-Variablen der Firmware-Funktion "MX_fSetDriveWarning"

Rückgabewert	Datentyp	Beschreibung
MX_fSetDriveWarning	BOOL	Ist der Wert von "WarningNo" im zugelassenen Bereich (1 bis 5), liefert die Funktion TRUE als Rückgabewert, andernfalls FALSE

Tab. 13-10: Rückgabewert der Firmware-Funktion "MX_fSetDriveWarning"

13.2.6 MX_fSetDriveWarningText

Mit der Firmware-Funktion "MX_fSetDriveWarningText" kann der Diagnose-Text der SPS-Warnungen E2011 bis E2017 geändert werden.

MX_PLCOpen

Durch Aus- und Einschalten des Antriebs werden die SPS-Warnungen wieder mit ihren Standard-Diagnosetexten (z. B. "SPS-Warnung Nr. 1") initialisiert.

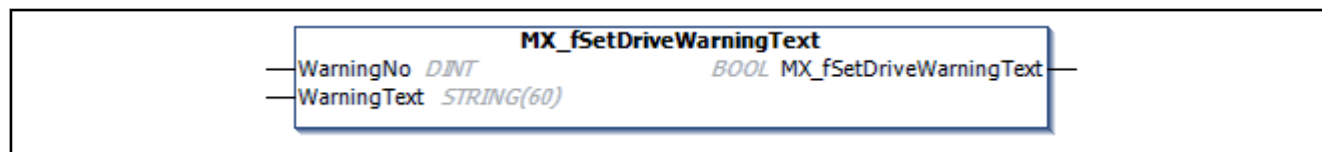


Abb. 13-6: Firmware-Funktion "MX_fSetDriveWarningText"

Eingangs-Variable	Datentyp	Beschreibung
WarningNo	DINT	Ordnungsnummer der SPS-Warnung, von dem der Diagnose-Text geändert werden soll: "E2011 SPS-Warnung Nr. 1": "WarningNo": "1" "E2012 SPS-Warnung Nr. 2": "WarningNo": "2" ... "E2017 SPS-Warnung Nr. 7": "WarningNo": "7"
ErrorText	STRING(60)	Zu ändernder Diagnose-Text; maximale Länge: 60 Zeichen (inklusive Stringende "\0")

Tab. 13-11: Eingangs-Variable der Firmware-Funktion "MX_fSetDriveWarning-Text"

Rückgabewert	Datentyp	Beschreibung
MX_fSetDriveWarningText	BOOL	Wurde der Diagnose-Text erfolgreich geändert, liefert die Funktion TRUE als Rückgabewert, andernfalls FALSE

Tab. 13-12: Rückgabewert der Firmware-Funktion "MX_fSetDriveWarningText"

13.2.7 MX_SetCmdState

Mit dem Firmware-Funktionsbaustein "MX_SetCmdState" kann der Kommandostatus von "P-0-1449, C4900 SPS-Kommando" gesetzt werden.

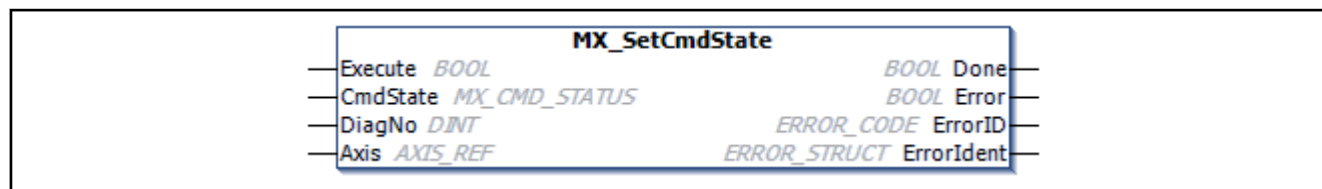


Abb. 13-7: Firmware-Funktion "MX_SetCmdState"

Eingangs-Variable	Datentyp	Beschreibung
Execute	BOOL	siehe " Signallaufverhalten "
CmdStatus	MX_CMD_STATUS	An dem Eingang "CmdStatus" können dem Funktionsbaustein die folgenden Werte übergeben werden: "MX_CMD_NO_ACTIVE": Kommando ist nicht gesetzt "MX_CMD_SETTING": Kommando ist gesetzt "MX_CMD_END_OK": Kommando ist fehlerfrei beendet "MX_CMD_IS_BREAK": Kommando ist unterbrochen "MX_CMD_IS_WORK": Kommando ist gesetzt und freigegeben "MX_CMD_BREAK_ERROR": Kommando ist mit Fehler abgebrochen

Eingangs-Variable	Datentyp	Beschreibung
DiagNo	DINT	Für den Fall "CmdStatus"==MX_CMD_BREAK_ERROR kann eine Kommandofehlernummer übergeben werden. 1: C4901 SPS Kommandofehler Nr. 1 2: C4902 SPS Kommandofehler Nr. 2 3: C4903 SPS Kommandofehler Nr. 3 4: C4904 SPS Kommandofehler Nr. 4
Axis	AXIS_REF	Achsnummer (hier kann nur die globale Variable "Axis1" übergeben werden)

Tab. 13-13: Eingangs-Variablen der Firmware-Funktion "MX_SetCmdState"

Ausgangs-Variable	Datentyp	Beschreibung
Done	BOOL	Mit "Done"=TRUE zeigt der Funktionsbaustein an, dass die Bearbeitung der Arbeitsanweisung erfolgreich abgeschlossen ist
Error	BOOL	Mit "Error"=TRUE meldet der Funktionsbaustein, dass ein Fehler vorliegt
ErrorID *	ERROR_CODE	"ErrorID" gibt eine grobe Fehlerbeschreibung wieder
ErrorIdent *	ERROR_STRUCT	"ErrorIdent" gibt über die Elemente: Table, Additional1 und Additional2 die genaue Fehlerursache wieder

* siehe nachfolgender Hinweis

Tab. 13-14: Ausgangs-Variablen des Firmware-Funktionsbausteins "MX_SetCmdState"



HINWEIS: Eine Auflistung aller möglichen Fehler finden Sie hier: ["Fehler-Referenzlisten"](#)

13.3 "Motion" - Funktionsbausteine/Funktionen zur Antriebsregelung

13.3.1 Allgemeine Eigenschaften

Die Funktionsbausteine/Funktionen zur Antriebsregelung haben die folgenden allgemeinen Eigenschaften:

- Positionier-Funktionsbausteine haben keinen Eingang für den Ruck ("Jerk")
- Funktionsbausteine haben "AXIS_REF" als "VAR_IN_OUT"
- Positionen bzw. Distanzen werden als REAL (Fließkomma) übergeben

13.3.2 MB_ChangeCamData

Kurzbeschreibung

Der Funktionsbaustein "MB_ChangeCamData" unterstützt das Laden von Kurvenscheiben-Profilen in die dafür vorgesehenen Antriebsparameter. Antriebe der Produktfamilie "Rexroth IndraDrive" unterstützen derzeit 8 Kurvenscheiben. 4 Kurvenscheiben mit maximal 1024 Kurvenscheiben-Elementen und 4 Kurvenscheiben mit maximal 128 Kurvenscheiben-Elementen.

Der Funktionsbaustein transportiert die durch den Eingang "NumberOfElements" spezifizierte Anzahl 4-Byte-Daten, beginnend ab dem Ort im SPS-Speicher auf den der "CamTableAdr"-Pointer zeigt, auf den über mit "CamTableID" ausgewählten Kurvenscheiben-Parameter.

MX_PLCOpen

Folgende Tabelle zeigt die Zuordnung zwischen dem Eingang "CamTableID", dem zugehörigen Antriebsparameter und die mögliche Anzahl Kurvenscheibenelemente für den Eingang "NumberOfElements".

CamTableID	Antriebs - Parameter	NumberOfElements
CAM_TABLE_1	P-0-0072, Kurventabelle 1	0 oder 3 bis 1024
CAM_TABLE_2	P-0-0092, Kurventabelle 2	0 oder 3 bis 1024
CAM_TABLE_3	P-0-0780, Kurventabelle 3	0 oder 3 bis 1024
CAM_TABLE_4	P-0-0781, Kurventabelle 4	0 oder 3 bis 1024
CAM_TABLE_5	P-0-0782, Kurventabelle 5	0 oder 3 bis 128
CAM_TABLE_6	P-0-0783, Kurventabelle 6	0 oder 3 bis 128
CAM_TABLE_7	P-0-0784, Kurventabelle 7	0 oder 3 bis 128
CAM_TABLE_8	P-0-0785, Kurventabelle 8	0 oder 3 bis 128

Tab. 13-15: Zuordnung zwischen "CamTableID", Antriebsparameter und "NumberOfElements"

Die zu übertragenden Einzeldaten dürfen in einem Bereich von -799.999.999 bis +799.999.999 liegen. Im Antrieb werden die zu übertragenden Einzeldaten in einen Prozent-Wert mit 6 Nachkommastellen umgewandelt; d.h. der zu übertragende Wert wird durch 1.000.000 dividiert.



Der Funktionsbaustein führt keine inhaltliche Prüfung der zu übertragenden Daten durch, so dass der Anwender für die korrekte Datenmenge und den richtigen Datentyp (DINT) der zu übertragenden Information zu sorgen hat.

Die benötigte Zeit für die Übertragung von Kurvenscheiben-Profilen hängt von verschiedenen Faktoren ab. Zu den Faktoren zählen

- Ort der Zielachse (lokale Achse oder entfernte CCD -Slaveachse)
- verwendetes Steuerteil,
- Anzahl der zu übertragenden Elemente,
- Zykluszeit der SPS-Task und die
- aktive CCD-Zykluszeit

Typische benötigten Zeiten um, **1024** Kurvenscheibenelemente zu übertragen sind:

Steuerteil	lokale Achse	entfernte CCD-Slaveachse
CSH1.1	6 - 15 ms	1024 * CCD-Zykluszeit + 250ms
CSB01.1 / CDB01.1	30 - 45 ms	1024 * CCD-Zykluszeit + 250ms

Tab. 13-16: Typische benötigten Zeiten um, 1024 Kurvenscheibenelemente zu übertragen



Die aufgeführten Zeiten sind **keine** absoluten Zeitangaben.



Wird das Kurvenscheibenprofil auf eine lokale Achse übertragen, dann läuft das SPS-Programm erst weiter, wenn der Funktionsbaustein "MB_ChangeCamData" mit "Done"=TRUE signalisiert, dass er die Daten erfolgreich geschrieben hat, oder mit "Error"=TRUE signalisiert, dass ein Fehler aufgetreten ist.

Anwendungshinweise:

Wenn der Funktionsbaustein "MB_ChangeCamData" zum Schreiben auf eine lokale Achse eingesetzt wird, sollte eine freilaufende SPS-Task oder eine zyklische SPS-Task mit großer Zykluszeit verwendet werden, um bei aktivierter Watchdog-Überwachung keinen Watchdog (F6010) zu generieren.



Ein Kurvenscheibenprofil wird im Antrieb mittels linearer oder kubischer Spline-Interpolation verfahren.

Für die kubische Spline-Interpolation werden beim Schreiben eines Kurvenscheibenprofils Koeffizienten berechnet, was dazu führt, dass es im Vergleich zu der linearen Kurvenscheiben-Interpolation ca. 5 ms länger dauert, um ein Kurvenscheibenprofil mit 1024 Elementen zu übertragen.

Die lineare Kurvenscheiben-Interpolation wird über "P-0-0086, Konfigurationswort Synchronbetriebsarten", Bit 7="1" aktiviert.



Antriebe der Produktfamilie "Rexroth IndraDrive" können zweierlei Kurvenscheibenformate verarbeiten (siehe "P-0-0072, Kurventabelle 1").

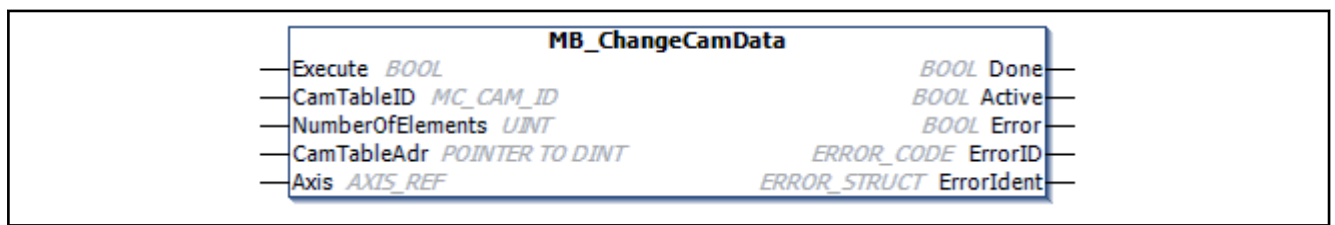


Abb. 13-8: Funktionsbaustein "MB_ChangeCamData"

Eingangs-Variable	Datentyp	Beschreibung
Execute	BOOL	siehe "Signallaufverhalten"
CamTableID	MC_CAM_ID	Nummer der zu beschreibenden Kurventabelle (1..8 entsprechend den Parametern P-0-0072, P-0-0092, P-0-0780..P-0-0785)
NumberOfElements	UINT	Anzahl der Stützpunkte, welche in die spezifizierte Kurventabelle übertragen werden sollen
CamTableAdr	POINTER TO DINT	Pointer auf SPS-Speicherbereich, der die zu transportierenden Daten enthält. Beispiel: CamTableAdr := ADR(arFeld1)

Tab. 13-17: Eingangs-Variablen des Firmware-Funktionsbausteins "MB_ChangeCamData"

MX_PLCOpen

Ausgangs-Variable	Datentyp	Beschreibung
Done	BOOL	Mit "Done"=TRUE zeigt der Baustein an, dass die Kurvenscheibendaten übertragen worden sind
Active	BOOL	Mit "Active"=TRUE angezeigt, dass der Funktionsbaustein die Bearbeitungsanweisung angenommen hat.
Error	BOOL	Mit "Error"=TRUE meldet der Funktionsbaustein, dass ein Fehler vorliegt
ErrorID *	ERROR_CODE	"ErrorID" gibt eine grobe Fehlerbeschreibung wieder
ErrorIdent *	ERROR_STRUCT	"ErrorIdent" gibt über die Elemente: Table, Additional1 und Additional2 die genaue Fehlerursache wieder

* siehe nachfolgender Hinweis

Tab. 13-18: *Ausgangs-Variablen des Firmware-Funktionsbausteins "MB_ChangeCamData"*



HINWEIS: Eine Auflistung aller möglichen Fehler finden Sie hier: ["Fehler-Referenzlisten"](#)

Eingangs-/Ausgangs-Variable (VAR_IN_OUT)	Datentyp	Beschreibung
Axis	AXIS_REF	Achsnummer ; hier können die folgenden globalen Variablen übergeben werden: "Axis1" (Achse bei MLD-S) "Axis1" (lokale Achse oder einzige Achse bei MLD-M), "Axis2" bis "Axis8" (erster bzw. letzter Slave in der Reihenfolge der Liste "P-0-4031, Geräteadressen-Übersicht")

Tab. 13-19: *Eingangs-/Ausgangs-Variable des Firmware-Funktionsbausteins "MB_ChangeCamData"*

13.3.3 MB_ChangeProfileSet

Mit Hilfe des Firmware-Funktionsbausteins "MB_ChangeProfileSet" kann der Satz eines Bewegungsprofils (MotionProfile) umgeschaltet werden.

Bei einer positiven Flanke an "Execute" werden die Daten übernommen, die den Satz eines Bewegungsprofils beschreiben. Dies sind

- die Anzahl der Bewegungsschritte,
- die Leitachsgeschwindigkeit, sowie die Listen für
 - Leitachs-Anfangsposition,
 - Hub,
 - Folgeachs-Geschwindigkeit und
 - Modus.

Für die Listen muss jeweils die Anfangsadresse eines Arrays angegeben werden, in dem sich die Daten befinden.



Die Anzahl der Elemente in den 4 Listen muss jeweils mindestens so groß sein wie die angegebene Anzahl der Bewegungsschritte.

Welcher Satz verändert wird, wird durch "SetNumber" festgelegt.

Die Einheiten der Daten, die in den Listen für "Distance" und "SlaveAxisVelocity" enthalten sind, sind durch die Wichtungparameter der Achse festgelegt (S-0-0076, S-0-0044).

Der Ausgang "Done" wird dann gesetzt, wenn der Satz komplett übertragen und das veränderte Profil überprüft worden ist. Der Ausgang "CheckOk" wird in Abhängigkeit vom Ergebnis der Prüfung gesetzt. Ist "CheckOk"=TRUE wurde keine Unstimmigkeit festgestellt.

Die mit "MB_ChangeProfileSet" übertragenen Daten werden nicht resident gespeichert. Der Baustein kann auf der lokalen Achse "Axis1" und auf entfernten Achsen "Axis2", ... arbeiten.

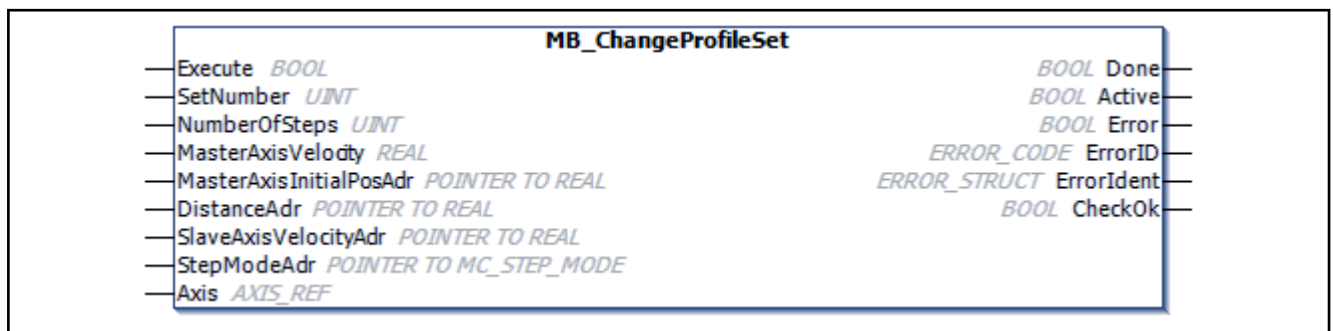


Abb. 13-9: Firmware-Funktionsbaustein "MB_ChangeProfileSet"

Input-Variable	Datentyp	Beschreibung
Execute	BOOL	siehe "Signallaufverhalten"
SetNumber	UINT	Auswahl des Bewegungsprofil-Satzes 0: Bewegungsprofil-Satz 0 (Parameter P-0-0703 bis P-0-0708) 1: Bewegungsprofil-Satz 1 (Parameter P-0-0710 bis P-0-0715)
NumberOfSteps	UINT	Anzahl der Bewegungsschritte Jeder Bewegungsprofil-Satz kann aus bis zu 8 Bewegungsschritten bestehen - Parameter P-0-0703 für Satz 0 - Parameter P-0-0710 für Satz 1
MasterAxisVelocity	REAL	Leitachsgeschwindigkeit Geschwindigkeit der Leitachse am Ausgang des elektronischen Getriebes in 1/min - Parameter P-0-0704 für Satz 0 - Parameter P-0-0711 für Satz 1
MasterAxisInitialPosAdr	POINTER TO REAL	Pointer auf ein Array mit den Leitachs-Anfangspositionen der Schritte Angabe der Leitachs-Anfangspositionen in Grad - Daten für Parameter P-0-0705 bei Satz 0 - Daten für Parameter P-0-0712 bei Satz 1

MX_PLCOpen

Input-Variable	Datentyp	Beschreibung
DistanceAdr	POINTER TO REAL	Pointer auf ein Array mit den Hub-Werten der Schritte - Daten für Parameter P-0-0707 bei Satz 0 - Daten für Parameter P-0-0714 bei Satz 1
SlaveAxisVelocityAdr	POINTER TO REAL	Pointer auf ein Array mit den Folgeachs-Geschwindigkeiten der Schritte - Daten für Parameter P-0-0708 bei Satz 0 - Daten für Parameter P-0-0715 bei Satz 1
StepModeAdr	POINTER TO MC_STEP_MODE	Pointer auf ein Array mit den Modus der Schritte - Daten für Parameter P-0-0706 bei Satz 0 - Daten für Parameter P-0-0713 bei Satz 1

Tab. 13-20: *Input-Variablen des Firmware-Funktionsbaustein "MB_ChangeProfileSet"*

Output-Variable	Datentyp	Beschreibung
Done	BOOL	Bei "Done"=TRUE wurden die Daten für den Bewegungssatz übernommen "P-0-0089, Statuswort Synchronbetriebsarten" (Bit12 für Satz 0, Bit 14 für Satz 1)
Active	BOOL	Bei "Active"=TRUE ist die Datenübernahme aktiv
Error	BOOL	Mit "Error"=TRUE meldet der Funktionsbaustein, dass ein Fehler vorliegt
ErrorID *	ERROR_CODE	"ErrorID" gibt eine grobe Fehlerbeschreibung wieder.
ErrorIdent *	ERROR_STRUCT	"ErrorIdent" gibt über die Elemente: "Table", "Additional1" und "Additional2" die genaue Fehlerursache wieder.
CheckOk	BOOL	Bei "CheckOk"=TRUE ist der Bewegungsprofil-Satz fehlerfrei "P-0-0089, Statuswort Synchronbetriebsarten" (Bit13 für Satz 0, Bit 15 für Satz 1)

* siehe nachfolgender Hinweis

Tab. 13-21: *Output-Variablen des Firmware-Funktionsbaustein "MB_ChangeProfileSet"*



HINWEIS: Eine Auflistung aller möglichen Fehler finden Sie hier:
["Fehler-Referenzlisten"](#)

Input-/Output-Variable	Datentyp	Beschreibung
Axis	AXIS_REF	Definiert die Achse. Folgende Achsen sind möglich: MLD-S: Axis1 MLD-M: - Axis1 (Master) oder - Axis2...Axis10 (insgesamt 9 Slaves / entfernte Achsen) Die in der Liste "P-0-1801.0.10, CCD: Adressen projektierter Antriebe" vorhandenen entfernten Achsen entsprechen der Reihenfolge nach Axis2 bis Axis10 (Axis2 wirkt auf den Antrieb des Listenelements 0 des P-0-1801.0.10,...)

Tab. 13-22: Input-/Output-Variable des Firmware-Funktionsbaustein "MB_ChangeProfileSet"

13.3.4 MB_ChangeProfileStep

Der Firmware-Funktionsbaustein "MB_ChangeProfileStep" dient dazu, die vier Parameter, die ein Segment eines Bewegungsprofil-Satzes definieren, zu ändern; dies sind

- Leitachsenanfangsposition,
- Hub,
- Folgeachs-Geschwindigkeit und
- Modus.

Wo sich der zu ändernde Schritt befindet, wird durch die Werte an den Eingängen "SetNumber" und "StepNumber" festgelegt.

Die Einheiten für die Bausteineingänge "Distance" und "SlaveAxisVelocity" werden durch die Wichtungparameter der Achse festgelegt (S-0-0076, S-0-0044).

Über den Eingang "StepMode" wird die Art des Schrittes festgelegt.

Der Ausgang "Done" wird dann gesetzt wenn das veränderte Profil überprüft worden ist. Der Ausgang "CheckOk" wird in Abhängigkeit des Ergebnisses der Prüfung gesetzt. Ist "CheckOk"=FALSE wurde eine Unstimmigkeit festgestellt.

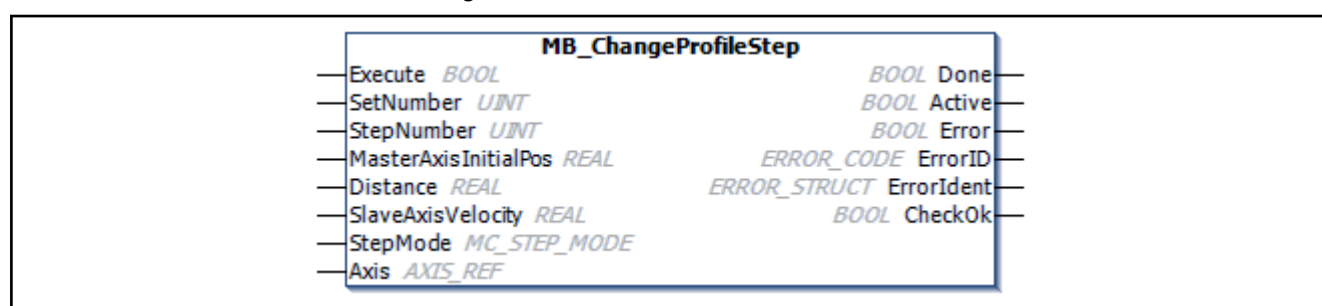


Abb. 13-10: Firmware-Funktionsbaustein "MB_ChangeProfileStep"

MX_PLCoPen

Input-Variable	Datentyp	Beschreibung
Execute	BOOL	siehe " Signallaufverhalten "
SetNumber	UINT	Auswahl des Bewegungsprofil-Satzes 0: Bewegungsprofil-Satz 0 (Parameter P-0-0705 bis P-0-0708) 1: Bewegungsprofil-Satz 1 (Parameter P-0-0712 bis P-0-0715)
StepNumber	UINT	Auswahl des zu ändernden Bewegungsschrittes Jeder Bewegungsprofil-Satz kann aus bis zu 8 Bewegungsschritten bestehen 1: Bewegungsschritt1 / Listenelement 0 von P-0-0705 bis P-0-0708 bzw. P-0-0712 bis P-0-0715) 2: Bewegungsschritt2 / Listenelement 1 von P-0-0705 bis P-0-0708 bzw. P-0-0712 bis P-0-0715) ... 8: Bewegungsschritt8 / Listenelement 7 von P-0-0705 bis P-0-0708 bzw. P-0-0712 bis P-0-0715)
MasterAxisInitialPos	REAL	Festlegung der Leitachs-Anfangsposition des ausgewählten Bewegungsschrittes in Grad P-0-0705, Liste der Leitachs-Anfangsposition, Satz 0 P-0-0712, Liste der Leitachs-Anfangsposition, Satz 1
Distance	REAL	Festlegung des Folgeachswegs des Bewegungsschrittes P-0-0707, Liste der Hübe, Satz 0 P-0-0714, Liste der Hübe, Satz 1
SlaveAxisVelocity	REAL	Festlegung der Folgeachs-Geschwindigkeit innerhalb des Bewegungsschrittes P-0-0708, Liste der Folgeachs-Geschwindigkeiten, Satz 0 P-0-0715, Liste der Folgeachs-Geschwindigkeiten, Satz 1
StepMode	MC_STEP_MODE	Festlegung des Verarbeitungsmodus des ausgewählten Bewegungsschrittes P-0-0706, Liste der Bewegungsgesetze, Satz 0 P-0-0713, Liste der Bewegungsgesetze, Satz 1 • REST_IN_REST_INCLINEDSINE (0) • REST_IN_REST_POLY5 (16#100) • REST_IN_REST_POLY5_VLIM (16#1200) • REST_IN_VELOCITY_POLY5 (16#200) • VELOCITY_IN_REST_POLY5 (16#400) • VELOCITY_IN_VELOCITY_POLY5 (16#700) • CONSTANT_VELOCITY (16#600) • CAM_TABLE_1 (1) • CAM_TABLE_2 (2) • CAM_TABLE_3 (3) • CAM_TABLE_4 (4) • CAM_TABLE_5 (5) • CAM_TABLE_6 (6) • CAM_TABLE_7 (7) • CAM_TABLE_8 (8)

Tab. 13-23: Input-Variablen des Firmware-Funktionsbaustein

"MB_ChangeProfileStep"

Output-Variable	Datentyp	Beschreibung
Done	BOOL	Daten für Bewegungsschritt wurden übernommen "P-0-0089, Statuswort Synchronbetriebsarten" (Bit12 für Satz 0, Bit 14 für Satz 1)
Active	BOOL	Datenübernahme aktiv
Error	BOOL	Mit "Error"=TRUE meldet der Funktionsbaustein, dass ein Fehler vorliegt
ErrorID *	ERROR_CODE	"ErrorID" gibt eine grobe Fehlerbeschreibung wieder.
ErrorIdent *	ERROR_STRUCT	"ErrorIdent" gibt über die Elemente: "Table", "Additional1" und "Additional2" die genaue Fehlerursache wieder.
CheckOk	BOOL	Bewegungsprofil-Satz fehlerfrei "P-0-0089, Statuswort Synchronbetriebsarten" (Bit13 für Satz 0, Bit 15 für Satz 1)

* siehe nachfolgender Hinweis
 Tab. 13-24: *Output-Variablen des Firmware-Funktionsbaustein "MB_ChangeProfileStep"*



HINWEIS: Eine Auflistung aller möglichen Fehler finden Sie hier:
["Fehler-Referenzlisten"](#)

Input-/Output-Variable	Datentyp	Beschreibung
Axis	AXIS_REF	Definiert die Achse. Folgende Achsen sind möglich: MLD-S: Axis1 MLD-M: - Axis1 (Master) oder - Axis2...Axis10 (insgesamt 9 Slaves / entfernte Achsen) Die in der Liste "P-0-1801.0.10, CCD: Adressen projektierter Antriebe" vorhandenen entfernten Achsen entsprechen der Reihenfolge nach Axis2 bis Axis10 (Axis2 wirkt auf den Antrieb des Listenelements 0 des P-0-1801.0.10,...)

Tab. 13-25: *Input-/Output-Variable des Firmware-Funktionsbaustein "MB_ChangeProfileStep"*

13.3.5 MB_ClearAllError

Der Firmware-Funktionsbaustein "MB_ClearAllError" dient dazu, Fehler bei allen realen aktivierten Geräten und bei allen aktivierten E/A-Modulen mit dem Kommando "S-0-0099, C0500 Reset Zustandsklasse 1" zurückzusetzen.

Damit der Funktionsbaustein das Resetkommando korrekt beenden kann, muss er solange zyklisch aufgerufen werden, bis er "Done"=TRUE oder im Fehlerfall "Error" meldet.

Wenn bereits bei einem Gerät oder E/A-Modul das Kommando "S-0-0099, C0500 Reset Zustandsklasse 1" aktiv ist, sollte es erst beendet werden, bevor "MB_ClearAllError" aufgerufen wird.

Wenn ein Fehler bei einem Gerät oder E/A-Modul über das Kommando "S-0-0099, C0500 Reset Zustandsklasse 1" nicht zurückgesetzt werden kann, so wird am Baustein **kein** Fehler ausgegeben.

MX_PLCOpen

Bei deaktivierten Achsen bzw. E/A-Modulen liefert der Baustein **keinen** Fehler.

Bei entfernten Achsen wird der Auftrag an den Servicekanal (Sercos III) abgegeben. Erst bei einem späteren Aufruf kann der Baustein "Done" melden. Damit ergeben sich für die Funktionsbaustein-Aufrufe selbst immer kurze Zeiten. Die Ausgänge "Done" und "Error" müssen hier zyklisch gepollt werden. Die Übertragung der Daten kann abhängig von der Belastung des Servicekanals auch mehrere Sekunden in Anspruch nehmen. Solange der Sercos Ring noch nicht in Phase 2 ist, können keine Daten mit entfernten Achsen ausgetauscht werden. Der Baustein meldet in diesem Fall einen Fehler.



Abb. 13-11: Firmware-Funktionsbaustein "MB_ClearAllError"

Eingangs-Variable	Datentyp	Beschreibung
Execute	BOOL	siehe "Signallaufverhalten"

Tab. 13-26: Eingangs-Variable des Firmware-Funktionsbausteins "MB_ClearAllError"

Ausgangs-Variable	Datentyp	Beschreibung
Done	BOOL	Mit "Done"=TRUE zeigt der Funktionsbaustein an, dass die Bearbeitung der Arbeitsanweisung erfolgreich abgeschlossen ist.
Error	BOOL	Mit "Error"=TRUE meldet der Funktionsbaustein, dass ein Fehler vorliegt
ErrorID *	ERROR_CODE	"ErrorID" gibt eine grobe Fehlerbeschreibung wieder.
ErrorIdent *	ERROR_STRUCT	"ErrorIdent" gibt über die Elemente: "Table", "Additional1" und "Additional2" die genaue Fehlerursache wieder.

* siehe nachfolgender Hinweis

Tab. 13-27: Ausgangs-Variablen des Firmware-Funktionsbausteins "MB_ClearAllError"



HINWEIS: Eine Auflistung aller möglichen Fehler finden Sie hier: "[Fehler-Referenzlisten](#)"

13.3.6 MB_GearInPos

Mit dem Funktionsbaustein "MB_GearInPos" kann eine winkelsynchrone Bewegung der Slave- zur Master-Achse realisiert werden.

Die Slave-Achse kann eine reale Achse oder die interne virtuelle Leitachse (VmAxis1) sein.

Wenn die Slave-Achse eine reale Achse ist, dann wird aus der Position der Master-Achse ein synchroner Lagesollwert für die Slave-Achse ermittelt.

Am Eingang "Master" muss die Master-Achse ausgewählt werden.



Wenn "VmAxis1" als Slave-Achse gewählt ist, dann darf diese Achse nicht gleichzeitig Master-Achse sein.

Der synchrone Lagesollwert " X_{Sync} " wird im Prinzip nach der folgenden Gleichung ermittelt (siehe auch Firmware-Funktionsbeschreibung "Winkelsynchronisation mit realer/virtueller Leitachse"):

$$X_{\text{Sync}} = X_{\text{Master}} * \frac{\text{RatioNumerator}}{\text{RatioDenominator}} * (1 + \text{MasterFineadjust})$$

Abb. 13-12: Berechnung des synchronen Lagesollwerts " X_{sync} "

An dem Eingang "StartMode" muss eingestellt werden, in welchem Modus der Aufsynchronisiervorgang durchgeführt werden soll. Es wird eine Geschwindigkeitsanpassung durchgeführt (relative Synchronisation) und bei absoluter Synchronisation auch eine Lageanpassung. "Anpassung" bedeutet, dass die Istwerte (Geschwindigkeit und Lage) zu den aus der Masterposition berechneten Sollwerten bewegt werden. Ist der "StartMode" "0" oder "1" eingestellt, findet der Aufsynchronisiervorgang nur bei der ersten positiven Flanke an "Execute" statt. Eine weitere positive Flanke führt nur zur Übernahme der Daten am Eingang des Funktionsbausteines.

Bei realen Achsen: Soll der Aufsynchronisiervorgang bei jeder positiven Flanke an "Execute" erfolgen, kann der StartMode "2" oder "3" eingestellt werden; bei der virtuellen Achse wird das wiederholte Aufsynchronisieren nicht unterstützt.

Bei einer realen Achse sind beim Aufsynchronisiervorgang die Parameter "P-0-0143, Aufsynchronisier-Geschwindigkeit" und "P-0-0142, Aufsynchronisier-Beschleunigung" wirksam; bei der virtuellen Achse die Parameter "P-0-0920, Aufsynchronisier-Beschleunigung, Folgeleitachse", und "P-0-0921, Aufsynchronisier-Geschwindigkeit, Folgeleitachse".

Das Aufsynchronisieren kann durch eine Bewegung in positiver Richtung, in negativer Richtung oder auf kürzestem Weg erfolgen. Dies muss an dem Eingang "SyncMode" vorgegeben werden.

Die Position der Masterachse wird direkt in eine Position der Folgeachse umgesetzt.



Wenn der Funktionsbaustein "MB_GearInPos" aktiv ist, ist bei realen Achsen auch der Lageregler des Antriebs aktiv. Einstellungen des Lagereglers (mit/ohne Schleppfehler, Lageistwert 1/2) müssen mit dem Funktionsbaustein "MB_SetPositionControlMode" getroffen werden.

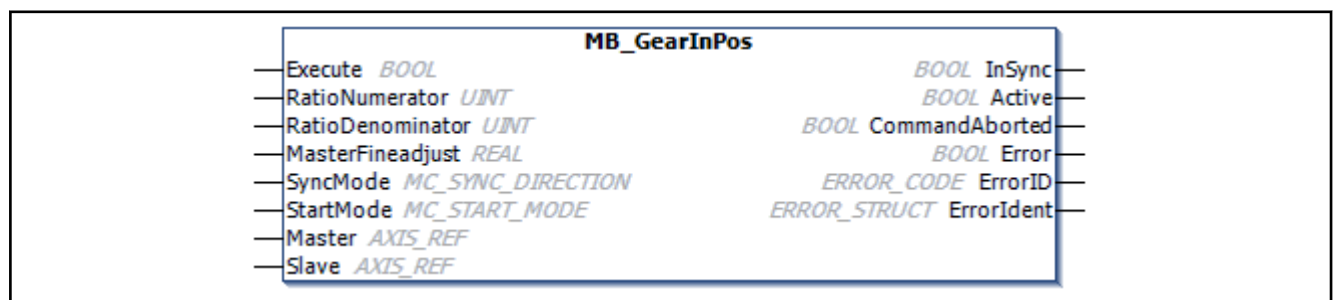


Abb. 13-13: Firmware-Funktionsbaustein "MB_GearInPos"

MX_PLCOpen

Eingangs-Variable	Datentyp	Beschreibung
Execute	BOOL	siehe " Signallaufverhalten "
RatioNumerator	UINT	Getriebeübersetzungsverhältnis - Zähler • "P-0-0157, Elektronisches Getriebe Ausgangsumdrehungen" oder • "P-0-0926, Elektronisches Getriebe Ausgangsumdrehungen, Folgeleitachse"
RatioDenominator	UINT	Getriebeübersetzungsverhältnis - Nenner • "P-0-0156, Elektronisches Getriebe Eingangsumdrehungen" oder • "P-0-0925, Elektronisches Getriebe Eingangsumdrehungen, Folgeleitachse"
MasterFineadjust	REAL	"P-0-0083, Feinabgleich Getriebeübersetzung" oder "P-0-0927, Elektronisches Getriebe Feinabgleich, Folgeleitachse" in %
SyncMode	MC_SYNC_DIRECTION	"P-0-0154, Aufsynchronisier-Richtung" oder "P-0-0922, Aufsynchronisier-Vorzugsrichtung, Folgeleitachse": 0:= kürzeste Entfernung 1:= positive Richtung 2:= negative Richtung
StartMode	MC_START_MODE	"P-0-0155, Aufsynchronisier-Modus", Bit 1 oder "P-0-0919, Aufsynchronisiermodus, Folgeleitachse" 0:= absolute Synchronisation 1:= relative Synchronisation 2:= absolute Synchronisation mit Aufsynchronisieren 3:= relative Synchronisation mit Aufsynchronisieren

Tab. 13-28: Eingangs-Variablen des Firmware-Funktionsbausteins
"MB_GearInPos"

Ausgangs-Variable	Datentyp	Beschreibung
InSync	BOOL	Mit dem Ausgangswert TRUE meldet der Funktionsbaustein, dass die Slave-Achse synchron ist. "InSync" entspricht P-0-0115, Bit12 der parametrisierten Achse. Das Bit12 des P-0-0115 entspricht für diesen Anwendungsfall Bit8 (Synchronbetriebsart in Synchronisation) von "P-0-0089, Statuswort Synchronbetriebsarten". Bei der virtuellen Achse entspricht "InSync" dem Bit 8 des "P-0-0768, Virtuelle Leitachse, Positionierstatus"
Active	BOOL	Mit "Active"=TRUE zeigt der Funktionsbaustein an, dass er die Achsbewegung bestimmt. "Active" bleibt solange TRUE bis "CommandAborted" oder "Error" gesetzt wird.
CommandAborted	BOOL	Mit "CommandAborted"=TRUE zeigt der Funktionsbaustein an, dass ein anderer Motion-Baustein aktiviert wurde oder ein Antriebsfehler vorlag. Ein Antriebsfehler wird über die Error-Ausgänge des "MC_Power" bzw. "MX_Power"
Error	BOOL	Mit "Error"=TRUE meldet der Funktionsbaustein, dass ein Fehler vorliegt
ErrorID *	ERROR_CODE	"ErrorID" gibt eine grobe Fehlerbeschreibung wieder.
ErrorIdent *	ERROR_STRUCT	"ErrorIdent" gibt über die Elemente: "Table", "Additional1" und "Additional2" die genaue Fehlerursache wieder.

* siehe nachfolgender Hinweis

Tab. 13-29: Ausgangs-Variablen des Firmware-Funktionsbausteins "MB_GearInPos"



HINWEIS: Eine Auflistung aller möglichen Fehler finden Sie hier: ["Fehler-Referenzlisten"](#)

MX_PLCOpen

Eingangs-/Ausgangs-Variable (VAR_IN_OUT)	Datentyp	Beschreibung	
		MLD-S	MLD-M
Master	AXIS_REF	Mögliche Auswahl der Master-Achse	
		• VmAxis1^{*1} / VmAxisInt: P-0-0761 • RmAxis1^{*1} / RmAxisInt: P-0-0052 • LinkAxis1^{*1} / LinkAxisInt: P-0-0053	• VmAxisInt: P-0-0761^{*3} • RmAxisInt: P-0-0052 • LinkAxisInt: P-0-0053 bei Axis1, P-0-0787 bei Axis2..10 • VmAxis1^{*2}: P-0-0761^{*3} • RmAxis1^{*2}: P-0-1820.0.3 • LinkAxis1^{*2}: P-0-0053 • Axisn^{*2}: P-0-1808.x.10 oder P-0-1821.x.1 (Auswahl in P-0-1820.0.4)
		*1: ab MPx-20 als Masterachse verwendbar	*2: muss in P-0-1820.0.1 als Leitachse konfiguriert sein, wird aufbereitet *3: damit P-0-0761 den richtigen Wert liefert, muss in P-0-0916 ein Auswahlparameter eingetragen sein
Slave	AXIS_REF	Mögliche Auswahl der Slave-Achse	
		• Axis1 • VmAxis1^{*4}	• VmAxis1^{*4} • Axis1 • Axis2..Axis10 Die in der Liste "P-0-1801.0.10, CCD: Adressen projizierte Antriebe" vorhandenen entfernten Achsen entsprechen der Reihenfolge nach Axis2 bis Axis10 (Axis2 wirkt auf den Antrieb des Listenelements 0 des P-0-1801.0.10,...)
		*4: Bei Verwendung von VmAxis1 als Slave wird der Folgeleitachsbetrieb aktiviert. Die virtuelle Leitachse kann zwei verschiedenen Leitachsen folgen: • RmAxisInt: P-0-0052 • LinkAxisInt: P-0-0053	

P-0-0052: Lageistwert Messgeber
P-0-0053: Leitachsposition
P-0-0761: Leitachsposition für Folgeachse
P-0-0787: Verbundachsposition-1
P-0-0916: Leitachsformatwandler Signalauswahl
P-0-1808.x.10: CCD: Aktiver Lageistwert
P-0-1820.0.1: CCD: Konfigurationsliste Leitachsen
P-0-1820.0.3: CCD: Lageistwert Messgeber
P-0-1820.0.4: CCD: Auswahl Achsposition
P-0-1821.x.1: CCD: Lagesollwert für Leitachse n

Tab. 13-30: Eingangs-/Ausgangs-Variablen des Firmware-Funktionsbausteins "MB_GearInPos"

13.3.7 MB_Home

Mit dem Funktionsbaustein "MB_Home" kann in der ausgewählten Achse das "Antriebsgeführte Referenzieren" durchgeführt werden.

Bei nicht absoluten Gebern kann so der Maßbezug hergestellt werden.

Im Referenzfahr-Parameter müssen Details zur Ausführung des Referenzierens festgelegt werden. Nach der Ausführung des Funktionsbausteins befindet sich die Achse im Stillstand.

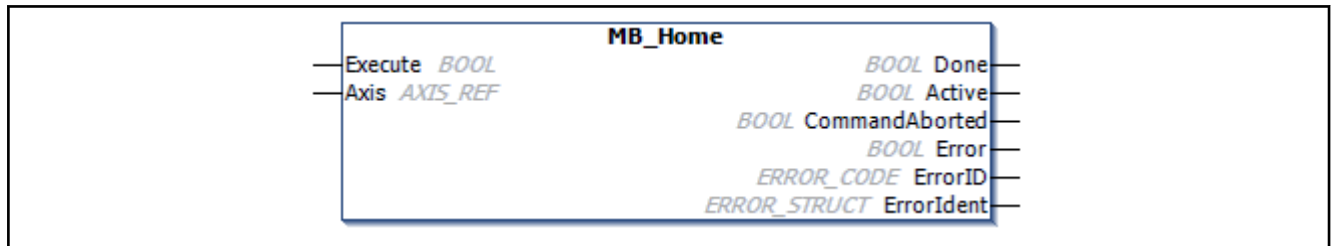


Abb. 13-14: Firmware-Funktionsbaustein "MB_Home"

Eingangs-Variable	Datentyp	Beschreibung
Execute	BOOL	siehe "Signallaufverhalten"

Tab. 13-31: Eingangs-Variable des Firmware-Funktionsbausteins "MB_Home"

Ausgangs-Variable	Datentyp	Beschreibung
Done	BOOL	Referenzieren abgeschlossen, Geschwindigkeit 0
Active	BOOL	Mit "Active"=TRUE zeigt der Funktionsbaustein an, dass er die Achsbewegung bestimmt. "Active" bleibt solange TRUE bis "CommandAborted" oder "Error" gesetzt wird.
CommandAborted	BOOL	Mit "CommandAborted"=TRUE zeigt der Funktionsbaustein an, dass ein anderer Motion-Baustein aktiviert wurde oder ein Antriebsfehler vorlag.
Error	BOOL	Mit "Error"=TRUE meldet der Funktionsbaustein, dass ein Fehler vorliegt
ErrorID *	ERROR_CODE	"ErrorID" gibt eine grobe Fehlerbeschreibung wieder.
ErrorIdent *	ERROR_STRUCT	"ErrorIdent" gibt über die Elemente: "Table", "Additional1" und "Additional2" die genaue Fehlerursache wieder.

* siehe nachfolgender Hinweis

Tab. 13-32: Ausgangs-Variablen des Firmware-Funktionsbausteins "MB_Home"



HINWEIS: Eine Auflistung aller möglichen Fehler finden Sie hier: ["Fehler-Referenzlisten"](#)

MX_PLCopen

Eingangs-/Ausgangs-Variable (VAR_IN_OUT)	Datentyp	Beschreibung
Axis	AXIS_REF	Definiert die Achse. Folgende Achsen sind möglich: MLD-S: Axis1 MLD-M: - Axis1 (Master) oder - Axis2...Axis10 (insgesamt 9 Slaves / entfernte Achsen) Die in der Liste "P-0-1801.0.10, CCD: Adressen projektierte Antriebe" vorhandenen entfernten Achsen entsprechen der Reihenfolge nach Axis2 bis Axis10 (Axis2 wirkt auf den Antrieb des Listenelements 0 des P-0-1801.0.10,...)

Tab. 13-33: Eingangs-/Ausgangs-Variablen des Firmware-Funktionsbausteins "MB_Home"

13.3.8 MB_MotionProfile

Der Funktionsbaustein "MB_MotionProfile" realisiert eine elektronische Kurvenscheibenfunktion. Die Kurvenscheiben sind dabei analytisch definiert durch die Parameter, die zu der Betriebsart "Elektronisches Bewegungsprofil" gehören. Aus der Position der Master-Achse wird ein synchroner Lagesollwert für die Slave-Achse ermittelt.

Am Eingang "Master" muss die Master-Achse ausgewählt werden.

Zur Berechnung des synchronen Lagesollwertes " φ_{Synch} " wird zunächst ein Tabellenzugriffswinkel (P-0-0227) nach der folgenden Gleichung ermittelt:

$$\varphi_{\text{Zugriff}} = \varphi_{\text{Master}} * \frac{\text{RatioNumerator}}{\text{RatioDenominator}} * (1 + \text{MasterFineadjust})$$

Abb. 13-15: Tabellenzugriffswinkel

Mit dem Tabellenzugriffswinkel als Argument wird mittels eines Polynoms 5. Ordnung ein synchroner Lagesollwert berechnet.

An dem Eingang "StartMode" muss eingestellt werden, in welchem Modus der Aufsynchronisiervorgang durchgeführt werden soll. Es wird eine Geschwindigkeitsanpassung durchgeführt (relative Synchronisation) und bei absoluter Synchronisation auch eine Lageanpassung. "Anpassung" bedeutet, dass die Istwerte (Geschwindigkeit und Lage) zu den aus der Masterposition berechneten Sollwerten bewegt werden. Ist der StartMode "0" oder "1" eingestellt, findet der Aufsynchronisiervorgang nur bei der ersten positiven Flanke an "Execute" statt. Eine weitere positive Flanke führt nur zur Übernahme der Daten am Eingang des Funktionsbausteines.

Soll der Aufsynchronisiervorgang bei jeder positiven Flanke an "Execute" erfolgen, kann der StartMode "2" oder "3" eingestellt werden.

Beim Aufsynchronisiervorgang sind die Parameter "P-0-0143, Aufsynchronisierung-Geschwindigkeit" und "P-0-0142, Aufsynchronisierung-Beschleunigung" wirksam.



Wenn der Funktionsbaustein "MB_MotionProfile" aktiv ist, ist auch der Lageregler des Antriebs aktiv. Einstellungen des Lagereglers (mit/ohne Schleppfehler, Lageistwert 1 / 2) müssen mit dem Funktionsbaustein "MB_SetPositionControlMode" getroffen werden.

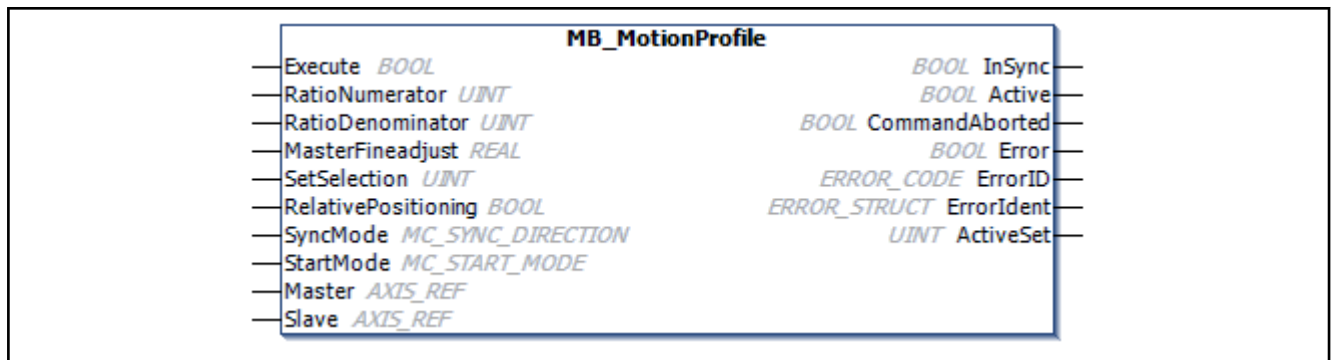


Abb. 13-16: Firmware-Funktionsbaustein "MB_MotionProfile"

Eingangs-Variable	Datentyp	Beschreibung
Execute	BOOL	siehe "Signallaufverhalten"
RatioNumerator	UINT	Getriebeübersetzungsverhältnis - Zähler P-0-0157, Elektronisches Getriebe Ausgangsumdrehungen
RatioDenominator	UINT	Getriebeübersetzungsverhältnis - Nenner P-0-0156, Elektronisches Getriebe Eingangsumdrehungen
MasterFineadjust	REAL	"P-0-0083, Feinabgleich Getriebeübersetzung" in %
SetSelection	UINT	P-0-0088, Bit9: Auswahl des Bewegungsprofilsatzes (0 oder 1)
RelativePositioning	BOOL	P-0-0088, Bit10: FALSE = Fester (absoluter) Lagebezug, TRUE = relative Positionierung.
SyncMode	MC_SYNC_DIRECTION	"P-0-0154, Aufsynchonisier-Richtung": 0:= kürzeste Entfernung 1:= positive Richtung 2:= negative Richtung
StartMode	MC_START_MODE	"P-0-0155, Aufsynchonisier-Modus", Bit 1 0:= absolute Synchronisation 1:= relative Synchronisation 2:= absolute Synchronisation mit Aufsynchonisieren 3:= relative Synchronisation mit Aufsynchonisieren

Tab. 13-34: Eingangs-Variablen des Firmware-Funktionsbausteins "MB_MotionProfile"

MX_PLCOpen

Ausgangs-Variable	Datentyp	Beschreibung
InSync	BOOL	Mit dem Ausgangswert TRUE meldet der Funktionsbaustein, dass die Slave-Achse synchron ist. "InSync" entspricht P-0-0115, Bit12 der parametrisierten Achse und SetSelection(P-0-0088.9) = ActiveSet(P-0-0089.3). Das Bit12 vom P-0-0115 entspricht für diesen Anwendungsfall Bit8 "Synchronbetriebsart in Synchronisation" vom Parameter "P-0-0089, Statuswort Synchronbetriebsarten".
Active	BOOL	Mit "Active"=TRUE zeigt der Funktionsbaustein an, dass er die Achsbewegung bestimmt. "Active" bleibt solange TRUE bis "CommandAborted" oder "Error" gesetzt wird.
CommandAborted	BOOL	Mit "CommandAborted"=TRUE zeigt der Funktionsbaustein an, dass ein anderer Motion-Baustein aktiviert wurde oder ein Antriebsfehler vorlag. Ein Antriebsfehler wird über die Error-Ausgänge des "MC_Power" bzw. "MX_Power".
Error	BOOL	Mit "Error"=TRUE meldet der Funktionsbaustein, dass ein Fehler vorliegt
ErrorID *	ERROR_CODE	"ErrorID" gibt eine grobe Fehlerbeschreibung wieder.
ErrorIdent *	ERROR_STRUCT	"ErrorIdent" gibt über die Elemente: "Table", "Additional1" und "Additional2" die genaue Fehlerursache wieder.
ActiveSet	UINT	P-0-0089, Bit3: Der Ausgang zeigt die Nummer des aktiven Bewegungsprofilsatzes an.

* siehe nachfolgender Hinweis

Tab. 13-35: Ausgangs-Variablen des Firmware-Funktionsbausteins
"MB_MotionProfile"HINWEIS: Eine Auflistung aller möglichen Fehler finden Sie hier:
["Fehler-Referenzlisten"](#)

Eingangs-/Ausgangs-Variable (VAR_IN_OUT)	Datentyp	Beschreibung	
		MLD-S	MLD-M
Master	AXIS_REF	Mögliche Auswahl der Master-Achse	
		• VmAxis1^{*1} / VmAxisInt: P-0-0761 • RmAxis1^{*1} / RmAxisInt: P-0-0052 • LinkAxis1^{*1} / LinkAxisInt: P-0-0053	• VmAxisInt: P-0-0761^{*3} • RmAxisInt: P-0-0052 • LinkAxisInt: P-0-0053 bei Axis1, P-0-0787 bei Axis2..10 • VmAxis1^{*2}: P-0-0761^{*3} • RmAxis1^{*2}: P-0-1820.0.3 • LinkAxis1^{*2}: P-0-0053 • Axisn^{*2}: P-0-1808.x.10 oder P-0-1821.x.1 (Auswahl in P-0-1820.0.4)
		*1: ab MPx-20 als Masterachse verwendbar	*2: muss in P-0-1820.0.1 als Leitachse konfiguriert sein, wird aufbereitet *3: damit P-0-0761 den richtigen Wert liefert, muss in P-0-0916 ein Auswahlparameter eingetragen sein
Slave	AXIS_REF	Mögliche Auswahl der Slave-Achse	
		Axis1	• Axis1 • Axis2..Axis10 Die in der Liste "P-0-1801.0.10, CCD: Adressen projizierte Antriebe" vorhandenen entfernten Achsen entsprechen der Reihenfolge nach Axis2 bis Axis10 (Axis2 wirkt auf den Antrieb des Listenelements 0 des P-0-1801.0.10,...)

- P-0-0052: Lageistwert Messgeber
 P-0-0053: Leitachseposition
 P-0-0761: Leitachseposition für Folgeachse
 P-0-0787: Verbundachseposition-1
 P-0-0916: Leitachsformatwandler Signalauswahl
 P-0-1808.x.10: CCD: Aktiver Lageistwert
 P-0-1820.0.1: CCD: Konfigurationsliste Leitachsen
 P-0-1820.0.3: CCD: Lageistwert Messgeber
 P-0-1820.0.4: CCD: Auswahl Achseposition
 P-0-1821.x.1: CCD: Lagesollwert für Leitachse n

Tab. 13-36: Eingangs-/Ausgangs-Variablen des Firmware-Funktionsbausteins

13.3.9 MB_Phasing

Mit dem Funktionsbaustein "MB_Phasing" kann ein Offset zur Master-Achse realisiert werden, der als Eingang an einem Synchronisations-Baustein anliegt.

Verändert wird der Parameter "P-0-0054, Leitachseposition additiv". Mit den Eingängen "Velocity" und "Acceleration" wird das Bewegungsprofil bestimmt, mit dem der Weg verfahren wird.

MX_PLCOpen

Die Einheiten der Eingänge, welche die Antriebsbewegung definieren, sind fest.

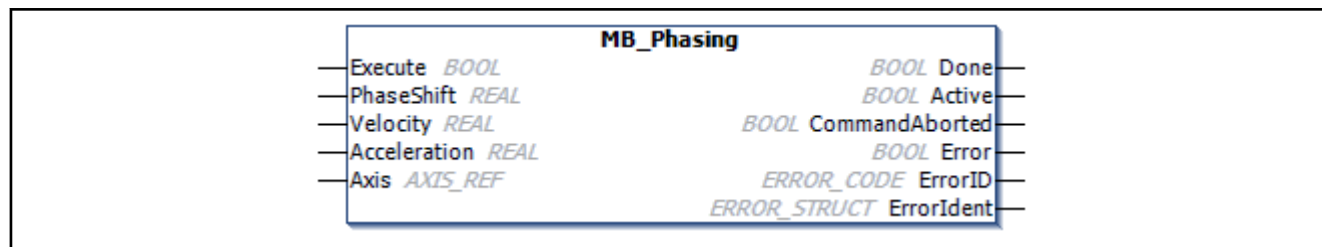


Abb. 13-17: Firmware-Funktionsbaustein "MB_Phasing"

Input-Variable	Datentyp	Beschreibung
Execute	BOOL	siehe "Signallaufverhalten"
PhaseShift	REAL	Leitachsposition additiv (P-0-0054) in Grad
Velocity	REAL	Leitachsposition additiv Positioniergeschwindigkeit (P-0-0688) in Upm
Acceleration	REAL	Leitachsposition additiv Positionierbeschleunigung (P-0-0689) in rad/s ²

Tab. 13-37: Input-Variablen des Firmware-Funktionsbausteins "MB_Phasing"

Output-Variable	Datentyp	Beschreibung
Done	BOOL	Mit "Done"=TRUE meldet der Firmware-Funktionsbaustein, dass PhaseShift erreicht wurde (Quittung durch Bit 6 des "P-0-0089, Statuswort Synchronbetriebsarten")
Active	BOOL	Mit TRUE zeigt der Ausgang "Active" an, dass der vorgegebene Wert noch nicht erreicht wurde
CommandAborted	BOOL	Der Ausgang "CommandAborted" wird auf TRUE gesetzt, wenn der Firmware-Funktionsbaustein noch nicht mit "Done"=TRUE gemeldet hat, dass er fertig ist und der Firmware-Funktionsbaustein durch einen anderen Motion-Funktionsbaustein abgebrochen wurde
Error	BOOL	Mit "Error"=TRUE meldet der Funktionsbaustein, dass ein Fehler vorliegt
ErrorID *	ERROR_CODE	"ErrorID" gibt eine grobe Fehlerbeschreibung wieder.
ErrorIdent *	ERROR_STRUCT	"ErrorIdent" gibt über die Elemente: "Table", "Additional1" und "Additional2" die genaue Fehlerursache wieder.

* siehe nachfolgender Hinweis

Tab. 13-38: Output-Variablen des Firmware-Funktionsbausteins "MB_Phasing"



HINWEIS: Eine Auflistung aller möglichen Fehler finden Sie hier: ["Fehler-Referenzlisten"](#)

Eingangs-/Ausgangs-Variable (VAR_IN_OUT)	Datentyp	Beschreibung
Axis	AXIS_REF	Definiert die Achse. Folgende Achsen sind möglich: MLD-S: Axis1 MLD-M: - Axis1 (Master) oder - Axis2...Axis10 (insgesamt 9 Slaves / entfernte Achsen) Die in der Liste "P-0-1801.0.10, CCD: Adressen projektierter Antriebe" vorhandenen entfernten Achsen entsprechen der Reihenfolge nach Axis2 bis Axis10 (Axis2 wirkt auf den Antrieb des Listenelements 0 des P-0-1801.0.10,...)

Tab. 13-39: Eingangs-/Ausgangs-Variable des Firmware-Funktionsbausteins "MB_Phasing"

13.3.10 MB_PhasingSlave

Mit dem Funktionsbaustein "MB_PhasingSlave" kann eine Verstellung der Slave-Achse durch einen Positions-Offset realisiert werden. Die Verstellung ist nur bei aktivem Synchronisations-Baustein ("MC_CamIn", "MB_GearInPos" oder "MB_MotionProfile") wirksam.

Ein neuer Wert für den Positions-Offset (Eingang "PhaseShift") wird unter Berücksichtigung der am Funktionsbaustein anliegenden Werte für Geschwindigkeit (Eingang "Velocity") und Beschleunigung (Eingang "Acceleration") angefahren.

Bei Verwendung des "MB_PhasingSlave" mit einer realen Achse (Axis1... Axis10) sind die Einheiten für die Eingänge "PhaseShift", "Velocity" und "Acceleration" durch die Wichtungsparameter der Achse wie folgt festgelegt:

- "PhaseShift": "S-0-0076, Wichtungsart für Lagedaten"
- "Velocity": "S-0-0044, Wichtungsart für Geschwindigkeitsdaten"
- "Acceleration": "S-0-0160, Wichtungsart für Beschleunigungsdaten").

Bei Verwendung des "MB_PhasingSlave" mit der virtuellen Leitachse V_{mAxis1} (Folgeleitachsbetrieb) sind die Einheiten für die Eingänge "PhaseShift", "Velocity" und "Acceleration" wie folgt festgelegt:

- "PhaseShift": "P-0-0928, Leitachsposition additiv, Folgeleitachse"
- "Velocity": "P-0-0929, Verstellgeschwindigkeit Leitachspos. add., Folgeleitachse"
- "Acceleration": "P-0-0930, Verstellbeschleunigung Leitachspos. add., Folgeleitachse").

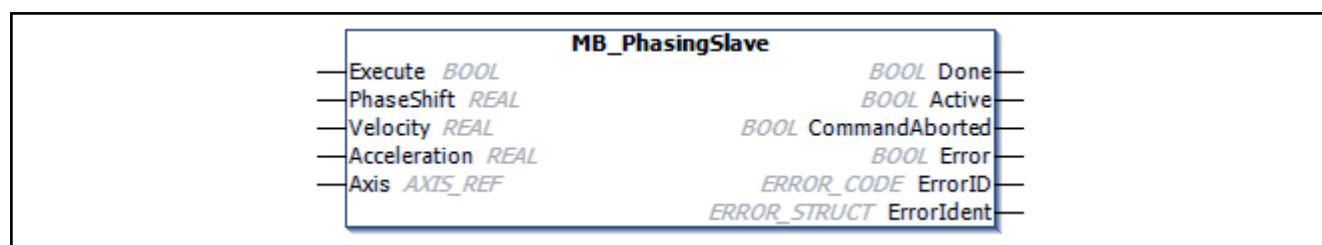


Abb. 13-18: Firmware-Funktionsbaustein "MB_PhasingSlave"

MX_PLCOpen

Eingangs-Variable	Datentyp	Beschreibung	
Execute	BOOL	siehe " Signallaufverhalten "	
		Reale Achse	Virtuelle Leitachse
PhaseShift	REAL	S-0-0048, Lagesollwert additiv	P-0-0928, Leitachspannung additiv, Folgeleitachse
Velocity	REAL	P-0-0686, Lagesollwert additiv Positioniergeschwindigkeit	P-0-0929, Verstellgeschwindigkeit Leitachspannung. add., Folgeleitachse
Acceleration	REAL	P-0-0687, Lagesollwert additiv Positionierbeschleunigung	P-0-0930, Verstellbeschleunigung Leitachspannung. add., Folgeleitachse

Tab. 13-40: Eingangs-Variablen des Firmware-Funktionsbausteins
"MB_PhasingSlave"

Ausgangs-Variable	Datentyp	Beschreibung
Done	BOOL	PhaseShift erreicht (Quittung durch "P-0-0152, Aufsynchonisieren beendet", Bit 0=1)
Active	BOOL	Mit "Active"=TRUE zeigt der Funktionsbaustein an, dass er zur Achsbewegung beiträgt
CommandAborted	BOOL	Mit "CommandAborted"=TRUE zeigt der Funktionsbaustein an, dass die Aufsynchonisierung abgebrochen wurde
Error	BOOL	Mit "Error"=TRUE meldet der Funktionsbaustein, dass ein Fehler vorliegt
ErrorID *	ERROR_CODE	"ErrorID" gibt eine grobe Fehlerbeschreibung wieder.
ErrorIdent *	ERROR_STRUCT	"ErrorIdent" gibt über die Elemente: "Table", "Additional1" und "Additional2" die genaue Fehlerursache wieder.

* siehe nachfolgender Hinweis

Tab. 13-41: Ausgangs-Variablen des Firmware-Funktionsbausteins
"MB_PhasingSlave"



HINWEIS: Eine Auflistung aller möglichen Fehler finden Sie hier:
["Fehler-Referenzlisten"](#)

Eingangs-/Ausgangs-Variable (VAR_IN_OUT)	Datentyp	Beschreibung
Axis	AXIS_REF	Definiert die Achse. Folgende Achsen sind möglich: MLD-S: Axis1 MLD-M: - Axis1 (Master) oder - Axis2...Axis10 (insgesamt 9 Slaves / entfernte Achsen) Die in der Liste "P-0-1801.0.10, CCD: Adressen projektierter Antriebe" vorhandenen entfernten Achsen entsprechen der Reihenfolge nach Axis2 bis Axis10 (Axis2 wirkt auf den Antrieb des Listenelements 0 des P-0-1801.0.10,...) Interne virtuelle Achse: "VmAxis1"=P-0-0761

Tab. 13-42: Eingangs-/Ausgangs-Variablen des Firmware-Funktionsbausteins "MB_PhasingSlave"

13.3.11 MB_PreSetMode

Mit dem Funktionsbaustein "MB_PreSetMode" kann die Achse auf eine Kommandierung vor dem Einschalten der Leistung vorbereitet werden (PreSetModus / Vorselektieren von Betriebsarten).

"MB_PreSetMode" kommandiert die Achse bereits mit den entsprechenden Daten, die Achse startet jedoch erst beim Einschalten der Leistung mit "MC_Power" und folgt dann direkt den Vorgaben.



"MB_PreSetMode" kann nur im Betriebsmodus ohne Leistung und ohne Fehler aktiviert werden. Nach dem Aktivieren mit steigender "Execute"-Flanke ist es erlaubt einen Motion-Funktionsbaustein (wie beispielsweise "MC_MoveVelocity") oder auch "MX_SetOpMode" zu aktivieren.

Nachfolgend beispielhaft ein typischer Ablauf bei Verwendung des Funktionsbausteins "MB_PreSetMode":

1. Einschalten des Antriebs
2. Umschalten oder abwarten bis Antrieb in Betriebsmodus ("bb")
3. Aktivieren von "MB_PreSetMode" mit steigender "Execute"-Flanke
4. Starten eines Motion-Funktionsbausteins (in "Ab")
5. Einschalten der Leistung mit "MC_Power"

Die Achse beginnt sofort mit der gewünschten Bewegung



Abb. 13-19: Firmware-Funktionsbaustein "MB_PreSetMode"

MX_PLCOpen

Eingangs-Variable	Datentyp	Beschreibung
Execute	BOOL	siehe "Signallaufverhalten"

Tab. 13-43: Eingangs-Variable des Firmware-Funktionsbausteins "MB_PreSetMode"

Ausgangs-Variable	Datentyp	Beschreibung
Done	BOOL	Mit "Done"=TRUE zeigt der Funktionsbaustein an, dass die Bearbeitung der Arbeitsanweisung erfolgreich abgeschlossen ist
Error	BOOL	Mit "Error"=TRUE meldet der Funktionsbaustein, dass ein Fehler vorliegt
ErrorID *	ERROR_CODE	"ErrorID" gibt eine grobe Fehlerbeschreibung wieder.
ErrorIdent *	ERROR_STRUCT	"ErrorIdent" gibt über die Elemente: "Table", "Additional1" und "Additional2" die genaue Fehlerursache wieder.
Value	DINT	Ist der Ausgang "Done"=TRUE, zeigt "Value" den Wert des ausgewählten Elements an

* siehe nachfolgender Hinweis

Tab. 13-44: Ausgangs-Variablen des Firmware-Funktionsbausteins "MB_PreSetMode"



HINWEIS: Eine Auflistung aller möglichen Fehler finden Sie hier: "[Fehler-Referenzlisten](#)"

Eingangs-/Ausgangs-Variable (VAR_IN_OUT)	Datentyp	Beschreibung
Axis	AXIS_REF	Definiert die Achse. Folgende Achsen sind möglich: MLD-S: Axis1 MLD-M: - Axis1 (Master) oder - Axis2...Axis10 (insgesamt 9 Slaves / entfernte Achsen) Die in der Liste "P-0-1801.0.10, CCD: Adressen projektierte Antriebe" vorhandenen entfernten Achsen entsprechen der Reihenfolge nach Axis2 bis Axis10 (Axis2 wirkt auf den Antrieb des Listenelements 0 des P-0-1801.0.10,...)

Tab. 13-45: Eingangs-/Ausgangs-Variable des Firmware-Funktionsbausteins "MB_SetPositionControlMode"

13.3.12 MB_SetPositionControlMode

Der Funktionsbaustein "MB_SetPositionControlMode" dient zur Einstellung des Lagereglers.



MoveAbsTransitionMode Der Eingang ist erst ab MPx-20 / HDx-20 verfügbar

Abb. 13-20: Firmware-Funktionsbaustein "MB_SetPositionControlMode"

Eingangs-Variable	Datentyp	Beschreibung
Execute	BOOL	siehe " Signallaufverhalten "
LaglessPositionControl	BOOL	Mit dem Wert an dem Eingang "LaglessPositionControl" wird Bit 2 von "S-0-0520, Achsregler-Steuerwort" beeinflusst: FALSE: Lageregelung mit Schleppfehler TRUE: schleppfehlerfreie Lageregelung
ExternalEncoder	BOOL	Mit dem Wert an dem Eingang "ExternalEncoder" wird Bit 0 von "S-0-0520, Achsregler-Steuerwort" beeinflusst: FALSE: Regelung mit Motorgeber (Lageistwert-1) TRUE: Regelung mit externem Geber (Lageistwert-2)
MoveAbsUseCmdPos	BOOL	Mit dem Wert an dem Eingang "MoveAbsUseCmdPos" wird Bit 4 von "S-0-0346, Positioniersteuerwort" beeinflusst: FALSE: aktueller Lageistwert als Bezugspunkt für relative Positioniersollwerte TRUE: letzte wirksame Zielposition als Bezugspunkt für relative Positioniersollwerte
MoveAbsTransitionMode	MB_TRANSITION_MODE	HINWEIS: Der Eingang ist erst ab MPx-20 / HDx-20 verfügbar. Mit dem Wert an dem Eingang "MoveAbsTransitionMode" werden die Bits 5, 6 und 7 von "S-0-0346, Positioniersteuerwort" beeinflusst: MB_ABORTING: Direkte Übernahme (default) MB_BUFFERED: Start nach Zwischenhalt MB_OLD_VELOCITY: Folgesatz mit alter Geschwindigkeit aufsetzen MB_NEW_VELOCITY: Folgesatz mit neuer Geschwindigkeit aufsetzen

Tab. 13-46: *Eingangs-Variablen des Firmware-Funktionsbausteins "MB_SetPositionControlMode"*

Ausgangs-Variable	Datentyp	Beschreibung
Done	BOOL	Mit "Done"=TRUE zeigt der Funktionsbaustein an, dass die Bearbeitung der Arbeitsanweisung erfolgreich abgeschlossen ist
Active	BOOL	Die Abarbeitung des Bausteins ist sehr schnell; der Zustandswechsel des Ausgangs "Active" kann bei IndraMotion MLD nicht ausgewertet werden
Error	BOOL	Mit "Error"=TRUE meldet der Funktionsbaustein, dass ein Fehler vorliegt
ErrorID*	ERROR_CODE	"ErrorID" gibt eine grobe Fehlerbeschreibung wieder.
ErrorIdent *	ERROR_STRUCT	"ErrorIdent" gibt über die Elemente: "Table", "Additional1" und "Additional2" die genaue Fehlerursache wieder.

* siehe nachfolgender Hinweis

Tab. 13-47: *Ausgangs-Variablen des Firmware-Funktionsbausteins "MB_SetPositionControlMode"*



HINWEIS: Eine Auflistung aller möglichen Fehler finden Sie hier: "[Fehler-Referenzlisten](#)"

MX_PLCOpen

Eingangs-/Ausgangs-Variable (VAR_IN_OUT)	Datentyp	Beschreibung
Axis	AXIS_REF	Definiert die Achse. Folgende Achsen sind möglich: MLD-S: Axis1 MLD-M: - Axis1 (Master) oder - Axis2...Axis10 (insgesamt 9 Slaves / entfernte Achsen) Die in der Liste "P-0-1801.0.10, CCD: Adressen projektierter Antriebe" vorhandenen entfernten Achsen entsprechen der Reihenfolge nach Axis2 bis Axis10 (Axis2 wirkt auf den Antrieb des Listenelements 0 des P-0-1801.0.10,...)

Tab. 13-48: Eingangs-/Ausgangs-Variable des Firmware-Funktionsbausteins "MB_SetPositionControlMode"

13.3.13 MB_Stop

Der Firmware-Funktionsbaustein "MB_Stop" dient dazu, den Antrieb in den Zustand STOP zu bringen.

Mit der steigenden Flanke an "ExecuteLock" wird das Anhalten begonnen.

Falls die SPS keine Kontrolle über den Antrieb hat, wird ein Fehler zurückgeliefert.

Solange der "ExecuteLock"-Eingang TRUE ist, kann kein anderer Motion-Funktionsbaustein gestartet werden. Dazu wird der Antrieb in den Zustand "AH" (Antrieb Halt) gebracht. Mit "ExecuteLock"=FALSE bleibt der Antrieb in "AH".



Die Funktion "Antrieb Halt" wird in der Funktionsbeschreibung zur Firmware beschrieben.

Der Antrieb wird mit der an "Deceleration" angegebenen Verzögerung zum Stillstand gebracht.

Wenn in einem Projekt mehrere Instanzen von "MC_Stop" oder "MB_Stop" aufgerufen werden, wirkt immer die letzte Instanz; **Ausnahme:** Ein aktiver "MC_Stop" kann nicht durch einen "MB_Stop" unterbrochen werden.

"MC_Stop" hat eine höhere Priorität als "MB_Stop"; d. h., dass ein laufender "MC_Stop" nicht durch einen "MB_Stop" abgebrochen werden kann.



Über den Funktionsbaustein "MB_Stop" kann auch der "P-0-0759, Virtuelle Leitachse, Geschwindigkeitsistwert" über eine definierte Rampe zum Stillstand gebracht werden. Dazu muss der Leitachsgenerator über "P-0-0917, Steuerwort Leitachsgenerator", Bit0="1" aktiviert werden. Um den virtuellen Lageistwert (P-0-0758) in die interne virtuelle Leitachsposition (VmAxis1, "P-0-0761, Leitachsposition") zu wandeln, muss der Parameter P-0-0758 in den Parameter "P-0-0916, Leitachsformatwandler Signalauswahl" geschrieben werden.

Siehe dazu auch Firmware-Funktionsbeschreibung "Leitachsgenerator".



Zum Testen des Funktionsbausteins kann die Instanzvariable "ActivateCounter" herangezogen werden. Sie zählt die positiven Flanken des "ExecuteLock"-Signals und zeigt damit an, ob der Baustein mit Flanke durchlaufen wird.

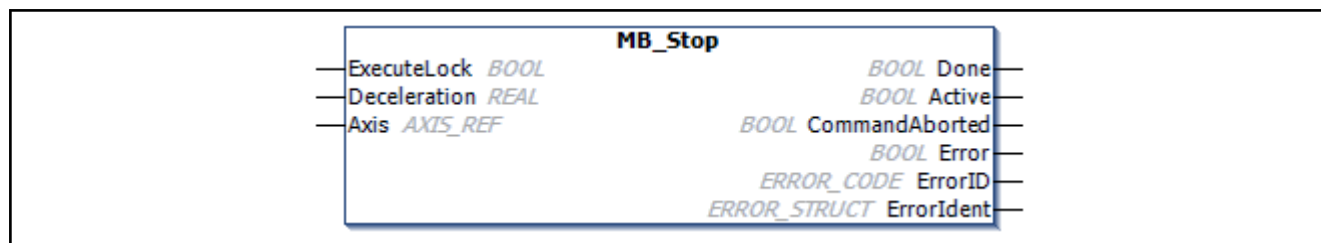


Abb. 13-21: Firmware-Funktionsbaustein "MB_Stop"

Input-Variable	Datentyp	Beschreibung
ExecuteLock	BOOL	Mit steigender Flanke an "ExecuteLock" wird der Antrieb in "AH" geschaltet.
Deceleration	REAL	Über diesen Eingang wird die Verzögerung eingestellt. Die angegebene Verzögerung beschreibt den Parameter "S-0-0372, Verzögerung Schnell-Halt".

Tab. 13-49: Input-Variablen des Firmware-Funktionsbausteins "MB_Stop"

Output-Variable	Datentyp	Beschreibung
Done	BOOL	Mit "Done"=TRUE (S-0-0331, Bit0="1") meldet der Firmware-Funktionsbaustein, dass der Antrieb steht. Nur wenn der "ExecuteLock"-Eingang=TRUE ist, wird der Ausgang "Done" gebildet
Active	BOOL	Mit "Active"=TRUE zeigt der Firmware-Funktionsbaustein an, dass er die Achsbewegung bestimmt
CommandAborted	BOOL	Mit "CommandAborted"=TRUE meldet der Funktionsbaustein, dass ein anderer Funktionsbaustein einen anderen Zustand kommandiert hat (weiterer "MC_Stop" oder "MB_Stop")
Error	BOOL	Mit "Error"=TRUE meldet der Funktionsbaustein, dass ein Fehler vorliegt
ErrorID *	ERROR_CODE	"ErrorID" gibt eine grobe Fehlerbeschreibung wieder.
ErrorIdent *	ERROR_STRUCT	"ErrorIdent" gibt über die Elemente: "Table", "Additional1" und "Additional2" die genaue Fehlerursache wieder.

* siehe nachfolgender Hinweis

Tab. 13-50: Output-Variablen des Firmware-Funktionsbausteins "MB_Stop"



HINWEIS: Eine Auflistung aller möglichen Fehler finden Sie hier: ["Fehler-Referenzlisten"](#)

MX_PLCOpen

Input-/Output-Variable (VAR_IN_OUT)	Datentyp	Beschreibung
Axis	AXIS_REF	Definiert die Achse. Folgende Achsen sind möglich: MLD-S: Axis1 MLD-M: - Axis1 (Master) oder - Axis2...Axis10 (insgesamt 9 Slaves / entfernte Achsen) Die in der Liste "P-0-1801.0.10, CCD: Adressen projektierte Antriebe" vorhandenen entfernten Achsen entsprechen der Reihenfolge nach Axis2 bis Axis10 (Axis2 wirkt auf den Antrieb des Listenelements 0 des P-0-1801.0.10,...)

Tab. 13-51: Input-/Output-Variable des Firmware-Funktionsbausteins "MB_Stop"

13.3.14 MC_CamIn

Mit dem Funktionsbaustein "MC_CamIn" kann eine elektronische Kurvenscheibe realisiert werden.

Aus der Position der Master-Achse wird ein synchroner Lagesollwert für die Slave-Achse ermittelt. Am Eingang "Master" muss die Master-Achse ausgewählt werden.

Zur Berechnung des synchronen Lagesollwertes "XSynch" wird zunächst ein Tabellenzugriffswinkel (P-0-0227) nach der folgenden Gleichung ermittelt:

$$\varphi_{\text{Zugriff}} = \varphi_{\text{Master}} * \frac{\text{RatioNumerator}}{\text{RatioDenominator}} * (1 + \text{MasterFineadjust})$$

Abb. 13-22: Berechnung des P-0-0227, Kurvenscheibentabelle, Zugriffswinkel (Tabellenzugriffswinkel)

Mit dem berechneten Tabellenzugriffswinkel wird in die an dem Eingang "CamTableID" ausgewählte Tabelle gegriffen und ein Prozentwert entnommen. Die Multiplikation dieses Tabellenwertes mit dem Lagewert an dem Eingang "CamShaftDistance" liefert den synchronen Lagesollwert.

An dem Eingang "StartMode" muss eingestellt werden, in welchem Modus der Aufsynchronisiervorgang durchgeführt werden soll. Es wird eine Geschwindigkeitsanpassung durchgeführt (relative Synchronisation) und bei absoluter Synchronisation auch eine Lageanpassung. "Anpassung" bedeutet, dass die Istwerte (Geschwindigkeit und Lage) zu den aus der Masterposition berechneten Sollwerten mit parametrierter Geschwindigkeit (P-0-0143, Aufsynchronisier-Geschwindigkeit) und Beschleunigung (P-0-0142, Aufsynchronisier-Beschleunigung) bewegt werden. Ist der StartMode "0" oder "1" eingestellt, findet der Aufsynchronisiervorgang nur bei der ersten positiven Flanke an "Execute" statt. Eine weitere positive Flanke führt nur zur Übernahme der Daten am Eingang des Funktionsbausteines. Die Eingangsgrößen (Getriebe, Hub, Kurvenscheibe) werden dann abhängig von der Konfiguration des "P-0-0086, Konfigurationswort Synchronbetriebsarten" wirksam.

Soll der Aufsynchronisiervorgang bei jeder positiven Flanke an "Execute" erfolgen, kann der StartMode "2" oder "3" eingestellt werden. Die Eingangsgrößen (Getriebe, Hub, Kurvenscheibe) werden damit unabhängig von P-0-0086 sofort wirksam.

Bei Modulo-Lagewichtung kann das Aufsynchronisieren durch eine Bewegung in positiver Richtung, in negativer Richtung oder auf kürzestem Weg erfolgen. Dies muss an dem Eingang "SyncMode" vorgegeben werden.



Wenn der Funktionsbaustein "MC_CamIn" aktiv ist, ist auch der Lageregler des Antriebs aktiv. Einstellungen des Lagereglers (mit/ ohne Schleppfehler, Lageistwert 1 / 2) müssen mit dem Funktionsbaustein "MB_SetPositionControlMode" getroffen werden.

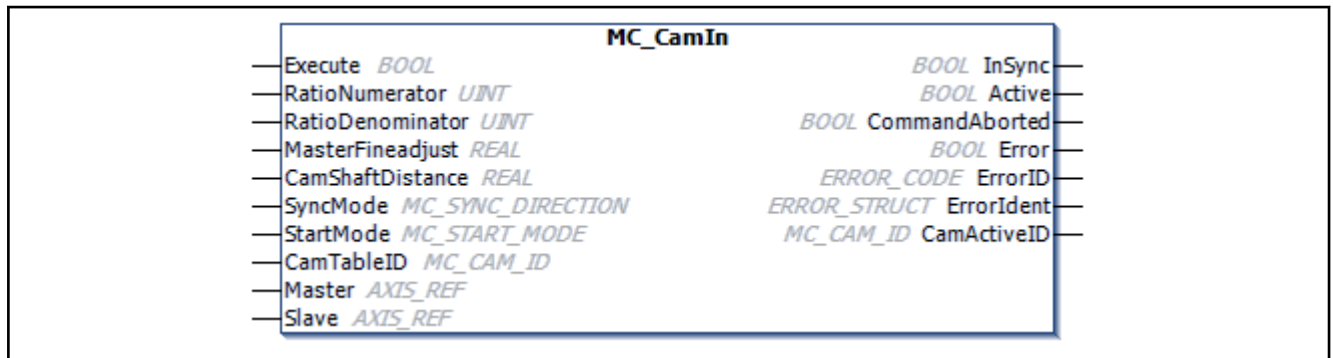


Abb. 13-23: Firmware-Funktionsbaustein "MC_CamIn"

Eingangs-Variable	Datentyp	Beschreibung
Execute	BOOL	siehe "Signallaufverhalten"
RatioNumerator	UINT	Getriebeübersetzungsverhältnis - Zähler P-0-0157, Elektronisches Getriebe Ausgangsumdrehungen
RatioDenominator	UINT	Getriebeübersetzungsverhältnis - Nenner P-0-0156, Elektronisches Getriebe Eingangsumdrehungen
MasterFineadjust	REAL	"P-0-0083, Feinabgleich Getriebeübersetzung" in %
CamShaftDistance	REAL	Kurvenscheibe Hub. Bewertung der prozentualen Tabellenwerte
SyncMode	MC_SYNC_DIRECTION	"P-0-0154, Aufsynchronisier-Richtung": 0:= kürzeste Entfernung 1:= positive Richtung 2:= negative Richtung
StartMode	MC_START_MODE	"P-0-0155, Aufsynchronisier-Modus", Bit 1 0:= absolute Synchronisation 1:= relative Synchronisation 2:= absolute Synchronisation mit Aufsynchronisieren 3:= relative Synchronisation mit Aufsynchronisieren
CamTableID	MC_CAM_ID	Nummer der Kurventabelle (1..8 entsprechend den Parametern P-0-0072, P-0-0092, P-0-0780..P-0-0785)

Tab. 13-52: Eingangs-Variablen des Firmware-Funktionsbausteins "MC_CamIn"

MX_PLCoPen

Ausgangs-Variable	Datentyp	Beschreibung
InSync	BOOL	Mit "InSync"=TRUE zeigt der Funktionsbaustein an, dass die Slave-Achse synchron ist. "InSync" entspricht • "P-0-0115, Gerätesteuerung: Statuswort", Bit12: "Sollwert erreicht"" der parametrisierten Achse und • "CamTableID"="CamActiveID" und • "P-0-0089, Statuswort Synchronbetriebsarten", Bit5="1" (Status Hubumschaltung)". Das Bit12 vom P-0-0115 entspricht für diesen Anwendungsfall Bit8 "Synchronbetriebsart in Synchronisation" vom Parameter "P-0-0089, Statuswort Synchronbetriebsarten".
Active	BOOL	Mit "Active"=TRUE zeigt der Funktionsbaustein an, dass er die Achsbewegung bestimmt. "Active" bleibt TRUE, bis "CommandAborted" oder "Error" gesetzt wird.
CommandAborted	BOOL	Mit "CommandAborted"=TRUE zeigt der Funktionsbaustein an, dass ein anderer Motion-Baustein aktiviert wurde oder dass ein Antriebsfehler vorlag. Ein Antriebsfehler wird über die "Error"-Ausgänge des "MC_Power" angezeigt.
Error	BOOL	Mit "Error"=TRUE meldet der Funktionsbaustein, dass ein Fehler vorliegt
ErrorID *	ERROR_CODE	"ErrorID" gibt eine grobe Fehlerbeschreibung wieder.
ErrorIdent *	ERROR_STRUCT	"ErrorIdent" gibt über die Elemente: "Table", "Additional1" und "Additional2" die genaue Fehlerursache wieder.
CamActiveID		Nummer der im Baustein aktiven Kurvenscheibe

* siehe nachfolgender Hinweis

Tab. 13-53: *Ausgangs-Variablen des Firmware-Funktionsbausteins "MC_CamIn"*

HINWEIS: Eine Auflistung aller möglichen Fehler finden Sie hier:
["Fehler-Referenzlisten"](#)

Eingangs-/Ausgangs-Variable (VAR_IN_OUT)	Datentyp	Beschreibung	
		MLD-S	MLD-M
Master	AXIS_REF	Mögliche Auswahl der Master-Achse	
		• VmAxis1^{*1} / VmAxisInt: P-0-0761 • RmAxis1^{*1} / RmAxisInt: P-0-0052 • LinkAxis1^{*1} / LinkAxisInt: P-0-0053	• VmAxisInt: P-0-0761^{*3} • RmAxisInt: P-0-0052 • LinkAxisInt: P-0-0053 bei Axis1, P-0-0787 bei Axis2..10 • VmAxis1^{*2}: P-0-0761^{*3} • RmAxis1^{*2}: P-0-1820.0.3 • LinkAxis1^{*2}: P-0-0053 • Axisn^{*2}: P-0-1808.x.10 oder P-0-1821.x.1 (Auswahl in P-0-1820.0.4)
		*1: ab MPx-20 als Masterachse verwendbar	*2: muss in P-0-1820.0.1 als Leitachse konfiguriert sein, wird aufbereitet *3: damit P-0-0761 den richtigen Wert liefert, muss in P-0-0916 ein Auswahlparameter eingetragen sein
Slave	AXIS_REF	Mögliche Auswahl der Slave-Achse	
		Axis1	• Axis1 • Axis2..Axis10 Die in der Liste "P-0-1801.0.10, CCD: Adressen projizierte Antriebe" vorhandenen entfernten Achsen entsprechen der Reihenfolge nach Axis2 bis Axis10 (Axis2 wirkt auf den Antrieb des Listenelements 0 des P-0-1801.0.10,...)

- P-0-0052: Lageistwert Messgeber
 P-0-0053: Leitachspannung
 P-0-0761: Leitachspannung für Folgeachse
 P-0-0787: Verbundachspannung-1
 P-0-0916: Leitachsformatwandler Signalauswahl
 P-0-1808.x.10: CCD: Aktiver Lageistwert
 P-0-1820.0.1: CCD: Konfigurationsliste Leitachsen
 P-0-1820.0.3: CCD: Lageistwert Messgeber
 P-0-1820.0.4: CCD: Auswahl Achspannung
 P-0-1821.x.1: CCD: Lagesollwert für Leitachse n

Tab. 13-54: Eingangs-/Ausgangs-Variablen des Firmware-Funktionsbausteins

13.3.15 MC_CamOut

Der Firmware-Funktionsbaustein "MC_CamOut" dient dazu, einen aktiven Synchronisations-Baustein zu beenden. Es erfolgt ein Übergang auf Geschwindigkeitsregelung. Als Geschwindigkeits-Sollwert wird der bei steigender Execute-Flanke vorliegende Geschwindigkeits-Istwert genommen.

MX_PLCoPen

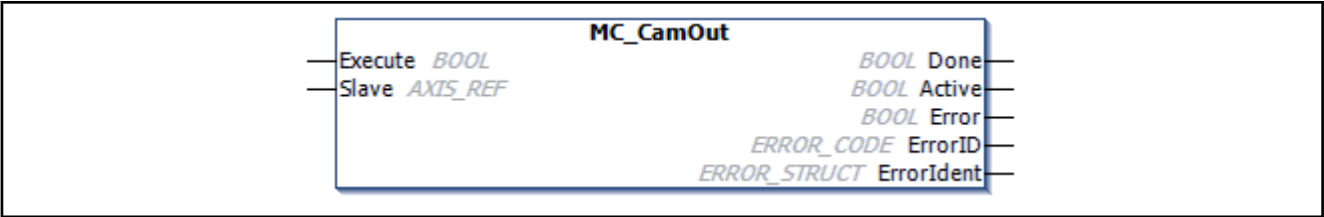


Abb. 13-24: Firmware-Funktionsbaustein "MC_CamOut"

Input-Variable	Datentyp	Beschreibung
Execute	BOOL	siehe "Signallaufverhalten"

Tab. 13-55: Input-Variablen des Firmware-Funktionsbausteins "MC_CamOut"

Output-Variable	Datentyp	Beschreibung
Done	BOOL	Mit "Done"=TRUE meldet der Firmware-Funktionsbaustein, dass die vorgegebene Geschwindigkeit erreicht wurde. (S-0-0330, Bit0=1)
Active	BOOL	Mit TRUE zeigt der Ausgang "Active" an, dass er die Achsbewegung bestimmt
Error	BOOL	Mit "Error"=TRUE meldet der Funktionsbaustein, dass ein Fehler vorliegt
ErrorID *	ERROR_CODE	"ErrorID" gibt eine grobe Fehlerbeschreibung wieder
ErrorIdent *	ERROR_STRUCT	"ErrorIdent" gibt über die Elemente: Table, Additional1 und Additional2 die genaue Fehlerursache wieder

* siehe nachfolgender Hinweis

Tab. 13-56: Output-Variablen des Firmware-Funktionsbausteins "MC_CamOut"



HINWEIS: Eine Auflistung aller möglichen Fehler finden Sie hier:
["Fehler-Referenzlisten"](#)

Input-/Output-Variable (VAR_IN_OUT)	Datentyp	Beschreibung
Slave	AXIS_REF	Definiert die Achse. Folgende Achsen sind möglich: MLD-S: Axis1 MLD-M: - Axis1 (Master) oder - Axis2...Axis10 (insgesamt 9 Slaves / entfernte Achsen) Die in der Liste "P-0-1801.0.10, CCD: Adressen projektierte Antriebe" vorhandenen entfernten Achsen entsprechen der Reihenfolge nach Axis2 bis Axis10 (Axis2 wirkt auf den Antrieb des Listenelements 0 des P-0-1801.0.10,...) "VmAxis1" (P-0-0761, Leitachposition für Folgeachse)

Tab. 13-57: Input-/Output-Variablen des Firmware-Funktionsbausteins
"MC_CamOut"

13.3.16 MC_GearIn

Mit dem Funktionsbaustein "MC_GearIn" kann eine geschwindigkeitssynchrone Bewegung der Slave- zur Master-Achse realisiert werden.

Bei Aktivierung des Funktionsbausteins beschleunigt die Slave-Achse mit der in "P-0-0142, Aufsynchronisier-Beschleunigung" festgelegten

Beschleunigung auf die synchrone Geschwindigkeit. Die synchrone Geschwindigkeit wird aus der Geschwindigkeit der Master-Achse und den Eingangsdaten des Bausteins nach folgender Gleichung gebildet.

$$V_{\text{sync}} = V_{\text{Master}} * \frac{\text{RatioNumerator}}{\text{RatioDenominator}} * (1 + \text{MasterFineadjust})$$

Abb. 13-25: Berechnung der synchronen Geschwindigkeit

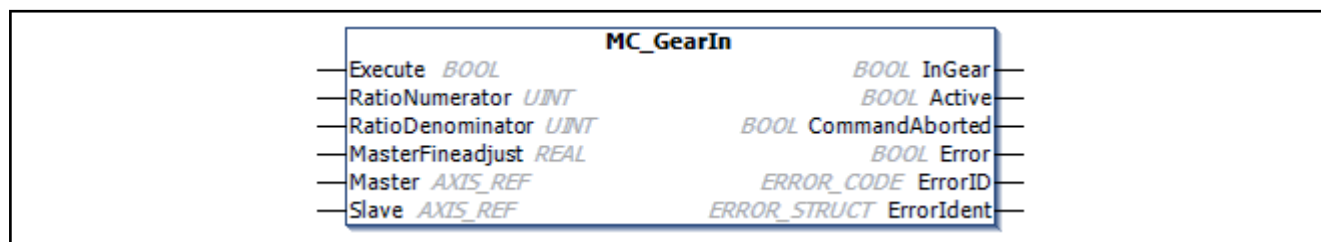


Abb. 13-26: Firmware-Funktionsbaustein "MC_GearIn"

Eingangs-Variable	Datentyp	Beschreibung
Execute	BOOL	siehe "Signallaufverhalten"
RatioNumerator	UINT	Getriebeübersetzungsverhältnis - Zähler P-0-0157, Elektronisches Getriebe Ausgangsumdrehungen
RatioDenominator	UINT	Getriebeübersetzungsverhältnis - Nenner P-0-0156, Elektronisches Getriebe Eingangsumdrehungen
MasterFineadjust	REAL	"P-0-0083, Feinabgleich Getriebeübersetzung" in %

Tab. 13-58: Eingangs-Variablen des Firmware-Funktionsbausteins "MC_GearIn"

Ausgangs-Variable	Datentyp	Beschreibung
InGear	BOOL	Mit "InGear"=TRUE zeigt der Funktionsbaustein an, dass die Slave-Achse geschwindigkeitssynchron ist. "InGear" entspricht "P-0-0115, Gerätesteuerung: Statuswort", Bit12 der parametrisierten Achse. Dieses Bit entspricht für diesen Anwendungsfall Bit8 "Synchronbetriebsart in Synchronisation" des Parameters "P-0-0089, Statuswort Synchronbetriebsarten".
Active	BOOL	Mit "Active"=TRUE zeigt der Funktionsbaustein an, dass er die Achsbewegung bestimmt. "Active" bleibt TRUE, bis "CommandAborted" oder "Error" gesetzt wird.
CommandAborted	BOOL	Mit "CommandAborted"=TRUE zeigt der Funktionsbaustein an, dass ein anderer Motion-Baustein aktiviert wurde oder dass ein Antriebsfehler vorlag. Ein Antriebsfehler wird über die Error-Ausgänge des "MC_Power" angezeigt
Error	BOOL	Mit "Error"=TRUE meldet der Funktionsbaustein, dass ein Fehler vorliegt
ErrorID *	ERROR_CODE	"ErrorID" gibt eine grobe Fehlerbeschreibung wieder.
ErrorIdent *	ERROR_STRUCT	"ErrorIdent" gibt über die Elemente: "Table", "Additional1" und "Additional2" die genaue Fehlerursache wieder.

* siehe nachfolgender Hinweis

Tab. 13-59: Ausgangs-Variablen des Firmware-Funktionsbausteins "MC_GearIn"



HINWEIS: Eine Auflistung aller möglichen Fehler finden Sie hier: "[Fehler-Referenzlisten](#)"

MX_PLCOpen

Eingangs-/Ausgangs-Variable (VAR_IN_OUT)	Datentyp	Beschreibung	
		MLD-S	MLD-M
Master	AXIS_REF	Mögliche Auswahl der Master-Achse	
		• VmAxis1^{*1} / VmAxisInt: P-0-0761 • RmAxis1^{*1} / RmAxisInt: P-0-0052 • LinkAxis1^{*1} / LinkAxisInt: P-0-0053	• VmAxisInt: P-0-0761^{*3} • RmAxisInt: P-0-0052 • LinkAxisInt: P-0-0053 bei Axis1, P-0-0787 bei Axis2..10 • VmAxis1^{*2}: P-0-0761^{*3} • RmAxis1^{*2}: P-0-1820.0.3 • LinkAxis1^{*2}: P-0-0053 • Axisn^{*2}: P-0-1808.x.10 oder P-0-1821.x.1 (Auswahl in P-0-1820.0.4)
		*1: ab MPx-20 als Masterachse verwendbar	*2: muss in P-0-1820.0.1 als Leitachse konfiguriert sein, wird aufbereitet *3: damit P-0-0761 den richtigen Wert liefert, muss in P-0-0916 ein Auswahlparameter eingetragen sein
Slave	AXIS_REF	Mögliche Auswahl der Slave-Achse	
		Axis1	• Axis1 • Axis2..Axis10 Die in der Liste "P-0-1801.0.10, CCD: Adressen projizierte Antriebe" vorhandenen entfernten Achsen entsprechen der Reihenfolge nach Axis2 bis Axis10 (Axis2 wirkt auf den Antrieb des Listenelements 0 des P-0-1801.0.10,...)

P-0-0052: Lageistwert Messgeber
 P-0-0053: Leitachsposition
 P-0-0761: Leitachsposition für Folgeachse
 P-0-0787: Verbundachsposition-1
 P-0-0916: Leitachsformatwandler Signalauswahl
 P-0-1808.x.10: CCD: Aktiver Lageistwert
 P-0-1820.0.1: CCD: Konfigurationsliste Leitachsen
 P-0-1820.0.3: CCD: Lageistwert Messgeber
 P-0-1820.0.4: CCD: Auswahl Achsposition
 P-0-1821.x.1: CCD: Lagesollwert für Leitachse n

Tab. 13-60: Eingangs-/Ausgangs-Variablen des Firmware-Funktionsbausteins

13.3.17 MC_GearOut

Der Firmware-Funktionsbaustein "MC_GearOut" dient dazu, einen aktiven Synchronisations-Baustein zu beenden. Es erfolgt ein Übergang auf Geschwindigkeitsregelung bei einer realen Achse. Als Geschwindigkeits-Sollwert wird der bei steigender "Execute"-Flanke vorliegende Geschwindigkeits-Istwert genommen.

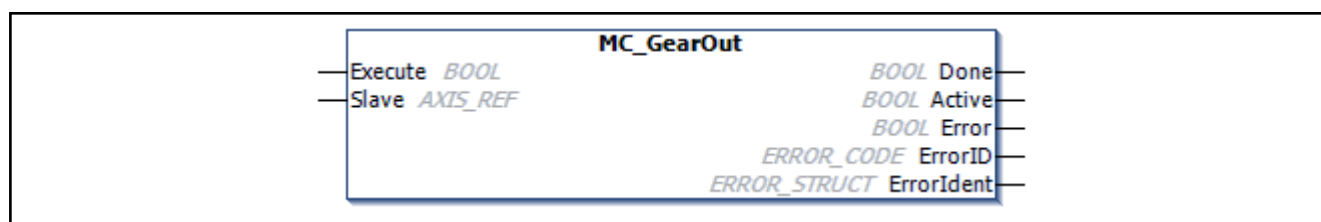


Abb. 13-27: Firmware-Funktionsbaustein "MC_GearOut"

Input-Variable	Datentyp	Beschreibung
Execute	BOOL	siehe "Signallaufverhalten"

Tab. 13-61: Input-Variablen des Firmware-Funktionsbausteins "MC_GearOut"

Output-Variable	Datentyp	Beschreibung
Done	BOOL	Mit "Done"=TRUE meldet der Firmware-Funktionsbaustein, dass die vorgegebene Geschwindigkeit erreicht wurde. (S-0-0330, Bit0="1")
Active	BOOL	Mit TRUE zeigt der Ausgang "Active" an, dass er die Achsbewegung bestimmt
Error	BOOL	Mit "Error"=TRUE meldet der Funktionsbaustein, dass ein Fehler vorliegt
ErrorID *	ERROR_CODE	"ErrorID" gibt eine grobe Fehlerbeschreibung wieder.
ErrorIdent *	ERROR_STRUCT	"ErrorIdent" gibt über die Elemente: "Table", "Additional1" und "Additional2" die genaue Fehlerursache wieder.

* siehe nachfolgender Hinweis

Tab. 13-62: Output-Variablen des Firmware-Funktionsbausteins "MC_GearOut"



HINWEIS: Eine Auflistung aller möglichen Fehler finden Sie hier: ["Fehler-Referenzlisten"](#)

Input-/Output-Variable (VAR_IN_OUT)	Datentyp	Beschreibung
Slave	AXIS_REF	Definiert die Achse. Folgende Achsen sind möglich: MLD-S: Axis1 MLD-M: - Axis1 (Master) oder - Axis2...Axis10 (insgesamt 9 Slaves / entfernte Achsen) Die in der Liste "P-0-1801.0.10, CCD: Adressen projektierter Antriebe" vorhandenen entfernten Achsen entsprechen der Reihenfolge nach Axis2 bis Axis10 (Axis2 wirkt auf den Antrieb des Listenelements 0 des P-0-1801.0.10,...) "VmAxis1" (P-0-0761, Leitachspolition für Folgeachse) (ab IndraDrive MP05)

Tab. 13-63: Input-/Output-Variable des Firmware-Funktionsbausteins "MC_GearOut"

13.3.18 MC_Jog

Der Firmware-Funktionsbaustein "MC_Jog" dient dazu, den Antrieb zu Tippen ("Endlos Fahren" positiv bzw. "Endlos Fahren" negativ). Intern wird dazu in der Betriebsart "Antriebsgeführtes Positionieren" der Tippbetrieb aktiviert.

MX_PLCOpen

Bei permanenter Kontrolle und bei entfernten Achsen wird die Nebenbetriebsart 3 und deren Parameter verwendet. Bei temporärer Kontrolle wird die SPS-Nebenbetriebsart 11 und deren Parameter verwendet. Die Tippgeschwindigkeit muss an dem Eingang "JogVelo" vorgegeben werden. Über die Eingänge "JogAcc" und "JogDec" muss bestimmt werden wie die Tippgeschwindigkeit erreicht wird.



In einem SPS-Zyklus kann nur eine Kommandierung vorgegeben werden.



Über den Firmware-Funktionsbaustein "MC_Jog" kann auch dem virtuellen Leitachsgenerator eine Tippgeschwindigkeit vorgegeben werden. Dazu muss der Leitachsgenerator über "P-0-0917, Steuerwort Leitachsgenerator", Bit0="1" aktiviert werden.

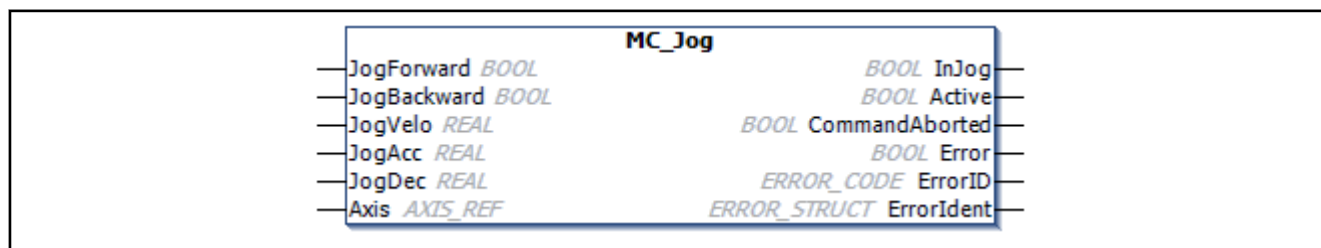


Abb. 13-28: Firmware-Funktionsbaustein "MC_Jog"

Input-Variable	Datentyp	Beschreibung
JogForward	BOOL	Mit einer steigenden Flanke werden die Eingänge "JogVelo", "JogAcc", "JogDec" und "Axis" übernommen. Es wird die Betriebsart "Antriebsgeführtes Positionieren" aktiviert und "Tippen positiv" gestartet [Positioniersteuerwort (reale Achse: S-0-0346 / virtuelle Leitachse: P-0-0760), Bit1/0=11]. Wird "JogForward" wieder zurückgesetzt (FALSE), dann wird Positionierhalt kommandiert (reale Achse: S-0-0346 / virtuelle Leitachse: P-0-0760, Bit2/1/0=111)
JogBackward	BOOL	Mit einer steigenden Flanke werden die Eingänge "JogVelo", "JogAcc", "JogDec" und "Axis" übernommen. Es wird die Betriebsart "Antriebsgeführtes Positionieren" aktiviert und "Tippen negativ" gestartet [Positioniersteuerwort (reale Achse: S-0-0346 / virtuelle Leitachse: P-0-0760), Bit2/1/0=101]. Wird "JogBackward" wieder zurückgesetzt (FALSE), dann wird Positionierhalt kommandiert (reale Achse: S-0-0346 / virtuelle Leitachse: P-0-0760, Bit2/1/0=111).
JogVelo	REAL	Positioniergeschwindigkeit (reale Achse: S-0-0259 / virtuelle Leitachse: P-0-0770) muss >="0" sein. Hinweis: Ist der Funktionsbaustein aktiv, dann wird der Eingang "JogVelo" zyklisch ausgewertet und bei einer Änderung übernommen.
JogAcc	REAL	Positionierbeschleunigung (reale Achse: S-0-0260 / virtuelle Leitachse: P-0-0771) muss >"0" sein. Hinweis: Ist der Funktionsbaustein aktiv, dann wird der Eingang "JogAcc" zyklisch ausgewertet und bei einer Änderung übernommen
JogDec	REAL	Positioniervverzögerung (reale Achse: S-0-0359 / virtuelle Leitachse: P-0-0772) muss >"0" sein. Hinweis: Ist der Funktionsbaustein aktiv, dann wird der Eingang "JogDec" zyklisch ausgewertet und bei einer Änderung übernommen

Tab. 13-64: Input-Variablen des Firmware-Funktionsbausteins "MC_Jog"

Output-Variable	Datentyp	Beschreibung
InJog	BOOL	Mit "InJog"]=TRUE zeigt der Funktionsbaustein an, dass sich die Istgeschwindigkeit außerhalb des Stillstandfensters (reale Achse: S-0-0124 / virtuelle Leitachse: P-0-0912) befindet.
Active	BOOL	Mit "Active"]=TRUE zeigt der Funktionsbaustein an, dass er die Achsbewegung bestimmt.
CommandAborted	BOOL	Mit "CommandAborted"]=TRUE zeigt der Funktionsbaustein an, dass ein anderer Motion-Baustein aktiviert wurde oder dass ein Antriebsfehler vorlag.
Error	BOOL	Mit "Error"]=TRUE meldet der Funktionsbaustein, dass ein Fehler vorliegt
ErrorID *	ERROR_CODE	"ErrorID" gibt eine grobe Fehlerbeschreibung wieder.
ErrorIdent *	ERROR_STRUCT	"ErrorIdent" gibt über die Elemente: "Table", "Additional1" und "Additional2" die genaue Fehlerursache wieder.

* siehe nachfolgender Hinweis

Tab. 13-65: Output-Variablen des Firmware-Funktionsbausteins "MC_Jog"



HINWEIS: Eine Auflistung aller möglichen Fehler finden Sie hier:
["Fehler-Referenzlisten"](#)

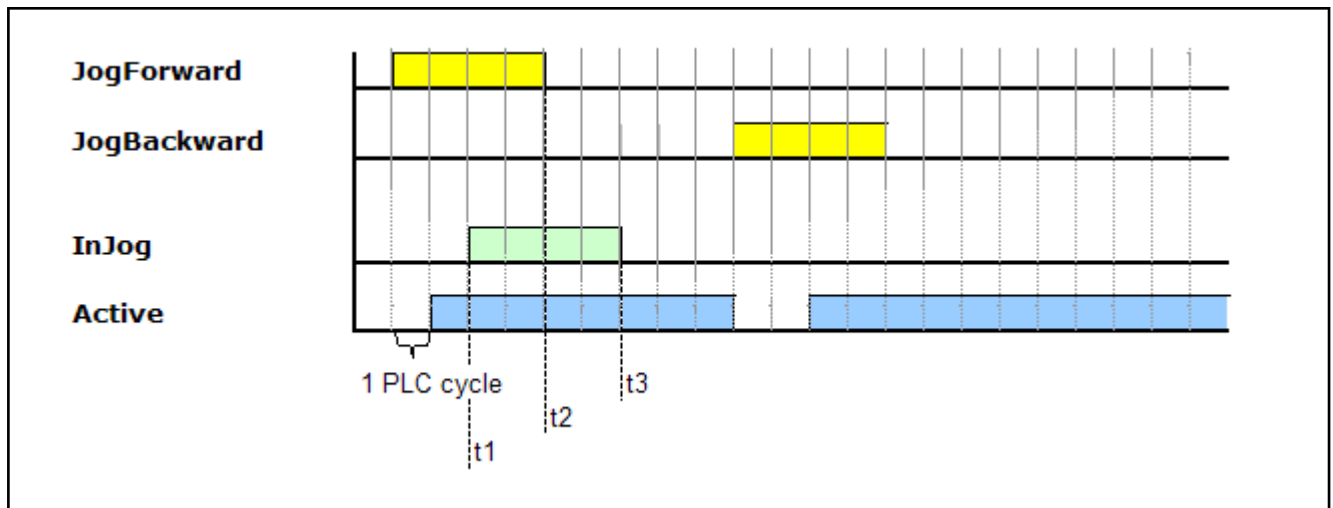
Eingangs-/Ausgangs-Variable (VAR_IN_OUT)	Datentyp	Beschreibung
Axis	AXIS_REF	Definiert die Achse. Folgende Achsen sind möglich: - Definiert die Achse. Folgende Achsen sind möglich: MLD-S: Axis1 MLD-M: - Axis1 (Master) oder - Axis2...Axis10 (insgesamt 9 Slaves / entfernte Achsen) Die in der Liste "P-0-1801.0.10, CCD: Adressen projektierter Antriebe" vorhandenen entfernten Achsen entsprechen der Reihenfolge nach Axis2 bis Axis10 (Axis2 wirkt auf den Antrieb des Listenelements 0 des P-0-1801.0.10,...) "VmAxis1" entspricht dem virtuellen Leitachsgenerator

Tab. 13-66: *Eingangs-/Ausgangs-Variable des Firmware-Funktionsbausteins "MC_Jog"*

Timingdiagramme

Die folgende Abbildung zeigt das Timingdiagramm bei aktivem Tippbetrieb ("Tippen positiv" bzw. "Tippen negativ"). Durch eine positive Flanke am Eingang "JogForward" werden die Eingangsparameter ("JogVelo" usw.) übernommen und in der Betriebsart "Antriebsgeführtes Positionieren" das Endlosfahren in positive Richtung (Tippen positiv) aktiviert. Der Ausgang "InJog" wird gesetzt, wenn die Istgeschwindigkeit größer als das Stillstandsfenster (S-0-0124) ist; "InJog" wird wieder weggenommen, wenn die Istgeschwindigkeit kleiner als das Stillstandsfenster ist. Bei dem virtuellen Leitachsgenerator wird "P-0-0912, Virtuelle Leitachse, Stillstandsfenster" verwendet.

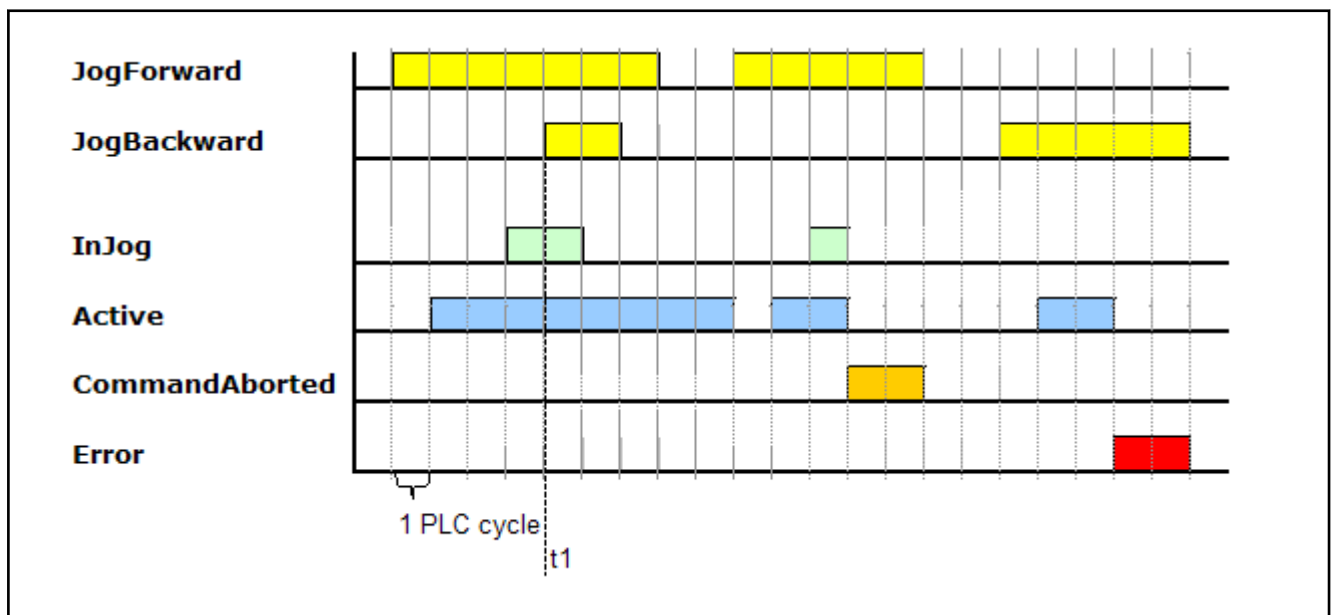
Der Ausgang "Active" bleibt solange gesetzt bis der Funktionsbaustein neu gestartet (kommandiert) wird bzw. durch eine andere Funktionsbaustein-Instanz (z. B. "MC_MoveAbsolute") unterbrochen wird.



- t1** Istgeschwindigkeit außerhalb Stillstandsfenster (reale Achse: S-0-0124 / virtuelle Leitachse: P-0-0912)
- t2** Achse wird gestoppt
- t3** Istgeschwindigkeit innerhalb Stillstandsfenster (reale Achse: S-0-0124 / virtuelle Leitachse: P-0-0912)

Abb. 13-29: Timing des "Active"-Ausgangs

Die folgende Abbildung zeigt das Timingverhalten der weiteren Standardausgänge.



- t1** Achse wird gestoppt

Abb. 13-30: Timing "Active"-, "CommandAborted"- und "Error"-Ausgang

13.3.19 MC_MoveAbsolute

Der Firmware-Funktionsbaustein "MC_MoveAbsolute" dient dazu, den Antrieb auf eine vorgegebene Absolutposition zu verfahren. Intern wird dazu die Betriebsart "Antriebsgeführtes Positionieren" aktiviert. Einstellungen des Lagereglers (mit/ohne Schleppfehler, Lageistwert 1 / 2) müssen mit dem Baustein "MB_SetPositionControlMode" getroffen werden.

MX_PLCoen

Kommandieren einer virtuellen
Leitachse

Bei permanenter Kontrolle und bei entfernten Achsen wird die Nebenbetriebsart 3 und deren Parameter verwendet. Bei temporärer Kontrolle wird die SPS-Nebenbetriebsart 11 und deren Parameter verwendet.
Die Zielposition muss an dem Eingang "Position" vorgegeben werden. Über die Eingänge "Velocity", "Acceleration" und "Deceleration" muss bestimmt werden wie die Zielposition angefahren wird (Bewegungsprofil).
Soll über den Firmware-Funktionsbaustein "MC_MoveAbsolute" ein virtueller Leitachsgenerator absolut positioniert werden, dann muss der Leitachsgenerator über "P-0-0917, Steuerwort Leitachsgenerator", Bit0="1" aktiviert werden. Um den virtuellen Lageistwert (P-0-0758) in die interne virtuelle Leitachsposition (VmAxis1; "P-0-0761, Leitachsposition") zu wandeln, muss "P-0-0758, Virtueller Lageistwert" in den Parameter "P-0-0916, Leitachsformatwandler Signalauswahl" geschrieben werden (siehe auch Firmware-Funktionsbeschreibung "Leitachsgenerator" und "Leitachsformatwandler").

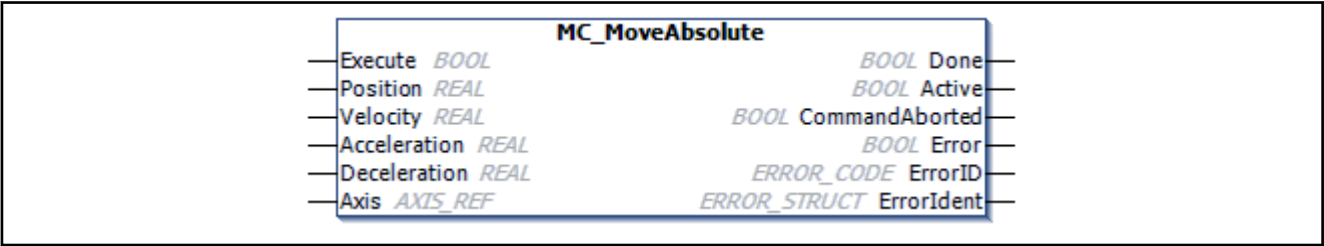


Abb. 13-31: Firmware-Funktionsbaustein "MC_MoveAbsolute"

-
- Die folgenden Parameter dürfen nicht zyklisch beschrieben werden, da sie durch diesen und ggf. weitere Funktionsbausteine beschrieben werden:
- S-0-0346, Steuerwort Positionieren
 - P-0-1454, SPS/Einrichtbetrieb Positioniersollwert-Übernahme
 - P-0-0760, Virtuelle Leitachse, Positioniersteuerwort

Input-Variable	Datentyp	Beschreibung	Verwendete Parameter		
Execute	BOOL	siehe "Signallaufverhalten"	-		
			①	②	③
Position	REAL	Am Eingang "Position" muss die absolute Position vorgegeben werden.	S-0-0259	P-0-1450	P-0-0766
Velocity	REAL	Positioniergeschwindigkeit in physikalischer Einheit*1)	S-0-0259	P-0-1451	P-0-0770
Acceleration	REAL	Positionierbeschleunigung in physikalischer Einheit*1)	S-0-0260	P-0-1452	P-0-0771
Deceleration	REAL	Positionierverzögerung in physikalischer Einheit*1)	S-0-0359	P-0-1453	P-0-0772

①

Wenn SPS permanente Kontrolle über die Achse hat oder wenn eine entfernte Achse kommandiert wird

②

Wenn SPS temporäre Kontrolle über die Achse hat

③

Bei virtueller Leitachse

*1)

Die Anzahl der möglichen Nachkommastellen richtet sich nach der parametrisierten Wichtung (siehe Antriebsdokumentation). Hat der angelegte Wert mehr Nachkommastellen, so werden diese abgeschnitten.

Tab. 13-67: Input-Variablen des Firmware-Funktionsbausteins "MC_MoveAbsolute"

Output-Variable	Datentyp	Beschreibung
Done	BOOL	Mit "Done"=TRUE meldet der Firmware-Funktionsbaustein, dass die absolute Zielposition erreicht wurde. Die Abarbeitung ist dann bis zur nächsten 0-1-Flanke am "Execute"-Eingang beendet. Gibt man eine 1-0-Flanke auf den "Execute"-Eingang bevor der Firmware-Funktionsbaustein meldet dass er fertig ist ("Done"=TRUE), wird der Ausgang "Done" für einen Aufruf auf TRUE gesetzt, wenn das Bewegungsziel erreicht wurde und die Abarbeitung des Firmware-Funktionsbausteins nicht durch einen Fehler oder durch einen anderen Firmware-Funktionsbaustein unterbrochen wurde. - reale Achsen: Der Ausgang wird auf TRUE gesetzt, wenn das Bit 2 im Parameter "S-0-0437, Positionier-Status" gesetzt wird. - virtuelle Leitachse: Der Ausgang wird auf TRUE gesetzt, wenn das Bit 2 im Parameter "P-0-0768, Virtuelle Leitachse, Positionierstatus" gesetzt wird.
Active	BOOL	Mit "Active"=TRUE zeigt der Funktionsbaustein an, dass er die Achsbewegung bestimmt.
CommandAborted	BOOL	Mit "CommandAborted"=TRUE zeigt der Funktionsbaustein an, dass ein anderer Motion-Baustein aktiviert wurde oder dass ein Antriebsfehler vorlag. Ein Antriebsfehler wird über die Error-Ausgänge des "MC_Power" angezeigt
Error	BOOL	Mit "Error"=TRUE meldet der Funktionsbaustein, dass ein Fehler vorliegt
ErrorID *	ERROR_CODE	"ErrorID" gibt eine grobe Fehlerbeschreibung wieder.
ErrorIdent *	ERROR_STRUCT	"ErrorIdent" gibt über die Elemente: "Table", "Additional1" und "Additional2" die genaue Fehlerursache wieder.

* siehe nachfolgender Hinweis

Tab. 13-68: *Output-Variablen des Firmware-Funktionsbausteins "MC_MoveAbsolute"*



HINWEIS: Eine Auflistung aller möglichen Fehler finden Sie hier: ["Fehler-Referenzlisten"](#)

Eingangs-/Ausgangs-Variable (VAR_IN_OUT)	Datentyp	Beschreibung
Axis	AXIS_REF	Definiert die Achse. Folgende Achsen sind möglich: - Definiert die Achse. Folgende Achsen sind möglich: MLD-S: Axis1 MLD-M: - Axis1 (Master) oder - Axis2...Axis10 (insgesamt 9 Slaves / entfernte Achsen) Die in der Liste "P-0-1801.0.10, CCD: Adressen projektierter Antriebe" vorhandenen entfernten Achsen entsprechen der Reihenfolge nach Axis2 bis Axis10 (Axis2 wirkt auf den Antrieb des Listenelements 0 des P-0-1801.0.10,...) "VmAxis1" entspricht dem virtuellen Leitachsgenerator

Tab. 13-69: *Eingangs-/Ausgangs-Variable des Firmware-Funktionsbausteins "MC_MoveAbsolute"*

MX_PLCopn

13.3.20 MC_MoveAdditive

Der Firmware-Funktionsbaustein "MC_MoveAdditive" dient dazu, den Antrieb relativ um einen Weg zusätzlich zur Zielposition zu verfahren. Die neue Zielposition ergibt sich also aus der alten Zielposition, addiert um eine Distanz. Der Weg (die Distanz) wird durch den Eingang "Distance" bestimmt. Mit den Eingängen "Velocity", "Acceleration" und "Deceleration" wird das Bewegungsprofil bestimmt.

Intern wird die Betriebsart "Antriebsgeführtes Positionieren" aktiviert. Einstellungen des Lagereglers (mit/ohne Schleppfehler, Lageistwert 1 / 2) müssen mit dem Baustein "MB_SetPositionControlMode" getroffen werden.

Bei permanenter Kontrolle und bei entfernten Achsen werden die Nebenbetriebsart 3 und deren Parameter verwendet. Bei temporärer Kontrolle werden die SPS-Nebenbetriebsart 11 und deren Parameter verwendet.

Kommandieren einer virtuellen Leitachse

Soll über den Firmware-Funktionsbaustein "MC_MoveAdditive" ein virtueller Leitachsgenerator relativ um einen Weg **zusätzlich** zur aktuellen Zielposition verfahren werden, dann muss der Leitachsgenerator über "P-0-0917, Steuerwort Leitachsgenerator", Bit0="1" aktiviert werden. Um den virtuellen Lageistwert (P-0-0758) in die interne virtuelle Leitachspolition (VmAxis1; "P-0-0761, Leitachspolition") zu wandeln, muss "P-0-0758, Virtueller Lageistwert" in den Parameter "P-0-0916, Leitachsformatwandler Signalauswahl" geschrieben werden (siehe auch Firmware-Funktionsbeschreibung "Leitachsgenerator" und "Leitachsformatwandler").

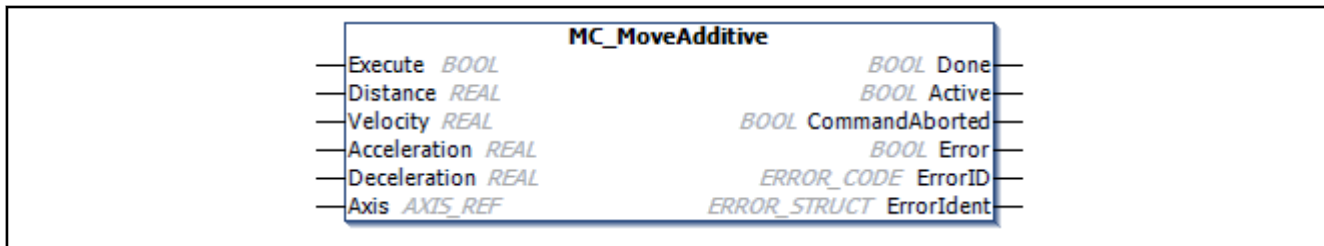


Abb. 13-32: Firmware-Funktionsbaustein "MC_MoveAdditive"



Die folgenden Parameter dürfen nicht zyklisch beschrieben werden, da sie durch diesen und ggf. weitere Funktionsbausteine beschrieben werden:

- S-0-0346, Steuerwort Positionieren
- P-0-1454, SPS/Einrichtbetrieb Positioniersollwert-Übernahme
- P-0-0760, Virtuelle Leitachse, Positioniersteuerwort

Input-Variable	Datentyp	Beschreibung	Verwendete Parameter		
Execute	BOOL	siehe "Signallaufverhalten"	-		
			①	②	③
Distance	REAL	Am Eingang "Distance" muss der relative Weg (Distanz) vorgegeben werden; die neue Zielposition ergibt sich aus der alten Zielposition und der Vorgabe am Eingang "Distance".	S-0-0282	P-0-1450	P-0-0766
Velocity	REAL	Positioniergeschwindigkeit in physikalischer Einheit* ¹⁾	S-0-0259	P-0-1451	P-0-0770
Acceleration	REAL	Positionierbeschleunigung in physikalischer Einheit* ¹⁾	S-0-0260	P-0-1452	P-0-0771
Deceleration	REAL	Positioniervverzögerung in physikalischer Einheit* ¹⁾	S-0-0359	P-0-1453	P-0-0772

- ① Wenn SPS permanente Kontrolle über die Achse hat oder wenn eine entfernte Achse kommandiert wird
 ② Wenn SPS temporäre Kontrolle über die Achse hat
 ③ Bei virtueller Leitachse
 *¹⁾ Die Anzahl der möglichen Nachkommastellen richtet sich nach der parametrisierten Wichtung (siehe Antriebsdokumentation). Hat der angelegte Wert mehr Nachkommastellen, so werden diese abgeschnitten.

Tab. 13-70: Input-Variablen des Firmware-Funktionsbausteins "MC_MoveAdditive"

Output-Variable	Datentyp	Beschreibung
Done	BOOL	Mit "Done"=TRUE meldet der Firmware-Funktionsbaustein, dass die Zielposition erreicht wurde. Die Abarbeitung ist dann bis zur nächsten 0-1-Flanke am "Execute"-Eingang beendet. Gibt man eine 1-0-Flanke auf den "Execute"-Eingang bevor der Firmware-Funktionsbaustein meldet dass er fertig ist ("Done"=TRUE), wird der Ausgang "Done" für einen Aufruf auf TRUE gesetzt, wenn das Bewegungsziel erreicht wurde und die Abarbeitung des Firmware-Funktionsbausteins nicht durch einen Fehler oder durch einen anderen Firmware-Funktionsbaustein unterbrochen wurde. - reale Achsen: Der Ausgang wird auf TRUE gesetzt, wenn das Bit 2 im Parameter "S-0-0437, Positionier-Status" gesetzt wird. - virtuelle Leitachse: Der Ausgang wird auf TRUE gesetzt, wenn das Bit 2 im Parameter P-0-0768; Virtuelle Leitachse, Positionierstatus gesetzt wird.
Active	BOOL	Mit "Active"=TRUE zeigt der Firmware-Funktionsbaustein an, dass er die Achsbewegung bestimmt.
CommandAborted	BOOL	Mit "CommandAborted"=TRUE zeigt der Firmware-Funktionsbaustein an, dass ein anderer Motion-Funktionsbaustein aktiviert wurde oder dass ein Antriebsfehler vorlag. Ein Antriebsfehler wird über die Error-Ausgänge des "MC_Power" angezeigt.
Error	BOOL	Mit "Error"=TRUE meldet der Funktionsbaustein, dass ein Fehler vorliegt
ErrorID *	ERROR_CODE	"ErrorID" gibt eine grobe Fehlerbeschreibung wieder.
ErrorIdent *	ERROR_STRUCT	"ErrorIdent" gibt über die Elemente: "Table", "Additional1" und "Additional2" die genaue Fehlerursache wieder.

* siehe nachfolgender Hinweis

Tab. 13-71: Output-Variablen des Firmware-Funktionsbausteins "MC_MoveAdditive"

MX_PLCoPen



HINWEIS: Eine Auflistung aller möglichen Fehler finden Sie hier:
["Fehler-Referenzlisten"](#)

Eingangs-/Ausgangs-Variable (VAR_IN_OUT)	Datentyp	Beschreibung
Axis	AXIS_REF	Definiert die Achse. Folgende Achsen sind möglich: - Definiert die Achse. Folgende Achsen sind möglich: MLD-S: Axis1 MLD-M: - Axis1 (Master) oder - Axis2...Axis10 (insgesamt 9 Slaves / entfernte Achsen) Die in der Liste "P-0-1801.0.10, CCD: Adressen projektierter Antriebe" vorhandenen entfernten Achsen entsprechen der Reihenfolge nach Axis2 bis Axis10 (Axis2 wirkt auf den Antrieb des Listenelements 0 des P-0-1801.0.10,...) "VmAxis1" entspricht dem virtuellen Leitachsgenerator

Tab. 13-72: *Eingangs-/Ausgangs-Variable des Firmware-Funktionsbausteins "MC_MoveAdditive"*

13.3.21 MC_MoveRelative

Der Firmware-Funktionsbaustein "MC_MoveRelative" dient dazu, den Antrieb ab der aktuellen Istposition relativ um einen Weg zu verfahren.

Intern wird dazu die Betriebsart "Antriebsgeführtes Positionieren" aktiviert. Einstellungen des Lagereglers (mit/ohne Schleppfehler, Lageistwert 1 / 2) müssen mit dem Baustein "MB_SetPositionControlMode" getroffen werden.

Bei permanenter Kontrolle und bei entfernten Achsen werden die Nebenbetriebsart 3 und deren Parameter verwendet. Bei temporärer Kontrolle werden die SPS-Nebenbetriebsart 11 und deren Parameter verwendet.

Der zu verfahrenende relative Weg muss an dem Eingang "Distance" vorgegeben werden. Über die Eingänge "Velocity", "Acceleration" und "Deceleration" muss das Bewegungsprofil bestimmt werden (wie der relative Weg verfahren wird).

Kommandieren einer virtuellen Leitachse

Soll über den Firmware-Funktionsbaustein "MC_MoveRelative" ein virtueller Leitachsgenerator ab der aktuellen virtuellen Istposition relativ um einen Weg verfahren werden, dann muss der Leitachsgenerator über "P-0-0917, Steuerwort Leitachsgenerator", Bit0="1" aktiviert werden. Um den virtuellen Lageistwert (P-0-0758) in die interne virtuelle Leitachsposition (VmAxis1; "P-0-0761, Leitachsposition") zu wandeln, muss "P-0-0758, Virtueller Lageistwert" in den Parameter "P-0-0916, Leitachsformatwandler Signalauswahl" geschrieben werden (siehe auch Firmware-Funktionsbeschreibung "Leitachsgenerator" und "Leitachsformatwandler").



Beachten Sie, dass die Sollwertvorgabe bei Modulowichtung des virtuellen Leitachsgenerators (P-0-0756) von "P-0-0769, Virtuelle Leitachse, Sollwertmodus" vorgegeben ist.

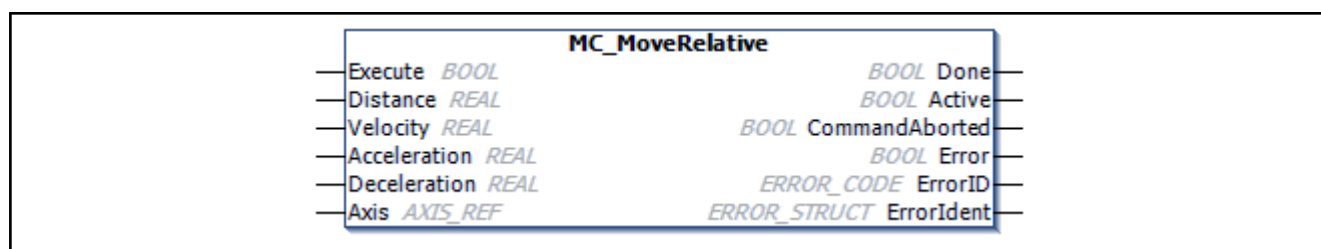


Abb. 13-33: Firmware-Funktionsbaustein "MC_MoveRelative"



Die folgenden Parameter dürfen nicht zyklisch beschrieben werden, da sie durch diesen und ggf. weitere Funktionsbausteine beschrieben werden:

- S-0-0346, Steuerwort Positionieren
- P-0-1454, SPS/Einrichtbetrieb Positioniersollwert-Übernahme
- P-0-0760, Virtuelle Leitachse, Positioniersteuerwort

Input-Variable	Datentyp	Beschreibung	Verwendete Parameter		
Execute	BOOL	siehe "Signallaufverhalten"	-		
			①	②	③
Distance	REAL	Am Eingang "Distance" muss der relative Weg vorgegeben werden, welcher ab der aktuellen Istposition verfahren werden soll ^{*1)}	S-0-0282	P-0-1450	P-0-0766
Velocity	REAL	Positioniergeschwindigkeit in physikalischer Einheit ^{*1)}	S-0-0259	P-0-1451	P-0-0770
Acceleration	REAL	Positionierbeschleunigung in physikalischer Einheit ^{*1)}	S-0-0260	P-0-1452	P-0-0771
Deceleration	REAL	Positionierverzögerung in physikalischer Einheit ^{*1)}	S-0-0359	P-0-1453	P-0-0772

- ① Wenn SPS permanente Kontrolle über die Achse hat oder wenn eine entfernte Achse kommandiert wird
- ② Wenn SPS temporäre Kontrolle über die Achse hat
- ③ Bei virtueller Leitachse
- *1) Die Anzahl der möglichen Nachkommastellen richtet sich nach der parametrisierten Wichtung (siehe Antriebsdokumentation). Hat der angelegte Wert mehr Nachkommastellen, so werden diese abgeschnitten.

Tab. 13-73: Input-Variablen des Firmware-Funktionsbausteins "MC_MoveRelative"

MX_PLCoPen

Output-Variable	Datentyp	Beschreibung
Done	BOOL	Mit "Done"=TRUE meldet der Firmware-Funktionsbaustein, dass die vorgegebene Distanz verfahren wurde. Die Abarbeitung ist dann bis zur nächsten 0-1-Flanke am "Execute"-Eingang beendet. Gibt man eine 1-0-Flanke auf den "Execute"-Eingang bevor der Firmware-Funktionsbaustein meldet dass er fertig ist ("Done"=TRUE), wird der Ausgang "Done" für einen Aufruf auf TRUE gesetzt, wenn das Bewegungsziel erreicht wurde und die Abarbeitung des Firmware-Funktionsbausteins nicht durch einen Fehler oder durch einen anderen Firmware-Funktionsbaustein unterbrochen wurde. - reale Achsen: Der Ausgang wird auf TRUE gesetzt, wenn das Bit 2 im Parameter "S-0-0437, Positionier-Status" gesetzt wird. - virtuelle Leitachse: Der Ausgang wird auf TRUE gesetzt, wenn das Bit 2 im Parameter "P-0-0768, Virtuelle Leitachse, Positionierstatus" gesetzt wird.
Active	BOOL	Mit TRUE zeigt der Ausgang "Active" an, dass er die Achsbewegung bestimmt
CommandAborted	BOOL	Mit "CommandAborted"=TRUE zeigt der Funktionsbaustein an, dass ein anderer Motion-Baustein aktiviert wurde oder dass ein Antriebsfehler vorlag. Ein Antriebsfehler wird über die Error-Ausgänge des "MC_Power" angezeigt
Error	BOOL	Mit "Error"=TRUE meldet der Funktionsbaustein, dass ein Fehler vorliegt
ErrorID *	ERROR_CODE	"ErrorID" gibt eine grobe Fehlerbeschreibung wieder.
ErrorIdent *	ERROR_STRUCT	"ErrorIdent" gibt über die Elemente: "Table", "Additional1" und "Additional2" die genaue Fehlerursache wieder.

* siehe nachfolgender Hinweis

Tab. 13-74: *Output-Variablen des Firmware-Funktionsbausteins "MC_MoveRelative"*



HINWEIS: Eine Auflistung aller möglichen Fehler finden Sie hier: ["Fehler-Referenzlisten"](#)

Eingangs-/Ausgangs-Variable (VAR_IN_OUT)	Datentyp	Beschreibung
Axis	AXIS_REF	Definiert die Achse. Folgende Achsen sind möglich: - Definiert die Achse. Folgende Achsen sind möglich: MLD-S: Axis1 MLD-M: - Axis1 (Master) oder - Axis2...Axis10 (insgesamt 9 Slaves / entfernte Achsen) Die in der Liste "P-0-1801.0.10, CCD: Adressen projektierter Antriebe" vorhandenen entfernten Achsen entsprechen der Reihenfolge nach Axis2 bis Axis10 (Axis2 wirkt auf den Antrieb des Listenelements 0 des P-0-1801.0.10,...) "VmAxis1" entspricht dem virtuellen Leitachsgenerator

Tab. 13-75: *Eingangs-/Ausgangs-Variable des Firmware-Funktionsbausteins "MC_MoveRelative"*

13.3.22 MC_MoveVelocity

Der Firmware-Funktionsbaustein "MC_MoveVelocity" dient dazu, dem Antrieb eine Geschwindigkeit vorzugeben. Da der Antrieb in Geschwindigkeitsregelung verfährt, kann der Firmware-Funktionsbaustein auch bei Open-Loop-Antrieben verwendet werden.

Bei permanenter Kontrolle und bei entfernten Achsen wird die Hauptbetriebsart und deren Parameter verwendet. Bei temporärer Kontrolle wird die SPS-Nebenbetriebsart 8 und deren Parameter verwendet.



Die Einheiten der Eingänge, welche die Antriebsbewegung definieren, sind falls sie wichtungsabhängig sind und nicht anders beschrieben sind, den Wichtungsparametern zu entnehmen.

- S-0-0044, Wichtungsart für Geschwindigkeitsdaten
- S-0-0076, Wichtungsart für Lagedaten
- S-0-0086, Wichtungsart für Drehmoment-/Kraftdaten

Kommandieren einer virtuellen Leitachse

Soll über den Firmware-Funktionsbaustein "MC_MoveVelocity" einem virtuellen Leitachsgenerator eine Geschwindigkeit vorgegeben werden, dann muss der Leitachsgenerator über "P-0-0917, Steuerwort Leitachsgenerator", Bit0="1" aktiviert werden. Um den virtuellen Lageistwert (P-0-0758) in die interne virtuelle Leitachsposition (VmAxis1; "P-0-0761, Leitachsposition") zu wandeln, muss "P-0-0758, Virtueller Lageistwert" in den Parameter "P-0-0916, Leitachsformatwandler Signalauswahl" geschrieben werden (siehe auch Firmware-Funktionsbeschreibung "Leitachsgenerator" und "Leitachsformatwandler").



Beachten Sie, dass die Sollwertvorgabe bei Modulowichtung des virtuellen Leitachsgenerators (P-0-0756) von "P-0-0769, Virtuelle Leitachse, Sollwertmodus" vorgegeben ist.

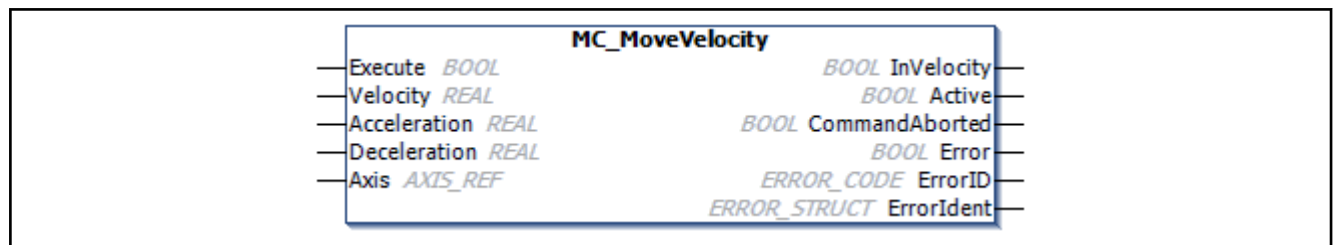


Abb. 13-34: Firmware-Funktionsbaustein "MC_MoveVelocity"

MX_PLCOpen

Input-Variable	Datentyp	Beschreibung	Verwendete Parameter		
Execute	BOOL	siehe "Signallaufverhalten"	-		
			①	②	③
Velocity	REAL	Positioniergeschwindigkeit in physikalischer Einheit ^{*1)}	S-0-0036	P-0-1460	P-0-0770
Acceleration	REAL	Positionierbeschleunigung in physikalischer Einheit ^{*1)}	P-0-1203	P-0-1461	P-0-0771
Deceleration	REAL	Positionierverzögerung in physikalischer Einheit ^{*1)}	P-0-1213	P-0-1463	P-0-0772

- ① Wenn SPS permanente Kontrolle über die Achse hat oder wenn eine entfernte Achse kommandiert wird
- ② Wenn SPS temporäre Kontrolle über die Achse hat
- ③ Bei virtueller Leitachse
- *1) Die Anzahl der möglichen Nachkommastellen richtet sich nach der parametrisierten Wichtung (siehe Antriebsdokumentation). Hat der angelegte Wert mehr Nachkommastellen, so werden diese abgeschnitten.

Tab. 13-76: *Input-Variablen des Firmware-Funktionsbausteins "MC_MoveVelocity"*

Output-Variable	Datentyp	Beschreibung
InVelocity	BOOL	Mit "InVelocity"=TRUE meldet der Firmware-Funktionsbaustein, dass die durch "Velocity" vorgegebene Geschwindigkeit erreicht wurde. Gibt man eine 1-0-Flanke auf den "Execute"-Eingang bevor der Firmware-Funktionsbaustein meldet, dass er die Sollgeschwindigkeit erreicht hat ("InVelocity"=TRUE), wird der Ausgang "InVelocity" für einen Aufruf auf TRUE gesetzt, wenn die Istgeschwindigkeit der Sollgeschwindigkeit entspricht. - reale Achsen: "Axis1" usw. InVelocity = "P-0-0115, Gerätesteuerung Statuswort", Bit12 der parametrisierten Achse. Bit12 vom P-0-0115 entspricht in diesem Anwendungsfall "S-0-0330, Meldung n_ist = n_soll". - virtuelle Leitachse: "VmAxis1" InVelocity = Bit4 AND Bit13 von "P-0-0768, Virtuelle Leitachse, Positionierstatus". - Bit4="1" → Generator liefert konst. Geschwindigkeit - Bit13="1" → $P-0-0770 - P-0-0759 < 1 \text{ Upm bzw. } < 1 \text{ mm/min}$
Active	BOOL	Mit "Active"=TRUE zeigt der Funktionsbaustein an, dass er die Achsbewegung bestimmt. "Active" bleibt TRUE, bis "CommandAborted" oder "Error" gesetzt wird.
CommandAborted	BOOL	Mit "CommandAborted"=TRUE zeigt der Funktionsbaustein an, dass ein anderer Motion-Baustein aktiviert wurde oder dass ein Antriebsfehler vorlag. Ein Antriebsfehler wird über die Error-Ausgänge des "MC_Power" angezeigt
Error	BOOL	Mit "Error"=TRUE meldet der Funktionsbaustein, dass ein Fehler vorliegt
ErrorID *	ERROR_CODE	"ErrorID" gibt eine grobe Fehlerbeschreibung wieder.
ErrorIdent *	ERROR_STRUCT	"ErrorIdent" gibt über die Elemente: "Table", "Additional1" und "Additional2" die genaue Fehlerursache wieder.

* siehe nachfolgender Hinweis

Tab. 13-77: *Output-Variablen des Firmware-Funktionsbausteins "MC_MoveVelocity"*



HINWEIS: Eine Auflistung aller möglichen Fehler finden Sie hier:
["Fehler-Referenzlisten"](#)

Eingangs-/Ausgangs-Variable (VAR_IN_OUT)	Datentyp	Beschreibung
Axis	AXIS_REF	Definiert die Achse. Folgende Achsen sind möglich: - Definiert die Achse. Folgende Achsen sind möglich: MLD-S: Axis1 MLD-M: - Axis1 (Master) oder - Axis2...Axis10 (insgesamt 9 Slaves / entfernte Achsen) Die in der Liste "P-0-1801.0.10, CCD: Adressen projektierter Antriebe" vorhandenen entfernten Achsen entsprechen der Reihenfolge nach Axis2 bis Axis10 (Axis2 wirkt auf den Antrieb des Listenelements 0 des P-0-1801.0.10,...) "VmAxis1" entspricht dem virtuellen Leitachsgenerator

Tab. 13-78: Eingangs-/Ausgangs-Variable des Firmware-Funktionsbausteins "MC_MoveVelocity"

13.3.23 MC_Power

Hat die interne SPS permanente oder temporäre Kontrolle über den Antrieb, kann mit diesem Funktionsbaustein die Reglerfreigabe aktiviert werden. Unabhängig vom Eingang "Enable" liefert der Ausgang "Status" den aktuellen Antriebszustand zurück.

Der Funktionsbaustein "MC_Power" setzt das SPS-interne Signal "EIN" auf TRUE und schaltet somit den Antrieb ein.



Bei dem "EIN"-Signal handelt es sich um das "EIN"-Signal der SPS, welches ggf. verzögert (max. 1ms bei ADVANCED, max. 2ms bei BASIC bzw. IndraDrive Mi) im Antrieb mit dem externen Signal verknüpft aktiv wird; beim Erzeugen von mehreren Flanken (z. B. "TRUE - FALSE - TRUE") ist diese Wartezeit zu berücksichtigen.

Wenn in der SPS ein WATCHDOG oder eine Exception auftritt, wird automatisch das SPS-interne "EIN"-Signal auf FALSE gesetzt und bis zum nächsten RESET der SPS auf FALSE gehalten.

Weitere Aufrufe von "MC_Power" mit TRUE melden Fehler.

Ist der Antrieb noch nicht bereit zur Leistungszuschaltung (z. B. nicht in "Ab"), so liefert der Ausgang "Status" FALSE und der Ausgang "Error" ist TRUE.



Ob der Antrieb zur Leistungszuschaltung bereit ist, können Sie z. B. mit der Abfrage `IF AxisData[Axis1.AxisNo].Axis_InAb THEN` prüfen; für diese Abfrage muss in der SPS-Konfiguration (P-0-1367) die Unterstützung der AxisData-Struktur aktiviert sein.

MX_PLCOpen



"MC_Power" liefert auch einen Fehler, wenn ein Motion-Funktionsbaustein einen Fehler erkannt und die Achse stillgelegt hat.

Nach jedem Fehler ("Error"=TRUE) muss am Eingang "Enable" eine neue 0-1-Flanke angelegt werden, um die Reglerfreigabe im Antrieb zu aktivieren.

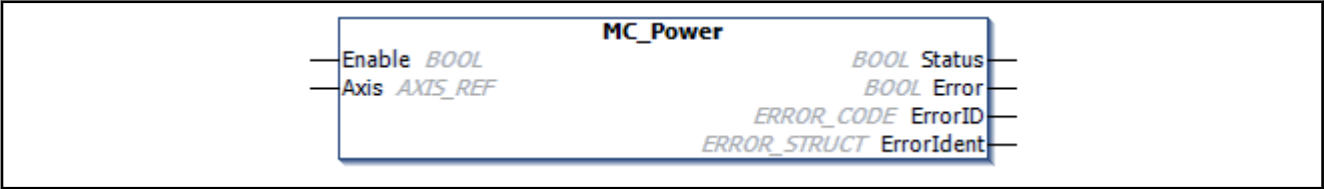


Abb. 13-35: Firmware-Funktionsbaustein "MC_Power"

Eingangs-Variable	Datentyp	Beschreibung
Enable	BOOL	Über den Eingang "Enable" kann dem Antrieb die Reglerfreigabe erteilt werden. Aus Sicherheitsgründen wirkt der Eingang flankengesteuert. Mit einer steigenden Flanke wird die Reglerfreigabe erteilt und mit einer fallenden Flanke wird die Reglerfreigabe wieder gesperrt.

Tab. 13-79: Eingangs-Variablen des Firmware-Funktionsbausteins "MC_Power"

Ausgangs-Variable	Datentyp	Beschreibung
Status	BOOL	Ist der Ausgang "Status"=TRUE ist die Reglerfreigabe aktiv und der Antrieb ist momentanbehaftet. Unabhängig vom Eingang "Enable" liefert der Ausgang "Status" den aktuellen Antriebszustand zurück. Dazu wird intern die Kombination "Antrieb in Regelung und kein Fehler" geprüft. Ausgang "Status" := (P115.14 AND P115.15 AND NOT P115.13) Somit geht der Ausgang "Status" bei Antriebsfehlern auf FALSE
Error	BOOL	Mit "Error"=TRUE meldet der Funktionsbaustein, dass ein Fehler vorliegt
ErrorID *	ERROR_CODE	"ErrorID" gibt eine grobe Fehlerbeschreibung wieder.
ErrorIdent *	ERROR_STRUCT	"ErrorIdent" gibt über die Elemente: "Table", "Additional1" und "Additional2" die genaue Fehlerursache wieder.

* siehe nachfolgender Hinweis

Tab. 13-80: Ausgangs-Variablen des Firmware-Funktionsbausteins "MC_Power"



HINWEIS: Eine Auflistung aller möglichen Fehler finden Sie hier: ["Fehler-Referenzlisten"](#)

Eingangs-/Ausgangs-Variable (VAR_IN_OUT)	Datentyp	Beschreibung
Axis	AXIS_REF	Definiert die Achse. Folgende Achsen sind möglich: MLD-S: Axis1 MLD-M: - Axis1 (Master) oder - Axis2...Axis10 (insgesamt 9 Slaves / entfernte Achsen) Die in der Liste "P-0-1801.0.10, CCD: Adressen projektierter Antriebe" vorhandenen entfernten Achsen entsprechen der Reihenfolge nach Axis2 bis Axis10 (Axis2 wirkt auf den Antrieb des Listenelements 0 des P-0-1801.0.10,...)

Tab. 13-81: Eingangs-/Ausgangs-Variable des Firmware-Funktionsbausteins "MC_Power"

13.3.24 MC_ReadStatus

Mit dem Funktionsbaustein "MC_ReadStatus" kann der aktuelle Status der Achse gelesen werden. Der Status der Achse wird dekodiert ausgegeben.



Eine Änderung der Achse am Eingang des Funktionsbausteines, setzt einen erneuten 0/1-Übergang an "Enable" voraus.

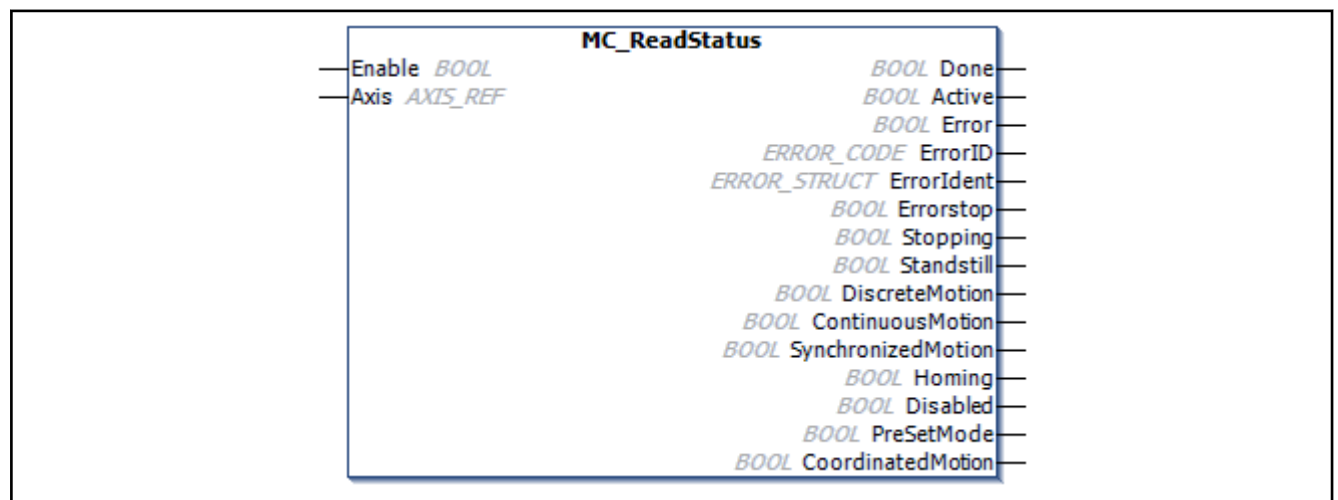


Abb. 13-36: Firmware-Funktionsbaustein "MC_ReadStatus"

Eingangs-Variable	Datentyp	Beschreibung
Enable	BOOL	siehe "Signallaufverhalten"

Tab. 13-82: Eingangs-Variable des Firmware-Funktionsbausteins "MC_ReadStatus"

MX_PLCOpen

Ausgangs-Variable	Datentyp	Beschreibung
Done	BOOL	Gültiger Wert ist verfügbar
Active	BOOL	Bearbeiten der Daten nach Abschluss der Vorverarbeitung läuft
Error	BOOL	Mit "Error"=TRUE meldet der Funktionsbaustein, dass ein Fehler vorliegt
ErrorID *	ERROR_CODE	"ErrorID" gibt eine grobe Fehlerbeschreibung wieder.
ErrorIdent *	ERROR_STRUCT	"ErrorIdent" gibt über die Elemente: "Table", "Additional1" und "Additional2" die genaue Fehlerursache wieder.
Errorstop	BOOL	Antrieb/Achse im Zustand "Errorstop"
Stopping	BOOL	Antrieb/Achse im Zustand "Stopping"
StandStill	BOOL	Antrieb/Achse im Zustand "StandStill"
DiscreteMotion	BOOL	Antrieb/Achse im Zustand "DiscreteMotion"
ContinuousMotion	BOOL	Antrieb/Achse im Zustand "ContinuousMotion"
SynchronizedMotion	BOOL	Antrieb/Achse im Zustand "SynchronizedMotion"
Homing	BOOL	Antrieb im Zustand "Homing"
Disabled	BOOL	Antrieb/Achse im Zustand "Disabled"
PreSetMode	BOOL	Antrieb im Zustand "PreSetMode"
CoordinatedMotion	BOOL	Antrieb/Achse im Zustand "CoordinatedMotion"; wird von dem Zielsystem "IndraDrive" nicht unterstützt!

* siehe nachfolgender Hinweis

Tab. 13-83: *Ausgangs-Variablen des Firmware-Funktionsbausteins "MC_ReadStatus"*



HINWEIS: Eine Auflistung aller möglichen Fehler finden Sie hier: ["Fehler-Referenzlisten"](#)

Eingangs-/Ausgangs-Variable (VAR_IN_OUT)	Datentyp	Beschreibung
Axis	AXIS_REF	Definiert die Achse. Folgende Achsen sind möglich: MLD-S: Axis1 MLD-M: - Axis1 (Master) oder - Axis2...Axis10 (insgesamt 9 Slaves / entfernte Achsen) Die in der Liste "P-0-1801.0.10, CCD: Adressen projektierte Antriebe" vorhandenen entfernten Achsen entsprechen der Reihenfolge nach Axis2 bis Axis10 (Axis2 wirkt auf den Antrieb des Listenelements 0 des P-0-1801.0.10,...)

Tab. 13-84: *Eingangs-/Ausgangs-Variablen des Firmware-Funktionsbausteins "MC_ReadStatus"*

13.3.25 MC_Reset

Der Funktionsbaustein "MC_Reset" dient dazu, Antriebsfehler mit dem Kommando "S-0-0099, C0500 Reset Zustandsklasse 1" zurückzusetzen; die Reglerfreigabe bleibt unberührt.

Damit der Funktionsbaustein das Resetkommando korrekt beenden kann, muss er solange zyklisch aufgerufen werden, bis er "Done"=TRUE oder im Fehlerfall "Error" meldet.



Wenn bereits das Kommando "S-0-0099, C0500 Reset Zustandsklasse 1" aktiv ist, sollte es erst beendet werden, bevor "MC_Reset" aufgerufen wird. Wenn "MC_Reset" aktiviert wurde und noch nicht beendet ist, kann mit steigender Flanke an "Execute" ein erneuter Reset veranlasst werden.



Bei einer entfernten Achse wird der Auftrag an den Servicekanal (Sercos III) abgegeben. Erst bei einem späteren Aufruf kann der Baustein "Done" melden. Damit ergeben sich für die Funktionsbaustein-Aufrufe selbst immer kurze Zeiten. Die Ausgänge "Done" und "Error" müssen hier zyklisch gepollt werden. Die Übertragung der Daten kann abhängig von der Belastung des Servicekanal auch Sekunden in Anspruch nehmen.

Solange der Sercos Ring noch nicht in Phase 2 ist, können keine Daten mit entfernten Achsen ausgetauscht werden. Der Baustein meldet in diesem Fall einen Fehler.

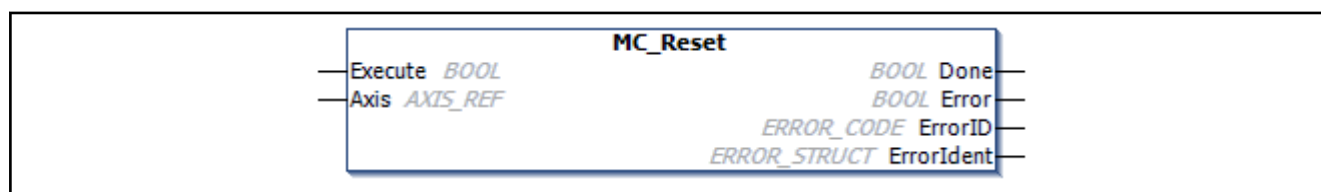


Abb. 13-37: Firmware-Funktionsbaustein "MC_Reset"

Eingangs-Variable	Datentyp	Beschreibung
Execute	BOOL	siehe "Signallaufverhalten"

Tab. 13-85: Eingangs-Variablen des Firmware-Funktionsbausteins "MC_Reset"

Ausgangs-Variable	Datentyp	Beschreibung
Done	BOOL	Mit "Done"=TRUE zeigt der Funktionsbaustein an, dass die Bearbeitung der Arbeitsanweisung erfolgreich abgeschlossen ist
Error	BOOL	Mit "Error"=TRUE meldet der Funktionsbaustein, dass ein Fehler vorliegt
ErrorID *	ERROR_CODE	"ErrorID" gibt eine grobe Fehlerbeschreibung wieder.
ErrorIdent *	ERROR_STRUCT	"ErrorIdent" gibt über die Elemente: "Table", "Additional1" und "Additional2" die genaue Fehlerursache wieder.

* siehe nachfolgender Hinweis

Tab. 13-86: Ausgangs-Variablen des Firmware-Funktionsbausteins "MC_Reset"



HINWEIS: Eine Auflistung aller möglichen Fehler finden Sie hier: "[Fehler-Referenzlisten](#)"

MX_PLCOpen

Eingangs-/Ausgangs-Variable (VAR_IN_OUT)	Datentyp	Beschreibung
Axis	AXIS_REF	Definiert die Achse. Folgende Achsen sind möglich: MLD-S: Axis1 MLD-M: - Axis1 (Master) oder - Axis2...Axis10 (insgesamt 9 Slaves / entfernte Achsen) Die in der Liste "P-0-1801.0.10, CCD: Adressen projektierter Antriebe" vorhandenen entfernten Achsen entsprechen der Reihenfolge nach Axis2 bis Axis10 (Axis2 wirkt auf den Antrieb des Listenelements 0 des P-0-1801.0.10,...)

Tab. 13-87: *Eingangs-/Ausgangs-Variable des Firmware-Funktionsbausteins "MC_Reset"*

13.3.26 MC_Stop

Der Firmware-Funktionsbaustein "MC_Stop" dient dazu, den Antrieb in den Zustand STOP zu bringen.

Mit der steigenden Flanke von "Execute" wird das Anhalten begonnen.

Falls die SPS keine Kontrolle über den Antrieb hat, wird ein Fehler zurückgeliefert.

Wird der Firmware-Funktionsbaustein "MC_Stop" über eine steigende Flanke am "Execute"-Eingang aktiviert, geht der Antrieb in den Zustand "AH" (Antrieb halt). Eine weitere Kommandierung eines Motion-Funktionsbausteins ist erst wieder möglich, wenn sich die Achse im Stillstand befindet und der "Execute"-Eingang den Wert FALSE hat.



Die Funktion "Antrieb Halt" wird in der Funktionsbeschreibung zur Firmware beschrieben.

Der Antrieb wird mit der an "Deceleration" angegebenen Verzögerung zum Stillstand gebracht.

Wenn in einem Projekt mehrere Instanzen von "MC_Stop" aufgerufen werden, wirkt immer die letzte Instanz. Ein aktiver "MC_Stop" kann nicht durch einen "MB_Stop" unterbrochen werden.

Wird eine Instanz mit einer Flanke an "Execute" aufgerufen, so melden andere Instanzen beim Aufruf (ohne Flanke) CommandAborted := TRUE.

"MC_Stop" hat eine höhere Priorität als "MB_Stop"; d. h., dass ein laufender "MC_Stop" nicht durch einen "MB_Stop" abgebrochen werden kann.



Zum Testen des Funktionsbausteins "MC_Stop" kann die Instanzvariable "ActivateCounter" herangezogen werden. Sie zählt die positiven Flanken des "Execute"-Signals und zeigt damit an, ob der Baustein mit Flanke durchlaufen wird.



Über den Firmware-Funktionsbaustein "MC_Stop" kann auch der "P-0-0759, Virtuelle Leitachse, Geschwindigkeitswert" über eine definierte Rampe (über Eingang "Deceleration" definiert) zum Stillstand gebracht werden. Dazu muss der Leitachsgenerator über "P-0-0917, Steuerwort Leitachsgenerator", Bit0=1 aktiviert werden. Um den virtuellen Lageistwert (P-0-0758) in die interne virtuelle Leitachsposition (VmAxis1; "P-0-0761, Leitachsposition") zu wandeln, muss "P-0-0758, Virtueller Lageistwert" in den Parameter "P-0-0916, Leitachsformatwandler Signalauswahl" geschrieben werden. Sie dazu auch Firmware-Funktionsbeschreibung "Leitachsgenerator" und "Leitachsformatwandler"

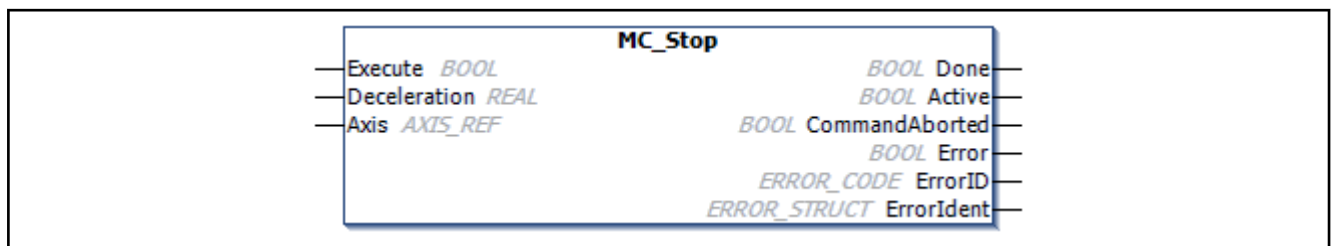


Abb. 13-38: Firmware-Funktionsbaustein "MC_Stop"

Input-Variable	Datentyp	Beschreibung
Execute	BOOL	Folgeachse ("Axis1" bis "Axis8"): Mit steigender Flanke an "Execute" wird der Antrieb in "AH" geschaltet. Leitachsgenerator ("VmAxis1"): Mit steigender Flanke an "Execute" wird der Leitachsgenerator in den Positioniermodus "Halten" gebracht ("P-0-0760, Virtuelle Leitachse, Positioniersteuerwort", Bit1 und Bit2 auf "1"). Siehe auch "Signallaufverhalten"
Deceleration	REAL	Über diesen Eingang wird die Verzögerung eingestellt. Die angegebene Verzögerung beschreibt den Parameter "S-0-0372, Verzögerung Schnell-Halt" der Folgeachse (reale Achse) bzw. den Parameter "P-0-0772, Virtuelle Leitachse, Positionierverzögerung" des virtuellen Leitachsgenerators.

Tab. 13-88: Input-Variablen des Firmware-Funktionsbausteins "MC_Stop"

MX_PLCOpen

Output-Variable	Datentyp	Beschreibung
Done	BOOL	Mit "Done"=TRUE meldet der Firmware-Funktionsbaustein, dass die Achse steht. Folgeachse (reale Achse): Wenn S-0-0331, Bit0=1 ist Virtueller Leitachsgenerator ("VmAxis1"): Wenn der Betrag von P-0-0759 kleiner als P-0-0912 ist
Active	BOOL	Mit "Active"=TRUE meldet der Firmware-Funktionsbaustein, dass er die Achsbewegung bestimmt
CommandAborted	BOOL	Mit "CommandAborted"=TRUE zeigt der Funktionsbaustein an, dass ein anderer "MC_Stop" aktiviert wurde oder dass ein Antriebsfehler vorlag. Ein Antriebsfehler wird über die Error-Ausgänge des "MC_Power" angezeigt
Error	BOOL	Mit "Error"=TRUE meldet der Funktionsbaustein, dass ein Fehler vorliegt
ErrorID *	ERROR_CODE	"ErrorID" gibt eine grobe Fehlerbeschreibung wieder.
ErrorIdent *	ERROR_STRUCT	"ErrorIdent" gibt über die Elemente: "Table", "Additional1" und "Additional2" die genaue Fehlerursache wieder.

* siehe nachfolgender Hinweis

Tab. 13-89: Output-Variablen des Firmware-Funktionsbausteins "MC_Stop"

HINWEIS: Eine Auflistung aller möglichen Fehler finden Sie hier:
["Fehler-Referenzlisten"](#)

Input-/Output-Variable (VAR_IN_OUT)	Datentyp	Beschreibung
Axis	AXIS_REF	Definiert die Achse. Folgende Achsen sind möglich: - Definiert die Achse. Folgende Achsen sind möglich: MLD-S: Axis1 MLD-M: - Axis1 (Master) oder - Axis2...Axis10 (insgesamt 9 Slaves / entfernte Achsen) Die in der Liste "P-0-1801.0.10, CCD: Adressen projektierte Antriebe" vorhandenen entfernten Achsen entsprechen der Reihenfolge nach Axis2 bis Axis10 (Axis2 wirkt auf den Antrieb des Listenelements 0 des P-0-1801.0.10,...) "VmAxis1" entspricht dem virtuellen Leitachsgenerator

Tab. 13-90: Input-/Output-Variable des Firmware-Funktionsbausteins "MC_Stop"

13.3.27 MC_TorqueControl

Der Firmware-Funktionsbaustein "MC_TorqueControl" dient dazu, die Betriebsart "Drehmoment-/Kraftregelung" zu aktivieren; der Antrieb folgt der Sollwertvorgabe.

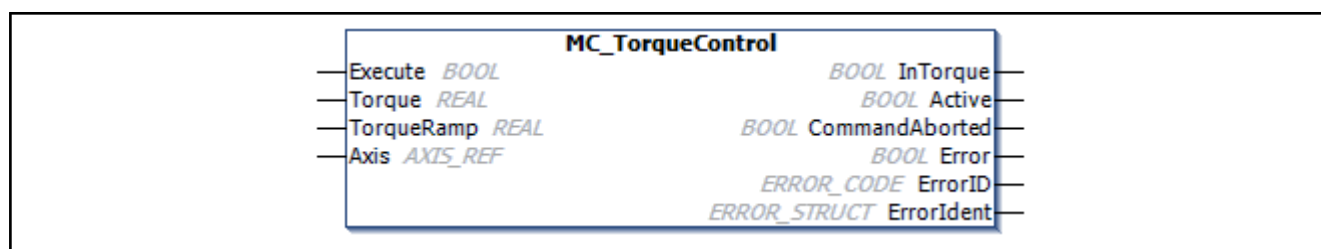


Abb. 13-39: Firmware-Funktionsbaustein "MC_TorqueControl"

Input-Variable	Datentyp	Beschreibung	Verwendete Parameter	
Execute	BOOL	siehe "Signallaufverhalten"	-	
			①	②
Torque	REAL	Drehmoment-/Kraftsollwert ^{*1)}	S-0-0080	P-0-1465
TorqueRamp	REAL	Drehmoment-/Kraftrampe in Bezug auf die Drehmoment- / Kraftrampenzeit ^{*1)} Hinweis: Wird der Eingang "TorqueRamp" auf "0" gesetzt, dann wird der Drehmoment-/Kraftsollwert sprunghaft vorgegeben.	S-0-0822 / S-0-0823	P-0-1466 / P-0-1467

- ① Wenn SPS permanente Kontrolle über die Achse hat oder wenn eine entfernte Achse kommandiert wird
- ② Wenn SPS temporäre Kontrolle über die Achse hat
- ^{*1)} Die Anzahl der möglichen Nachkommastellen richtet sich nach der parametrisierten Wichtung (siehe Antriebsdokumentation). Hat der angelegte Wert mehr Nachkommastellen, so werden diese abgeschnitten.

Tab. 13-91: Input-Variablen des Firmware-Funktionsbausteins "MC_TorqueControl"

Output-Variable	Datentyp	Beschreibung
InTorque	BOOL	Drehmoment-/Kraftsollwert wurde erreicht. Hinweis: Mit dem Parameter "S-0-0826, Drehmoment- / Kraftfenster" wird festgelegt, wann der Funktionsbaustein die Meldung "InTorque" liefert.
Active	BOOL	Funktionsbaustein bzw. Betriebsart ist aktiv
CommandAborted	BOOL	Funktionsbaustein wurde abgebrochen
Error	BOOL	Mit "Error"=TRUE meldet der Funktionsbaustein, dass ein Fehler vorliegt
ErrorID *	ERROR_CODE	"ErrorID" gibt eine grobe Fehlerbeschreibung wieder.
ErrorIdent *	ERROR_STRUCT	"ErrorIdent" gibt über die Elemente: "Table", "Additional1" und "Additional2" die genaue Fehlerursache wieder.

* siehe nachfolgender Hinweis

Tab. 13-92: Output-Variablen des Firmware-Funktionsbausteins "MC_TorqueControl"



HINWEIS: Eine Auflistung aller möglichen Fehler finden Sie hier: "[Fehler-Referenzlisten](#)"

MX_PLCopen

Input-/Output-Variable (VAR_IN_OUT)	Datentyp	Beschreibung
Slave	AXIS_REF	Definiert die Achse. Folgende Achsen sind möglich: MLD-S: Axis1 MLD-M: - Axis1 (Master) oder - Axis2...Axis10 (insgesamt 9 Slaves / entfernte Achsen) Die in der Liste "P-0-1801.0.10, CCD: Adressen projektierte Antriebe" vorhandenen entfernten Achsen entsprechen der Reihenfolge nach Axis2 bis Axis10 (Axis2 wirkt auf den Antrieb des Listenelements 0 des P-0-1801.0.10,...)

Tab. 13-93: Input-/Output-Variablen des Firmware-Funktionsbausteins "MC_TorqueControl"

13.3.28 MX_MoveAbsolute

Der Firmware-Funktionsbaustein "MX_MoveAbsolute" dient dazu, den Antrieb auf eine vorgegebene Absolutposition zu verfahren. Intern wird dazu die SPS-Nebenbetriebsart "Antriebsgeführtes Positionieren" aktiviert (siehe Firmware-Funktionsbeschreibung "Antriebsgeführtes Positionieren"). Einstellungen des Lagereglers (mit/ohne Schleppfehler, Lageistwert 1 / 2) müssen mit dem Baustein "MB_SetPositionControlMode" getroffen werden.

Bei permanenter Kontrolle und bei entfernten Achsen wird die Nebenbetriebsart 3 und deren Parameter verwendet. Bei temporärer Kontrolle wird die SPS-Nebenbetriebsart 11 und deren Parameter verwendet.

Die Zielposition muss an dem Eingang "Position" vorgegeben werden. Über die Eingänge "Velocity", "Acceleration" und "Deceleration" muss bestimmt werden wie die Zielposition angefahren wird (Bewegungsprofil).

Kommandieren einer virtuellen
Leitachse

Soll über den Firmware-Funktionsbaustein "MX_MoveAbsolute" ein virtueller Leitachsgenerator absolut positioniert werden, dann muss der Leitachsgenerator über "P-0-0917, Steuerwort Leitachsgenerator", Bit0="1" aktiviert werden. Um den virtuellen Lageistwert (P-0-0758) in die interne virtuelle Leitachsposition (VmAxis1; "P-0-0761, Leitachsposition") zu wandeln, muss "P-0-0758, Virtueller Lageistwert" in den Parameter "P-0-0916, Leitachsformatwandler Signalauswahl" geschrieben werden (siehe auch Firmware-Funktionsbeschreibung "Leitachsgenerator" und "Leitachsformatwandler").



Die folgenden Parameter dürfen nicht zyklisch beschrieben werden, da sie durch diesen und ggf. weitere Funktionsbausteine beschrieben werden:

- S-0-0346, Steuerwort Positionieren
- P-0-1454, SPS/Einrichtbetrieb Positioniersollwert-Übernahme
- P-0-0760, Virtuelle Leitachse, Positioniersteuerwort

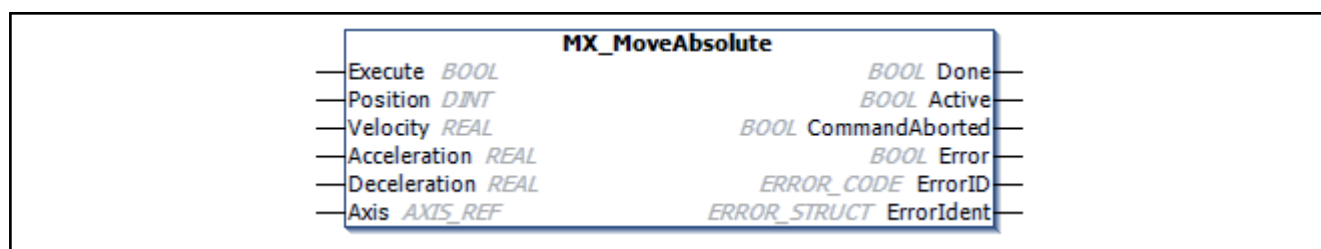


Abb. 13-40: Firmware-Funktionsbaustein "MX_MoveAbsolute"

Input-Variable	Datentyp	Beschreibung	Verwendete Parameter		
Execute	BOOL	siehe "Signallaufverhalten"	-		
			①	②	③
Position	DINT	An "Position" muss die Zielposition im Festkommaformat vorgegeben werden. Um die Position richtig vorgeben zu können, muss die Anzahl der Nachkommastellen des Parameters bekannt sein. Bei Vorzugswichtung (siehe Firmware-Funktionsbeschreibung "Wichtungsarten") hat der Parameter 4 Nachkommastellen. Beispiel: Soll auf eine Zielposition von 100,25° verfahren werden, muss an "Position" der Wert 1002500 geschrieben werden.	S-0-0282	P-0-1450	P-0-0766
Velocity	REAL	Positioniergeschwindigkeit in physikalischer Einheit* ¹⁾	S-0-0259	P-0-1451	P-0-0770
Acceleration	REAL	Positionierbeschleunigung in physikalischer Einheit* ¹⁾	S-0-0260	P-0-1452	P-0-0771
Deceleration	REAL	Positionierverzögerung in physikalischer Einheit* ¹⁾	S-0-0359	P-0-1453	P-0-0772

- ① Wenn SPS permanente Kontrolle über die Achse hat oder wenn eine entfernte Achse kommandiert wird
- ② Wenn SPS temporäre Kontrolle über die Achse hat
- ③ Bei virtueller Leitachse
- *¹⁾ Die Anzahl der möglichen Nachkommastellen richtet sich nach der parametrisierten Wichtung (siehe Antriebsdokumentation). Hat der angelegte Wert mehr Nachkommastellen, so werden diese abgeschnitten.

Tab. 13-94: Input-Variablen des Firmware-Funktionsbausteins "MX_MoveAbsolute"

MX_PLCoPen

Output-Variable	Datentyp	Beschreibung
Done	BOOL	Mit "Done"=TRUE meldet der Firmware-Funktionsbaustein, dass die absolute Zielposition erreicht wurde. Die Abarbeitung ist dann bis zur nächsten 0-1-Flanke am "Execute"-Eingang beendet. Gibt man eine 1-0-Flanke auf den "Execute"-Eingang bevor der Firmware-Funktionsbaustein meldet dass er fertig ist ("Done"=TRUE), wird der Ausgang "Done" für einen Aufruf auf TRUE gesetzt, wenn das Bewegungsziel erreicht wurde und die Abarbeitung des Firmware-Funktionsbausteins nicht durch einen Fehler oder durch einen anderen Firmware-Funktionsbaustein unterbrochen wurde. Reale Achse: Der Ausgang wird auf TRUE gesetzt, wenn das Bit 2 im Parameter "S-0-0437, Positionier-Status" gesetzt wird. Virtueller Leitachsgenerator (VmAxis1): Der Ausgang wird auf TRUE gesetzt, wenn das Bit 2 im Parameter "P-0-0768, Virtuelle Leitachse, Positionierstatus" gesetzt wird.
Active	BOOL	Mit "Active"=TRUE zeigt der Funktionsbaustein an, dass er die Achsbewegung bestimmt. "Active" bleibt TRUE, bis "CommandAborted" oder "Error" gesetzt wird.
CommandAborted	BOOL	Mit "CommandAborted"=TRUE zeigt der Funktionsbaustein an, dass ein anderer Motion-Baustein aktiviert wurde oder dass ein Antriebsfehler vorlag. Ein Antriebsfehler wird über die Error-Ausgänge des "MC_Power" angezeigt
Error	BOOL	Mit "Error"=TRUE meldet der Funktionsbaustein, dass ein Fehler vorliegt
ErrorID *	ERROR_CODE	"ErrorID" gibt eine grobe Fehlerbeschreibung wieder.
ErrorIdent *	ERROR_STRUCT	"ErrorIdent" gibt über die Elemente: "Table", "Additional1" und "Additional2" die genaue Fehlerursache wieder.

*

siehe nachfolgender Hinweis

Tab. 13-95: *Output-Variablen des Firmware-Funktionsbausteins "MX_MoveAbsolute"*



HINWEIS: Eine Auflistung aller möglichen Fehler finden Sie hier: ["Fehler-Referenzlisten"](#)

Eingangs-/Ausgangs-Variable (VAR_IN_OUT)	Datentyp	Beschreibung
Axis	AXIS_REF	Definiert die Achse. Folgende Achsen sind möglich: - Definiert die Achse. Folgende Achsen sind möglich: MLD-S: Axis1 MLD-M: - Axis1 (Master) oder - Axis2...Axis10 (insgesamt 9 Slaves / entfernte Achsen) Die in der Liste "P-0-1801.0.10, CCD: Adressen projektierter Antriebe" vorhandenen entfernten Achsen entsprechen der Reihenfolge nach Axis2 bis Axis10 (Axis2 wirkt auf den Antrieb des Listenelements 0 des P-0-1801.0.10,...) "VmAxis1" entspricht dem virtuellen Leitachsgenerator

Tab. 13-96: *Eingangs-/Ausgangs-Variable des Firmware-Funktionsbausteins "MX_MoveAbsolute"*

13.3.29 MX_MoveAdditive

Der Firmware-Funktionsbaustein "MX_MoveAdditive" dient dazu, den Antrieb relativ um einen Weg zusätzlich zur Zielposition zu verfahren. Die neue Zielposition ergibt sich also aus der alten Zielposition, addiert um eine Distanz. Der Weg (die Distanz) wird durch den Eingang "Distance" bestimmt. Mit den Eingängen "Velocity", "Acceleration" und "Deceleration" wird das Bewegungsprofil bestimmt.

Intern wird dazu die SPS-Nebenbetriebsart "Antriebsgeführtes Positionieren" aktiviert (siehe Firmware-Funktionsbeschreibung "Antriebsgeführtes Positionieren"). Einstellungen des Lagereglers (mit/ohne Schleppfehler, Lageistwert 1 / 2) müssen mit dem Baustein "MB_SetPositionControlMode" getroffen werden.

Bei permanenter Kontrolle und bei entfernten Achsen werden die Nebenbetriebsart 3 und deren Parameter verwendet. Bei temporärer Kontrolle werden die SPS-Nebenbetriebsart 11 und deren Parameter verwendet.

Kommandieren einer virtuellen Leitachse

Soll über den Firmware-Funktionsbaustein "MX_MoveAdditive" ein virtueller Leitachsgenerator relativ um einen Weg **zusätzlich** zur aktuellen Zielposition verfahren werden, dann muss der Leitachsgenerator über "P-0-0917, Steuerwort Leitachsgenerator", Bit0="1" aktiviert werden. Um den virtuellen Lageistwert (P-0-0758) in die interne virtuelle Leitachseposition (VmAxis1; "P-0-0761, Leitachseposition") zu wandeln, muss "P-0-0758, Virtueller Lageistwert" in den Parameter "P-0-0916, Leitachsformatwandler Signalauswahl" geschrieben werden (siehe auch Firmware-Funktionsbeschreibung "Leitachsgenerator" und "Leitachsformatwandler").

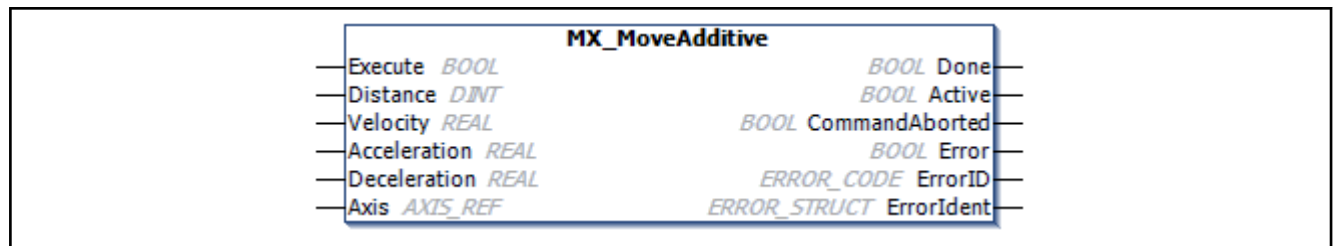


Abb. 13-41: Firmware-Funktionsbaustein "MX_MoveAdditive"



Die folgenden Parameter dürfen nicht zyklisch beschrieben werden, da sie durch diesen und ggf. weitere Funktionsbausteine beschrieben werden:

- S-0-0346, Steuerwort Positionieren
- P-0-1454, SPS/Einrichtbetrieb Positioniersollwert-Übernahme
- P-0-0760, Virtuelle Leitachse, Positioniersteuerwort

MX_PLCOpen

Input-Variable	Datentyp	Beschreibung	Verwendete Parameter		
Execute	BOOL	siehe "Signallaufverhalten"	-		
			①	②	③
Distance	DINT	An "Distance" muss der relative Weg im Festkommaformat vorgegeben werden. Um den Weg richtig vorgeben zu können, muss die Anzahl der Nachkommastellen des Parameters bekannt sein. Bei Vorzugswichtung (siehe Firmware-Funktionsbeschreibung "Wichtungsarten") hat der Parameter 4 Nachkommastellen. Beispiel: Soll ein Weg von 100,25° verfahren werden, muss an "Distance" der Wert 1002500 geschrieben werden.	S-0-0282	P-0-1450	P-0-0766
Velocity	REAL	Positioniergeschwindigkeit in physikalischer Einheit ^{*1)}	S-0-0259	P-0-1451	P-0-0770
Acceleration	REAL	Positionierbeschleunigung in physikalischer Einheit ^{*1)}	S-0-0260	P-0-1452	P-0-0771
Deceleration	REAL	Positionierverzögerung in physikalischer Einheit ^{*1)}	S-0-0359	P-0-1453	P-0-0772

- ① Wenn SPS permanente Kontrolle über die Achse hat oder wenn eine entfernte Achse kommandiert wird
- ② Wenn SPS temporäre Kontrolle über die Achse hat
- ③ Bei virtueller Leitachse
- *1) Die Anzahl der möglichen Nachkommastellen richtet sich nach der parametrisierten Wichtung (siehe Antriebsdokumentation). Hat der angelegte Wert mehr Nachkommastellen, so werden diese abgeschnitten.

Tab. 13-97: *Input-Variablen des Firmware-Funktionsbausteins "MX_MoveAdditive"*

Output-Variable	Datentyp	Beschreibung
Done	BOOL	Mit "Done"=TRUE meldet der Firmware-Funktionsbaustein, dass die Zielposition erreicht wurde. Die Abarbeitung ist dann bis zur nächsten 0-1-Flanke am "Execute"-Eingang beendet. Gibt man eine 1-0-Flanke auf den "Execute"-Eingang bevor der Firmware-Funktionsbaustein meldet dass er fertig ist ("Done"=TRUE), wird der Ausgang "Done" für einen Aufruf auf TRUE gesetzt, wenn das Bewegungsziel erreicht wurde und die Abarbeitung des Firmware-Funktionsbausteins nicht durch einen Fehler oder durch einen anderen Firmware-Funktionsbaustein unterbrochen wurde. • reale Achsen: Der Ausgang wird auf TRUE gesetzt, wenn das Bit 2 im Parameter "S-0-0437, Positionier-Status" gesetzt wird. • virtuelle Leitachse: Der Ausgang wird auf TRUE gesetzt, wenn das Bit 2 im Parameter "P-0-0768, Virtuelle Leitachse, Positionierstatus" gesetzt wird.
Active	BOOL	Mit "Active"=TRUE zeigt der Firmware-Funktionsbaustein an, dass er die Achsbewegung bestimmt.
CommandAborted	BOOL	Mit "CommandAborted"=TRUE zeigt der Firmware-Funktionsbaustein an, dass ein anderer Motion-Funktionsbaustein aktiviert wurde oder dass ein Antriebsfehler vorlag. Ein Antriebsfehler wird über die Error-Ausgänge des "MC_Power" angezeigt.
Error	BOOL	Mit "Error"=TRUE meldet der Funktionsbaustein, dass ein Fehler vorliegt
ErrorID *	ERROR_CODE	"ErrorID" gibt eine grobe Fehlerbeschreibung wieder.
ErrorIdent *	ERROR_STRUCT	"ErrorIdent" gibt über die Elemente: "Table", "Additional1" und "Additional2" die genaue Fehlerursache wieder.

* siehe nachfolgender Hinweis
 Tab. 13-98: *Output-Variablen des Firmware-Funktionsbausteins "MX_MoveAdditive"*



HINWEIS: Eine Auflistung aller möglichen Fehler finden Sie hier: ["Fehler-Referenzlisten"](#)

Eingangs-/Ausgangs-Variable (VAR_IN_OUT)	Datentyp	Beschreibung
Axis	AXIS_REF	Definiert die Achse. Folgende Achsen sind möglich: • Definiert die Achse. Folgende Achsen sind möglich: MLD-S: Axis1 MLD-M: - Axis1 (Master) oder - Axis2...Axis10 (insgesamt 9 Slaves / entfernte Achsen) Die in der Liste "P-0-1801.0.10, CCD: Adressen projektierte Antriebe" vorhandenen entfernten Achsen entsprechen der Reihenfolge nach Axis2 bis Axis10 (Axis2 wirkt auf den Antrieb des Listenelements 0 des P-0-1801.0.10,...) • "VmAxis1" entspricht dem virtuellen Leitachsgenerator

Tab. 13-99: *Eingangs-/Ausgangs-Variable des Firmware-Funktionsbausteins "MX_MoveAdditive"*

MX_PLCoPen

13.3.30 MX_MoveRelative

Der Firmware-Funktionsbaustein "MX_MoveRelative" dient dazu, den Antrieb ab der aktuellen Istposition relativ um einen Weg zu verfahren. Intern wird dazu die SPS-Nebenbetriebsart "Antriebsgeführtes Positionieren" aktiviert (siehe Firmware-Funktionsbeschreibung "Antriebsgeführtes Positionieren").



Intern wird die Betriebsart "Antriebsgeführtes Positionieren" aktiviert. Einstellungen des Lagereglers (mit/ohne Schleppfehler, Lageistwert 1 / 2) müssen mit dem Baustein "MB_SetPositionControlMode" getroffen werden.

Bei permanenter Kontrolle und bei entfernten Achsen werden die Nebenbetriebsart 3 und deren Parameter verwendet. Bei temporärer Kontrolle werden die SPS-Nebenbetriebsart 11 und deren Parameter verwendet.

Der zu verfahrens relative Weg muss an dem Eingang "Distance" vorgegeben werden. Über die Eingänge "Velocity", "Acceleration" und "Deceleration" muss das Bewegungsprofil bestimmt werden (wie der relative Weg verfahren wird).

Kommandieren einer virtuellen Leitachse

Soll über den Firmware-Funktionsbaustein "MX_MoveRelative" ein virtueller Leitachsgenerator ab der aktuellen virtuellen Istposition relativ um einen Weg verfahren werden, dann muss der Leitachsgenerator über "P-0-0917, Steuerwort Leitachsgenerator", Bit0="1" aktiviert werden. Um den virtuellen Lageistwert (P-0-0758) in die interne virtuelle Leitachsisposition (VmAxis1; "P-0-0761, Leitachsisposition") zu wandeln, muss "P-0-0758, Virtueller Lageistwert" in den Parameter "P-0-0916, Leitachsformatwandler Signalauswahl" geschrieben werden (siehe auch Firmware-Funktionsbeschreibung "Leitachsgenerator" und "Leitachsformatwandler").



Die folgenden Parameter dürfen nicht zyklisch beschrieben werden, da sie durch diesen und ggf. weitere Funktionsbausteine beschrieben werden:

- S-0-0346, Steuerwort Positionieren
- P-0-1454, SPS/Einrichtbetrieb Positioniersollwert-Übernahme
- P-0-0760, Virtuelle Leitachse, Positioniersteuerwort

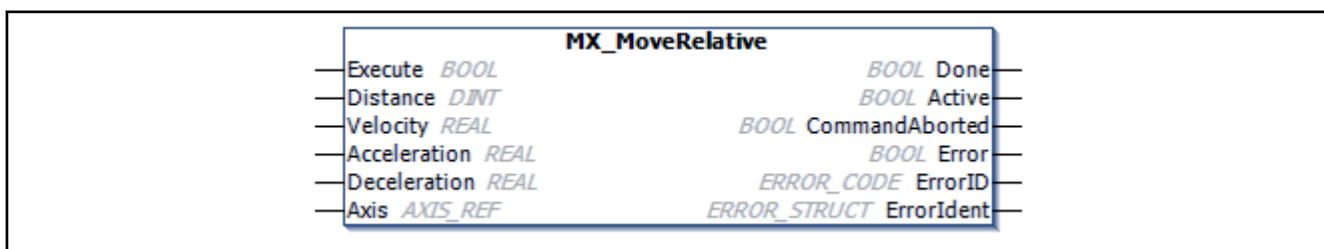


Abb. 13-42: Firmware-Funktionsbaustein "MX_MoveRelative"

Input-Variable	Datentyp	Beschreibung	Verwendete Parameter		
Execute	BOOL	siehe "Signallaufverhalten"	-		
			①	②	③
Distance	DINT	An "Distance" muss der zu verfahrenende relative Weg im Festkommaformat vorgegeben werden. Um den Weg richtig vorgeben zu können, muss die Anzahl der Nachkommastellen des Parameters bekannt sein. Bei Vorzugswichtung (siehe Firmware-Funktionsbeschreibung "Wichtungsarten") hat der Parameter 4 Nachkommastellen. Beispiel: Soll ein Weg von 100,25° verfahren werden, muss an "Distance" der Wert 1002500 geschrieben werden.	S-0-0282	P-0-1450	P-0-0766
Velocity	REAL	Positioniergeschwindigkeit in physikalischer Einheit ^{*1)}	S-0-0259	P-0-1451	P-0-0770
Acceleration	REAL	Positionierbeschleunigung in physikalischer Einheit ^{*1)}	S-0-0260	P-0-1452	P-0-0771
Deceleration	REAL	Positioniervverzögerung in physikalischer Einheit ^{*1)}	S-0-0359	P-0-1453	P-0-0772

- ① Wenn SPS permanente Kontrolle über die Achse hat oder wenn eine entfernte Achse kommandiert wird
- ② Wenn SPS temporäre Kontrolle über die Achse hat
- ③ Bei virtueller Leitachse
- *1) Die Anzahl der möglichen Nachkommastellen richtet sich nach der parametrisierten Wichtung (siehe Antriebsdokumentation). Hat der angelegte Wert mehr Nachkommastellen, so werden diese abgeschnitten.

Tab. 13-100: *Input-Variablen des Firmware-Funktionsbausteins "MX_MoveRelative"*

MX_PLCOpen

Output-Variable	Datentyp	Beschreibung
Done	BOOL	Mit "Done"=TRUE meldet der Firmware-Funktionsbaustein, dass die vorgegebene Distanz verfahren wurde. Die Abarbeitung ist dann bis zur nächsten 0-1-Flanke am "Execute"-Eingang beendet. Gibt man eine 1-0-Flanke auf den "Execute"-Eingang bevor der Firmware-Funktionsbaustein meldet dass er fertig ist ("Done"=TRUE), wird der Ausgang "Done" für einen Aufruf auf TRUE gesetzt, wenn das Bewegungsziel erreicht wurde und die Abarbeitung des Firmware-Funktionsbausteins nicht durch einen Fehler oder durch einen anderen Firmware-Funktionsbaustein unterbrochen wurde. - reale Achsen: Der Ausgang wird auf TRUE gesetzt, wenn das Bit 2 im Parameter "S-0-0437, Positionier-Status" gesetzt wird. - virtuelle Leitachse: Der Ausgang wird auf TRUE gesetzt, wenn das Bit 2 im Parameter "P-0-0768, Virtuelle Leitachse, Positionierstatus" gesetzt wird.
Active	BOOL	Mit TRUE zeigt der Ausgang "Active" an, dass er die Achsbewegung bestimmt
CommandAborted	BOOL	Mit "CommandAborted"=TRUE zeigt der Funktionsbaustein an, dass ein anderer Motion-Baustein aktiviert wurde oder dass ein Antriebsfehler vorlag. Ein Antriebsfehler wird über die Error-Ausgänge des "MC_Power" angezeigt
Error *	BOOL	Mit "Error"=TRUE meldet der Funktionsbaustein, dass ein Fehler vorliegt
ErrorID *	ERROR_CODE	"ErrorID" gibt eine grobe Fehlerbeschreibung wieder.
ErrorIdent *	ERROR_STRUCT	"ErrorIdent" gibt über die Elemente: "Table", "Additional1" und "Additional2" die genaue Fehlerursache wieder.

* siehe nachfolgender Hinweis

Tab. 13-101: *Output-Variablen des Firmware-Funktionsbausteins "MX_MoveRelative"*



HINWEIS: Eine Auflistung aller möglichen Fehler finden Sie hier: ["Fehler-Referenzlisten"](#)

Eingangs-/Ausgangs-Variable (VAR_IN_OUT)	Datentyp	Beschreibung
Axis	AXIS_REF	Definiert die Achse. Folgende Achsen sind möglich: - Definiert die Achse. Folgende Achsen sind möglich: MLD-S: Axis1 MLD-M: - Axis1 (Master) oder - Axis2...Axis10 (insgesamt 9 Slaves / entfernte Achsen) Die in der Liste "P-0-1801.0.10, CCD: Adressen projektierter Antriebe" vorhandenen entfernten Achsen entsprechen der Reihenfolge nach Axis2 bis Axis10 (Axis2 wirkt auf den Antrieb des Listenelements 0 des P-0-1801.0.10,...) "VmAxis1" entspricht dem virtuellen Leitachsgenerator

Tab. 13-102: *Eingangs-/Ausgangs-Variable des Firmware-Funktionsbausteins "MX_MoveRelative"*

13.3.31 MX_SetControl

Der Funktionsbaustein "MX_SetControl" dient dazu, die Kontrolle zwischen Führungskommunikation und SPS umzuschalten; dies ist dann sinnvoll, wenn zunächst die Achse über eine Führungskommunikation gesteuert wird und in einer bestimmten Situation (z. B. Fehler) die interne SPS die Achse weiter steuern soll.



Wenn die Achse generell mit der internen SPS verfahren werden soll, sollte die permanente Kontrolle aktiviert werden.

Bei "PLCCControl"=TRUE bekommt die SPS die Kontrolle über den Antrieb, bei "PLCCControl"=FALSE bekommt die externe Führungskommunikation die Kontrolle über den Antrieb.

Es werden die Signale "Antrieb Halt" (AH), "Antrieb Ein" (EIN) und die Betriebsartenauswahl umgeschaltet. Bei SPS-Kontrolle werden die Signale "AH" und "EIN" nur von intern bedient. Das Signal "Antrieb Halt" wird durch den Funktionsbaustein "MB_Stop" bzw. Anhalten der SPS angesteuert. Das Signal "EIN" kann mit dem Funktionsbaustein "MX_Power" gesteuert werden.

Sind die entsprechenden Funktionsblöcke nicht verwendet, so liegen die internen Signale auf TRUE.

Beim Umschalten auf temporäre Kontrolle wird das interne "EIN"-Signal auf TRUE gesetzt. Daraus ergibt sich folgendes Verhalten, wenn mit "MX_SetControl" die temporäre Kontrolle geholt wurde:

- **Ausschalten:** Mit dem Firmware-Funktionsbaustein "MC_Power".
Damit der Firmware-Funktionsbaustein an seinem Enable-Eingang einen TRUE/FALSE-Übergang erkennen kann, muss er mindestens einmal mit Enable=TRUE aufgerufen werden.
- **Einschalten:** Mit dem Firmware-Funktionsbaustein "MC_Power".
Damit der Firmware-Funktionsbaustein an seinem Enable-Eingang einen FALSE/TRUE Übergang erkennen kann, muss der Firmware-Funktionsbaustein mindestens einmal mit "Enable"=FALSE aufgerufen werden. Diese Maßnahme ist nicht erforderlich, wenn zwischen dem Aufruf von "MX_SetControl" und "MC_Power" eine Umschaltung von Parametriermodus nach Betriebsmodus gemacht wird.



Der Funktionsbaustein "MX_SetControl" liefert Fehler, wenn schon eine permanente Kontrolle aktiv ist (Einstellung über "P-0-1367, SPS Konfiguration").

Während einer laufenden Umschaltung ("Done" und "Error" noch nicht TRUE) kann erneut mit einer "Execute"-Flanke eine Umschaltung gestartet werden.

Bei einem Programmdownload (kein Online-Change) oder bei einem SPS-Reset wird eine temporäre Kontrolle automatisch wieder abgegeben.



Zur Beachtung bei PLCopen-Motion-Funktionsbausteinen ab dem Zielsystem "IndraDrive MP04":

- Bei **temporärer** Kontrolle werden die SPS-Nebenbetriebsarten verwendet.
- Bei **permanenter** Kontrolle werden die Haupt- und die Nebenbetriebsarten 1-6 parametrierung und verwendet.

MX_PLCOpen

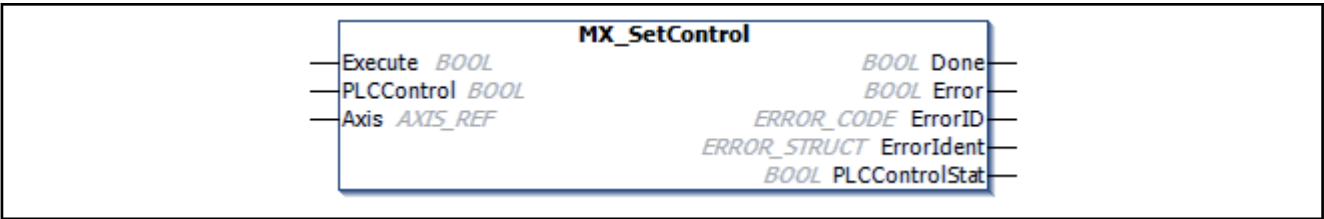


Abb. 13-43: Firmware-Funktionsbaustein "MX_SetControl"

Eingangs-Variable	Datentyp	Beschreibung
Execute	BOOL	Mit steigender Flanke an "Execute" wird der Funktionsbaustein aktiviert. Alle Ausgänge, mit Ausnahme des Ausgangs "PLCControlStat", werden mit fallender Flanke an "Execute" zurückgesetzt. Unabhängig von "Execute" liefert "PLCControlStat" den Istwert.
PLCControl	BOOL	Bei "PLCControl"=TRUE bekommt die interne SPS bei der nächsten 0-1-Flanke an "Execute" temporäre Kontrolle über den Antrieb. Bei "PLCControl"=FALSE wird mit der nächsten 0-1-Flanke an "Execute" die Kontrolle an die Führungskommunikation abgegeben.

Tab. 13-103: Eingangs-Variablen des Firmware-Funktionsbausteins "MX_SetControl"

Ausgangs-Variable	Datentyp	Beschreibung
Done	BOOL	Mit "Done"=TRUE zeigt der Funktionsbaustein an, dass die Bearbeitung der Arbeitsanweisung erfolgreich abgeschlossen ist
Error	BOOL	Mit "Error"=TRUE meldet der Funktionsbaustein, dass ein Fehler vorliegt
ErrorID *	ERROR_CODE	"ErrorID" gibt eine grobe Fehlerbeschreibung wieder.
ErrorIdent *	ERROR_STRUCT	"ErrorIdent" gibt über die Elemente: "Table", "Additional1" und "Additional2" die genaue Fehlerursache wieder.
PLCControlStat	BOOL	Bei "PLCControlStat"=TRUE hat die antriebsinterne SPS Kontrolle über den Antrieb. Bei "PLCControlStat"=FALSE hat die Führungskommunikation Kontrolle über den Antrieb. Unabhängig von "Execute" liefert "PLCControlStat" den Istwert.

* siehe nachfolgender Hinweis
Tab. 13-104: Ausgangs-Variablen des Firmware-Funktionsbausteins "MX_SetControl"

 HINWEIS: Eine Auflistung aller möglichen Fehler finden Sie hier: ["Fehler-Referenzlisten"](#)

Eingangs-/Ausgangs-Variable (VAR_IN_OUT)	Datentyp	Beschreibung
Axis	AXIS_REF	Achsnummer (hier kann nur die globale Variable "Axis1" übergeben werden)

Tab. 13-105: Eingangs-/Ausgangs-Variable des Firmware-Funktionsbausteins "MX_SetControl"

13.3.32 MX_SetDeviceMode

Der Funktionsbaustein "MX_SetDeviceMode" dient dazu, den Antrieb in den Parametrier- bzw. Betriebsmodus zu schalten.

MX_PLCOpen

Wenn keine Umschaltung erlaubt ist ("P-0-4086, Führungskommunikation Status", Bit 0="0"), meldet der Baustein über "Error" einen Fehler zurück.

Ein Modus wird durch eine steigende Flanke von "Execute" eingestellt. Unabhängig von "Execute" wird der aktuelle Zustand im Ausgang "ActOperationMode" angezeigt.



"MX_SetDeviceMode" sollte beim Umschalten solange aufgerufen werden, bis der Ausgang "Done" oder der Ausgang "Error" TRUE wird. Ansonsten kann unter Umständen nicht wieder direkt umgeschaltet werden, da sich "MX_SetDeviceMode" dann ggf. nicht im Initialzustand befindet.

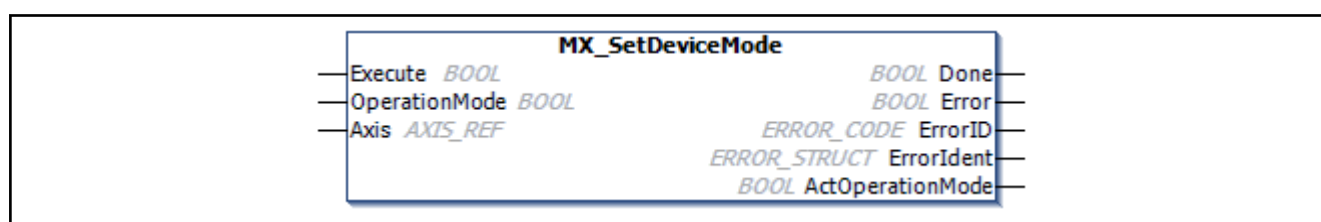


Abb. 13-44: Firmware-Funktionsbaustein "MX_SetDeviceMode"

Eingangs-Variable	Datentyp	Beschreibung
Execute	BOOL	Mit steigender Flanke an "Execute" wird der Funktionsbaustein aktiviert. Alle Ausgänge, mit Ausnahme des Ausgangs "ActOperationMode", werden mit fallender Flanke an "Execute" zurückgesetzt. Unabhängig von "Execute" wird an "ActOperationMode" die aktuelle Kommunikationsphase angezeigt.
OperationMode	BOOL	Bei "OperationMode"=TRUE wird der Antrieb mit der nächsten 0-1-Flanke an "Execute" in den Betriebsmodus (Phase 4) geschaltet; der Baustein startet zu diesem Zweck das Antriebskommando C0200 (S-0-0422). Bei "OperationMode"=FALSE wird der Antrieb mit der nächsten 0-1-Flanke an "Execute" in den Parametriermodus (Phase 2) geschaltet; der Baustein startet zu diesem Zweck das Antriebskommando C0400 (S-0-0420).

Tab. 13-106: Eingangs-Variablen des Firmware-Funktionsbausteins "MX_SetDeviceMode"

MX_PLCOpen

Ausgangs-Variable	Datentyp	Beschreibung
Done	BOOL	Mit "Done"=TRUE zeigt der Funktionsbaustein an, dass die Bearbeitung der Arbeitsanweisung erfolgreich abgeschlossen ist
Error	BOOL	Mit "Error"=TRUE meldet der Funktionsbaustein, dass ein Fehler vorliegt
ErrorID *	ERROR_CODE	"ErrorID" gibt eine grobe Fehlerbeschreibung wieder.
ErrorIdent *	ERROR_STRUCT	"ErrorIdent" gibt über die Elemente: "Table", "Additional1" und "Additional2" die genaue Fehlerursache wieder.
ActOperationMode	BOOL	Bei "ActOperationMode"=TRUE befindet sich der Antrieb im Betriebsmodus (Phase 4). Bei "ActOperationMode"=FALSE befindet sich der Antrieb im Parametriermodus (2<=aktuelle Phase 4). Unabhängig von "Execute" wird an "ActOperationMode" die aktuelle Kommunikationsphase angezeigt.

* siehe nachfolgender Hinweis

Tab. 13-107: *Ausgangs-Variablen des Firmware-Funktionsbausteins "MX_SetDeviceMode"*



HINWEIS: Eine Auflistung aller möglichen Fehler finden Sie hier: ["Fehler-Referenzlisten"](#)

Eingangs-/Ausgangs-Variable (VAR_IN_OUT)	Datentyp	Beschreibung
Axis	AXIS_REF	Achsnummer (hier kann nur die globale Variable "Axis1" übergeben werden)

Tab. 13-108: *Eingangs-/Ausgangs-Variable des Firmware-Funktionsbausteins "MX_SetDeviceMode"*

13.3.33 MX_SetOpMode

Der Funktionsbaustein "MX_SetOpMode" dient dazu, direkt die Betriebsart des Antriebs umzuschalten.

Die aktuelle Betriebsart wird unabhängig von "Execute" geliefert. Die Hauptbetriebsart entspricht dabei dem Wert "0" und die Nebenbetriebsarten entsprechen den Werten "1" bis "3" bei der Firmware MPH02VRS bzw. "1" bis "7" bei MPx03VRS-Firmwaren.



"MX_SetOpMode" ist für die direkte Steuerung des Antriebs über seine Betriebsarten und deren Parameter vorgesehen. Die eingestellte Betriebsart wird im folgenden Lagereglerzyklus aktiviert.

Wird nach dem Umschalten der Antrieb z. B. mit einem "MX_MoveRelative" verfahren, so wird implizit eine Betriebsart >3 bei MPH02VRS und >7 ab MPx03VRS eingestellt.

"MX_SetOpMode" sollte beim Umschalten solange aufgerufen werden, bis der Ausgang "Done" oder der Ausgang "Error" TRUE wird. Ansonsten kann unter Umständen nicht wieder direkt umgeschaltet werden, da sich "MX_SetOpMode" dann ggf. nicht im Initialzustand befindet.

Es dürfen nicht mehrere Instanzen gleichzeitig das Umschalten aktivieren!

Bei der Verwendung des PLCopen-Profiles zur Bewegungssteuerung darf "MX_SetOpMode" nicht verwendet werden. Die SPS-Nebenbetriebsarten sind als "NewOpMode" nicht erlaubt und intern für das PLCopen-Profil der SPS vorgesehen.



Zur Beachtung ab dem Zielsystem "IndraDrive MP04":

- Ab dem Zielsystem sind bei permanenter Kontrolle die ersten 6 Betriebsarten fest parametrierbar und dürfen nicht verändert werden.
- Die 7. Nebenbetriebsart kann frei parametrierbar und mit dem Funktionsbaustein "MX_SetOpMode" im Betrieb angewählt werden.
- Wird der Funktionsbaustein "MB_PreSetMode" verwendet, so schaltet der Antrieb sofort nach dem Einschalten des Antriebs in diese Nebenbetriebsart, ohne vorher kurz in "Antrieb Halt" zu verweilen.
- Aktive Motion-Funktionsbausteine melden "CommandAborted", wenn mit "MX_SetOpMode" eine neue Betriebsart eingestellt wird.
- Die Achse wechselt beim Kommando "MX_SetOpMode" von "AH" nach "AF" bzw. "AU".
- **Achtung:** Bei entfernten Achsen im Mehrachssystem MLD-M sind bereits die Sollwert-Parameter der von den PLCopen-Bausteinen genutzten Betriebsarten im Multiplexkanal (CCD-Prozessdaten) konfiguriert. Diese können somit nicht mehr beschrieben werden!



Das Umschalten der Betriebsart darf nicht durch mehrere Instanzen gleichzeitig erfolgen und wird daher vom System verhindert.

MX_PLCopen

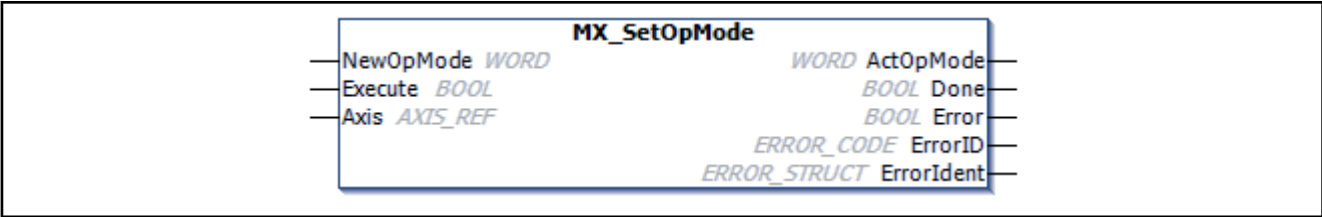


Abb. 13-45: Firmware-Funktionsbaustein "MX_SetOpMode"

Eingangs-Variable	Datentyp	Beschreibung
NewOpMode	WORD	Die an "NewOpMode" eingestellte Betriebsart wird durch eine 0-1-Flanke an "Execute" aktiviert. "NewOpMode" kann zurzeit folgende Werte annehmen: • 0: S-0-0032, Hauptbetriebsart • 1: S-0-0033, Nebenbetriebsart 1 • 2: S-0-0034, Nebenbetriebsart 2 • 3: S-0-0035, Nebenbetriebsart 3 Ab dem Zielsystem "IndraDrive MP03": • 4: S-0-0284, Nebenbetriebsart 4 • 5: S-0-0285, Nebenbetriebsart 5 • 6: S-0-0286, Nebenbetriebsart 6 • 7: S-0-0287, Nebenbetriebsart 7
Execute	BOOL	Mit steigender Flanke an "Execute" wird der Funktionsbaustein aktiviert. Alle Ausgänge, mit Ausnahme des Ausganges "ActOpMode", werden mit fallender Flanke an "Execute" zurückgesetzt. Unabhängig von "Execute" wird an "ActOpMode" die aktuelle Betriebsart angezeigt.

Tab. 13-109: Eingangs-Variablen des Firmware-Funktionsbausteins "MX_SetOp-Mode"

Ausgangs-Variable	Datentyp	Beschreibung
ActOpMode	WORD	"ActOpMode" zeigt an, welche Betriebsart bei Reglerfreigabe im Antrieb aktiv ist. Der Ausgang nimmt den an "NewOpMode" eingestellten Wert an.
Done	BOOL	Mit "Done"=TRUE zeigt der Funktionsbaustein an, dass die Bearbeitung der Arbeitsanweisung erfolgreich abgeschlossen ist
Error	BOOL	Mit "Error"=TRUE meldet der Funktionsbaustein, dass ein Fehler vorliegt
ErrorID *	ERROR_CODE	"ErrorID" gibt eine grobe Fehlerbeschreibung wieder.
ErrorIdent *	ERROR_STRUCT	"ErrorIdent" gibt über die Elemente: "Table", "Additional1" und "Additional2" die genaue Fehlerursache wieder.

* siehe nachfolgender Hinweis
Tab. 13-110: Ausgangs-Variablen des Firmware-Funktionsbausteins "MX_SetOp-Mode"

 HINWEIS: Eine Auflistung aller möglichen Fehler finden Sie hier: ["Fehler-Referenzlisten"](#)

Eingangs-/Ausgangs-Variable (VAR_IN_OUT)	Datentyp	Beschreibung
Axis	AXIS_REF	Definiert die Achse. Folgende Achsen sind möglich: MLD-S: Axis1 MLD-M: - Axis1 (Master) oder - Axis2...Axis10 (insgesamt 9 Slaves / entfernte Achsen) Die in der Liste "P-0-1801.0.10, CCD: Adressen projektierte Antriebe" vorhandenen entfernten Achsen entsprechen der Reihenfolge nach Axis2 bis Axis10 (Axis2 wirkt auf den Antrieb des Listenelements 0 des P-0-1801.0.10,...)

Tab. 13-111: Eingangs-/Ausgangs-Variable des Firmware-Funktionsbausteins "MX_SetOpMode"

13.4 "Parameter" - Funktionsbausteine/Funktionen für Parameter

Laufzeit bei Schreib-/Lesezugriffen

Die Werte einiger Parameter werden bei Schreib-/Lesezugriffen mit Funktionsbausteinen/Funktionen auch zu dem Prozessor für die integrierte Sicherheitstechnik übertragen. Aus diesem Grund dauern Schreib-/Lesezugriffe bei den nachfolgend genannten Parametern vergleichsweise lange (ca. 8 ms). Um Probleme mit der Task-Zykluszeit zu vermeiden, wird empfohlen den Schreib-/Lesezugriff auf die nachfolgend genannten Parameter in einer Task mit mindestens 8 ms Zykluszeit laufen zu lassen (eventuell ohne Watchdog):

- Sicherheitstechnik-Parameter (mit Ausnahme der Parameter deren Parameter-Attribut "zyklisch="AT"" ist): P-0-32xx, P-0-33xx, S-0-1800.x.x bis S-0-1830.x.x
- P-0-0117
- P-0-0119

Schreibschutz durch Passwort

Wenn Parameter durch ein Passwort schreibgeschützt sind, ist das Beschreiben von Parameterwerten über Funktionen oder Funktionsbausteine (z. B. MX_WriteParamDINT) nicht möglich.

13.4.1 MB_Command

Kurzbeschreibung

Der Funktionsbaustein "MB_Command" dient zur Ausführung, zur Überwachung und zum Abbruch von Kommandos.

Schnittstellenbeschreibung

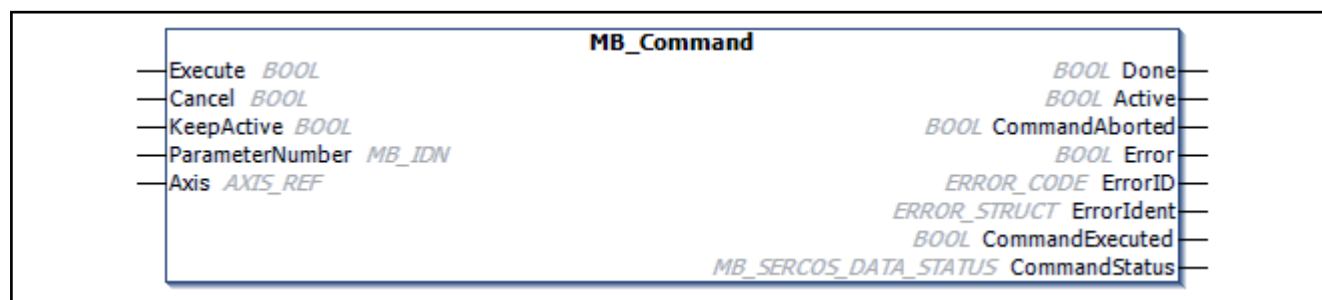


Abb. 13-46: Funktionsbaustein "MB_Command"

Schnittstellenvariablen

MX_PLCOpen

	Name	Typ	Kommentar
VAR_IN_OUT	Axis	AXIS_REF	Referenz zur Achse
VAR_INPUT	Execute	BOOL	Freigabe des Funktionsbausteins (einmalig, flankengesteuert)
	Cancel	BOOL	Abbruch der Kommandoausführung (einmalig, flankengesteuert)
	KeepActive	BOOL	Aktivierung des Kommandos nach Ausführung aktiv halten (permanent, zustandsgesteuert)
	ParameterNumber	MB_IDN	Kommando
VAR_OUTPUT	Done	BOOL	Kommando erfolgreich abgeschlossen und zurückgesetzt
	Active	BOOL	Kommandoausführung aktiv
	CommandAborted	BOOL	Kommando erfolgreich abgebrochen und zurückgesetzt
	Error	BOOL	Mit "Error"=TRUE meldet der Funktionsbaustein, dass ein Fehler vorliegt
	ErrorID	ERROR_CODE	"ErrorID" gibt eine grobe Fehlerbeschreibung wieder.
	ErrorIdent	ERROR_STRUCT	"ErrorIdent" gibt über die Elemente: "Table", "Additional1" und "Additional2" die genaue Fehlerursache wieder.
	CommandExecuted	BOOL	Kommando erfolgreich abgeschlossen
	CommandStatus	MB_SERCOS_DATA_STATUSES	Status der Kommandoausführung

Tab. 13-112: Schnittstellenvariablen Funktionsbaustein MB_Command

Signal-Zeit-Diagramm

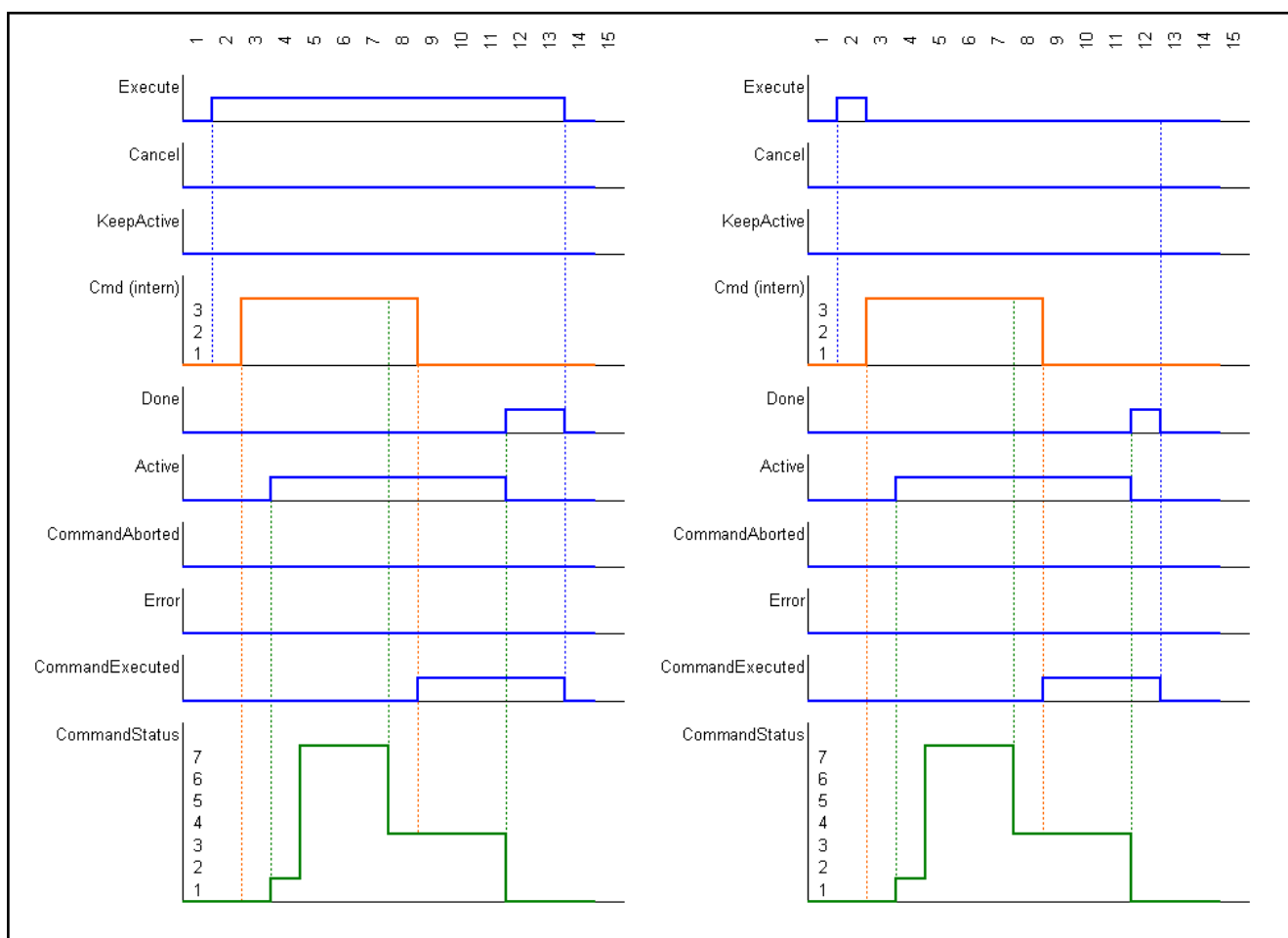


Abb. 13-47: Signal-Zeit-Diagramm für Signalverlauf "Execute-Done"

MX_PLCOpen

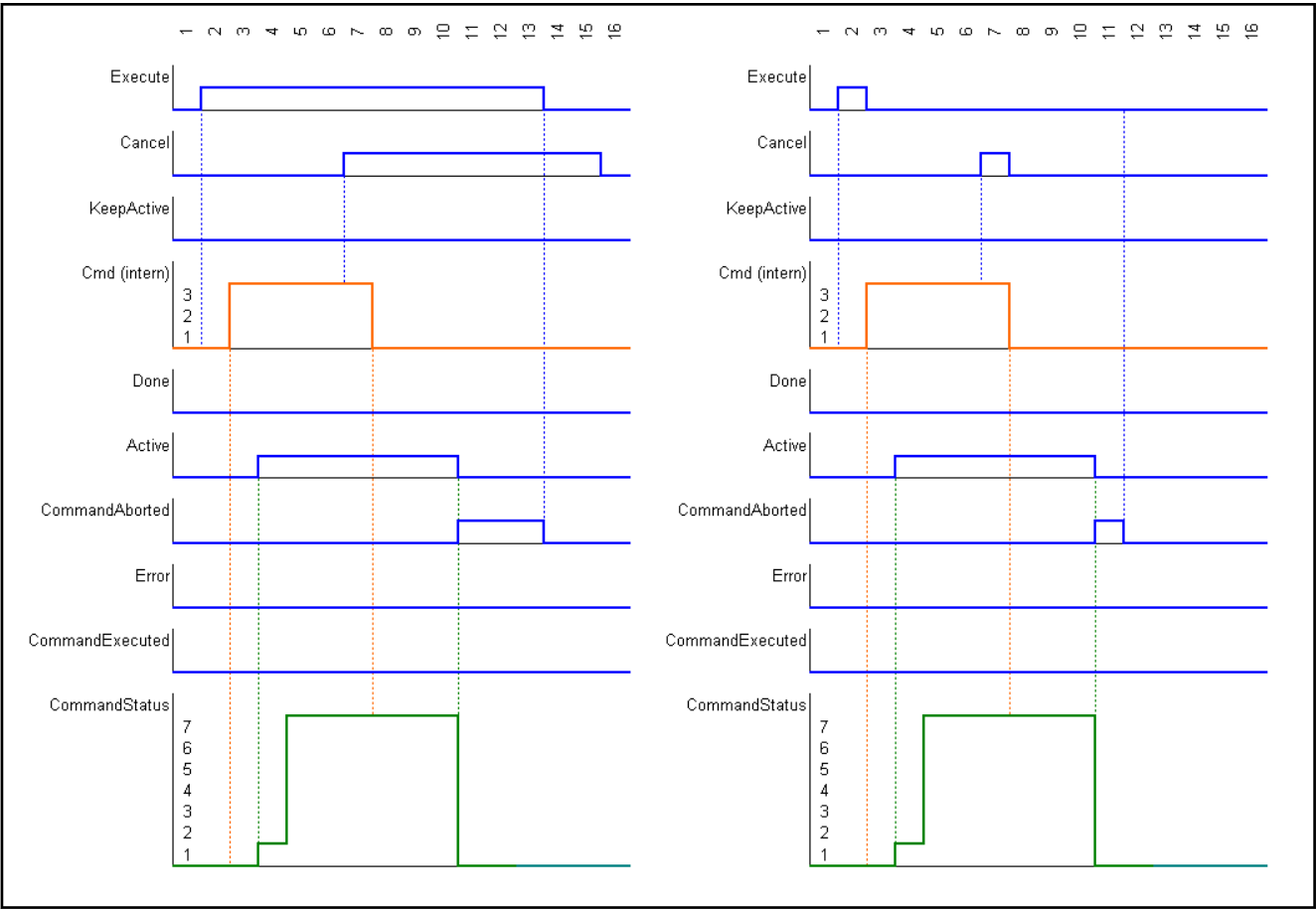


Abb. 13-48: Signal-Zeit-Diagramm für Signalverlauf "Execute-Cancel-Command-Aborted "

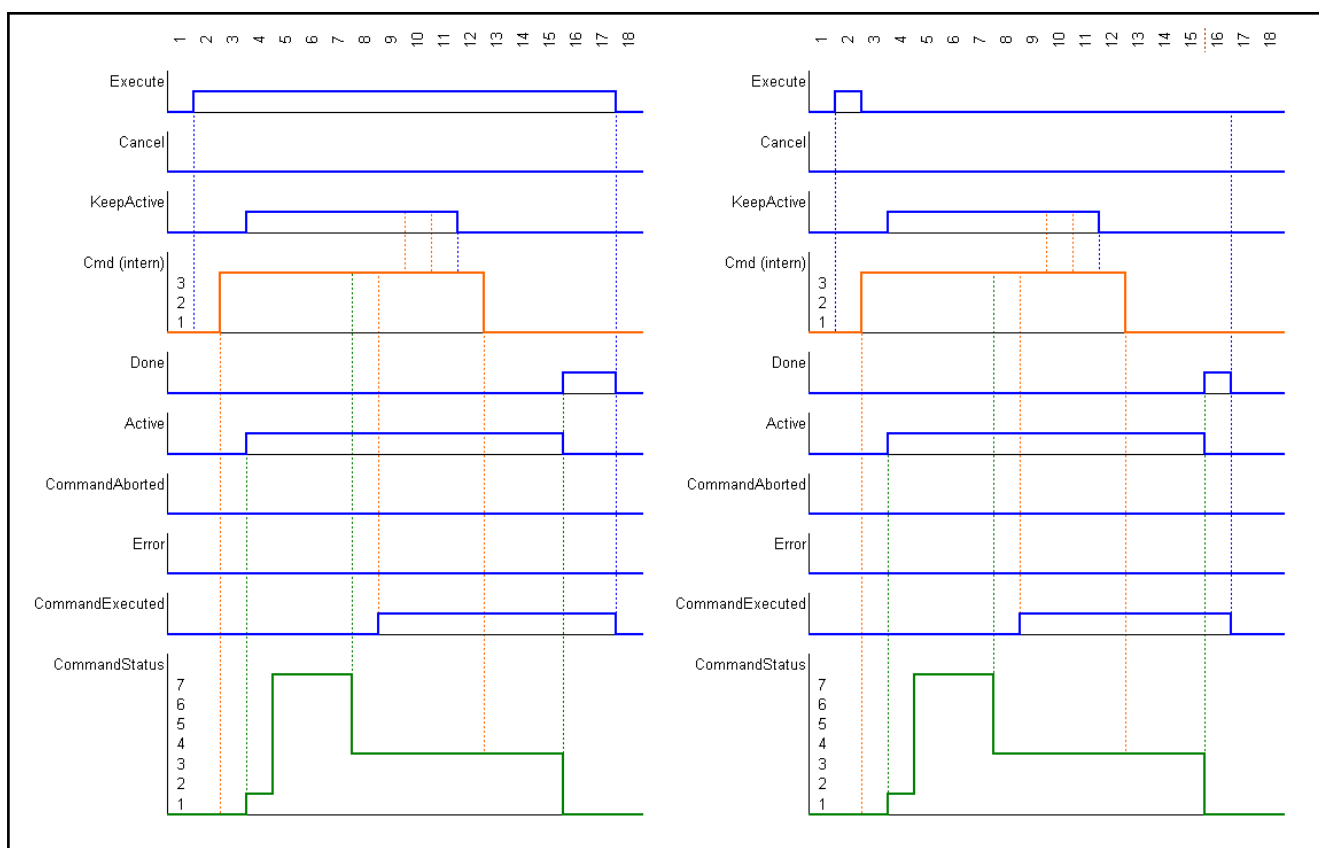


Abb. 13-49: Signal-Zeit-Diagramm für Signalverlauf "Execute-KeepActive-Done "

MX_PLCOpen

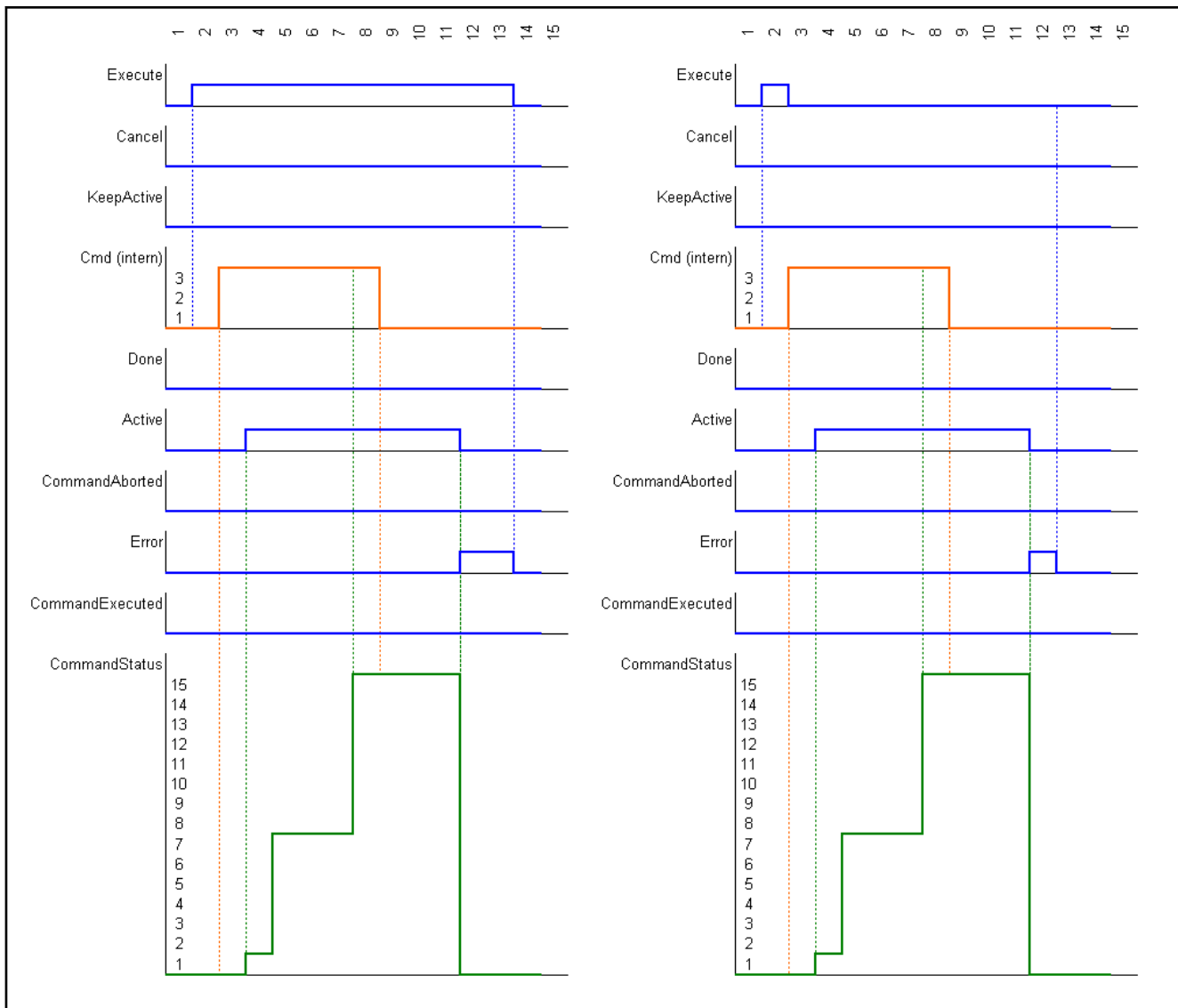


Abb. 13-50: Signal-Zeit-Diagramm für Signalverlauf "Execute-Error"

Funktionsbeschreibung

Der Funktionsbaustein "MB_Command" startet nach der Bearbeitungsfreigabe mittels "Execute" die Kommandoausführung des über den Eingang "ParameterNumber" festgelegten Kommandos. Das Kommando wird im allgemeinen nach Abschluss der Bearbeitung automatisch zurückgesetzt. Das Kommando muss im Parameter-Identnummerformat nach Sercos Norm angegeben werden. Neben dieser direkten Angabe ist es weiterhin möglich, eine entsprechend vordefinierte Konstante aus dem Bereich der globalen Variablen der Basisbibliothek (z. B. "ML_Base") bzw. eine Umwandlungsfunktion aus der Bibliothek "RIL_Uilities" zu verwenden. Die Kommandoausführung kann, abhängig vom Kommando selbst, jederzeit über den Eingang "Cancel" abgebrochen werden. Der Eingang "KeepActive" ermöglicht es, das Kommando nach Ausführung weiter aktiv zu halten. Wird im Folgenden der Eingang "KeepActive" deaktiviert, wird das Kommando ebenfalls automatisch zurückgesetzt. Der erfolgreiche Abschluss bzw. erfolgreiche Abbruch eines Kommandos wird über die Ausgänge "Done" bzw. "CommandAborted" für die Zeitdauer von mindestens einem Aufruf signalisiert. Die aktive Bearbeitung des Kommandos wird über den Ausgang "Active" sowie mittels des entsprechenden Datums am Ausgang "CommandStatus" signalisiert.

"CommandStatus" dargestellt. Der aktuelle Status der Kommandoausführung wird permanent am Ausgang "CommandStatus" signalisiert, der Abschluss der Kommandoausführung wird durch den Ausgang "CommandExecuted" quittiert. Tritt während der Bearbeitung des Kommandos ein Fehler auf, wird dies über den Ausgang "Error" sowie über die Elemente der Fehlerklassifizierung (ErrorID, ErrorIdent) für die Zeitdauer von mindestens einem Aufruf signalisiert.

Fehlerbehandlung Der Funktionsbaustein "MB_Command" nutzt die Fehlertabellen **F_RELATED_TABLE**, 16#0170, sowie **INDRV_TABLE**, 16#0070. Er kann in Additional1 und Additional2 die, in der nachfolgenden Tabelle aufgeführten Fehler erzeugen. Zusätzlich zu diesem Fehler können die eingebundene Funktionsbausteine "MB_WriteParameter", "MB_ReadSercosDataStatus", "MB_ReadParameter" sowie "MB_ReadSercosAttribute" weitere Fehler melden.

ErrorID	Additional1	Additional2	Beschreibung
INPUT_RANGE_ERROR, 16#0006	16#0002	16#0001	ungültiger Eingang - Die Retriggerung der Kommandoausführung wird durch den Baustein "MB_Command" nicht unterstützt.
INPUT_RANGE_ERROR, 16#0006	16#0002	16#0002	ungültiger Eingang - Die Eingänge "Cancel" bzw. "KeepActive" werden durch Steuerungskommandos nicht unterstützt.
INPUT_RANGE_ERROR, 16#0006	16#0002	16#0003	ungültiger Eingang - Der Eingang "ParameterNumber" enthält kein gültiges Kommando.
DEVICE_ERROR, 16#0008	16#1701	16#0000	Fehler bei der Kommandoausführung (Steuerung)
STATE_MACHINE_ERROR, 16#0005	16#0006	16#????	ungültiger Zustand der Zustandsmaschine 16#???? gibt den ungültigen Zustand an

Tab. 13-113: Fehlernummern der Fehlertabelle "F_RELATED_TABLE", verursacht durch "MB_Command"

ErrorID	Additional1	Additional2	Beschreibung
DEVICE_ERROR, 16#0008	16#????	16#0000	Fehler bei der Kommandoausführung (Drive) - 16#???? enthält den aktuellen Wert aus S-0-0390.

Tab. 13-114: Fehlernummern der Fehlertabelle "INDRV_TABLE", verursacht durch "MB_Command"

13.4.2 MB_ReadListParameter

Mit dem Funktionsbaustein "MB_ReadListParameter" kann der aktuelle Wert eines Listen-Parameters gelesen werden.

Die Identnummer des zu lesenden Listenparameters muss am Eingang "ParameterNumber" angelegt werden. An dem Eingang "NoOfBytes" ist die Anzahl der zu lesenden Bytes anzugeben. An dem Eingang "ValueAdr" ist die Anfangsadresse des Feldes anzugeben, in das die gelesenen Bytes geschrieben werden sollen. Die Bearbeitung des Funktionsbausteins wird mit einer steigenden Flanke am Eingang "Execute" ("Execute"=TRUE) gestartet.

 Zum Testen des Funktionsbausteins "MB_ReadListParameter" kann die Instanzvariable "ActivateCounter" herangezogen werden. Sie zählt die positiven Flanken des Execute-Signals und zeigt damit an, ob der Baustein mit Flanke durchlaufen wird.

Lokale Achse Wird der Funktionsbaustein bei einer lokalen Achse verwendet, so werden bei seinem Aufruf sofort alle Listenelemente des ausgewählten Parameters gelesen. Nach dem Auslesen der Listenelemente kehrt die Kontrolle an das Programm zurück und am Ausgang "Done" liegt TRUE an.

 Es sollte beachtet werden, dass der Lesevorgang in Abhängigkeit der Menge der zu lesenden Daten einige Zeit in Anspruch nehmen kann; falls notwendig sollte eine eigene Task benutzt werden.

Beispiel: Um alle Listenelemente des Parameters P-0-0072 zu lesen (1024 Langworte), werden mehrere 100 µs benötigt.

Entfernte Achse  Der Zugriff auf Parameter von entfernten Achsen ist im MLD-M-Systemmodus, im CCD-Basismodus und im CCD-Systemmodus möglich.

Wird der Funktionsbaustein bei entfernter Achse verwendet, wird der Leseauftrag an den Servicekanal (Sercos III) abgegeben. Erst bei einem späteren Aufruf kann der Funktionsbaustein "Done" melden.

Die Ausgänge "Done", "Active" und "Error" müssen hier zyklisch gepollt werden.

Die Übertragung der Daten kann abhängig von der Menge mehrere Sekunden in Anspruch nehmen.

 Solange der Sercos Ring noch nicht in Phase2 ist, können keine Daten mit entfernten Achsen ausgetauscht werden. Der Baustein meldet in diesem Fall einen Fehler.

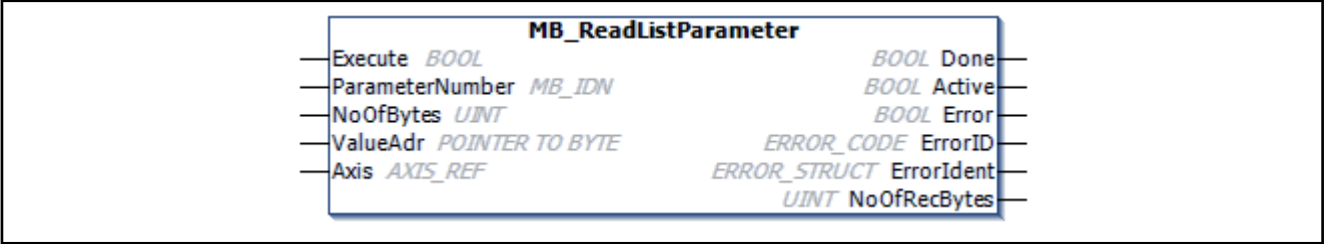


Abb. 13-51: Firmware-Funktionsbaustein "MB_ReadListParameter"

Eingangs-Variable	Datentyp	Beschreibung
Execute	BOOL	siehe "Signallaufverhalten"
ParameterNumber	MB_IDN	Identnummer des auszulesenden Parameters

Eingangs-Variable	Datentyp	Beschreibung
NoOfBytes	UINT	An dem Eingang "NoOfBytes" ist die Anzahl der zu lesenden Bytes anzugeben. Beispiel: NoOfBytes:=SIZEOF(arFeld1)
ValueAdr	POINTER TO BYTE	An dem Eingang "ValueAdr" ist die Anfangsadresse des Feldes anzugeben, in das die gelesenen Bytes geschrieben werden sollen. Beispiel: ValueAdr := ADR(arFeld1) Hinweis: Die Adresse muss zur Speicherausrichtung (Alignment) der Daten passen, auf die zugegriffen wird: • 1-Byte-Listen: jede Adresse • 2-Byte-Listen: jede gerade Adresse • 4-Byte-Listen: jede durch 4 teilbare Adresse

Tab. 13-115: *Eingangs-Variablen des Firmware-Funktionsbausteins "MB_ReadListParameter"*

Ausgangs-Variable	Datentyp	Beschreibung
Done	BOOL	Mit "Done"=TRUE zeigt der Baustein an, dass das Lesen des Datumwertes des Listenparameters erfolgreich war
Active	BOOL	Bei entfernten Achsen wird mit "Active"=TRUE angezeigt, dass der Funktionsbaustein die Bearbeitungsanweisung angenommen hat, bei Einzelachsen wird dieser Ausgang nicht gesetzt!
Error	BOOL	Mit "Error"=TRUE meldet der Funktionsbaustein, dass ein Fehler vorliegt
ErrorID *	ERROR_CODE	"ErrorID" gibt eine grobe Fehlerbeschreibung wieder.
ErrorIdent *	ERROR_STRUCT	"ErrorIdent" gibt über die Elemente: "Table", "Additional1" und "Additional2" die genaue Fehlerursache wieder.
NoOfRecBytes	UINT	Der Ausgang "NoOfRecBytes" zeigt an, wie viele Bytes tatsächlich gelesen wurden

* siehe nachfolgender Hinweis

Tab. 13-116: *Ausgangs-Variablen des Firmware-Funktionsbausteins "MB_ReadListParameter"*



HINWEIS: Eine Auflistung aller möglichen Fehler finden Sie hier: ["Fehler-Referenzlisten"](#)

MX_PLCOpen

Eingangs-/Ausgangs-Variable (VAR_IN_OUT)	Datentyp	Beschreibung
Axis	AXIS_REF	Definiert die Achse. Folgende Achsen sind möglich: MPB (IndraDrive) und HDC (HydraulicDrive): Axis1 MPC (IndraDrive): - Axis1 (Master) oder - Axis2...Axis10 (insgesamt 9 Slaves / entfernte Achsen) Die in der Liste "P-0-1801.0.10, CCD: Adressen projektierte Antriebe" vorhandenen entfernten Achsen entsprechen der Reihenfolge nach Axis2 bis Axis10 (Axis2 wirkt auf den Antrieb des Listenelements 0 des P-0-1801.0.10,...)

Tab. 13-117: Eingangs-/Ausgangs-Variable des Firmware-Funktionsbausteins "MB_ReadListParameter"

13.4.3 MB_ReadMaxRealValue

Mit dem Funktionsbaustein "MB_ReadMaxRealValue" kann der maximal zulässige Wert eines Parameters ausgelesen werden. Der Wert wird im REAL-Format zurückgegeben.

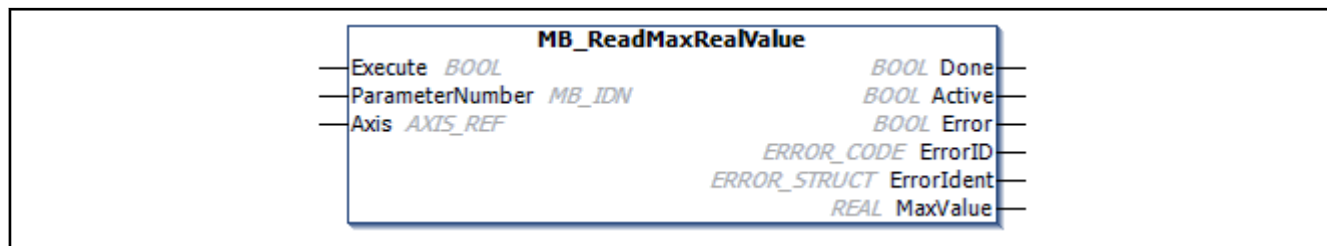


Abb. 13-52: Firmware-Funktionsbaustein "MB_ReadMaxRealValue"

Eingangs-Variable	Datentyp	Beschreibung
Execute	BOOL	siehe "Signallaufverhalten"
ParameterNumber	MB_IDN	An dem Eingang "ParameterNumber" ist die Identnummer des Parameters anzulegen, dessen Maximalwert gelesen werden soll

Tab. 13-118: Eingangs-Variablen des Firmware-Funktionsbausteins "MB_ReadMaxRealValue"

Ausgangs-Variable	Datentyp	Beschreibung
Done	BOOL	Mit "Done"=TRUE zeigt der Funktionsbaustein an, dass die Bearbeitung der Arbeitsanweisung erfolgreich abgeschlossen ist
Active	BOOL	Bei entfernten Achsen wird mit "Active"=TRUE angezeigt, dass der Funktionsbaustein die Bearbeitungsanweisung angenommen hat, bei Einzelachsen wird dieser Ausgang nicht gesetzt!
Error	BOOL	Mit "Error"=TRUE meldet der Funktionsbaustein, dass ein Fehler vorliegt
ErrorID *	ERROR_CODE	"ErrorID" gibt eine grobe Fehlerbeschreibung wieder.

Ausgangs-Variable	Datentyp	Beschreibung
ErrorIdent *	ERROR_STRUCT	"ErrorIdent" gibt über die Elemente: "Table", "Additional1" und "Additional2" die genaue Fehlerursache wieder.
MaxValue	REAL	Ist "Done"=TRUE, zeigt "MaxValue" den maximalen Eingabewert des ausgewählten Parameters an. Hinweis: Je nach Wichtung besitzt der Eingabewert des Parameters Nachkommastellen die hier beachtet und verrechnet werden müssen

* siehe nachfolgender Hinweis

Tab. 13-119: *Ausgangs-Variablen des Firmware-Funktionsbausteins "MB_ReadMaxRealValue"*



HINWEIS: Eine Auflistung aller möglichen Fehler finden Sie hier: "[Fehler-Referenzlisten](#)"

Eingangs-/Ausgangs-Variable (VAR_IN_OUT)	Datentyp	Beschreibung
Axis	AXIS_REF	Definiert die Achse. Folgende Achsen sind möglich: MPB (IndraDrive) und HDC (HydraulicDrive): Axis1 MPC (IndraDrive): - Axis1 (Master) oder - Axis2...Axis10 (insgesamt 9 Slaves / entfernte Achsen) Die in der Liste "P-0-1801.0.10, CCD: Adressen projektierter Antriebe" vorhandenen entfernten Achsen entsprechen der Reihenfolge nach Axis2 bis Axis10 (Axis2 wirkt auf den Antrieb des Listenelements 0 des P-0-1801.0.10,...) Bezüglich Zugriff auf Parameter von entfernten Achsen bitte nachfolgende Hinweise beachten!

Tab. 13-120: *Eingangs-/Ausgangs-Variable des Firmware-Funktionsbausteins "MB_ReadMaxRealValue"*



Bei einer entfernten Achse wird der Auftrag an den Servicekanal (Sercos III) abgegeben. Erst bei einem späteren Aufruf kann der Baustein "Done"="TRUE" melden. Damit ergeben sich für die Funktionsbaustein-Aufrufe selbst immer kurze Zeiten. Die Ausgänge "Done", "Active" und "Error" müssen hier zyklisch gepollt werden. Die Übertragung der Daten kann abhängig von der Belastung des Servicekanal auch Sekunden in Anspruch nehmen.

Solange der Sercos Ring noch nicht in Phase2 ist, können keine Daten mit entfernten Achsen ausgetauscht werden. Der Baustein meldet in diesem Fall einen Fehler.



Der Zugriff auf Parameter von entfernten Achsen ist im MLD-M-Systemmodus, im CCD-Basismodus und im CCD-Systemmodus möglich.

MX_PLCoOpen

13.4.4 MB_ReadMaxValue

Mit dem Funktionsbaustein "MB_ReadMaxValue" kann der maximal zulässige Wert eines Parameters ausgelesen werden. Der Wert wird im DINT-Format zurückgegeben.

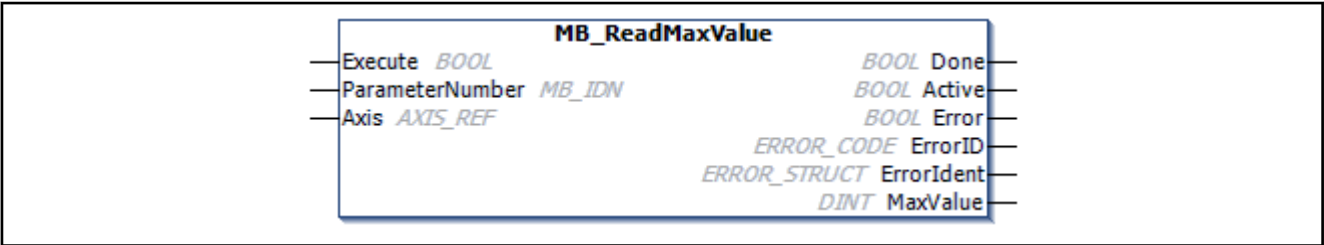


Abb. 13-53: Firmware-Funktionsbaustein "MB_ReadMaxValue"

Eingangs-Variable	Datentyp	Beschreibung
Execute	BOOL	siehe "Signallaufverhalten"
ParameterNumber	MB_IDN	An dem Eingang "ParameterNumber" ist die Identnummer des Parameters anzulegen, dessen Maximalwert gelesen werden soll

Tab. 13-121: Eingangs-Variablen des Firmware-Funktionsbausteins "MB_ReadMaxValue"

Ausgangs-Variable	Datentyp	Beschreibung
Done	BOOL	Mit "Done"=TRUE zeigt der Funktionsbaustein an, dass die Bearbeitung der Arbeitsanweisung erfolgreich abgeschlossen ist
Active	BOOL	Bei entfernten Achsen wird mit "Active"=TRUE angezeigt, dass der Funktionsbaustein die Bearbeitungsanweisung angenommen hat, bei Einzelachsen wird dieser Ausgang nicht gesetzt!
Error	BOOL	Mit "Error"=TRUE meldet der Funktionsbaustein, dass ein Fehler vorliegt
ErrorID *	ERROR_CODE	"ErrorID" gibt eine grobe Fehlerbeschreibung wieder.
ErrorIdent *	ERROR_STRUCT	"ErrorIdent" gibt über die Elemente: "Table", "Additional1" und "Additional2" die genaue Fehlerursache wieder.
MaxValue	DINT	Ist "Done"=TRUE, zeigt "MaxValue" den maximalen Eingabewert des ausgewählten Parameters an. Hinweis: Je nach Wichtung besitzt der Eingabewert des Parameters Nachkommastellen die hier beachtet und verrechnet werden müssen

* siehe nachfolgender Hinweis

Tab. 13-122: Ausgangs-Variablen des Firmware-Funktionsbausteins "MB_ReadMaxValue"

HINWEIS: Eine Auflistung aller möglichen Fehler finden Sie hier:
["Fehler-Referenzlisten"](#)

Eingangs-/Ausgangs-Variable (VAR_IN_OUT)	Datentyp	Beschreibung
Axis	AXIS_REF	Definiert die Achse. Folgende Achsen sind möglich: MPB (IndraDrive) und HDC (HydraulicDrive): Axis1 MPC (IndraDrive): - Axis1 (Master) oder - Axis2...Axis10 (insgesamt 9 Slaves / entfernte Achsen) Die in der Liste "P-0-1801.0.10, CCD: Adressen projektierter Antriebe" vorhandenen entfernten Achsen entsprechen der Reihenfolge nach Axis2 bis Axis10 (Axis2 wirkt auf den Antrieb des Listenelements 0 des P-0-1801.0.10,...) Bezüglich Zugriff auf Parameter von entfernten Achsen bitte nachfolgende Hinweise beachten!

Tab. 13-123: Eingangs-/Ausgangs-Variable des Firmware-Funktionsbausteins "MB_ReadMaxValue"



Bei einer entfernten Achse wird der Auftrag an den Servicekanal (Sercos III) abgegeben. Erst bei einem späteren Aufruf kann der Baustein "Done"="TRUE" melden. Damit ergeben sich für die Funktionsbaustein-Aufrufe selbst immer kurze Zeiten. Die Ausgänge "Done", "Active" und "Error" müssen hier zyklisch gepollt werden. Die Übertragung der Daten kann abhängig von der Belastung des Servicekanal auch Sekunden in Anspruch nehmen.

Solange der Sercos Ring noch nicht in Phase2 ist, können keine Daten mit entfernten Achsen ausgetauscht werden. Der Baustein meldet in diesem Fall einen Fehler.



Der Zugriff auf Parameter von entfernten Achsen ist im MLD-M-Systemmodus, im CCD-Basismodus und im CCD-Systemmodus möglich.

13.4.5 MB_ReadMinRealValue

Mit dem Funktionsbaustein "MB_ReadMinRealValue" kann der minimal zulässige Wert eines Parameters ausgelesen werden. Der Wert wird im REAL-Format zurückgegeben.



Abb. 13-54: Firmware-Funktionsbaustein "MB_ReadMinRealValue"

MX_PLCoPen

Eingangs-Variable	Datentyp	Beschreibung
Execute	BOOL	siehe "Signallaufverhalten"
ParameterNumber	MB_IDN	An dem Eingang "ParameterNumber" ist die Identnummer des Parameters anzulegen, dessen Minimalwert gelesen werden soll

Tab. 13-124: Eingangs-Variablen des Firmware-Funktionsbausteins
"MB_ReadMinRealValue"

Ausgangs-Variable	Datentyp	Beschreibung
Done	BOOL	Mit "Done"=TRUE zeigt der Funktionsbaustein an, dass die Bearbeitung der Arbeitsanweisung erfolgreich abgeschlossen ist
Active	BOOL	Bei entfernten Achsen wird mit "Active"=TRUE angezeigt, dass der Funktionsbaustein die Bearbeitungsanweisung angenommen hat, bei Einzelachsen wird dieser Ausgang nicht gesetzt!
Error	BOOL	Mit "Error"=TRUE meldet der Funktionsbaustein, dass ein Fehler vorliegt
ErrorID *	ERROR_CODE	"ErrorID" gibt eine grobe Fehlerbeschreibung wieder.
ErrorIdent *	ERROR_STRUCT	"ErrorIdent" gibt über die Elemente: "Table", "Additional1" und "Additional2" die genaue Fehlerursache wieder.
MinValue	REAL	Ist "Done"=TRUE, zeigt "MinValue" den minimalen Eingabewert des ausgewählten Parameters an. Hinweis: Je nach Wichtung besitzt der Eingabewert des Parameters Nachkommastellen die hier beachtet und verrechnet werden müssen

* siehe nachfolgender Hinweis

Tab. 13-125: Ausgangs-Variablen des Firmware-Funktionsbausteins
"MB_ReadMinRealValue"



HINWEIS: Eine Auflistung aller möglichen Fehler finden Sie hier:
"Fehler-Referenzlisten"

Eingangs-/Ausgangs-Variable (VAR_IN_OUT)	Datentyp	Beschreibung
Axis	AXIS_REF	Definiert die Achse. Folgende Achsen sind möglich: MPB (IndraDrive) und HDC (HydraulicDrive): Axis1 MPC (IndraDrive): - Axis1 (Master) oder - Axis2...Axis10 (insgesamt 9 Slaves / entfernte Achsen) Die in der Liste "P-0-1801.0.10, CCD: Adressen projektierte Antriebe" vorhandenen entfernten Achsen entsprechen der Reihenfolge nach Axis2 bis Axis10 (Axis2 wirkt auf den Antrieb des Listenelements 0 des P-0-1801.0.10,...) Bezüglich Zugriff auf Parameter von entfernten Achsen bitte nachfolgende Hinweise beachten!

Tab. 13-126: Eingangs-/Ausgangs-Variable des Firmware-Funktionsbausteins
"MB_ReadMinRealValue"



Bei einer entfernten Achse wird der Auftrag an den Servicekanal (Sercos III) abgegeben. Erst bei einem späteren Aufruf kann der Baustein "Done"="TRUE" melden. Damit ergeben sich für die Funktionsbaustein-Aufrufe selbst immer kurze Zeiten. Die Ausgänge "Done", "Active" und "Error" müssen hier zyklisch gepollt werden. Die Übertragung der Daten kann abhängig von der Belastung des Servicekanal auch Sekunden in Anspruch nehmen.

Solange der Sercos Ring noch nicht in Phase2 ist, können keine Daten mit entfernten Achsen ausgetauscht werden. Der Baustein meldet in diesem Fall einen Fehler.



Der Zugriff auf Parameter von entfernten Achsen ist im MLD-M-Systemmodus, im CCD-Basismodus und im CCD-Systemmodus möglich.

13.4.6 MB_ReadMinValue

Mit dem Funktionsbaustein "MB_ReadMinValue" kann der minimal zulässige Wert eines Parameters ausgelesen werden. Der Wert wird im DINT-Format zurückgegeben.

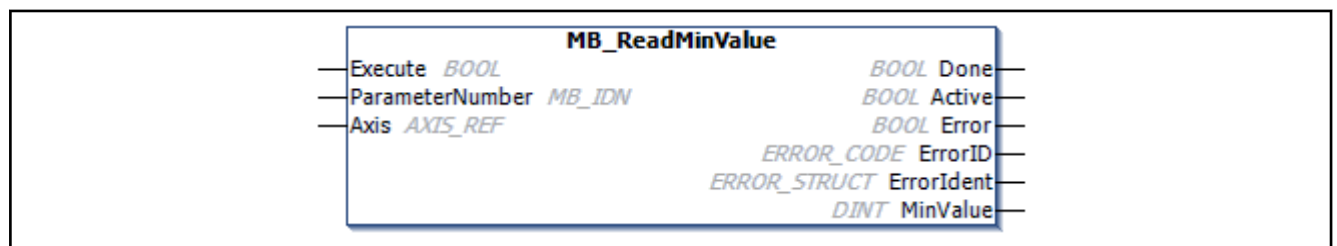


Abb. 13-55: Firmware-Funktionsbaustein "MB_ReadMinValue"

Eingangs-Variable	Datentyp	Beschreibung
Execute	BOOL	siehe "Signallaufverhalten"
ParameterNumber	MB_IDN	An dem Eingang "ParameterNumber" ist die Identnummer des Parameters anzulegen, dessen Minimalwert gelesen werden soll

Tab. 13-127: Eingangs-Variablen des Firmware-Funktionsbausteins "MB_ReadMinValue"

Ausgangs-Variable	Datentyp	Beschreibung
Done	BOOL	Mit "Done"=TRUE zeigt der Funktionsbaustein an, dass die Bearbeitung der Arbeitsanweisung erfolgreich abgeschlossen ist
Active	BOOL	Bei entfernten Achsen wird mit "Active"=TRUE angezeigt, dass der Funktionsbaustein die Bearbeitungsanweisung angenommen hat, bei Einzelachsen wird dieser Ausgang nicht gesetzt!
Error	BOOL	Mit "Error"=TRUE meldet der Funktionsbaustein, dass ein Fehler vorliegt
ErrorID *	ERROR_CODE	"ErrorID" gibt eine grobe Fehlerbeschreibung wieder.

MX_PLCoPen

Ausgangs-Variable	Datentyp	Beschreibung
ErrorIdent *	ERROR_STRUCT	"ErrorIdent" gibt über die Elemente: "Table", "Additional1" und "Additional2" die genaue Fehlerursache wieder.
MinValue	DINT	Ist "Done"=TRUE, zeigt "MinValue" den minimalen Eingabewert des ausgewählten Parameters an. Hinweis: Je nach Wichtung besitzt der Eingabewert des Parameters Nachkommastellen die hier beachtet und verrechnet werden müssen

* siehe nachfolgender Hinweis

Tab. 13-128: *Ausgangs-Variablen des Firmware-Funktionsbausteins "MB_ReadMinValue"*



HINWEIS: Eine Auflistung aller möglichen Fehler finden Sie hier: ["Fehler-Referenzlisten"](#)

Eingangs-/Ausgangs-Variable (VAR_IN_OUT)	Datentyp	Beschreibung
Axis	AXIS_REF	Definiert die Achse. Folgende Achsen sind möglich: MPB (IndraDrive) und HDC (HydraulicDrive): Axis1 MPC (IndraDrive): - Axis1 (Master) oder - Axis2...Axis10 (insgesamt 9 Slaves / entfernte Achsen) Die in der Liste "P-0-1801.0.10, CCD: Adressen projektierter Antriebe" vorhandenen entfernten Achsen entsprechen der Reihenfolge nach Axis2 bis Axis10 (Axis2 wirkt auf den Antrieb des Listenelements 0 des P-0-1801.0.10,...) Bezüglich Zugriff auf Parameter von entfernten Achsen bitte nachfolgende Hinweise beachten!

Tab. 13-129: *Eingangs-/Ausgangs-Variable des Firmware-Funktionsbausteins "MB_ReadMinValue"*



Bei einer entfernten Achse wird der Auftrag an den Servicekanal (Sercos III) abgegeben. Erst bei einem späteren Aufruf kann der Baustein "Done"="TRUE" melden. Damit ergeben sich für die Funktionsbaustein-Aufrufe selbst immer kurze Zeiten. Die Ausgänge "Done", "Active" und "Error" müssen hier zyklisch gepollt werden. Die Übertragung der Daten kann abhängig von der Belastung des Servicekanal auch Sekunden in Anspruch nehmen.

Solange der Sercos Ring noch nicht in Phase2 ist, können keine Daten mit entfernten Achsen ausgetauscht werden. Der Baustein meldet in diesem Fall einen Fehler.



Der Zugriff auf Parameter von entfernten Achsen ist im MLD-M-Systemmodus, im CCD-Basismodus und im CCD-Systemmodus möglich.

13.4.7 MB_ReadName

Mit Hilfe des Funktionsbausteins "MB_ReadName" kann der Name eines Parameters gelesen werden.

An dem Eingang "ParameterNumber" muss die Parameternummer des Parameters angelegt werden, dessen Name gelesen werden soll. Mit einer steigenden Flanke an dem Eingang "Execute" wird die Bearbeitung des Funktionsbausteins gestartet. Eine fehlerfreie Bearbeitung wird durch "Done"=TRUE angezeigt. Eine fehlerhafte Bearbeitung wird durch "Error"=TRUE signalisiert. In einem Fehlerfall wird eine detaillierte Fehlerbeschreibung an den Ausgängen "ErrorID" und "ErrorIdent" ausgegeben.

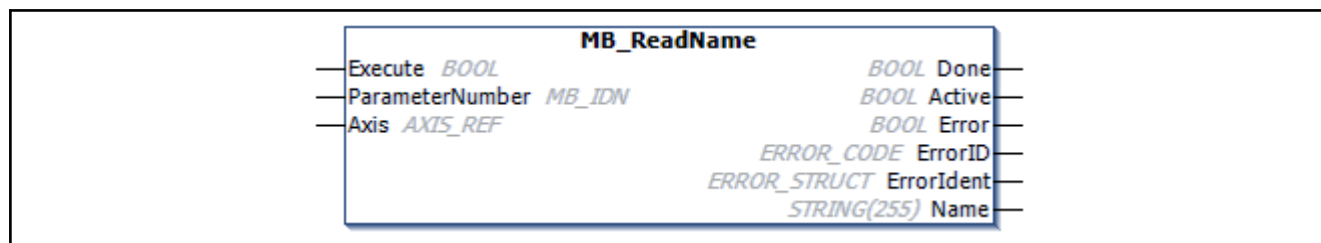


Abb. 13-56: Firmware-Funktionsbaustein "MB_ReadName"

Eingangs-Variable	Datentyp	Beschreibung
Execute	BOOL	siehe "Signallaufverhalten"
ParameterNumber	MB_IDN	An dem Eingang "ParameterNumber" ist die Identnummer des Parameters anzulegen, dessen Name gelesen werden soll

Tab. 13-130: Eingangs-Variablen des Firmware-Funktionsbausteins "MB_ReadName"

Ausgangs-Variable	Datentyp	Beschreibung
Done	BOOL	Mit "Done"=TRUE zeigt der Funktionsbaustein an, dass die Bearbeitung der Arbeitsanweisung erfolgreich abgeschlossen ist
Active	BOOL	Bei entfernten Achsen wird mit "Active"=TRUE angezeigt, dass der Funktionsbaustein die Bearbeitungsanweisung angenommen hat, bei Einzelachsen wird dieser Ausgang nicht gesetzt!
Error	BOOL	Mit "Error"=TRUE meldet der Funktionsbaustein, dass ein Fehler vorliegt
ErrorID *	ERROR_CODE	"ErrorID" gibt eine grobe Fehlerbeschreibung wieder.
ErrorIdent *	ERROR_STRUCT	"ErrorIdent" gibt über die Elemente: "Table", "Additional1" und "Additional2" die genaue Fehlerursache wieder.
Name	STRING(255)	Ist "Done"=TRUE, zeigt "Name" den Namen des ausgewählten Parameters an

* siehe nachfolgender Hinweis

Tab. 13-131: Ausgangs-Variablen des Firmware-Funktionsbausteins "MB_ReadName"



HINWEIS: Eine Auflistung aller möglichen Fehler finden Sie hier: "[Fehler-Referenzlisten](#)"

MX_PLCOpen

Eingangs-/Ausgangs-Variable (VAR_IN_OUT)	Datentyp	Beschreibung
Axis	AXIS_REF	Definiert die Achse. Folgende Achsen sind möglich: MPB (IndraDrive) und HDC (HydraulicDrive): Axis1 MPC (IndraDrive): - Axis1 (Master) oder - Axis2...Axis10 (insgesamt 9 Slaves / entfernte Achsen) Die in der Liste "P-0-1801.0.10, CCD: Adressen projektierter Antriebe" vorhandenen entfernten Achsen entsprechen der Reihenfolge nach Axis2 bis Axis10 (Axis2 wirkt auf den Antrieb des Listenelements 0 des P-0-1801.0.10,...) Bezüglich Zugriff auf Parameter von entfernten Achsen bitte nachfolgende Hinweise beachten!

Tab. 13-132: *Eingangs-/Ausgangs-Variable des Firmware-Funktionsbausteins "MB_ReadName"*



Bei einer entfernten Achse wird der Auftrag an den Servicekanal (Sercos III) abgegeben. Erst bei einem späteren Aufruf kann der Baustein "Done"="TRUE" melden. Damit ergeben sich für die Funktionsbaustein-Aufrufe selbst immer kurze Zeiten. Die Ausgänge "Done", "Active" und "Error" müssen hier zyklisch gepollt werden. Die Übertragung der Daten kann abhängig von der Belastung des Servicekanal auch Sekunden in Anspruch nehmen.

Solange der Sercos Ring noch nicht in Phase2 ist, können keine Daten mit entfernten Achsen ausgetauscht werden. Der Baustein meldet in diesem Fall einen Fehler.



Der Zugriff auf Parameter von entfernten Achsen ist im MLD-M-Systemmodus, im CCD-Basismodus und im CCD-Systemmodus möglich.

13.4.8 MB_ReadParameter

Mit dem Funktionsbaustein "MB_ReadParameter" kann der aktuelle Wert eines Parameters gelesen werden. Der Rückgabewert ist DINT.



Bei dem Funktionsbaustein "MB_ReadParameter" werden nur bei 0-1-Flanke an dem Eingang "Enable" die Eingangs-Variablen übernommen.



Listen- und String-Parameter können mit dem Funktionsbaustein nicht gelesen werden.

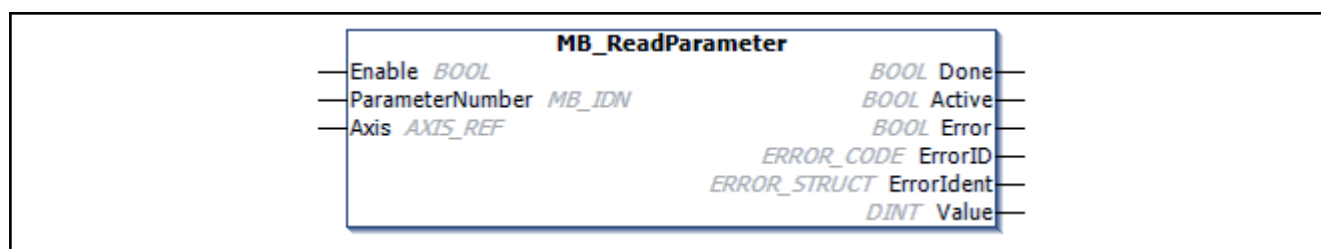


Abb. 13-57: Firmware-Funktionsbaustein "MB_ReadParameter"

Eingangs-Variable	Datentyp	Beschreibung
Enable	BOOL	Bei 0-1-Flanke an dem Eingang "Enable" werden die Eingangs-Variablen übernommen
ParameterNumber	MB_IDN	An dem Eingang "ParameterNumber" ist die Identnummer des Parameters (z. B. FP_S_0_0052) anzulegen, dessen aktueller Wert gelesen werden soll

Tab. 13-133: Eingangs-Variablen des Firmware-Funktionsbausteins "MB_ReadParameter"

Ausgangs-Variable	Datentyp	Beschreibung
Done	BOOL	Mit "Done"=TRUE zeigt der Funktionsbaustein an, dass die Bearbeitung der Arbeitsanweisung erfolgreich abgeschlossen ist
Active	BOOL	Bei entfernten Achsen wird mit "Active"=TRUE angezeigt, dass der Funktionsbaustein die Bearbeitungsanweisung angenommen hat, bei Einzelachsen wird dieser Ausgang nicht gesetzt!
Error	BOOL	Mit "Error"=TRUE meldet der Funktionsbaustein, dass ein Fehler vorliegt
ErrorID *	ERROR_CODE	"ErrorID" gibt eine grobe Fehlerbeschreibung wieder.
ErrorIdent *	ERROR_STRUCT	"ErrorIdent" gibt über die Elemente: "Table", "Additional1" und "Additional2" die genaue Fehlerursache wieder.
Value	DINT	Ist "Done"=TRUE, zeigt "Value" den Wert des ausgewählten Parameters an. Je nach Wichtung besitzt der Parameter Nachkommastellen, die hier beachtet und verrechnet werden müssen

* siehe nachfolgender Hinweis

Tab. 13-134: Ausgangs-Variablen des Firmware-Funktionsbausteins "MB_ReadParameter"



HINWEIS: Eine Auflistung aller möglichen Fehler finden Sie hier:
["Fehler-Referenzlisten"](#)

MX_PLCOpen

Eingangs-/Ausgangs-Variable (VAR_IN_OUT)	Datentyp	Beschreibung
Axis	AXIS_REF	Definiert die Achse. Folgende Achsen sind möglich: MPB (IndraDrive) und HDC (HydraulicDrive): Axis1 MPC (IndraDrive): - Axis1 (Master) oder - Axis2...Axis10 (insgesamt 9 Slaves / entfernte Achsen) Die in der Liste "P-0-1801.0.10, CCD: Adressen projektierter Antriebe" vorhandenen entfernten Achsen entsprechen der Reihenfolge nach Axis2 bis Axis10 (Axis2 wirkt auf den Antrieb des Listenelements 0 des P-0-1801.0.10,...) Bezüglich Zugriff auf Parameter von entfernten Achsen bitte nachfolgende Hinweise beachten!

Tab. 13-135: *Eingangs-/Ausgangs-Variable des Firmware-Funktionsbausteins "MB_ReadParameter"*



Bei einer entfernten Achse wird der Auftrag an den Servicekanal (Sercos III) abgegeben. Erst bei einem späteren Aufruf kann der Baustein "Done"="TRUE" melden. Damit ergeben sich für die Funktionsbaustein-Aufrufe selbst immer kurze Zeiten. Die Ausgänge "Done", "Active" und "Error" müssen hier zyklisch gepollt werden. Die Übertragung der Daten kann abhängig von der Belastung des Servicekanal auch Sekunden in Anspruch nehmen.

Solange der Sercos Ring noch nicht in Phase2 ist, können keine Daten mit entfernten Achsen ausgetauscht werden. Der Baustein meldet in diesem Fall einen Fehler.



Der Zugriff auf Parameter von entfernten Achsen ist im MLD-M-Systemmodus, im CCD-Basismodus und im CCD-Systemmodus möglich.

13.4.9 MB_ReadRealListParameter

Mit dem Funktionsbaustein "MB_ReadRealListParameter" kann die aktuelle Liste eines Listen-Parameters gelesen werden.

Die Identnummer des zu lesenden Listenparameters muss am Eingang "ParameterNumber" angelegt werden. An dem Eingang "NoOfBytes" ist die Anzahl der zu lesenden Bytes anzugeben. An dem Eingang "ValueAdr" ist die Anfangsadresse des Feldes anzugeben, in das die gelesenen Bytes geschrieben werden sollen. Die Bearbeitung des Funktionsbausteins wird mit einer steigenden Flanke am Eingang "Execute" ("Execute"=TRUE) gestartet.



Zum Testen des Funktionsbausteins "MB_ReadRealListParameter" kann die Instanzvariable "ActivateCounter" herangezogen werden. Sie zählt die positiven Flanken des Execute-Signals und zeigt damit an, ob der Baustein mit Flanke durchlaufen wird.

Lokale Achse

Wird der Funktionsbaustein bei einer lokalen Achse verwendet, so werden bei seinem Aufruf sofort alle Listenelemente des ausgewählten Parameters

gelesen. Nach dem Auslesen der Listenelemente kehrt die Kontrolle an das Programm zurück und am Ausgang "Done" liegt TRUE an.



Es sollte beachtet werden, dass der Lesevorgang in Abhängigkeit der Menge der zu lesenden Daten einige Zeit in Anspruch nehmen kann; falls notwendig sollte eine eigene Task benutzt werden.

Beispiel: Um alle Listenelemente des Parameters P-0-0072 zu lesen (1024 Langworte), werden mehrere 100 µs benötigt.

Entfernte Achse



Der Zugriff auf Parameter von entfernten Achsen ist im MLD-M-Systemmodus, im CCD-Basismodus und im CCD-Systemmodus möglich.

Wird der Funktionsbaustein bei entfernter Achse verwendet, wird der Leseauftrag an den Servicekanal (Sercos III) abgegeben. Erst bei einem späteren Aufruf kann der Funktionsbaustein "Done"=TRUE melden.

Die Ausgänge "Done", "Active" und "Error" müssen hier zyklisch gepollt werden.

Die Übertragung der Daten kann abhängig von der Menge mehrere Sekunden in Anspruch nehmen.



Solange der Sercos Ring noch nicht in Phase2 ist, können keine Daten mit entfernten Achsen ausgetauscht werden. Der Baustein meldet in diesem Fall einen Fehler.

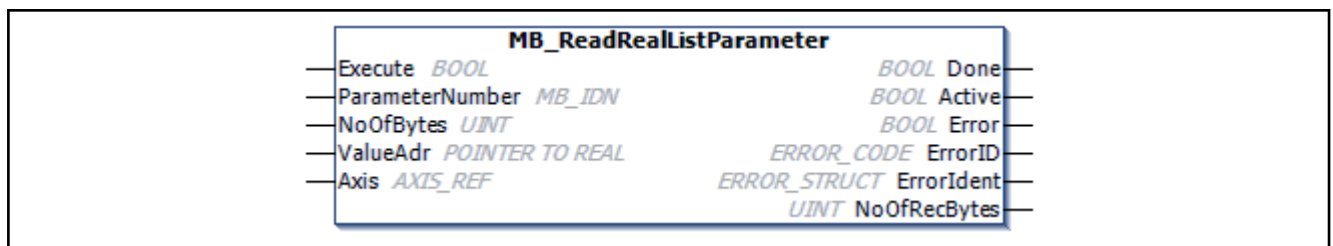


Abb. 13-58: Firmware-Funktionsbaustein "MB_ReadRealListParameter"

Eingangs-Variable	Datentyp	Beschreibung
Execute	BOOL	siehe "Signallaufverhalten"
ParameterNumber	MB_IDN	Identnummer des auszulesenden Parameters
NoOfBytes	UINT	An dem Eingang "NoOfBytes" ist die Anzahl der zu lesenden Bytes anzugeben Beispiel: NoOfBytes := SIZEOF(arFeld1)
ValueAdr	POINTER TO REAL	An dem Eingang "ValueAdr" ist die Anfangsadresse des Feldes anzugeben, in das die gelesenen Bytes geschrieben sollen. Beispiel: ValueAdr := ADR(arFeld1)

Tab. 13-136: Eingangs-Variablen des Firmware-Funktionsbausteins "MB_ReadRealListParameter"

MX_PLCOpen

Ausgangs-Variable	Datentyp	Beschreibung
Done	BOOL	Mit "Done"=TRUE zeigt der Baustein an, dass das zyklische Lesen des Datumwertes des Parameters erfolgreich war
Active	BOOL	Bei entfernten Achsen wird mit "Active"=TRUE angezeigt, dass der Funktionsbaustein die Bearbeitungsanweisung angenommen hat, bei Einzelachsen wird dieser Ausgang nicht gesetzt!
Error	BOOL	Mit "Error"=TRUE meldet der Funktionsbaustein, dass ein Fehler vorliegt
ErrorID *	ERROR_CODE	"ErrorID" gibt eine grobe Fehlerbeschreibung wieder.
ErrorIdent *	ERROR_STRUCT	"ErrorIdent" gibt über die Elemente: "Table", "Additional1" und "Additional2" die genaue Fehlerursache wieder.
NoOfRecBytes	UINT	Der Ausgang "NoOfRecBytes" zeigt an, wie viele Bytes tatsächlich gelesen wurden

* siehe nachfolgender Hinweis

Tab. 13-137: *Ausgangs-Variablen des Firmware-Funktionsbausteins "MB_ReadRealListParameter"*



HINWEIS: Eine Auflistung aller möglichen Fehler finden Sie hier: "[Fehler-Referenzlisten](#)"

Eingangs-/Ausgangs-Variable (VAR_IN_OUT)	Datentyp	Beschreibung
Axis	AXIS_REF	Definiert die Achse. Folgende Achsen sind möglich: MPB (IndraDrive) und HDC (HydraulicDrive): Axis1 MPC (IndraDrive): - Axis1 (Master) oder - Axis2...Axis10 (insgesamt 9 Slaves / entfernte Achsen) Die in der Liste "P-0-1801.0.10, CCD: Adressen projektierte Antriebe" vorhandenen entfernten Achsen entsprechen der Reihenfolge nach Axis2 bis Axis10 (Axis2 wirkt auf den Antrieb des Listenelements 0 des P-0-1801.0.10,...) Bezüglich Zugriff auf Parameter von entfernten Achsen bitte nachfolgenden Hinweis beachten!

Tab. 13-138: *Eingangs-/Ausgangs-Variable des Firmware-Funktionsbausteins "MB_ReadRealListParameter"*



Der Zugriff auf Parameter von entfernten Achsen ist im MLD-M-Systemmodus, im CCD-Basismodus und im CCD-Systemmodus möglich.

13.4.10 MB_ReadRealParameter

Mit dem Funktionsbaustein "MB_ReadRealParameter" kann der aktuelle Wert eines Parameters gelesen werden. Der Rückgabewert ist REAL.



Listen- und String-Parameter können mit dem Funktionsbaustein "MB_ReadRealParameter" nicht gelesen werden.

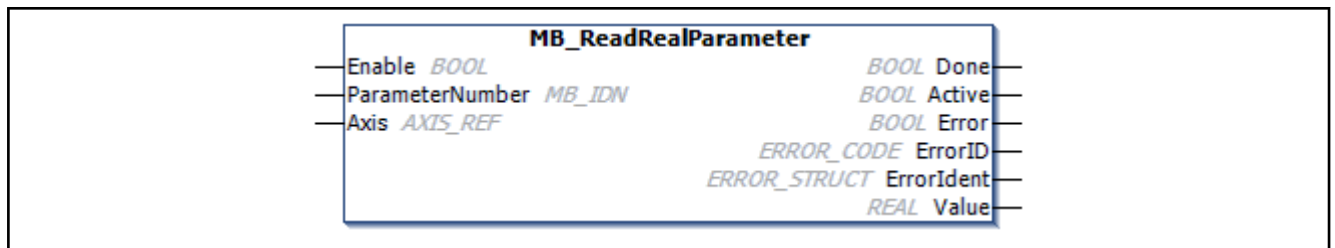


Abb. 13-59: Firmware-Funktionsbaustein "MB_ReadRealParameter"

Eingangs-Variable	Datentyp	Beschreibung
Enable	BOOL	Bei 0-1-Flanke an dem Eingang "Enable" werden die Eingangs-Variablen übernommen
ParameterNumber	MB_IDN	An dem Eingang "ParameterNumber" ist die Identnummer des Parameters anzulegen, dessen aktueller Wert gelesen werden soll. Eine neue Parameternummer wird erst mit der nächsten positiven Flanke am "Enable"-Eingang übernommen

Tab. 13-139: Eingangs-Variablen des Firmware-Funktionsbausteins "MB_ReadRealParameter"

Ausgangs-Variable	Datentyp	Beschreibung
Done	BOOL	Mit "Done"=TRUE zeigt der Funktionsbaustein an, dass die Bearbeitung der Arbeitsanweisung erfolgreich abgeschlossen ist
Active	BOOL	Bei entfernten Achsen wird mit "Active"=TRUE angezeigt, dass der Funktionsbaustein die Bearbeitungsanweisung angenommen hat, bei Einzelachsen wird dieser Ausgang nicht gesetzt!
Error	BOOL	Mit "Error"=TRUE meldet der Funktionsbaustein, dass ein Fehler vorliegt
ErrorID *	ERROR_CODE	"ErrorID" gibt eine grobe Fehlerbeschreibung wieder.
ErrorIdent *	ERROR_STRUCT	"ErrorIdent" gibt über die Elemente: "Table", "Additional1" und "Additional2" die genaue Fehlerursache wieder.
Value	REAL	Ist "Done"=TRUE, zeigt "Value" den Wert des ausgewählten Parameters an

* siehe nachfolgender Hinweis

Tab. 13-140: Ausgangs-Variablen des Firmware-Funktionsbausteins "MB_ReadRealParameter"



HINWEIS: Eine Auflistung aller möglichen Fehler finden Sie hier:
["Fehler-Referenzlisten"](#)

MX_PLCOpen

Eingangs-/Ausgangs-Variable (VAR_IN_OUT)	Datentyp	Beschreibung
Axis	AXIS_REF	Definiert die Achse. Folgende Achsen sind möglich: MPB (IndraDrive) und HDC (HydraulicDrive): Axis1 MPC (IndraDrive): - Axis1 (Master) oder - Axis2...Axis10 (insgesamt 9 Slaves / entfernte Achsen) Die in der Liste "P-0-1801.0.10, CCD: Adressen projektierter Antriebe" vorhandenen entfernten Achsen entsprechen der Reihenfolge nach Axis2 bis Axis10 (Axis2 wirkt auf den Antrieb des Listenelements 0 des P-0-1801.0.10,...) Bezüglich Zugriff auf Parameter von entfernten Achsen bitte nachfolgende Hinweise beachten!

Tab. 13-141: *Eingangs-/Ausgangs-Variable des Firmware-Funktionsbausteins "MB_ReadRealParameter"*



Bei einer entfernten Achse wird der Auftrag an den Servicekanal (Sercos III) abgegeben. Erst bei einem späteren Aufruf kann der Baustein "Done"="TRUE" melden. Damit ergeben sich für die Funktionsbaustein-Aufrufe selbst immer kurze Zeiten. Die Ausgänge "Done", "Active" und "Error" müssen hier zyklisch gepollt werden. Die Übertragung der Daten kann abhängig von der Belastung des Servicekanal auch Sekunden in Anspruch nehmen.

Solange der Sercos Ring noch nicht in Phase2 ist, können keine Daten mit entfernten Achsen ausgetauscht werden. Der Baustein meldet in diesem Fall einen Fehler.



Der Zugriff auf Parameter von entfernten Achsen ist im MLD-M-Systemmodus, im CCD-Basismodus und im CCD-Systemmodus möglich.

13.4.11 MB_ReadSercosAttribute

Mit Hilfe des Funktionsbausteins "MB_ReadSercosAttribute" kann das Sercos Attribut eines Parameters gelesen werden.

An dem Eingang "ParameterNumber" muss die Parameternummer des Parameters angelegt werden, dessen Sercos Attribut gelesen werden soll. Mit einer steigenden Flanke an dem Eingang "Execute" wird die Bearbeitung des Funktionsbausteins gestartet. Eine fehlerfreie Bearbeitung wird durch "Done"=TRUE angezeigt. Eine fehlerhafte Bearbeitung wird durch "Error"=TRUE signalisiert. In einem Fehlerfall wird eine detaillierte Fehlerbeschreibung an den Ausgängen "ErrorID" und "ErrorIdent" ausgegeben.

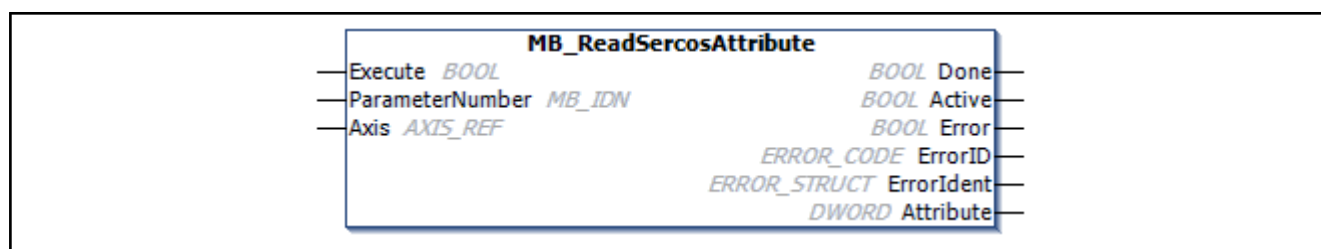


Abb. 13-60: Firmware-Funktionsbaustein "MB_ReadSercosAttribute"

Eingangs-Variable	Datentyp	Beschreibung
Execute	BOOL	siehe "Signallaufverhalten"
ParameterNumber	MB_IDN	An dem Eingang "ParameterNumber" muss die Parameternummer des Parameters angelegt werden, dessen Sercos Attribut gelesen werden soll. HINWEIS: Eine neue Parameternummer wird erst mit der nächsten positiven Flanke am Execute-Eingang übernommen.

Tab. 13-142: Eingangs-Variablen des Firmware-Funktionsbausteins "MB_ReadSercosAttribute"

Ausgangs-Variable	Datentyp	Beschreibung
Done	BOOL	Mit "Done"=TRUE zeigt der Funktionsbaustein an, dass das zyklische Lesen des Sercos Attributs erfolgreich läuft
Active	BOOL	Bei entfernten Achsen wird mit "Active"=TRUE angezeigt, dass der Funktionsbaustein die Bearbeitungsanweisung angenommen hat, bei Einzelachsen wird dieser Ausgang nicht gesetzt!
Error	BOOL	Mit "Error"=TRUE meldet der Funktionsbaustein, dass ein Fehler vorliegt
ErrorID *	ERROR_CODE	"ErrorID" gibt eine grobe Fehlerbeschreibung wieder.
ErrorIdent *	ERROR_STRUCT	"ErrorIdent" gibt über die Elemente: "Table", "Additional1" und "Additional2" die genaue Fehlerursache wieder.
Attribute	DWORD	Ist "Done"=TRUE, zeigt "Attribute" das Attribut des ausgewählten Parameters an

* siehe nachfolgender Hinweis

Tab. 13-143: Ausgangs-Variablen des Firmware-Funktionsbausteins "MB_ReadSercosAttribute"



HINWEIS: Eine Auflistung aller möglichen Fehler finden Sie hier: "[Fehler-Referenzlisten](#)"

MX_PLCOpen

Eingangs-/Ausgangs-Variable (VAR_IN_OUT)	Datentyp	Beschreibung
Axis	AXIS_REF	Definiert die Achse. Folgende Achsen sind möglich: MPB (IndraDrive) und HDC (HydraulicDrive): Axis1 MPC (IndraDrive): - Axis1 (Master) oder - Axis2...Axis10 (insgesamt 9 Slaves / entfernte Achsen) Die in der Liste "P-0-1801.0.10, CCD: Adressen projektierter Antriebe" vorhandenen entfernten Achsen entsprechen der Reihenfolge nach Axis2 bis Axis10 (Axis2 wirkt auf den Antrieb des Listenelements 0 des P-0-1801.0.10,...) Bezüglich Zugriff auf Parameter von entfernten Achsen bitte nachfolgende Hinweise beachten!

Tab. 13-144: *Eingangs-/Ausgangs-Variable des Firmware-Funktionsbausteins "MB_ReadSercosAttribute"*



Bei einer entfernten Achse wird der Auftrag an den Servicekanal (Sercos III) abgegeben. Erst bei einem späteren Aufruf kann der Baustein "Done"="TRUE" melden. Damit ergeben sich für die Funktionsbaustein-Aufrufe selbst immer kurze Zeiten. Die Ausgänge "Done", "Active" und "Error" müssen hier zyklisch gepollt werden. Die Übertragung der Daten kann abhängig von der Belastung des Servicekanal auch Sekunden in Anspruch nehmen.

Solange der Sercos Ring noch nicht in Phase2 ist, können keine Daten mit entfernten Achsen ausgetauscht werden. Der Baustein meldet in diesem Fall einen Fehler.



Der Zugriff auf Parameter von entfernten Achsen ist im MLD-M-Systemmodus, im CCD-Basismodus und im CCD-Systemmodus möglich.

13.4.12 MB_ReadSercosDataStatus

Mit Hilfe des Funktionsbausteins "MB_ReadSercosDataStatus" kann der Status eines Parameters gelesen werden.

An dem Eingang "ParameterNumber" muss die Parameternummer des Parameters angelegt werden, dessen Status gelesen werden soll.

Eine fehlerfreie Bearbeitung wird durch "Done"=TRUE angezeigt. Eine fehlerhafte Bearbeitung wird durch "Error"=TRUE signalisiert. In einem Fehlerfall wird eine detaillierte Fehlerbeschreibung an den Ausgängen "ErrorID" und

"ErrorIdent" ausgegeben.



Abb. 13-61: Firmware-Funktionsbaustein "MB_ReadSercosDataStatus"

Eingangs-Variable	Datentyp	Beschreibung
Enable	BOOL	Der Funktionsbaustein ist zustandsgesteuert. Mit der positiven Flanke von "Enable" werden die Eingangsvariablen übernommen. Neue Eingangswerte werden erst über eine erneute positive Flanke von "Enable" wirksam.
ParameterNumber	MB_IDN	An dem Eingang "ParameterNumber" muss die Parameternummer des Parameters angelegt werden, dessen Status gelesen werden soll

Tab. 13-145: Eingangs-Variablen des Firmware-Funktionsbausteins "MB_ReadSercosDataStatus"

Ausgangs-Variable	Datentyp	Beschreibung
Done	BOOL	Mit "Done"=TRUE zeigt der Funktionsbaustein an, dass die Bearbeitung des Funktionsbausteins erfolgreich abgeschlossen ist
Active	BOOL	Bei entfernten Achsen wird mit "Active"=TRUE angezeigt, dass der Funktionsbaustein die Bearbeitungsanweisung angenommen hat, bei Einzelachsen wird dieser Ausgang nicht gesetzt!
Error	BOOL	Mit "Error"=TRUE meldet der Funktionsbaustein, dass ein Fehler vorliegt
ErrorID *	ERROR_CODE	"ErrorID" gibt eine grobe Fehlerbeschreibung wieder.
ErrorIdent *	ERROR_STRUCT	"ErrorIdent" gibt über die Elemente: "Table", "Additional1" und "Additional2" die genaue Fehlerursache wieder.
Status	MB_SERCOS_DATA_STATUS	Der Ausgang zeigt den aktuellen Status des Parameters an. Enum-Werte für Ausgang "Status": "MB_SERCOS_DATA_NOT_SET": Kommando ist nicht gesetzt "MB_SERCOS_DATA_SETTING": Kommando ist gesetzt "MB_SERCOS_DATA_END_OK": Kommando fehlerfrei beendet "MB_SERCOS_DATA_INTERRUPTED": Kommando unterbrochen "MB_SERCOS_DATA_IN_PROCESS": Kommando in Bearbeitung "MB_SERCOS_DATA_BREAK_ERROR": Kommando mit Fehler abgebrochen "MB_SERCOS_DATA_DATUM_FAULT": Betriebsdatum ist ungültig

* siehe nachfolgender Hinweis

Tab. 13-146: Ausgangs-Variablen des Firmware-Funktionsbausteins "MB_ReadSercosDataStatus"



HINWEIS: Eine Auflistung aller möglichen Fehler finden Sie hier:
["Fehler-Referenzlisten"](#)

MX_PLCOpen

Eingangs-/Ausgangs-Variable (VAR_IN_OUT)	Datentyp	Beschreibung
Axis	AXIS_REF	Definiert die Achse. Folgende Achsen sind möglich: MPB (IndraDrive) und HDC (HydraulicDrive): Axis1 MPC (IndraDrive): - Axis1 (Master) oder - Axis2...Axis10 (insgesamt 9 Slaves / entfernte Achsen) Die in der Liste "P-0-1801.0.10, CCD: Adressen projektierter Antriebe" vorhandenen entfernten Achsen entsprechen der Reihenfolge nach Axis2 bis Axis10 (Axis2 wirkt auf den Antrieb des Listenelements 0 des P-0-1801.0.10,...) Bezüglich Zugriff auf Parameter von entfernten Achsen bitte nachfolgende Hinweise beachten!

Tab. 13-147: *Eingangs-/Ausgangs-Variable des Firmware-Funktionsbausteins "MB_ReadSercosDataStatus"*



Bei einer entfernten Achse wird der Auftrag an den Servicekanal (Sercos III) abgegeben. Erst bei einem späteren Aufruf kann der Baustein "Done"="TRUE" melden. Damit ergeben sich für die Funktionsbaustein-Aufrufe selbst immer kurze Zeiten. Die Ausgänge "Done", "Active" und "Error" müssen hier zyklisch gepollt werden. Die Übertragung der Daten kann abhängig von der Belastung des Servicekanal auch Sekunden in Anspruch nehmen.

Solange der Sercos Ring noch nicht in Phase2 ist, können keine Daten mit entfernten Achsen ausgetauscht werden. Der Baustein meldet in diesem Fall einen Fehler.



Der Zugriff auf Parameter von entfernten Achsen ist im MLD-M-Systemmodus, im CCD-Basismodus und im CCD-Systemmodus möglich.

13.4.13 MB_ReadStringParameter

Mit Hilfe des Funktionsbausteins "MB_ReadStringParameter" kann ein Parameter gelesen werden, dessen Betriebsdatum ASCII-Format hat (String).

An dem Eingang "ParameterNumber" muss die Parameternummer des Parameters angelegt werden, dessen Betriebsdatum gelesen werden soll. Mit einer steigenden Flanke an dem Eingang "Execute" wird die Bearbeitung des Funktionsbausteins gestartet.



Zum Testen des Funktionsbausteins kann die Instanzvariable "ActivateCounter" herangezogen werden. Sie zählt die positiven Flanken des "Execute"-Signals und zeigt damit an, ob der Funktionsbaustein mit Flanke durchlaufen wird.

Lokale Achse

Wird der Funktionsbaustein bei einer lokalen Achse verwendet, so werden bei seinem Aufruf sofort alle Zeichen des ausgewählten ASCII-Parameters gelesen. Danach kehrt die Kontrolle an das Programm zurück und am Ausgang "Done" liegt TRUE an.

Entfernte Achse



Der Lesevorgang kann in Abhängigkeit von der Textlänge einige Zeit in Anspruch nehmen. Falls notwendig sollte eine eigene Task benutzt werden.



Der Zugriff auf Parameter von entfernten Achsen ist im MLD-M-Systemmodus, im CCD-Basismodus und im CCD-Systemmodus möglich.

Wird der Funktionsbaustein bei entfernter Achse verwendet, wird der Leseauftrag an den Servicekanal (Sercos III) abgegeben. Erst bei einem späteren Aufruf kann der Baustein "Done" melden.

Die Ausgänge "Done", "Active" und "Error" müssen hier zyklisch gepollt werden.

Die Übertragung der Daten kann abhängig von der Textlänge einige Millisekunden in Anspruch nehmen.



Solange der Sercos Ring noch nicht in Phase2 ist, können keine Daten mit entfernten Achsen ausgetauscht werden. Der Baustein meldet in diesem Fall einen Fehler.

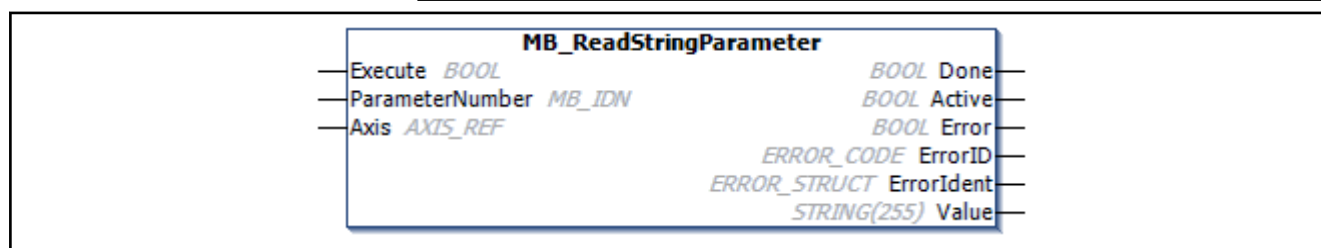


Abb. 13-62: Firmware-Funktionsbaustein "MB_ReadStringParameter"

Eingangs-Variable	Datentyp	Beschreibung
Execute	BOOL	siehe "Signallaufverhalten"
ParameterNumber	DINT	An dem Eingang "ParameterNumber" muss die Parameternummer des Parameters angelegt werden, dessen Betriebsdatum gelesen werden soll

Tab. 13-148: Eingangs-Variablen des Firmware-Funktionsbausteins "MB_ReadStringParameter"

Ausgangs-Variable	Datentyp	Beschreibung
Done	BOOL	Mit "Done"=TRUE zeigt der Funktionsbaustein an, dass die Bearbeitung der Arbeitsanweisung erfolgreich abgeschlossen ist
Active	BOOL	Bei entfernten Achsen wird mit "Active"=TRUE angezeigt, dass der Funktionsbaustein die Bearbeitungsanweisung angenommen hat, bei Einzelachsen wird dieser Ausgang nicht gesetzt!
Error	BOOL	Mit "Error"=TRUE meldet der Funktionsbaustein, dass ein Fehler vorliegt
ErrorID *	ERROR_CODE	"ErrorID" gibt eine grobe Fehlerbeschreibung wieder.
ErrorIdent *	ERROR_STRUCT	"ErrorIdent" gibt über die Elemente: "Table", "Additional1" und "Additional2" die genaue Fehlerursache wieder.
Value	STRING(255)	"Value" liefert das Betriebsdatum des ausgewählten Parameters

* siehe nachfolgender Hinweis

Tab. 13-149: Ausgangs-Variablen des Firmware-Funktionsbausteins "MB_ReadStringParameter"



HINWEIS: Eine Auflistung aller möglichen Fehler finden Sie hier:
["Fehler-Referenzlisten"](#)

Eingangs-/Ausgangs-Variable (VAR_IN_OUT)	Datentyp	Beschreibung
Axis	AXIS_REF	Definiert die Achse. Folgende Achsen sind möglich: MPB (IndraDrive) und HDC (HydraulicDrive): Axis1 MPC (IndraDrive): - Axis1 (Master) oder - Axis2...Axis10 (insgesamt 9 Slaves / entfernte Achsen) Die in der Liste "P-0-1801.0.10, CCD: Adressen projektierte Antriebe" vorhandenen entfernten Achsen entsprechen der Reihenfolge nach Axis2 bis Axis10 (Axis2 wirkt auf den Antrieb des Listenelements 0 des P-0-1801.0.10,...)

Tab. 13-150: *Eingangs-/Ausgangs-Variable des Firmware-Funktionsbausteins "MB_ReadStringParameter"*

13.4.14 MB_ReadUnit

Mit Hilfe des Funktionsbausteins "MB_ReadUnit" kann die Einheit eines Parameters gelesen werden.

An dem Eingang "ParameterNumber" muss die Parameternummer des Parameters angelegt werden, dessen Einheit gelesen werden soll. Mit einer steigenden Flanke an dem Eingang "Execute" wird die Bearbeitung des Funktionsbausteins gestartet. Eine fehlerfreie Bearbeitung wird durch "Done"=TRUE angezeigt. Eine fehlerhafte Bearbeitung wird durch "Error"=TRUE signalisiert. In einem Fehlerfall wird eine detaillierte Fehlerbeschreibung an den Ausgängen "ErrorID" und "ErrorIdent" ausgegeben.

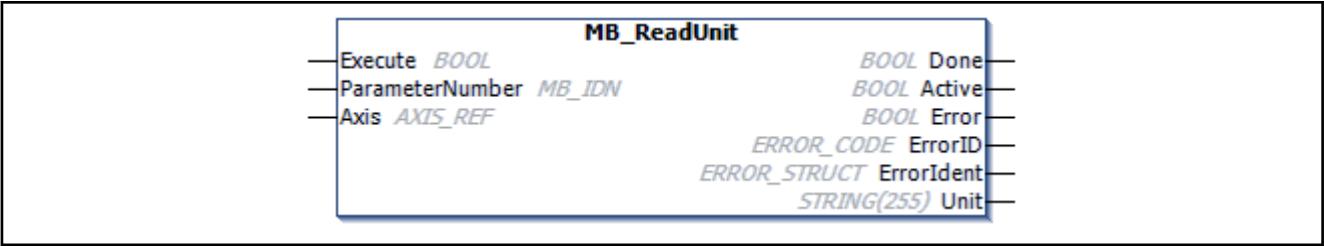


Abb. 13-63: *Firmware-Funktionsbaustein "MB_ReadUnit"*

Eingangs-Variable	Datentyp	Beschreibung
Execute	BOOL	siehe "Signallaufverhalten"
ParameterNumber	MB_IDN	An dem Eingang "ParameterNumber" muss die Parameternummer des Parameters angelegt werden, dessen Einheit gelesen werden soll

Tab. 13-151: *Eingangs-Variablen des Firmware-Funktionsbausteins "MB_ReadUnit"*

Ausgangs-Variable	Datentyp	Beschreibung
Done	BOOL	Mit "Done"=TRUE zeigt der Funktionsbaustein an, dass die Bearbeitung der Arbeitsanweisung erfolgreich abgeschlossen ist
Active	BOOL	Bei entfernten Achsen wird mit "Active"=TRUE angezeigt, dass der Funktionsbaustein die Bearbeitungsanweisung angenommen hat, bei Einzelachsen wird dieser Ausgang nicht gesetzt!
Error	BOOL	Mit "Error"=TRUE meldet der Funktionsbaustein, dass ein Fehler vorliegt
ErrorID *	ERROR_CODE	"ErrorID" gibt eine grobe Fehlerbeschreibung wieder.
ErrorIdent *	ERROR_STRUCT	"ErrorIdent" gibt über die Elemente: "Table", "Additional1" und "Additional2" die genaue Fehlerursache wieder.
Unit	STRING(255)	Ist "Done"=TRUE, zeigt "Unit" die Einheit des ausgewählten Parameters an

* siehe nachfolgender Hinweis

Tab. 13-152: *Ausgangs-Variablen des Firmware-Funktionsbausteins "MB_ReadUnit"*



HINWEIS: Eine Auflistung aller möglichen Fehler finden Sie hier: ["Fehler-Referenzlisten"](#)

Eingangs-/Ausgangs-Variable (VAR_IN_OUT)	Datentyp	Beschreibung
Axis	AXIS_REF	Definiert die Achse. Folgende Achsen sind möglich: MPB (IndraDrive) und HDC (HydraulicDrive): Axis1 MPC (IndraDrive): - Axis1 (Master) oder - Axis2...Axis10 (insgesamt 9 Slaves / entfernte Achsen) Die in der Liste "P-0-1801.0.10, CCD: Adressen projektierter Antriebe" vorhandenen entfernten Achsen entsprechen der Reihenfolge nach Axis2 bis Axis10 (Axis2 wirkt auf den Antrieb des Listenelements 0 des P-0-1801.0.10,...) Bezüglich Zugriff auf Parameter von entfernten Achsen bitte nachfolgende Hinweise beachten!

Tab. 13-153: *Eingangs-/Ausgangs-Variable des Firmware-Funktionsbausteins "MB_ReadUnit"*



Bei einer entfernten Achse wird der Auftrag an den Servicekanal (Sercos III) abgegeben. Erst bei einem späteren Aufruf kann der Baustein "Done"=TRUE melden. Damit ergeben sich für die Funktionsbaustein-Aufrufe selbst immer kurze Zeiten. Die Ausgänge "Done", "Active" und "Error" müssen hier zyklisch gepollt werden. Die Übertragung der Daten kann abhängig von der Belastung des Servicekanal auch Sekunden in Anspruch nehmen.

Solange der Sercos Ring noch nicht in Phase2 ist, können keine Daten mit entfernten Achsen ausgetauscht werden. Der Baustein meldet in diesem Fall einen Fehler.

MX_PLCoPen



Der Zugriff auf Parameter von entfernten Achsen ist im MLD-M-Systemmodus, im CCD-Basismodus und im CCD-Systemmodus möglich.

13.4.15 MB_WriteListParameter

Mit dem Funktionsbaustein "MB_WriteListParameter" kann ein Listen-Parameter mit definierten Werten beschrieben werden.

Die Identnummer des zu beschreibenden Parameters muss an dem Eingang "ParameterNumber" angelegt werden.



Abhängig vom Parameter "S-0-0269, Speicherungsmodus" wird der geänderte Parameterwert nach jeder Änderung sofort oder erst durch die Ausführung des Kommandos "S-0-0264, C2200 Kommando Arbeitsspeicher sichern" im geräteinternen, nicht flüchtigen Speicher ("Flash") gespeichert.

HINWEIS

Zerstörung des Parameter-Festwertspeichers durch ständiges Aktivieren/Ausführen des Funktionsbausteins !

Bei häufiger Ausführung des Funktionsbausteins "S-0-0269, Speicherungsmodus" so einstellen, dass geänderte Parameterwerte nur durch die Ausführung des Kommandos "S-0-0264, C2200 Kommando Arbeitsspeicher sichern" im geräteinternen, nicht flüchtigen Speicher ("Flash") gespeichert werden.



Zum Testen des Funktionsbausteins "MB_WriteListParameter" kann die Instanzvariable "ActivateCounter" herangezogen werden. Sie zählt die positiven Flanken des Execute-Signals und zeigt damit an, ob der Baustein mit Flanke durchlaufen wird.

Lokale Achse

Wird der Funktionsbaustein bei einer lokalen Achse verwendet, so werden bei seinem Aufruf sofort alle Listenelemente des angelegten Arrays geschrieben. Danach kehrt die Kontrolle an das Programm zurück und am Ausgang "Done" liegt TRUE an.



Es sollte beachtet werden, dass der Schreibvorgang in Abhängigkeit der Menge der zu schreibenden Daten einige Zeit in Anspruch nehmen kann; falls notwendig sollte eine eigene Task benutzt werden.

Beispiel: Um alle Listenelemente des Parameters P-0-0072 zu beschreiben (1024 Langworte), werden mehrere Millisekunden benötigt.

Besser ist es, wenn nur die tatsächlich benötigten Elemente geschrieben werden; sollen beispielsweise 8 Elemente des Parameters "P-0-0072, Kurventabelle1" geschrieben werden (ein Element des Parameters P-0-0072 besteht aus 4 Bytes), dann muss an "NoOfBytes"=32 angelegt werden (8*4).

Entfernte Achse



Der Zugriff auf Parameter von entfernten Achsen ist im MLD-M-Systemmodus, im CCD-Basismodus und im CCD-Systemmodus möglich.

Wird der Funktionsbaustein bei entfernter Achse verwendet, wird der Schreibauftrag an den Servicekanal (Sercos III) abgegeben. Erst bei einem späteren Aufruf kann der Funktionsbaustein "Done" melden.

Die Ausgänge "Done", "Active" und "Error" müssen hier zyklisch gepollt werden.

Die Übertragung der Daten kann abhängig von der Menge mehrere Sekunden in Anspruch nehmen.



Solange der Sercos Ring noch nicht in Phase2 ist, können keine Daten mit entfernten Achsen ausgetauscht werden. Der Baustein meldet in diesem Fall einen Fehler.

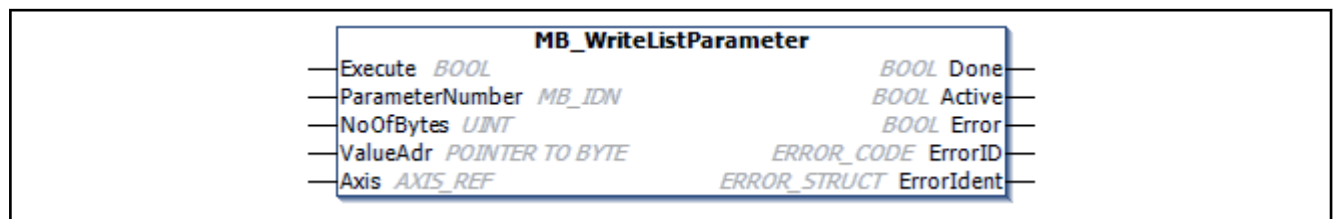


Abb. 13-64: Firmware-Funktionsbaustein "MB_WriteListParameter"

Eingangs-Variable	Datentyp	Beschreibung
Execute	BOOL	siehe "Signallaufverhalten"
ParameterNumber	MB_IDN	An den Eingang "ParameterNumber" muss die Identnummer des zu beschreibenden Listen-Parameters angelegt werden
NoOfBytes	UINT	An dem Eingang "NoOfBytes" ist die Anzahl der zu schreibenden Bytes anzugeben. Beispiel: NoOfBytes:=SIZEOF(arFeld1)
ValueAdr	POINTER TO BYTE	An dem Eingang "ValueAdr" ist die Adresse des Feldes anzugeben, in dem die zu schreibenden Daten/Elemente für den Listen-Parameter liegen. Beispiel: ValueAdr := ADR(arFeld1) Hinweis: Die Adresse muss zur Speicherausrichtung (Alignment) der Daten passen, auf die zugegriffen wird: • 1-Byte-Listen: jede Adresse • 2-Byte-Listen: jede gerade Adresse • 4-Byte-Listen: jede durch 4 teilbare Adresse

Tab. 13-154: Eingangs-Variablen des Firmware-Funktionsbausteins "MB_WriteListParameter"

Ausgangs-Variable	Datentyp	Beschreibung
Done	BOOL	Mit "Done"=TRUE zeigt der Funktionsbaustein an, dass die Bearbeitung der Arbeitsanweisung erfolgreich abgeschlossen ist
Active	BOOL	Bei entfernten Achsen wird mit "Active"=TRUE angezeigt, dass der Funktionsbaustein die Bearbeitungsanweisung angenommen hat, bei Einzelachsen wird dieser Ausgang nicht gesetzt!
Error	BOOL	Mit "Error"=TRUE meldet der Funktionsbaustein, dass ein Fehler vorliegt

MX_PLCOpen

Ausgangs-Variable	Datentyp	Beschreibung
ErrorID *	ERROR_CODE	"ErrorID" gibt eine grobe Fehlerbeschreibung wieder.
ErrorIdent *	ERROR_STRUCT	"ErrorIdent" gibt über die Elemente: "Table", "Additional1" und "Additional2" die genaue Fehlerursache wieder.

* siehe nachfolgender Hinweis

Tab. 13-155: *Ausgangs-Variablen des Firmware-Funktionsbausteins "MB_WriteListParameter"*



HINWEIS: Eine Auflistung aller möglichen Fehler finden Sie hier: ["Fehler-Referenzlisten"](#)

Eingangs-/Ausgangs-Variable (VAR_IN_OUT)	Datentyp	Beschreibung
Axis	AXIS_REF	Definiert die Achse. Folgende Achsen sind möglich: MPB (IndraDrive) und HDC (HydraulicDrive): Axis1 MPC (IndraDrive): - Axis1 (Master) oder - Axis2...Axis10 (insgesamt 9 Slaves / entfernte Achsen) Die in der Liste "P-0-1801.0.10, CCD: Adressen projektierte Antriebe" vorhandenen entfernten Achsen entsprechen der Reihenfolge nach Axis2 bis Axis10 (Axis2 wirkt auf den Antrieb des Listenelements 0 des P-0-1801.0.10,...)

Tab. 13-156: *Eingangs-/Ausgangs-Variable des Firmware-Funktionsbausteins "MB_WriteListParameter"*

13.4.16 MB_WriteParameter

Mit dem Funktionsbaustein "MB_WriteParameter" kann ein Parameter mit einem definierten Wert beschrieben werden. Dazu muss an dem Eingang "ParameterNumber" die Parameternummer des Parameters angelegt werden, dessen Wert beschrieben werden soll. Der Wert, mit dem der Parameter beschrieben werden soll, muss an dem Eingang "Value" angelegt werden und vom Typ DINT sein.



Listen- und String-Parameter können mit "MB_WriteParameter" nicht beschrieben werden.



Je nach Wichtung besitzt der Parameter Nachkommastellen, die beachtet und verrechnet werden müssen!



Abhängig vom Parameter "S-0-0269, Speicherungsmodus" wird der geänderte Parameterwert nach jeder Änderung sofort oder erst durch die Ausführung des Kommandos "S-0-0264, C2200 Kommando Arbeitsspeicher sichern" im geräteinternen, nicht flüchtigen Speicher ("Flash") gespeichert.

HINWEIS

Zerstörung des Parameter-Festwertspeichers durch ständiges Aktivieren/Ausführen des Funktionsbausteins !

Bei häufiger Ausführung des Funktionsbausteins "S-0-0269, Speicherungsmodus" so einstellen, dass geänderte Parameterwerte nur durch die Ausführung des Kommandos "S-0-0264, C2200 Kommando Arbeitsspeicher sichern" im geräteinternen, nicht flüchtigen Speicher ("Flash") gespeichert werden.



Zum Testen des Funktionsbausteins "MB_WriteParameter" kann die Instanzvariable "ActivateCounter" herangezogen werden. Sie zählt die positiven Flanken des "Execute"-Signals und zeigt damit an, ob der Baustein mit Flanke durchlaufen wird.

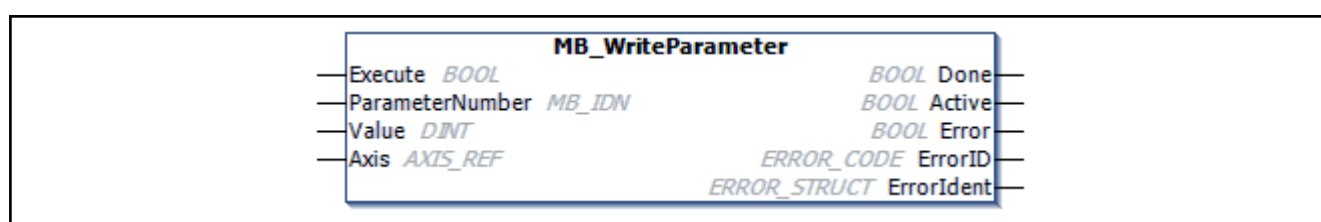


Abb. 13-65: Firmware-Funktionsbaustein "MB_WriteParameter"

Eingangs-Variable	Datentyp	Beschreibung
Execute	BOOL	siehe "Signallaufverhalten"
ParameterNumber	MB_IDN	An dem Eingang "ParameterNumber" muss die Parameternummer des zu beschreibenden Parameters angelegt werden
Value	DINT	An den Eingang "Value" muss der Wert gelegt werden, mit dem der Parameter beschrieben werden soll. Die Wichtung des ausgewählten Parameters ist dabei zu beachten

Tab. 13-157: Eingangs-Variablen des Firmware-Funktionsbausteins "MB_WriteParameter"

Ausgangs-Variable	Datentyp	Beschreibung
Done	BOOL	Mit "Done"=TRUE zeigt der Funktionsbaustein an, dass die Bearbeitung der Arbeitsanweisung erfolgreich abgeschlossen ist
Active	BOOL	Bei entfernten Achsen wird mit "Active"=TRUE angezeigt, dass der Funktionsbaustein die Bearbeitungsanweisung angenommen hat, bei Einzelachsen wird dieser Ausgang nicht gesetzt!
Error	BOOL	Mit "Error"=TRUE meldet der Funktionsbaustein, dass ein Fehler vorliegt
ErrorID *	ERROR_CODE	"ErrorID" gibt eine grobe Fehlerbeschreibung wieder.
ErrorIdent *	ERROR_STRUCT	"ErrorIdent" gibt über die Elemente: "Table", "Additional1" und "Additional2" die genaue Fehlerursache wieder.

* siehe nachfolgender Hinweis

Tab. 13-158: Ausgangs-Variablen des Firmware-Funktionsbausteins "MB_WriteParameter"



HINWEIS: Eine Auflistung aller möglichen Fehler finden Sie hier: ["Fehler-Referenzlisten"](#)

MX_PLCOpen

Eingangs-/Ausgangs-Variable (VAR_IN_OUT)	Datentyp	Beschreibung
Axis	AXIS_REF	Definiert die Achse. Folgende Achsen sind möglich: MPB (IndraDrive) und HDC (HydraulicDrive): Axis1 MPC (IndraDrive): - Axis1 (Master) oder - Axis2...Axis10 (insgesamt 9 Slaves / entfernte Achsen) Die in der Liste "P-0-1801.0.10, CCD: Adressen projektierter Antriebe" vorhandenen entfernten Achsen entsprechen der Reihenfolge nach Axis2 bis Axis10 (Axis2 wirkt auf den Antrieb des Listenelements 0 des P-0-1801.0.10,...) Bezüglich Zugriff auf Parameter von entfernten Achsen bitte nachfolgende Hinweise beachten!

Tab. 13-159: *Eingangs-/Ausgangs-Variable des Firmware-Funktionsbausteins "MB_WriteParameter"*



Bei einer entfernten Achse wird der Auftrag an den Servicekanal (Sercos III) abgegeben. Erst bei einem späteren Aufruf kann der Baustein "Done"="TRUE" melden. Damit ergeben sich für die Funktionsbaustein-Aufrufe selbst immer kurze Zeiten. Die Ausgänge "Done", "Active" und "Error" müssen hier zyklisch gepollt werden. Die Übertragung der Daten kann abhängig von der Belastung des Servicekanal auch Sekunden in Anspruch nehmen.

Solange der Sercos Ring noch nicht in Phase2 ist, können keine Daten mit entfernten Achsen ausgetauscht werden. Der Baustein meldet in diesem Fall einen Fehler.



Der Zugriff auf Parameter von entfernten Achsen ist im MLD-M-Systemmodus, im CCD-Basismodus und im CCD-Systemmodus möglich.

13.4.17 MB_WriteRealListParameter

Mit dem Funktionsbaustein "MB_WriteRealListParameter" kann ein Listen-Parameter mit definierten Werten beschrieben werden.

Die Identnummer des zu beschreibenden Listen-Parameters muss an dem Eingang "ParameterNumber" angelegt werden.



Abhängig vom Parameter "S-0-0269, Speicherungsmodus" wird der geänderte Parameterwert nach jeder Änderung sofort oder erst durch die Ausführung des Kommandos "S-0-0264, C2200 Kommando Arbeitsspeicher sichern" im geräteinternen, nicht flüchtigen Speicher ("Flash") gespeichert.

HINWEIS

Zerstörung des Parameter-Festwertspeichers durch ständiges Aktivieren/Ausführen des Funktionsbausteins !

Bei häufiger Ausführung des Funktionsbausteins "S-0-0269, Speicherungsmodus" so einstellen, dass geänderte Parameterwerte nur durch die Ausführung des Kommandos "S-0-0264, C2200 Kommando Arbeitsspeicher sichern" im geräteinternen, nicht flüchtigen Speicher ("Flash") gespeichert werden.



Zum Testen des Funktionsbausteins "MB_WriteRealListParameter" kann die Instanzvariable "ActivateCounter" herangezogen werden. Sie zählt die positiven Flanken des Execute-Signals und zeigt damit an, ob der Baustein mit Flanke durchlaufen wird.

Die Eingänge "NoOfBytes" und "ValueAdr" sollten wie folgt verwendet werden:

Beispiel:

```
inst_MB_WriteRealListParameter : MB_WriteRealListParameter;  
arFeld1 : ARRAY[1..16] OF REAL;  
inst_MB_WriteRealListParameter(...,  
NoOfBytes:=SIZEOF(arFeld1), ValueAdr:=ADR(arFeld1), ...);
```

Lokale Achse

Wird der Funktionsbaustein bei einer lokalen Achse verwendet, so werden bei seinem Aufruf sofort alle Listenelemente des angelegten Arrays geschrieben. Danach kehrt die Kontrolle an das Programm zurück und am Ausgang "Done" liegt TRUE an.



Es sollte beachtet werden, dass der Schreibvorgang in Abhängigkeit der Menge der zu schreibenden Daten einige Zeit in Anspruch nehmen kann. Falls notwendig sollte eine eigene Task benutzt werden.

Beispiel: Um beispielsweise alle Listenelemente des Parameters P-0-0072 zu beschreiben (1024 Langworte), werden mehrere Millisekunden benötigt.

Entfernte Achse



Der Zugriff auf Parameter von entfernten Achsen ist im MLD-M-Systemmodus, im CCD-Basismodus und im CCD-Systemmodus möglich.

Wird der Funktionsbaustein bei entfernten Achsen verwendet, wird der Schreibauftrag an den Servicekanal (Sercos III) abgegeben. Erst bei einem späteren Aufruf kann der Funktionsbaustein "Done"=TRUE melden.

Die Ausgänge "Done", "Active" und "Error" müssen hier zyklisch gepollt werden.

Die Übertragung der Daten kann abhängig von der Menge mehrere Sekunden in Anspruch nehmen.



Solange der Sercos Ring noch nicht in Phase2 ist, können keine Daten mit entfernten Achsen ausgetauscht werden. Der Baustein meldet in diesem Fall einen Fehler.

MX_PLCoPen

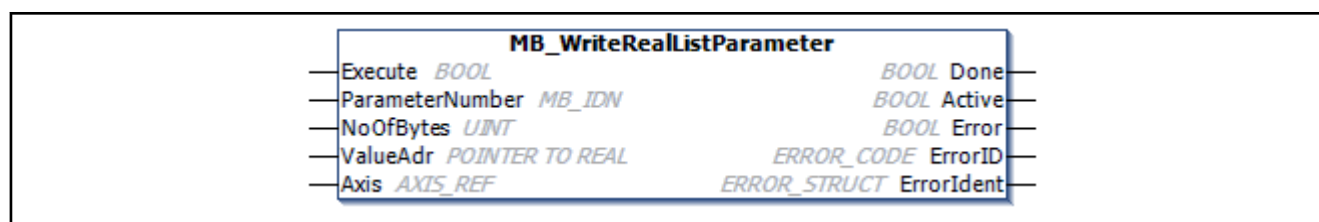


Abb. 13-66: Firmware-Funktionsbaustein "MB_WriteRealListParameter"

Eingangs-Variable	Datentyp	Beschreibung
Execute	BOOL	siehe "Signallaufverhalten"
ParameterNumber	MB_IDN	An den Eingang "ParameterNumber" muss die Identnummer des zu beschreibenden Listen-Parameters angelegt werden
NoOfBytes	UINT	An dem Eingang "NoOfBytes" ist die Anzahl der zu schreibenden Bytes anzugeben Beispiel: NoOfBytes := SIZEOF(arFeld1); Hinweis: Da die Listenelemente in einen REAL-Array (REAL-Datentyp = 4-Byte) abgelegt sind, berechnen sich die NoOfBytes wie folgt: NoOfBytes = 4 * (Anzahl der Elemente)
ValueAdr	POINTER TO REAL	An dem Eingang "ValueAdr" ist die Anfangsadresse des Feldes anzugeben, in das die gelesenen Bytes geschrieben werden sollen. Beispiel: ValueAdr := ADR(arFeld1); Hinweis: Das angelegte Feld muss vom Datentyp REAL sein.

Tab. 13-160: Eingangs-Variablen des Firmware-Funktionsbausteins "MB_WriteRealListParameter"

Ausgangs-Variable	Datentyp	Beschreibung
Done	BOOL	Mit "Done"=TRUE zeigt der Funktionsbaustein an, dass die Bearbeitung der Arbeitsanweisung erfolgreich abgeschlossen ist
Active	BOOL	Bei entfernten Achsen wird mit "Active"=TRUE angezeigt, dass der Funktionsbaustein die Bearbeitungsanweisung angenommen hat, bei Einzelachsen wird dieser Ausgang nicht gesetzt!
Error	BOOL	Mit "Error"=TRUE meldet der Funktionsbaustein, dass ein Fehler vorliegt
ErrorID *	ERROR_CODE	"ErrorID" gibt eine grobe Fehlerbeschreibung wieder.
ErrorIdent *	ERROR_STRUCT	"ErrorIdent" gibt über die Elemente: "Table", "Additional1" und "Additional2" die genaue Fehlerursache wieder.

* siehe nachfolgender Hinweis

Tab. 13-161: Ausgangs-Variablen des Firmware-Funktionsbausteins "MB_WriteRealListParameter"



HINWEIS: Eine Auflistung aller möglichen Fehler finden Sie hier:
"Fehler-Referenzlisten"

Eingangs-/Ausgangs-Variable (VAR_IN_OUT)	Datentyp	Beschreibung
Axis	AXIS_REF	Definiert die Achse. Folgende Achsen sind möglich: MPB (IndraDrive) und HDC (HydraulicDrive): Axis1 MPC (IndraDrive): - Axis1 (Master) oder - Axis2...Axis10 (insgesamt 9 Slaves / entfernte Achsen) Die in der Liste "P-0-1801.0.10, CCD: Adressen projektierte Antriebe" vorhandenen entfernten Achsen entsprechen der Reihenfolge nach Axis2 bis Axis10 (Axis2 wirkt auf den Antrieb des Listenelements 0 des P-0-1801.0.10,...)

Tab. 13-162: Eingangs-/Ausgangs-Variable des Firmware-Funktionsbausteins "MB_WriteRealListParameter"

13.4.18 MB_WriteRealParameter

Mit dem Funktionsbaustein "MB_WriteRealParameter" kann ein definierter Wert in den zur Parameternummer gehörenden Parameter geschrieben werden. Das Datenformat des Parameters, welcher beschrieben werden soll, muss DEC_OV, DEC_MV oder FLOAT sein; nicht beschreibbar sind Parameter mit dem Datenformat BIN, HEX und IDN.



Abhängig vom Parameter "S-0-0269, Speicherungsmodus" wird der geänderte Parameterwert nach jeder Änderung sofort oder erst durch die Ausführung des Kommandos "S-0-0264, C2200 Kommando Arbeitsspeicher sichern" im geräteinternen, nicht flüchtigen Speicher ("Flash") gespeichert.

HINWEIS

Zerstörung des Parameter-Festwertspeichers durch ständiges Aktivieren/Ausführen des Funktionsbausteins !

Bei häufiger Ausführung des Funktionsbausteins "S-0-0269, Speicherungsmodus" so einstellen, dass geänderte Parameterwerte nur durch die Ausführung des Kommandos "S-0-0264, C2200 Kommando Arbeitsspeicher sichern" im geräteinternen, nicht flüchtigen Speicher ("Flash") gespeichert werden.



Listen- und String-Parameter können mit dem Funktionsbaustein "MB_WriteRealParameter" nicht beschrieben werden; für das Schreiben von Listen-Parametern verwenden Sie bitte den Funktionsbaustein "MB_WriteListParameter" oder "MB_WriteRealListParameter".

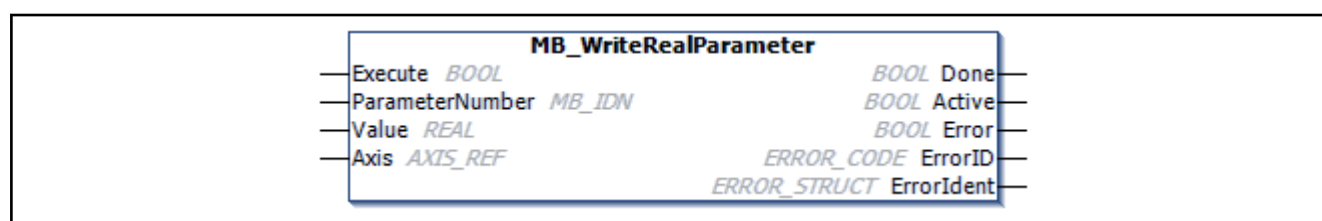


Abb. 13-67: Firmware-Funktionsbaustein "MB_WriteRealParameter"

MX_PLCOpen

Eingangs-Variable	Datentyp	Beschreibung
Execute	BOOL	siehe " Signallaufverhalten "
ParameterNumber	MB_IDN	An den Eingang "ParameterNumber" muss die Identnummer des zu beschreibenden Listen-Parameters angelegt werden
Value	REAL	An den Eingang "Value" muss der Wert gelegt werden, mit dem der Parameter beschrieben werden soll. Die Wichtung des ausgewählten Parameters ist dabei zu beachten

Tab. 13-163: *Eingangs-Variablen des Firmware-Funktionsbausteins "MB_WriteRealParameter"*

Ausgangs-Variable	Datentyp	Beschreibung
Done	BOOL	Mit "Done"=TRUE zeigt der Funktionsbaustein an, dass die Bearbeitung der Arbeitsanweisung erfolgreich abgeschlossen ist
Active	BOOL	Bei entfernten Achsen wird mit "Active"=TRUE angezeigt, dass der Funktionsbaustein die Bearbeitungsanweisung angenommen hat, bei Einzelachsen wird dieser Ausgang nicht gesetzt!
Error	BOOL	Mit "Error"=TRUE meldet der Funktionsbaustein, dass ein Fehler vorliegt
ErrorID *	ERROR_CODE	"ErrorID" gibt eine grobe Fehlerbeschreibung wieder.
ErrorIdent *	ERROR_STRUCT	"ErrorIdent" gibt über die Elemente: "Table", "Additional1" und "Additional2" die genaue Fehlerursache wieder.

* siehe nachfolgender Hinweis

Tab. 13-164: *Ausgangs-Variablen des Firmware-Funktionsbausteins "MB_WriteRealParameter"*



HINWEIS: Eine Auflistung aller möglichen Fehler finden Sie hier: "[Fehler-Referenzlisten](#)"

Eingangs-/Ausgangs-Variable (VAR_IN_OUT)	Datentyp	Beschreibung
Axis	AXIS_REF	Definiert die Achse. Folgende Achsen sind möglich: MPB (IndraDrive) und HDC (HydraulicDrive): Axis1 MPC (IndraDrive): - Axis1 (Master) oder - Axis2...Axis10 (insgesamt 9 Slaves / entfernte Achsen) Die in der Liste "P-0-1801.0.10, CCD: Adressen projektierter Antriebe" vorhandenen entfernten Achsen entsprechen der Reihenfolge nach Axis2 bis Axis10 (Axis2 wirkt auf den Antrieb des Listenelements 0 des P-0-1801.0.10,...) Bezüglich Zugriff auf Parameter von entfernten Achsen bitte nachfolgende Hinweise beachten!

Tab. 13-165: *Eingangs-/Ausgangs-Variable des Firmware-Funktionsbausteins "MB_WriteRealParameter"*



Bei einer entfernten Achse wird der Auftrag an den Servicekanal (Sercos III) abgegeben. Erst bei einem späteren Aufruf kann der Baustein "Done"="TRUE" melden. Damit ergeben sich für die Funktionsbaustein-Aufrufe selbst immer kurze Zeiten. Die Ausgänge "Done", "Active" und "Error" müssen hier zyklisch gepollt werden. Die Übertragung der Daten kann abhängig von der Belastung des Servicekanal auch Sekunden in Anspruch nehmen.

Solange der Sercos Ring noch nicht in Phase2 ist, können keine Daten mit entfernten Achsen ausgetauscht werden. Der Baustein meldet in diesem Fall einen Fehler.



Der Zugriff auf Parameter von entfernten Achsen ist im MLD-M-Systemmodus, im CCD-Basismodus und im CCD-Systemmodus möglich.

13.4.19 MB_WriteStringParameter

Mit Hilfe des Funktionsbausteins "MB_WriteStringParameter" kann ein Parameter geschrieben werden, dessen Betriebsdatum ASCII-Format hat (String).

An dem Eingang "ParameterNumber" muss die Parameternummer des Parameters angelegt werden, dessen Betriebsdatum geschrieben werden soll. Mit einer steigenden Flanke an dem Eingang "Execute" wird die Bearbeitung des Funktionsbausteins gestartet.



Abhängig vom Parameter "S-0-0269, Speicherungsmodus" wird der geänderte Parameterwert nach jeder Änderung sofort oder erst durch die Ausführung des Kommandos "S-0-0264, C2200 Kommando Arbeitsspeicher sichern" im geräteinternen, nicht flüchtigen Speicher ("Flash") gespeichert.

HINWEIS

Zerstörung des Parameter-Festwertspeichers durch ständiges Aktivieren/Ausführen des Funktionsbausteins !

Bei häufiger Ausführung des Funktionsbausteins "S-0-0269, Speicherungsmodus" so einstellen, dass geänderte Parameterwerte nur durch die Ausführung des Kommandos "S-0-0264, C2200 Kommando Arbeitsspeicher sichern" im geräteinternen, nicht flüchtigen Speicher ("Flash") gespeichert werden.



Zum Testen des Funktionsbausteins kann die Instanzvariable "ActivateCounter" herangezogen werden. Sie zählt die positiven Flanken des "Execute"-Signals und zeigt damit an, ob der Funktionsbaustein mit Flanke durchlaufen wird.

Lokale Achse

Wird der Funktionsbaustein bei einer lokalen Achse verwendet, so werden bei seinem Aufruf sofort alle Zeichen des ausgewählten ASCII-Parameters geschrieben. Danach kehrt die Kontrolle an das Programm zurück und am Ausgang "Done" liegt TRUE an.

MX_PLCOpen

Entfernte Achse



Der Schreibvorgang kann in Abhängigkeit von der Textlänge einige Zeit in Anspruch nehmen. Falls notwendig sollte eine eigene Task benutzt werden.



Der Zugriff auf Parameter von entfernten Achsen ist im MLD-M-Systemmodus, im CCD-Basismodus und im CCD-Systemmodus möglich.

Wird der Funktionsbaustein bei entfernten Achse verwendet, wird der Schreibauftrag an den Servicekanal (Sercos III) abgegeben. Erst bei einem späteren Aufruf kann der Baustein "Done" melden.

Die Ausgänge "Done", "Active" und "Error" müssen hier zyklisch gepollt werden.

Die Übertragung der Daten kann abhängig von der Textlänge einige Millisekunden in Anspruch nehmen.



Solange der Sercos Ring noch nicht in Phase2 ist, können keine Daten mit entfernten Achsen ausgetauscht werden. Der Baustein meldet in diesem Fall einen Fehler.

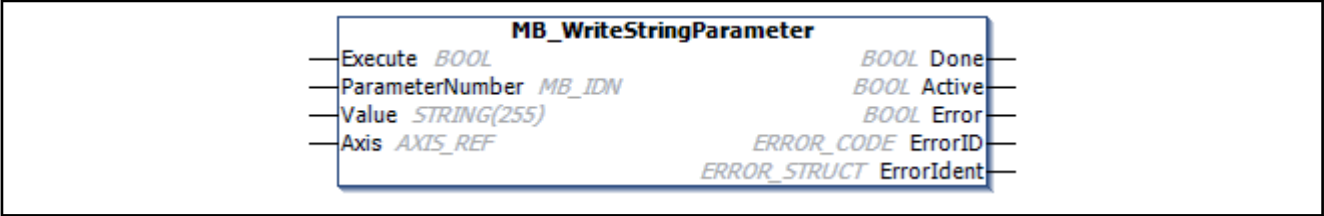


Abb. 13-68: Firmware-Funktionsbaustein "MB_WriteStringParameter"

Eingangs-Variable	Datentyp	Beschreibung
Execute	BOOL	siehe "Signallaufverhalten"
ParameterNumber	DINT	An dem Eingang "ParameterNumber" muss die Parameternummer des Parameters angelegt werden, dessen Betriebsdatum geschrieben werden soll

Tab. 13-166: Eingangs-Variablen des Firmware-Funktionsbausteins "MB_WriteStringParameter"

Ausgangs-Variable	Datentyp	Beschreibung
Done	BOOL	Mit "Done"=TRUE zeigt der Funktionsbaustein an, dass die Bearbeitung der Arbeitsanweisung erfolgreich abgeschlossen ist
Active	BOOL	Bei entfernten Achsen wird mit "Active"=TRUE angezeigt, dass der Funktionsbaustein die Bearbeitungsanweisung angenommen hat, bei Einzelachsen wird dieser Ausgang nicht gesetzt!
Error	BOOL	Mit "Error"=TRUE meldet der Funktionsbaustein, dass ein Fehler vorliegt
ErrorID *	ERROR_CODE	"ErrorID" gibt eine grobe Fehlerbeschreibung wieder.
ErrorIdent *	ERROR_STRUCT	"ErrorIdent" gibt über die Elemente: "Table", "Additional1" und "Additional2" die genaue Fehlerursache wieder.
Value	STRING(255)	"Value" liefert das Betriebsdatum des ausgewählten Parameters

* siehe nachfolgender Hinweis
Tab. 13-167: Ausgangs-Variablen des Firmware-Funktionsbausteins "MB_WriteStringParameter"



HINWEIS: Eine Auflistung aller möglichen Fehler finden Sie hier:
["Fehler-Referenzlisten"](#)

Eingangs-/Ausgangs-Variable (VAR_IN_OUT)	Datentyp	Beschreibung
Axis	AXIS_REF	Definiert die Achse. Folgende Achsen sind möglich: MPB (IndraDrive) und HDC (HydraulicDrive): Axis1 MPC (IndraDrive): - Axis1 (Master) oder - Axis2...Axis10 (insgesamt 9 Slaves / entfernte Achsen) Die in der Liste "P-0-1801.0.10, CCD: Adressen projektierter Antriebe" vorhandenen entfernten Achsen entsprechen der Reihenfolge nach Axis2 bis Axis10 (Axis2 wirkt auf den Antrieb des Listenelements 0 des P-0-1801.0.10,...)

Tab. 13-168: Eingangs-/Ausgangs-Variable des Firmware-Funktionsbausteins "MB_WriteStringParameter"

13.4.20 MX_Command

Der Funktionsbaustein "MX_Command" dient dazu ein Antriebskommando auszuführen.

Mit dem Eingang "Enable" wird das Kommando aktiviert bzw. deaktiviert. Liegt am Eingang TRUE an, so wird das Kommando durch Schreiben einer "3" (dez.) gestartet. Bei FALSE wird das Kommando durch Schreiben einer "0" abgebrochen/beendet.

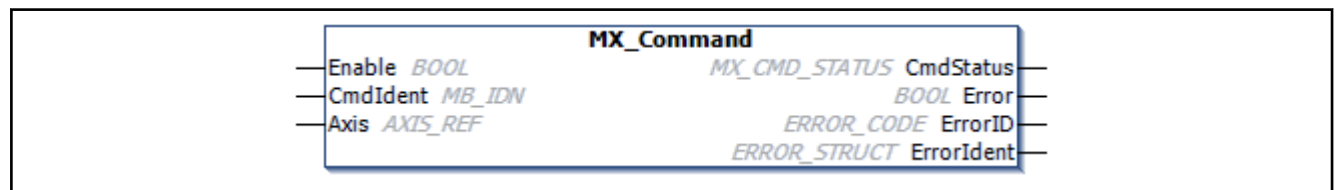


Abb. 13-69: Firmware-Funktionsbaustein "MX_Command"

MX_PLCOpen

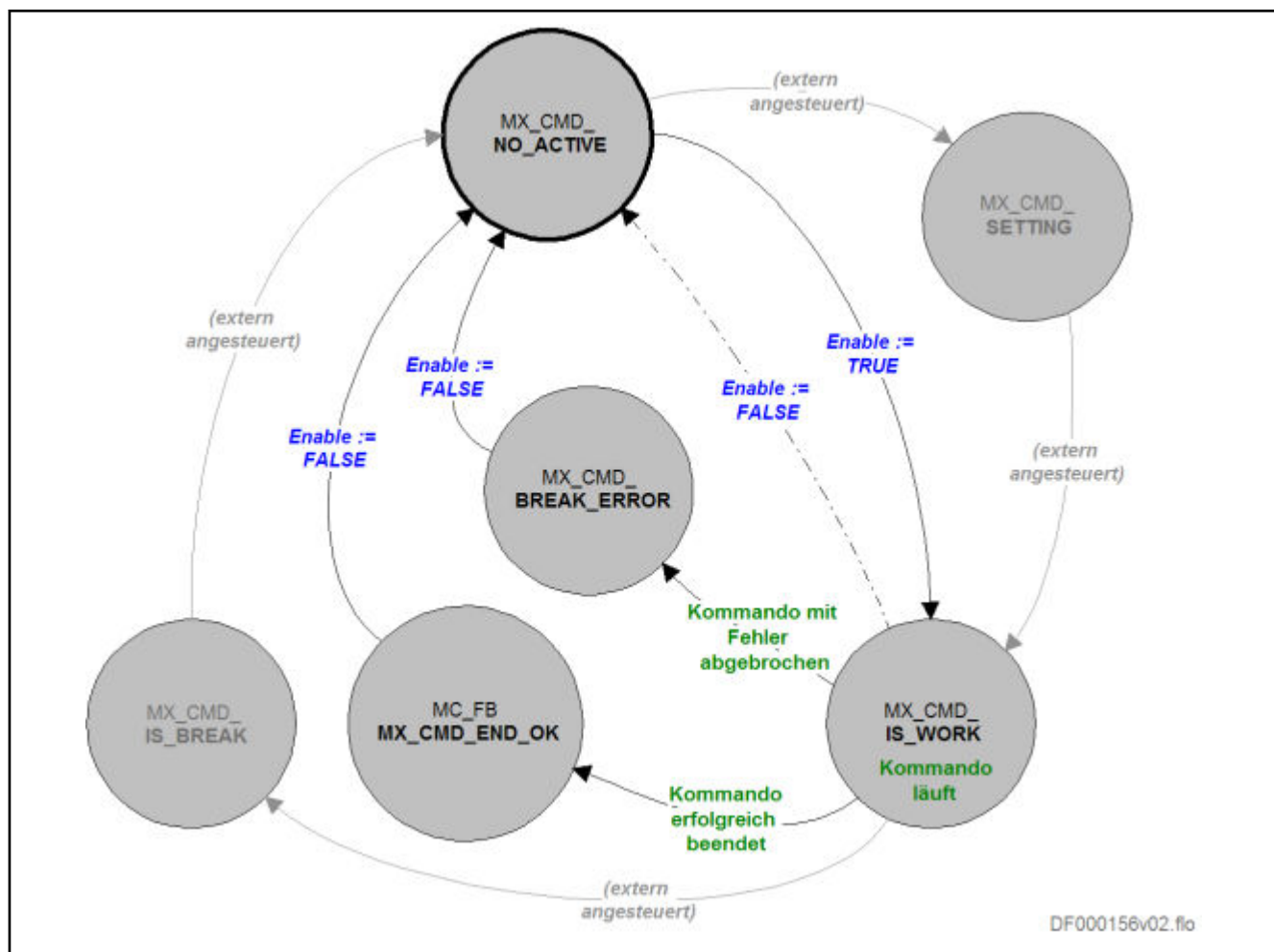


Abb. 13-70: Ausführung eines Antriebskommandos



Bei einer entfernten Achse wird der Auftrag an den Servicekanal (Sercos III) abgegeben. Erst bei einem späteren Aufruf kann "CmdStatus" "MX_CMD_END_OK" melden. Damit ergeben sich für die Funktionsbaustein-Aufrufe selbst immer kurze Zeiten. Die Ausgänge "CmdStatus" und "Error" müssen hier zyklisch gepollt werden. Die Übertragung der Daten kann abhängig von der Belastung des Servicekanal auch Sekunden in Anspruch nehmen.

Solange der Sercos Ring noch nicht in Phase2 ist, können keine Daten mit entfernten Achsen ausgetauscht werden. Der Baustein meldet in diesem Fall einen Fehler.

Eingangs-Variable	Datentyp	Beschreibung
Enable	BOOL	Der Funktionsbaustein ist zustandsgesteuert. Mit "Enable"=TRUE wird das Kommando gestartet, mit "Enable"=FALSE wird es gelöscht. Nach Setzen des Eingangs auf TRUE wird in der Regel gewartet ,bis der Ausgang "CmdStatus"=MX_CMD_END_OK oder "CmdStatus"=MX_CMD_BREAK_ERROR anzeigt. Nach dem Rücksetzen des Eingangs sollte darauf gewartet werden, bis der Ausgang "CmdStatus"=MX_CMD_NO_ACTIVE anzeigt. (Siehe auch Kommandobeschreibung und "Signallaufverhalten")
CmdIdent	MB_IDN	Identnummer des Kommandos

Tab. 13-169: Eingangs-Variablen des Firmware-Funktionsbausteins "MX_Command"

Ausgangs-Variable	Datentyp	Beschreibung
CmdStatus	MX_CMD_STATUS	Der Ausgang zeigt den aktuellen Status des Kommandos an. Enum-Werte für Ausgang "CmdStatus": "MX_CMD_NO_ACTIVE": Kommando ist nicht gesetzt "MX_CMD_SETTING": Kommando ist gesetzt "MX_CMD_END_OK": Kommando fehlerfrei beendet "MX_CMD_IS_BREAK": Kommando unterbrochen "MX_CMD_IS_WORK": Kommando in Bearbeitung "MX_CMD_BREAK_ERROR": Kommando mit Fehler abgebrochen
Error	BOOL	Mit "Error"=TRUE meldet der Funktionsbaustein, dass ein Fehler vorliegt
ErrorID *	ERROR_CODE	"ErrorID" gibt eine grobe Fehlerbeschreibung wieder.
ErrorIdent *	ERROR_STRUCT	"ErrorIdent" gibt über die Elemente: "Table", "Additional1" und "Additional2" die genaue Fehlerursache wieder.

* siehe nachfolgender Hinweis

Tab. 13-170: Ausgangs-Variablen des Firmware-Funktionsbausteins "MX_Command"



HINWEIS: Eine Auflistung aller möglichen Fehler finden Sie hier: "[Fehler-Referenzlisten](#)"

MX_PLCOpen

Eingangs-/Ausgangs-Variable (VAR_IN_OUT)	Datentyp	Beschreibung
Axis	AXIS_REF	Definiert die Achse. Folgende Achsen sind möglich: MPB (IndraDrive) und HDC (HydraulicDrive): Axis1 MPC (IndraDrive): - Axis1 (Master) oder - Axis2...Axis10 (insgesamt 9 Slaves / entfernte Achsen) Die in der Liste "P-0-1801.0.10, CCD: Adressen projektierter Antriebe" vorhandenen entfernten Achsen entsprechen der Reihenfolge nach Axis2 bis Axis10 (Axis2 wirkt auf den Antrieb des Listenelements 0 des P-0-1801.0.10,...) Hinweis: Der Eingang wird nur bei steigender Flanke des Enable-Eingangs übernommen. Sollen mehrere Achsen angesteuert werden, dann sollte für jede Achse eine separate Instanz verwendet werden.

Tab. 13-171: Eingangs-/Ausgangs-Variable des Firmware-Funktionsbausteins "MX_Command"

13.4.21 MX_fReadParamDINT

Die Firmware-Funktion "MX_fReadParamDINT" dient dazu, einen Parameter-Wert zu lesen und als DINT zurückzuliefern.

Mit dieser Funktion ist zunächst nicht erkennbar, ob das Lesen erfolgreich war. Hierzu eignet sich der Funktionsbaustein "MX_ReadParamDINT".

Bei Parametern mit 32 Bit ohne Vorzeichen kann der Rückgabewert einfach mit der Funktion "DINT_TO_UDINT" gewandelt werden.



Abb. 13-71: Firmware-Funktion "MX_fReadParamDINT"

Eingangs-Variable	Datentyp	Beschreibung
Ident	MB_IDN	An dem Eingang "Ident" ist die Identnummer des Parameters anzulegen, dessen Wert gelesen werden soll
ListElem	INT	Es kann der Wert eines Einzelparameters, aber auch die Werte von Listenparametern können gelesen werden. Zum Lesen eines Einzelparameters ist an "ListElem" "0" anzulegen. Bei Listenparameter wird mit "0" das nullte Element, mit "1" das erste Element, mit "2" das zweite Element, ... gelesen. Mit "-1" an "ListElem" wird die Istlänge, mit "-2" die Maximallänge eines Listenparametern gelesen
Axis	AXIS_REF	Achsnummer (hier kann nur die globale Variable "Axis1" übergeben werden)

Tab. 13-172: Eingangs-Variablen der Firmware-Funktion "MX_fReadParamDINT"

Rückgabewert	Datentyp	Beschreibung
MX_fReadParamDINT	DINT	Liefert den Wert des mit "Ident" und "ListElem" spezifizierten Elements. Hinweis: Falls der Parameter bzw. das Listenelement nicht gelesen werden kann, wird "0" zurückgeliefert

Tab. 13-173: Rückgabewert der Firmware-Funktion "MX_fReadParamDINT"

13.4.22 MX_fReadStringParam

Mit Hilfe der Firmware-Funktion "MX_fReadStringParam" kann ein Parameter gelesen werden, dessen Betriebsdatum ASCII-Format hat (String).



Zusätzlich zu "MaxLen" wird ein abschließendes 0-Zeichen kopiert. Der Returnwert gibt die tatsächlich gelesenen Zeichen (ohne 0) zurück.

Beim Aufruf der Funktion wird sofort der komplette Text gelesen. Danach kehrt die Kontrolle an das Programm zurück.



Der Lesevorgang kann in Abhängigkeit von der Textlänge einige Zeit in Anspruch nehmen. Falls notwendig sollte eine eigene Task benutzt werden.

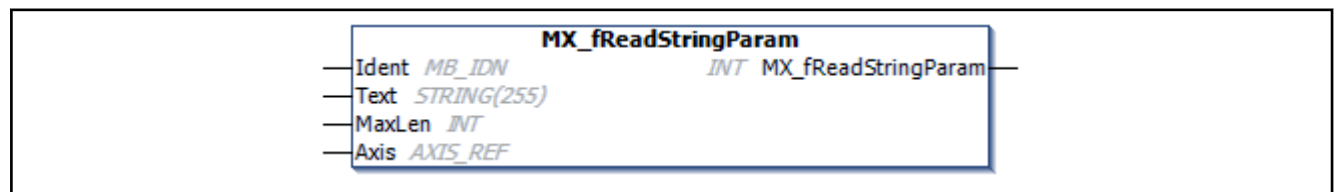


Abb. 13-72: Firmware-Funktion "MX_fReadStringParam"

Eingangs-Variable	Datentyp	Beschreibung
Ident	MB_IDN	Identnummer des Parameters
Text	STRING(255)	Textbuffer 255 Zeichen
MaxLen	INT	Anzahl der zu lesenden Zeichen (ohne abschließende 0)
Axis	AXIS_REF	Achsennummer (hier kann nur die globale Variable "Axis1" übergeben werden)

Tab. 13-174: Eingangs-Variablen der Firmware-Funktion "MX_fReadStringParam"

Rückgabewert	Datentyp	Beschreibung
MX_fReadStringParam	INT	Gibt die tatsächlich gelesenen Zeichen (ohne 0) zurück

Tab. 13-175: Rückgabewert der Firmware-Funktion "MX_fReadStringParam"

13.4.23 MX_fSetParamLimits

Mit der Firmware-Funktion "MX_fSetParamLimits" können die Grenzwerte der SPS-Globalen-Register-Parameter gesetzt werden.



Durch Ausschalten oder durch "SPS-Urladen" [C0730 Kommando Urladen (SPS Urladen)] werden die Grenzwerte wieder auf ihre Defaultwerte gesetzt.

MX_PLCoPen



Diese Funktion darf nicht aus dem Systemereignis "PositionLoop" aufgerufen werden. Wenn die Funktion dort aufgerufen wird, kommt es zu einer SPS-Exception und die Funktion liefert den Wert "0" zurück.



Abb. 13-73: Firmware-Funktion "MX_fSetParamLimits"

Eingangs-Variable	Datentyp	Beschreibung
Ident	MB_IDN	Hier kann eine Konstante aus den globalen Variablen übergeben werden
MinValue	DINT	Zu setzender Minimalwert
MaxValue	DINT	Zu setzender Maximalwert

Tab. 13-176: Eingangs-Variablen der Firmware-Funktion "MX_fSetParamLimits"

Rückgabewert	Datentyp	Beschreibung
MX_fSetParamLimits	UINT	Die Funktion kann die folgenden Werte zurück geben: 0: Grenzwerte wurden erfolgreich gesetzt 1: Fehler, unzulässige IDN wurde angegeben

Tab. 13-177: Rückgabewert der Firmware-Funktion "MX_fSetParamLimits"

13.4.24 MX_fSetParamName

Mit der Firmware-Funktion "MX_fSetParamName" können die Namen der SPS-Globalen-Register-Parameter und der SPS-I/O-Parameter gesetzt werden.



Durch Ausschalten werden die Namen der SPS-Globalen-Register-Parameter und der SPS-I/O-Parameter wieder auf ihre Defaultwerte gesetzt.

Wurden die Namen der SPS-Globalen-Register-Parameter oder der SPS-I/O-Parameter verändert, werden Sie durch "SPS-Urladen" [C0730 Kommando Urladen (SPS Urladen)] auf den jeweiligen englischen Namen gesetzt. Dieses Verhalten ist technisch bedingt.



Diese Funktion darf nicht aus dem Systemereignis "PositionLoop" aufgerufen werden. Wenn die Funktion dort aufgerufen wird, kommt es zu einer SPS-Exception und die Funktion liefert den Wert "0" zurück.



Abb. 13-74: Firmware-Funktion "MX_fSetParamName"

Eingangs-Variable	Datentyp	Beschreibung
Ident	MB_IDN	Hier kann eine Konstante aus den globalen Variablen übergeben werden
NewName	STRING (60)	Zu setzender Name (maximale Länge: 60 Zeichen)

Tab. 13-178: Eingangs-Variablen der Firmware-Funktion "MX_fSetParamName"

Rückgabewert	Datentyp	Beschreibung
MX_fSetParamName	UINT	Die Funktion kann die folgenden Werte zurück geben: 0: Name wurde erfolgreich gesetzt 1: Fehler, unzulässige IDN wurde angegeben

Tab. 13-179: Rückgabewert der Firmware-Funktion "MX_fSetParamName"

13.4.25 MX_fSetParamUnit

Mit der Firmware-Funktion "MX_fSetParamUnit" können die Einheiten der SPS-Globalen-Register-Parameter gesetzt werden.



Durch Ausschalten oder durch "SPS-Urladen" [C0730 Kommando Urladen (SPS Urladen)] werden die Einheiten wieder auf ihre Defaultwerte gesetzt.



Diese Funktion darf nicht aus dem Systemereignis "PositionLoop" aufgerufen werden. Wenn die Funktion dort aufgerufen wird, kommt es zu einer SPS-Exception und die Funktion liefert den Wert "0" zurück.



Abb. 13-75: Firmware-Funktion "MX_fSetParamUnit"

Eingangs-Variable	Datentyp	Beschreibung
Ident	MB_IDN	Hier kann eine Konstante aus den globalen Variablen übergeben werden
NewUnit	STRING (20)	Zu setzende Einheit (maximale Länge: 20 Zeichen)

Tab. 13-180: Eingangs-Variablen der Firmware-Funktion "MX_fSetParamUnit"

Rückgabewert	Datentyp	Beschreibung
MX_fSetParamUnit	UINT	Die Funktion kann die folgenden Werte zurück geben: 0: Einheit wurde erfolgreich gesetzt 1: Fehler, unzulässige IDN wurde angegeben

Tab. 13-181: Rückgabewert der Firmware-Funktion "MX_fSetParamUnit"

13.4.26 MX_fWriteParamDINT

Die Firmware-Funktion "MX_fWriteParamDINT" dient dazu, einen Parameter zu beschreiben.

Es kann ungepuffert oder gepuffert geschrieben werden.



Beim gepufferten Schreiben werden die Parameterwerte jedes Mal in den nichtflüchtigen Speicher transferiert. Da dies viel Rechenzeit verbraucht, sollte nur gepuffert geschrieben werden, wenn es notwendig ist.

Bei Listenparameter sollte nur der letzte Schreibzugriff mit "WriteBuffered"=TRUE erfolgen, da bei jedem Zugriff die gesamte Liste gepuffert wird.

Während der Zeit, in der ein Parameter gepuffert wird, schlägt ein weiterer Zugriff mit Pufferung fehl (ACCESS_ERROR).

HINWEIS

Zerstörung des geräteinternen, nicht flüchtigen Speichers ("Flash") durch ständiges gepuffertes Schreiben der Parameterwerte!

Bei häufiger Ausführung des Funktionsbausteins nur dann Parameterwerte gepuffert schreiben ("WriteBuffered"=TRUE), wenn es notwendig ist. Bei Listenparametern sollte z.B. nur der letzte Schreibzugriff mit "WriteBuffered"=TRUE erfolgen.

Mit dieser Funktion ist zunächst nicht erkennbar, ob das Schreiben erfolgreich war. Hierzu eignet sich der Funktionsbaustein "MX_WriteParamDINT".

Bei Parametern mit 32 Bit ohne Vorzeichen kann der Eingangswert "Value" mit der Funktion "UDINT_TO_DINT" gewandelt werden.

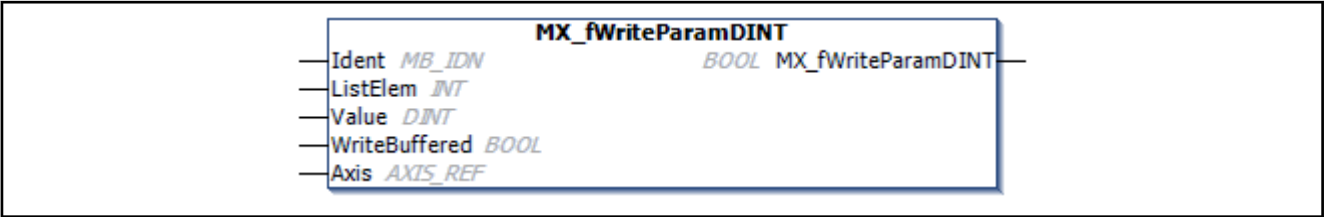


Abb. 13-76: Firmware-Funktion "MX_fWriteParamDINT"

Eingangs-Variable	Datentyp	Beschreibung
Ident	MB_IDN	Identnummer des spezifizierten Parameters
ListElem	INT	Listenelementnummer: bei Listenparameter ab "0" und bei Einzelparameter immer "0" "-1" zum Schreiben der Istlänge (nur bei Listenparametern erlaubt)
Value	INT	Wert mit dem das ausgewählte Element beschrieben werden soll
WriteBuffered	BOOL	"WriteBuffered"=TRUE: Wert wird gepuffert; "WriteBuffered"=FALSE: Wert wird nicht gepuffert
Axis	AXIS_REF	Achsnummer (hier kann nur die globale Variable "Axis1" übergeben werden)

Tab. 13-182: Eingangs-Variablen der Firmware-Funktion "MX_fWriteParamDINT"

Rückgabewert	Datentyp	Beschreibung
MX_fWriteParamDINT	BOOL	Dummy-Ausgang ohne Bedeutung

Tab. 13-183: Rückgabewert der Firmware-Funktion "MX_fWriteParamDINT"

13.4.27 MX_fWriteStringParam

Die Firmware-Funktion "MX_fWriteStringParam" dient dazu, einen Text-Parameter zu beschreiben.

Es kann ungepuffert oder gepuffert geschrieben werden.



Beim gepufferten Schreiben werden die Parameterwerte jedes Mal in den nichtflüchtigen Speicher transferiert. Da dies viel Rechenzeit verbraucht, sollte nur gepuffert geschrieben werden, wenn es notwendig ist.

Während der Zeit, in der ein Parameter gepuffert wird, schlägt ein weiterer Zugriff mit Pufferung fehl (ACCESS_ERROR).

Beim Aufruf der Funktion wird sofort der komplette Text geschrieben. Danach kehrt die Kontrolle an das Programm zurück.



Der Schreibvorgang kann in Abhängigkeit von der Textlänge einige Zeit in Anspruch nehmen. Falls notwendig sollte eine eigene Task benutzt werden.

HINWEIS

Zerstörung des geräteinternen, nicht flüchtigen Speichers ("Flash") durch ständiges gepuffertes Schreiben der Parameterwerte!

Bei häufiger Ausführung des Funktionsbausteins nur dann Parameterwerte gepuffert schreiben ("WriteBuffered"=TRUE), wenn es notwendig ist. Bei Listenparametern sollte z.B. nur der letzte Schreibzugriff mit "WriteBuffered"=TRUE erfolgen.

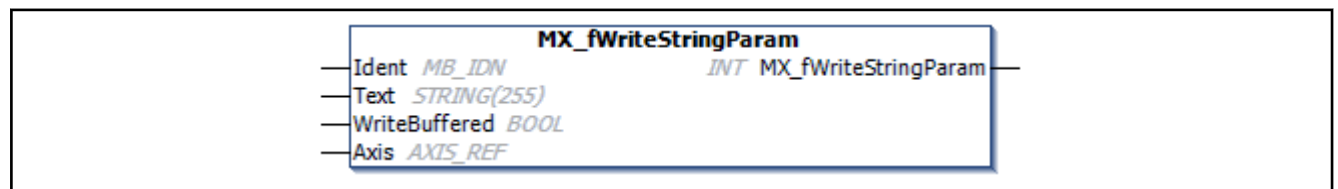


Abb. 13-77: Firmware-Funktion "MX_fWriteStringParam"

Eingangs-Variable	Datentyp	Beschreibung
Ident	MB_IDN	Identnummer des spezifizierten Parameters
Text	STRING(255)	Textstring
WriteBuffered	BOOL	"WriteBuffered"=TRUE: Wert wird gepuffert "WriteBuffered"=FALSE: Wert wird nicht gepuffert
Axis	AXIS_REF	Achsnummer (hier kann nur die globale Variable "Axis1" übergeben werden)

Tab. 13-184: Eingangs-Variablen der Firmware-Funktion "MX_fWriteStringParam"

Rückgabewert	Datentyp	Beschreibung
MX_fWriteStringParam	INT	Der Returnwert gibt die tatsächlich geschriebenen Zeichen (ohne 0) zurück

Tab. 13-185: Rückgabewert der Firmware-Funktion "MX_fWriteStringParam"

MX_PLCOpen

13.4.28 MX_ReadParamDINT

Der Funktionsbaustein "MX_ReadParamDINT" dient dazu, einen Parameter-Wert zu lesen und als DINT zurückgeliefert.

Der Funktionsbaustein gibt unter anderem Rückmeldung, ob das Lesen erfolgreich war.

Bei Parametern mit 32 Bit ohne Vorzeichen kann der Rückgabewert mit der Funktion "DINT_TO_UDINT" gewandelt werden.

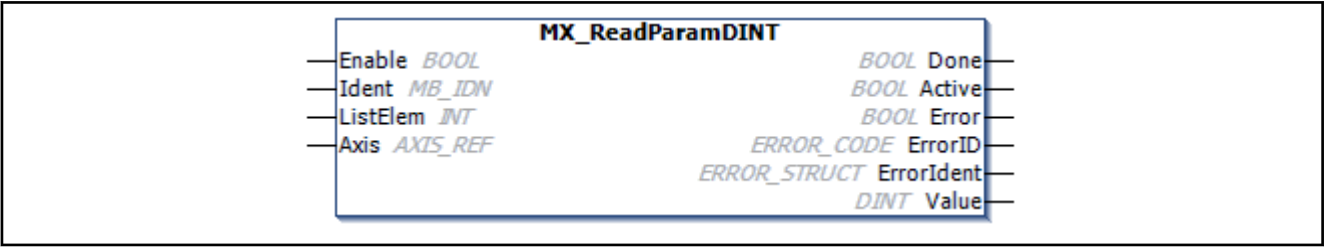


Abb. 13-78: Firmware-Funktionsbaustein "MX_ReadParamDINT"

Eingangs-Variable	Datentyp	Beschreibung
Enable	BOOL	Mit "Enable"=TRUE wird der Funktionsbaustein aktiviert. Alle Ausgänge werden mit "Enable"=FALSE zurückgesetzt
Ident	DINT	Identnummer des Parameters
ListElem	INT	Listenelementnummer: bei Listenparameter ab "0" und bei Einzelparameter immer "0" "-1" zum Lesen der Istlänge (nur bei Listenparametern erlaubt) "-2" zum Lesen der Maxlänge (nur bei Listenparametern erlaubt)

Tab. 13-186: Eingangs-Variablen des Firmware-Funktionsbausteins "MX_ReadParamDINT"

Ausgangs-Variable	Datentyp	Beschreibung
Done	BOOL	Mit "Done"=TRUE zeigt der Funktionsbaustein an, dass die Bearbeitung der Arbeitsanweisung erfolgreich abgeschlossen ist
Active	BOOL	Bei entfernten Achsen wird mit "Active"=TRUE angezeigt, dass der Funktionsbaustein die Bearbeitungsanweisung angenommen hat, bei Einzelachsen wird dieser Ausgang nicht gesetzt!
Error	BOOL	Mit "Error"=TRUE meldet der Funktionsbaustein, dass ein Fehler vorliegt
ErrorID *	ERROR_CODE	"ErrorID" gibt eine grobe Fehlerbeschreibung wieder.
ErrorIdent *	ERROR_STRUCT	"ErrorIdent" gibt über die Elemente: "Table", "Additional1" und "Additional2" die genaue Fehlerursache wieder.
Value	DINT	Ist der Ausgang "Done"=TRUE, zeigt "Value" den Wert des ausgewählten Elements an

* siehe nachfolgender Hinweis

Tab. 13-187: Ausgangs-Variablen des Firmware-Funktionsbausteins "MX_ReadParamDINT"

 HINWEIS: Eine Auflistung aller möglichen Fehler finden Sie hier: ["Fehler-Referenzlisten"](#)

Eingangs-/Ausgangs-Variable (VAR_IN_OUT)	Datentyp	Beschreibung
Axis	AXIS_REF	Achsnummer (hier kann nur die globale Variable "Axis1" übergeben werden)

Tab. 13-188: Eingangs-/Ausgangs-Variable des Firmware-Funktionsbausteins "MX_ReadParamDINT"

13.4.29 MX_WriteParamDINT

Der Firmware-Funktionsbaustein "MX_WriteParamDINT" dient dazu, einen Parameter zu beschreiben. Es kann ungepuffert oder gepuffert geschrieben werden.



Beim gepufferten Schreiben werden die Parameterwerte jedes Mal in den nichtflüchtigen Speicher transferiert. Da dies viel Rechenzeit verbraucht, sollte nur gepuffert geschrieben werden, wenn es notwendig ist.

Bei Listenparameter sollte nur der letzte Schreibzugriff mit "WriteBuffered"=TRUE erfolgen, da bei jedem Zugriff die gesamte Liste gepuffert wird.

Während der Zeit, in der ein Parameter gepuffert wird, schlägt ein weiterer Zugriff mit Pufferung fehl (ACCESS_ERROR).

HINWEIS

Zerstörung des geräteinternen, nicht flüchtigen Speichers ("Flash") durch ständiges gepuffertes Schreiben der Parameterwerte!

Bei häufiger Ausführung des Funktionsbausteins nur dann Parameterwerte gepuffert schreiben ("WriteBuffered"=TRUE), wenn es notwendig ist. Bei Listenparametern sollte z.B. nur der letzte Schreibzugriff mit "WriteBuffered"=TRUE erfolgen.

Bei Parametern mit 32 Bit ohne Vorzeichen kann der Eingangswert "Value" einfach mit der Funktion "UDINT_TO_DINT" gewandelt werden.

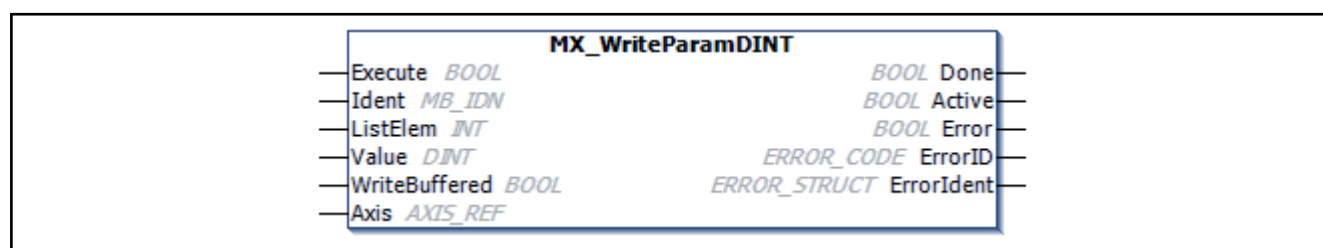


Abb. 13-79: Firmware-Funktionsbaustein "MX_WriteParamDINT"

Eingangs-Variable	Datentyp	Beschreibung
Execute	BOOL	siehe "Signallaufverhalten"
Ident	MB_IDN	Identnummer des Parameters

MX_PLCOpen

Eingangs-Variable	Datentyp	Beschreibung
ListElem	INT	Mit dem Eingang "ListElem" muss das Element, welches geschrieben werden soll, spezifiziert werden. Listenelementnummer: bei Listenparameter ab "0" und bei Einzelparameter immer "0". "-1" zum Schreiben der Istlänge (nur bei Listenparametern erlaubt)
Value	DINT	An "Value" muss der Wert anlegt werden, mit dem der Parameter beschrieben werden soll. Die Wichtung des ausgewählten Parameters ist dabei zu beachten. Je nach Wichtung besitzt der Parameter Nachkommastellen, die hier beachtet und verrechnet werden müssen
WriteBuffered	BOOL	Mit "WriteBuffered"=TRUE wird der Wert gepuffert geschrieben, falls der Parameter dafür ausgelegt ist. Mit FALSE wird der Parameter immer ungepuffert geschrieben

Tab. 13-189: Eingangs-Variablen des Firmware-Funktionsbausteins
"MX_WriteParamDINT"

Ausgangs-Variable	Datentyp	Beschreibung
Done	BOOL	Mit "Done"=TRUE zeigt der Funktionsbaustein an, dass die Bearbeitung der Arbeitsanweisung erfolgreich abgeschlossen ist
Active	BOOL	Bei entfernten Achsen wird mit "Active"=TRUE angezeigt, dass der Funktionsbaustein die Bearbeitungsanweisung angenommen hat, bei Einzelachsen wird dieser Ausgang nicht gesetzt!
Error	BOOL	Mit "Error"=TRUE meldet der Funktionsbaustein, dass ein Fehler vorliegt
ErrorID *	ERROR_CODE	"ErrorID" gibt eine grobe Fehlerbeschreibung wieder.
ErrorIdent *	ERROR_STRUCT	"ErrorIdent" gibt über die Elemente: "Table", "Additional1" und "Additional2" die genaue Fehlerursache wieder.

* siehe nachfolgender Hinweis

Tab. 13-190: Ausgangs-Variablen des Firmware-Funktionsbausteins
"MX_WriteParamDINT"



HINWEIS: Eine Auflistung aller möglichen Fehler finden Sie hier:
["Fehler-Referenzlisten"](#)

Eingangs-/Ausgangs-Variable (VAR_IN_OUT)	Datentyp	Beschreibung
Axis	AXIS_REF	Achsnummer (hier kann nur die globale Variable "Axis1" übergeben werden)

Tab. 13-191: Eingangs-/Ausgangs-Variable des Firmware-Funktionsbausteins
"MX_WriteParamDINT"

13.5 "Scaling" - Funktionen für Skalierung

13.5.1 MX_fDINT_AccTo_REAL

Die Firmware-Funktion "MX_fDINT_AccTo_REAL" wandelt einen Beschleunigungswert im Festkommaformat des Datentyps DINT gemäß eingestellter Wichtung in einen REAL-Wert um.

Falls die Achsreferenz ungültig ist, wird 0.0 zurückgeliefert.

Die Funktion kann für die virtuelle Leitachse / den Leitachsgeber und für die realen Achsen "Axis1", "Axis2", ... verwendet werden.



Abb. 13-80: Firmware-Funktion "MX_fDINT_AccTo_REAL"

Input-Variable	Datentyp	Beschreibung
Acceleration	DINT	Beschleunigungswert im DINT-Festkomma-Format
Axis	AXIS_REF	Achsadresse: hier kann die entsprechende Achse z. B. "Axis1" aus den globalen Achsvariablen "AxisAddressing" von "MX_Base.lib" übergeben werden

Tab. 13-192: Input-Variablen der Firmware-Funktion "MX_fDINT_AccTo_REAL"

Rückgabewert	Datentyp	Beschreibung
MX_fDINT_AccTo_REAL	REAL	Liefert einen Beschleunigungswert gemäß eingestellter Wichtung im REAL-Format zurück

Tab. 13-193: Rückgabewert der Firmware-Funktion "MX_fDINT_AccTo_REAL"

13.5.2 MX_fDINT_DistTo_REAL

Die Firmware-Funktion "MX_fDINT_DistTo_REAL" wandelt einen Lagewert im Festkommaformat des Datentyps DINT gemäß eingestellter Wichtung in einen REAL-Wert um.

Falls die Achsreferenz ungültig ist, wird 0.0 zurückgeliefert.

Die Funktion kann für die virtuelle Leitachse / den Leitachsgeber und für die realen Achsen "Axis1", "Axis2", ... verwendet werden.

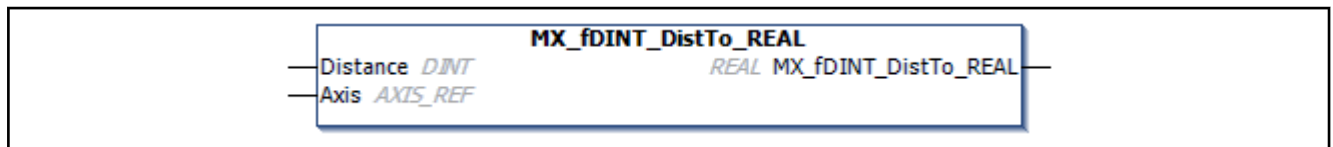


Abb. 13-81: Firmware-Funktion "MX_fDINT_DistTo_REAL"

Input-Variable	Datentyp	Beschreibung
Distance	DINT	Lagewert im DINT-Festkomma-Format
Axis	AXIS_REF	Achsadresse: hier kann die entsprechende Achse z. B. "Axis1" aus den globalen Achsvariablen "AxisAddressing" von "MX_Base.lib" übergeben werden

Tab. 13-194: Input-Variablen der Firmware-Funktion "MX_fDINT_DistTo_REAL"

Rückgabewert	Datentyp	Beschreibung
MX_fDINT_DistTo_REAL	REAL	Liefert einen Lagewert gemäß eingestellter Wichtung im REAL-Format zurück

Tab. 13-195: Rückgabewert der Firmware-Funktion "MX_fDINT_DistTo_REAL"

13.5.3 MX_fDINT_VelTo_REAL

Die Firmware-Funktion "MX_fDINT_VelTo_REAL" wandelt einen Geschwindigkeitswert im Festkommaformat des Datentyps DINT gemäß eingestellter Wichtung in einen REAL-Wert um.

MX_PLCoPen

Falls die Achsreferenz ungültig ist, wird 0 zurückgeliefert.

Die Funktion kann für die virtuelle Leitachse / den Leitachsgeber und für die realen Achsen "Axis1", "Axis2", ... verwendet werden.

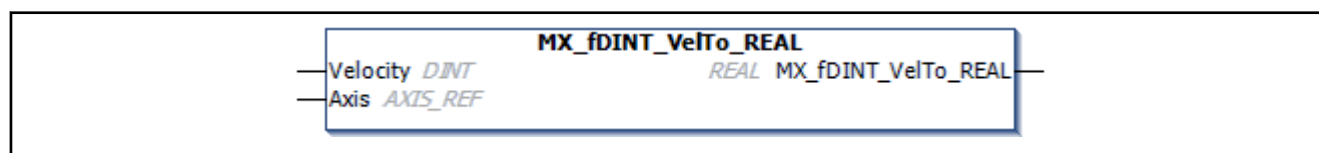


Abb. 13-82: Firmware-Funktion "MX_fDINT_VelTo_REAL"

Input-Variable	Datentyp	Beschreibung
Velocity	DINT	Geschwindigkeitswert im DINT-Festkomma-Format
Axis	AXIS_REF	Achsadresse: hier kann die entsprechende Achse z. B. "Axis1" aus den globalen Achsvariablen "AxisAddressing" von "MX_Base.lib" übergeben werden

Tab. 13-196: Input-Variablen der Firmware-Funktion "MX_fDINT_VelTo_REAL"

Rückgabewert	Datentyp	Beschreibung
MX_fDINT_VelTo_REAL	REAL	Liefert einen Geschwindigkeitswert gemäß eingestellter Wichtung im REAL-Format zurück

Tab. 13-197: Rückgabewert der Firmware-Funktion "MX_fDINT_VelTo_REAL"

13.5.4 MX_fREAL_AccTo_DINT

Die Firmware-Funktion "MX_fREAL_AccTo_DINT" wandelt einen Beschleunigungswert im REAL-Format gemäß eingestellter Wichtung in einen DINT-Festkomma-Wert um.

Falls die Achsreferenz ungültig ist, wird 0.0 zurückgeliefert.

Die Funktion kann für die virtuelle Leitachse / den Leitachsgeber und für die realen Achsen "Axis1", "Axis2", ... verwendet werden.

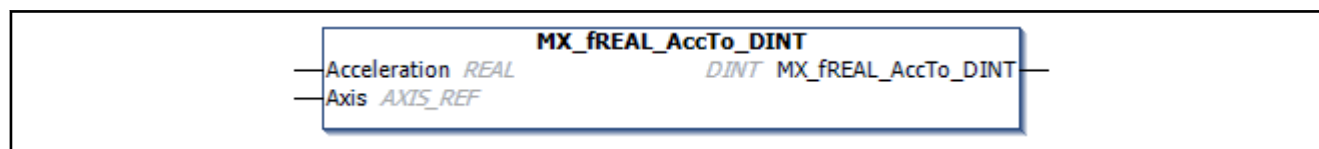


Abb. 13-83: Firmware-Funktion "MX_fREAL_AccTo_DINT"

Input-Variable	Datentyp	Beschreibung
Acceleration	REAL	Beschleunigungswert im REAL-Format
Axis	AXIS_REF	Achsadresse: hier kann die entsprechende Achse z. B. "Axis1" aus den globalen Achsvariablen "AxisAddressing" von "MX_Base.lib" übergeben werden

Tab. 13-198: Input-Variablen der Firmware-Funktion "MX_fREAL_AccTo_DINT"

Rückgabewert	Datentyp	Beschreibung
MX_fREAL_AccTo_DINT	DINT	Liefert einen Beschleunigungswert gemäß eingestellter Wichtung im DINT-Festkomma-Format zurück

Tab. 13-199: Rückgabewert der Firmware-Funktion "MX_fREAL_AccTo_DINT"

13.5.5 MX_fREAL_DistTo_DINT

Die Firmware-Funktion "MX_fREAL_DistTo_DINT" wandelt einen Lagewert im REAL-Format gemäß eingestellter Wichtung in einen DINT-Festkomma-Wert um.

Falls die Achsreferenz ungültig ist, wird 0 zurückgeliefert.

Die Funktion kann für die virtuelle Leitachse / den Leitachsgeber und für die realen Achsen "Axis1", "Axis2", ... verwendet werden.

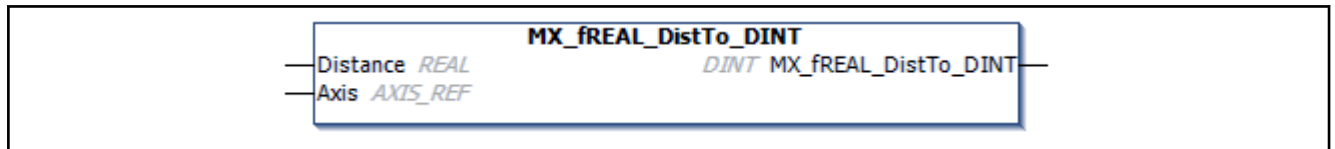


Abb. 13-84: Firmware-Funktion "MX_fREAL_DistTo_DINT"

Input-Variable	Datentyp	Beschreibung
Distance	REAL	Lagewert im REAL-Format
Axis	AXIS_REF	Achsadresse: hier kann die entsprechende Achse z. B. "Axis1" aus den globalen Achsvariablen "AxisAddressing" von "MX_Base.lib" übergeben werden

Tab. 13-200: Input-Variablen der Firmware-Funktion "MX_fREAL_DistTo_DINT"

Rückgabewert	Datentyp	Beschreibung
MX_fDINT_DistTo_REAL	DINT	Liefert einen Lagewert gemäß eingestellter Wichtung im DINT-Festkomma-Format zurück

Tab. 13-201: Rückgabewert der Firmware-Funktion "MX_fREAL_DistTo_DINT"

13.5.6 MX_fREAL_TorqueRampTo_DINT

Die Firmware-Funktion "MX_fREAL_TorqueRampTo_DINT" wandelt einen Drehmoment-/Krafttrampenwert im REAL-Format gemäß eingestellter Wichtung in einen DINT-Festkomma-Wert um.

Die Funktion kann für die realen Achsen "Axis1", "Axis2", ... verwendet werden.

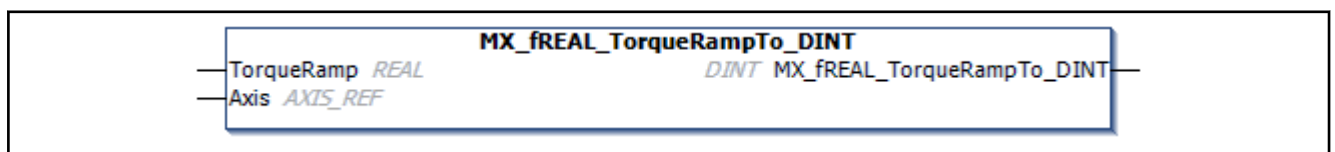


Abb. 13-85: Firmware-Funktion "MX_fREAL_TorqueRampTo_DINT"

Input-Variable	Datentyp	Beschreibung
TorqueRamp	REAL	Drehmoment-/Krafttrampenwert im REAL-Format
Axis	AXIS_REF	Achsadresse: hier kann die entsprechende Achse z. B. "Axis1" aus den globalen Achsvariablen "AxisAddressing" von "MX_Base.lib" übergeben werden

Tab. 13-202: Input-Variablen der Firmware-Funktion "MX_fREAL_TorqueRampTo_DINT"

MX_PLCOpen

Rückgabewert	Datentyp	Beschreibung
MX_fREAL_TorqueRampTo_DINT	DINT	Liefert einen Drehmoment-/Krafttrampenwert gemäß eingestellter Wichtung im DINT-Festkomma-Format zurück

Tab. 13-203: Rückgabewert der Firmware-Funktion "MX_fREAL_TorqueRampTo_DINT"

13.5.7 MX_fREAL_TorqueTo_DINT

Die Firmware-Funktion "MX_fREAL_TorqueTo_DINT" wandelt einen Drehmoment-/Kraftwert im REAL-Format gemäß eingestellter Wichtung in einen DINT-Festkomma-Wert um.

Die Funktion kann für die realen Achsen "Axis1", "Axis2", ... verwendet werden.

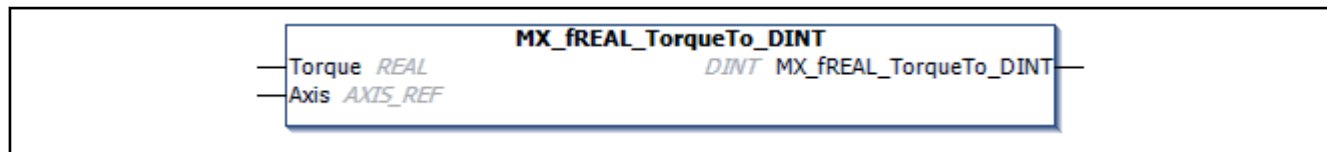


Abb. 13-86: Firmware-Funktion "MX_fREAL_TorqueTo_DINT"

Input-Variable	Datentyp	Beschreibung
Torque	REAL	Drehmoment-/Kraftwert im REAL-Format
Axis	AXIS_REF	Achsadresse: hier kann die entsprechende Achse z. B. "Axis1" aus den globalen Achsvariablen "AxisAddressing" von "MX_Base.lib" übergeben werden

Tab. 13-204: Input-Variablen der Firmware-Funktion "MX_fREAL_TorqueTo_DINT"

Rückgabewert	Datentyp	Beschreibung
MX_fREAL_TorqueTo_DINT	DINT	Liefert einen Drehmoment-/Kraftwert gemäß eingestellter Wichtung im DINT-Festkomma-Format zurück

Tab. 13-205: Rückgabewert der Firmware-Funktion "MX_fREAL_TorqueTo_DINT"

13.5.8 MX_fREAL_VelTo_DINT

Die Firmware-Funktion "MX_fREAL_VelTo_DINT" wandelt einen Geschwindigkeitswert im REAL-Format gemäß eingestellter Wichtung in einen DINT-Festkomma-Wert um.

Falls die Achsreferenz ungültig ist, wird 0 zurückgeliefert.

Die Funktion kann für die virtuelle Leitachse / den Leitachsgeber und für die realen Achsen "Axis1", "Axis2", ... verwendet werden.



Abb. 13-87: Firmware-Funktion "MX_fREAL_VelTo_DINT"

Input-Variable	Datentyp	Beschreibung
Velocity	REAL	Geschwindigkeitswert im REAL-Format
Axis	AXIS_REF	Achsadresse: hier kann die entsprechende Achse z. B. "Axis1" aus den globalen Achsvariablen "AxisAddressing" von "MX_Base.lib" übergeben werden

Tab. 13-206: Input-Variablen der Firmware-Funktion "MX_fREAL_VelTo_DINT"

Rückgabewert	Datentyp	Beschreibung
MX_fREAL_VelTo_DINT	DINT	Liefert einen Geschwindigkeitswert gemäß eingestellter Wichtung im DINT-Festkomma-Format zurück

Tab. 13-207: Rückgabewert der Firmware-Funktion "MX_fREAL_VelTo_DINT"

13.6 "Tools" - allgemeine Funktionen

13.6.1 MX_fGetFreeTicks

Die Firmware-Funktion "MX_fGetFreeTicks" dient dazu, die noch für die SPS verfügbare Zeit der letzten SPS-Zeitscheibe zu ermitteln. Diese SPS-Zeitscheibe ist der Teil einer Millisekunde, welcher nach Bearbeitung der Antriebsregler- und Steuertasks verbleibt.

Es werden Timerticks zurückgeliefert, die mit der Funktion "MX_fHighResTimerTicks_to_us" in Mikrosekunden umgewandelt werden können.



- Wenn eine freilaufende Task aktiv ist, wird hier stets "0" zurückgeliefert, da diese Task die gesamte Zeitscheibe arbeitet.
- Wenn "MX_fGetFreeTicks" in einer Task aufgerufen wird, die langsamer als 1 ms getaktet wird, so wird nur die verfügbare Zeit der letzten ms-Zeitscheibe angezeigt.
- Braucht eine zyklische Task für ihre Abarbeitung länger als die eingestellte Zykluszeit, liefert "MX_fGetFreeTicks" stets den Wert "0".



Abb. 13-88: Firmware-Funktion "MX_fGetFreeTicks"

Rückgabewert	Datentyp	Beschreibung
MX_fGetFreeTicks	DINT	Liefert Timerticks zurück. Die Timerticks können mit der Funktion "MX_fHighResTimerTicks_to_us" in µs umgewandelt werden

Tab. 13-208: Rückgabewert der Firmware-Funktion "MX_fGetFreeTicks"

13.6.2 MX_fGetHighResTime

Die Firmware-Funktion "MX_fGetHighResTime" liefert einen hochlaufenden Timerwert mit hoher Auflösung.

Es werden Timerticks zurückgeliefert, die mit der Funktion "MX_fHighResTimerTicks_to_us" in Mikrosekunden umgewandelt werden können.

MX_PLCOpen



Abb. 13-89: Firmware-Funktion "MX_fGetHighResTime"

Rückgabewert	Datentyp	Beschreibung
MX_fGetHighResTime	UDINT	Liefert Timerticks zurück. Die Timerticks können mit der Funktion "MX_fHighResTimerTicks_to_us" in µs umgewandelt werden

Tab. 13-209: Rückgabewert der Firmware-Funktion "MX_fGetHighResTime"

13.6.3 MX_fHighResTimerTicks_to_us

Die Firmware-Funktion "MX_fHighResTimerTicks_to_us" dient dazu, Timerticks in Mikrosekunden umzuwandeln und den Wert als REAL zurückzuliefern.



Abb. 13-90: Firmware-Funktion "MX_fHighResTimerTicks_to_us"

Eingangs-Variable	Datentyp	Beschreibung
HighResTimerTicks	UDINT	Übergabewerte sind Timerticks (siehe z. B. "MX_fGetHighResTime")

Tab. 13-210: Eingangs-Variable der Firmware-Funktion "MX_fHighResTimerTicks_to_us"

Rückgabewert	Datentyp	Beschreibung
MX_fHighResTimerTicks_to_us	REAL	Liefert die Timerticks in µs zurück

Tab. 13-211: Rückgabewert der Firmware-Funktion "MX_fHighResTimerTicks_to_us"

13.6.4 MX_IECTaskGetLoad

Der Funktionsbaustein "MX_IECTaskGetLoad" dient dazu die erweiterte Laufzeitmessung zu aktivieren und zeigt Informationen über die Taskbelastung an (siehe auch Anwendungsbeschreibung zu IndraMotion MLD, Kapitel "Laufzeitmessungen").

- 

Sobald eine Bausteininstanz im Projekt angelegt ist, wird die erweiterte Laufzeitmessung im Antrieb aktiviert.
- 

Der Funktionsbaustein "MX_IECTaskGetLoad" kann in der "eigenen" Task, aber auch in einer anderen (langsameren) Task aufgerufen werden, um Rechenzeit zu sparen. Die maximale Taskbelastung wird auch gemessen, wenn der Baustein nicht in jedem Zyklus aufgerufen wird.

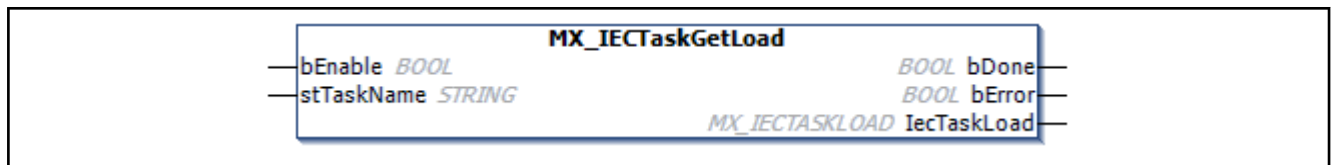


Abb. 13-91: Firmware-Funktionsbaustein "MX_IECTaskGetLoad"

Eingangs-Variable	Datentyp	Beschreibung
bEnable	BOOL	Bei "bEnable"=TRUE wird die Taskinformation der am Eingang "stTaskName" angegebenen Task zurückgeliefert.
stTaskName	STRING	An "stTaskName" muss der Name der Task angegeben werden, dessen Laufzeitinformationen angezeigt werden sollen

Tab. 13-212: Eingangs-Variablen des Firmware-Funktionsbausteins "MX_IECTaskGetLoad"

Ausgangs-Variable	Datentyp	Beschreibung
bDone	BOOL	"bDone" liefert TRUE zurück, wenn die Task existiert und die Ausgänge gültig sind
bError	BOOL	Mit "bError"=TRUE meldet der Funktionsbaustein, dass die Task nicht gefunden wurde (falscher Taskname)
IecTaskLoad	MX_IECTASKLOAD	Laufzeitinformationen der Task Elemente in "IecTaskLoad": "rLoad" zeigt die Auslastung der Task im letzten Zyklus in Prozent bezogen auf die Zykluszeit an "rLoadMax" zeigt die maximale Auslastung der Task in Prozent bezogen auf die Zykluszeit an. "rFreeTime" zeigt die ungenutzte verbleibende Rechenzeit der Task bis zu ihrem nächsten Start vom letzten Zyklus in µs an. "rMinFreeTime" zeigt die minimale ungenutzte verbleibende Rechenzeit der Task bis zu ihrem nächsten Start in µs an.

Tab. 13-213: Ausgangs-Variablen des Firmware-Funktionsbausteins "MX_IECTaskGetLoad"

13.6.5 MX_PositionLoopEventInfo



Die Nutzung des Funktionsbausteins "MX_PositionLoopEventInfo" ist Mitarbeitern von Bosch Rexroth vorbehalten.

14 Basisbibliotheken

14.1 MX_CommonTypes

14.1.1 Einleitung

Die Bibliothek "MX_CommonTypes" enthält allgemeine Datentypen und Strukturen. (Die "MX_CommonTypes" wurde für "IndraMotion MLD" von der "RIL_CommonTypes" abgeleitet.)

Nachfolgend werden nur die Datentypen und Strukturen beschrieben, die für den Anwender von Bedeutung sind.

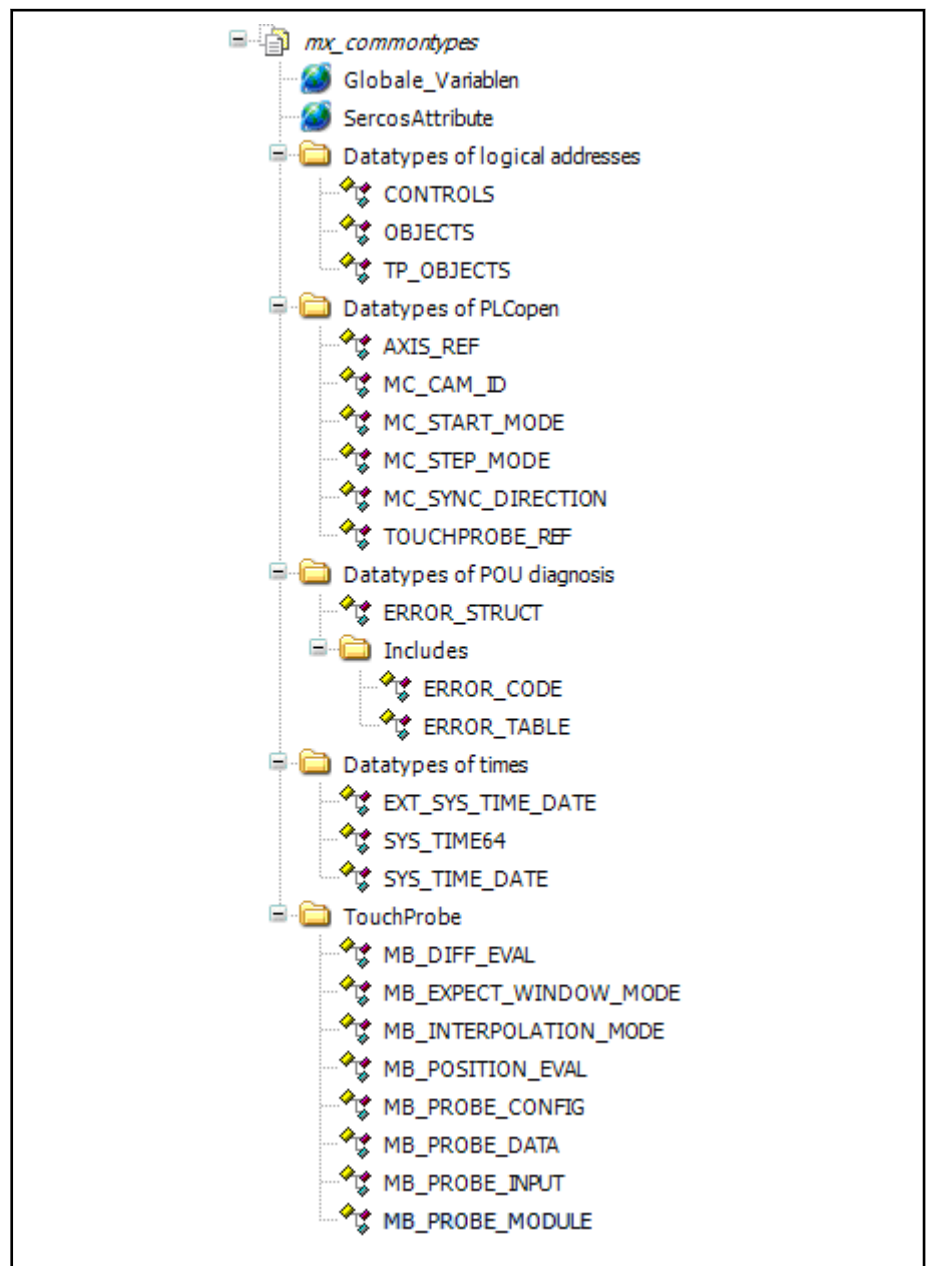


Abb. 14-1: Aufbau der Bibliothek "MX_CommonTypes"

Basisbibliotheken

14.1.2 Globale Variablen

IL_NULL

Kurzbeschreibung Die globale Variable "IL_NULL" ist ein "Nullzeiger" mit dem Datentyp "POINTER TO BYTE".

Zuordnung Zielsystem/Bibliothek

Zielsystem	Bibliothek / Ordner
IndraMotion MLD 2G	MX_CommonTypes / Globale_Variablen
Grundpaket PLC	

Tab. 14-1: Referenztable IL_NULL

Typdefinition

```
VAR_GLOBAL CONSTANT
  IL_NULL: POINTER TO BYTE := 0;
END_VAR
```

14.1.3 Sercos Attribute

Siehe "[Sercos Attribut](#)"

14.1.4 Datatypes of logical addresses

CONTROLS

Kurzbeschreibung Mit dem Datentyp "CONTROLS" kann die lokale Steuerung über IndraMotion-Funktionsbausteine adressiert werden. Die lokale Steuerung ist über zwei Möglichkeiten erreichbar: die Nummer "0" oder über ihre eigene Adresse "LOCAL_CTRL".

Zuordnung Zielsystem/Bibliothek

Zielsystem	Bibliothek / Ordner
IndraMotion MLD 2G	MX_CommonTypes / Datatypes of logical addresses
Grundpaket PLC	

Tab. 14-2: Referenztable CONTROLS

Typdefinition

```
// Enumeration of valid logical control addresses
TYPE CONTROLS :
(
  LOCAL_CNTRL := 0,    // Local control (default)
);
END_TYPE
```

OBJECTS

Kurzbeschreibung Die Aufzählung der verfügbaren Achsen enthält alle Objekte eines IndraMotion-Verbundes, auf die per "AXIS_REF" durch IndraMotion-Bausteine zugegriffen werden kann.

Zuordnung Zielsystem/Bibliothek

Zielsystem	Bibliothek / Ordner
IndraMotion MLD 2G	MX_CommonTypes / Datatypes of logical addresses
Grundpaket PLC	

Tab. 14-3: Referenztable OBJECTS

Typdefinition

```
// not specified object
NO_OBJECT := 0,           // no object specified
// drives 1..19
  AXIS_1 := 1,             // drive 1
  // ...
  AXIS_19 := 19,          // drive 19
// virtual master axis
  VMA_1 := 20,             // virtual master axis 1
  VMA_2 := 21,             // virtual master axis 2
  VMA_INT := 22,           // internal virtual master axis
// real master axis
  RMA_1 := 23,             // real master axis 1
  RMA_2 := 24,             // real master axis 2
  RMA_INT := 25,           // internal real master axis
// link axis
  MA_LINK_1 := 26,         // link axis 1
  MA_LINK_2 := 27,         // link axis 2
  MA_LINK_INT := 28,       // internal link axis
```



Die Nummer "0", "NO_OBJECT" (es wurde kein Objekt angegeben) wird für deaktivierte Achsen verwendet.



Die Vergabe des Namens und der Nummern erfolgt beim Anlegen der jeweiligen Achse/ des Objektes in der IndraWorks-Suite.

TP_OBJECTS

Kurzbeschreibung

Die Aufzählung der verfügbaren Messtaster (Touch Probe) enthält Objekte eines IndraMotion-Verbundes, auf die durch Messtaster-Bausteine zugegriffen werden kann.

Zuordnung Zielsystem/Bibliothek

Zielsystem	Bibliothek /Ordner
IndraMotion MLD 2G Grundpaket PLC	MX_CommonTypes / Datatypes of logical addresses

Tab. 14-4: Referenztabelle TP_OBJECTS

Typdefinition

```
(* Enumeration of valid logical object addresses *)
TYPE TP_OBJECTS :
(
  (* not specified object *)
  NO_TOUCHPROBE := 0,           (* no object specified *)

  (* Touch Probe 1..100, realized on controller *)
  TOUCHPROBE_1 := 01,          (* Touch Probe 1 *)
  (* ... *)
  TOUCHPROBE_100 := 100,       (* Touch Probe 100 *)

  (* Touch Probe 1001..1002, realized on drive *)
  TOUCHPROBE_DRIVE_1 := 1001,  (* Touch Probe Drive 1 *)
  TOUCHPROBE_DRIVE_2 := 1002,  (* Touch Probe Drive 2 *)
);
END_TYPE
```



Die Vergabe des Namens und der Nummern erfolgt beim manuellen Anlegen des jeweiligen Messtasters in der IndraWorks-Suite.



Die Anzahl der realisierbaren Messtaster hängt vom Steuerungstyp und der verwendeten Firmwareversion ab.

Basisbibliotheken

14.1.5 Datatypes of PLCopen

AXIS_REF

Kurzbeschreibung Die Struktur enthält Informationen zur gewünschten Steuerung und dem von dieser Steuerung gesteuerten Objekt (Achse, Nockenschaltwerk, Oszilloskop...).

Zuordnung Zielsystem/Bibliothek

Zielsystem	Bibliothek / Ordner
IndraMotion MLD 2G	MX_CommonTypes / Datatypes of
Grundpaket PLC	PLCopen

Tab. 14-5: Referenztabelle AXIS_REF

Typdefinition

```
(* Structure of PLCopen element AXIS_REF. The AXIS_REF is a structure that contains *)
(* information on the corresponding Object. *)
(* It is used as a VAR_IN_OUT CONSTANT in all Motion Control Function Blocks. *)
TYPE AXIS_REF :
STRUCT
  CntrlNo: CONTROLS := LOCAL_CNTRL; (* control reference number *)
  AxisNo: OBJECTS := NO_OBJECT; (* Steuerungsnummer, Init: lokale Steuerung *)
  (* object reference number *)
  (* Achsreferenznummer, Init: kein Objekt *)
END_STRUCT
END_TYPE
```

Funktionsbeschreibung Die Informationsübergabe erfolgt ab Version 14VRS per **VAR_IN_OUT CONSTANT** an den jeweiligen PLCopen-Funktionsbaustein.

Die AXIS_REF-Struktur enthält zwei 16-Bit-Adresselemente:

- die Steuerungsadresse (Element von [Kap. "CONTROLS" auf Seite 198](#)) und
- die Objektadresse (Element von [Kap. "OBJECTS" auf Seite 198](#))

Die Elemente können per logischer oder physischer Adresse angesprochen werden. Im Falle einer logischen Adresse zeigt das Element auf eine Parameterstruktur mit einer physikalischen Adresse (z.B. Ethernet-Adresse der Steuerung, Sercos-Adresse des Antriebs o.ä.).



Die Informationsübergabe erfolgte bis einschließlich Version 13VRS per **VAR_IN_OUT**.

Ab Version 14VRS erfolgt die Informationsübergabe per **VAR_IN_OUT CONSTANT**.

MC_START_MODE

Kurzbeschreibung Dieser Parameter gibt die Art der automatischen dynamischen Synchronisation bei Phasensynchronisation an.

Zuordnung Zielsystem/Bibliothek

Zielsystem	Bibliothek / Ordner
IndraMotion MLD 2G Grundpaket PLC	MX_CommonTypes / Datatypes of PLCopen

Tab. 14-6: Referenztabelle MC_START_MODE

Typdefinition

```

TYPE MC_START_MODE :
(
  ABSOLUTE      := 0, (* Aufsynchronisieren mit absolute Winkelbezug, *)
                  (* nur bei erster Ausführung *)
  RELATIVE      := 1, (* Aufsynchronisieren auf die synchrone Geschwindigkeit, *)
                  (* nur bei erster Ausführung *)
  ABSOLUTE_RAMPIN := 2, (* Aufsynchronisieren mit absolute Winkelbezug, *)
                  (* bei jeder Ausführung *)
  RELATIVE_RAMPIN := 3 (* Aufsynchronisieren auf die synchrone Geschwindigkeit, *)
                  (* bei jeder Ausführung *)
);
END_TYPE
    
```

Absolute Synchronisation

Die Folgeachsposition beim Nulldurchgang der Leitachsposition ist eindeutig definiert und ändert sich nicht bei jedem Aufruf von MB_MotionProfile/ MC_CamIn.

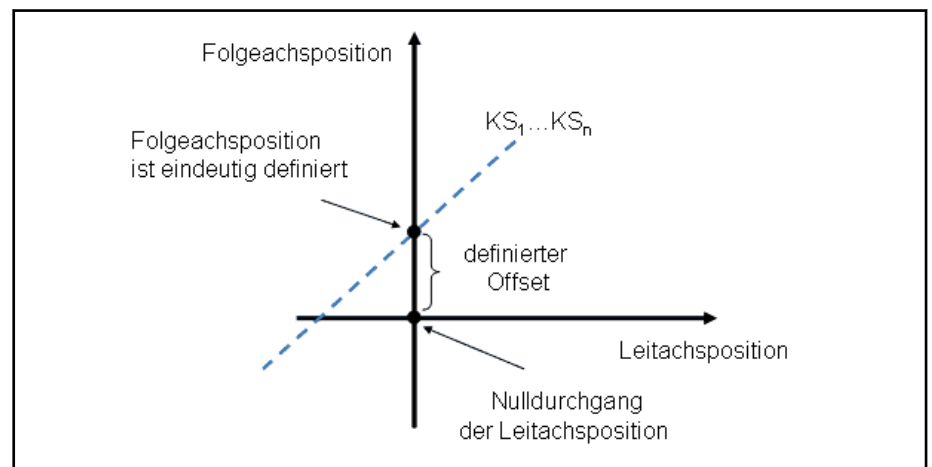


Abb. 14-2: Absolute Synchronisation

Relative Synchronisation

Die Folgeachsposition beim Nulldurchgang der Leitachsposition ergibt sich zufällig und ändert sich bei jedem Aufruf von MB_MotionProfile/ MC_CamIn.

Basisbibliotheken

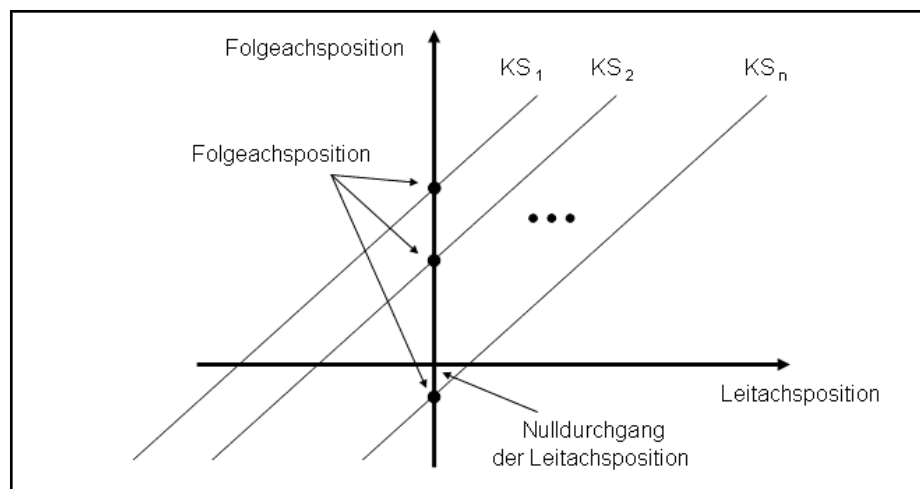


Abb. 14-3: Relative Synchronisation

"Harte" Synchronisation

- StartMode= 0: absolute Synchronisation, "hartes Umschalten" bei Profilwechsel,
- StartMode= 1: relative Synchronisation, "hartes Umschalten" bei Profilwechsel.

Die Umschaltung erfolgt nach Überfahren der Leitachs-Umschaltposition (P-0-0700) durch die Leitachse.

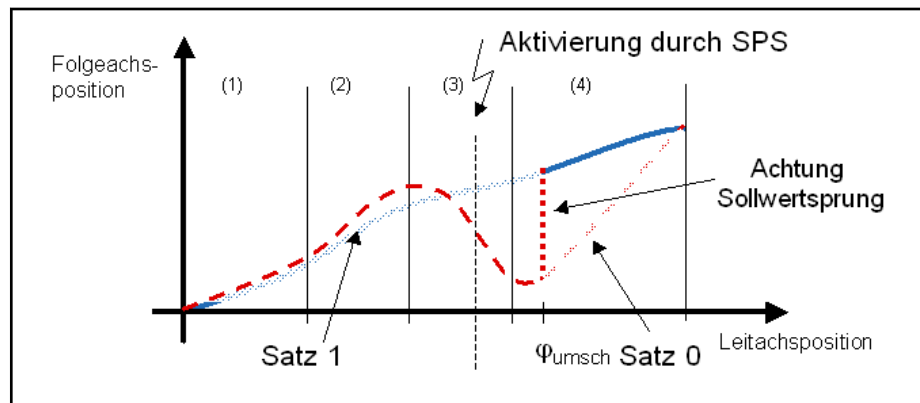


Abb. 14-4: Hartes Umschalten

"Weiche" Synchronisation

- StartMode= 2: absolute Synchronisation, "weiches Umschalten" bei Profilwechsel,
- StartMode= 3: relative Synchronisation, "weiches Umschalten" bei Profilwechsel.

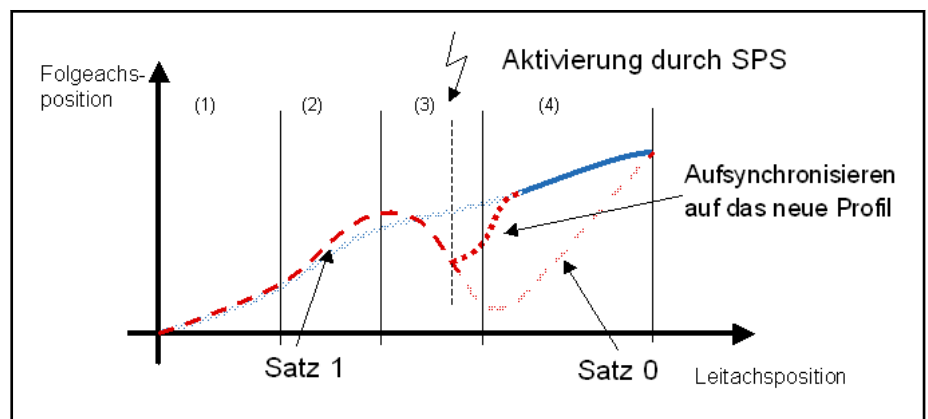


Abb. 14-5: Weiches Umschalten

Die Umschaltung erfolgt sofort nach Starten des Funktionsbausteins, es wird nicht auf das Überfahren der Leitachs-Umschaltposition (P-0-0700) durch die Leitachse gewartet.



Das "Weiche Aufsynchronisieren" setzt die Berechnung auf der Steuerung voraus.

MC_SYNC_DIRECTION

Kurzbeschreibung

Bei Modulo-Achsen kann die Bewegung der Achse in positiver oder negativer Richtung erfolgen.

Durch die Aufsynchronisierrichtung wird festgelegt, in welche Richtung sich die Achse bewegen soll.

Zuordnung Zielsystem/Bibliothek

Zielsystem	Bibliothek / Ordner
IndraMotion MLD 2G Grundpaket PLC	MX_CommonTypes / Datatypes of PLCopen

Tab. 14-7: Referenztable MC_SYNC_DIRECTION

Typdefinition

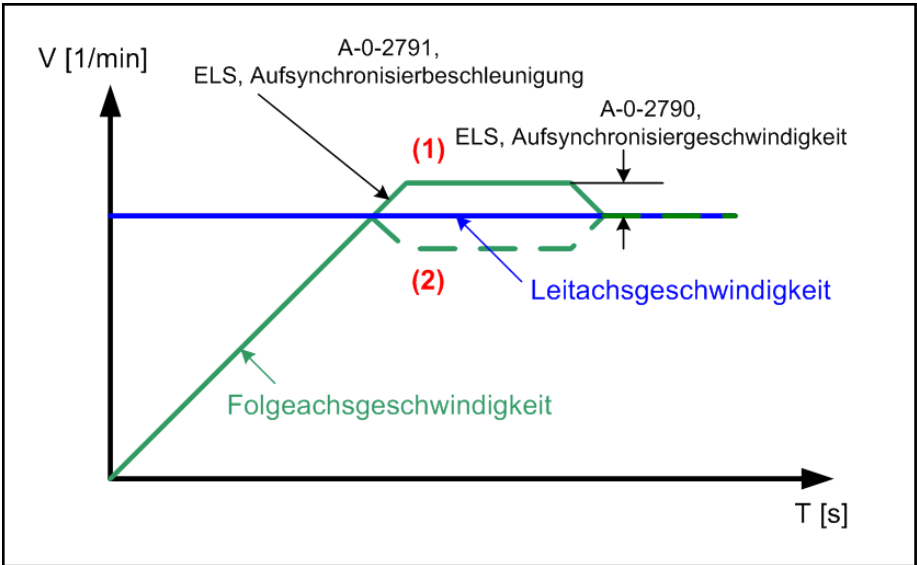
```

TYPE MC_SYNC_DIRECTION :
(
    SHORTEST_WAY      := 0,  (* kürzester Weg /shortest distance *)
    CATCH_UP          := 1,  (* immer vorwärts / always forward *)
    SLOW_DOWN         := 2
);
END_TYPE
    
```



Ist der kürzeste Weg zur absoluten Synchronisation jedoch kleiner als das "Aufsynchronisierfenster", bei Modulo-Format, wird der kürzeste Weg verfahren und die vorgegebene Aufsynchronisierrichtung ignoriert.

Beispiel MLC:



- ① Folgeachse läuft Leitachse nach
- ② Leitachse läuft Folgeachse nach

Abb. 14-6: Übergang geschwindigkeitssynchron nach lagesynchron

In dem gezeigten Beispiel beschleunigt zunächst die Folgeachse auf die gleiche Geschwindigkeit wie die Leitachse. Danach muss die Winkeldifferenz zwischen Leit- und Folgeachse abgebaut werden.

Beim Aufsynchronisieren wirken die beiden Parameter P-0-0142 (Aufsynchronisier-Beschleunigung/-Verzögerung) und P-0-0143 (Aufsynchronisier-Geschwindigkeit).

Der Vorgang ist abgeschlossen, wenn beide geschwindigkeits- und lagesynchron laufen.

TOUCHPROBE_REF

Kurzbeschreibung

Die Struktur enthält Informationen zur gewünschten Steuerung und dem von dieser Steuerung gesteuerten Objekt (Messtaster).

Zuordnung Zielsystem/Bibliothek

Zielsystem	Bibliothek / Ordner
IndraMotion MLD 2G	MX_CommonTypes / Datatypes of
Grundpaket PLC	PLCopen

Tab. 14-8: Referenztable TOUCHPROBE_REF

Typdefinition

(* Structure of PLCopen element TOUCHPROBE_REF. The TOUCHPROBE_REF is a structure that *)
(* contains information on the corresponding object. *)

```
TYPE TOUCHPROBE_REF :  
STRUCT  
  CntrlNo:          CONTROLS    := LOCAL_CNTRL; (* control reference number *)  
                                     (* Steuerungsnummer, Init: lokale Steuerung *)  
END_STRUCT  
END_TYPE
```

Funktionsbeschreibung

Die Informationsübergabe erfolgt ab Version 14VRS per VAR_IN_OUT CONSTANT an den jeweiligen PLCopen-Funktionsbaustein.

Die TOUCHPROBE_REF-Struktur enthält ein 16-Bit-Adresselement der Steuerungsadresse (Element von Kap. "CONTROLS" auf Seite 198).



Die Informationsübergabe erfolgte bis einschließlich Version 13VRS per **VAR_IN_OUT**.

Ab Version 14VRS erfolgt die Informationsübergabe per **VAR_IN_OUT CONSTANT**.

14.1.6 Struktur "ERROR_STRUCT"

Die Struktur "ERROR_STRUCT" wird für den Ausgang "ErrorIdent" verwendet. "ERROR_STRUCT" beinhaltet Information die den aufgetretenen Fehler näher beschreibt.

Die Elemente der Struktur "ERROR_STRUCT" sind:

- **Table**

"Table" ist vom Typ "ERROR_TABLE". Der Wert des Elements "Table" beinhaltet die Information, mit Hilfe welcher Fehlertabelle [INDRV_TABLE, SERCOS_TABLE oder MLD_TABLE] der Fehlercode des Elements "Additional1" entschlüsselt werden muss.

- **Additional1**

"Additional1" ist vom Typ "DWORD". Das Element "Additional1" beinhaltet den Fehlercode zur genauen Spezifizierung des aufgetretenen Fehlers.

- **Additional2**

"Additional2" ist vom Typ "DWORD". "Additional2" beinhaltet Zusatzinformationen (wenn vorhanden).

14.1.7 Datentypen zur Wiedergabe des Datums und absoluter Zeitangaben bei IndraMotion-Bausteinen

Wiedergabe des Datums und absoluter Zeitangaben, Einführung und Übersicht

Zur Wiedergabe des Datums und anderer absoluter Zeitangaben existieren drei Strukturen:

- [SYS_TIME64, Seite 206](#), Systemzeit in Mikrosekunden seit 01.01.1970, 00:00 Uhr.
- [EXT_SYS_TIME_DATE, Seite 206](#), Systemzeit in Mikrosekunden seit 01.01.1970, 00:00 Uhr, Zusatzdaten Jahr, Monat, Tag, Stunden, Minuten, Sekunden, Millisekunden und Wochentag werden aus der Systemzeit berechnet.
- [EXT_SYS_TIME_DATE, Seite 206](#), Jahr, Monat, Tag, Stunden, Minuten, Sekunden, Millisekunden und Wochentag, berechnet auf der Basis der Systemzeit in Mikrosekunden seit 01.01.1970, 00:00 Uhr.

Basisbibliotheken

SYS_TIME64

Kurzbeschreibung Struktur zur Wiedergabe der Systemzeit in Mikrosekunden seit 01.01.1970, 00:00 Uhr.

Zuordnung Zielsystem/Bibliothek

Zielsystem	Bibliothek / Ordner
IndraMotion MLD 2G Grundpaket PLC	MX_CommonTypes / Datatypes of times

*Tab. 14-9: Referenztable SYS_TIME64**Typdefinition*

```
(* Structure of system time - Microseconds since 1970 *)
TYPE SYS_TIME64 :
STRUCT
    LowValue:      DWORD;      (* Low value [us]      *)
    HighValue:     DWORD;      (* High value [us]    *)
END_STRUCT
END_TYPE
```

EXT_SYS_TIME_DATE

Kurzbeschreibung Struktur zur Wiedergabe der Systemzeit in Mikrosekunden seit 01.01.1970, 00:00 Uhr und der Zusatzdaten Jahr, Monat, Tag, Stunden, Minuten, Sekunden, Millisekunden und Wochentag, die aus der Systemzeit berechnet werden.

Zuordnung Zielsystem/Bibliothek

Zielsystem	Bibliothek / Ordner
IndraMotion MLD 2G Grundpaket PLC	MX_CommonTypes / Datatypes of times

*Tab. 14-10: Referenztable EXT_SYS_TIME_DATE**Typdefinition*

```
(* Extended structure of system time *)
TYPE EXT_SYS_TIME_DATE :
STRUCT
    LowValue: DWORD;      (* Low value [us]      *)
    HighValue: DWORD;     (* High value [us]    *)
    Year : UINT;          (* Year                *)
    Month : UINT;         (* Month               *)
    Day : UINT;           (* Day                 *)
    Hour : UINT;          (* Hour                *)
    Minute : UINT;        (* Minute              *)
    Second : UINT;        (* Second              *)
    Milliseconds : UINT;  (* Milliseconds        *)
    DayOfWeek: UINT;      (* Day of week         *)
END_STRUCT
END_TYPE
```

SYS_TIME_DATE

Kurzbeschreibung Struktur zur Wiedergabe der Daten Jahr, Monat, Tag, Stunden, Minuten, Sekunden, Millisekunden und Wochentag, die aus einer Systemzeit in Mikrosekunden seit 01.01.1970, 00:00 Uhr berechnet werden.

Zuordnung Zielsystem/Bibliothek

Zielsystem	Bibliothek / Ordner
IndraMotion MLD 2G	MX_CommonTypes / Datatypes of times
Grundpaket PLC	

Tab. 14-11: Referenztable SYS_TIME_DATE

Typdefinition

```
(* Structure of system time *)
TYPE SYS_TIME_DATE :
STRUCT
    Year : UINT;          (* Year *)
    Month : UINT;         (* Month *)
    Day : UINT;           (* Day *)
    Hour : UINT;          (* Hour *)
    Minute : UINT;        (* Minute *)
    Second : UINT;        (* Second *)
    Milliseconds : UINT;  (* Milliseconds *)
    DayOfWeek: UINT;     (* Day of week *)
END_STRUCT
END_TYPE
```

14.1.8 Datentypen zur Unterstützung von Messtaster-Funktionen

Datentypen zur Unterstützung von Messtaster-Funktionen, allgemeines

Der Abschnitt enthält Datentypen, die die Messtaster-Funktionen sowohl bei Abarbeitung im Antrieb als auch in der Steuerung unterstützen:

- [Kap. "MB_DIFF_EVAL" auf Seite 208](#), Einstellung der Methoden der Differenzmessung
- [Kap. "MB_EXPECT_WINDOW_MODE" auf Seite 208](#), Konfiguration und Aktivierung des Erwartungsfensters
- [Kap. "MB_INTERPOLATION_MODE" auf Seite 209](#), Aktivierung der Interpolation
- [Kap. "MB_POSITION_EVAL" auf Seite 210](#), Auswahl und Aktivierung des Triggersignals
- [Kap. "MB_PROBE_DATA" auf Seite 211](#), Auswahl der Messtaster-Bezugssignalquelle
- [Kap. "MB_PROBE_MODULE" auf Seite 212](#), Auswahl der Triggerquelle für den Messtaster
- [Kap. "MB_PROBE_INPUT" auf Seite 211](#), Auswahl des gewünschten Einganges auf dem ausgewählten Modul

Die zu einer Messtaster-Funktion gehörenden Daten sind in nachstehender Struktur zusammengestellt:

- [Kap. "MB_PROBE_CONFIG" auf Seite 210](#)

Basisbibliotheken

MB_DIFF_EVAL

Kurzbeschreibung Der Aufzählungstyp "MB_DIFF_EVAL" unterstützt die Einstellung der Methoden der Differenzmessung.

Zuordnung Zielsystem/Bibliothek

Zielsystem	Bibliothek / Ordner
IndraMotion MLD 2G Grundpaket PLC	MX_CommonTypes / TouchProbe

Tab. 14-12: Referenztabelle MB_DIFF_EVAL

Typdefinition

```
TYPE MB_DIFF_EVAL :
(
  DIFF_DISABLED      := 0, (* diff.measurement is disabled *)
  DIFF_STANDARD MODE := 1, (* diff. last two measured values*)
  DIFF_NEG_TO_NEG_EDGE:= 2, (* neg. edge to neg. edge diff. *)
  DIFF_POS_TO_NEG_EDGE:= 3, (* pos. edge to neg. edge diff. *)
  DIFF_NEG_TO_POS_EDGE:= 4, (* neg. edge to pos. edge diff. *)
  DIFF_POS_TO_POS_EDGE:= 5 (* pos. edge to pos. edge diff. *)
);
END_TYPE
```

Funktionsbeschreibung Folgende Einstellungen werden unterstützt:

- 0 - Differenzmessung ist abgeschaltet, Differenz-Signale für ML_TouchProbe bzw. ML_TouchProbe_Continuous werden nicht berechnet.
- 1 - Differenz zwischen den letzten zwei Messwerten, je nach aktivierter Flanke.
Beispiel: beide Flanken sind aktiviert, erfasst wird Differenz pos/neg Flanke, weiter neg/pos Flanke, weiter pos/neg Flanke.
- Im nachstehenden Bild sind die erweiterten Methoden 2..5 der Differenzmessung dargestellt:

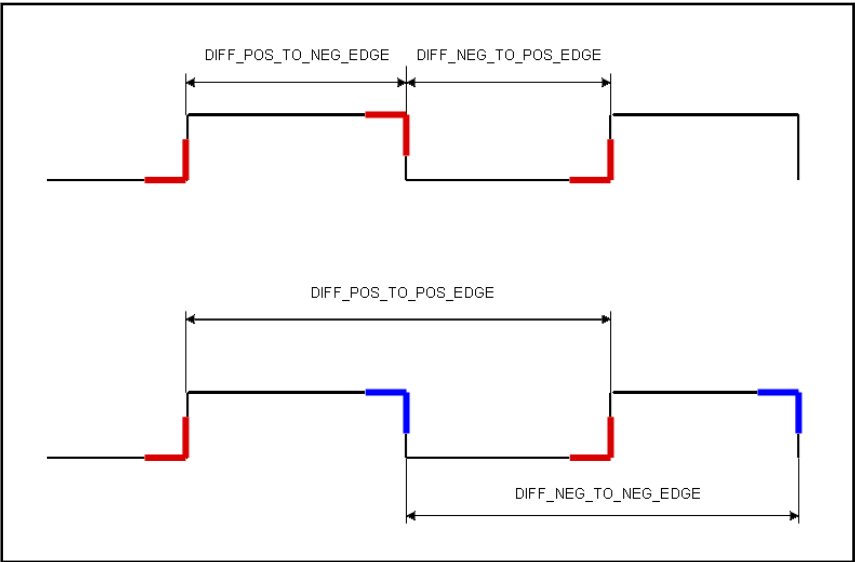


Abb. 14-7: Methoden der Differenzmessung

MB_EXPECT_WINDOW_MODE

Kurzbeschreibung Der Aufzählungstyp "MB_EXPECT_WINDOW_MODE" unterstützt die Konfiguration und Aktivierung des Erwartungsfensters.

Zuordnung Zielsystem/Bibliothek

Zielsystem	Bibliothek / Ordner
IndraMotion MLD 2G Grundpaket PLC	MX_CommonTypes / TouchProbe

Tab. 14-13: Referenztablette MB_EXPECT_WINDOW_MODE
 Typdefinition

```

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
  
```

Funktionsbeschreibung

Der Anwender kann sich getrennt für die positive und die negative Flanke des Messtasterwertes für ein Erwartungsfenster entscheiden.

Entscheidet sich der Anwender für die Nutzung eines Erwartungsfensters um , so kann er dessen Grenzen auf zwei unterschiedliche Arten definieren:

1. Erwartungsfenster deaktiviert:
Die jeweilige Flanke des Messtastersignals wird ungefiltert verwendet.
2. Erwartungsfenster absolut definiert:
Die Startposition und die Endposition des Erwartungsfensters sind absolut anzugeben. Der Mittelpunktwert ist in diesem Fall bedeutungslos.
3. Erwartungsfenster relativ definiert:
Der Mittelpunktwert des Erwartungsfensters sind absolut anzugeben. Die Startposition und die Endposition geben die Entfernung zum Mittelpunktswert an.

MB_INTERPOLATION_MODE

Kurzbeschreibung

Der Aufzählungstyp "MB_INTERPOLATION_MODE" unterstützt die Aktivierung der Interpolation.

Zuordnung Zielsystem/Bibliothek

Zielsystem	Bibliothek / Ordner
IndraMotion MLD 2G Grundpaket PLC	MX_CommonTypes / TouchProbe

Tab. 14-14: Referenztablette MB_INTERPOLATION_MODE
 Typdefinition

```

TYPE MB_INTERPOLATION_MODE:
(
  NO_INTERPOLATION      := 0, (* no interpolation *)
  LINEAR_INTERPOLATION  := 1, (* linear interpolation *)
  CUBIC_3_INTERPOLATION := 2, (* cubic interpolation with 3 setpoints *)
  CUBIC_4_INTERPOLATION := 3, (* cubic interpolation with 4 setpoints *)
);
END_TYPE
  
```

Basisbibliotheken

MB_POSITION_EVAL**Kurzbeschreibung**

Der Aufzählungstyp "MB_POSITION_EVAL" unterstützt die Auswahl und Aktivierung des Triggersignals.

Mit diesen Enumerationswerten kann ausgewählt werden, ob auf die positive oder die negative Flanke bzw. auf beide Flanken getriggert werden soll.



Der Defaultwert "Null" blockiert die Triggerung!

Zuordnung Zielsystem/Bibliothek

Zielsystem	Bibliothek / Ordner
IndraMotion MLD 2G	MX_CommonTypes / TouchProbe
Grundpaket PLC	

Tab. 14-15: Referenztable MB_POSITION_EVAL

Typdefinition

```

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, (* pos.and neg. edge are evaluated *)
);
END_TYPE

```

MB_PROBE_CONFIG**Kurzbeschreibung**

Die Struktur "MB_PROBE_CONFIG" dient der Aufnahme der Daten einer Messtaster-Funktion.

Zuordnung Zielsystem/Bibliothek

Zielsystem	Bibliothek / Ordner
IndraMotion MLD 2G	MX_CommonTypes / TouchProbe
Grundpaket PLC	

Tab. 14-16: Referenztable MB_PROBE_CONFIG

Typdefinition

```

TYPE MB_PROBE_CONFIG :
STRUCT
  ModuloValueDrive      :REAL;          (* Modulo-Wert *)
  ProcessingFormat       :BOOL;         (* Wichtung Lage, abs./modulo *)
  ProbePositionEval     :MB_POSITION_EVAL; (* Aktivierung Triggersignal, def.: Aus *)
  DifferenzMeasurment    :MB_DIFF_EVAL;  (* pos., neg., beide Flanken *)
  ExpectWindowPosMode    :MB_EXPECT_WINDOW_MODE; (* Methode der Differenzmessung, def.: Aus *)
  ExpectWindowNegMode    :MB_EXPECT_WINDOW_MODE; (* Mode des Erwartungsfensters pos. Flanke, *)
  ExpectWindowPosStart   :REAL;          (* def.: Aus, absolut / relativ zum Setpoint *)
  ExpectWindowPosEnd     :REAL;          (* Startpos. Erwartungsfensters pos. Flanke *)
  ExpectWindowNegStart   :REAL;          (* Endpos. Erwartungsfensters pos. Flanke *)
  ExpectWindowNegEnd     :REAL;          (* Startpos. Erwartungsfensters neg. Flanke *)
  ExpectWindowNegEnd     :REAL;          (* Endpos. Erwartungsfensters neg. Flanke *)
END_STRUCT
END_TYPE

```

"MB_PROBE_CONFIG" nutzt die folgenden Datentypen

- [Kap. "MB_DIFF_EVAL" auf Seite 208](#)
- [Kap. "MB_EXPECT_WINDOW_MODE" auf Seite 208](#)
- [Kap. "MB_EXPECT_WINDOW_MODE" auf Seite 208](#)

MB_PROBE_DATA

Kurzbeschreibung Der Aufzählungstyp "MB_PROBE_DATA" unterstützt die Auswahl der Messtaster-Bezugssignalquelle.

Zuordnung Zielsystem/Bibliothek

Zielsystem	Bibliothek / Ordner
IndraMotion MLD 2G Grundpaket PLC	MX_CommonTypes / TouchProbe

Tab. 14-17: Referenztable MB_PROBE_DATA

Typdefinition

```
TYPE MB_PROBE_DATA :  
(  
    TP_ACT_POS_FEEDBACK1      := 0, // S-0-0051  
    TP_ACT_POS_FEEDBACK2      := 1, // S-0-0053  
    TP_ACT_POSIN_ACT_CYCLE     := 2, // P-0-0753  
    TP_SYNC_CMD_POS           := 3, // P-0-0778  
    TP_PROBE_TIME              := 4, // Zeitmessung  
    TP_ACT_MEASURING_ENC_POS   := 5, // P-0-0052  
    TP_RES_MASTER_AXIS_POS     := 6, // P-0-0775  
    TP_EFF_MASTER_AXIS_POS     := 7, // P-0-0776  
    TP_CAM_SHAFT_PROF_ACCESS_ANGLE := 8, // P-0-0227  
);  
END_TYPE
```

MB_PROBE_INPUT

Kurzbeschreibung Der Aufzählungstyp "MB_PROBE_INPUT" unterstützt die Auswahl des gewünschten Einganges auf dem ausgewählten Modul.



Für das OnBoard -Modul stehen die Eingänge 1..8 zur Verfügung.
Die Fast IO- Module haben die Eingänge 1..16 als Triggerquelle.

Zuordnung Zielsystem/Bibliothek

Zielsystem	Bibliothek / Ordner
IndraMotion MLD 2G Grundpaket PLC	MX_CommonTypes / TouchProbe

Tab. 14-18: Referenztable MB_PROBE_INPUT

Typdefinition

```
TYPE MB_PROBE_INPUT :  
(  
    D_INPUT_1 := 1, (* digital input 1 *)  
    D_INPUT_2 := 2, (* digital input 2 *)  
    D_INPUT_3 := 3, (* digital input 3 *)  
    D_INPUT_4 := 4, (* digital input 4 *)  
    D_INPUT_5 := 5, (* digital input 5 *)  
    D_INPUT_6 := 6, (* digital input 6 *)  
    D_INPUT_7 := 7, (* digital input 7 *)  
    D_INPUT_8 := 8, (* digital input 8 *)  
    D_INPUT_9 := 9, (* digital input 9 *)  
    D_INPUT_10 := 10, (* digital input 10 *)  
    D_INPUT_11 := 11, (* digital input 11 *)  
    D_INPUT_12 := 12, (* digital input 12 *)  
    D_INPUT_13 := 13, (* digital input 13 *)  
    D_INPUT_14 := 14, (* digital input 14 *)  
    D_INPUT_15 := 15, (* digital input 15 *)  
    D_INPUT_16 := 16, (* digital input 16 *)  
);  
END_TYPE
```

Basisbibliotheken

MB_PROBE_MODULE

Kurzbeschreibung Der Aufzählungstyp "MB_PROBE_MODULE" unterstützt die Auswahl der Triggerquelle für den Messtaster.

Zuordnung Zielsystem/Bibliothek

Zielsystem	Bibliothek / Ordner
IndraMotion MLD 2G Grundpaket PLC	MX_CommonTypes / TouchProbe

Tab. 14-19: Referenztable MB_PROBE_MODULE

Typdefinition

```

TYPE MB_PROBE_MODULE :
(
  ONBOARD_MODULE      := 0, (* Onboard module MLC,      Inputs 1...8      *)
  FUNCTION_MODULE_1 := 1, (* Function module 1 MLC, Inputs 1..16 (CFL-01.1-E2, Fast IO) *)
  FUNCTION_MODULE_2 := 2, (* Function module 2 MLC, Inputs 1..16 (CFL-01.1-E2, Fast IO) *)
  FUNCTION_MODULE_3 := 3, (* Function module 3 MLC, Inputs 1..16 (CFL-01.1-E2, Fast IO) *)
  FUNCTION_MODULE_4 := 4, (* Function module 4 MLC, Inputs 1..16 (CFL-01.1-E2, Fast IO) *)
);
END_TYPE

```

14.1.9 Datentypen zur Unterstützung von Kurvenscheiben, Motion- und Flex-Profilen

Datentyp zum Umschaltung zwischen Kurvenscheiben, allgemeines

Rexroth IndraDrive unterstützt bis zu 8 Kurventabellen, welche sich in folgende Gruppen unterteilen lassen:

- 4 Kurventabellen mit 3 bis 1024 Stützpunkten (P-0-0072, P-0-0092, P-0-0780, P-0-0781)
- 4 Kurventabellen mit 3 bis 128 Stützpunkten (P-0-0782, P-0-0783, P-0-0784, P-0-0785)

Im laufenden Betrieb kann zwischen den Kurventabellen umgeschaltet werden. Auswahl und Umschaltung können über den Funktionsbaustein MC_CamIn oder über P-0-0088 erfolgen.

MC_CAM_ID

Kurzbeschreibung Der Aufzählungstyp unterstützt die Auswahl der gewünschten Kurventabelle.

Zuordnung Zielsystem/Bibliothek

Zielsystem	Bibliothek / Ordner
IndraMotion MLD 2G Grundpaket PLC	MX_CommonTypes / Datatypes of PLCopen

Tab. 14-20: Referenztable MC_CAM_ID

Typdefinition

```

TYPE MC_CAM_ID :
(
  CAM_TABLE_1 := 1, (* Kurventabelle 1 *)
  CAM_TABLE_2 := 2, (* Kurventabelle 2 *)
  (* ... *)
  CAM_TABLE_8 := 8, (* Kurventabelle 8 *)
);
END_TYPE

```

Funktionsbeschreibung Im laufenden Betrieb kann zwischen den Kurventabellen umgeschaltet werden. Auswahl und Umschaltung können über den Funktionsbaustein MC_CamIn oder über P-0-0088 erfolgen.

MC_STEP_MODE

Kurzbeschreibung Der Aufzählungsdatentyp MC_STEP_MODE beschreibt die Abhängigkeit der Folgeachseposition von der Leitachse während eines Bewegungsschrittes. Die Abhängigkeit kann gegeben sein als funktionale Abhängigkeit oder in Form einer Kurventabelle.

Zuordnung Zielsystem/Bibliothek

Zielsystem	Bibliothek / Ordner
IndraMotion MLD 2G	MX_CommonTypes / Datatypes of
Grundpaket PLC	PLCopen

Tab. 14-21: Referenztable MC_STEP_MODE

Typdefinition

```

TYPE MC_STEP_MODE :
(
    REST_IN_REST_INCLINEDSINE          := 16#0000, (* R-R: Rast in Rast, geneigte Sinuslinie *)
    REST_IN_REST_POLY5                  := 16#0100, (* R-R: Rast in Rast, Polynom 5. Ordnung *)
    REST_IN_VELOCITY_POLY5              := 16#0200, (* R-G: Rast in Geschw., Polynom 5. Ordnung *)
    REST_IN_VELOCITY_POLY7              := 16#0300, (* R-G: Rast in Geschw., Polynom 7. Ordnung *)
    VELOCITY_IN_REST_POLY5              := 16#0400, (* G-R: Geschw. in Rast, Polynom 5. Ordnung *)
    VELOCITY_IN_REST_POLY7              := 16#0500, (* G-R: Geschw. in Rast, Polynom 7. Ordnung *)
    CONSTANT_VELOCITY                   := 16#0600, (* G: konstante Geschwindigkeit *)
    VELOCITY_IN_VELOCITY_POLY5          := 16#0700, (* G-G: Geschw. in Geschw., Polynom 5. Ordnung *)

    REST_IN_REST_POLY5_VLIM              := 16#1200, (* MotionProfile, FlexProfile *)

    (* Kurventabellen *)
    CAMTABLE_1                          := 1,        (* Kurventabelle 1, P-0-0072 *)
    CAMTABLE_2                          := 2,        (* Kurventabelle 2, P-0-0092 *)
    CAMTABLE_3                          := 3,        (* Kurventabelle 3, P-0-0780 *)
    CAMTABLE_4                          := 4,        (* Kurventabelle 4, P-0-0781 *)
    CAMTABLE_5                          := 5,        (* Kurventabelle 5, P-0-0782 *)
    CAMTABLE_6                          := 6,        (* Kurventabelle 6, P-0-0783 *)
    CAMTABLE_7                          := 7,        (* Kurventabelle 7, P-0-0784 *)
    CAMTABLE_8                          := 8,        (* Kurventabelle 8, P-0-0785 *)
);
END_TYPE
    
```

Funktionsbeschreibung Als Modus eines Schrittes können verschiedene Normprofile oder eine Kurventabelle definiert werden.

Als Modi stehen zur Verfügung:

- R-R: Rast in Rast, Polynom 5. Ordnung,
- R-R: Rast in Rast, geneigte Sinuslinie,
- R-G: Rast in Geschwindigkeit, Polynom 5. Ordnung,
- G-R: Geschwindigkeit in Rast, Polynom 5. Ordnung,
- G: konstante Geschwindigkeit,
- G-G: Geschwindigkeit in Geschwindigkeit, Polynom 5. Ordnung,
- Kurventabellen 1 bis 8.

R-R: Rast in Rast Für das Profil "Rast in Rast" stehen zehn Profile zur Verfügung. Es kann zwischen den normierten Bewegungsgesetzen "Polynom 5. Ordnung" und "Geneigte Sinuslinie" ... gewählt werden.

Diese Profile sind durch den Hub und Leitachsbereich definiert. Die Endgeschwindigkeit des Schrittes wird vom Folgeschritt bestimmt.

Basisbibliotheken

Das Profil für die **"Polynom 5. Ordnung"** ist durch nachstehende Funktion bestimmt:

$$f(z) = 6z^5 - 15z^4 + 10z^3$$

Abb. 14-8: Rast in Rast, Polynom 5. Ordnung

Die möglichen Randwerte für das Bewegungsgesetz sind:

Randwert	z=0	z=1
1. Ordnung (Geschwindigkeit)	0	0
2. Ordnung (Beschleunigung)	0	0
3. Ordnung (Ruck)	automatisch	automatisch

Tab. 14-22: Randwerte für das Bewegungsgesetz - Polynom 5. Ordnung

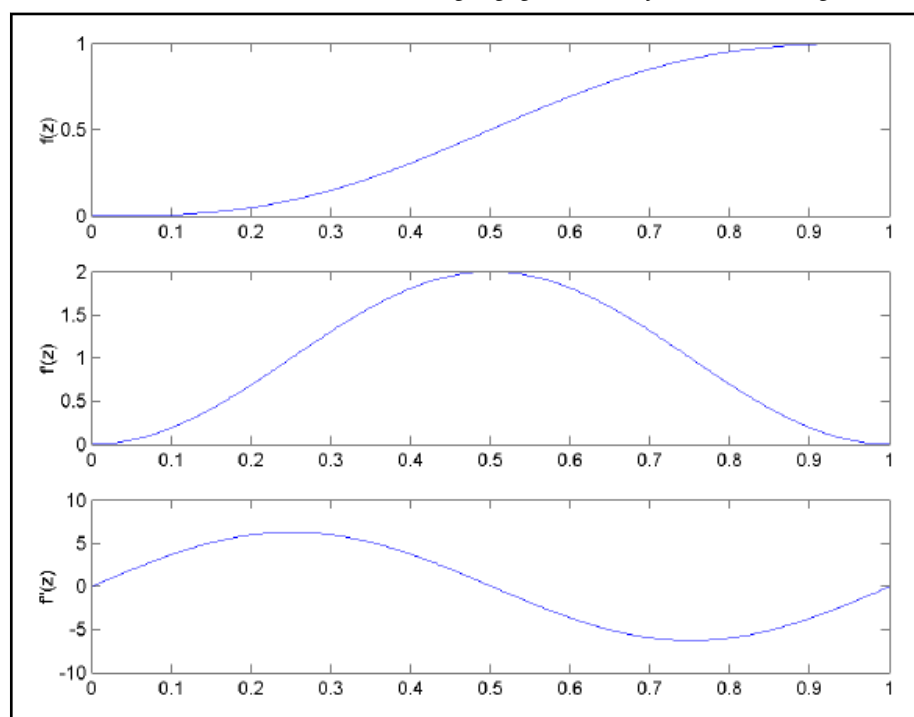


Abb. 14-9: Rast in Rast, Polynom 5. Ordnung

Für das **"Polynom 7. Ordnung"** gilt die Formel :

$$f(z) = -20z^7 + 70z^6 - 84z^5 - 35z^4$$

Abb. 14-10: Rast in Rast, Polynom 7. Ordnung

Die möglichen Randwerte für das Bewegungsgesetz sind:

Randwert	z=0	z=1
1. Ordnung (Geschwindigkeit)	0	0
2. Ordnung (Beschleunigung)	0	0
3. Ordnung (Ruck)	0	0

Tab. 14-23: Randwerte für das Bewegungsgesetz - Polynom 7. Ordnung

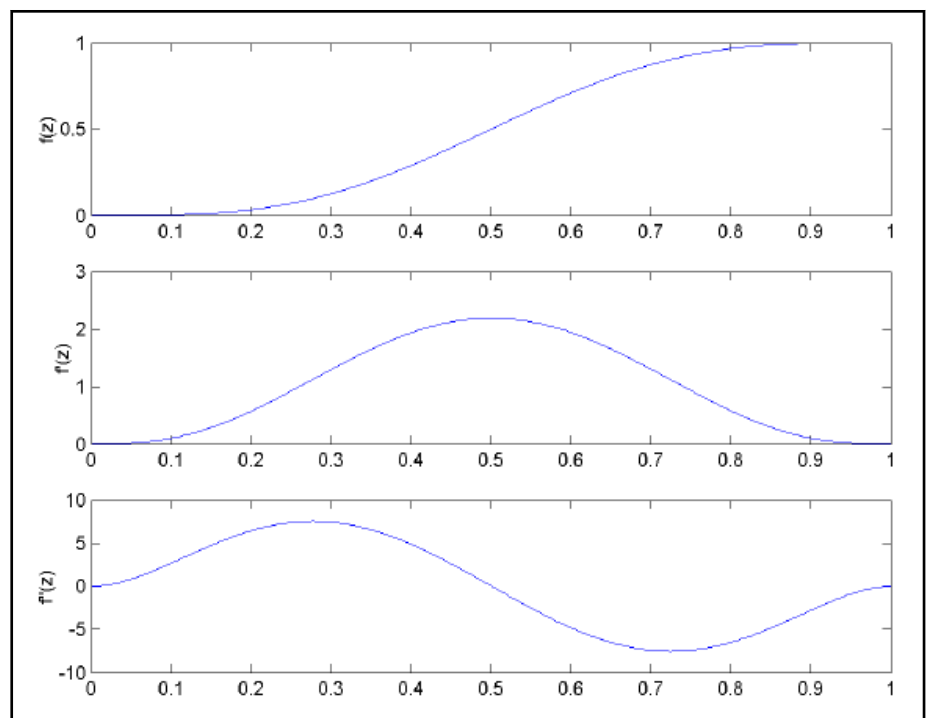


Abb. 14-11: Rast in Rast, Polynom 7. Ordnung

Das Profil für die **"Geneigte Sinuslinie"** ist durch nachstehende Funktion bestimmt:

$$f(z) = z - \frac{1}{2\pi} \sin(2\pi z)$$

Abb. 14-12: Rast in Rast, Geneigte Sinuslinie

Die möglichen Randwerte für das Bewegungsgesetz sind:

Randwert	z=0	z=1
1. Ordnung (Geschwindigkeit)	0	0
2. Ordnung (Beschleunigung)	0	0
3. Ordnung (Ruck)	automatisch	automatisch

Tab. 14-24: Randwerte für Bewegungsgesetz - geneigte Sinuslinie

Basisbibliotheken

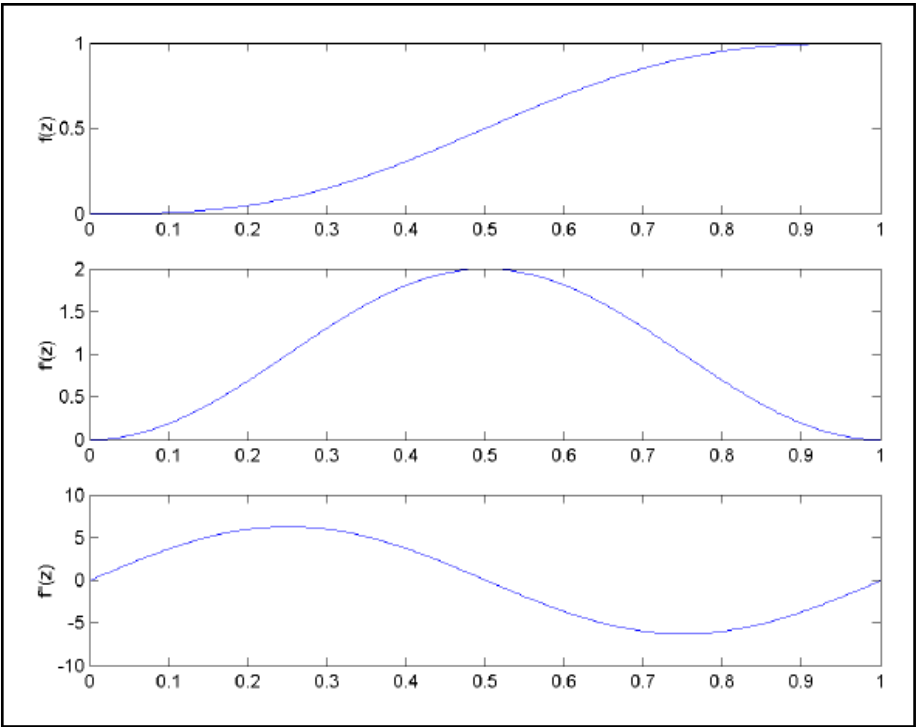


Abb. 14-13: *Rast in Rast, Geneigte Sinuslinie (BESTEHORN)*

Das Profil für die **"Sinuslinie (einfache Sinoide)"** ist durch nachstehende Funktion bestimmt:

$$f(z) = \frac{1}{2} \left[1 - \cos(\pi z) \right]$$

Abb. 14-14: *Rast in Rast, Sinuslinie*

Die möglichen Randwerte für das Bewegungsgesetz sind:

Randwert	z=0	z=1
1. Ordnung (Geschwindigkeit)	0	0
2. Ordnung (Beschleunigung)	automatisch	automatisch
3. Ordnung (Ruck)	automatisch	automatisch

Tab. 14-25: *Randwerte für das Bewegungsgesetz Sinuslinie*

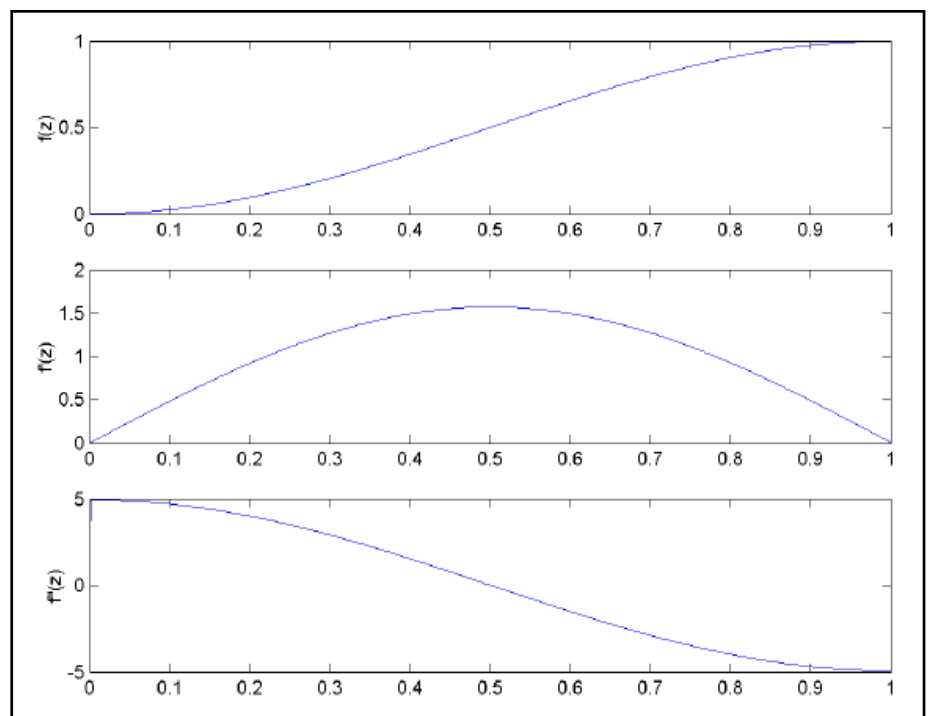


Abb. 14-15: Rast in Rast, Sinuslinie (einfache Sinoide)

Das Profil für die **"Lineare Rast"** ist durch nachstehende Funktion bestimmt:

$$f(z) = 0$$

Abb. 14-16: Lineare Rast

Die möglichen Randwerte für das Bewegungsgesetz sind:

Randwert	z=0	z=1
1. Ordnung (Geschwindigkeit)	0	0
2. Ordnung (Beschleunigung)	0	0
3. Ordnung (Ruck)	0	0

Tab. 14-26: Randwerte für Bewegungsgesetze - Lineare Rast

Basisbibliotheken

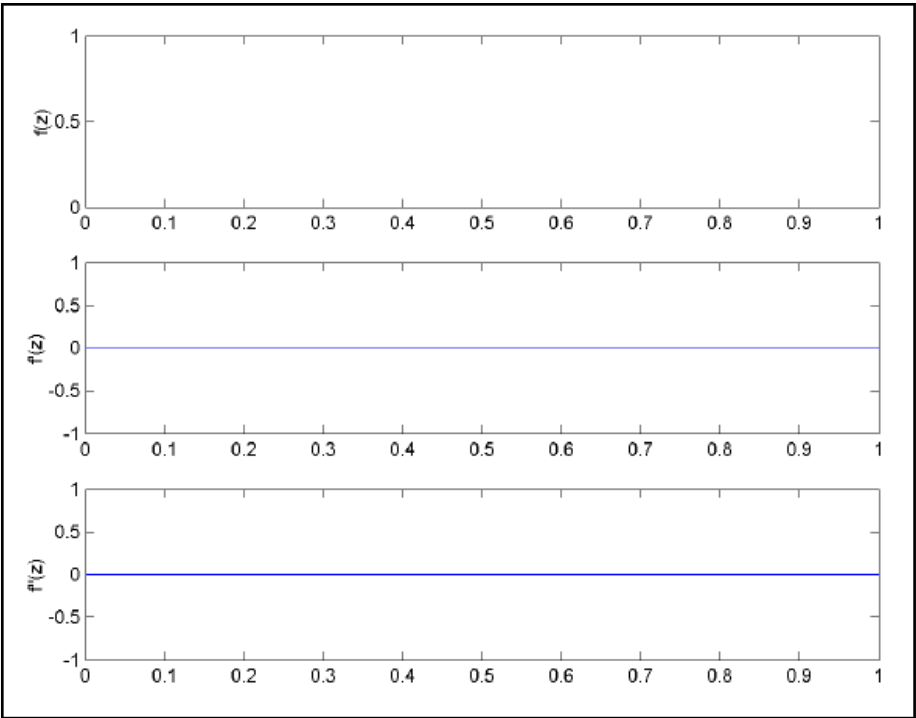


Abb. 14-17: Lineare Rast

Das Profil für die **"Sinoide von Gutman"** ist durch nachstehende Funktion bestimmt:

$$f(z) = z - \frac{15}{32\pi} \sin(2\pi z) - \frac{1}{96\pi} \sin(6\pi z)$$

Abb. 14-18: Sinoide nach Gutman

Die möglichen Randwerte für das Bewegungsgesetz sind:

Randwert	z=0	z=1
1. Ordnung (Geschwindigkeit)	0	0
2. Ordnung (Beschleunigung)	0	0
3. Ordnung (Ruck)	automatisch	automatisch

Tab. 14-27: Randwerte für das Bewegungsgesetz - Sinoide von Gutman

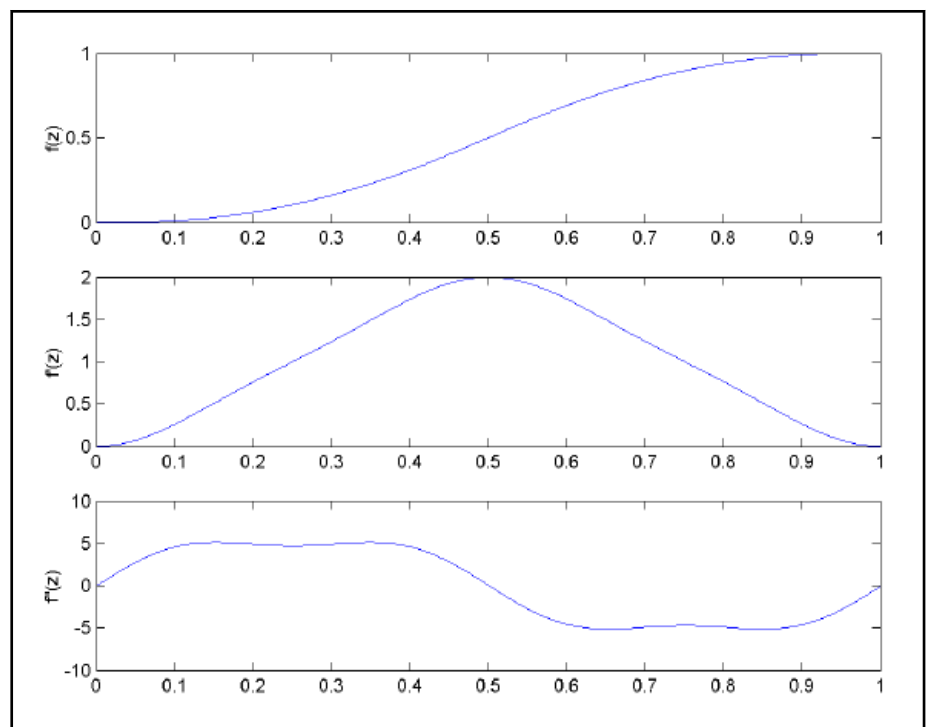


Abb. 14-19: Rast in Rast, Sinoide von Gutman

Das Profil für die **"Beschleunigungsoptimale geneigte Sinuslinie"** ist durch nachstehende Funktion bestimmt:

$$f(z) = z - \frac{1}{2\pi} \sin(2\pi z)$$

$$g(z) = z - \frac{0.41}{2\pi} \sin(2\pi z)$$

Abb. 14-20: Beschleunigungsoptimale geneigte Sinuslinie

Die möglichen Randwerte für das Bewegungsgesetz sind:

Randwert	z=0	z=1
1. Ordnung (Geschwindigkeit)	0	0
2. Ordnung (Beschleunigung)	0	0
3. Ordnung (Ruck)	automatisch	automatisch

Tab. 14-28: Randwerte für das Bewegungsgesetz - Beschleunigungsoptimale geneigte Sinuslinie

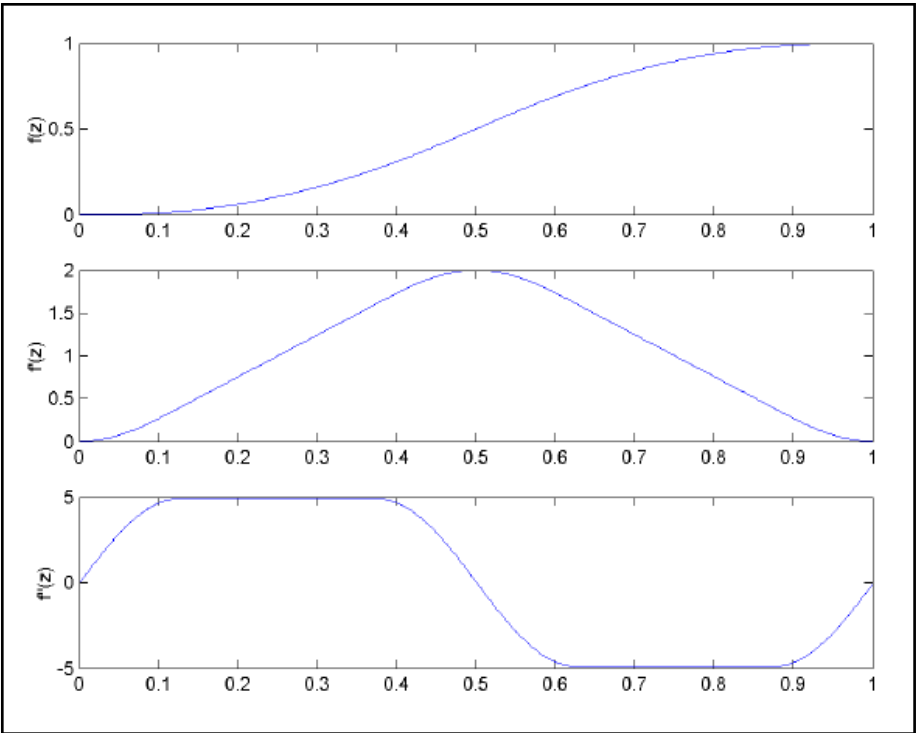


Abb. 14-21: Rast in Rast, Beschleunigungsoptimale geneigte Sinuslinie
 Das Profil für die **"Momentenoptimale geneigte Sinuslinie"** ist durch nachstehende Funktion bestimmt:

$$f(z) = z - \frac{1}{2\pi} \sin(2\pi z)$$

$$g(z) = z - \frac{0.134}{2\pi} \sin(2\pi z)$$

Abb. 14-22: Momentenoptimale geneigte Sinuslinie
 Die möglichen Randwerte für das Bewegungsgesetz sind:

Randwert	z=0	z=1
1. Ordnung (Geschwindigkeit)	0	0
2. Ordnung (Beschleunigung)	0	0
3. Ordnung (Ruck)	automatisch	automatisch

Tab. 14-29: Randwerte für das Bewegungsgesetz - Momentenoptimale geneigte Sinuslinie

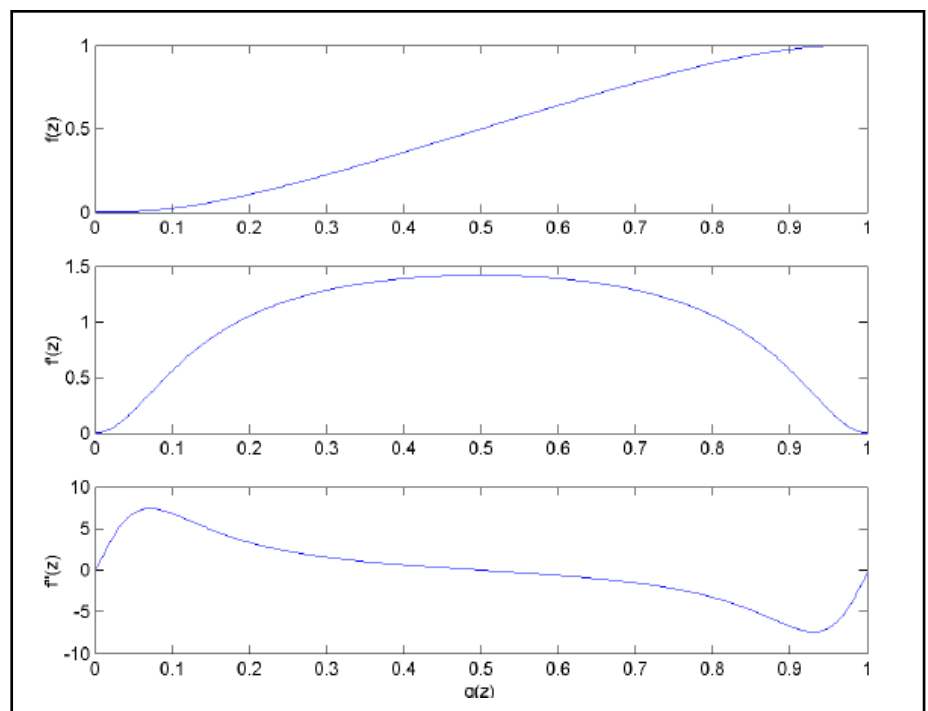


Abb. 14-23: Rast in Rast, Momentenoptimale geneigte Sinuslinie

Das Profil für die **"Modifizierte Sinoide"** ist durch nachstehende Funktion bestimmt:

$$\begin{aligned}
 0 \leq z \leq \frac{1}{8}: f(z) &= \frac{1}{4+\pi} \left[\pi z - \frac{1}{4} \sin(4\pi z) \right] \\
 \frac{1}{8} \leq z \leq \frac{7}{8}: f(z) &= \frac{1}{4+\pi} \left[2 + \pi z - \frac{4}{9} \sin\left(\frac{4\pi}{3}z + \frac{\pi}{3}\right) \right] \\
 \frac{7}{8} \leq z \leq 1: f(z) &= \frac{1}{4+\pi} \left[4 + \pi z - \frac{1}{4} \sin(4\pi z) \right]
 \end{aligned}$$

Abb. 14-24: Modifizierte Sinoide

Die möglichen Randwerte für das Bewegungsgesetz sind:

Randwert	z=0	z=1
1. Ordnung (Geschwindigkeit)	0	0
2. Ordnung (Beschleunigung)	0	0
3. Ordnung (Ruck)	automatisch	automatisch

Tab. 14-30: Randwerte für das Bewegungsgesetz - Modifizierte Sinoide

Basisbibliotheken

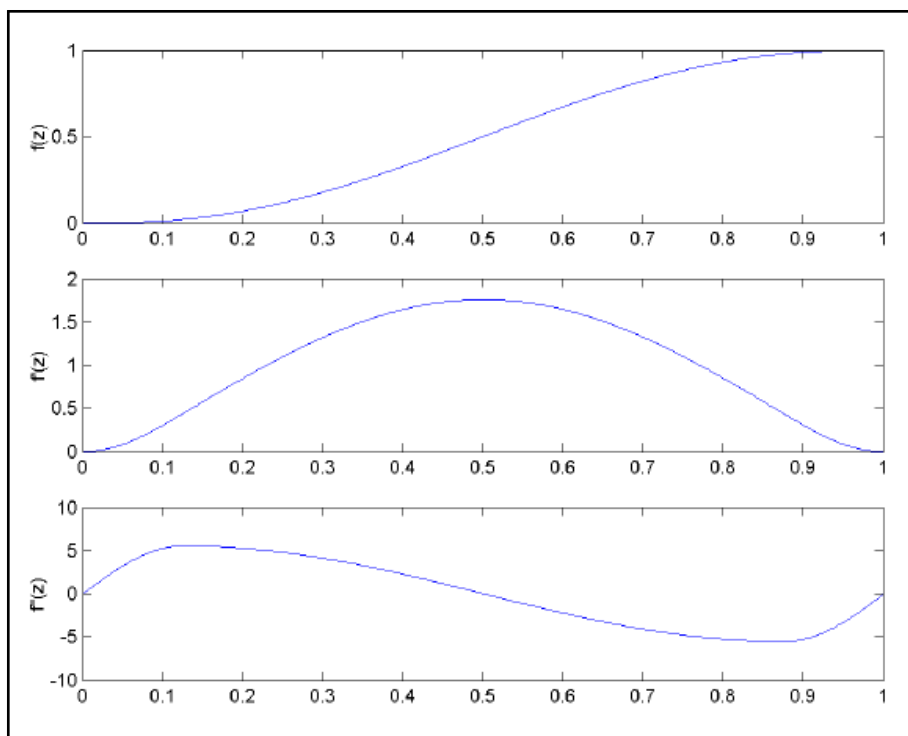


Abb. 14-25: Rast in Rast, Modifizierte Sinoide

Das Profil für die **"Modifiziertes Beschleunigungstrapez"** ist durch nachstehende Funktion bestimmt:

$$\begin{aligned}
 0 \leq z \leq \frac{1}{8}: f(z) &= \frac{1}{2+\pi} \left[2z - \frac{1}{2\pi} \sin(4\pi z) \right] \\
 \frac{1}{8} \leq z \leq \frac{3}{8}: f(z) &= \frac{1}{2+\pi} \left[-\frac{1}{2\pi} + 2z + 4\pi \left(z + \frac{1}{8} \right)^2 \right] \\
 \frac{3}{8} \leq z \leq \frac{1}{2}: f(z) &= \frac{1}{2+\pi} \left[-\frac{\pi}{2} + 2(1+\pi)z - \frac{1}{2\pi} \sin(4\pi z - \pi) \right] \\
 \frac{1}{2} \leq z \leq 1: f(z) &= 1 - f(1-z)
 \end{aligned}$$

Abb. 14-26: Modifiziertes Beschleunigungstrapez

Die möglichen Randwerte für das Bewegungsgesetz sind:

Randwert	z=0	z=1
1. Ordnung (Geschwindigkeit)	0	0
2. Ordnung (Beschleunigung)	0	0
3. Ordnung (Ruck)	automatisch	automatisch

Tab. 14-31: Randwerte für das Bewegungsgesetz - Modifiziertes Beschleunigungstrapez

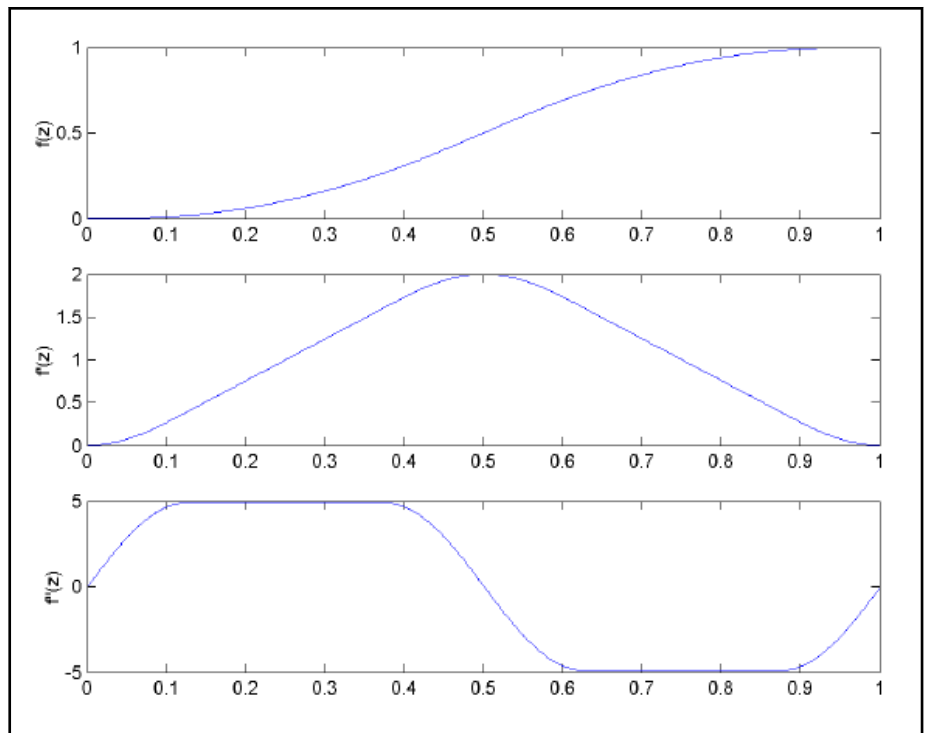


Abb. 14-27: Rast in Rast, Modifiziertes Beschleunigungstrapez

R-G: Rast in Geschwindigkeit

Dieses Profil dient zum Übergang aus dem Stillstand in eine bestimmte Geschwindigkeit.

Das Profil wird mit einem "Polynom 5. Ordnung" berechnet :

$$f(z) = A_5 z^5 + A_4 z^4 + A_3 z^3$$

Abb. 14-28: Polynom 5. Ordnung

Die möglichen Randwerte für das Bewegungsgesetz sind:

Randwert	z=0	z=1
1. Ordnung (Geschwindigkeit)	0	wählbar
2. Ordnung (Beschleunigung)	0	wählbar
3. Ordnung (Ruck)	automatisch	automatisch

Tab. 14-32: Randwerte für das Bewegungsgesetz Polynom 5. Ordnung

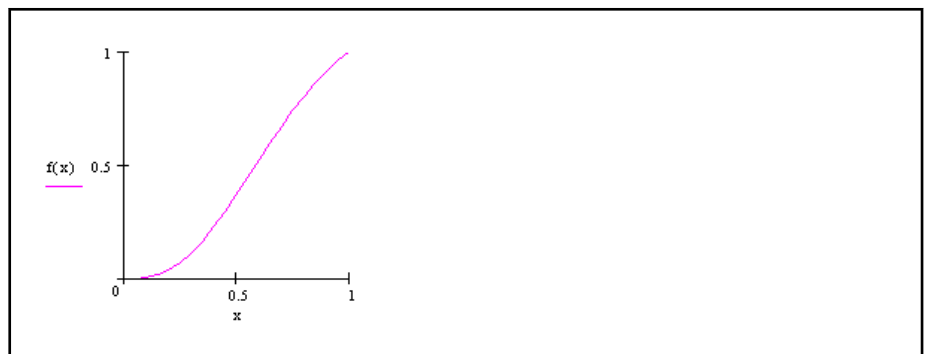


Abb. 14-29: Rast in Geschwindigkeit, Polynom 5. Ordnung

Dieses Profil dient zum Übergang aus dem Stillstand in eine bestimmte Geschwindigkeit.

Basisbibliotheken

Das Profil wird mit einem **"Polynom 7. Ordnung"** berechnet :

$$f(z) = A_7 z^7 + A_6 z^6 + A_5 z^5 + A_4 z^4 + A_3 z^3$$

Abb. 14-30: Polynom 7. Ordnung

Die möglichen Randwerte für das Bewegungsgesetz sind:

Randwert	z=0	z=1
1. Ordnung (Geschwindigkeit)	0	wählbar
2. Ordnung (Beschleunigung)	0	wählbar
3. Ordnung (Ruck)	automatisch	wählbar

Tab. 14-33: Randwerte für das Bewegungsgesetz - Polynom 7. Ordnung

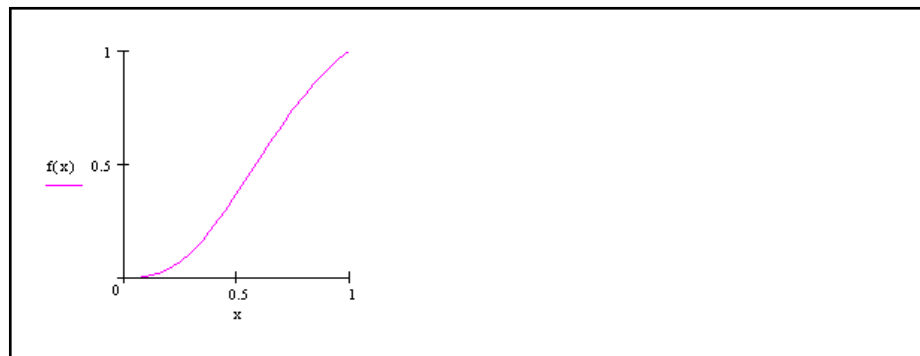


Abb. 14-31: Rast in Geschwindigkeit, Polynom 7. Ordnung

G-R: Geschwindigkeit in Rast

Das Profil wird zum Übergang aus einer definierten Geschwindigkeit in den Stillstand verwendet.

Bei diesem Bewegungsprofil wird ein Weg verfahren, der durch den Hub für diesen Bewegungsschritt festgelegt ist.

Das Profil wird mit einem **"Polynom 5. Ordnung"** berechnet :

$$f(z) = A_5 z^5 + A_4 z^4 + A_3 z^3 + A_1 z$$

Abb. 14-32: Geschwindigkeit in Rast, Polynom 5. Ordnung

Die möglichen Randwerte für das Bewegungsgesetz sind:

Randwert	z=0	z=1
1. Ordnung (Geschwindigkeit)	wählbar	0
2. Ordnung (Beschleunigung)	wählbar	0
3. Ordnung (Ruck)	automatisch	0

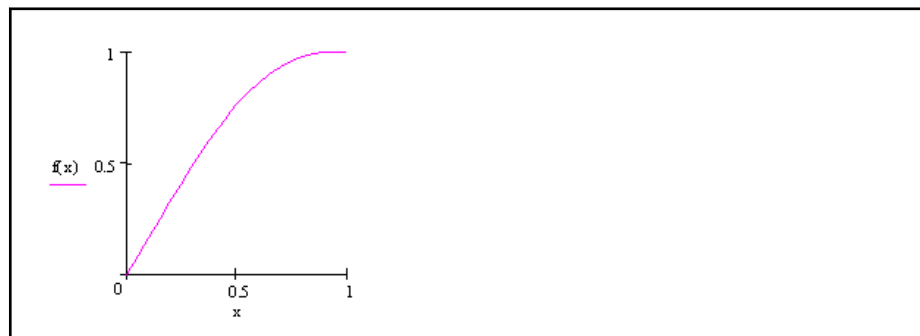


Abb. 14-33: Geschwindigkeit in Rast, Polynom 5. Ordnung

Das Profil wird zum Übergang aus einer definierten Geschwindigkeit in den Stillstand verwendet.

Das Profil wird mit einem **"Polynom 7. Ordnung"** berechnet :

$$f(z) = A_7 z^7 + A_6 z^6 + A_5 z^5 + A_4 z^4 + A_3 z^3 + A_2 z^2 + A_1 z$$

Abb. 14-34: Rast in Geschwindigkeit, Polynom 7. Ordnung

Die möglichen Randwerte für das Bewegungsgesetz sind:

Randwert	z=0	z=1
1. Ordnung (Geschwindigkeit)	wählbar	0
2. Ordnung (Beschleunigung)	wählbar	0
3. Ordnung (Ruck)	wählbar	0

Tab. 14-34: Randwerte für das Bewegungsgesetz - Polynom 7. Ordnung

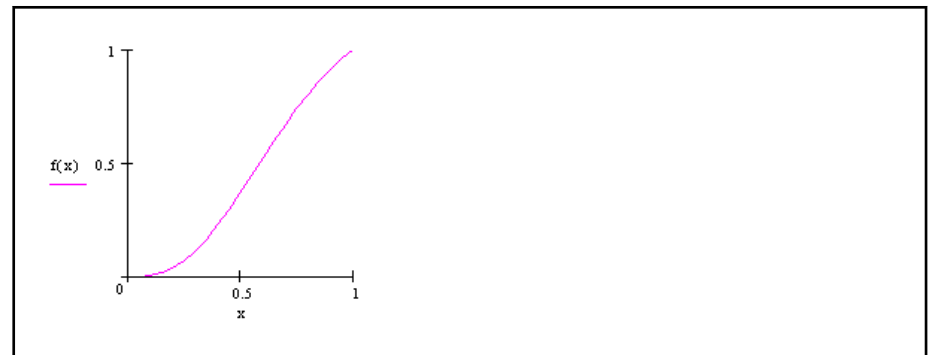


Abb. 14-35: Rast in Geschwindigkeit, Polynom 7. Ordnung

G: konstante Geschwindigkeit

Das Profil dieses Bewegungsschrittes ist eine **"Gerade"**. Die Geschwindigkeit innerhalb dieses Schrittes ist konstant. Es wird ein Weg verfahren, der durch den Hub für diesen Bewegungsschritt festgelegt ist.

Die Formel für das Profil lautet in normierter Form:

$$f(z) = z.$$

Die möglichen Randwerte für das Bewegungsgesetz sind:

Randwert	z=0	z=1
1. Ordnung (Geschwindigkeit)	automatisch	automatisch
2. Ordnung (Beschleunigung)	automatisch	automatisch
3. Ordnung (Ruck)	automatisch	automatisch

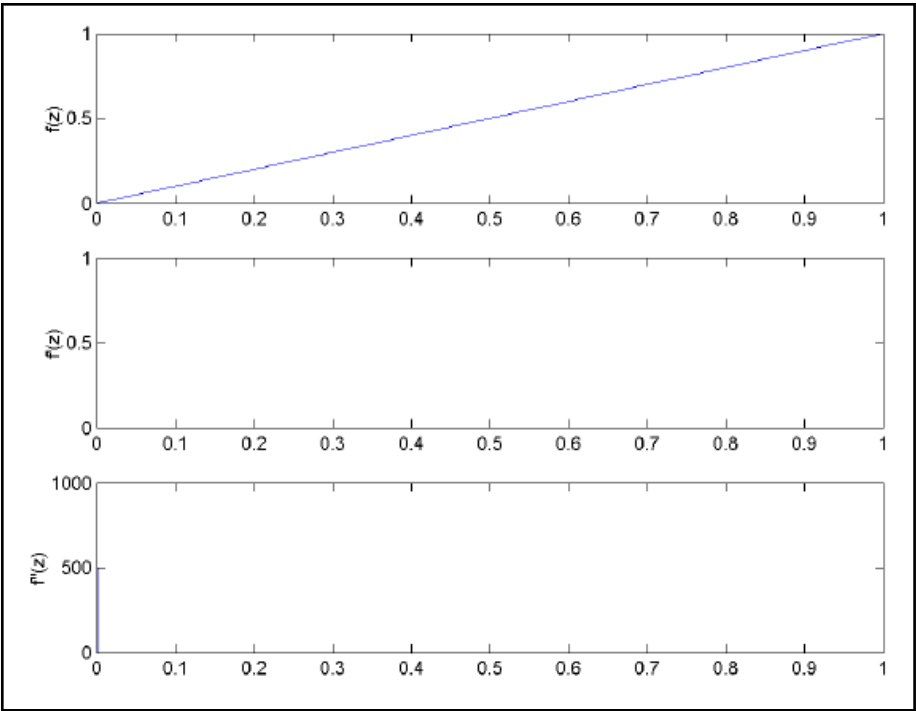


Abb. 14-36: Konstante Geschwindigkeit

Die Folgeachs-Geschwindigkeit wird aus den für diesen Bewegungsschritt vorgegebenen Werten bestimmt:

- Leitachsbereich
- Hub
- Leitachsgeschwindigkeit.

Die Eingabe eines Wertes für die Folgeachs-Geschwindigkeit ist nicht erforderlich.

G-G: Geschwindigkeit in Geschwindigkeit

Das Profil für die "Polynom 5. Ordnung" ist durch nachstehende Funktion bestimmt:

$$f(z) = A_5 z^5 + A_4 z^4 + A_3 z^3 + A_2 z^2 + A_1 z$$

Abb. 14-37: Polynom 5. Ordnung

Die möglichen Randwerte für das Bewegungsgesetz sind:

Randwert	z=0	z=1
1. Ordnung (Geschwindigkeit)	wählbar	wählbar
2. Ordnung (Beschleunigung)	wählbar	wählbar
3. Ordnung (Ruck)	automatisch	automatisch

Tab. 14-35: Randwerte für das Bewegungsgesetz - Polynom 5. Ordnung

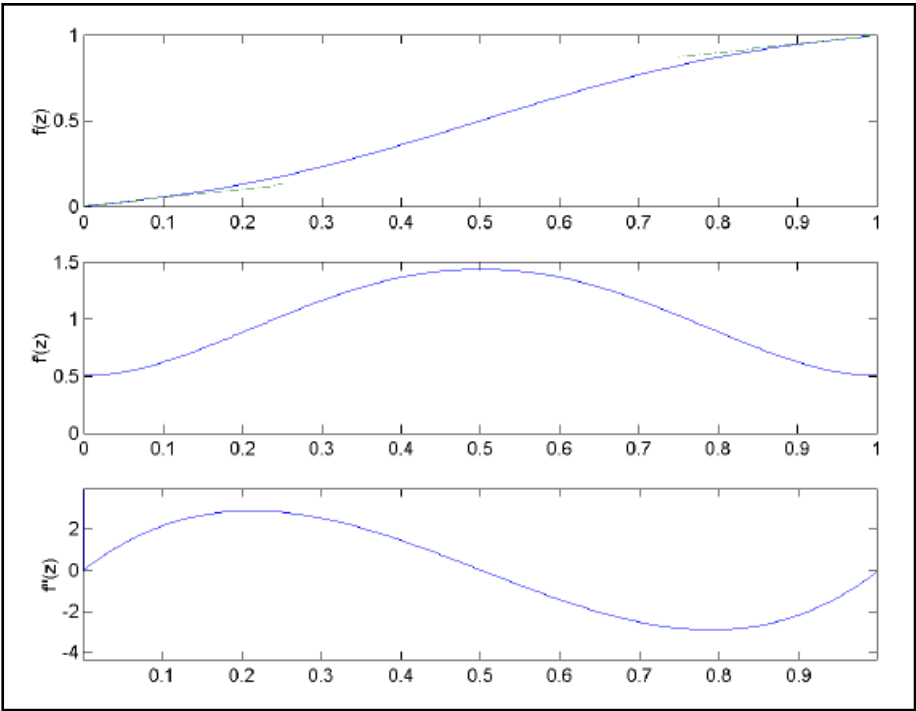


Abb. 14-38: Geschwindigkeit in Geschwindigkeit, Polynom 5. Ordnung
Das Profil für die "Polynom 7. Ordnung" ist durch nachstehende Funktion bestimmt:

$$f(z) = A_7 z^7 + A_6 z^6 + A_5 z^5 + A_4 z^4 + A_3 z^3 + A_2 z^2 + A_1 z$$

Abb. 14-39: Polynom 7. Ordnung
Die möglichen Randwerte für das Bewegungsgesetz sind:

Randwert	z=0	z=1
1. Ordnung (Geschwindigkeit)	wählbar	wählbar
2. Ordnung (Beschleunigung)	wählbar	wählbar
3. Ordnung (Ruck)	wählbar	wählbar

Tab. 14-36: Randwerte für das Bewegungsgesetz - Polynom 7. Ordnung

Basisbibliotheken

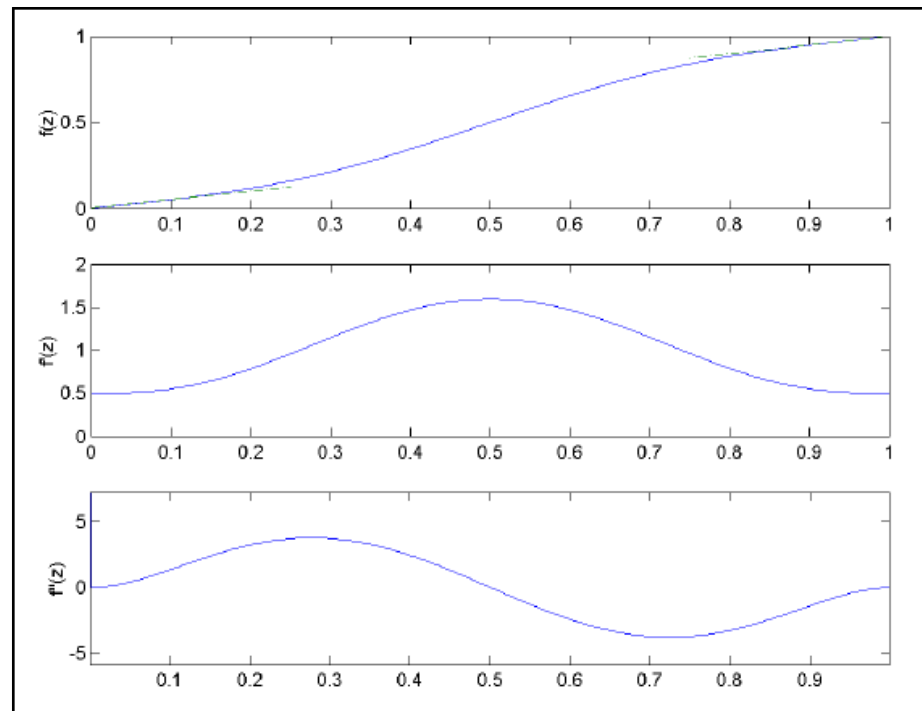


Abb. 14-40: Geschwindigkeit in Geschwindigkeit, Polynom 7. Ordnung

Das Profil für die **"Modifizierte Sinuslinie"** ist durch nachstehende Funktion bestimmt:

$$\begin{aligned}
 0 \leq z \leq \frac{\lambda}{8}: f'(z) &= C_a \frac{\lambda}{2\pi} \left(z - \frac{\lambda}{2\pi} \sin \left(2\pi \frac{z}{\lambda} \right) \right) + f'(0)z \\
 \frac{\lambda}{8} \leq z \leq \lambda: f'(z) &= C_a \frac{\lambda}{2\pi} \left(z - \frac{9\lambda}{2\pi} \sin \left(2\pi \frac{z + \frac{\lambda}{2}}{3\lambda} \right) + \frac{4\lambda}{\pi} \right) + f'(0)z \\
 \lambda \leq z \leq 1 - \frac{1-\lambda}{4}: f'(z) &= C_b \frac{1-\lambda}{2\pi} \left(z - \frac{9(1-\lambda)}{2\pi} \sin \left(2\pi \frac{z + \frac{3-5\lambda}{3(1-\lambda)}}{3\lambda} \right) + \frac{4\lambda}{\pi} \right) \\
 &+ f'(1) + (f'(0) - f'(1)) \left(\frac{3}{4} - \frac{1}{\pi} \right) + \lambda \\
 1 - \frac{1-\lambda}{4} \leq z \leq 1: f'(z) &= -C_b \frac{1-\lambda}{2\pi} \left(z - \frac{1-\lambda}{2\pi} \sin \left(2\pi \frac{z + \lambda}{1-\lambda} \right) - 1 \right) \\
 &+ 1 + f'(1) + (z-1)
 \end{aligned}$$

Abb. 14-41: Modifizierte Sinuslinie

Die möglichen Randwerte für das Bewegungsgesetz sind:

Randwert	z=0	z=1
1. Ordnung (Geschwindigkeit)	wählbar	wählbar
2. Ordnung (Beschleunigung)	0	0
3. Ordnung (Ruck)	0	0

Tab. 14-37: Randwerte für das Bewegungsgesetz - Modifizierte Sinuslinie

Erweiterte Bewegungsgesetze

Bei den erweiterten Bewegungsgesetzen handelt es sich um spezielle Bewegungsgesetze. Sie unterscheiden sich von den in den VDI-Richtlinien beschriebenen Bewegungsgesetzen durch neue Freiheitsgrade. Während ein Standard-Bewegungsgesetz durch die Angabe seiner linken und rechten Randwerte eindeutig definiert ist, erlauben die hier dargestellten Bewegungsgesetze noch weitere unterschiedliche Einstellungsmöglichkeiten, wie z. B. eine geforderte maximale Geschwindigkeit. In dem Fall ist der Leitachsabschnitt des Bewegungsschrittes nicht anzugeben, er ergibt sich aus der Parametrierung des Bewegungsgesetzes und der Festlegung von Kennwerten. So erlaubt beispielsweise die beschleunigungsbegrenzte Bewegung die Angabe von Hub, Anfangs- und Endgeschwindigkeit, nicht aber die Vorgabe eines Leitachsabschnitts. Der Leitachsabschnitt, und damit die Dauer des Bewegungsschrittes, ergibt sich aus den ebenfalls festlegbaren Werten einer maximalen Beschleunigung und einer maximalen Geschwindigkeit.

	Hub	Leitachsabschnitt	$v_{\max}$	$a_{\max}$	$j_{\max}$
Überschwingungsfreies Polynom 5. Ordnung	wählbar	wählbar	wählbar	automatisch	automatisch
Geschwindigkeitsbegrenzte Polynom 5. Ordnung	wählbar	wählbar	wählbar	automatisch	automatisch
Beschleunigungsbegrenzte Bewegung	wählbar	automatisch	wählbar	wählbar	automatisch
Beschleunigungsbegrenzte Sinoidenprofil	wählbar	automatisch	wählbar	wählbar	automatisch
Ruckbegrenzte Bewegung	wählbar	automatisch	wählbar	wählbar	wählbar

Tab. 14-38: Erweiterte Bewegungsgesetze

Das Profil für **"Geschwindigkeitsbegrenzte Polynom 5. Ordnung"** kann wie folgt beschrieben werden:

Das in der VDI-Richtlinie beschriebene Polynom 5. Ordnung ermöglicht keine Kontrolle der maximalen Verfahrensgeschwindigkeit. Zur Lösung dieses Problems wird das Bewegungsgesetz in drei Sektionen aufgeteilt. Die mittlere Sektion entspricht einer Geraden mit der geforderten Geschwindigkeit. Umgeben wird diese von zwei Polynomen 5. Grades. Diese sorgen für einen glatten Übergang von der Anfangs- bzw. Endgeschwindigkeit auf die Verfahrensgeschwindigkeit. Zu ermitteln sind die Schnittpunkte zwischen der ersten und zweiten sowie der zweiten und dritten Sektion. Die Punkte werden so gewählt, dass sich für die beiden Polynome kein Wendepunkt einstellt.

Basisbibliotheken

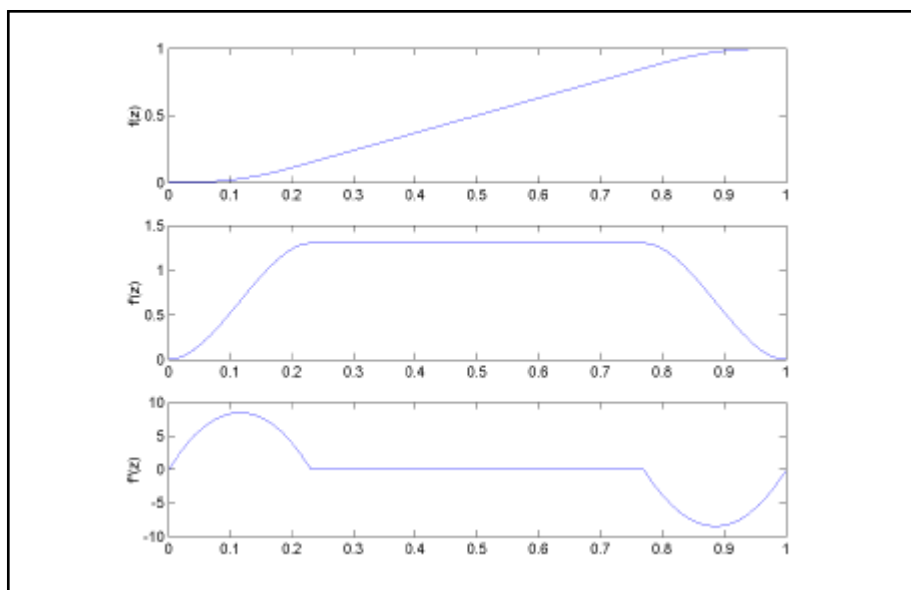


Abb. 14-42: Geschwindigkeitsbegrenztes Polynom 5. Ordnung

Das Profil für **"Überschwingungsfreies Polynom 5. Ordnung"** kann wie folgt beschrieben werden:

Das hier gezeigte Bewegungsgesetz erkennt und verhindert die mögliche Polynom-Überschwingung. Wie diese verändert wird, hängt dabei von der Kombination der normierten Geschwindigkeitsrandwerte ab. Es ergeben sich vier Fälle. Für Kombinationen, in denen die normierte Anfangsgeschwindigkeit " v_0 " und die normierte Endgeschwindigkeit " v_1 " unterschiedliche Vorzeichen haben, kann dieses Bewegungsgesetz nicht verwendet werden. Dies würde bedeuten, dass die Bewegungsbahn in einem Randpunkt einen Knick hätte, wenn sie den normierten Bereich von $0 \leq z \leq 1$ und $0 \leq f(z) \leq 1$ nicht verlassen dürfte.

- Fall 1:

$$V_0 \leq 1 \text{ und } V_1 \leq 1$$

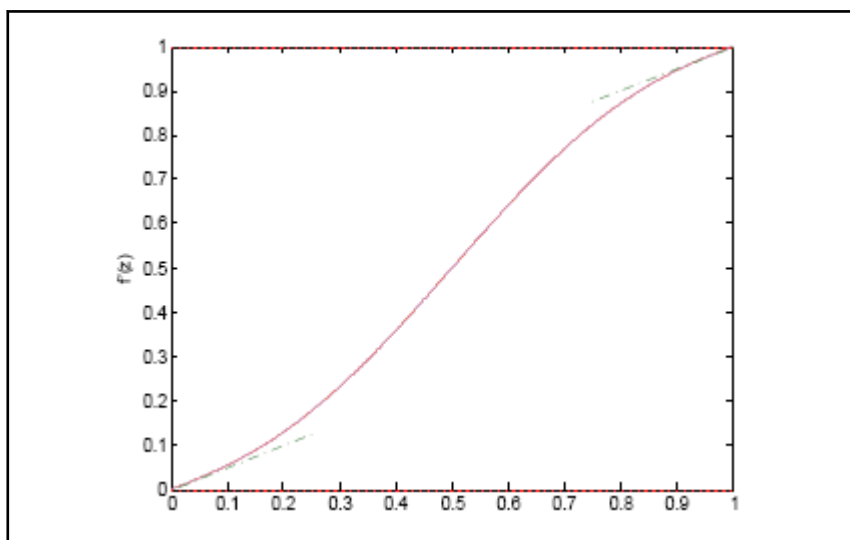


Abb. 14-43: Fall 1

- Fall 2:

$$V_0 \leq 1 \text{ und } V_1 > 1$$

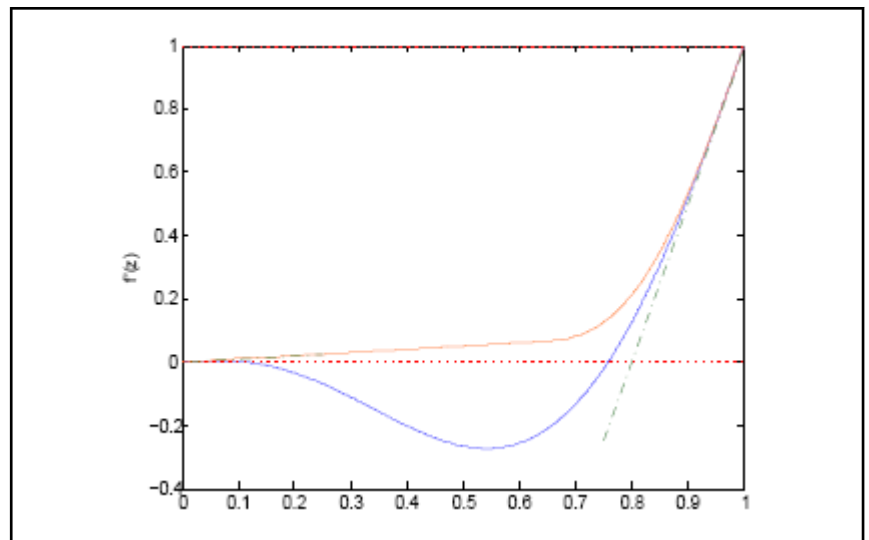


Abb. 14-44: Fall 2

- Fall 3:
 $V_0 > 1$ und $V_1 \leq 1$

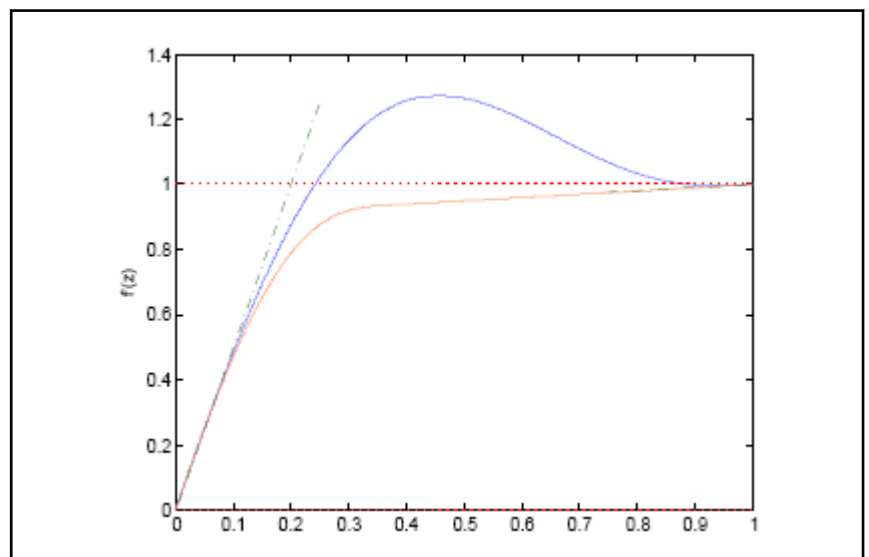


Abb. 14-45: Fall 3

- Fall 4:
 $V_0 > 1$ und $V_1 > 1$

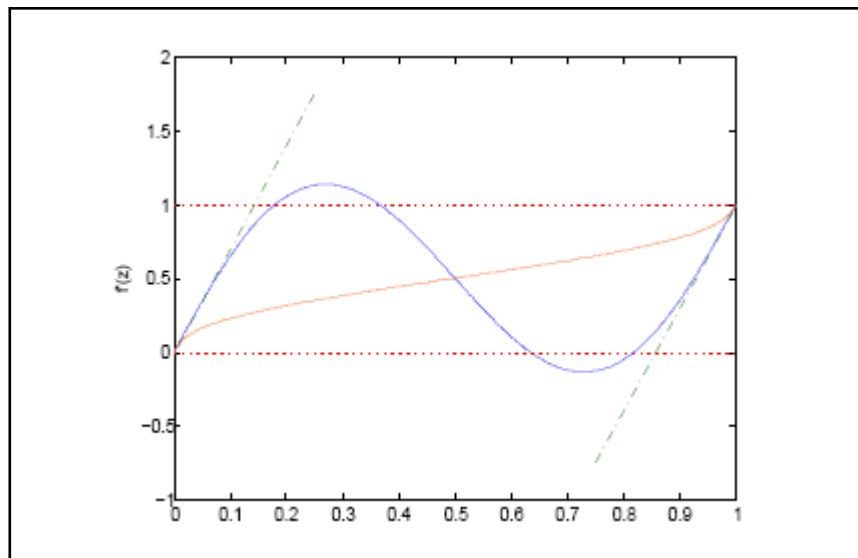


Abb. 14-46: Fall 4

Das Profil für die **"Beschleunigungsbegrenzte Bewegung"** kann wie folgt beschrieben werden:

Als einfachste Art der Bahninterpolation kann das Trapezprofil, oft auch Rampenprofil genannt, gesehen werden. Soll eine Wegstrecke "s" verfahren werden, so lässt sich hierbei der Bahnverlauf in drei Phasen aufteilen. In eine erste Phase, in der mit einem vorgegebenen Beschleunigungswert auf eine vorgegebene Verfahrgeschwindigkeit beschleunigt wird, in eine zweite Phase konstanter Geschwindigkeit und schließlich in eine Bremsphase in der ebenfalls mit einer angegebenen Verzögerung wieder abgebremst wird. Aufgabe der Bahninterpolation ist es, aus diesen vorgegebenen Beschleunigungswerten a_1 und a_3 sowie der gewünschten Geschwindigkeit v_m und der zurückzulegenden Wegstrecke "s", die Start- und Endzeitpunkte t_1 , t_2 und t_3 bzw. die einzelnen Verfahrzeiten T_1 , T_2 und T_3 zu berechnen. In der hier vorgestellten Variante wird außerdem auch eine geforderte Startgeschwindigkeit v_0 sowie Endgeschwindigkeit v_3 berücksichtigt.

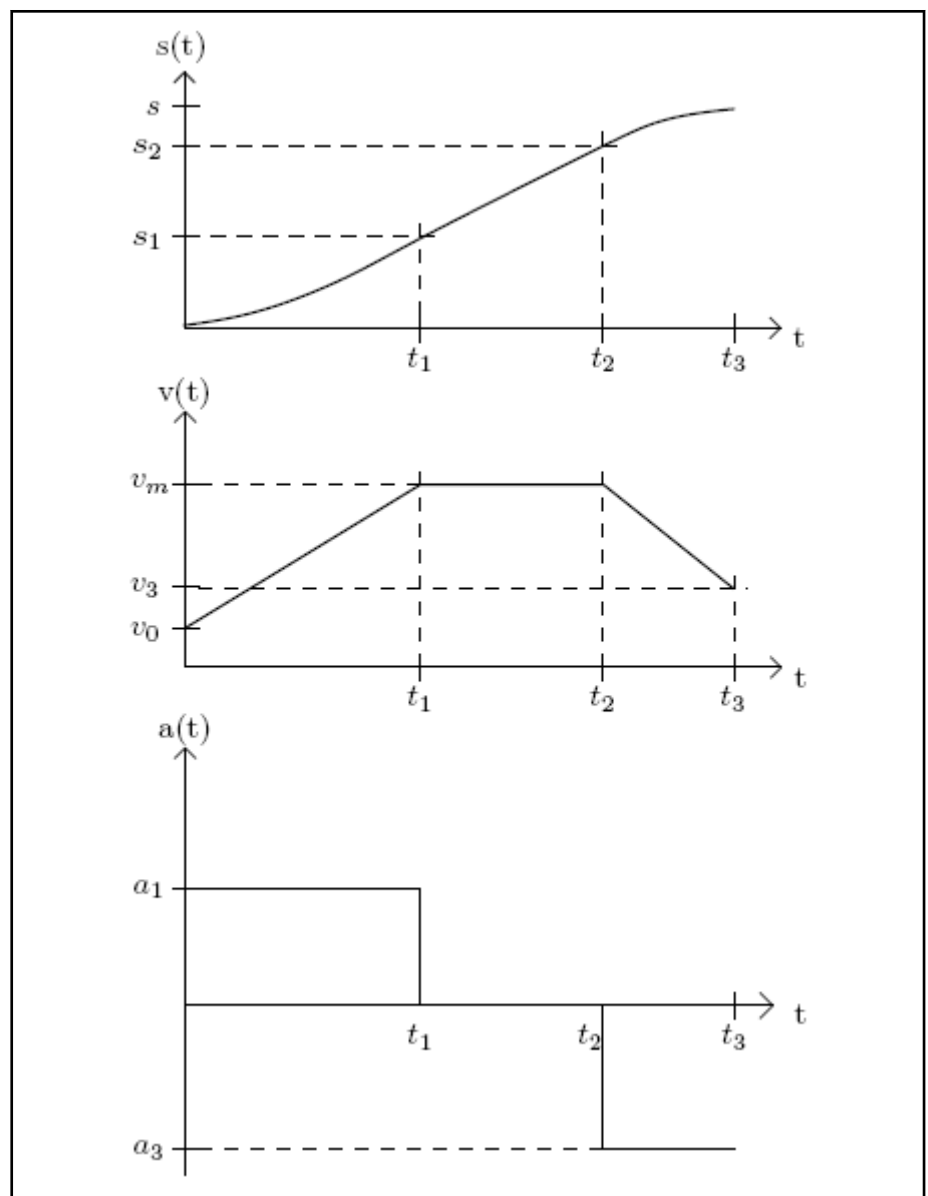


Abb. 14-47: Beschleunigungsbegrenzte Bewegung

Die möglichen Randwerte für das Bewegungsgesetz sind:

Randwert	z=0	z=1
1. Ordnung (Geschwindigkeit)	wählbar	wählbar
2. Ordnung (Beschleunigung)	automatisch	automatisch
3. Ordnung (Ruck)	automatisch	automatisch

Tab. 14-39: Randwerte für Bewegungsgesetz - Beschleunigungsbegrenzte Bewegung

Das Profil für die **"Beschleunigungsbegrenzte Bewegung mit Sinoidenprofil"** kann wie folgt beschrieben werden:

Beim Sinoidenprofil kann die Start- und Endgeschwindigkeiten ungleich 0 sein, sowie unterschiedliche Beschleunigungsgrößen a_1 und a_3 für die Beschleunigungs- und Bremsphase haben. Weiterhin werden auch die Sonderfälle berücksichtigt, in denen die Startgeschwindigkeit v_0 bereits

Basisbibliotheken

größer als die gewünschte Verfahrensgeschwindigkeit v_m ist. Gleiches gilt auch für die Endgeschwindigkeit v_3 .

Die möglichen Randwerte für das Bewegungsgesetz sind:

Randwert	z=0	z=1
1. Ordnung (Geschwindigkeit)	wählbar	wählbar
2. Ordnung (Beschleunigung)	automatisch	automatisch
3. Ordnung (Ruck)	automatisch	automatisch

Tab. 14-40: Randwerte für Bewegungsgesetz - Beschleunigungsbegrenzte Bewegung mit Sinoidenprofil

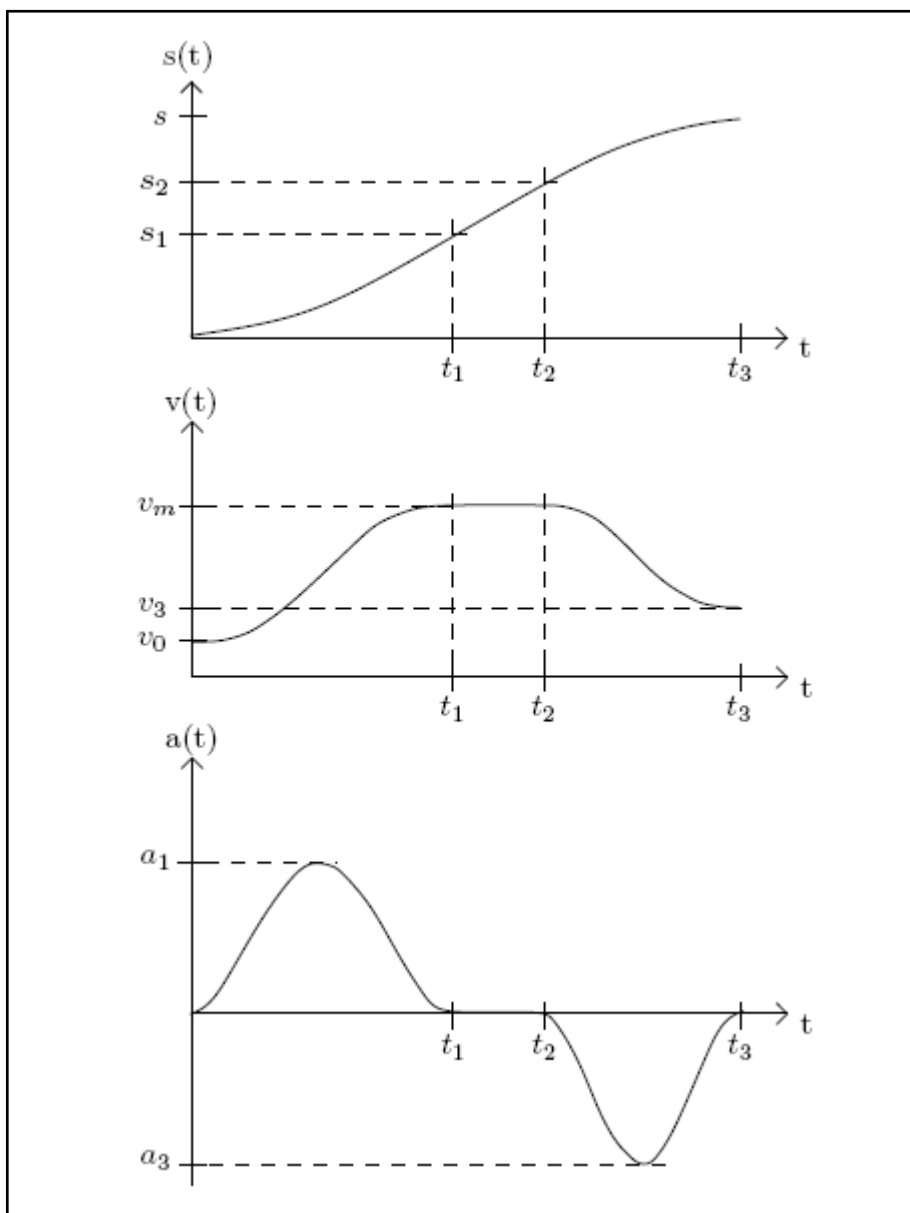


Abb. 14-48: Beschleunigungsbegrenzte Bewegung mit Sinoidenprofil

Das Profil für die **"Ruckbegrenzte Bewegung"** kann wie folgt beschrieben werden:

Für viele anspruchsvollere Bewegungsaufgaben gilt die Einhaltung einer Beschleunigung mit Ruckbegrenzung. Beispielsweise für den Transport von Flüssigkeiten oder aber auch bei leicht schwingungsfähigen Maschinen.

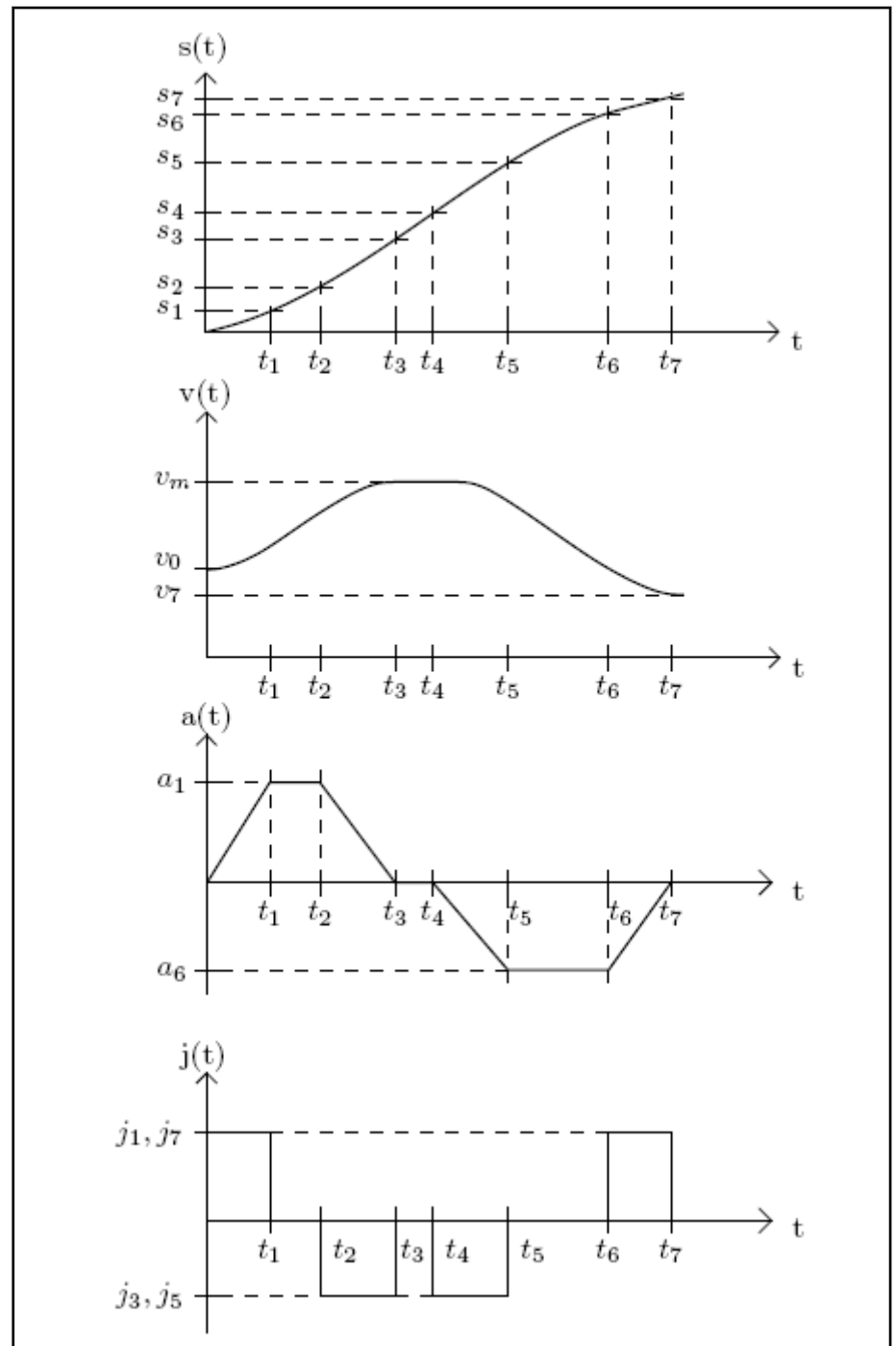


Abb. 14-49: Ruckbegrenzte Bewegung

14.2 RIL_HMI_Utilities

Die Bibliothek "RIL_HMI_Utilities" enthält Funktionsbausteine, die die HMI-Geräte unterstützen.

Die Bibliothek wird von mehreren Zielsystemen unterstützt und daher in der Dokumentation "Basis-Bibliotheken, IndraLogic 2G" beschrieben.

14.3 RIL_LoopControl

Die Bibliothek "RIL_LoopControl" stellt Grundelemente und Regler der Regelungstechnik zur Verfügung.

Um die Bibliothek nutzen zu können müssen die Bibliotheken "RIL_CommonTypes" und "RIL_Uutilities" in das Projekt eingebunden werden.

Die Bibliothek wird von mehreren Zielsystemen unterstützt und daher in der Dokumentation "Basis-Bibliotheken, IndraLogic 2G" beschrieben.

14.4 RIL_ModbusTCP

Das Feldbus-Protokoll "Modbus TCP" dient dem Datenaustausch zwischen Ethernet-Geräten, die das Modbus-Protokoll unterstützen.

Die antriebsinterne SPS übernimmt hierbei die Rolle des "Modbus TCP"-Servers (Master), mit dem sich die "Modbus TCP"-Clients (Slave) verbinden können.

Mit Hilfe der in der Bibliothek "RIL_ModbusTCP" enthaltenen Funktionsbausteine, können sowohl ein "Modbus TCP"-Server als auch ein "Modbus TCP"-Client realisiert werden.

Die Bibliothek wird von mehreren Zielsystemen unterstützt und daher in der Dokumentation "Basis-Bibliotheken, IndraLogic 2G" beschrieben.

14.5 RIL_SercosIII

14.5.1 Allgemeines

Die Bibliothek "RIL_SercosIII" ist die Schnittstelle zwischen SPS-Programmierungsumgebung und Sercos III-Teilnehmern. Über diese Schnittstelle können zum Beispiel die Sercos III-IO-Teilnehmer diagnostiziert werden oder azyklisch Sercos Parameter ausgetauscht werden.

Die Bibliothek wird von mehreren Zielsystemen unterstützt und daher in der Dokumentation "Feldbus-Bibliotheken" beschrieben.

14.6 RIL_SocketComm

Die Bibliothek "RIL_SocketComm" enthält Funktionen und Funktionsbausteine, um die TCP-/UDP-Kommunikation nutzen zu können.

Die Bibliothek wird von mehreren Zielsystemen unterstützt und daher in der Dokumentation "Basis-Bibliotheken, IndraLogic 2G" beschrieben.



Bedingt durch das Betriebssystem der Antriebe können die Funktion "IL_Ping" und der Funktionsbaustein "IL_PingAsync" nicht genutzt werden. Diese bilden das ICMP-Protokoll nach und kollidieren mit der Realisierung des ICMP-Protokolls im Betriebssystem. Als Ersatz kann die Funktion "SysSockPing" aus der Bibliothek "SysSocket" genutzt werden, welche den Ping synchron an das Betriebssystem weiterleitet.

14.7 RIL_Uutilities

Die Bibliothek "RIL_Uutilities" enthält Funktionen und Funktionsbausteine für die Konvertierung und Beeinflussung verschiedener Datentypen.

Die Bibliothek wird von mehreren Zielsystemen unterstützt und daher in der Dokumentation "Basis-Bibliotheken, IndraLogic 2G" beschrieben.

14.8 RMB_SercosIII_Util

Die Bibliothek "RMB_SercosIII_Util" stellt folgende Funktionen zur Verfügung:

- Funktionen zur vereinfachten Achsansteuerung von Antrieben über den Sercos Bus.

Hinweis: Für die Achsansteuerung über den Sercos Bus muss der Antrieb in der Steuerung als SPS-Gerät angelegt werden.

- Funktionen zum Betreiben eines Frequenzumrichters des Typs "EFC" in der Betriebsart "Frequenzsteuerung".

Die Bibliothek wird von mehreren Zielsystemen unterstützt und daher in der Dokumentation "Feldbus-Bibliotheken" beschrieben.

15 Systembibliotheken

15.1 Allgemeines

Die nachfolgend aufgeführten Systembibliotheken werden von Rexroth IndraMotion MLD ab MPx-18 (IndraDrive) / HDx-18 (HydraulicDrive) unterstützt.

HINWEIS

Sachschäden durch das Einbinden von nicht unterstützten Bibliotheken.

Es dürfen nur Bibliotheken eingebunden werden, die hier aufgeführt sind.

15.2 Standard

Die Bibliothek "Standard.library" wird standardmäßig mit dem IndraLogic 2G-Entwicklungssystem bereitgestellt.

Sie enthält alle Funktionen und Funktionsbausteine, die von der IEC61131-3 als Standardbausteine für ein IEC-Programmiersystem gefordert werden. Der Unterschied zwischen einer Standardfunktion und einem Operator ist, dass der Operator implizit dem Programmiersystem bekannt ist, während die Standardbausteine als Bibliothek an das Projekt gebunden werden müssen.

Die Bibliothek wird von mehreren Zielsystemen unterstützt und daher in der Dokumentation "SPS-System-Bibliotheken, IndraLogic 2G" beschrieben.

15.3 Util

Die Bibliothek "Util.library" enthält Funktionsbausteine, die für BCD-Konvertierung, Bit/Byte-Funktionen, mathematische Hilfsfunktionen, als Regler, Signalgeneratoren, Funktionsmanipulatoren und zur Analogwertverarbeitung verwendet werden können.

Die Bibliothek wird von mehreren Zielsystemen unterstützt und daher in der Dokumentation "SPS-System-Bibliotheken, IndraLogic 2G" beschrieben.

15.4 SysDir

15.4.1 Allgemeines

Mit den Funktionen der Bibliothek "SysDir" kann synchron auf ein Datei-Verzeichnissystem auf dem Zielsystem zugegriffen werden. Verzeichniseinträge können gelesen werden.

Die Bibliothek kann über den Bibliotheksverwalter hinzugefügt werden (Kategorie "System"- "SysLibs").



Einschränkung: Die Funktionen "SysDirGetCurrent" und "SysDirSetCurrent" werden nicht unterstützt.

Die Bibliothek wird von mehreren Zielsystemen unterstützt und daher in der Dokumentation "SPS-System-Bibliotheken, IndraLogic 2G" beschrieben.

15.4.2 Verwendungshinweise

Allgemeines

Es ist nur ein Zugriff auf das Verzeichnis "User" auf der externen Speicherkarte möglich. Ein Zugriff aus dem Anwenderprogramm auf das Stammverzeichnis wird umgeleitet auf das Verzeichnis "User".

Systembibliotheken

Beim Listen der Verzeichnisse wird nur der Inhalt des "User"-Verzeichnisses und ggf. darin liegende Unterordner angezeigt. Das Verzeichnis "User" selbst wird nicht angezeigt, so als wäre "User" eine eigene Partition. Zugriffe auf den Root-Knoten aus Anwendersicht können ohne Verzeichnisangabe, mit einfachem "\" und mit doppelten "\\" (Windows-kompatibel) erreicht werden.

Laufzeitverhalten

Wenn synchrone Dateisystemzugriffe in einer zyklischen MLD-Task verwendet werden, kann unter Umständen das Zeitverhalten dieser Task verändert werden. Daher ist es ratsam die Dateisystemzugriffe aus einer separaten, freilaufenden Task durchzuführen.

15.5 SysFile

15.5.1 Allgemeines

Mit den Funktionen der Bibliothek "SysFile" werden **synchrone** Dateizugriffe unterstützt. "Synchroner Dateizugriff" heißt, dass eine Funktion aufgerufen wird, und erst nach Beendigung der Ausgabe zurückkehrt.

Mit den Funktionen der Bibliotheken können Dateien

- geöffnet,
- geschlossen,
- gelöscht und
- umbenannt werden.
- Außerdem ist es möglich Daten in Dateien zu schreiben oder Daten aus Dateien zu lesen.
- Weitere Funktionen dienen
 - dem Ermitteln der Dateigröße oder
 - des Zeitpunkts des letzten Zugriffs, wie auch
 - dem Feststellen bzw. Verändern des aktuellen Offsets.

Die Bibliothek wird von mehreren Zielsystemen unterstützt und daher in der Dokumentation "SPS-System-Bibliotheken, IndraLogic 2G" beschrieben.

15.5.2 Verwendungshinweise

Allgemeines

Es ist nur ein Zugriff auf das Verzeichnis "User" auf der externen Speicherkarte möglich. Ein Zugriff aus dem Anwenderprogramm auf das Stammverzeichnis wird umgeleitet auf das Verzeichnis "User".

Beim Listen der Verzeichnisse wird nur der Inhalt des "User"-Verzeichnisses und ggf. darin liegende Unterordner angezeigt. Das Verzeichnis "User" selbst wird nicht angezeigt, so als wäre "User" eine eigene Partition. Zugriffe auf den Root-Knoten aus Anwendersicht können ohne Verzeichnisangabe, mit einfachem "\" und mit doppelten "\\" (Windows-kompatibel) erreicht werden.

Laufzeitverhalten

Wenn synchrone Dateisystemzugriffe in einer zyklischen MLD-Task verwendet werden, kann unter Umständen das Zeitverhalten dieser Task verändert werden. Daher ist es ratsam die Dateisystemzugriffe aus einer separaten, freilaufenden Task durchzuführen.

15.6 SyslecTasks23

Aus der Bibliothek "SyslecTasks23" darf nur die Funktion "SysIECTaskGetConfig" verwendet werden; alle anderen Funktionen werden von IndraMotion MLD 2G nicht unterstützt und dürfen nicht verwendet werden.

Mit Hilfe der Funktion "SysIECTaskGetConfig" kann die Task-Konfiguration ausgelesen werden.

Die Tasks werden der Priorität nach aufgelistet, die Zykluszeit wird in Millisekunden angezeigt.

15.7 SysMem

Die Bibliothek "SysMem" dient der Speicherverwaltung.

Die Bibliotheksfunktionen können verwendet werden, um Speicherplatz zu reservieren (allokieren) und wieder freizugeben, um Speicher zu definieren, um zwei Speicherbereiche zu vergleichen, um Daten von einem Speicher zum anderen zu kopieren oder zu verschieben, oder um zu swappen.

Die Abarbeitung erfolgt synchron.

Die Bibliothek wird von mehreren Zielsystemen unterstützt und daher in der Dokumentation "SPS-System-Bibliotheken, IndraLogic 2G" beschrieben.

15.8 SysCallback23

Die Bibliothek "SysCallback23" enthält Funktionen, um definierte Callback-Funktionen für Laufzeitergebnisse zu aktivieren.

Die Bibliothek wird von mehreren Zielsystemen unterstützt und daher in der Dokumentation "SPS-System-Bibliotheken, IndraLogic 2G" beschrieben.

15.9 SysSem

Die Bibliothek "SysSem" bietet Funktionen um Semaphore für die Synchronisation von Tasks zu erzeugen und zu benutzen. Semaphore dienen dazu, den gleichzeitigen Zugriff auf kritische Daten zu verhindern, die von mehreren Tasks verwendet werden.

Die Abarbeitung erfolgt synchron.



Bei einem SPS-Reset werden keine Semaphore freigegeben. Der Anwender muss selbst darauf achten, die Semaphore wieder freizugeben.

Die Bibliothek wird von mehreren Zielsystemen unterstützt und daher in der Dokumentation "SPS-System-Bibliotheken, IndraLogic 2G" beschrieben.

15.10 SysSockets

Die Bibliothek "SysSockets" bietet Funktionen, mit denen der Zugriff auf Sockets zur Kommunikation über TCP/IP und UDP unterstützt wird.

Bitte überprüfen Sie zunächst, ob Ihre Anforderungen durch die Bibliothek "RIL_SocketComm" erfüllt werden können.

Wenn das Zielsystem es unterstützt, können über die Bibliotheksfunktionen die entsprechenden Funktionen des Betriebssystems aufgerufen werden. Für detaillierte Information zu den Betriebssystemfunktionen lesen Sie bitte die entsprechende Hilfe.

Bitte beachten:

- Das Verhalten der Funktionen kann zielsystemabhängig unterschiedlich sein.
- Das Öffnen/Schließen von Sockets kann viel Zeit beanspruchen, wenn viele Sockets gleichzeitig geöffnet/geschlossen werden.

Systembibliotheken



Die Bibliothek "SysSockets" ist nicht kompatibel zur Bibliothek "SysLibSockets" der IndraMotion MLD 1G.

Die Bibliothek "SysSockets23" stellt zwar das Programmierinterface der Bibliothek "SysLibSockets" der IndraMotion MLD 1G zur Verfügung, die Funktionalität ist aber nicht vollständig abgedeckt. **Diese Bibliothek darf daher nicht verwendet werden!**

Die Bibliothek wird von mehreren Zielsystemen unterstützt und daher in der Dokumentation "SPS-System-Bibliotheken, IndraLogic 2G" beschrieben.

15.11 SysStr23

15.11.1 Übersicht

Strings vergleichen

Die Bibliothek "SysStr23" bietet Funktionen für den Umgang mit Strings.

- [SysStrCmp, Seite 242](#)
- [SysStrCmpl, Seite 242](#)
- [SysStrCmpN, Seite 243](#)
- [SysStrCmpNI, Seite 243](#)

Strings kopieren

[SysStrCpy, Seite 243](#)

Stringlänge ermitteln

[SysStrLen, Seite 244](#)

Die Abarbeitung der Funktionen erfolgt synchron.

15.11.2 SysStrCmp

Mit der Funktion "SysStrCmp" vom Typ DINT kann ein lexikographischer Vergleich zweier Strings durchgeführt werden.

Input-Variable	Datentyp	Beschreibung
sString1	STRING	erster String
sString2	STRING	zweiter String

Die Funktion liefert eines der folgenden Ergebnisse:

Rückgabewert < 0	String1 kleiner als String2
Rückgabewert = 0	String1 = String2
Rückgabewert > 0	String1 größer als String2

15.11.3 SysStrCmpl

Mit der Funktion "SysStrCmpl" vom Typ DINT kann überprüft werden, ob zwei Strings identisch sind.

Input-Variable	Datentyp	Beschreibung
sString1	STRING	erster String
sString2	STRING	zweiter String

Die Funktion liefert eines der folgenden Ergebnisse:

Rückgabewert < 0	String1 kleiner als String2
Rückgabewert = 0	String1 = String2
Rückgabewert > 0	String1 größer als String2

15.11.4 SysStrCmpN

Mit der Funktion "SysStrCmpN" vom Typ DINT kann die Größe zweier Strings verglichen werden, wobei eine definierte Stellenzahl ab dem Stringanfang betrachtet wird.

Input-Variable	Datentyp	Beschreibung
sString1	STRING	erster String
sString2	STRING	zweiter String
diChars	DINT	Anzahl der Stellen, für die - von Stringbeginn an gezählt -, die Größe der Strings verglichen werden soll

Die Funktion liefert eines der folgenden Ergebnisse:

Rückgabewert < 0	String1 kleiner als String2
Rückgabewert = 0	String1 = String2
Rückgabewert > 0	String1 größer als String2

15.11.5 SysStrCmpNI

Mit der Funktion "SysStrCmpNI" vom Typ DINT kann verglichen werden, ob die Anfangsbereiche zweier Strings identisch sind. Die Anzahl der Stellen, für die von Stringbeginn an gezählt wird, muss mit "diChars" vorgegeben werden.

Input-Variable	Datentyp	Beschreibung
sString1	STRING	erster String
sString2	STRING	zweiter String
diChars	DINT	Anzahl der Stellen, für die von Stringbeginn an gezählt, geprüft wird, ob der Inhalt bei beiden Strings identisch ist

Die Funktion liefert eines der folgenden Ergebnisse:

Rückgabewert < 0	String1 kleiner als String2
Rückgabewert = 0	String1 = String2
Rückgabewert > 0	String1 größer als String2

15.11.6 SysStrCpy

Mit der Funktion "SysStrCpy" vom Typ DWORD kann ein String (sString2) auf einen anderen String (sString1) kopiert werden.

Input-Variable	Datentyp	Beschreibung
sString1	STRING	String, auf den kopiert werden soll (Ziel)
sString2	STRING	String, der kopiert werden soll (Quelle)

Als Rückgabewert erhält man einen Pointer auf den Ziel-String (sString1).

Systembibliotheken

15.11.7 SysStrLen

Mit der Funktion "SysStrLen" vom Typ DINT kann die Länge eines Strings ermittelt werden.

Input-Variable	Datentyp	Beschreibung
sString1	STRING	String, dessen Länge geprüft werden soll

Als Rückgabewert erhält man die Anzahl der Stellen, wobei "terminal NULL" nicht mitgezählt wird.

15.12 SysTimeRTC.library

15.12.1 Einleitung

Ab der Firmware-Version 20 werden folgende Funktionen der Systembibliothek "SysTimeRTC.library" unterstützt:

- SysTimeRtcGet
- SysTimeRtcSet
- SysTimeRtcHighResGet
- SysTimeRtcHighResSet

Mit Hilfe der Funktionen kann die Systemzeit des Geräts im IEC 61588-Format ausgelesen bzw. beschrieben werden ("S-0-1305.0.1, Systemzeit").

15.12.2 SysTimeRtcGet

Kurzbeschreibung Mit dieser Funktion kann die aktuelle RTC-Systemzeit des Geräts in **Sekunden** gelesen werden.

Schnittstellenbeschreibung

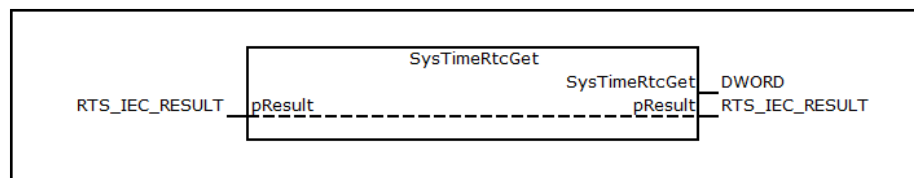


Abb. 15-1: FUN SysTimeRtcGet

Ausgangs-Variable	Datentyp	Beschreibung
SysTimeRtcGet	DWORD	RTC-Zeit in Sekunden seit 01.01.1970 00:00:00.

Eingangs-/Ausgangs-Variable (VAR_IN_OUT)	Datentyp	Beschreibung
pResult	RTS_IEC_RESU LT	Mit "ERR_OK" (16#0000) wird der Erfolg der Operation gemeldet

15.12.3 SysTimeRtcSet

Kurzbeschreibung Mit dieser Funktion kann die aktuelle RTC-Systemzeit des Geräts in **Sekunden** gesetzt werden.

Schnittstellenbeschreibung

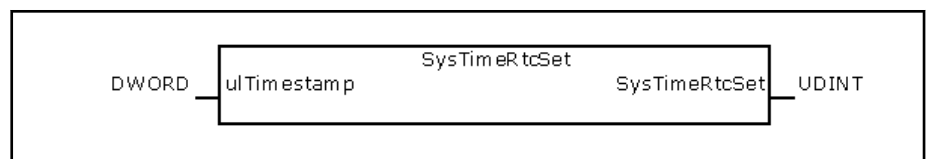


Abb. 15-2: FUN SysTimeRtcSet

Eingangs-Variable	Datentyp	Beschreibung
ulTimestamp	DWORD	RTC-Zeit in Sekunden seit 01.01.1970 00:00:00

Ausgangs-Variable	Datentyp	Beschreibung
SysTimeRtcSet	UDINT	ERR_OK (16#0000): Ok ERR_FAILED 16#0001): Allgemeiner interner Fehler

15.12.4 SysTimeRtcHighResGet

Kurzbeschreibung Mit dieser Funktion kann die aktuelle RTC-Systemzeit des Geräts in **Millisekunden** gelesen werden.

Schnittstellenbeschreibung

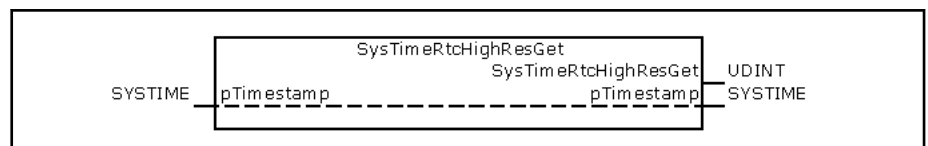


Abb. 15-3: FUN SysTimeRtcHighResGet

Ausgangs-Variable	Datentyp	Beschreibung
SysTimeRtcHighResGet	UDINT	Mit "ERR_OK" (16#0000) wird der Erfolg der Operation gemeldet

Eingangs-/Ausgangs-Variable (VAR_IN_OUT)	Datentyp	Beschreibung
pTimestamp	SYSTIME	RTC-Zeit in Millisekunden seit 01.01.1970 00:00:00.

15.12.5 SysTimeRtcHighResSet

Kurzbeschreibung Mit dieser Funktion kann die aktuelle RTC-Systemzeit des Geräts in **Millisekunden** gesetzt werden.

Schnittstellenbeschreibung

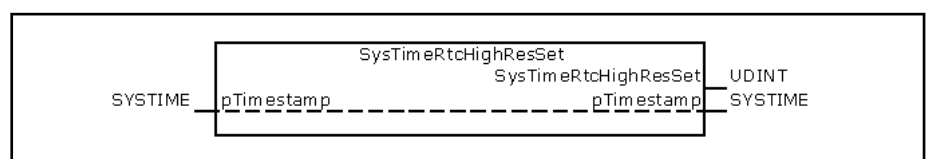


Abb. 15-4: FUN SysTimeRtcHighResSet

Eingangs-/Ausgangs-Variable VAR_IN_OUT	Datentyp	Beschreibung
pTimestamp	SYSTIME	Zeit in Millisekunden seit 01.01.1970 00:00:00, die gesetzt werden soll

Ausgangs-Variable	Datentyp	Beschreibung
SysTimeRtcHighResSet	UDINT	Mit "ERR_OK" (16#0000) wird der Erfolg der Operation gemeldet

15.13 IoDrvEtherNetIP.library

Stand der Beschreibung/Verfügbarkeit der Bibliothek:

- Firmware: MPC-20V12
- Engineering: IndraWorks MLD 14V16
- MLD-Projektierung / Compiler: CoDeSys 3.5.6.40
- Bibliothek: IoDrvEtherNetIP.library 3.5.7.0

Mit der Bibliothek "IoDrvEtherNetIP.library" wird ein EtherNet/IP-Scanner zur Verfügung gestellt. Mit ihm ist es möglich andere EtherNet/IP-Geräte als E/A-Slaves einzubinden (z. B. Barcode-Leser, RFID, Durchflussmesser, Laser-Entfernungsmesser, I/O, HMI,...).

Die Bibliothek "IoDrvEtherNetIP.library" wird mit der Applikation in die MLD geladen.

Die Konfiguration erfolgt über Standard-E/A-Mechanismen, indem dem Projekt der Scanner und die Adapter (E/A-Slaves) als Geräte hinzugefügt werden (siehe auch "[Projektierung des EtherNet/IP-Scanners](#)"). Scanner und Adapter sind in der Gerätedatenbank abgelegt. Fehlende Adapter können über EDS-Beschreibungsdateien einfach hinzugefügt werden. Es besteht auch die Möglichkeit einen Adapter über ein generisches, frei konfigurierbares Gerät zu projektieren.

Der EtherNet/IP-Scanner hat auf Grund seiner Realisierung im IEC-Code und den technischen Eigenschaften des Antriebs einen eingeschränkten Einsatzbereich, der bei der Projektierung eines EtherNet/IP-Scanners zu berücksichtigen ist.



Der EtherNet/IP-Scanner läuft im Kontext einer Task der MLD-Applikation, die im folgenden als Buszyklustask bezeichnet wird. Er kann daher die Anforderungen an harte Echtzeit nicht erfüllen. Auch ist durch den EtherNet/IP-Scanner eine erkennbare Belastung der Zykluszeit und ein erhöhter Speicherbedarf bei der Projektierung der MLD zu berücksichtigen.

In den nachfolgenden Kapiteln sind zunächst die Merkmale beschrieben, die im Kontext der MLD zu berücksichtigen sind, gefolgt von einer Übersicht über die Editoren, die für die Projektierung des EtherNet/IP-Scanners und der Adapter zur Verfügung gestellt werden.

15.13.1 Lizenzierung und Betrieb

Der EtherNet/IP-Scanner benötigt eine Lizenz, die in binärer Form in der Firmware hinterlegt ist und über den Geräteschlüssel (Funktionspaketfreischaltung) "MLD Advanced with EtherNet/IP" freigeschaltet wird (Ausführung "ME").



Wenn die Lizenz fehlt, läuft der Stack ½ Stunde bevor er stoppt. Zum Starten (Laden der Testlizenz) muss an der MLD ein Reset "Warm" ausgeführt werden.

In Geräten, in denen der Betrieb des EtherNet/IP-Scanner generell nicht unterstützt wird, wird ein Fehler bzgl. unaufgelöster Referenzen beim Download generiert, sobald ein EtherNet/IP-Scanner in der MLD-Applikation projiziert ist.

15.13.2 Performance und Zykluszeit der MLD

Die Verarbeitung der zyklischen Daten belastet die Buszyklustask der MLD typisch mit etwa 2 ms bei 4 gleichzeitig betriebenen EtherNet/IP-Adaptern. Bei hoher Netzwerklast kann die Verarbeitung in der Buszyklustask sporadisch um bis zu 5 ms zusätzlich belastet werden.

- Die zyklischen Daten sollten nicht schneller als mit einer RPI=10 ms konfiguriert werden.
- Bei einer RPI=10 ms sollte die Buszyklustask ausschließlich für die EtherNet/IP-Kommunikation genutzt werden.

Bei der Konfiguration des Watchdogs für die Buszyklustask ist in der MLD die sporadische Belastung der Buszykluszeit durch Netzwerkspitzen zu berücksichtigen. Zudem wird empfohlen den Watchdog erst nach der Initialisierung des EtherNet/IP-Scanners über `IecTaskEnableWatchdog()` einzuschalten.

15.13.3 Ressourcenbedarf im Bootprojekt

Da Daten und Code des EtherNet/IP-Scanners und der unterlagerten Adapter im IEC-Code abgelegt werden, ist der dadurch verursachte Verbrauch an Ressourcen bei der Projektierung einer Anlage zu berücksichtigen. Neben Code- und Datenspeicher werden für die IP-basierte Kommunikation zudem Sockets belegt.

Der Ressourcenbedarf kann aus folgenden Angaben abgeleitet werden:

EtherNet/IP-Scanner

- Code: ca. 300 kBytes
- Globale Daten: ca. 35 kBytes
- Dynamisch allozierter Speicher: ca. 5 kByte
- Sockets: 3

EtherNet/IP-Adapter

- Code: ca. 0,5 kBytes
- Globale Daten: ca. 8 kBytes
- Dynamisch allozierter Speicher: ca. 1,5 kByte
- Sockets: 1 (bei azyklischer Kommunikation)



Der Betrieb des EtherNet/IP-Scanners ist unabhängig von einer µSD-Karte möglich. Aufgrund des hohen Speicherbedarfs des EtherNet/IP-Scanners wird aber in der Regel eine µSD-Karte für das Bootprojekt der MLD erforderlich sein.

15.13.4 Unterstützte Schnittstellen und Koexistenz mit Sercos

Da der EtherNet/IP-Scanner auf der Socket-Bibliothek aufbaut, ist die Verwendung des Scanners prinzipiell nicht an eine bestimmte Ethernet-Schnittstelle gebunden. Die Wahl der Schnittstelle hat aber Auswirkung auf die Leistungsfähigkeit des Scanners, da die Ethernet-Schnittstellen unterschiedlich leistungsfähig sind. Von den am Gerät vorhandenen Ethernet-Schnittstellen eignet sich nur das Master Sercos Interface (X24 und X25) für die Anbindung von EtherNet/IP-Adaptern. Dies ist bei der Projektierung zu berücksichtigen.

Die relevanten Kenndaten des Master Sercos Interface sind:

Systembibliotheken

Schnittstelle	Verfügbare Speicher		Anzahl der pufferbaren IP-Nachrichten	
	Empfangspuffer	Sendepuffer	Empfangspuffer	Sendepuffer
Sercos Master	6 kByte (pro Port, segmentiert)	6 kByte (shared, segmentiert)	8 (pro Port)	8 (pro Port)

Tab. 15-1: Kenndaten des Master Sercos Interface



Die Anzahl der gleichzeitig betreibbaren EtherNet/IP-Adapter ist prinzipiell auf 8 pro Port (X24 und X25) eingeschränkt.

Eine gemeinsame Nutzung der Protokolle Sercos und EtherNet/IP ist nur in einer Linientopologie und unter einer der beiden folgenden Bedingungen möglich:

1. Es wird „Single Wire Coexistence“ (SWC, Support of Industrial Ethernet protocols via UCC) genutzt und die EtherNet/IP-Adapter werden am letzten freien Port der Sercos Slaves angekoppelt. SWC muss in den Einstellungen des CCD-Verbunds aktiviert werden.
2. Jedes Protokoll nutzt einen Port (X24 oder X25); d. h. es wird am Sercos Interface ein Port exklusiv für Sercos und ein Port exklusiv für EtherNet/IP genutzt.

Dies ist die Vorzugsvariante, da SWC nur von wenigen Sercos Teilnehmern unterstützt wird und besondere Anforderungen an die Verdrahtung stellt.

15.13.5 Anforderungen an das IP-Netzwerk

Wird ein Interface sowohl für EtherNet/IP als auch für andere IP-Protokolle genutzt (Engineering, Datentransfer usw.), so hat dies Auswirkungen auf die Verfügbarkeit der Sende- und Empfangspuffer der Schnittstelle für die EtherNet/IP-Kommunikation. Die Anzahl der gleichzeitig betreibbaren Adapter kann sich verringern. Die Einschränkungen sind vom Anwendungsfall abhängig und müssen bei der Evaluierung vom Anwender abgeschätzt werden. Folgende Effekte sind hierbei zu beachten:

1. Der EtherNet/IP-Kommunikation steht unter Umständen nur noch ein Teil der Sende- und Empfangspuffer der Schnittstelle zur Verfügung:
 - Bei Lastspitzen kann es dazu kommen, dass kein Empfangspuffer für den Telegrammempfang zur Verfügung steht, was zum Verlust der Daten führt.
 - Sporadische Lastspitzen können auftreten, weil die EtherNet/IP-Adapter untereinander nicht synchronisiert sind.
 - Damit die Daten in der Buszyklustask konzentriert gesendet werden können, müssen die Sendepuffer in der Schnittstelle entsprechend viele IP-Nachrichten aufnehmen können. Ist dies nicht der Fall, kann es zu Verzögerungen in der Buszyklustask kommen, wenn der Sendepuffer in der Schnittstelle durch andere Protokolle bereits belegt wurde.

Dadurch kann die Zykluszeit der Buszyklustask in der MLD ansteigen.

2. Aufgrund der Socket-basierten Realisierung des EtherNet/IP-Scanners nutzt dieser den im Antrieb vorhandenen TCP/IP-Stack. Durch konkurrierende Zugriffe, bedingt durch andere Protokolle, kann es hierbei zu Verzögerungen kommen. Insbesondere dann, wenn das hierbei über andere Protokolle transferierte Datenvolumen besonders

hoch ist. Hierbei ist auch zu berücksichtigen, dass alle Schnittstellen des Antriebs über einen gemeinsamen TCP/IP-Stack angebunden sind.

Dadurch kann die Zykluszeit der Buszyklustask in der MLD ansteigen.

15.13.6 Einschränkungen des Funktionsumfangs der Bibliothek

Folgende Einschränkungen im Funktionsumfang sind bei der Nutzung der Bibliothek "IoDrvEtherNetIP.library" im Kontext der MLD zu beachten:

1. Es können nicht mehr als 4 EtherNet/IP-Adapter projektiert und betrieben werden. Abhängig von der Nutzung des Netzwerks kann sich die Anzahl gleichzeitig betreibbarer Adapter weiter verringern. Details hierzu siehe "[Kap. 15.13.4 "Unterstützte Schnittstellen und Koexistenz mit Sercos" auf Seite 247](#)" und "[Kap. 15.13.5 "Anforderungen an das IP-Netzwerk" auf Seite 248](#)".



Über die Device Diagnose besteht die Möglichkeit, EtherNet/IP-Adapter bei laufender Steuerung vom Bus zu trennen oder mit dem Bus zu verbinden. Hierzu muss die Device Diagnose in den Einstellungen der SPS aktiviert werden, was dazu führt, dass die Bibliothek "CAA Device Diagnosis" angezogen wird. Weitere Details hierzu können der Beschreibung zur Bibliothek "CAA Device Diagnosis" entnommen werden.

2. Es werden nur Point-to-Point-Verbindungen unterstützt. Multicast-Nachrichten können vom Antrieb gesendet, aber nicht empfangen werden
3. Aus der Auswahl der Schnittstelle ergibt sich die IP-Adresse des Scanners, welche im Konfigurationseditor des Scanners eingetragen werden muss. Die IP-Adressen der Adapter müssen im Projekt ebenfalls passend eingestellt werden. Alle IP-Adressen müssen in den Einstellungen des EtherNet/IP-Scanners und den Adaptern manuell eingegeben werden.
4. Aufgrund der Einschränkungen und der Tatsache, dass ein EtherNet/IP-Scanner nicht zertifiziert werden muss, ist dieser im Kontext der MLD nicht zertifiziert.

HINWEIS

Die EtherNet/IP-Kommunikation ist nicht für eine sichere Kommunikation geeignet. Ein sporadischer Verlust von zyklischen Daten ist zu tolerieren. Aufgrund dessen sieht die EtherNet/IP-Spezifikation auch eine Timeout-Überwachung mit einem Time-Multiplikator von minimal 4 vor.

Wenn Sie die obigen Kenndaten beachten, ist ein sicherer Betrieb ohne Telegrammverluste zu erwarten.

15.13.7 Projektierung des EtherNet/IP-Scanners

Der EtherNet/IP-Scanner ist ein IEC-basierter I/O-Treiber und kann somit mit den Standard-Editoren für die Konfiguration der Ein- und Ausgänge projektiert werden.

Hinzufügen der Geräteknoten



Da der Ethernet-Knoten nur für Ethernet-basierte E/A-Protokolle erforderlich ist, ist dieser variabel und muss manuell hinzugefügt werden. Auch wird dieser nicht mit dem Standardprojekt angelegt.

Nachfolgend die Schritte, um einem MLD-Projekt einen EtherNet/IP-Scanner und ein EtherNet/IP-Target (bzw. einen EtherNet/IP-Adapter) hinzuzufügen:

Vorbedingung: Ein geeignetes MLD-Projekt (MPC20) ist angelegt.

1. Klicken Sie im Projektextplorer mit der rechten Maustaste auf den MLD-Knoten, um das Kontextmenü aufzurufen.

2. Wählen Sie **Hinzufügen ▶ Ethernet**.

Der Knoten "Ethernet (Ethernet)" wird unterhalb des MLD-Knotens eingefügt.

3. Rufen Sie an dem Knoten "Ethernet (Ethernet)" das Kontextmenü auf.

4. Wählen Sie **Hinzufügen ▶ Gerät hinzufügen ▶ MLD Ethernet/IP Scanner**.

Der Knoten "MLD_Ethernet_IP_Scanner (MLD Ethernet/IP Scanner)" wird unterhalb des MLD-Knotens eingefügt.



Die Bibliothek "IoDrvEtherNetIP.library" wird durch das Anlegen des Knotens für den EtherNet/IP-Scanners automatisch hinzugefügt. Die Version der Bibliothek ist hierbei durch die Version des Geräts festgelegt. Die Version kann nachträglich über den Bibliotheksplatzhalter angepasst werden.

5. Rufen Sie an dem Knoten "MLD_Ethernet_IP_Scanner (MLD Ethernet/IP Scanner)" das Kontextmenü auf.

6. Wählen Sie **Hinzufügen ▶ Gerät hinzufügen ▶ Communications Adapter**. Wählen Sie hier das gewünschte Gerät aus.

(In Vorbereitung: Einzelne Adapter können im Gerätebaum über die Selektionsboxen, die über die Maus unter dem Symbol des Geräteknotens aufgedeckt werden können, im Projekt aktiviert oder deaktiviert werden.)

Taskkonfiguration

Mit dem Anlegen des Scanner-Knotens werden jeweils eine Task für die zyklischen IO-Daten ("ENIPScannerIOTask") und eine Task für die azyklische Kommunikation mit Explicit Messages angelegt ("ENIPScannerServiceTask") (siehe auch "[Azyklische Kommunikation \(Explicit Messaging Service\)](#)"). Diese Tasks können ergänzt werden um eigene Aufrufe. Zudem ist es möglich die Tasks umzubenennen, womit letztendlich wie gewohnt bis zu 4 Tasks zur freien Verwendung in der Applikation zur Verfügung stehen.



Per Drag&Drop können die automatisch hinzugefügten Aufrufe aus den angelegten Tasks in andere Anwendertasks verschoben werden.

Werden die Tasks vom Anwender gelöscht, können diese nur wieder hergestellt werden, indem der Geräteknoten "MLD Ethernet/IP-Scanner" gelöscht und erneut angelegt wird.

IP-Adressierung

Basis für die Nutzung des EtherNet/IP-Scanners ist eine Schnittstelle (Port), über die eine Kommunikation über TCP und UDP (Unicast) möglich ist. Dies ist prinzipiell bei allen Ports des IndraDrive gegeben, aber nur für einen CCD-Port empfehlenswert. Aus der Auswahl des Ports ergibt sich die IP-Adresse des Scanners, welche im Konfigurationseditor des Scanners eingetragen werden muss. Die IP-Adressen der Adapter müssen im Projekt ebenfalls passend eingestellt werden. Alle IP-Adressen müssen in den Einstellungen des EtherNet/IP-Scanners und den Adaptern manuell eingegeben werden und dürfen daher im Betrieb nicht dynamisch vergeben werden.



Auch die Adapter müssen so konfiguriert sein, dass die IP-Adressen fest vergeben sind. "BOOTP" und "DHCP" sind in den Adaptern zu deaktivieren. Andernfalls kann es sein, dass zu den Adaptern keine Kommunikation möglich ist. Das Vorgehen hierzu ist der jeweiligen Dokumentation des Adapters zu entnehmen.



IP-Adresse des EtherNet/IP-Scanners einstellen: Siehe "[MLD EtherNet/IP-Scanner](#)"

IP-Adresse der EtherNet/IP-Adapter einstellen: Siehe "[EtherNet/IP-Target](#)"

E/A-Konfiguration

Das Vorgehen bei der E/A-Konfiguration entspricht dem Vorgehen bei anderen Ein- und Ausgängen. Die E/A-Adressen werden wie bei den Sercos-E/A im Abbild der Adapter beim Hinzufügen der Adapter automatisch vergeben. In den Eigenschaften des Adapters können den Ein- und Ausgängen in der Applikation definierte Variablen zugeordnet werden (Mapping) oder anderen E/A-Adressen zugewiesen werden.



Das Verhalten der Ausgänge im Stopp ist durch den Bus bestimmt. Bei EtherNet/IP sieht die Spezifikation vor, dass bei SPS-Stopp im Protokoll das RUN-IDLE-Bit auf IDLE (0) gesetzt wird. Der Adapter wertet dieses Bit aus und setzt abhängig von seiner Konfiguration die Ausgänge entsprechend dem Abbild oder nutzt eigene Ersatzwerte. Die SPS-Einstellung "E/A bei Stopp" bezieht sich somit auf das Abbild der Ausgänge und stimmt nicht zwangsläufig mit den Ausgängen an den einzelnen Adaptern überein.

Ist die Buszyklustask im Stopp ebenfalls gestoppt, findet keine Kommunikation mit dem Adapter statt und der Adapter setzt abhängig von seiner Konfiguration seine Ausgänge.



Es sind maximal 511 Byte pro Connection möglich. Da pro Adapter nur eine "Exclusive Owner Connection" (= Verbindung mit Ein- und Ausgängen) existieren kann, ist die maximale Größe der Ein- und Ausgänge auf 511 Byte je Adapter beschränkt. Hintergrund: Das Feld für die Verbindungsgröße ist nur 9 Bit groß. Die EtherNet/IP-Erweiterung "Large Forward Open" wird nicht unterstützt.

Anwenderparameter

Neben dem zyklischen Austausch der E/A-Daten besteht die Möglichkeit, zusätzliche Parameter - Anwenderparameter - zu definieren, die einmal beim Start des zyklischen Datenaustauschs (Bussystem) an den Adapter übermittelt werden. Der Umfang der zur Verfügung stehenden Parameter wird vom verwendeten E/A-Gerät bestimmt. Anwenderparameter werden in den Eigenschaften des Adapters angelegt. Sind die Parameter in der Gerätebeschreibungsdatei beschrieben, wird eine Auswahl dieser Parameter beim Anlegen im Editor angeboten. Andernfalls müssen diese manuell angelegt werden.

15.13.8 Editoren für die Konfiguration von EtherNet/IP

Ein EtherNet/IP-Netzwerk besteht aus einem EtherNet/IP-Scanner und einem oder mehreren EtherNet/IP-Targets. Dabei stellt der Scanner den Master und die Targets die Slaves im Netzwerk dar.

Die MLD unterstützt nur die Funktion als Scanner. Um den Antrieb als Adapter zu nutzen, ist die Führungskommunikation entsprechend zu konfigurieren.

IndraWorks bietet Ihnen Dialoge zum Einstellen der Parameter sowie zum Abbilden der Ein-/Ausgänge auf Variablen.

Ethernet

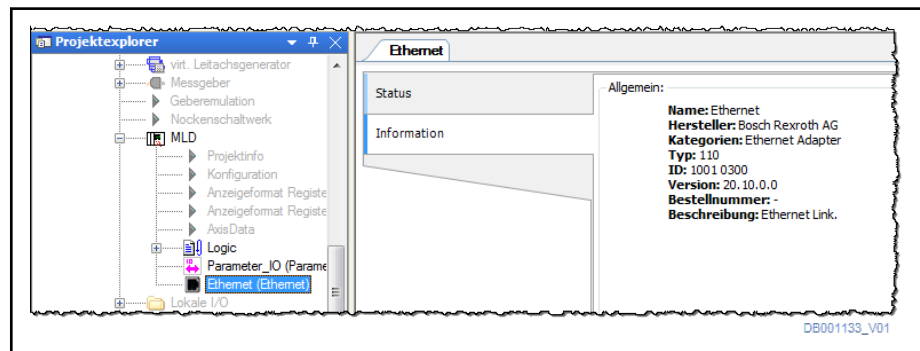


Abb. 15-5: Dialog "Ethernet"

Status Die Registerkarte "Status" zeigt Statusinformationen zum Device und der Funktion an. Beim Ethernet Device ist der Status immer "Running".

Information Die Registerkarte "Information" zeigt allgemeine Informationen an, die aus der Gerätebeschreibungsdatei stammen:

- Name
- Hersteller
- Kategorien
- Typ
- ID
- Version
- Bestellnummer
- Beschreibung

MLD EtherNet/IP-Scanner

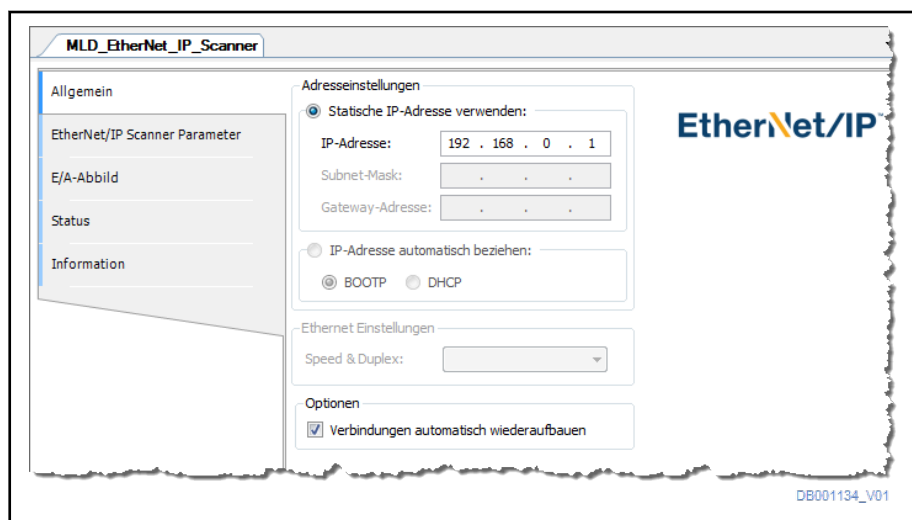


Abb. 15-6: Dialog "MLD EtherNet/IP-Scanner"

Allgemein

Die Registerkarte "Allgemein" im Konfigurator des EtherNet/IP-Scanners enthält die Grundeinstellungen für die Kommunikation im Netzwerk.

Adresseinstellungen

Statische IP-Adresse verwenden:	
IP-Adresse: Subnet-Mask: Gateway-Adresse:	Diese Einträge belegen jeweils vier Byte und dienen zur Identifizierung des Scanners innerhalb der Netzwerkumgebung. Die Einstellungen "Subnet-Mask" und "Gateway-Adresse" werden nicht unterstützt.
IP-Adresse automatisch beziehen: <i>Die Option wird nicht unterstützt.</i>	
BOOTP	Zuweisung der IP-Adresse durch einen Server mittels Bootstrap Protocol (BOOTP)
DHCP	Automatische Konfiguration der Netzwerkeinstellungen durch den Host mittels Dynamic Host Configuration Protocol (DHCP)

Ethernet-Einstellungen

Speed & Duplex <i>Die Option wird nicht unterstützt.</i>	Bitrate der Übertragung. Bei "Auto-negotiation" wird automatisch die höchste der jeweils verfügbaren Bitraten gewählt.
---	---

Optionen

Verbindungen automatisch wiederaufbauen	Wenn diese Option ausgewählt ist, versucht der Scanner eine unterbrochene Verbindung automatisch wieder aufzubauen.
---	---

EtherNet/IP Scanner Parameter

Die Registerkarte "EtherNet/IP Scanner Parameter" im Konfigurator des EtherNet/IP-Scanners entspricht dem generischen Konfigurationsdialog. Der Dialog zeigt die "ConfigVersion" an. Dieser Wert wird automatisch erzeugt und zeigt die Version des PlugIn in IndraWorks an, mit der Scanner konfiguriert wurde.

Systembibliotheken

Ist die Option "Generische Konfigurationseditoren anzeigen" aktiviert, wird der Dialog um einen weiteren Dialog ergänzt, der die gerätespezifischen Parameter in Listenform anzeigt. Dieser Dialog ist für Testzwecke bestimmt. Werte sollten darin nur von Experten geändert werden.

E/A-Abbild

Die Registerkarte "E/A-Abbild" zeigt den Namen der automatisch erzeugten Instanz des Adapters unter "IEC-Objekte" in der Spalte "Variable" an. Damit kann über die Applikation auf diese Instanz zugegriffen werden. Somit kann zum Beispiel ein Neustart, ein Stop oder eine Statusprüfung des Scanners über eine Applikation erfolgen.

Beim EtherNet/IP-Scanner können keine Projektvariablen auf die Ausgänge und Eingänge abgebildet werden.

Buszyklusoptionen

Buszyklus-Task	Die Auswahlliste bietet alle Tasks an, die in der Task-Konfiguration der aktiven Applikation definiert sind. Bei "Zykluseinstellungen des übergeordneten Busses verwenden" werden die Einstellungen des Vaterknotens verwendet. Wird hier keine Task definiert, gilt die in den SPS-Einstellungen eingestellte Task. Ist dort ebenfalls keine Task eingestellt, dann wird die höchstprioräre Task als Buszyklustask verwendet.
----------------	--

Status

Die Registerkarte "Status" zeigt Statusinformationen (z. B. "Running", "Stopped") und spezifische Diagnosemeldungen an. Die Statusinformation wird für das Gerät und das Bussystem getrennt angezeigt. Quittierbare Diagnosemeldungen können quittiert werden.

Information

Die Registerkarte "Information" zeigt allgemeine Informationen an, die aus der Gerätebeschreibungsdatei stammen:

- Name
- Hersteller
- Kategorien
- Typ
- ID
- Version
- Bestellnummer
- Beschreibung

EtherNet/IP-Target

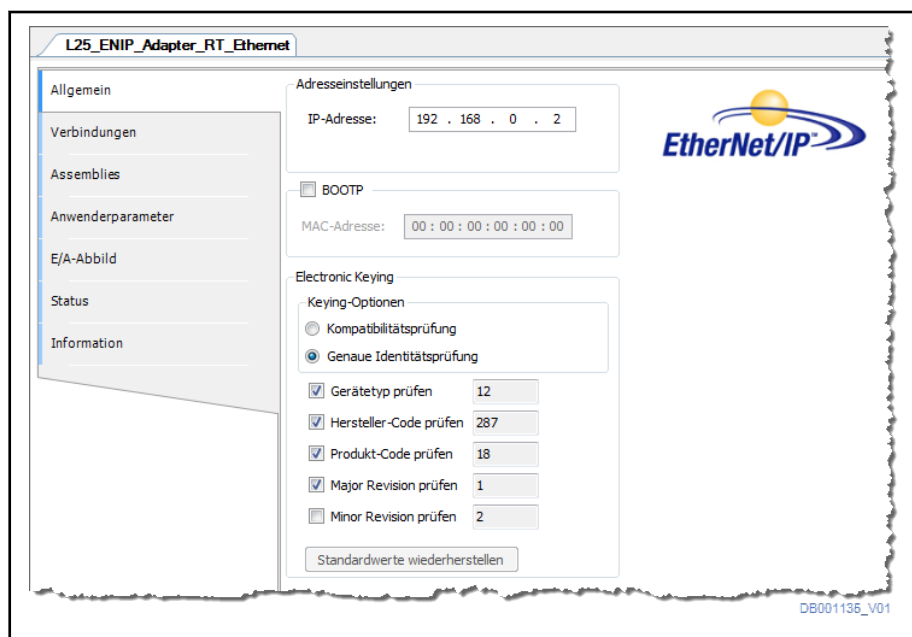


Abb. 15-7: Dialog eines EtherNet/IP-Targets

Allgemein

Die Registerkarte "Allgemein" im Konfigurationseditor des EtherNet/IP-Targets enthält die Grundeinstellungen für die Kommunikation im Netzwerk.

Adresseinstellungen

IP-Adresse	Adresse zur Identifizierung des EtherNet/IP-Target-Geräts.
------------	--



Auch die Adapter müssen so konfiguriert sein, dass die IP-Adressen fest vergeben sind. "BOOTP" und "DHCP" sind in den Adaptern zu deaktivieren. Andernfalls kann es sein, dass zu den Adaptern keine Kommunikation möglich ist. Das Vorgehen hierzu ist der jeweiligen Dokumentation des Adapters zu entnehmen.

BOOTP

Bootstrap Protocol (BOOTP) <i>Diese Option wird vom Scanner nicht unterstützt.</i>	
MAC-Adresse	Gerätespezifische MAC-Adresse des Slaves

Electronic Keying

Keying-Optionen	Kompatibilitätsprüfung: Target (Adapter) soll eine Kompatibilitätsprüfung der Keying-Werte aus der Gerätestammdatei (*.EDS) mit den eigenen Keying-Werten durchführen Genauere Identitätsprüfung: Target (Adapter) soll eine exakte Prüfung der Keying-Werte aus der Gerätestammdatei (*.EDS) mit den eigenen Keying-Werten durchführen Wenn die Überprüfung fehlschlägt, wird keine E/A-Verbindung zum Gerät aufgebaut und eine Fehlermeldung auf der Statusseite ausgegeben.
-----------------	---

Systembibliotheken

Gerätetyp prüfen	Wenn die jeweilige Option ausgewählt ist, wird die Information für die Kompatibilitäts- oder Identitätsprüfung berücksichtigt. Bei generischen Geräten können diese Werte verändert werden.
Hersteller-Code prüfen	
Produkt-Code prüfen	
Major Revision prüfen	
Minor Revision prüfen	

Verbindungen

Die Registerkarte "Verbindungen" zeigt im oberen Teil eine Liste aller konfigurierten Verbindungen. Wenn die "Exclusive owner"-Verbindung in der Gerätestammdatei (*.EDS) vorhanden ist, wird sie beim Hinzufügen des Targets automatisch eingefügt. Die Konfigurationsdaten der Verbindungen können im unteren Teil des Dialogs verändert werden.

Die Konfigurationsdaten sind in der Gerätestammdatei (*.EDS) definiert. Sie werden beim Öffnen der Verbindung zum Adapter übertragen.

RPI (ms)	Requested Packet Interval: Austauschintervall der Ein- und Ausgangsdaten
O -> T Größe (Bytes)	Größe der Producerdaten vom Scanner an den Adapter (Originator --> Target)
T -> O Größe (Bytes)	Größe der Consumer-Daten vom Adapter zum Scanner (T --> O)
Config #1 Größe (Bytes)	Größe der Konfigurationsdaten 1
Config #2 Größe (Bytes)	Größe der Konfigurationsdaten 2
Verbindungspfad	Adresse der - Konfigurationsobjekte - Eingangsobjekte - Ausgangsobjekte
Verbindung hinzufügen...	Durch Klick auf die Schaltfläche wird der Dialog "Neue Verbindung" geöffnet. In dem Dialog können die Parameter für die neue Verbindung festgelegt werden.
Verbindung löschen	Durch Klick auf die Schaltfläche wird die selektierte Verbindung aus der Liste gelöscht.
Verbindung bearbeiten...	Durch Klick auf die Schaltfläche wird der Dialog "Verbindung bearbeiten" geöffnet. In dem Dialog können die Parameter für die bestehende Verbindung geändert werden.

Konfigurationsdaten

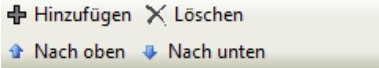
Symbolische Werte	Wenn diese Option ausgewählt ist, dann wird der Wert in Textform angezeigt, falls in der Gerätestammdatei (*.EDS) definiert.
Voreinstellungen	Durch Klick auf die Schaltfläche werden die Werte auf die Voreinstellungen zurückgesetzt.
Wert	Führen Sie einen Doppelklick auf einen Wert aus, um den Wert zu verändern.

Assemblies

Die Registerkarte zeigt im oberen Teil eine Liste aller konfigurierten Verbindungen. Wenn Sie eine Verbindung selektieren, werden die zugehörigen Assemblies im unteren Teil des Dialogs angezeigt.

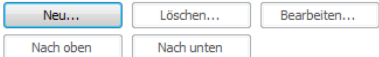
Eine Beschreibung der Spalten finden Sie unter "[Verbindungen](#)" der Beschreibung des EtherNet/IP-Targets.

Ausgangs-Assembly, Eingangs-Assembly

	Durch Klick auf die Schaltfläche Hinzufügen wird der Dialog " Parameter auswählen " geöffnet.
	Durch Klick auf die Schaltfläche Löschen werden alle selektierten Parameter gelöscht.
	Durch Klick auf die Schaltfläche Nach oben/Nach unten werden alle selektierten Parameter innerhalb der Liste verschoben. Die Reihenfolge in der Liste bestimmt die Reihenfolge im E/A-Abbild.
Name	
Bitlänge	Führen Sie einen Doppelklick auf einen Wert aus, um den Wert zu verändern.
Hilfe-String	

Anwenderparameter

Die Registerkarte "Anwenderparameter" zeigt alle zusätzlichen Parameter, die einmalig in der dafür vorgesehenen Phase des Startvorgangs in das Bussystem übertragen werden.

	Durch Klick auf die Schaltfläche Neu... wird der Dialog " Parameter auswählen " geöffnet, um einen neuen Parameter hinzuzufügen. Der neue Parameter wird vor der selektierten Zeile eingefügt.
	Durch Klick auf die Schaltfläche Bearbeiten... wird der Dialog " Parameter auswählen " geöffnet, um einen bestehenden Parameter zu verändern.
	Durch Klick auf die Schaltfläche Löschen... wird der Dialog " Parameter auswählen " geöffnet, um einen bestehenden Parameter zu löschen.
	Durch Klick auf die Schaltfläche Nach oben/Nach unten wird die Reihenfolge der Anwenderparameter geändert. Die Reihenfolge der Parameter in der Liste entspricht der Reihenfolge bei der Initialisierung.
Wert	Der Wert des jeweiligen Parameters kann durch Doppelklick auf den Wert direkt verändert werden. Gegebenenfalls öffnet sich ein Listenfeld mit möglichen Werten.
Im Fehlerfall abbrechen	Bei Auftreten eines Fehlers wird die komplette Übertragung der Parameter abgebrochen.
Im Fehlerfall Sprung zu Zeile	Im Fehlerfall wird mit der Zeile fortgefahren, die in der Spalte "Nächste Zeile" angegeben ist. Es kann somit während der Initialisierung ein kompletter Block übersprungen werden oder auch ein Rücksprung definiert werden. Hinweis: Ein Rücksprung kann zu einer Endlosschleife führen, wenn es nie möglich ist einen bestimmten Parameter zu schreiben.

EtherNet/IP Parameters

Die Registerkarte "EtherNet/IP Parameters" im Konfigurator des EtherNet/IP-Scanners entspricht dem generischen Konfigurationsdialog und wird angezeigt, wenn die Option "Generische Konfigurationseditoren anzeigen" in den Optionen von IndraWorks aktiv ist.

Es werden die gerätespezifischen Parameter in Listenform angezeigt. Dieser Dialog ist für Testzwecke bestimmt. Werte sollten darin nur von Experten geändert werden.

Systembibliotheken

E/A-Abbild

Die Registerkarte "E/A-Abbild" zeigt den Namen der automatisch erzeugten Instanz des Targets unter "IEC-Objekte" in der Spalte "Variable" an. Damit kann über die Applikation auf diese Instanz zugegriffen werden. Somit kann zum Beispiel ein Neustart, ein Stop oder eine Statusprüfung des Targets über eine Applikation erfolgen.

In der Registerkarte werden Projektvariablen auf die Ausgänge und Eingänge des Targets abgebildet.

Die Beschreibung entspricht der Beschreibung des allgemeinen E/A-Abbild-Dialogs.

Variablen immer aktualisieren	Wenn diese Option ausgewählt ist, werden alle Variablen in jedem Zyklus der Buszyklus-Task aktualisiert, egal ob sie verwendet werden und ob sie auf einen Ein- oder Ausgangskanal abgebildet sind.
Buszyklus-Task	Die Auswahlliste bietet alle Tasks an, die in der Task-Konfiguration der aktiven Applikation definiert sind, wie zum Beispiel 'MainTask'. Bei "Zykluseinstellungen des übergeordneten Busses verwenden" werden die Einstellungen des Vaterknotens verwendet. Wenn hier keine Task definiert wird, gilt die in den SPS-Einstellungen eingestellte Task. Wenn dort ebenfalls keine Task eingestellt ist, dann wird die höchstpriorisierte Task als Buszyklus-Task verwendet.

Status

Die Registerkarte "Status" zeigt Statusinformationen (z. B. "Running", "Stopped") und spezifische Diagnosemeldungen an. Quittierbare Diagnosemeldungen können quittiert werden.

Information

Die Registerkarte "Information" zeigt allgemeine Informationen an, die aus der Gerätebeschreibungsdatei stammen:

- Name
- Hersteller
- Kategorien
- Typ
- ID
- Version
- Bestellnummer
- Beschreibung

Dialog "Parameter auswählen"**Dialog "Parameter auswählen"**

Parametergruppen anzeigen	Wenn diese Option ausgewählt ist, zeigt der Dialog alle Parameter aus der Gerätestammdatei (*.EDS) in Gruppen zusammengefasst. Wenn diese Option nicht ausgewählt ist, zeigt der Dialog alle Parameter aus der Gerätestammdatei (*.EDS) in flacher Struktur. Sie können einzelne Parameter dieser Liste selektieren und mit OK der Liste der Assemblies hinzufügen.
Generischer Parameter	Wenn diese Option ausgewählt ist, können Sie generische Parameter hinzufügen. Dabei sind die einzelnen Werte des Parameters editierbar.

15.13.9 Azyklische Kommunikation (Explicit Messaging Service)

Die Bibliothek "EtherNet/IP Services" beinhaltet Funktionsbausteine, um eine azyklische Kommunikation mit dem Adapter umzusetzen. Diese setzen die Funktionen und Attribute der CIP-Objekte um, wie diese in der CIP-Spezifikation der Open DeviceNet Vendor Association (ODVA) beschrieben sind:

- Get_Attribute_Single
- Set_Attribute_Single
- Get_Attributes_All
- Set_Attributes_All
- Reset
- NOP (No Operation)
- Start
- Stop
- Apply_Attributes

Bibliothek "EtherNet/IP Services"

Get_Attribute_Single

Verwenden Sie diesen Funktionsbaustein, um das Attribut einer bestimmten Instanz eines CIP-Objekts abzufragen.

Eingänge

Name	Datentyp	Beschreibung
xExecute	BOOL	Steigende Flanke: Die Aktion startet Fallende Flanke: Rücksetzen der Ausgänge. Wenn eine fallende Flanke auftritt bevor der Funktionsbaustein seine Aktion beendet hat, werden die Ausgänge nur dann zurückgesetzt, wenn entweder die Aktion beendet oder ein Fehler aufgetreten ist. In diesem Fall stehen die entsprechenden Ausgangswerte (xDone, xError, eError) für genau einen Zyklus am Ausgang an.
itfEtherNetIPDevice	IEtherNetIPService	EtherNet/IP-Gerät, dass das EtherNet/IP-Services-Interface implementiert
eClass	WORD	Klasse, die den Service ausführen soll
dwInstance	DWORD	Instanz, die den Service ausführen soll (0: Class level, 1..x: Instance level)
pData	POINTER TO BYTE	Datenpuffer, in dem die Daten des Attributs geladen werden sollen
udiDataSize	USINT	Größe des Puffers
wAttribute	Word	Attribut, das abgefragt wird

Tab. 15-2: Eingänge des Funktionsbausteins "Get_Attribute_Single"

Ausgänge

Name	Datentyp	Beschreibung
xDone	BOOL	Aktion erfolgreich beendet
xBusy	BOOL	Funktionsbaustein aktiv

Systembibliotheken

Name	Datentyp	Beschreibung
xError	BOOL	0: kein Fehler 1: Fehler aufgetreten, der Funktionsbaustein bricht die Aktion ab
eError	Error	Fehler (0-255: CIP-Fehler; 256-x: Bibliotheksfehler)

Tab. 15-3: Eingänge des Funktionsbausteins "Get_Attribute_Single"

Beispiel

Dieser Instanzaufruf fragt den Host-Namen (= Name des Gerätes im Netzwerk) von dem Gerät DEVICE_123_ETHERNET ab, sowie xExecute gleich TRUE wird.

Programm:

```

getAttributeSingle(
    xExecute:=,
    (* instance of the device (instance is found in the I/O Mapping of the device) *)
    itfEtherNetIPDevice:= DEVICE_123_ETHERNET,
    (* cip class which contains the desired attribute *)
    eClass:= ENIP.CIPClass.TCPIPInterfaceObject,
    (* value of 0 is class level, range from 1..x is instance level *)
    usiInstance:= 1,
    (* data buffer *)
    pData:= ADR(strHostnameToGet),
    (* size of the data buffer *)
    udiDataSize:= SIZEOF(strHostnameToGet) ,
    (* attribute no. 6 of the tcp/ip interface object is the Hostname *)
    (* (CIP Spec. Vol.2 Chapter 5-3.3.2.6) *)
    usiAttribute:= 6,
    xDone=> ,
    xBusy=> ,
    xError=> ,
    eError=> ,
    (* if the request was successfull, udiReceivedData contains the number of bytes *)
    (* which have been received *)
    udiReceivedDataSize=> udiReceivedData);

```

Set_Attribute_Single

Der Funktionsbaustein wird verwendet, um das Attribut einer bestimmten Instanz eines CIP-Objekts zu setzen.

Die Bedeutung der Ein-/Ausgänge ist in "Get_Attribut_Single" oder in der Dokumentation der Bibliothek im Bibliotheksverwalter beschrieben.

Beispiel

Dieser Instanzaufruf setzt den Host-Namen (= Name des Gerätes im Netzwerk) von dem Gerät DEVICE_123_ETHERNET, sowie xExecute gleich TRUE wird.

Programm:

```

strHostnameToSet.uiLength := INT_TO_UINT(Standard.LEN(strHostnameToSet.sString));
setAttributeSingle(
    xExecute:= ,
    (* instance of the device (instance is found in the I/O Mapping of the device) *)
    itfEtherNetIPDevice:= DEVICE_123_ETHERNET,
    (* cip class which contains the desired attribute *)
    eClass:= ENIP.CIPClass.TCPIPInterfaceObject,
    (* value of 0 is class level, range from 1..x is instance level *)
    usiInstance:= 1,
    (* data to be assigned to the attribute *)
    pData:= ADR(strHostnameToSet),
    (* size of the data *)
    (* here: size of the Length-Parameter PLUS the length of the string *)
    udiDataSize:= SIZEOF(strHostnameToSet.uiLength)
    + strHostnameToSet.uiLength,
    (* attribute no. 6 of the tcp/ip interface object is the Hostname *)
    (* (CIP Spec. Vol.2 Chapter 5-3.3.2.6) *)
    usiAttribute:= 6,
    xDone=> ,
    xBusy=> ,

```

```
xError=> ,  
eError=> );
```

Get_Attributes_All

Der Funktionsbaustein wird verwendet, um alle Attribute einer bestimmten Instanz eines CIP-Objekts **abzufragen**.

Die Bedeutung der Ein-/Ausgänge ist in "[Get_Attribut_Single](#)" oder in der Dokumentation der Bibliothek im Bibliotheksverwalter beschrieben.

Set_Attributes_All

Der Funktionsbaustein wird verwendet, um alle Attribute einer bestimmten Instanz eines CIP-Objekts **zu setzen**.

Die Bedeutung der Ein-/Ausgänge ist in "[Get_Attribut_Single](#)" oder in der Dokumentation der Bibliothek im Bibliotheksverwalter beschrieben.

Reset

Der Funktionsbaustein wird verwendet, um den Reset-Dienst einer bestimmten Instanz eines CIP-Objekts aufzurufen. Die genaue Auswirkung dieses Dienstes ist abhängig von dem CIP-Objekt (siehe Spezifikationen CIP Volume 1 und Volume 2).

Die Bedeutung der Ein-/Ausgänge ist in "[Get_Attribut_Single](#)" oder in der Dokumentation der Bibliothek im Bibliotheksverwalter beschrieben.

Beispiel

Dieser Instanzaufruf setzt das Gerät DEVICE_123_ETHERNET zurück, sowie xExecute gleich TRUE wird.

Programm:

```
reset(  
  xExecute:= ,  
  (* instance of the device *)  
  (* (instance is found in the I/O Mapping of the device) *)  
  itfEtherNetIPDevice:= DEVICE_123_ETHERNET,  
  (* cip class which contains the desired attribute *)  
  eClass:= ENIP.CIPClass.IdentityObject,  
  (* value of 0 is class level, range from 1..x is instance level *)  
  usiInstance:= 1,  
  xDone=> ,  
  xBusy=> ,  
  xError=> ,  
  eError=> );
```

NOP (No Operation)

Der Funktionsbaustein wird verwendet, um den "NOP"-Service einer bestimmten Instanz eines CIP-Objekts aufzurufen. Dieser Dienst wird üblicherweise verwendet, um zu prüfen ob der Adapter noch im Netzwerk vorhanden ist.

Die Bedeutung der Ein-/Ausgänge können Sie der Beschreibung von "[Get_Attribut_Single](#)" oder der Dokumentation der Bibliothek im Bibliotheksverwalter entnehmen.

Start

Der Funktionsbaustein wird verwendet, um den "Start"-Service einer bestimmten Instanz eines CIP-Objekts aufzurufen. Die genaue Auswirkung dieses Dienstes ist abhängig von der CIP-Klasse (siehe Spezifikationen CIP Volume 1 und Volume 2).

Die Bedeutung der Ein-/Ausgänge können Sie der Beschreibung von "[Get_Attribut_Single](#)" oder der Dokumentation der Bibliothek im Bibliotheksverwalter entnehmen.

Systembibliotheken

Stop

Der Funktionsbaustein wird verwendet, um den "Stop"-Service einer bestimmten Instanz eines CIP-Objekts aufzurufen. Die genaue Auswirkung dieses Dienstes ist abhängig von der CIP-Klasse (siehe Spezifikationen CIP Volume 1 und Volume 2).

Die Bedeutung der Ein-/Ausgänge können Sie der Beschreibung von "[Get_Attribut_Single](#)" oder der Dokumentation der Bibliothek im Bibliotheksverwalter entnehmen.

Apply_Attributes

Der Funktionsbaustein wird verwendet, um den "Apply_Attributes"-Dienst einer bestimmten Instanz eines CIP-Objekts aufzurufen. Dadurch werden die vorher mit "Get_Attribut_Single" oder "Get_Attribut_All" gesetzten Attribute vom Adapter übernommen und gespeichert.

Die Bedeutung der Ein-/Ausgänge können Sie der Beschreibung von "[Get_Attribut_Single](#)" oder der Dokumentation der Bibliothek im Bibliotheksverwalter entnehmen.

16 Technologie-Bibliotheken

16.1 Einleitung

Die Technologie-Bibliotheken sind ab der Firmware-Version 20 verfügbar. Sie erweitern die Grundfunktionalität des Zielsystems und stellen applikationsspezifische Funktionen zur Verfügung.

16.2 MX_TechInterface

16.2.1 Übersicht

Vorstellung des "TechInterface"

Mit der Bibliothek "MX_TechInterface.library" wird die Funktion "AxisInterface" zur Verfügung gestellt.

Das Achs-Interface bündelt und erweitert PLCopen-Bewegungs-Funktionsbausteine und stellt eine einfach zu bedienende Schnittstelle für die Antriebsfunktionalität zur Verfügung. Weniger Code und leistungsfähigere Kommandos beschleunigen die Programmentwicklung von Applikationen.

Das Achs-Interface enthält Steuersignale und Parameter für die verschiedenen Betriebsarten der Leitachse und Folgeachse sowie Einstellmöglichkeiten für angewählte Prozesswerte.

Die Bibliothek "MX_TechInterface" wird in der Online-Hilfe von IndraWorks und in der gedruckten Dokumentation "Rexroth IndraMotion MLC 14VRS, Technologie-Bibliotheken" beschrieben.

Vergleich zwischen IndraMotion XLC/MLC und IndraMotion MLD

	IndraMotion XLC/MLC	IndraMotion MLD
FlexProfil	✓	–
MotionProfile	✓	✓
Parametrierebene (ParamLevel)	✓	✓
Diagnosen	A-Parameter	S-0-0390 / S-0-0095
Element "arAxisStatus_gb[].Admin.AxisFeatures"	✓	–



Weitere Besonderheiten, die bei der Anwendung des Achs-Interface im System IndraMotion MLD zu beachten sind, sind in dem Kapitel ["Anwendung des Achs-Interface im System IndraMotion MLD"](#) beschrieben.

16.2.2 Anwendung des Achs-Interface im System IndraMotion MLD

Einführung

Dieses Kapitel beschreibt die Besonderheiten, die bei der Anwendung des Achs-Interface im System IndraMotion MLD zu beachten sind.

Das Achs-Interface kann nicht in allen Firmwarevarianten genutzt werden:

- MPC: Anwendung als MLD-M mit bis zu 10 realen Achsen und einer virtuellen Achse
- MPB: Anwendung als MLD-S für eine Achse und einer virtuellen Achse.

Zur Anwendung des Achs-Interface muss das additive Funktionspaket "ML - IndraMotion MLD mit Technologiefunktionen" freigeschaltet sein. Das

Funktionspaket "MA - erweiterte Technologiefunktionsbausteine" wird nicht benötigt.

Parametrierung

Um das Achs-Interface auf einer MLD zu nutzen, muss die folgende Parametrierung ausgeführt werden:

- SPS hat permanente Kontrolle über den Antrieb: "P-0-1367, SPS Konfiguration", Bit 4 gesetzt
- Unterstützung der AxisData-Struktur: "P-0-1367, SPS Konfiguration", Bit 6 gesetzt

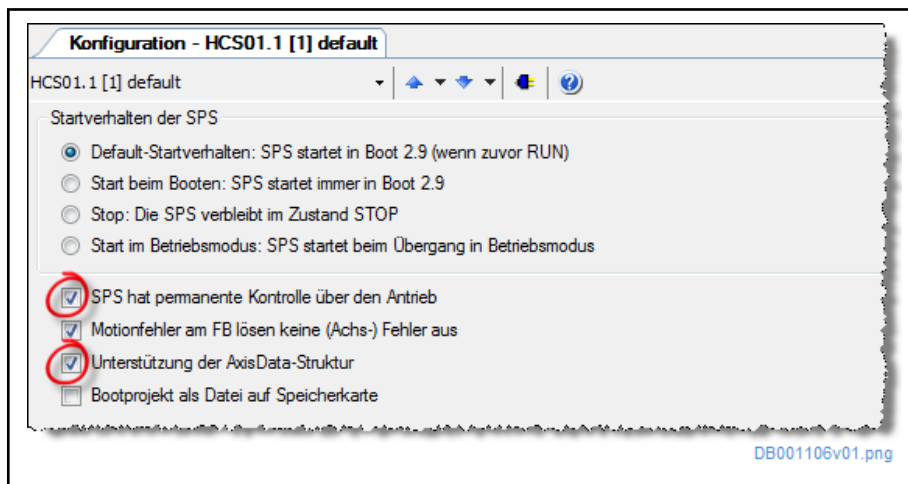


Abb. 16-1: Parametrierung MLD für Achs-Interface

Ein Bedienteil mit optionaler Speicherkarte muss für das Bootprojekt nicht vorhanden sein. Die Speicherkarte erweitert den nutzbaren Speicherplatz für das Bootprojekt.

Um den Leitachsgenerator als virtuelle Achse für das Achs-Interface nutzen zu können, sind folgenden Einstellungen notwendig:

- Aktivierung Leitachsgenerator: "P-0-0917, Steuerwort Leitachsgenerator", Bit 0 gesetzt
Wird der Leitachsgenerator im CCD-Verbund aktiviert, gilt folgender Parameter: "P-0-1820.0.1, CCD: Konfigurationsliste Leitachsen"
- Leitachsbildung: "P-0-0916, Leitachsformatwandler Signalauswahl", P-0-0758 eingetragen
Wird die Leitachsbildung im CCD-Verbund aktiviert, gilt folgender Parameter: "P-0-1822.0.1, CCD: Auswahl Default-Leitachsen für die IndraDrive-Achsen"
- Hier als Beispiel die Einstellungen für eine rotatorische virtuelle Achse, die in beiden Richtungen rotieren kann:
 - rotatorisch modulo gewichtet: "P-0-0756, Virtuelle Leitachse, Wichtungsart", Bit 2/1/0="010", Bit 7 gesetzt
 - kürzester Weg: "P-0-0769, Virtuelle Leitachse, Sollwertmodus", Bit 1/0="10"

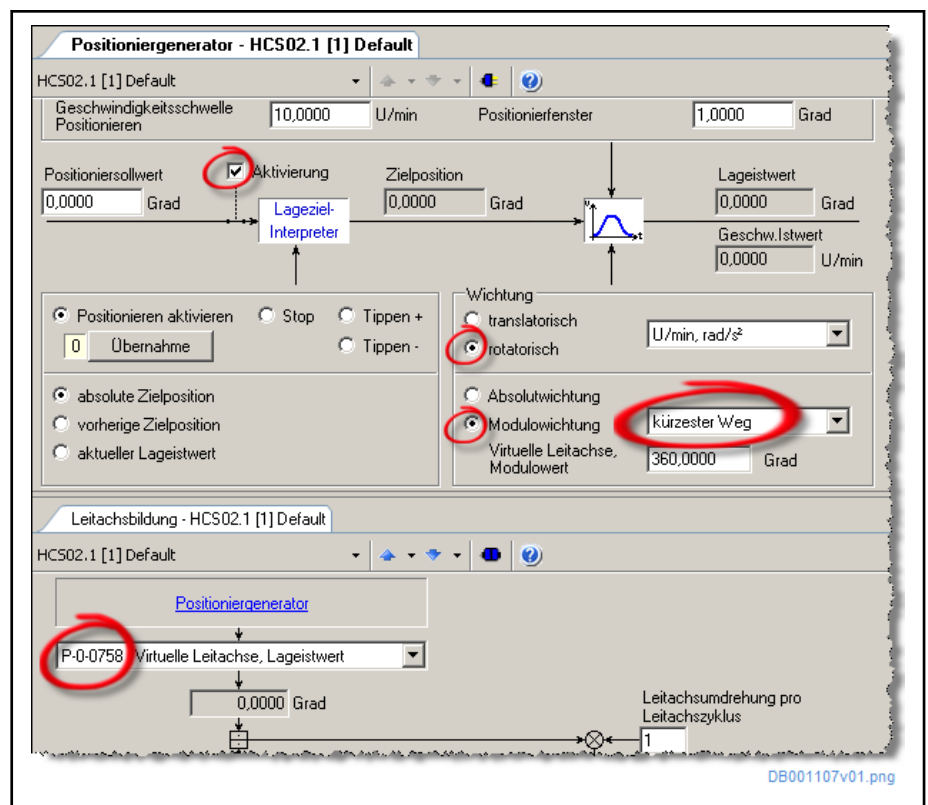


Abb. 16-2: Parametrierung MLD-Leitachsgenerator als virtuelle Achse

Falls die oben beschriebene Parametrierung nicht durchgeführt wird, wird die Initialisierung des AchsInterface-Beispielprojekts nicht komplett durchlaufen; es sind Anpassungen durch den Anwender nötig.

SPS-Programmierung

Die Achs-Interface Strukturen sind für 11 Achsen definiert:

- Index1 bis Index10: reale Achsen
- Index20: virtuelle Achse (=Leitachsgenerator) adressiert über den Define "VMA_1"



Da die Achs-Interface-Strukturen genau für 11 Achsen definiert sind, führt ein Indizieren dieser Strukturen mit den vordefinierten Achsreferenzen "VmAxisInt.AxisNo" oder "VmAxisExt.AxisNo" zu der Fehlermeldung 2021 "Array-Grenze überschritten...".

Hinweise zur virtuellen Achse

- Zugriff auf die Strukturen im Achs-Interface mit dem Define "VMA_1 (Index 20)"
- Mit der Funktion "MB_GetAxisInterfaceIndex()" kann der Index systemunabhängig auf MLC und MLD ermittelt werden.
- Das Achs-Interface bildet einen Teil der in der Struktur "AxisData" für reale Achsen angebotenen Informationen für die virtuelle Achse nach. Auf diese Informationen kann über einen Zeiger in der Struktur "AxisStatus" zugegriffen werden.



Dieser Zeiger wird vom Achs-Interface-Init-Funktionsbaustein "MX_AxisInitType01" initialisiert. Auf den Zeiger und damit auf die Informationen kann erst nach erfolgreicher Durchführung der Initialisierung zugegriffen werden.

Verfügbare Informationen der virtuellen Achsen

Für virtuelle Achsen sind die Informationen der realen Achsen vorhanden.

Verfügbare Information der virtuellen Achsen (diese Informationen werden im Betriebsmodus immer auf "TRUE" gesetzt!)

Verfügbare Informationen	Bedeutung
arAxisStatus[VMA_1].PtrAxisData^.Axis_Inbb;	Anzeige "Antriebsstatus bb"
arAxisStatus[VMA_1].PtrAxisData^.Axis_InAb;	Anzeige "Antriebsstatus Ab"
arAxisStatus[VMA_1].PtrAxisData^.Axis_Power;	Anzeige "Leistung zugeschaltet"
arAxisStatus[VMA_1].PtrAxisData^.Axis_Homed;	Anzeige "Achse referenziert"

16.3 MX_TechBase

Die Bibliothek "MX_TechBase" beinhaltet Basis-Funktionalitäten, wie Kontroll- und Steuerungsfunktionen [IndraMotion Control (IMC) Interface], Funktionen, um Systeminformationen abzufragen und trigonometrische Funktionen.

Die Bibliothek "MX_TechBase" wird in der Online-Hilfe von IndraWorks und in der gedruckten Dokumentation "Rexroth IndraMotion MLC 14VRS, Technologie-Bibliotheken" beschrieben.

16.4 MX_TechMotionBase

Die Bibliothek "MX_TechMotionBase" beinhaltet Funktionalitäten, wie Messtasterfunktion, Positionsüberwachung, Nockenschaltwerk und Funktionen zur Verwaltung des zyklischen Datenkanals.

Die Bibliothek "MX_TechMotionBase" wird in der Online-Hilfe von IndraWorks und in der gedruckten Dokumentation "Rexroth IndraMotion MLC 14VRS, Technologie-Bibliotheken" beschrieben.

16.5 MX_TechMotion

Die Bibliothek "MX_TechMotion" beinhaltet den Funktionsbaustein "MB_TensionControlLoadCellType01". Mit dem Funktionsbaustein wird ein Zugkraftregler zur Verfügung gestellt, der die Zugkraft auf einen Sollwert regelt, indem der Messwert einer Kraftmessdose ausgewertet wird.

Die Bibliothek "MX_TechMotion" wird in der Online-Hilfe von IndraWorks und in der gedruckten Dokumentation "Rexroth IndraMotion MLC 14VRS, Technologie-Bibliotheken" beschrieben.

16.6 MX_TechWinder

Die Bibliothek "MX_TechWinder" beinhaltet Funktionsbausteine, die Funktionen für das Auf- und Abwickeln von Stoffbahnen (Wicklern) bereitstellen.

Die Bibliothek "MX_TechWinder" wird in der Online-Hilfe von IndraWorks und in der gedruckten Dokumentation "Rexroth IndraMotion MLC 14VRS, Technologie-Bibliotheken" beschrieben.

16.7 MX_TechCrosscutCrossseal

Die Bibliothek "MX_TechCrosscutCrossseal" beinhaltet Funktionsbausteine für Quersiegler- und Querschneider-Anwendungen.

Mit IndraMotion MLD (2G) kann der Funktionsbaustein "MB_CrossCutterCalcType05" genutzt werden. Der Funktionsbaustein berechnet eine Kurvenscheibe für die Anwendung als Querschneider. Die Querschneider-Funktionalität ist dabei konfigurierbar.

Die Bibliothek "MX_TechCrosscutCrossseal" wird in der Online-Hilfe von IndraWorks und in der gedruckten Dokumentation "Rexroth IndraMotion MLC 14VRS, Technologie-Bibliotheken" beschrieben.

16.8 RMB_TechCam

Die Bibliothek "RMB_TechCam" beinhaltet Funktionsbausteine zum Erstellen von Kurvenscheibentabellen.

Die Funktionsbausteine sind für unterschiedliche Anwendungen wie Vorschubbewegung, Querschneider, Slotter etc. verfügbar.

Die Bibliothek "RMB_TechCam" wird in der Online-Hilfe von IndraWorks und in der gedruckten Dokumentation "Rexroth IndraMotion MLC 14VRS, Technologie-Bibliotheken" beschrieben.

17 Freie SPS-Bibliothek von "OSCAT"

Die quelloffene (freie) SPS-Bibliothek "Basic" von "OSCAT" ("Open Source Community for Automation Technology") beinhaltet eine Vielzahl nützlicher Funktionen in den Bereichen "Automatisierungstechnik" und "Gebäudeautomation".

Bibliotheken von "OSCAT" sind unabhängig von Zielsystem und Hersteller.

Die Bibliothek "Basic" enthält unter anderem Bausteine zu folgenden Themen:

- Mathematische Berechnungen
- String-, Listen- und Buffer-Verarbeitung
- Logikmodule
- Datums- und Zeitfunktionen
- Regelungstechnik
- Signalverarbeitung
- Netzwerk- und Kommunikations-Funktionen
- Messtechnik und Sensorik
- Gerätetreiber für Standardperipherie

OSCAT-Bibliotheken können unter der Adresse www.oscat.de bezogen werden. Bitte beachten Sie, dass es sich hierbei nicht um Produkte von Rexroth handelt, aus diesem Grund schließen wir jede Haftung für die Vollständigkeit, Funktionalität und Betriebssicherheit aus und bieten keinen Support für diese Bibliotheken an.



Bibliotheken können auf dem lokalen System in verschiedenen "Repositories" (Verzeichnisse, Speicherorte) verwaltet werden. Bevor eine Bibliothek in ein Projekt eingebunden werden kann, muss sie auf dem lokalen System in ein solches Repository installiert werden. Dies erfolgt in IndraLogic 2G im Dialog "Bibliotheks-Repository".



Um Compilerfehler durch Namensüberschneidungen mit anderen Bibliotheken zu vermeiden, sollten nur die Bausteine und Datentypen der OSCAT-Bibliotheken in ein Projekt eingebunden werden, die auch verwendet werden. Achten Sie daher darauf, dass im Bibliotheksverwalter bei den Eigenschaften der OSCAT-Bibliotheken der Haken bei **"Alle IEC Symbole im Projekt sichtbar machen, ..."** nicht gesetzt ist.

Index

A

Abarbeitung	
synchron.....	242
Analogwertverarbeitung.....	239
AXIS_REF.....	200

B

BCD-Konvertierung.....	239
Befehlsabarbeitung in einem PLCopen-FB	17
Bibliothek	
Debug.....	57
RIL_SercosIII.....	236
SysStr23.....	242
Bit/Byte-Funktionen.....	239
Buffer-Verarbeitung.....	269
Buszyklustask	
IoDrvEtherNetIP.....	247

C

CheckBounds.....	55
CheckDivLReal.....	55
CheckDivReal.....	55
CheckPointer.....	55
CheckRangeSigned.....	55
CheckRangeUnsigned.....	55
CheckRtv.....	55
CONTROLS.....	198

D

Darstellungsmittel	
Hinweise.....	8
Konventionen der Schreibweise.....	7
Schreibweisen.....	7
Dateizugriffe	
asynchrone.....	240
synchrone.....	240
Datentypen zur Unterstützung von Kurven-	
scheiben, Motion- und FlexProfilen.....	212
Datentypen zur Unterstützung von Mess-	
taster-Funktionen.....	207
Datentypen zur Wiedergabe des Datums	
und absoluter Zeitangaben bei IndraMotion-	
Bausteinen.....	205
Datumsfunktionen.....	269

E

Eingangssignale "Execute" / "Enable".....	17
Error.....	25
ERROR_CODE.....	25
ERROR_STRUCT.....	25, 205
ErrorID.....	25
ErrorIdent.....	25
EtherNet und Sercos	
Koexistenz.....	248
EtherNet/IP	

Anforderungen an das IP-Netzwerk.....	248
EtherNet/IP und Sercos	
Topologie.....	248
EtherNet/IP-Adapter	
aktivieren/deaktivieren.....	250
vom Bus trennen.....	249
EtherNet/IP-Scanner	
Ressourcenbedarf.....	247
Schnittstelle.....	247
Taskkonfiguration.....	250
EXT_SYS_TIME_DATE.....	206

F

Fehlerbehandlung.....	25
-----------------------	----

I

IL_NULL.....	198
INDRV_TABLE.....	31
IoDrvEtherNetIP	
Einleitung.....	246
Funktionsumfang.....	249
Lizenzierung.....	246
Performance und Zykluszeit.....	247
Testlizenz.....	246
Watchdog.....	247
IoDrvEtherNetIP.library	
hinzufügen.....	250
Version ändern.....	250

K

Kommunikation über TCP/IP.....	241
Kommunikation über UDP.....	241

L

Laufzeitereignis.....	241
Laufzeitfehler	
melden / prüfen.....	55
Listen- Verarbeitung.....	269
Logikmodule.....	269

M

Master Sercos Interface	
Kenndaten.....	247
Mathematische Berechnungen.....	269
MB_ChangeCamData.....	63
MB_ChangeProfileSet.....	66
MB_ChangeProfileStep.....	69
MB_ClearAllError.....	71
MB_Command.....	135
MB_DIFF_EVAL.....	208
MB_EXPECT_WINDOW_MODE.....	208
MB_GearInPos.....	72
MB_GetAxisInterfaceIndex.....	265
MB_Home.....	77
MB_INTERPOLATION_MODE.....	209

Index

MB_MotionProfile.....	78	MX_fCANL2_Status.....	49
MB_Phasing.....	81	MX_fCANtoHI.....	51
MB_PhasingSlave.....	83	MX_fCANtoHs.....	52
MB_POSITION_EVAL.....	210	MX_fCheckCmdRequest.....	59
MB_PreSetMode.....	85	MX_fDINT_AccTo_REAL.....	188
MB_PROBE_CONFIG.....	210	MX_fDINT_DistTo_REAL.....	189
MB_PROBE_DATA.....	211	MX_fDINT_VelTo_REAL.....	189
MB_PROBE_INPUT.....	211	MX_fGetDriveWarning.....	59
MB_PROBE_MODULE.....	212	MX_fGetFreeTicks.....	193
MB_ReadListParameter.....	141	MX_fGetHighResTime.....	193
MB_ReadMaxRealValue.....	144	MX_fHighResTimerTicks_to_us.....	194
MB_ReadMaxValue.....	146	MX_fHtoCANI.....	52
MB_ReadMinRealValue.....	147	MX_fHtoCANS.....	53
MB_ReadMinValue.....	149	MX_fReadParamDINT.....	180
MB_ReadName.....	150	MX_fReadStringParam.....	181
MB_ReadParameter.....	152	MX_fREAL_AccTo_DINT.....	190
MB_ReadRealListParameter.....	154	MX_fREAL_DistTo_DINT.....	191
MB_ReadRealParameter.....	156	MX_fREAL_TorqueRampTo_DINT.....	191
MB_ReadSercosAttribute.....	158	MX_fREAL_TorqueTo_DINT.....	192
MB_ReadSercosDataStatus.....	160	MX_fREAL_VelTo_DINT.....	192
MB_ReadStringParameter.....	162	MX_fSetDriveError.....	60
MB_ReadUnit.....	164	MX_fSetDriveErrorText.....	60
MB_SetPositionControlMode.....	86	MX_fSetDriveWarning.....	61
MB_Stop.....	88	MX_fSetDriveWarningText.....	61
MB_WriteListParameter.....	166	MX_fSetParamLimits.....	181
MB_WriteParameter.....	168	MX_fSetParamName.....	182
MB_WriteRealListParameter.....	170	MX_fSetParamUnit.....	183
MB_WriteRealParameter.....	173	MX_fWriteParamDINT.....	183
MB_WriteStringParameter.....	175	MX_fWriteStringParam.....	185
MC_CAM_ID.....	212	MX_LECTaskGetLoad.....	194
MC_CamIn.....	90	MX_MoveAbsolute.....	120
MC_CamOut.....	93	MX_MoveAdditive.....	123
MC_GearIn.....	94	MX_MoveRelative.....	126
MC_GearOut.....	96	MX_PositionLoopEventInfo.....	195
MC_Jog.....	97	MX_ReadParamDINT.....	186
MC_MoveAbsolute.....	101	MX_SetCmdState.....	62
MC_MoveAdditive.....	104	MX_SetControl.....	129
MC_MoveRelative.....	106	MX_SetDeviceMode.....	130
MC_MoveVelocity.....	109	MX_SetOpMode.....	132
MC_Power.....	111	MX_WriteParamDINT.....	187
MC_ReadStatus.....	113		
MC_Reset.....	114	O	
MC_START_MODE.....	201	OBJECTS.....	198
MC_STEP_MODE.....	213		
MC_Stop.....	116	P	
MC_SYNC_DIRECTION.....	203	Parameterstruktur.....	35
MC_TorqueControl.....	118	Aufbau.....	35
MLD_TABLE.....	26	Sercos Attribut.....	35
MX_CanL2.....	43	Performance und Zykluszeit	
MX_CANL2_FifoReceive.....	43	IoDrvEtherNetIP.....	247
MX_CANL2_FifoSend.....	44		
MX_CANL2_Init.....	44	R	
MX_CANL2_SetLED.....	46	RIL_SercosIII.....	236
MX_Command.....	177		
MX_fCANL2_MBoxReceive.....	47	S	
MX_fCANL2_MBoxRecStatus.....	49	Scaling.....	188
MX_fCANL2_MBoxSend.....	48	Semaphore.....	241
MX_fCANL2_MBoxSndStatus.....	50		

SERCOS_TABLE.....	30
Signalgeneratoren.....	239
Signallaufverhalten.....	17
Sockets.....	241
Speicherverwaltung.....	241
String-Verarbeitung.....	269
Stringlänge ermitteln.....	242
Strings.....	242
Strings kopieren.....	242
Strings vergleichen.....	242
synchrone Dateizugriffe.....	240
SYS_TIME_DATE.....	207
SYS_TIME64.....	206
SysStr23.....	242
SysStrCmp.....	242
SysStrCmpl.....	242
SysStrCmpN.....	243
SysStrCmpNI.....	243
SysStrCpy.....	243
SysStrLen.....	244
Systembibliotheken.....	239
SysTimeRtcGet.....	244
SysTimeRtcHighResGet.....	245
SysTimeRtcHighResSet.....	245
SysTimeRtcSet.....	244
T	
TOUCHPROBE_REF.....	204
TP_OBJECTS.....	199
V	
Verzeichniszugriff.....	239
W	
Watchdog	
IoDrvEtherNetIP.....	247
Z	
Zeitfunktionen.....	269

Notizen

Notizen

Bosch Rexroth AG

Electric Drives and Controls

Postfach 13 57

97803 Lohr, Deutschland

Bgm.-Dr.-Nebel-Str. 2

97816 Lohr, Deutschland

Tel. +49 9352 18 0

Fax +49 9352 18 8400

www.boschrexroth.com/electrics



R911338915