

IndraDrive

IndraMotion MLD (2G)
ab MPx-18

Anwendungsbeschreibung
R911338913

Ausgabe 05



Titel	IndraDrive IndraMotion MLD (2G) ab MPx-18
Art der Dokumentation	Anwendungsbeschreibung
Dokumentations-Type	DOK-INDRV*-MLD3-**VRS*-AP05-DE-P
Interner Ablagevermerk	RS-77461bce0f8a914b0a6846a501d9a219-5-de-DE-5
Zweck der Dokumentation	Diese Dokumentation beschreibt die für IndraMotion MLD spezifischen Funktionalitäten ab der Firmware MPx-18. Da ab dieser Firmwareversion IndraLogic 2G unterstützt wird, wird in dieser Dokumentation außerdem die Konvertierung von IndraLogic 1.x-Projekten in IndraLogic 2G-Projekte erläutert.
Änderungsverlauf	Ausgabe 05, 2019-05 Ausgaben siehe ""Ausgaben dieser Dokumentation" auf Seite 1"
Schutzvermerk	© Bosch Rexroth AG 2019 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	Abt. DC-AE/EPI5

Inhaltsverzeichnis

	Seite
1 Einleitung.....	1
1.1 Über diese Dokumentation.....	1
1.2 Aufbau und Struktur der Dokumentation.....	4
1.3 Dokumentationen.....	4
1.3.1 Übersicht.....	4
1.3.2 Antriebssysteme, Systemkomponenten.....	5
1.3.3 Motoren.....	5
1.3.4 Kabel.....	6
1.3.5 Firmware.....	7
1.3.6 MLD-Technologiefunktionen.....	9
1.4 Mitgeltende Dokumentationen.....	9
1.5 Produktvorstellung.....	9
1.5.1 Einleitung und Übersicht.....	9
1.5.2 Kurzbeschreibung.....	10
1.5.3 IndraMotion MLD-S.....	12
1.5.4 IndraMotion MLD-M.....	13
1.5.5 Erweiterungen der MLD-M.....	14
1.5.6 Begriffe, Grundlagen.....	16
1.5.7 Funktionsübersicht und Systemeigenschaften.....	18
2 Wichtige Gebrauchshinweise	31
2.1 Bestimmungsgemäßer Gebrauch	31
2.1.1 Einführung.....	31
2.1.2 Einsatz- und Anwendungsbereiche.....	31
2.2 Nicht-bestimmungsgemäßer Gebrauch.....	32
3 Sicherheitshinweise für elektrische Antriebe und Steuerungen.....	35
3.1 Begriffsdefinitionen.....	35
3.2 Grundsätzliche Hinweise.....	36
3.2.1 Benutzung und Weitergabe der Sicherheitshinweise.....	36
3.2.2 Voraussetzungen für den sicheren Gebrauch.....	36
3.2.3 Gefahren durch falschen Gebrauch.....	38
3.3 Gefahrenbezogene Hinweise.....	38
3.3.1 Schutz gegen Berühren elektrischer Teile und von Gehäusen.....	38
3.3.2 Schutzkleinspannung als Schutz gegen elektrischen Schlag	40
3.3.3 Schutz vor gefährlichen Bewegungen.....	40
3.3.4 Schutz vor elektromagnetischen und magnetischen Feldern bei Betrieb und Montage.....	42
3.3.5 Schutz gegen Berühren heißer Teile.....	42
3.3.6 Schutz bei Handhabung und Montage.....	43
3.3.7 Schutz beim Umgang mit Batterien.....	43
3.3.8 Schutz vor unter Druck stehenden Leitungen.....	44
3.4 Erläuterung der Signalwörter und der Signalgrafik.....	45

	Seite
4 Grundfunktionen von Rexroth IndraMotion MLD	47
4.1 Übersicht	47
4.2 Begriffsklärung	47
4.3 Gerätesteuerung	49
4.3.1 Allgemeines	49
4.3.2 Führungskommunikation	49
4.3.3 Achssteuerung	49
4.3.4 Achse	49
4.3.5 SPS-Runtime	50
4.3.6 Eingriffsmöglichkeiten der MLD auf die Achssteuerung	50
4.4 Achskommandierung	53
4.4.1 Grundlegendes zur Achskontrolle	53
4.4.2 Umsetzung der Motionkommandierung in der Achse	58
4.4.3 Grundsätzliche Funktionsweise von PLCopen-konformen Motion-Funktionsbausteinen	59
4.4.4 Achsadressierung	61
4.4.5 Anwendungs- und Programmierhinweise	76
4.5 Tasksystem	78
4.5.1 Begriffserklärungen	78
4.5.2 Kurzbeschreibung	78
4.5.3 Task-Eigenschaften	80
4.5.4 Motions-task	80
4.5.5 Task-Überwachung (Watchdog)	83
4.5.6 Task-Stacküberprüfung	85
4.5.7 Laufzeitmessungen	85
4.5.8 Task-Konfiguration	89
4.5.9 Task-Zykluszeiten und Timing	89
4.6 Fehlerbehandlung der IndraMotion MLD	91
4.6.1 Betriebsverhalten der IndraMotion MLD	91
4.6.2 Allgemein (für MLD-S und MLD-M)	92
4.6.3 Konfiguration der Fehlerreaktion bei MLD-S	97
4.6.4 Fehlerbehandlung und -Reaktion bei MLD-M	98
5 Kommunikationsschnittstellen und Datenkanäle der MLD	105
5.1 Einleitung und Übersicht	105
5.2 Datenkanäle der IndraMotion MLD	107
5.2.1 Zyklische Datenkanäle	107
5.2.2 Azyklische Datenkanäle / Schnittstellen	126
5.2.3 Konfiguration der Ein- und Ausgänge (E/A-Konfiguration der MLD)	141
5.2.4 Zyklische Daten im MLD-M-Systemmodus	151
5.3 Sercos I/O	156
6 Visualisierung, Engineering und Anbindung über OCI	159
6.1 Übersicht	159
6.2 Schnittstellen zur Anbindung von Kommunikationspartnern	161
6.2.1 Ethernet Schnittstelle (TCP/IP)	161

	Seite
6.2.2	Führungskommunikations-Schnittstelle..... 163
6.3	Anbindung von IndraControl V..... 164
6.3.1	Einleitung und Übersicht..... 164
6.3.2	Konfiguration der Symbole für Variablen-/ Parameterzugriff..... 166
6.3.3	Anbindung eines IndraControl VE* über Ethernet..... 167
6.4	Open Core Interface für Antriebe..... 170
6.4.1	Zielplattformen..... 172
6.4.2	Unterstützte Geräte..... 172
6.4.3	Anwendungsmöglichkeiten..... 172
6.4.4	SDK: Voraussetzungen und Installation..... 174
6.4.5	SDK: Inhalte..... 174
6.4.6	Anwendungsbeispiele..... 177
7	Inbetriebnahme- und Anwendungshinweise..... 189
7.1	Voraussetzungen für die Nutzung von Rexroth IndraMotion MLD..... 189
7.1.1	Firmware- und Hardware-Voraussetzungen..... 189
7.1.2	Software-Voraussetzungen..... 189
7.1.3	Voraussetzungen für die Kommunikation 189
7.2	Programmerstellung in IndraWorks MLD..... 190
7.2.1	Nutzung von Technologiefunktionen..... 191
7.2.2	Freie Programmierung von Rexroth IndraMotion MLD..... 194
7.2.3	Nutzung des "AxisInterface" für die Programmierung von Rexroth IndraMotion MLD..... 207
7.2.4	Automatisierte Programmerstellung mit "GAT compact"..... 208
7.3	SPS-Projektaufbau..... 213
7.3.1	Allgemeines..... 213
7.3.2	Projektablage im Antrieb..... 215
7.3.3	Bootprojekt..... 216
7.3.4	Dateisystem im Antrieb..... 217
7.3.5	Externe Speicherkarte als Speichermedium..... 218
7.3.6	IndraWorks-Package..... 219
7.3.7	Bibliotheken..... 219
7.4	Arbeitsschritte während der Programmierung und Inbetriebnahme..... 220
7.4.1	Neues Projekt anlegen..... 220
7.4.2	SPS-Projekt importieren..... 221
7.4.3	IndraLogic-Projekt (*.project-Datei) aus IndraWorks MLD exportieren..... 222
7.4.4	IndraLogic-Projekt in den Antrieb laden..... 222
7.4.5	IndraWorks-Projekt archivieren / wiederherstellen..... 224
7.4.6	Quellcode laden..... 224
7.4.7	Projekt in der Steuerung öffnen..... 224
7.4.8	MLD-Projekt archivieren..... 225
7.4.9	Exportiertes MLD-Projekt importieren..... 234
7.4.10	Update der IndraLogic-Geräteversion..... 236
7.4.11	Ändern der für ein SPS-Projekt gültigen Bibliotheken 236
7.4.12	Import einer externen Bibliotheksdatei..... 237
7.5	Antrieb als SPS-Gerät..... 238
7.5.1	Allgemeines..... 238

	Seite
7.5.2 Sercos III Antrieb als SPS-Gerät.....	239
7.6 Versorgungsgerät als SPS-Gerät.....	248
7.6.1 Allgemeines.....	248
7.6.2 Parametrierung.....	248
7.6.3 Konfiguration der zyklischen Daten.....	249
7.6.4 SPS-Programmierung.....	252
7.7 Vergleich der Antriebsfamilie "IndraDrive Cs" mit der Antriebsfamilie "IndraDrive C"/"IndraDrive M".....	254
7.7.1 Einleitung.....	254
7.7.2 Physikalische Schnittstellen.....	254
7.7.3 Byte-Reihenfolge.....	254
7.7.4 MLD-Leistungsvergleich.....	258
8 Programmierhinweise.....	261
8.1 Industrie-Standards für die Programmierung	261
8.1.1 Allgemeines.....	261
8.1.2 IEC 61131.....	261
8.1.3 PLCopen.....	261
8.2 Bibliotheken für Rexroth IndraMotion MLD.....	263
8.2.1 Allgemeine Eigenschaften der Bibliotheken.....	263
8.2.2 Bibliotheken.....	263
8.3 Retaindaten sichern und wieder herstellen.....	267
8.3.1 Retaindaten manuell sichern und wieder herstellen.....	267
8.3.2 Persistente Variablen bei "Reset Ursprung" sichern und bei SPS-Neustart mit gesicherten persistenten Daten initialisieren.....	271
8.4 Dateizugriffe auf die optionale Speicherkarte.....	275
8.5 Startverhalten und Bootprojekt.....	276
8.5.1 Allgemeines.....	276
8.5.2 Kurzbeschreibung.....	276
8.5.3 Anwendungs- und Programmierhinweise.....	278
8.6 Richtlinien für die Programmierung mit IndraLogic.....	278
8.6.1 Übersicht.....	278
8.6.2 Strukturierung von SPS-Projekten.....	278
8.6.3 Programmiersprachen.....	279
8.6.4 Globale Daten.....	280
8.6.5 Programm-Header.....	281
8.6.6 History.....	282
8.6.7 Typbezeichner.....	282
8.6.8 Instanzbezeichner.....	285
8.6.9 Definition von Standardschnittstellen an Funktionsbausteinen.....	287
8.6.10 Fehlerhandling.....	291
8.6.11 Versionierung von Bibliotheken.....	291
8.6.12 Codeoptimierung.....	292
8.7 VCS – Version Control System.....	294

	Seite
9 Diagnose- und Servicefunktionen.....	297
9.1 Übersicht und Einleitung.....	297
9.2 Geräte-Editor.....	297
9.2.1 Geräte-Editor, Allgemeines.....	297
9.2.2 Status.....	297
9.2.3 Dateien.....	298
9.2.4 Taskaufstellung.....	298
9.2.5 Log.....	298
9.2.6 SPS-Einstellungen.....	300
9.2.7 SPS Shell.....	301
9.2.8 Information.....	301
9.3 Diagnosefunktionen.....	302
9.3.1 Standard-Diagnosemöglichkeiten des Antriebs.....	302
9.3.2 Diagnosefunktionen der MLD.....	303
9.4 Servicefunktionen.....	307
9.4.1 Servicefunktionen der SPS (IndraMotion MLD).....	307
9.4.2 Firmware-Tausch.....	308
9.4.3 Firmware-Versions-Upgrade.....	309
10 Projektkonvertierung.....	313
10.1 Einleitung.....	313
10.2 Voraussetzungen in MLD-1G.....	313
10.3 Einfügen in MLD-2G.....	315
10.3.1 Projekt einfügen.....	315
10.3.2 Übersetzungsfehler nach der Projektkonvertierung.....	316
10.3.3 Anpassungen in MLD-2G.....	317
10.4 Projektkonvertierung MPx18 --> MPx20.....	318
11 Service und Support.....	319
Index.....	321

1 Einleitung

1.1 Über diese Dokumentation

Ausgaben dieser Dokumentation

Ausgabe	Stand	Bemerkung
05	2019-05	- Allgemeine Korrekturen, Ergänzungen und Verbesserungen - Auflistung der Dokumentationen zu den MLD-Technologiefunktionen aufgenommen
04	2018-11	- Ab MPB-21 ist der interne Codespeicher (RAM im Steuerteil) auf 1,2 MB reduziert (siehe "Kap. "Mengengerüst (Speicher)" auf Seite 20") "Sequential Motion Control (SMC)" erwähnt - Allgemeine Korrekturen, Ergänzungen und Verbesserungen
03	2017-06	- Erweiterung um "IndraDrive ML" (Firmware "PSB") - Erweiterung um MPx-19 - Erweiterungen und Änderungen zur MPx-20 - Neue Bibliotheken (siehe "Unterstützte Bibliotheken") - VmAxis1, RmAxis1, LinkAxis1 sind bei den Synchronisationsbausteinen auch bei MLD-S als Masterachse verwendbar (siehe "Achsadressierung") - Beschreibung zu "AxisInterface" eingefügt - Beschreibung zu "GAT compact" eingefügt - Beschreibung zu "Version Control System (VCS)" eingefügt - Kurzbeschreibung zu "IndraControl S20" und zu "Antrieb oder Frequenzumrichter als Sercos Peripherie" eingefügt - Beschreibungen zu "Antrieb als SPS-Gerät" und zu "Versorgungsgeräte als SPS-Gerät" eingefügt - Beschreibungen zu "Open Core Interface für Antriebe" eingefügt
02	2014-08	- Korrekturen - Angaben zum verfügbaren Speicher - Ergänzungen: - Notwendige Einstellung in WinStudio bei Anbindung eines HMI (IndraControl VE*) über Ethernet - Beschreibung des Kontextmenüs von "Application" - Hinweis darauf, dass wenn bei MPC Systemmode "MLD-M" konfiguriert ist, Performancelevel "Advanced" nicht eingestellt werden kann
01	2014-01	Erstausgabe

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...
Wichtige Sachverhalte, die im Fließtext hervorgehoben werden sollen	Fettschrift	Bei "IndraDrive Cs" steht ein analoger Eingang zur Verfügung. Ein analoger Ausgang existiert nicht.
Parameternamen, Diagnosenamen, Funktionsbezeichnungen	Anführungszeichen	Die Eingänge wurden in der E/A-Konfiguration (IndraWorks) mit dem Parameter "P-0-1390, SPS Eingang WORD0" verknüpft.

Tab. 1-2: Konventionen in der Schreibweise

Hinweise und Tipps sind im Text besonders hervorgehoben. Sie sind jeweils mit einem Symbol versehen, das Ihnen sagt, um welche Art es sich handelt:



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



Mit dieser Darstellung werden nützliche Tipps und Tricks gekennzeichnet.

Ihr Feedback

Auf Gefahren wird gemäß ANSI Z535.6-2011 aufmerksam gemacht (siehe "[Erläuterung der Signalwörter und der Signalgrafik](#)").

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 auf der Rückseite der Titelseite unter "Interner Ablagevermerk" angegeben ist.
- Die Seitenzahl.

1.2 Aufbau und Struktur der Dokumentation

Der Aufbau der Dokumentation ist zielgruppenorientiert gestaltet, um eine möglichst einfache und schnelle Navigation zu erreichen. Diese Dokumentation ist in folgende Hauptkapitel gegliedert, welche jeweils auf eine definierte Zielgruppe bzw. einen Verwendungsfall zugeschnitten sind:

- **Kap. 1.5 "Produktvorstellung" auf Seite 9**
Liefert einen Überblick bzgl. der Hauptmerkmale und des Verwendungszwecks der "IndraMotion MLD"
- **Kap. 7 "Inbetriebnahme- und Anwendungshinweise" auf Seite 189**
Hinweise für die Inbetriebnahme und die Verwendung der "IndraMotion MLD"
- **Kap. 9 "Diagnose- und Servicefunktionen" auf Seite 297**
Tritt ein Problem in Verbindung mit der "IndraMotion MLD" auf, erläutert dieses Kapitel die Möglichkeiten der Fehlersuche und Fehlerbehebung
- **Kap. 10 "Projektkonvertierung" auf Seite 313**
In diesem Kapitel wird die Konvertierung von IndraLogic 1.x-Projekten (MLD-1G) in IndraLogic 2G-Projekte (MLD-2G) beschrieben

1.3 Dokumentationen

1.3.1 Übersicht

Die Beschreibung der Funktionalität der "IndraMotion MLD" ist wie folgt gegliedert:

- **Anwendungsbeschreibung für Rexroth IndraMotion MLD-2G** (diese Dokumentation)
Beschreibt die für "IndraMotion MLD" spezifischen Funktionalitäten und erläutert die Eingliederung der SPS in den Antrieb sowie die technischen Möglichkeiten und Funktionen
- **Erste Schritte mit Rexroth IndraMotion MLD-2G**
Beschreibt die erforderlichen Schritte für die erste Nutzung von Rexroth IndraMotion MLD (Vorbereitung und Aufbau, Inbetriebnahme, Programmerstellung)
- **Bibliothek für Rexroth IndraMotion MLD-2G**
 - Beschreibt die für "IndraMotion MLD" speziell erstellten Funktionen bzw. Funktionsbausteine, welche in einer speziellen Bibliothek zusammengefasst sind
 - Enthält Verweise auf die von mehreren System genutzten **Systembibliotheken**
 - Enthält Verweise auf die von mehreren System genutzten **Basisbibliotheken**



Über die Online-Hilfe in IndraWorks haben Sie komfortablen Zugriff auf die Beschreibungen der System- und Basisbibliotheken.

- **MLD-Technologiefunktionen**
MLD-Technologiefunktionen werden im Antrieb mit Hilfe der antriebsintegrierten SPS (IndraMotion MLD) realisiert. Technologiefunktionen werden allgemein als kompiliertes SPS-Projekt (*.par- Dateien) zur Verfügung gestellt, das die Technologiefunktionen ähnlich wie eine Firmware-

funktion bereitstellt. Für die Nutzung der Technologiefunktionen, die als kompiliertes SPS-Projekt zur Verfügung gestellt werden, sind keine Programmierkenntnisse erforderlich.

Für jede MLD-Technologiefunktion existiert eine eigene Dokumentation.

- **Bedien- und Programmieranleitung** für IndraLogic 2G

In dieser Dokumentation wird speziell die Programmierumgebung, sowie die Programmierung bzw. Erstellung von SPS-Programmen beschrieben

- **Parameterbeschreibung** für IndraDrive

In der Parameterbeschreibung sind neben den SPS-spezifischen Parametern auch alle übrigen Antriebsparameter dokumentiert

- **Diagnosebeschreibung** für IndraDrive

In der Diagnosebeschreibung (auch "Hinweise zur Störungsbeseitigung") sind neben den SPS-spezifischen Diagnosen auch alle übrigen Antriebsdiagnosen dokumentiert

1.3.2 Antriebssysteme, Systemkomponenten

Titel	Dokumentationsart	Dokumentations-Type ¹⁾	Materialnummer R911...
Rexroth IndraControl VR 21 Bedienpanel	Betriebsanleitung	DOK-SUPPL*-VR21**-.01*-ITxx-DE-P	339475
IndraControl VH 2110.01 Handbediengerät	Betriebsanleitung	DOK-SUPPL*-VH*21**-.01*-ITxx-DE-P	346749
Rexroth IndraControl VEP / VEH	Projektierungsanleitung	DOK-SUPPL*-VEH/VEP****-PRxx-DE-P	309681
Rexroth IndraControl S20	Betriebsanleitung	DOK-CONTRL-S20*SYS*INS-APxx-DE-P	335987
Die Automatisierungsklemmen der Produktfamilie Rexroth Inline	Anwendungsbeschreibung	DOK-CONTRL-ILSYSINS***-AWxx-DE-P	317017

1) In den Dokumentations-Typen ist "xx" ein Platzhalter für den aktuellen Ausgabestand der Dokumentation (Beispiel: AW01 bedeutet die erste Ausgabe einer Anwendungsbeschreibung)

Tab. 1-3: Dokumentationen – Übersicht

1.3.3 Motoren

Titel	Dokumentationsart	Dokumentations-Type ¹⁾ DOK-MOTOR*-...	Materialnummer R911...
MAD / MAF Asynchronmotoren MAD / MAF	Projektierungsbeschreibung	MAD/MAF****-PRxx-DE-P	295054
MBS-H Synchron-Bausatz-Spindelmotoren	Projektierungsbeschreibung	MBS-H*****-PRxx-DE-P	297894
MLF Synchron-Linearmotoren	Projektierungsbeschreibung	MLF*****-PRxx-DE-P	293634

Einleitung

Titel	Dokumentationsart	Dokumentations-Type ¹⁾ DOK-MOTOR*... R911...	Materialnummer
MCL Eisenlose Linearmotoren MCL	Projektierungsbeschreibung	MCL*****-PRxx-DE-P	330591
MKE Synchronmotoren Synchron-Servomotoren für explosionsgefährdete Bereiche nach ATEX und UL / CSA	Projektierungsbeschreibung	MKE*GEN2***-PRxx-DE-P	297662
MSK Synchron-Servomotoren	Projektierungsbeschreibung	MSK*****-PRxx-DE-P	296288
MSK Synchron-Servomotoren für explosionsgefährdete Bereiche	Projektierungsbeschreibung	MSK*EXGIIK3-PRxx-DE-P	312708
MSM Synchron-Servomotoren	Datenblatt	MSM*****-DAxx-DE-P	329337
MS2E Synchron-Servomotoren nach ATEX-Richtlinie 2014/34/EU	Projektierungsbeschreibung	MS2E*****-PR01-DE-P	394139
MS2N Synchron-Servomotoren	Projektierungsbeschreibung	MS2N*****-PRxx-DE-P	347582
MBT Synchron-Torquemotoren	Projektierungsbeschreibung	MBT*****-PRxx-DE-P	291224

1) In den Dokumentations-Typen ist "xx" ein Platzhalter für den aktuellen Ausgabestand der Dokumentation (Beispiel: PR01 bedeutet die erste Ausgabe einer Projektierungsbeschreibung)

Tab. 1-4: Dokumentationen – Motoren

1.3.4 Kabel

Titel	Dokumentationsart	Dokumentations-Type ¹⁾ DOK-CONNEC-... R911...	Materialnummer
Rexroth Anschlusskabel IndraDrive und IndraDyn	Auswahldaten	CABLE*INDRV-CAxx-DE-P	322948

1) In den Dokumentations-Typen ist "xx" ein Platzhalter für den aktuellen Ausgabestand der Dokumentation (Beispiel: CA02 bedeutet die zweite Ausgabe der Dokumentation "Auswahldaten")

Tab. 1-5: Dokumentationen – Kabel

1.3.5 Firmware

Titel	Dokumentationsart	Dokumentations-Type ¹⁾ DOK-INDRV*-...	Materialnummer R911...
IndraDrive MPx-21 Funktionen	Anwendungsbeschreibung	MP*-21VRS**-APxx-DE-P	385759
IndraDrive MPx-20 Funktionen	Anwendungsbeschreibung	MP*-20VRS**-APxx-DE-P	345607
IndraDrive MPx-20 Version Notes	Release Notes	MP*-20VRS**-RNxx-DE-P	345605
IndraDrive Power Supply Basic PSB-21 Funktionen	Anwendungsbeschreibung	PSB-21VRS**-APxx-DE-P	385755
IndraDrive Power Supply Basic PSB-21 Version Notes	Release Notes	PSB-21VRS**-RNxx-DE-P	385753
IndraDrive Power Supply Basic PSB-20 Funktionen	Anwendungsbeschreibung	PSB-20VRS**-APxx-DE-P	345609
Rexroth IndraDrive Power Supply Basic PSB-19 Funktionen	Anwendungsbeschreibung	PSB-19VRS**-APxx-DE-P	345601
Rexroth IndraDrive MPx-18 Funktionen	Anwendungsbeschreibung	MP*-18VRS**-APxx-DE-P	338674
Rexroth IndraDrive MPx-18 Version Notes	Release Notes	MP*-18VRS**-RNxx-DE-P	338657
Rexroth IndraDrive MPx-17 Funktionen	Anwendungsbeschreibung	MP*-17VRS**-APxx-DE-P	331235
Rexroth IndraDrive MPx-17 Version Notes	Release Notes	MP*-17VRS**-RNxx-DE-P	331587
Rexroth IndraDrive MPx-16 Funktionen	Anwendungsbeschreibung	MP*-16VRS**-APxx-DE-P	326484

Einleitung

Titel	Dokumentationsart	Dokumentations-Type ¹⁾ DOK-INDRV*-...	Materialnummer R911...
Rexroth IndraDrive MPx-16 Version Notes	Release Notes	MP*-16VRS**-RNxx-DE-P	329271
IndraDrive MPx-16 bis MPx-21 und PSB Parameter	Referenz	GEN1-PARA**-RExx-DE-P	328650
IndraDrive MPx-16 bis MPx-21 und PSB Diagnosen	Referenz	GEN1-DIAG**-RExx-DE-P	326539
Rexroth IndraDrive Integrierte Sicherheitstechnik "Safe Torque Off" (ab MPx-16)	Anwendungsbeschreibung	SI3-**-VRS**-APxx-DE-P	332633
IndraDrive Integrierte Sicherheitstechnik "Safe Motion" (ab MPx-18)	Anwendungsbeschreibung	SI3*SMO-VRS-APxx-DE-P	338919
Rexroth IndraDrive Rexroth IndraMotion MLD Bibliotheken ab MPx-17	Referenz	MLD-SYSLIB2-RExx-DE-P	332626
IndraDrive Rexroth IndraMotion MLD Bibliotheken ab MPx-18	Referenz	MLD-SYSLIB3-RExx-DE-P	338915
Rexroth IndraDrive Rexroth IndraMotion MLD ab MPx-17	Anwendungsbeschreibung	MLD2-**-VRS*-APxx-DE-P	334350
IndraDrive IndraMotion MLD ab MPx-18	Anwendungsbeschreibung	MLD3-**-VRS*-APxx-DE-P	338913

1)

In den Dokumentations-Typen ist "xx" ein Platzhalter für den aktuellen Ausgabestand der Dokumentation (Beispiel: RE02 bedeutet die zweite Ausgabe einer Referenz-Dokumentation)

Tab. 1-6:

Dokumentationen – Firmware

1.3.6 MLD-Technologiefunktionen

Titel	Dokumentationsart	Dokumentations-Type ¹⁾	Materialnummer
MLD-Technologiefunktion "Prozess-regler"	Anwendungsbeschreibung	DOK-MLD***-TECFUNC*PC*-APxx-DE-P	R911397136
MLD-Technologiefunktion "Schwingungsdämpfung und -vermeidung"	Anwendungsbeschreibung	DOK-MLD***-TECFUNC*VDA-APxx-DE-P	R911397134
MLD-Technologiefunktion "Kinetic Buffering"	Anwendungsbeschreibung	DOK-MLD***-TECFUNC*KB*-APxx-DE-P	R911397138

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

Tab. 1-7: Dokumentationen – MLD-Technologiefunktionen

1.4 Mitgeltende Dokumentationen

Titel	Dokumentationsart	Dokumentations-Type ¹⁾	Materialnummer
IndraWorks 14VRS Basis-Bibliotheken IndraLogic 2G	Bibliotheksbeschreibung	DOK-IL*2G*-BASLIB**V14-LIxx-DE-P	R911343918
IndraWorks 14VRS SPS-Programmiersystem IndraLogic 2G	Anwendungsbeschreibung	DOK-IWORKS-IL2GPRO*V14-APxx-DE-P	R911343570
IndraMotion for Metal Forming 14VRS Sequential Motion Control	Referenz	DOK-IM*MF*-TF*SMC**V14-RExx-DE-P	R911343864
Rexroth IndraMotion for Handling 13VRS Funktionsbeschreibung	Referenz	DOK-IM*HA*-FUNC****V13-RExx-DE-P	R911343860

1) In den Dokumentations-Typen ist "xx" ein Platzhalter für den aktuellen Ausgabestand der Dokumentation (Beispiel: RE02 bedeutet die zweite Ausgabe einer Referenz-Dokumentation)

Tab. 1-8: Mitgeltende Dokumentationen

1.5 Produktvorstellung

1.5.1 Einleitung und Übersicht

Bei "IndraMotion MLD" (**M**otion **L**ogic **D**rive-based) handelt es sich um eine im Antrieb integrierte SPS, die als Firmware-Funktionspaket für die Antriebsfamilie "Rexroth IndraDrive" angeboten wird.

Mit "IndraMotion MLD" verschmelzen Motion- und SPS-Funktionen zu einer modernen, offenen Automatisierungsplattform für modulare Maschinenkonzepte. Die dezentrale Steuerungsarchitektur bildet ein kompaktes Motion-Logic-System auf Basis der skalierbaren Antriebsplattform IndraDrive und macht übergeordnete Steuerungen überflüssig.

- Elektronische Synchronisierung von bis zu 10 Servoachsen
- Unterstützung von bis zu 4 Sercos-III-I/O-Teilnehmern
- Visualisierung über "IndraControl Vxx"

- Anwendungen: Bandanlagen, Walzenvorschübe, Vorschubrichtmaschinen, Abwickler, Richtmaschinen

"IndraMotion MLD" gibt es in den Ausprägungen MLD-S (Einzelachs-"Motion Logic Control") und MLD-M (Mehrachs-"Motion Logic Control").

	IndraMotion MLD (Antriebsintegrierte SPS)	
	IndraMotion MLD-S: Einzelachs-"Motion Logic Control"	IndraMotion MLD-M: Mehrachs-"Motion Logic Control" (nur Firmware "MPC")
"IndraDrive C" / "IndraDrive M" (mit Steuerteilen Cxx02)	X	X
"IndraDrive Cs"	X	X
"IndraDrive Mi"	X	-
"IndraDrive ML"	X	X



In diesem Dokument wird die Funktionalität und die Verwendung der antriebsintegrierten SPS "IndraMotion MLD" mit den Geräten des Typs "IndraDrive Cs" beschrieben.

1.5.2 Kurzbeschreibung

"IndraMotion MLD" ist eine Ausprägung der "IndraMotion"-Familie, die durch die Einbettung in den Antriebsregelgeräten der Antriebsfamilie "Rexroth IndraDrive" gekennzeichnet ist. Es ist hierfür keine zusätzliche Hardwareoption, sondern lediglich eine Lizenzierung eines Firmware-Funktionspakets erforderlich ["ML", "MA" oder "TF" (ab MPx-20)].

Siehe auch Firmware-Funktionsbeschreibung "Funktionspakete"

Weitere Ausprägungen der "IndraMotion"-Familie:

- IndraMotion MLC
- IndraMotion MTX



Alle IndraMotion-Varianten können mit dem Engineeringtool "IndraWorks Engineering" konfiguriert und bedient werden.

Wird nur die IndraMotion-Variante "IndraMotion MLD" verwendet, dann steht dafür das Engineeringtool "IndraWorks MLD" zur Verfügung.

Die **Programmierung** von IndraMotion ermöglicht das in IndraWorks eingebettete Programmierwerkzeug "IndraLogic".

Die mit "IndraLogic" erstellten Programme sind portierbar, sofern die Randbedingungen eingehalten werden (siehe "[Programmierhinweise](#)").

Hauptmerkmale von "IndraMotion MLD"

Nachfolgend sind die wesentlichen Merkmale von "IndraMotion MLD" aufgeführt:

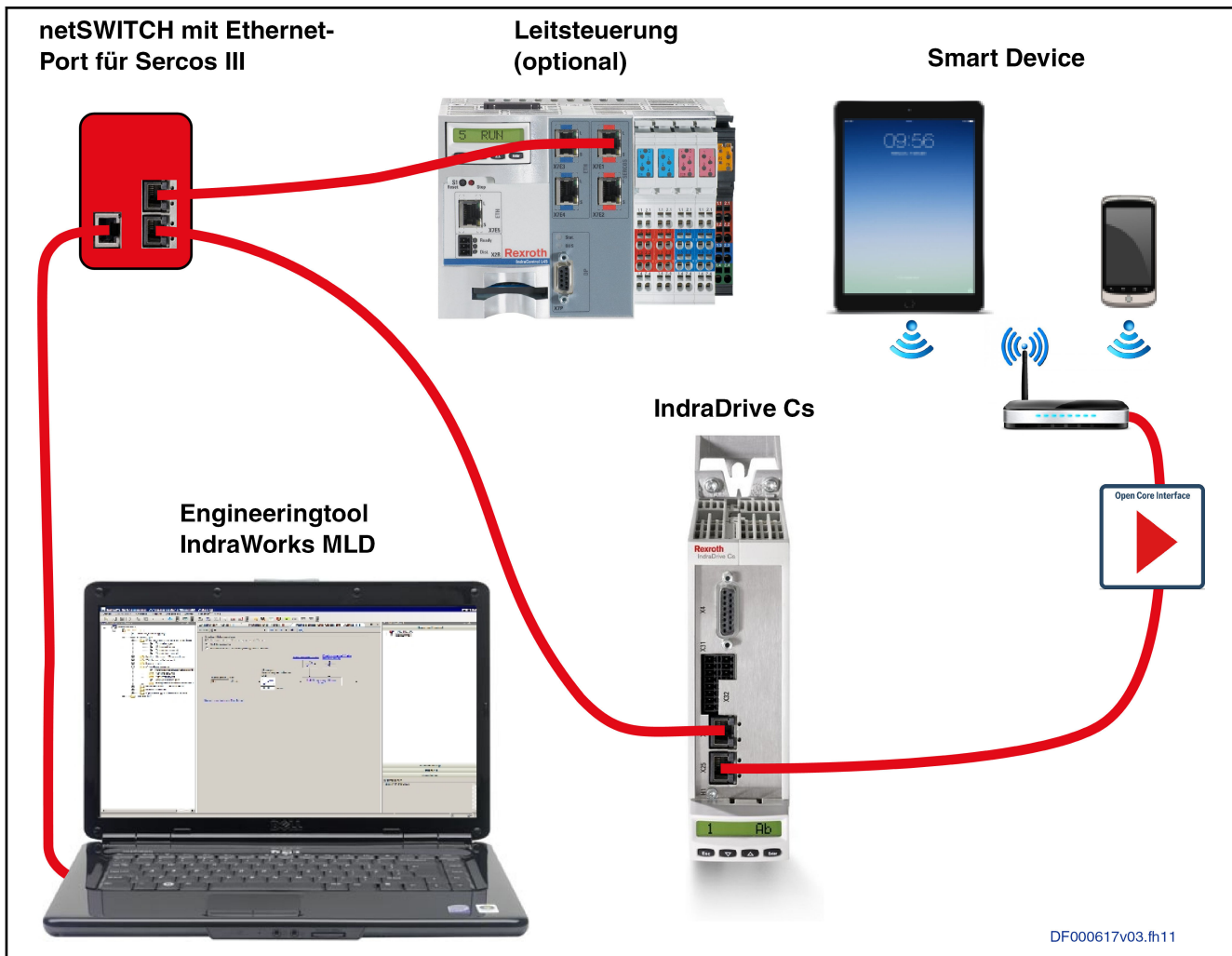
- "IndraMotion MLD" ist ideal, um wirtschaftlich Bewegungs- und Logik-Aufgaben mit Firmware-Funktionen des Antriebs zu lösen (es wird keine zusätzliche Hardware benötigt)
- "Offener Antrieb", d.h. direkter und transparenter Zugriff auf alle Antriebsfunktionen und Parameter

- **Open Core Interface (OCI):** ermöglicht den Einsatz eigener, echtzeitfähiger Anwendungen direkt auf der Steuerung oder individueller IT-Anwendungen auf externen Geräten (ab Firmware-Version 20)
- Programmierung von Funktionserweiterungen oder eigenen Motionlösungen nach **IEC 61131-3** auf Basis umfangreicher Standard- und Zusatzbibliotheken ("Technologie-Bibliotheken")
- Ab Firmware-Version 20: Vereinfachte Motion-Programmierung mit Hilfe von "AxisInterface" und "GAT compact"
- Flexible Kommunikationserweiterung durch On-Board-Schnittstellen

"IndraMotion MLD" gibt es in den folgenden Ausprägungen:

- **IndraMotion MLD-S** (Einzelachslösung; Firmware "MPB" und "MPC")
 - Als intelligente Servoachse (Erweiterung der Antriebsfunktionalität)
 - Als autarke Einzelachs-"Motion Logic Control"
- **IndraMotion MLD-M** (Mehrachslösung; nur Firmware "MPC")
 - Als autarke Mehrachs-"Motion Logic Control" unter Verwendung von CCD [Cross Communication Drives (Antriebs-Querkommunikation)] auf Basis von Sercos III

1.5.3 IndraMotion MLD-S



Anschluss Über welche Schnittstellen die Komponenten miteinander verbunden werden, entnehmen Sie bitte den jeweiligen Projektierungen

Abb. 1-1: Grundsätzliche Systemstruktur von "IndraMotion MLD-S"

Merkmale von "IndraMotion MLD-S"

"IndraMotion MLD-S" zeichnet sich durch die folgenden Hauptmerkmale aus:

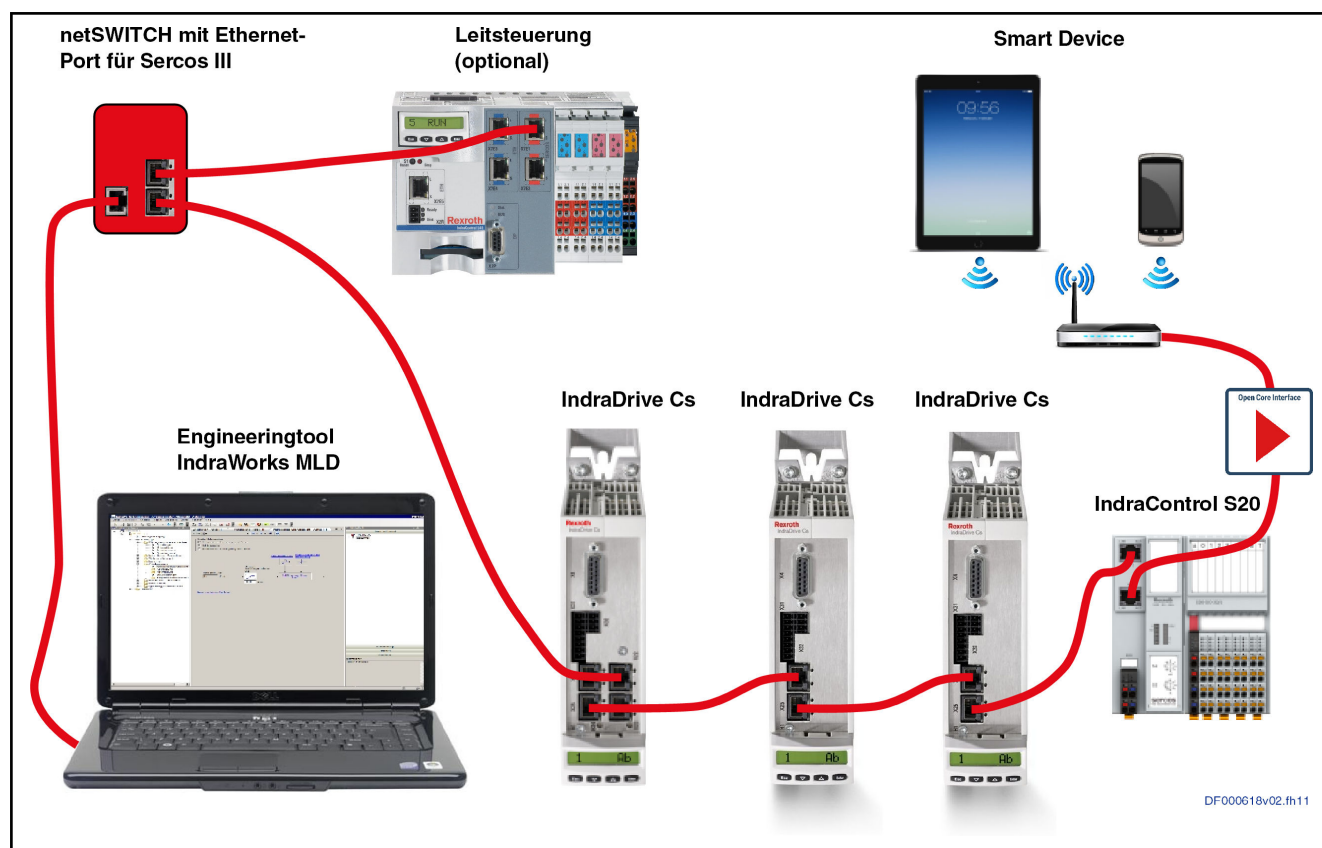
- Ausprägung "Intelligente Servoachse" (Erweiterung der Antriebsfunktionalität) oder "Autarke Einzelachs-"Motion Logic Control"
- Kommandierung der lokalen Achse mit direktem Durchgriff auf die Gerätesteuerung des Antriebs
- Direkter Zugriff auf alle Parameter des Antriebs über systemglobale SPS-Variablen, Funktionen, Funktionsbausteine oder einen konfigurierbaren Synchronkanal
- Direkter Zugriff auf die digitalen Ein-/Ausgänge und den analogen Eingang des Antriebs (lokale Achse)

In der Ausprägung "Autarke Einzelachs-"Motion Logic Control" zusätzlich:

- Motion-Control-Funktionalität für 1 Achse durch Kommandierung der lokalen Achse durch entsprechende Motion-Control-Bibliothek nach PLCopen

1.5.4 IndraMotion MLD-M

Bei "IndraMotion MLD-M" werden neben der lokalen Achse auch entfernte Achsen (so genannte "CCD-Slaves") bedient.



Anschluss Über welche Schnittstellen die Komponenten miteinander verbunden werden, entnehmen Sie bitte den jeweiligen Projektierungen

Abb. 1-2: Grundsätzliche Systemstruktur von "IndraMotion MLD-M"

Merkmale von "IndraMotion MLD-M"

"IndraMotion MLD-M" zeichnet sich zusätzlich durch die folgenden Hauptmerkmale aus:

- Nur verfügbar mit der Firmware "MPC"
- Motion-Control-Funktionalität für bis zu 10 Achsen durch entsprechende Motion-Control-Bibliothek nach PLCopen
- CCD-Querkommunikationsschnittstelle zu den "entfernten Achsen" auf Basis von Sercos III
 - Zur Motion-Kommandierung der lokalen und entfernten Achsen
 - Zur Anbindung weiterer Sercos-Peripherie-Geräte ("Rexroth Inline", "Rexroth IndraControl S20", Antrieb, Frequenzumrichter) (siehe "[Erweiterungen der MLD-M](#)")
- Direkter Zugriff auf die digitalen und analogen Ein-/Ausgänge der Antriebe (lokale und entfernte Achsen)
- Zugriffsmöglichkeit auf alle Parameter der Antriebe (lokale und entfernte Achsen)
- Sercos-III-Schnittstelle zu den Slaveachsen:

- **Achszahl und Zykluszeit**
max. 9 Slaves (T=500 µs bei 1 Slave, T=1000 µs bei 2 Slaves bis T=4000 µs bei 9 Slaves)
- zyklischer Datenkanal (MDT, AT) mit maximal je 48 Byte und 16 Parameter
- Parameter- bzw. Servicekanal (Infocontainer je 4 Byte)

1.5.5 Erweiterungen der MLD-M

Rexroth Inline

Mit der Produktfamilie "Rexroth Inline" werden digitale und analoge E/A-Erweiterungen in zwei Varianten angeboten:

- Modular-IOs: modular anreihbare I/O-Module ("Rexroth Inline Modular IO")
- Block-IOs: kompakte Remote-I/O-Module ("Rexroth Inline Block IO")

Die IO-Module "Rexroth Inline Block IO" und "Rexroth Inline Modular IO" können im Mischbetrieb eingesetzt werden. Die Gesamtanzahl von 4 Block-IO-Modulen und Modular-IO-Modulen (Buskoppler) darf nicht überschritten werden.

Der Betrieb der I/O-Module ist nur dann möglich, wenn folgendes gegeben ist:

- Die I/O-Module können nur in Verbindung mit einem MLD-M-System (Advanced-Gerät mit MPC-Firmware) betrieben werden.
- Ein Firmware-Funktionspaket für "IndraMotion MLD" muss zum Betrieb freigeschaltet sein.
- Die I/O-Module müssen im Projekt korrekt konfiguriert sein.
- Das MLD-Projekt wurde in die Steuerung geladen; d. h. die I/O-Konfiguration ist im MLD-Projekt enthalten.
- Der Sercos Ring ist in Sercos Phase 4. (Eine Aktualisierung der IOs ist nur in Sercos Phase 4 möglich.)



In der MLD-2G können die Parameter der I/O-Module zur Diagnose über den Servicekanal ausgelesen werden, es ist aber nicht möglich die Prozessdaten der I/O-Module in die zyklische Kommunikation der Führungskommunikation zu übernehmen. Im zyklischen Betrieb können die I/O-Module nur über die MLD gelesen/geschrieben werden.



Die an den I/O-Modulen "Inline Block I/O digital fast" anliegenden Signale werden in der Regel über eine CCD-synchrone Task geschrieben/gelesen, dies erfordert die Konfiguration des MLD-M-Systemmodus (P-0-1800.0.1, CCD: Konfiguration).

"Rexroth Inline Block IO" in Verbindung mit "IndraMotion MLD"

"Rexroth Inline Block IO" ist die ideale Lösung für Anwendungen mit begrenztem E/A-Umfang und geringer Komplexität. Durch die kompakte Bauweise eignen sie sich als platzsparende Lösung zum Einbau in flache Schaltkästen.

Maximal 4 "Rexroth Inline Block IO" können an einem Antriebsregelgerät betrieben werden.

Die Verdrahtung erfolgt über den aufrastbaren Inline-Anschluss-Stecker. Durch die Mehrleiter-Anschlussstechnik kann auf zusätzliche Klemmenebenen verzichtet werden.

Für die Sercos-III-Anschaltung existieren verschiedene Varianten von "Rexroth Inline Block IO":

- **Inline Block IO mit 16 digitalen Eingängen und 16 konfigurierbaren digitalen Ein- oder Ausgängen** (R-ILB S3 24 DI16 DIO16)
Dokumentation: "Rexroth Inline Block IO für Sercos III mit digitalen Ein- und Ausgängen", DOK-CONTRL-S3DI16DIO16-KB, R911170499
- **Inline Block IO mit 4 analogen Eingängen und 2 analogen Ausgängen** (R-ILB S3 24 AI4 AO2)
Dokumentation: "Rexroth Inline Block IO für Sercos III mit analogen Ein- und Ausgängen", DOK-CONTRL-S3AI4AO2***-KB, R911170557
- **Inline Block IO mit 16 digitalen Highspeed-Eingängen und 16 digitalen Highspeed-Ein- oder Ausgängen, Timestamp und Oversampling** (R-ILB S3 24 DI16 DIO16/F)
Dokumentation: In Vorbereitung
- **Inline Block IO mit 6 analogen Highspeed-Eingängen und 4 analogen Highspeed-Ausgängen, Oversampling** (R-ILB S3 AI6 AO4/F)
Dokumentation: In Vorbereitung

"R-ILB S3 24 DI16 DIO16" / "R-ILB S3 24 DI16 DIO16/F"

Die Inline Block IO "R-ILB S3 24 DI16 DIO16" ist zum Einsatz innerhalb Sercos III vorgesehen. Sie dient zur Erfassung von digitalen Eingangssignalen und zur Ausgabe digitaler Signale. "R-ILB S3 24 DI16 DIO16" verfügt über 16 digitale Eingänge und 16 frei konfigurierbare digitale Ein- oder Ausgänge.

"R-ILB S3 24 DI16 DIO16/F" verfügt ebenfalls über 16 digitale Eingänge und 16 frei konfigurierbare digitale Ein- oder Ausgänge. Durch die schnelle Signalverarbeitung eignet es sich für Einsatz in zeitkritischen Applikationen.

"R-ILB S3 24 AI4 AO2" / "R-ILB S3 24 AI6 AO4/F"

Die Inline Block IO "R-ILB S3 24 AI4 AO2" ist zum Einsatz innerhalb Sercos III vorgesehen. Sie kann zur Erfassung von Strom- oder Spannungssignalen sowie zur Erfassung von Temperatursensoren verwendet werden. Des Weiteren bietet sie die Möglichkeit analoge Strom- und Spannungssignale auszugeben. "R-ILB S3 24 AI4 AO2" verfügt über 4 analoge Eingänge und 2 analoge Ausgänge.

"R-ILB S3 24 AI6 AO4/F" verfügt über 6 analoge Eingänge und 2 analoge Ausgänge. Durch die schnelle Signalverarbeitung eignet es sich für Einsatz in zeitkritischen Applikationen.



Allgemeine Informationen zu der Produktfamilie "Rexroth Inline" finden Sie in der Dokumentation "Die Automatisierungsklemmen der Produktfamilie Rexroth Inline", Dokumentations-Type "DOK-CONTRL-ILSYSINS***-AWxx-DE-P".

"Rexroth Inline Modular IO" in Verbindung mit "IndraMotion MLD"

Es können maximal 4 Buskoppler für Sercos III mit jeweils maximal 16 Scheiben (Teilnehmer/Module) an einem CCD-Master ausgewertet werden.

Es werden alle Scheiben unterstützt, die als einfache digitale oder analoge E/As betrieben werden können.

Es existieren die folgenden Geräte (Produktfamilie "Rexroth Inline"):

- Buskoppler für Sercos III mit digitalen Ein- und Ausgängen (R-IB IL S3 BK DI8 DO4-PAC)
- Module mit analogen Eingängen (R-IB IL AI xx) oder analogen Ausgängen (R-IB IL AO xx)

- Module mit digitalen Eingängen (R-IB IL 24 DI xx) oder digitalen Ausgängen (R-IB IL 24 DO xx)
- Systemklemmen für verschiedene Funktionen, z. B. PWM, serielle Kommunikation RS232, RS485, Geberklemmen für SSI-Geber, etc.



Allgemeine Informationen zu der Produktfamilie "Rexroth Inline" finden Sie in der Dokumentation "Die Automatisierungsklemmen der Produktfamilie Rexroth Inline", Dokumentations-Type "DOK-CTRL-ILSYSINS***-AWxx-DE-P".

Rexroth IndraControl S20

Aufbau einer IndraControl S20-Station

"Rexroth IndraControl S20" ist ein I/O-System in modularer Bauweise für den Schaltschrank. Offen für alle Ethernet-basierten Kommunikationsprotokolle ermöglicht IndraControl S20 höchste Flexibilität. Darüber hinaus ist IndraControl S20 schnell in der Reaktionszeit und Installation, robust im Design und in der Mechanik und gleichzeitig besonders einfach in der Handhabung. Es dient zur Übertragung der Prozess-Signale an eine übergeordnete Steuerung. Dabei werden verschiedene Netzwerke unterstützt.

Eine IndraControl S20-Station setzt sich aus Einzelmodulen zusammen, die auf eine Tragschiene aufgerastet werden. Den Kopf der Station bildet eine Steuerung oder ein Buskoppler. An ihn werden I/O-Module angereicht. Die Verbindung der einzelnen Module untereinander und mit dem Stationskopf wird mittels Bussockelmodulen realisiert. Diese werden in der Tragschiene aneinandergerastet und bilden den Lokalbus.

Weiterführende Informationen erhalten Sie in der Anwendungsbeschreibung "Rexroth IndraControl S20" (Material-Nr. R911335987).

Antrieb oder Frequenzumrichter als Sercos Peripherie

Es können bis zu 4 IndraDrive-Geräte oder Frequenzumrichter des Typs "EFC" als Sercos-III-Peripherie konfiguriert werden. Dadurch erhöht sich die Anzahl der maximal projektierbaren IOs.

Die notwendigen Funktionsbausteine zur Steuerung von "IndraDrive" oder "EFC" als Sercos-III-Peripherie sind in der Bibliothek "RMB_SercosIII_Util.library" enthalten.



Es können nur insgesamt 4 weitere Peripherie-Geräte konfiguriert werden. Zu den Peripherie-Geräten zählen Antriebe, Frequenzumrichter, Inline Block IO oder IO-Buskoppler (Inline Modular IO, IndraControl S20, ...).

1.5.6 Begriffe, Grundlagen

Allgemeines

- Unter einer **Technologiefunktion** bzw. einem **Technologiepaket** versteht man eine abgeschlossene, umfangreiche Funktionalität (z. B. präventive Diagnosemöglichkeiten).
- Aufbauend auf Technologie-Bausteinen und -Paketen werden auch anwendungsspezifische Komplettlösungen von Bosch Rexroth angeboten, z. B. "Sequential Motion Control (SMC)" und "IndraMotion for Handling".
 - "Sequential Motion Control (SMC)" für IndraDrive erweitert die Servoantriebe IndraDrive um sequentielle Programmierung und Funktionen für Achskopplungen. Diese vorkonfigurierte Systemlösung für alle sequentiellen Bewegungsaufgaben bietet umfangreiche Funktionen für Automatik- und Handbetrieb.

- "IndraMotion for Handling" ist eine steuerungs-basierte, vorkonfigurierte und zugleich offene Motion-Control-Lösung. Das System können Sie für alle einfachen und kartesischen Bewegungsaufgaben in der Automation verwenden.

SPS-Programmierung

Nachfolgend Definitionen einiger Grundbegriffe bei der SPS-Programmierung:

- Unter **Ressourcen** versteht man Objekte zum Organisieren eines Projekts, zum Verfolgen von Variablenwerten und zum Konfigurieren des Projekts für die Anwendung auf dem Zielsystem und im Netzwerk (globale Variablen, Variablenkonfiguration, Traceaufzeichnung, Steuerungskonfiguration, Taskkonfiguration, Watch- und Rezepturverwalter).
 - Die **Variablenkonfiguration** dient zur Konfiguration von globalen Variablen, die im ganzen Projekt bzw. Netzwerk verwendet werden können.
 - Die **Traceaufzeichnung** bietet eine ähnliche Funktionalität wie ein Speicheroszilloskop: SPS-Daten (Variablenwerte) können aufgezeichnet bzw. verfolgt werden.
 - Mit der **Steuerungskonfiguration** ist es möglich die SPS-Hardware zu konfigurieren. Bei "IndraMotion MLD-S" wird keine Steuerungskonfiguration benötigt, da sich die Geräte-Eingänge/Geräte-Ausgänge mit der Inbetriebnahmesoftware zwischen Antrieb bzw. SPS aufteilen und adressieren lassen.
 - Es ist möglich, aber nicht zwingend notwendig, die Projektabarbeitung (Steuerung des Programms) über Tasks zu steuern; liegt keine **Taskkonfiguration** vor, muss das Projekt den Baustein "PLC_PRG" enthalten.
 - Mit dem **Watchverwalter** können im Online-Modus Werte der eingegebenen Variablen angezeigt werden. Der **Rezepturverwalter** ermöglicht die Vorbelegung von Variablen mit konstanten Werten.
- Eine **Task** ist eine zeitliche Ablaufeinheit eines IEC-Programms. Sie ist definiert durch einen Namen, eine Priorität und einen Typ. Mit dem Typ wird festgelegt welche Bedingung den Start der Task auslöst. Diese Bedingung kann entweder zeitlich (Zyklusintervall, freilaufend) oder durch ein internes oder ein externes Ereignis definiert sein, bei dessen Eintreten die Task ausgeführt werden soll; beispielsweise die steigende Flanke einer globalen Projektvariable oder ein Interrupt-Event der Steuerung.
- Eine **Funktion** ist ein Baustein mit folgenden Eigenschaften:
 - Eine Funktion liefert als Ergebnis der Ausführung genau ein Datum (das auch mehrelementig sein kann, wie z. B. Felder oder Strukturen) zurück.
 - Der Aufruf einer Funktion kann in textuellen Sprachen als ein Operand in Ausdrücken vorkommen.
 - Eine Funktion hat kein "Gedächtnis"; d. h. Variableninhalte werden nicht gespeichert.
 - Eine Funktion benötigt keine Instanz.
- Ein **Funktionsbaustein** besitzt folgende Eigenschaften:
 - Ein Funktionsbaustein kann mehrere Werte zurückliefern.
 - Ein Funktionsbaustein hat ein "Gedächtnis"; d. h. Variableninhalte werden gespeichert.

- Für einen Funktionsbaustein muss eine Instanz angegeben werden.
- **Programme** sind global im gesamten Projekt bekannt. Programme werden nicht instanziiert. Die Variablen des Programms sind einmal vorhanden. Programme können in einer Task aufgerufen werden. Eine Task kann mehrere Programme aufrufen und ein Programm kann Funktionen, Funktionsbausteine oder weitere Programme aufrufen.
- Das **Bootprojekt** ist ein Projekt, das auf dem Zielsystem (hier: ein Antriebsregelgerät der Antriebsfamilie "Rexroth IndraDrive" mit dem Firmware-Funktionspaket für "IndraMotion MLD") geladen wurde und beim Booten des Antriebsregelgeräts je nach Einstellung (vgl. "P-0-1367, SPS Konfiguration") gestartet wird.

PLCopen

Die PLCopen ist eine Organisation, in der sich verschiedene Hersteller zusammengefunden haben, um Programmierschnittstellen, Inbetriebnahme und Wartung zu vereinheitlichen, und so den programmiertechnischen Zugriff auf die Bewegungssteuerung (Motion Control) zu harmonisieren.

Die Unabhängigkeit von der Steuerungsarchitektur wird durch Kapselung von Funktionalität und Daten in Funktionsbausteinen erreicht; d. h. Applikationen, die nach dem in der PLCopen erarbeiteten Standard ("PLCopen Motion Control Profil", einem Satz von IEC 61131-3 Funktionsbausteinen) programmiert sind, sind unabhängig von der ausführenden Hardware und so über Plattformgrenzen hinweg wiederverwendbar.

IEC 61131

Mit den IndraWorks-Packages der "IndraMotion MLD" werden, wie in der IEC 61131 üblich, Bibliotheken zur Verfügung gestellt. Diese enthalten fertige Funktionen, Funktionsbausteine und Variablen, die als Blackbox verwendet werden. Es gibt zwei Arten von Bibliotheken:

- Firmware-Bibliotheken: Sie enthalten keinen Code, sondern nur Deklarationen. Der Code ist fest in der Firmware kodiert.
- IEC-Bibliotheken: Sie sind IEC 61131-3-konform geschrieben und können bei Bedarf gegen Ändern oder Lesen des Codes geschützt werden.

1.5.7 Funktionsübersicht und Systemeigenschaften

Allgemeine Grundfunktionen der integrierten SPS

Frei programmierbare SPS nach IEC 61131

Durch die Definition des Industrie-Standards für die Programmierung in der Automatisierung in Form der IEC 61131-3 sind erstmals die Grundlagen für eine kombinierte Logik- und Bewegungssteuerung gelegt. Im Wesentlichen sind dies:

- Definition der Ablaufsprache **SFC (Sequential Function Chart)** als leistungsfähiges Programmierwerkzeug für datenflussorientierte Programme wie z. B. Regler
- Definition der pascalähnlichen Hochsprache **ST (Structured Text)** als leistungsfähiges Instrument auch zur Programmierung von komplexen (z. B. mathematischen) Funktionalitäten
- Definition von **Basis-Datentypen** von BOOL bis DINT (32 Bit Integer) oder REAL (32 Bit Float) sowie Felder und Strukturen
- Definition von **instanzierbaren Funktionsbausteinen**

- Definition von **Standard-Funktionsbausteinen** (Counter, Timer, Flanken-erkennung,...) sowie Standard-Funktionen (mathematische Funktionen, String-Verarbeitung, Vergleiche,...)

Der Trend für eine kombinierte Logik- und Bewegungssteuerung wird durch die offizielle Freigabe der Spezifikation der "PLCopen Motion Task Force" verstärkt. Darin sind Funktionen und Funktionsbausteine zur Einzelachspositionierung, sowie für elektronische Getriebefunktionen definiert. Außerdem wurde eine grundsätzliche Verfahrensweise zur Implementierung von bewegungsauslösenden Funktionsbausteinen festgelegt.

Programmiersprachen

"IndraMotion MLD" kann mittels der folgenden Programmiersprachen programmiert werden:

- Anweisungsliste (AWL)
- Strukturierter Text (ST)
- Kontaktplan (KOP)
- Funktionsplan (FUP)
- Ablaufsprache (AS)
- Freigrafischer Funktionsplan (CFC)

Siehe "[Programmiersprachen](#)"

Zur Programmerstellung steht die in IndraWorks integrierte Programmierumgebung "IndraLogic 2G" zur Verfügung (siehe Dokumentation "IndraLogic 2G Programmieranleitung").

Unterstützte Bibliotheken

Die von "IndraMotion MLD-2G" unterstützten Bibliotheken sind in der Dokumentation "DOK-INDRV*-MLD-SYSLIB3-RExx-DE-P" beschrieben.

HINWEIS

Sachschäden durch das Einbinden von nicht unterstützten Bibliotheken.

Es dürfen nur Bibliotheken eingebunden werden, die in der genannten Dokumentation aufgeführt sind.

Unterstützte Schnittstellen und Zugriffsmechanismen

siehe "Kommunikationsschnittstellen und Datenkanäle der MLD"

Mengengerüst (Speicher)

"IndraMotion MLD" verfügt über folgende Ressourcen:

- Interner **Codespeicher** (RAM im Steuerteil):



Für die Funktion "Online-Change" wird ca. 33 % der hier angegebenen Speicher-Ressource reserviert, d. h. es sind etwa 66 % nutzbar.

- Bis MPB-20: **2 MB** (nicht speichernd), ab MPB-21: **1,2 MB** (nicht speichernd)
- MPC: **12 MB** (nicht speichernd)

- **Programmgröße** (Bootprogramm):

- ca. **650 kB** mit Basic-Bedienteil und Advanced-Bedienteil im Parameterspeicher
- ca. **8 MB** mit Advanced-Bedienteil und µSD-Speicherkarte (optionale Speicherkarte muss über P-0-1367 freigeschaltet werden)



Das **Bootprojekt** wird beim Einschalten des Geräts aus den Programmparametern in den **Codespeicher** (RAM) geladen. Daher muss der **Programmspeicher** groß genug sein, um das Projekt aufzunehmen.

- **Datenspeicher:**



Für die Funktion "Online-Change" wird ca. 10 % der hier angegebenen Speicher-Ressource reserviert, d. h. es sind etwa 90 % nutzbar.

MPB: 3 MB

MPC: 9 MB

Es gibt mehrere Datenbereiche mit unterschiedlicher Verwendung (Instanzen sind Variablen):

- **Global:** Speicher für globale Variablen
2048 kByte
Hinweis: Die Instanzdaten hängen an globalen Variablen. Alle Daten von Funktionen werden nicht in diesem Segment, sondern im Stack abgelegt.
- **Memory:** Alle mit "%M" adressierten Merker-Variablen
32 kByte
- **Input:** Alle mit "%I" adressierten Eingangs-Variablen
16 kByte
- **Output:** Alle mit "%O" adressierten Ausgangs-Variablen
16 kByte
- **Retain-Variablen** und **persistente Variablen**
(VAR_GLOBAL PERSISTENT RETAIN):
 - 472 Byte auf Basic-Bedienteil (MPB)
 - 31704 Byte auf Advanced-Bedienteil (MPC) in FRAM



Retain-Variablen und persistente Variablen können nur in einer "Globalen Persistenten Variablenliste" deklariert werden, die zu einer Applikation gehört. Pro Applikation kann jeweils nur eine solche Liste hinzugefügt werden.

Persistente Variablen werden nur bei einem Reset (Ursprung) der Steuerung neu initialisiert.



Die Funktionalität der Retain-Variablen und der persistenten Variablen ist in der Dokumentation "IndraLogic 2G, SPS-Programmiersystem" beschrieben: "Remanente Variablen - RETAIN, PERSISTENT".

- **Sockets**

Sockets werden zum Aufbau einer Verbindung und zum Datentransfer mit TCP (Transmission Control Protocol) und UDP (User Datagram Protocol) benötigt.

Es sind 20 Sockets verfügbar. Die für das Anwenderprogramm verfügbaren Sockets gelten, wenn z. B. IndraWorks mit MLD, HMI aktiv sind. Falls noch weitere Ethernet-Kommunikationskanäle (z. B. FTP) genutzt werden, reduziert sich die für die MLD verfügbare Anzahl der Sockets.

- **Symbole**

Die Anzahl der Symbole ist von der Ablage des Bootprojekts abhängig.

- Bootprojekt in Parametern:

Bei MPB- und MPC-Firmware sind bei Ablage des Bootprojekts in Parametern je nach Datentyp bis zu ca. 2000 Symbolvariablen möglich; Voraussetzung ist, dass kein Code programmiert ist, sondern nur Symbolvariablen deklariert sind. Je mehr Code die Applikation enthält, umso geringer ist die maximal mögliche Anzahl an Symbolvariablen.

- Bootprojekt auf µSD-Card:

Bei MPC-Firmware sind im Vergleich zu der Ablage des Bootprojekts in Parametern bei der Ablage des Bootprojekts auf der µSD-Card sehr viel mehr Symbolvariablen möglich, da die Begrenzung der Codespeicher ist. Als praxisrelevanter Richtwert dient folgendes Beispiel: 10000 Symbolvariablen benötigen ca. 1,5 MB Codespeicher.

- **Freie Parameter**

- 32-Bit Parameter

- 32 gepufferte Parameter (P-0-1316 bis P-0-1331 und P-0-1370 bis P-0-1385)

- 32 ungepufferte Parameter (P-0-1270 bis P-0-1301)

- 32-Bit Listenparameter

- "P-0-1368, SPS Globales Register AL0" (Listenregister mit 8192 Werten, nicht gepuffert)

- "P-0-1389, SPS Globales Register GL0" (Listenregister mit 1024 Werten, gepuffert)

- "P-0-1311, SPS Globales Register GL1" (Listenregister mit 1024 Werten, gepuffert)

- "P-0-1312, SPS Globales Register GL2" (Listenregister mit 1024 Werten, gepuffert)
- Zwei globale **Textregister** mit je 255 Zeichen
- Anzeigeformat der freien Parameter ist parametrierbar über "P-0-1386, SPS Anzeigeformat Globaler Register"
- Name, Einheit und Grenzwerte der freien Parameter sind über SPS-Funktionen konfigurierbar
- **Prozessabbild-Eingang (PAE)** (max. 72 Byte)
 - "P-0-1390, SPS Eingang WORD0 AT %IB0" bis "P-0-1397, SPS Eingang WORD7 AT %IB14"
 - "P-0-1398, SPS Eingang WORD8 AT %IB16" bis "P-0-1409, SPS Eingang WORD19 AT %IB38"
 - "P-0-1440, SPS Eingang DWORD25 AT %IB100" bis "P-0-1447, SPS Eingang DWORD32 AT %IB128"
- **Prozessabbild-Ausgang (PAA)** (max. 40 Byte):
 - "P-0-1410, SPS Ausgang WORD0 AT %QB0" bis "P-0-1417, SPS Ausgang WORD7 AT %QB14"
 - "P-0-1418, SPS Ausgang WORD8 AT %QB16" bis "P-0-1429, SPS Ausgang WORD19 AT %QB38"

Einschränkungen

Folgende Einschränkungen bzw. Begrenzungen existieren:

- Der Codespeicher enthält das aktuelle laufende Projekt und nicht das "Bootprojekt". Beim Starten des Antriebs, oder ggf. beim Wechsel vom Parametriermodus (PM) in den Betriebsmodus (OM), wird das Bootprojekt aus den Parametern P-0-1352 bis P-0-1358 in den Codespeicher geladen.
- Wird "Online-Change" durchführt, so wird das geänderte Programm in den zweiten Codebereich geladen und dann dynamisch in den zweiten Codebereich umgeschaltet.
- Der Retain-Speicher wird automatisch benutzt, wenn Variablen mit dem Schlüsselwort "RETAIN" definiert sind. Werden zu viele Retain-Variablen angelegt, so dass die Größe des verfügbaren Retain-Speichers überstiegen ist, so kommt beim Download eine Fehlermeldung.

Leistungs- und Performance-Angaben

"IndraMotion MLD" zeichnet sich durch folgende Leistungsdaten aus:

- Anzahl Bausteine: Der Wert ist fest auf den Maximalwert eingestellt (2048 Bausteine möglich)
- Datentypen: Einzelne Bits bis 32 Bit Float, sowie eigene Typen und Strukturen
- Die minimale SPS-Zykluszeit ist steuerteilunabhängig, sie beträgt 1 ms. Siehe auch Firmware-Funktionsbeschreibung "Performance-Angaben"
- Die Leistung ist abhängig von ...
 - ... der für die SPS-Task zur Verfügung stehenden Zeitscheibe
 - ... der Codegröße
 - ... der Art der Operationen: REAL, DWORD, WORD, BYTE, BIT, etc. [Durch die integrierte FPU (Floating Point Unit) auf dem Prozessor ist die Verarbeitungsgeschwindigkeit von REAL sehr hoch.]



Die Verarbeitungszeit hängt von mehreren Faktoren, wie Auslastung durch Regelung/Betriebsart, Art der Rechenoperationen ab. Im günstigsten Fall liegt die Verarbeitungszeit für 1000 Zeilen Anweisungsliste unter 4 µs, im schlechtesten Fall liegt sie über 100 µs.

Bei einem Testprogramm, bei dem 1000 gemischte Anweisungen verarbeitet wurden, wurden folgende Verarbeitungszeiten gemessen:

- BASIC-Steuerteil: 99 µs
- ADVANCED-Steuerteil: 49 µs

Einschränkungen

Folgende Einschränkungen bzw. Begrenzungen existieren:

- Die für die SPS verfügbare Zeitscheibe ist abhängig von der Regelung und der Auslastung des Antriebs durch die Betriebsarten.
- Bei "Online-Change" gibt es eine kurze Verzögerung der SPS, da der Cache verworfen und neu geladen werden muss.
- Die Bitverarbeitung ist systembedingt deutlich langsamer als die Bearbeitung von 8-, 16- oder 32-Bit-Werten.

Verwendungshinweise und Anwendungsgebiete

Allgemeines

In den nachfolgenden Abschnitten werden die erforderlichen Voraussetzungen zur Nutzung von "IndraMotion MLD" genannt. Außerdem werden Hinweise für die Verwendung bzw. die Einsatzgebiete von "IndraMotion MLD" gegeben.

Zunächst eine Übersicht aller Abhängigkeiten für die Nutzung von "IndraMotion MLD":

	Gerät (Leistungsteil)	Firmware	IndraMotion MLD-S		IndraMotion MLD-M	
IndraDrive Cs	HCS01.1	ab MPB18VRS	ML	MA	-	-
IndraDrive Cs	HCS01.1	ab MPC18VRS	ML	MA	ML	MA
IndraDrive C / IndraDrive M mit Steuerteil CSB02.x	Hinweis: Steuerteile CSB02.x werden nur durch Leistungsteile unterstützt, die ab 2007 produziert wurden	ab MPB18VRS	ML	MA	-	-
IndraDrive C / IndraDrive M mit Steuerteil CSH02.3	Hinweis: Steuerteile CSH02.x werden nur durch Leistungsteile unterstützt, die ab 2007 produziert wurden	ab MPC18VRS	ML	MA	ML	MA
IndraDrive Mi	KMS02 KSM02	ab MPB18VRS	ML	MA	-	-

ML frei programmierbare SPS (IEC61131)

MA frei programmierbare SPS (IEC61131) für komplexe Aufgaben

Tab. 1-9: System-Voraussetzungen für die Nutzung von "IndraMotion MLD"

Hardware-Voraussetzungen

Die Nutzung der antriebsintegrierten SPS (Rexroth IndraMotion MLD) setzt folgende Hardware voraus:

Steuerteil / Leistungsteil: IndraDrive Cs

- HCS01.1E-Wxxxx-A-0y-**B**-... (ab FWA-INDRV-MPB18VRS)
- HCS01.1E-Wxxxx-A-0y-**A**-... (ab FWA-INDRV-MPC18VRS und MLD-M)



Bei Economy-Geräten (HCS01.1E-Wxxxx-A-0y-**E**-...) ist die Nutzung der MLD nicht möglich!

Für Mehrachsgeräte (HCQ01.1 bzw. HCT01.1) ist die MLD nicht verfügbar.



Die Nutzung der SPS-Funktionalität erfordert kein spezielles Optionsmodul bzw. keine spezielle Steuerteilkonfiguration, da es sich um eine SPS handelt, welche im Echtzeitkern des Antriebsprozessors mitläuft.

Firmware-Voraussetzungen

Um die antriebsintegrierte SPS (Rexroth IndraMotion MLD) nutzen zu können, ist das Firmware-Grundpaket und ein Firmware-Funktionspaket (optionales Erweiterungspaket) erforderlich. Das Firmware-Funktionspaket muss freigeschaltet / lizenziert sein (siehe [Auslieferung und Lizenzierung der SPS](#)).

Additives Funktionspaket

Folgende Ausführungen des additiven Funktionspaketes stehen zur Verfügung:

- Die Ausführung **"TF"** ermöglicht das Laden und Verwenden von fertigen SPS-Programmen (Technologiefunktionen) von Rexroth (siehe "Technologiefunktionen").
- Die Ausführung **"ML"** ermöglicht die freie Programmierung der Rexroth IndraMotion MLD-S / MLD-M.
- Die Ausführung **"MA"** ergänzt die Ausführung "ML" um die Nutzbarkeit von fertigen Technologiefunktionsbausteinen (Bibliothek).
- Die Ausführung **"ME"** ergänzt die Ausführung "MA" um eine Ethernet/IP-Scanner Funktionalität, die in Form einer MLD-Bibliothek zur Verfügung gestellt wird.



Eine Technologiefunktion ist ein komplettes (übersetztes) SPS-Projekt, welches sich aus mehreren Bausteinen (IEC-konform) zusammensetzen kann. Ein Projekt wird durch Kombination von IEC- oder Firmwarebausteinen erstellt und bietet meist eine komplexe und umfangreiche Funktionalität zur Lösung einer speziellen technologischen Anwendung an.

Die Technologiefunktionen können über IndraWorks geladen werden (siehe auch MLD-Anwendungsbeschreibung: "Nutzung von Technologiefunktionen").

Beispiele für Technologiefunktionen: mitlaufende Trennvorrichtung, Pick&Place, Prozessregler (Registerregler, Wickelrechner, ...), vorbeugende Wartung, Sytronix...

Nachfolgend eine Übersicht der Firmware-Grundpakete und der verfügbaren Ausführungen des additiven Funktionspakets "IndraMotion MLD":

Firmware-Version	Ausprägung	MLD-S				MLD-M			
	Option	ML	MA	ME ¹⁾	TF	ML	MA	ME	TF
FWA-INDRV*-MP*18VRS	MPB	X	X	-	-	-	-	-	-
	MPC	X	X	-	-	X	X	-	-
	MPE	-	-	-	-	-	-	-	-
	MPM	-	-	-	-	-	-	-	-
FWA-INDRV*-MP*19VRS	MPB	X	X	-	X	-	-	-	-
	MPC	X	X	-	X	X	X	-	-
	MPE	-	-	-	-	-	-	-	-
	MPM	-	-	-	-	-	-	-	-
FWA-INDRV*-MP*20VRS / FWA-INDRV*-MP*21VRS	MPB	X	X	X	X	-	-	-	-
	MPC	X	X	X	X	X	X	X	-
	MPE	-	-	-	-	-	-	-	-
	MPM	-	-	-	-	-	-	-	-
FWA-INDRV*-PSB19VRS	PSB	X	X	-	X	-	-	-	-
FWA-INDRV*-PSB20VRS	PSB	X	X	-	X	-	-	-	-

- 1) IndraDrive-Pakete mit Closed-Loop-Antriebsregelgerät. "ME" (inklusive "MA" und "ML") Ethernet-IP-Scanner
 2) Funktionspakete immer gesetzt
 X Additives Funktionspaket für diese Firmware möglich
 - Additives Funktionspaket für diese Firmware nicht möglich

Tab. 1-10: Übersicht der Firmware-Grundpakete und der verfügbaren Ausführungen des additiven Funktionspakets "IndraMotion MLD"

Siehe auch Firmware-Funktionsbeschreibung "Firmware-Typen"

Siehe auch Firmware-Funktionsbeschreibung "Additive Funktionspakete"

Auslieferung und Lizenzierung der SPS

Wird bei der Bestellung der Firmware zu dem Firmware-Grundpaket das Firmware-Funktionspaket sofort mitbestellt, dann erfolgt die Freischaltung der "IndraMotion MLD" bereits werksseitig. Wird hingegen das Firmware-Funktionspaket nachbestellt, so ist eine so genannte Nachlizenzierung erforderlich; d. h. die SPS-Funktionalität muss durch den Anwender nachträglich freigeschaltet bzw. lizenziert werden.



Wenn die SPS nicht freigeschaltet ist, verweigert der Antrieb die Kommunikation mit dem SPS-Programmiersystem. Es wird eine Meldung generiert. Weiterhin ist zu beachten, dass bei nicht lizenzierter Nutzung der SPS die Gewährleistung für den Antrieb erlischt!

Siehe auch Firmware-Funktionsbeschreibung "Funktionspaketfreischaltung"

Software-Voraussetzungen

Zur Parametrierung, Inbetriebnahme und Programmierung des Antriebs mit integrierter SPS ist das Engineeringtool "IndraWorks MLD" erforderlich.

Anwendungsmöglichkeiten der IndraMotion MLD

Übersicht

"IndraMotion MLD" kann für die Realisierung von einfachen Programmen zur Flexibilisierung / Erweiterung der Antriebsfunktionalität ("intelligente Servoachse"), der Nutzung als Motion-System für die Kommandierung von einer oder mehreren Achsen (autarke "Motion Logic Control") oder der Steuerung von Versorgungsgeräten (ab Firmware-Version 20) verwendet werden. Hierfür stehen diverse Bibliotheken und Technologiefunktionen zur Verfügung.

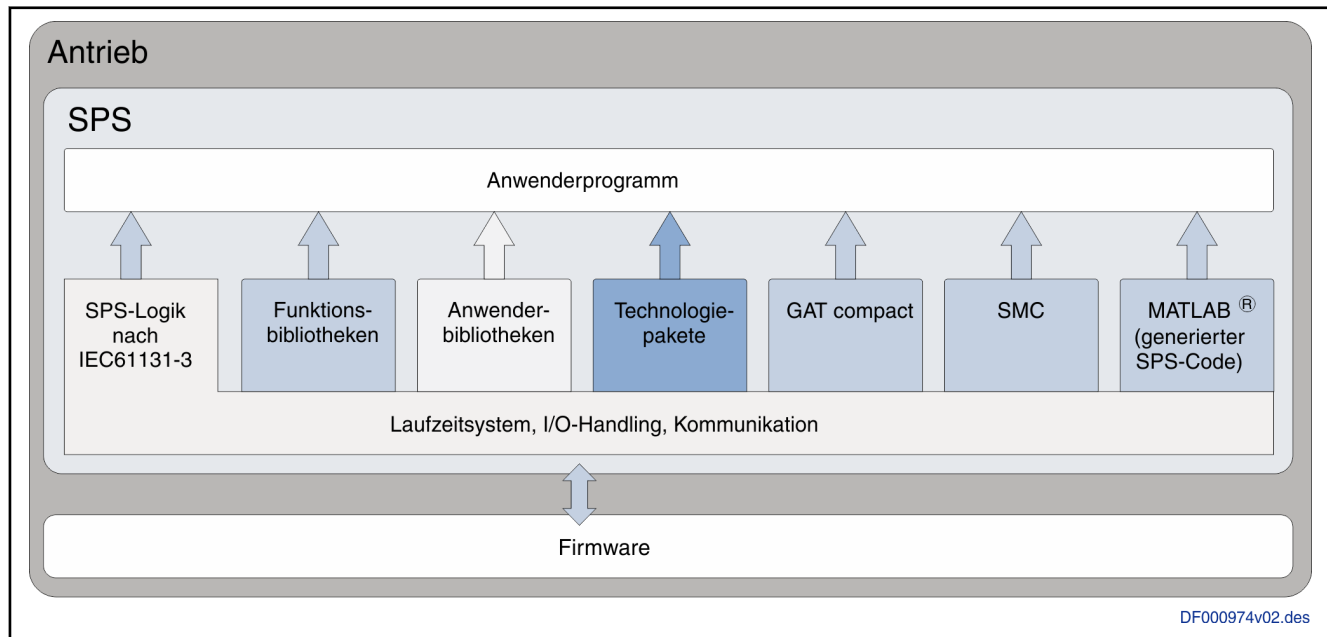


Abb. 1-3: Übersicht der Verwendungsmöglichkeiten im Antriebsregelgerät

IndraMotion MLD als intelligente Servoachse

Die Erweiterung bzw. Flexibilisierung der Antriebsfunktionalität kann im einfachsten Fall durch einen Funktionsbaustein oder ein einfaches Programm erfolgen.

Merkmale Die Ausprägung "Intelligente Servoachse" zeichnet sich durch folgende Merkmale aus:

- Der Antrieb wird von einer übergeordneten Steuerung (NC/SPS) verwaltet bzw. kontrolliert.
- Die übergeordnete Steuerung erzeugt die zyklischen Sollwerte und bestimmt die Betriebsart.
- Über das Engineeringtool "IndraWorks MLD" kann eine Funktion aus den Makro-Bibliotheken in den Antrieb geladen werden. Die geladene Funktion kann anschließend wie eine "normale Firmwarefunktion" vom Kunden parametrisiert werden. Für den Anwender wirken Makrofunktionen wie implementierte Firmwarefunktionen.
- Der Antrieb kann sich temporär die Kontrolle holen, um dann die Bewegungssteuerung selbst zu übernehmen.

Beispiel Beispielsweise kann durch einen Funktionsbaustein oder ein einfaches Programm eine intelligente Fehlerreaktion realisiert werden:

Bei Auftreten eines Fehlers schaltet dann die SPS auf interne Kontrolle und führt eine Reaktion mit Bewegung des Antriebs durch. Anschließend gibt die SPS die Kontrolle wieder an die Führungskommunikation (externe SPS) ab.

IndraMotion MLD als autarke "Motion Logic Control"

Soll "IndraMotion MLD" als Motion-System genutzt werden, dann besteht die Möglichkeit Steuerungs-Funktionen in die im Antrieb integrierte SPS (MLD) zu verlagern.

- Der Antrieb übernimmt Steuerungsfunktionalität und kommandiert die Achse(n), d. h. es wird eine Abfolge von Verfahrbefehlen frei programmiert.
 - Einzelachs-"Motion Logic Control" ("IndraMotion MLD-S", ab MPB18VRS)
 - Mehrachs-"Motion Logic Control" ("IndraMotion MLD-M", ab MPC18VRS)

Merkmale Bei der Verwendung von "IndraMotion MLD" als "Autarke "Motion Logic Control"" sind folgende Punkte zu beachten:

- Der Antrieb kann ohne eine übergeordnete Steuerung (SPS) arbeiten.
- Die integrierte Steuerung kommandiert die Achse(n) und kann auch die zyklischen Sollwerte und die Betriebsart festlegen.
- Eigenständige Motion-Programme können erstellt werden.
- Zur Programmierung stehen neben den Standard-Funktionsbausteinen auch entsprechende Hersteller-Funktionsbausteine zur Verfügung. Die Hersteller-Funktionsbausteine sind in Bibliotheken zusammengefasst.

Anwendung / Nutzung von IndraMotion MLD

Frei programmierbare SPS nach IEC 61131

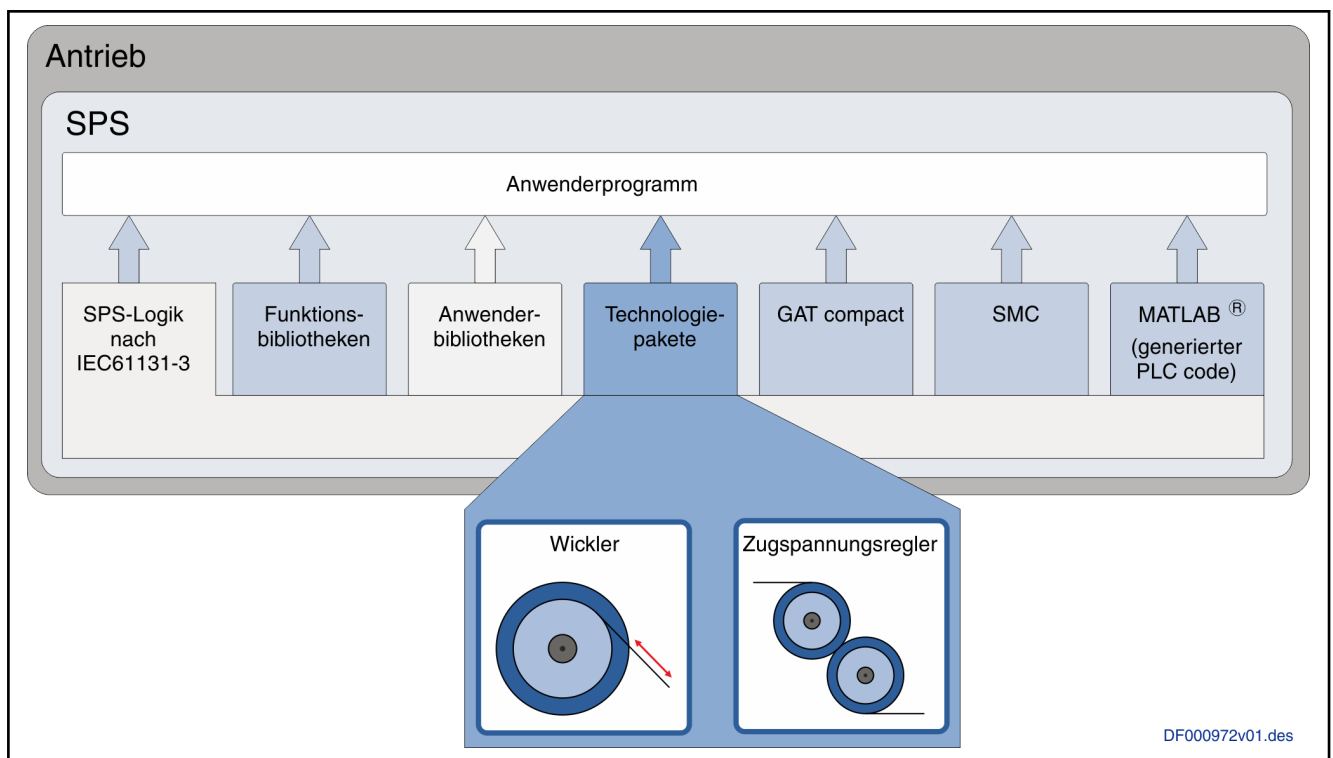


Abb. 1-4: Möglichkeiten zur Gestaltung eines Anwenderprogramms bzw. Nutzung der "IndraMotion MLD"

Bosch Rexroth bietet bei der Erstellung von Anwenderprogrammen folgende Unterstützung an:

- Funktionsbibliotheken

- Grundlegende IEC-Funktionsbausteine
- PLCopen-Funktionsbausteine
- AxisInterface
- **Technologie-Bibliotheken**
Die Technologiebausteine werden branchen- und themenorientiert bei Bosch Rexroth entwickelt und in Form von Technologie-Bibliotheken zur Verfügung gestellt (z. B. Wickler, Zugspannungsregler):
 - **Motionfunktionen** (MX_TechMotion)
 - Erweiterte **Antriebsfunktionen** (MX_TechWinder)
- **Anwenderbibliotheken**
Außerdem besteht für den Anwender die Möglichkeit, selbst erstellte Funktionsbausteine bzw. -pakete in eigenen Anwenderbibliotheken zusammenzufassen.

Das **Anwenderprogramm** kann aus einer beliebigen Kombination von Funktionsbibliotheken, Technologiepaketen und Anwenderbibliotheken bestehen.

Nutzung von Technologiepaketen und weitere Programmierlösungen

Neben der Möglichkeit "IndraMotion MLD" frei zu programmieren, kann die integrierte SPS auch als Basisfunktion für die Nutzung vorgefertigter abgeschlossener Lösungen (z.B. Technologiepakete) verwendet werden.

Es werden folgende Ausprägungen unterschieden:

- **Technologiepaket** (gekapselte Technologiefunktion)
Compiliertes SPS-Projekt welches als abgeschlossene ".spj"-Datei (Parameterdatei) in den Antrieb geladen wird. Die Technologiefunktion wird wie eine "normale Firmwarefunktion" über P-Parameter bedient und erfordert keine Programmierkenntnisse. Man unterscheidet hierbei folgende Typen:
 - **Motionfunktionen** (z. B. Prozessregler, Wickler, Querschneider, ...)
 - Erweiterte **Antriebsfunktionen** (z. B. Eventfunktionen, programmierbare Fehlerreaktionen, Schnellhalt, ...)
 - Erweiterte **Regelungsfunktionen** (z. B. "Prozessregler", "Schwingungsdämpfung und -vermeidung", "Kinetic Buffering")
- **Turnkey-Lösungen** ("Rundum-Sorglos-Paket")
Abgeschlossene (fertige) Systemlösungen mit Systemdokumentation, welche als eigene Produkte erworben werden können. Es sind keine Programmierkenntnisse erforderlich. Ggf. vorhandene Abhängigkeiten sind berücksichtigt bzw. werden abgefangen.



Gekapselte Technologiefunktionen bzw. "Turnkey-Lösungen" können den erweiterten SPS-Speicher nicht nutzen.



Technologiepakete sind in IEC programmiert und können als Projekte in Form von Parametersätzen über die Inbetriebnahmesoftware geladen oder als Funktionsbausteine aus Bibliotheken in ein Projekt integriert werden.

Im Betrieb wird nur noch die Runtime der SPS als Plattform für die Technologiefunktionen und erweiterten Antriebsfunktionen benötigt.

- **GAT compact Beispielprojekt**

Einschaltfertiger Programmierrahmen, in dem der Applikationscode an vordefinierten Stellen eingetragen wird. Durch die dialogbasierte Codegenerierung des GAT-Wizards werden professionell gut strukturierte Applikationsprogramme mit vorbereiteten Elementen erstellt. GATcompact ist ab IndraWorks 14V14 mit MPx20 verfügbar (siehe "[Automatisierte Programmerstellung mit GAT compact](#)").

- **Sequential Motion Control (SMC)**

Sequentielle Programmierung und Funktionen für Achskopplungen. Vor-konfigurierte Systemlösung für alle sequentiellen Bewegungsaufgaben; bietet umfangreiche Funktionen für Automatik- und Handbetrieb.

2 Wichtige Gebrauchshinweise

2.1 Bestimmungsgemäßer Gebrauch

2.1.1 Einführung

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

WARNUNG

Personen- und Sachschäden durch falschen Gebrauch der Produkte!

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



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

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

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

2.1.2 Einsatz- und Anwendungsbereiche

Antriebsregelgeräte von Rexroth sind dazu bestimmt, elektrische Motoren zu regeln und deren Betrieb zu überwachen.

Zur Regelung und Überwachung der Antriebsregelgeräte kann es notwendig sein, dass zusätzliche Sensoren und Aktoren angeschlossen werden müssen.



Die Antriebsregelgeräte dürfen nur mit den in dieser Dokumentation angegebenen Zubehör- und Anbauteilen benutzt werden. Nicht ausdrücklich genannte Komponenten dürfen weder angebaut noch angeschlossen werden. Gleiches gilt für Kabel und Leitungen.

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

Antriebsregelgeräte müssen vor der Inbetriebnahme programmiert werden, damit der Motor die für die Anwendung spezifischen Funktionen ausführt.

Antriebsregelgeräte aus der Baureihe IndraDrive sind für den Einsatz in ein- und mehrachsigen Antriebs- und Steuerungsaufgaben entwickelt worden.

Für den applikationsspezifischen Einsatz der Antriebsregelgeräte stehen Gerätetypen mit unterschiedlicher Antriebsleistung und unterschiedlichen Schnittstellen zur Verfügung.

Typische Anwendungsbereiche sind beispielsweise:

- Handhabungs- und Montagesysteme,
- Verpackungs- und Lebensmittelmaschinen,
- Druck- und Papierverarbeitungsanlagen und
- Werkzeugmaschinen.

Antriebsregelgeräte dürfen nur unter den in dieser Dokumentation angegebenen Montage- und Installationsbedingungen, in der angegebenen Gebrauchslage und unter den angegebenen Umweltbedingungen (Temperatur, Schutzart, Feuchte, EMV u. a.) betrieben werden.

2.2 Nicht-bestimmungsgemäßer Gebrauch

Die Verwendung der Antriebsregelgeräte außerhalb der in der Dokumentation beschriebenen Betriebsbedingungen und angegebenen technischen Daten und Spezifikationen gilt als "nicht bestimmungsgemäß".

Antriebsregelgeräte dürfen nicht eingesetzt werden, wenn ...

- sie Betriebsbedingungen ausgesetzt werden, die die vorgeschriebenen Umgebungsbedingungen nicht erfüllen. Untersagt sind z. B. der Betrieb unter Wasser, unter extremen Temperaturschwankungen oder extremen Maximaltemperaturen.
- Außerdem dürfen Antriebsregelgeräte nicht bei Anwendungen eingesetzt werden, die von Rexroth nicht ausdrücklich freigegeben sind. Beachten Sie hierzu bitte unbedingt die Aussagen in den allgemeinen Sicherheitshinweisen!



Komponenten des Antriebssystems IndraDrive sind **Produkte der Kategorie C3** (mit eingeschränkter Erhältlichkeit) nach IEC 61800-3. Diese Kategorie umfasst EMV-Grenzwerte zur leitungsgeführten und gestrahlten Störaussendung. Zur Einhaltung dieser Kategorie (Grenzwerte) müssen im Antriebssystem entsprechende Entstörmaßnahmen angewendet werden (z. B. Netzfilter, Schirmmaßnahmen).

Diese Komponenten sind nicht vorgesehen für den Einsatz in einem öffentlichen Niederspannungsnetz, das Wohngebiete speist. Wenn diese Komponenten in einem solchen Netz betrieben werden, sind Hochfrequenzstörungen zu erwarten. Zusätzliche Entstörmaßnahmen können dann erforderlich sein.

HINWEIS

Gefahr von Störungen bzw. einem Steuerungsausfall aufgrund unautorisierter Zugriffe über eine direkte ungesicherte Verbindung zum Internet!

Die Ethernet-Schnittstelle der Antriebsregelgeräte bietet keinen sicheren Schutz gegen unautorisierten Zugriff. Aus diesem Grund sollten die Antriebsregelgeräte nur in abgesicherten Netzwerken eingesetzt werden (Security).

3 Sicherheitshinweise für elektrische Antriebe und Steuerungen

3.1 Begriffsdefinitionen

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

Projektierungsbeschreibung	Eine Projektierungsbeschreibung ist Teil der Anwendungsdokumentation zur Hilfestellung bei der Auslegung und Planung von Systemen, Maschinen oder Anlagen.
Qualifiziertes Personal	Im Sinne dieser Anwendungsdokumentation umfasst das qualifizierte Personal diejenigen Personen, die mit der Installation, Montage, Inbetriebnahme und Betrieb der Komponenten des elektrischen Antriebs- und Steuerungssystems sowie den damit verbundenen Gefahren vertraut sind und über die ihrer Tätigkeit entsprechenden Qualifikationen verfügen. Zu derartigen Qualifikationen gehören u. a.: • Eine Ausbildung oder Unterweisung bzw. Berechtigung, um Stromkreise und Geräte sicher ein- und auszuschalten, zu erden und zu kennzeichnen. • Eine Ausbildung oder Unterweisung für die Pflege und den Gebrauch angemessener Sicherheitsausrüstung. • Eine Schulung in Erster Hilfe.
Qualifiziertes Personal für den Umgang mit funktional sicheren Produkten	Personen, die funktional sichere Produkte projektieren, in Betrieb nehmen und betreiben, müssen über die unter " Qualifiziertes Personal " genannten Kenntnisse verfügen. Zusätzlich müssen diese Personen über Kenntnisse von technischen Sicherheitskonzepten, sowie den geltenden Normen und Vorschriften im Bereich der funktionalen Sicherheit verfügen.
Steuerungssystem	Ein Steuerungssystem umfasst mehrere miteinander verbundene Steuerungskomponenten, die als eine einzige Funktionseinheit in Verkehr gebracht werden.

3.2 Grundsätzliche Hinweise

3.2.1 Benutzung und Weitergabe der Sicherheitshinweise

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

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

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

3.2.2 Voraussetzungen für den sicheren Gebrauch

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

- Bei Schäden infolge von Nichtbeachtung der Sicherheitshinweise übernimmt Rexroth keine Haftung.
- Vor der Inbetriebnahme sind die Betriebs-, Wartungs- und Sicherheitshinweise durchzulesen. Wenn die Anwendungsdokumentation in der

vorliegenden Sprache nicht einwandfrei verstanden wird, bitte beim Lieferanten anfragen und diesen informieren.

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

Der Maschinenhersteller und Anlagenerrichter muss für seine individuelle Anwendung die Eignung

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

Die Einhaltung der durch die nationalen Vorschriften geforderten Grenzwerte liegt in der Verantwortung des Herstellers der Anlage oder Maschine.

- Die technischen Daten, die Anschluss- und Installationsbedingungen der Komponenten sind den zugehörigen Anwendungsdokumentationen zu entnehmen und unbedingt einzuhalten.

Länderspezifische Vorschriften, die vom Anwender zu berücksichtigen sind

- Europäische Länder: entsprechend Euronormen EN
- Vereinigte Staaten von Amerika (USA):

- Nationale Vorschriften für Elektrik (NEC)
- Nationale Vereinigung der Hersteller von elektrischen Anlagen (NEMA) sowie regionale Bauvorschriften
- Vorschriften der National Fire Protection Association (NFPA)
- Kanada: Canadian Standards Association (CSA)
- Andere Länder:
 - International Organization for Standardization (ISO)
 - International Electrotechnical Commission (IEC)

3.2.3 Gefahren durch falschen Gebrauch

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

3.3 Gefahrenbezogene Hinweise

3.3.1 Schutz gegen Berühren elektrischer Teile und von Gehäusen



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

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

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

- Bedienung, Wartung und/oder Instandsetzung der Komponenten des elektrischen Antriebs- und Steuerungssystems darf nur durch qualifiziertes Personal erfolgen.
- Beachten Sie die allgemeinen Errichtungs- und Sicherheitsvorschriften zu Arbeiten an Starkstromanlagen.

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

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

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

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

Tab. 3-1: Mindestquerschnitt der Schutzleiterverbindung

3.3.2 Schutzkleinspannung als Schutz gegen elektrischen Schlag

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

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

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

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

3.3.3 Schutz vor gefährlichen Bewegungen

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

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

- Fehler in der Software oder in der Firmware

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

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

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

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

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

Vermeidung von Unfällen, Körperverletzung und/oder Sachschaden:

- Halten Sie sich nicht im Bewegungsbereich der Maschine und Maschinenteilen auf. Verhindern Sie den unbeabsichtigten Zugang für Personen, z. B. durch
 - Schutzzaun
 - Schutzgitter
 - Schutzabdeckung
 - Lichtschranke
- Stellen Sie eine ausreichende Festigkeit der Schutzzäune und Schutzabdeckungen gegen die maximal mögliche Bewegungsenergie sicher.
- Ordnen Sie NOT-HALT-Schalter leicht zugänglich und schnell erreichbar an. Prüfen Sie die Funktion der NOT-HALT-Einrichtung vor der Inbetriebnahme. Unterlassen Sie den Betrieb der Maschine bei Fehlfunktion des NOT-HALT-Schalters.
- Stellen Sie sicher, dass es nicht zu einem unbeabsichtigten Anlauf kommt. Schalten Sie den Leistungsanschluss der Antriebe über Ausschalter/-taster frei oder verwenden Sie eine sichere Anlaufsperre.
- Bringen Sie vor dem Zugriff oder Zutritt in den Gefahrenbereich die Antriebe sicher zum Stillstand.
- Sichern Sie zusätzlich vertikale Achsen gegen Herabfallen oder Absinken nach Abschalten des Motors, z. B. durch
 - mechanische Verriegelung der vertikalen Achse,
 - externe Brems-/ Fang-/ Klemmeinrichtung oder
 - ausreichenden Gewichtsausgleich der Achse.
- Die serienmäßig gelieferte **Motor-Haltebremse** oder eine externe, vom Antriebsregelgerät angesteuerte Haltebremse **alleine ist nicht für den Personenschutz geeignet!**

- Schalten Sie die Komponenten des elektrischen Antriebs- und Steuerungssystems über den Hauptschalter spannungsfrei und sichern Sie diese gegen Wiedereinschalten bei:
 - Wartungsarbeiten und Instandsetzung
 - Reinigungsarbeiten
 - langen Betriebsunterbrechungen
- Vermeiden Sie den Betrieb von Hochfrequenz-, Fernsteuer- und Funkgeräten in der Nähe von Komponenten des elektrischen Antriebs- und Steuerungssystems und deren Zuleitungen. Wenn ein Gebrauch dieser Geräte unvermeidlich ist, prüfen Sie bei Erstinbetriebnahme des elektrischen Antriebs- und Steuerungssystems die Maschine oder Anlage auf mögliche Fehlfunktionen bei Betrieb solcher Hochfrequenz-, Fernsteuer- oder Funkgeräte in deren möglichen Gebrauchslagen. Eventuell ist eine spezielle EMV-Prüfung notwendig.

3.3.4 Schutz vor elektromagnetischen und magnetischen Feldern bei Betrieb und Montage

Elektromagnetische und magnetische Felder!

Gesundheitsgefahr für Personen mit aktiven Körperhilfsmitteln (AIMD) wie Herzschrittmachern oder passiven metallischen Implantaten.

- Gefahr durch elektromagnetische und magnetische Felder in unmittelbarer Nähe von Antriebsregelgeräten und dazugehörigen stromführenden Leiter für oben genannte Personengruppen.
- Zutritt zu diesen Bereichen kann für oben genannte Personengruppen ein erhöhtes Risiko darstellen, es sollte der behandelnde Arzt konsultiert werden.
- Bei möglichen Auswirkungen auf oben genannte Personen während des Betriebs von Antriebsregelgeräten und Zubehöerteilen die gefährdeten Personen aus der Nähe von Zuleitungen und Geräten bringen.

3.3.5 Schutz gegen Berühren heißer Teile

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

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

Endanwendung vom Hersteller der Maschine oder Anlage vorzunehmen. Diese Maßnahmen können beispielsweise sein: Warnhinweise an der Maschine oder Anlage, trennende Schutzeinrichtung (Abschirmung oder Absperrung) oder Sicherheitshinweise in der Anwendungsdokumentation.

3.3.6 Schutz bei Handhabung und Montage

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

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

3.3.7 Schutz beim Umgang mit Batterien

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

Verletzungsgefahr durch unsachgemäße Handhabung!

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



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

3.3.8 Schutz vor unter Druck stehenden Leitungen

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

Verletzungsgefahr durch unsachgemäße Handhabung von unter Druck stehenden Leitungen!

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



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

3.4 Erläuterung der Signalwörter und der Signalgrafik

Die Sicherheitshinweise in der vorliegenden Anwendungsdokumentation beinhalten bestimmte Signalwörter (Gefahr, Warnung, Vorsicht, Hinweis) und ggf. eine Signalgrafik (nach ANSI Z535.6-2011).

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

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

GEFAHR

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

WARNUNG

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

VORSICHT

Bei Nichtbeachtung dieses Sicherheitshinweises können mittelschwere oder leichte Körpervletzung eintreten.

HINWEIS

Bei Nichtbeachtung dieses Sicherheitshinweises können Sachschäden eintreten.

4 Grundfunktionen von Rexroth IndraMotion MLD

4.1 Übersicht

Das Laufzeitsystem der antriebsintegrierten SPS in ihren Ausführungsformen MLD-S bzw. MLD-M bettet sich als Modul in das vorhandene Antriebslaufzeitsystem mit seinen Funktionen und Schnittstellen ein.

Die elementaren Grundfunktionen eines Antriebs mit integrierter SPS und ihr Zusammenspiel zeigt nachfolgende Grafik (hier MLD-S):

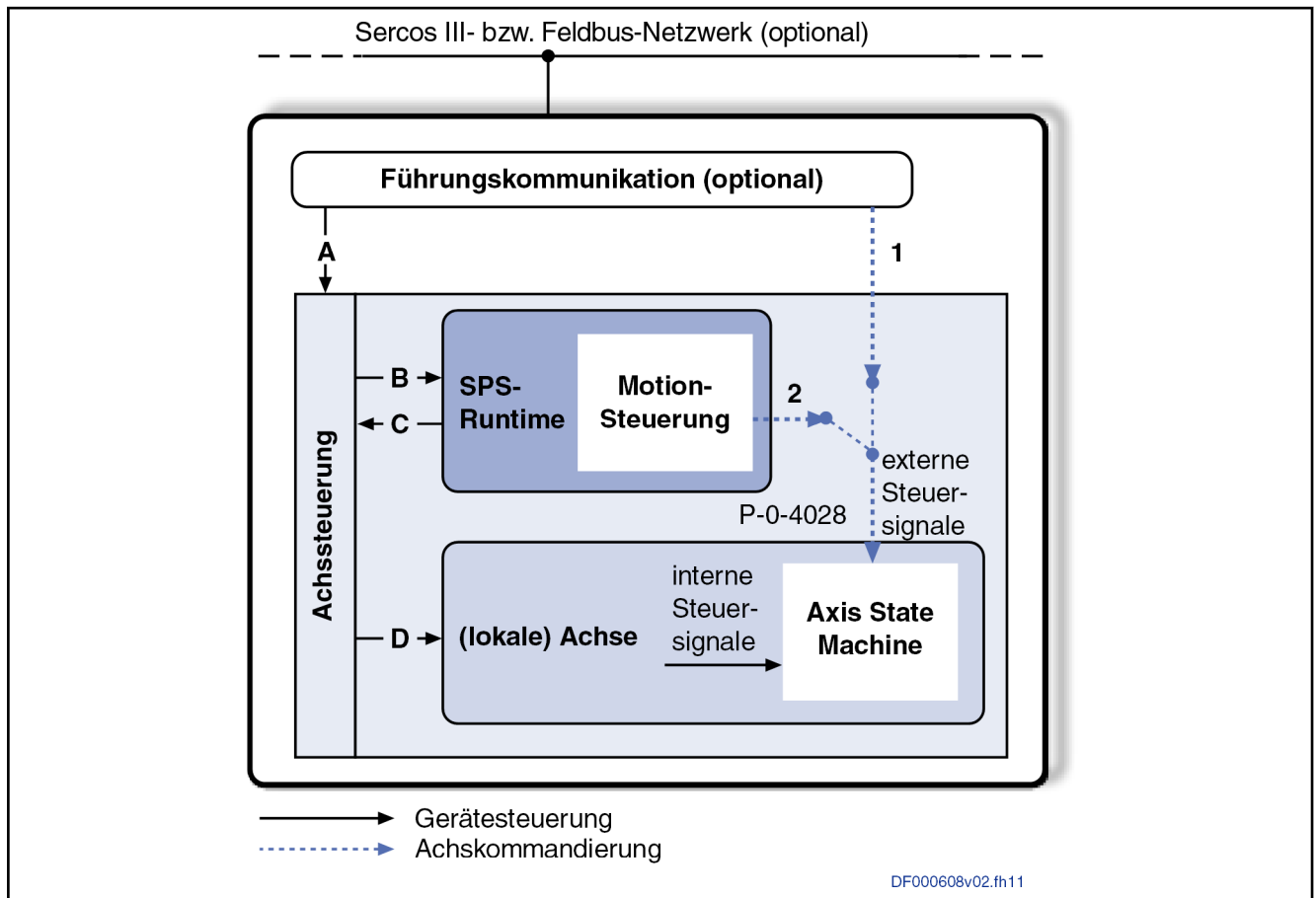


Abb. 4-1: Übersicht Gerätesteuerung und Achskontrolle, MLD-S

4.2 Begriffsklärung

MLD-S, MLD-M, MLD

Die antriebsintegrierte SPS "Rexroth IndraMotion MLD" ist in unterschiedlichen Varianten verfügbar.

Die Bezeichnung der Variante für Einzelachsenanwendungen lautet "MLD-S". Die Ansteuermöglichkeiten der SPS beschränken sich dann geräteintern auf die "lokale Achse" und den virtuellen Leitachsgenerator.

Sind über die "CC"-Option [Sercos III Master (Querkommunikation)] Antriebe mit Sercos III Slave interface angeschlossen, kann die SPS diese Achsen ansteuern. Die Bezeichnung dieser Variante lautet "MLD-M".

Die allgemeine Kurzform "MLD" wird verwendet, wenn unabhängige Funktionalitäten beschrieben werden.

Motion-Steuerung	Motion-Steuerung ist der Überbegriff für die Regelung von Position, Geschwindigkeit, Beschleunigung oder einer Kombination daraus. • Technologie-Bausteine Bausteine zur Erweiterung der Antriebsfunktionalität • PLCopen-Bausteine Die Motionsteuerung über das SPS-Programm wird mittels Funktionsbausteinen nach PLCopen durchgeführt. Die PLCopen hat hierfür eine Reihe von IEC 61131-Bausteine definiert, mit deren Hilfe man die Achsen ansteuern kann. Neben den durch die PLCopen bereits definierten Bausteinen existieren noch zusätzliche an die Norm angelehnte Bausteine. • AxisInterface Das Achs-Interface bündelt und erweitert PLCopen-Bewegungs-Funktionsbausteine und stellt ein einfach zu bedienendes Interface für die Antriebs-Funktionalität zur Verfügung. Das Achs-Interface enthält Steuerungssignale und Parameter für die verschiedenen Betriebsarten der Leitachse und Folgeachse sowie Einstellmöglichkeiten für angewählte Prozesswerte. Das Achs-Interface besteht aus SPS-Funktionsbausteinen, die im Anwendungsprogramm mitlaufen. • GAT compact Das GAT compact ist ein Beispielprojekt, basierend auf IEC 61131-3. Es wird unter anderem mit dem IndraWorks Engineering Framework für IndraMotion MLD zur Verfügung gestellt und lässt sich Wizard-unterstützt laden und in Betrieb nehmen.
CCD	"CCD" ist die Abkürzung für Cross Communication Drives , der auf Sercos III basierenden Schnittstelle für die Querkommunikation. Geräte der Antriebsfamilie "Rexroth IndraDrive" können mit "CCD"-Option konfiguriert sein, um eine elektronische (digitale) Kopplung von Antrieben und E/A-Geräten zu ermöglichen.
CCD-Master /CCD-Slave	Als "CCD-Master" wird ein Antrieb mit "CCD"-Option (Sercos III Master interface) bezeichnet, der für "CCD-Slaves" [Antrieb mit "CCD"-Option (Sercos III Slave interface)] eines CCD-Verbunds eine externe Steuerung darstellt.
Lokale Achse	Als "lokale Achse" wird bei MLD-S die Achse des Antriebs und bei MLD-M die Achse im Antrieb mit der "CCD"-Option (Sercos III Master interface) bezeichnet.
Entfernte Achsen	Als "entfernte Achsen" werden bei MLD-M die Achsen in den CCD-Slaves bezeichnet.
Virtuelle Leitachse	Der Antrieb enthält einen Leitachsgenerator. Dieser kann ähnlich wie eine reale Achse angesteuert werden und simuliert dann eine Achsbewegung. Die Position dieser Leitachse kann als Vorgabe für die lokale oder auch entfernte Achse herangezogen werden. Die MLD kann diese Leitachse über die Motion-Funktionsbausteine ansteuern.
AT	"AT" ist die Abkürzung für Antriebstelegramm . Das Antriebstelegramm wird über den Echtzeitdatenkanal vom Slave zum Master gesendet.
MDT	"MDT" ist die Abkürzung für Master-Daten-Telegramm . Das Master-Daten-Telegramm wird über den Echtzeitdatenkanal vom Master zum Slave gesendet.
PAE	"PAE" ist die Abkürzung für "Prozessabbild der Eingänge" . Im Taskzyklus werden vor dem Aufruf des Programmcodes die Schaltzustände an den Eingängen gelesen und im PAE gespeichert. Mit diesen Informationen wird anschließend das Steuerungsprogramm versorgt und abgearbeitet.

PAA "PAA" ist die Abkürzung für "**P**rozess**a**bbild der **A**usgänge". Die Zustände im PAA werden am Taskende nach der Abarbeitung des Programmcodes an die physikalischen Ausgänge übertragen.

Hochlauf-Modus Bei Antrieben mit Sercos III interface als Führungskommunikation, kann über den Hochlauf-Modus die Verbindung zwischen Kommunikationsphase und Gerätezustand entkoppelt werden. Bei Antrieben ohne Führungskommunikationsschnittstelle oder bei Antrieben mit einer von Sercos III interface verschiedenen Führungskommunikationsschnittstelle, definiert der Hochlauf-Modus welchen Gerätezustand der Antrieb nach dem Booten erreicht (Betriebs- oder Parametriermodus).

4.3 Gerätesteuerung

4.3.1 Allgemeines

Über die Gerätesteuerung wird definiert, wie die grundlegenden Betriebszustände der elementaren Systemblöcke zusammenwirken.

4.3.2 Führungskommunikation

Über die Führungskommunikation (Sercos Slave interface bzw. Feldbus-Slave-Interface) kann der Antrieb (hier mit eingebetteter MLD) an eine Leitsteuerung angekoppelt sein. Die busspezifische Zustandsmaschine steuert die Initialisierung des Kommunikationssystems (bis zur Etablierung des zyklischen Datenaustauschs; Beispiel: Sercos Kommunikationsphasen 0..4). Die Steuerung dieser Zustandsmaschine erfolgt durch den Busmaster bzw. den Zustand des Busses selbst.

Eine busfähige Führungskommunikationsschnittstelle ist optional.

Siehe auch Firmware-Funktionsbeschreibung: "Kommunikationsphasen der Führungskommunikation"

4.3.3 Achssteuerung

Die Achssteuerung bildet den elementaren Verwaltungsrahmen für alle im Antrieb vorhandenen Funktionen, wie Achse oder integrierte SPS. Sie verfügt über die Zustände Initialisierung, Parametriermodus und Betriebsmodus.

Die Zustandsmaschine der Achssteuerung ist bei Führungskommunikation Sercos an diese gekoppelt, d.h. die Initialisierung des Sercos Rings (Umschaltung bis Kommunikationsphase 4) führt automatisch zur Umschaltung der Achssteuerung in den Betriebsmodus [siehe **(A)** in [Kap. 4.1 "Übersicht auf Seite 47](#)]. Über den Hochlauf-Modus besteht bei Sercos III interface die Möglichkeit, die Kopplung zwischen Kommunikationsphase und Geräte-Zustandsmaschine aufzuheben.

Bei nicht vorhandener Führungskommunikation bzw. allen Führungskommunikations-Schnittstellen außer Sercos III, kann die Achse unabhängig vom Bussystem gesteuert werden. Über den Hochlauf-Modus kann definiert werden, bis in welchen Gerätezustand (Betriebs- oder Parametriermodus) die Achse nach dem Booten hochläuft. Ab der Firmware-Version MPx17 ist dies mittels der Kommandos "S-0-0420, C0400 Kommando Parametrierebene 1 aktivieren" und "S-0-0422, C0200 Kommando Parametrierebene beenden" auch bei Sercos möglich.

Siehe auch Firmware-Funktionsbeschreibung "Geräteinterne Zustandsmaschine".

4.3.4 Achse

Die (lokale) Achse ist für die Erzeugung bzw. Aufbereitung der Achssollwerte und für die Antriebsregelung zuständig. Die Implementierung ist gemäß

Sercos Spezifikation ausgeführt. Die elementaren externen Steuerinformationen zur Achskommandierung sind folgende:

- "Antrieb Ein"
- "Antrieb Freigabe"
- "Antrieb Halt"
- Sollbetriebsart-Vorgabe

Diese Signale steuern die Zustandsmaschine der Achse.

Die Steuerinformationen wirken erst, wenn sich die Achssteuerung im Zustand "Betriebsmodus" befindet. Die Zustandsmaschine der Achssteuerung überlagert somit die Achs-Zustandsmaschine [siehe (D) in [Kap. 4.1 "Übersicht" auf Seite 47](#)].



Im Parametriermodus der Achssteuerung ist keine Achskommandierung möglich. Die Achse befindet sich dann ebenfalls im Parametriermodus!

4.3.5 SPS-Runtime

Das Laufzeitsystem der antriebsintegrierten SPS verfügt über die Elementarzustände "STOP" und "RUN". Die Umschaltung von "STOP" auf "RUN" während der Systeminitialisierung erfolgt in Abhängigkeit der Gerätesteuerung [siehe (B) in [Kap. 4.1 "Übersicht" auf Seite 47](#)]. Über "P-0-1367, SPS Konfiguration" (Bits 1, 0) oder den IndraWorks-Dialog sind mehrere Optionen einstellbar:

- "00" / "Default-Startverhalten": Die SPS startet beim Übergang der Achssteuerung von Initialisierungsmodus in den Parametriermodus, falls sie beim Ausschalten in RUN war. Sonst verbleibt sie in STOP (Default-Startverhalten).
- "01" / "Start beim Booten": Wie "00", jedoch unabhängig vom letzten Zustand (geht in RUN).
- "10" / "Stop": Die SPS verbleibt im Zustand STOP.
- "11" / "Start im Betriebsmodus": Die SPS startet beim Übergang der Achssteuerung vom Parametriermodus in den Betriebsmodus.

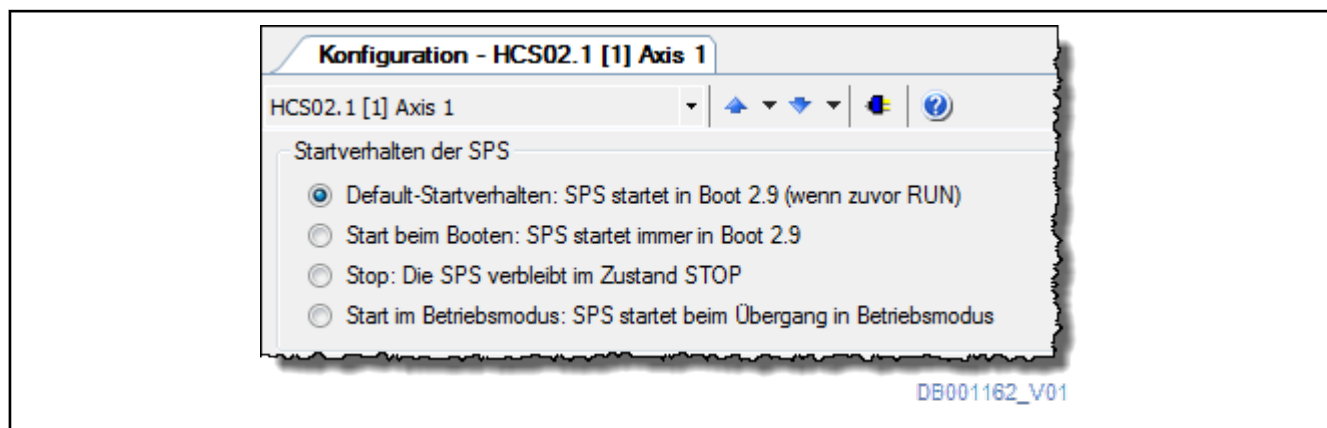


Abb. 4-2: Einstellung Startverhalten der SPS-Runtime

4.3.6 Eingriffsmöglichkeiten der MLD auf die Achssteuerung

Allgemeines

Durch den Firmware-Funktionsbaustein "MX_SetDeviceMode" kann im SPS-Anwenderprogramm Einfluss auf den Initialisierungszustand der lokalen

Achssteuerung und damit auch der lokalen Achse genommen werden [siehe (C) in [Kap. "Gerätesteuerung bei MLD-M" auf Seite 51](#)]. Dies ist notwendig, wenn beispielsweise elementare Konfigurationsänderungen, d.h. Parametermodifikationen, die nur im Parametriermodus möglich sind, durchgeführt werden sollen. Dies ist auch bei Sercos als Führungskommunikation möglich.

Gerätesteuerung bei MLD-M

Die Gerätesteuerung des CCD-Verbunds bei der Ausführungsform MLD-M basiert auf den Grundfunktionen der MLD-S.

Folgende Ergänzungen zur Ausführungsform MLD-S sind zu beachten (siehe auch nachfolgende Grafik):

- Als zusätzlicher Systemblock in der MLD-M (im Gerät "CCD-Master") ist das Sercos III Master interface vorhanden. Dieses definiert mit seiner Zustandsmaschine den CCD-Verbund (Sercos III-Netzwerk) und damit auch den Hochlauf der angeschlossenen CCD-Slave-Geräte.
- Die Kopplungseigenschaft zwischen der Zustandsmaschine des CCD-Verbunds und der Achssteuerung im CCD-Master kann konfiguriert werden (P-0-1800.0.1, Bit 5). Sind diese nicht gekoppelt, dann bewirkt der Parametriermodus der Achssteuerung im CCD-Master keine Umschaltung des CCD-Verbunds in Kommunikationsphase 2.

- Im gekoppelten Modus (P-0-1800.0.1, Bit 5="1") wird die Zustandsmaschine des Sercos III Master interfaces durch die Achssteuerung im CCD-Master vorgegeben (E), d.h. Parametriermodus der Achssteuerung im CCD-Master bedeutet Kommunikationsphase 2 im CCD-Verbund bzw. Betriebsmodus der Achssteuerung im CCD-Master bedeutet Kommunikationsphase 4 im CCD-Verbund.

Im gekoppelten Modus kann die MLD-M über den Firmware-Funktionsbaustein "MX_SetDeviceMode" den Zustand der Achssteuerung und damit auch den Zustand des CCD-Verbunds steuern. "MX_SetDeviceMode" kann nur auf die lokale Achse (und damit ggf. auf den Verbund als Ganzes) wirken. Der Aufruf von "MX_SetDeviceMode" (lokale Achse im CCD-Master) mit Eingang "OperationMode"=FALSE bewirkt, dass das der CCD-Master in den Parametriermodus und damit der CCD-Verbund und die daran angeschlossenen CCD-Slaves in Kommunikationsphase 2 geschaltet werden. Dies bewirkt bei den CCD-Slaves ebenfalls ein Schalten in den Parametriermodus; der komplette CCD-Verbund wird also in den Parametriermodus gesetzt.

- Im ungekoppelten Modus (P-0-1800.0.1, Bit 5="0") muss mit Hilfe des Kommandos "P-0-1802.0.1, C7400 CCD: Umschaltung nach Phase 2" zuerst die CCD-Kommunikation in Phase 2 geschaltet werden, um Parameter im CCD-Verbund schreiben zu können.

Grundfunktionen von Rexroth IndraMotion MLD

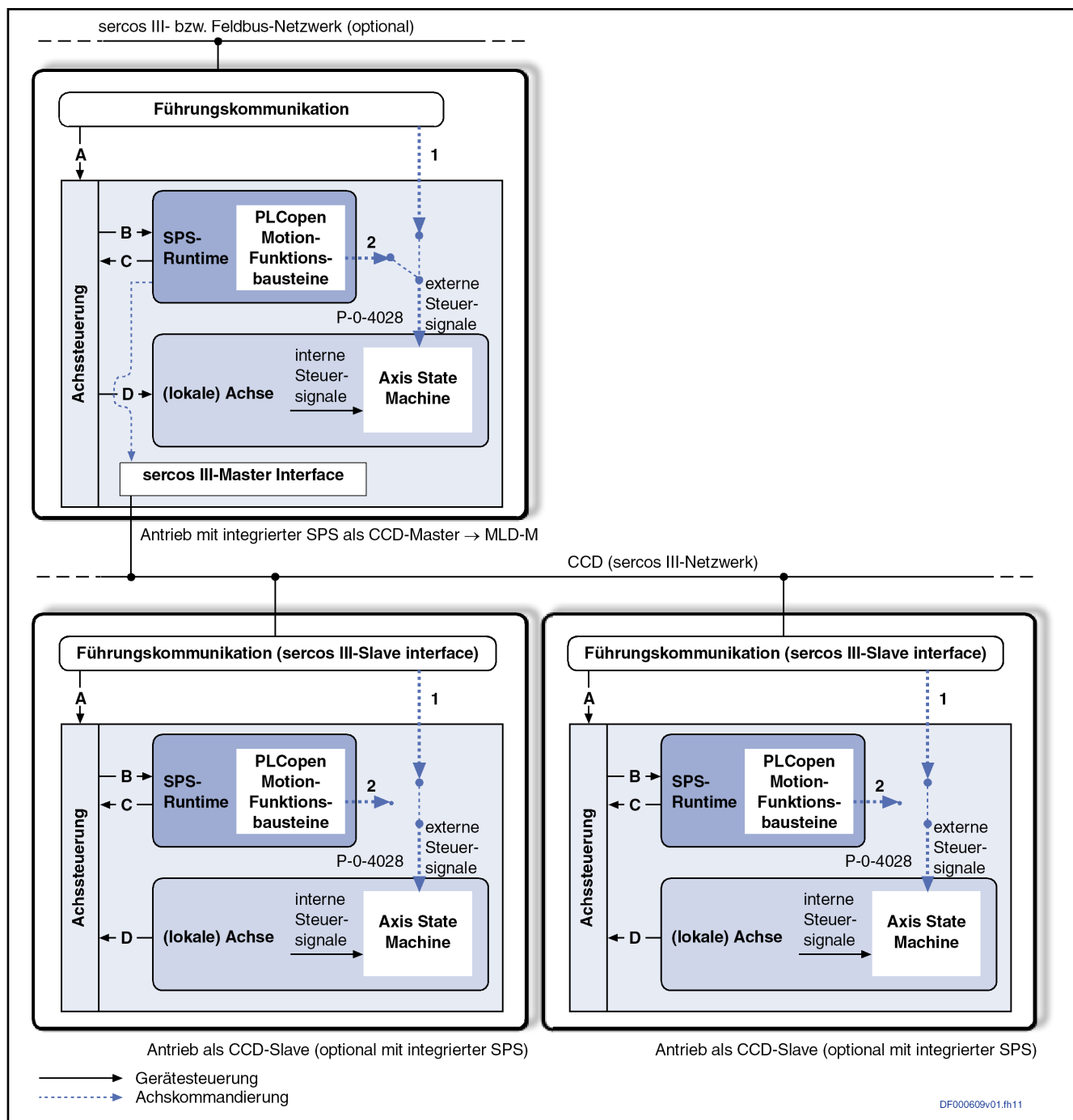


Abb. 4-3: Übersicht Gerätesteuerung und Achskommandierung MLD-M

Funktionsbaustein und systemglobale Variable der MLD zur Achssteuerung

Funktionsbaustein:

- "MX_SetDeviceMode": Firmware-Funktionsbaustein zur Achssteuerung; verwendet S-0-0420, S-0-0422, S-0-0424

Systemglobale Variable:

- MX_AXISDATA AxisData[<Axis>].dwDriveStatus_i, Bits 15,14: Anzeige des elementaren Betriebszustandes von Achse und Achssteuerung (siehe Marginalie "Aufbau der AxisData")

Parameter und Diagnosen zur Gerätesteuerung

Zugehörige Parameter	• S-0-0128, C5200 Umschaltvorbereitung auf Komm.-Phase 4 • S-0-0420, C0400 Kommando Parametrierebene 1 aktivieren • S-0-0422, C0200 Kommando Parametrierebene beenden • S-0-0423, IDN-Liste der ungültigen Daten der Parametrierebenen • S-0-0424, Status Parametrierebene • P-0-1350, SPS Steuerwort • P-0-1367, SPS Konfiguration • P-0-1802.0.1, C7400 CCD: Umschaltung nach Phase 2 • P-0-1802.0.2, C7500 CCD: Umschaltung nach Phase 4 • P-0-4086, Führungskommunikation Status
Zugehörige Diagnosen	• Diagnosen der Betriebszustände der Sercos Führungskommunikation: P-1, P0, .., P3 • Diagnose der Betriebszustände der Achssteuerung: OM, PM • Diagnose der Betriebszustände der Achse: bb, Ab, AF, .. • Kommandos der Gerätesteuerung / Zustandsumschaltung der Führungskommunikation, Achssteuerung und Achse: C0100, C0200, C0400, C5200, C7400, C7500 • Kommandofehler bei der Gerätesteuerung: C01xx, C02xx, C04xx, C52xx, C74xx, C75xx

4.4 Achskommandierung

4.4.1 Grundlegendes zur Achskontrolle

Allgemeines

Der Betriebszustand der Achse wird durch externe und interne Steuersignale (Fehler, Leistung_Vorhanden, ..) definiert. Die elementaren externen Steuerinformationen zur Achskommandierung sind folgende

- "Antrieb Ein"
- "Antrieb Freigabe"
- "Antrieb Halt"
- Sollbetriebsart-Vorgabe

Außerdem existieren Antriebs-Steuerkommandos zur Aktivierung von komplexen, vorkonfigurierten Kommandos, wie "Antriebsgeführtes Referenzieren", "Autojustage-Funktionen" etc.

Die Vorgabe der externen Steuersignale erfolgt abhängig vom Anwendungsfall durch eine externe Steuerung über die Führungskommunikation oder durch die integrierte SPS (MLD). Daraus ergeben sich die folgenden Anwendungsfälle mit der dafür notwendigen Konfiguration:

Grundfunktionen von Rexroth IndraMotion MLD

Anwendungsfall	Konfiguration Achskontrolle in MLD-Gerät ("P-0-1367, SPS Konfiguration", Bit 4)	Konfiguration (keine MLD aktiv) oder Achskontrolle in CCD-Slaves ("P-0-1367, SPS Konfiguration", Bit 4)	Weiterführende Hinweise
MLD-S als "Intelligente Servoachse"	0: keine permanente Kontrolle	nicht relevant	Achskontrolle durch die MLD nur temporär möglich
MLD-S als "Autarke Einzelachs-"Motion Logic Control"	1: permanente Kontrolle	nicht relevant	--
MLD-M als "Autarke Mehrachs-"Motion Logic Control"	1: permanente Kontrolle	(keine MLD aktiv) oder 0: keine permanente Kontrolle	P-0-1800.0.1, CCD Konfiguration, Bits 3, 4 = "01": MLD-M-Systemmodus notwendig!

Tab. 4-1: Vorgabe der externen Steuersignale

Konfiguration der MLD mit IndraWorks

Die Konfiguration der MLD erfolgt über IndraWorks in folgendem Dialog:

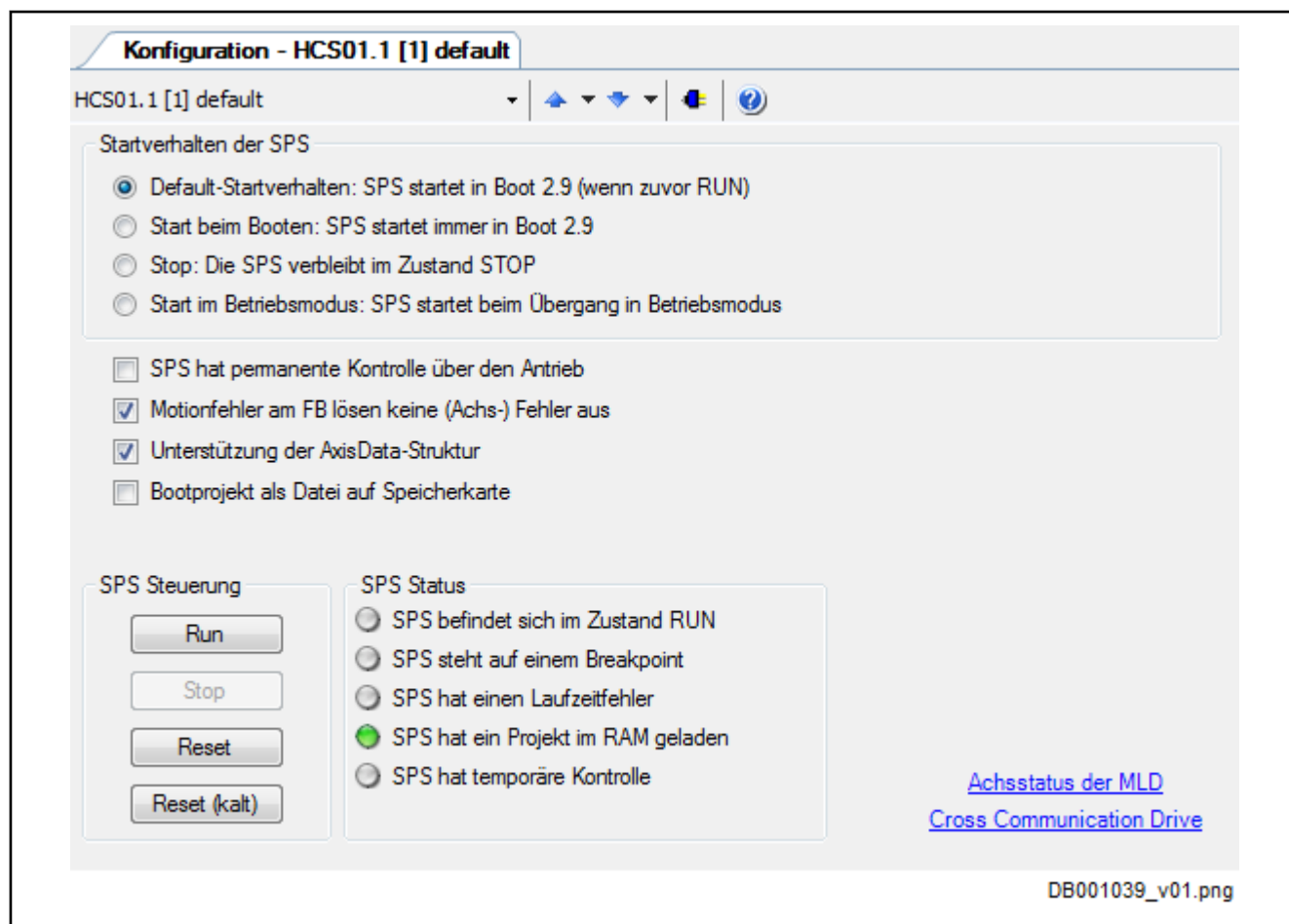


Abb. 4-4: Konfiguration der MLD mit IndraWorks

Bei der Verwendung der MLD-M als Mehrachssteuerung wird der MLD-M-Systemmodus aktiviert, damit die angeschlossenen Achsen den Vorgaben der MLD-M folgen.

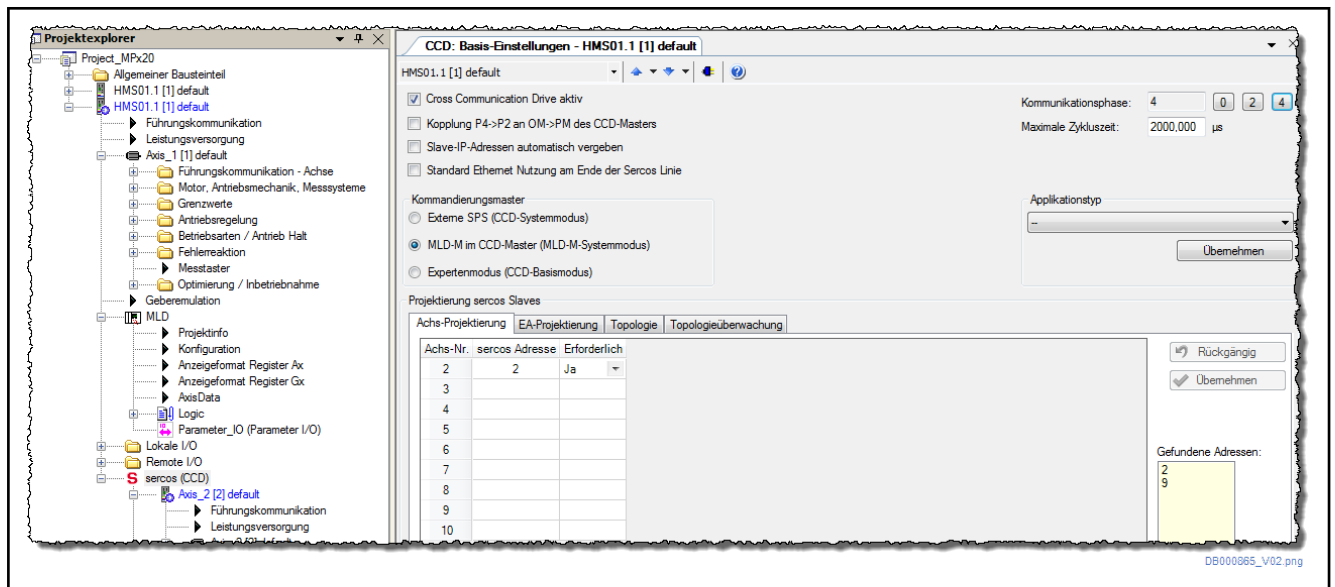


Abb. 4-5: Konfiguration der MLD mit IndraWorks, Auswahl CCD-Einstellung "MLD-M-Systemmodus"

MLD-S als "Intelligente Servoachse"

Bei der Verwendung der MLD-S als "Intelligente Servoachse" besitzt standardmäßig die externe Steuerung die Kontrolle über die (lokale) Achse (6). Die externen Steuersignale werden über das spezifische Steuerwort der jeweiligen Führungskommunikations-Schnittstelle vorgegeben.

Die MLD kann sich bei Bedarf über den Firmware-Funktionsbaustein "MX_SetControl" temporär die Achskontrolle erteilen (7). Dies kann beispielsweise bei einer intelligenten, dezentralen Fehlerreaktion notwendig werden. Damit ist die MLD temporär in der Lage die Achse über PLCopen-konforme Motion-Funktionsbausteine zu steuern.

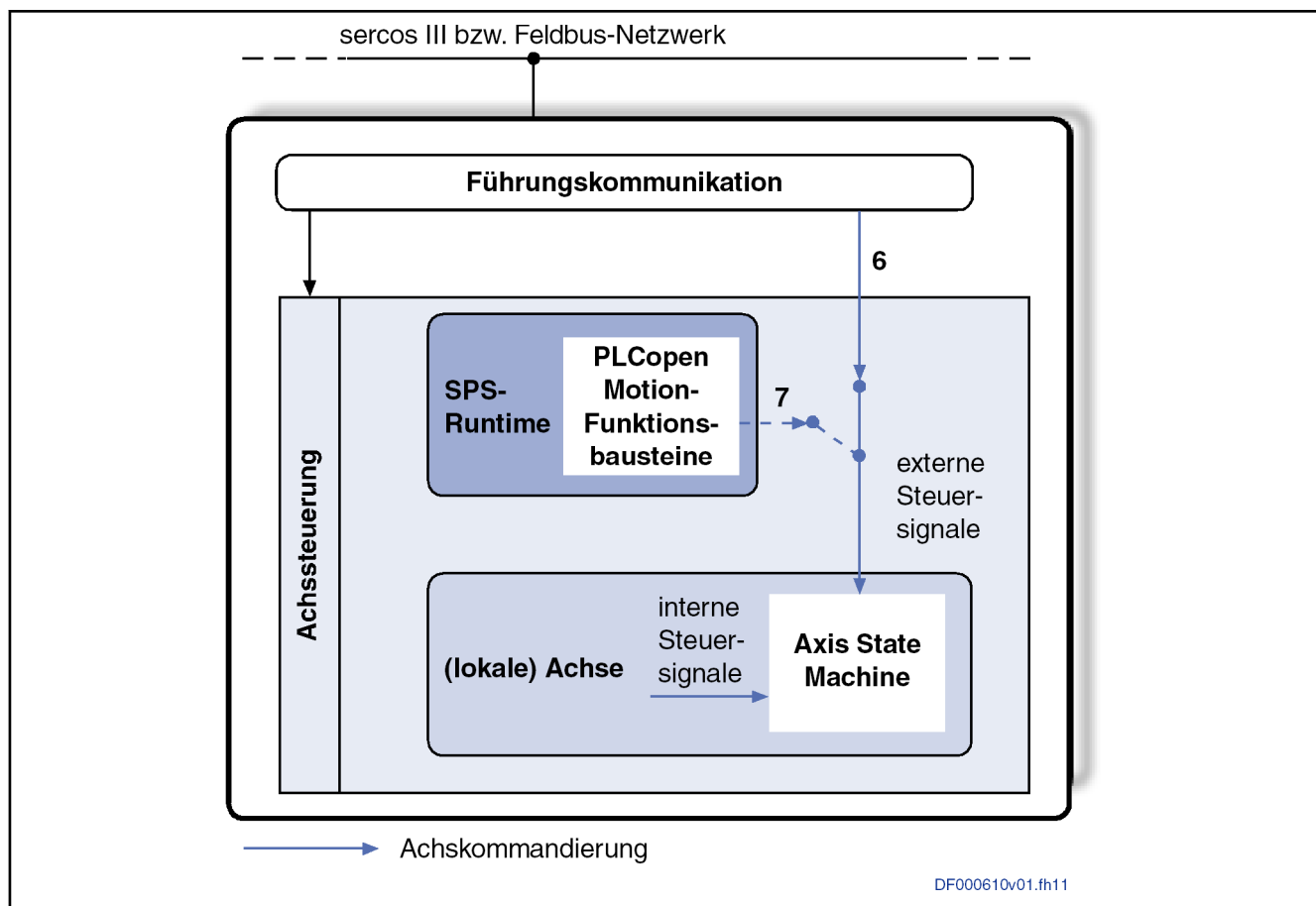


Abb. 4-6: Achskommandierung MLD-S als "Intelligente Servoachse"

MLD-S als "Autarke Einzelachs-"Motion Logic Control"

Bei der Verwendung der MLD-S als "Autarke Einzelachs-"Motion Logic Control" besitzt die MLD ständig die Achskontrolle und kann damit eine externe Motion-Kommandierung komplett ersetzen (7). Ist eine Führungskommunikation vorhanden, so werden darüber im Normalfall nur noch Prozessdaten mit dem Antrieb bzw. der MLD ausgetauscht und ein Steuerwort zur Achskommandierung wird nicht mit übertragen. Ist ein Steuerwort trotzdem aufgelegt, so wird lediglich das Signal "Antrieb-Freigabe" wirksam, sofern dies das busspezifische Steuerwort unterstützt (nur bei Sercos).

Die externen Steuersignale "Antrieb Ein", "Antrieb Halt" sowie die Sollbetriebsart-Vorgabe werden indirekt durch die PLCopen Motion-Funktionsbausteine, die im SPS-Anwenderprogramm aufgerufen werden, bedient.

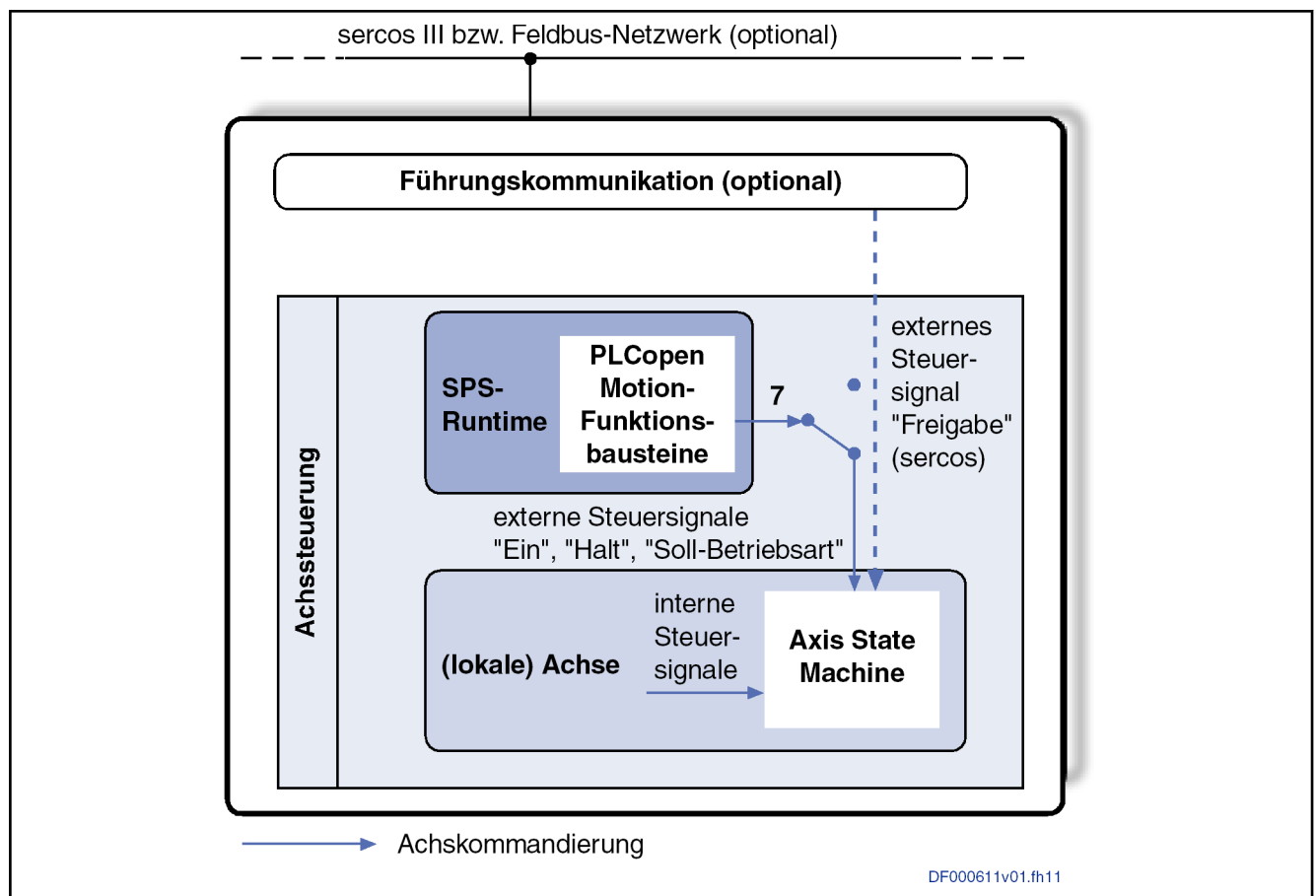


Abb. 4-7: Achskommandierung MLD-S als "Autarke Motion Logic Control"

MLD-M als "Autarke Mehrachs-"Motion Logic Control"

Bei der Verwendung der MLD-M als "Autarke Mehrachs-"Motion Logic Control" kommandiert die integrierte SPS im CCD-Master (MLD-M) zusätzlich zu der lokalen Achse auch noch bis zu 9 entfernte Achsen in den CCD-Slaves. Die MLD-M im CCD-Master stellt für diese CCD-Slaves eine externe Steuerung dar, die über die Führungskommunikation Sercos III deren Achsen kontrolliert.

Diese Struktur stellt somit eine Mischform der Verwendung der MLD-S als "Intelligente Servoachse" und der Verwendung der MLD-S als "Autarke Einzelachs-"Motion Logic Control" dar. Während die lokale Achse im CCD-Master unter permanenter Kontrolle durch die im gleichen Gerät vorhandene MLD-M steht, sind die in den CCD-Slaves integrierten speicherprogrammierbaren Steuerungen nur optional vorhanden und dienen maximal zur Realisierung einer "intelligenten Servoachse". Die Kommandierung erfolgt standardmäßig durch die MLD-M im CCD-Master.

Grundfunktionen von Rexroth IndraMotion MLD

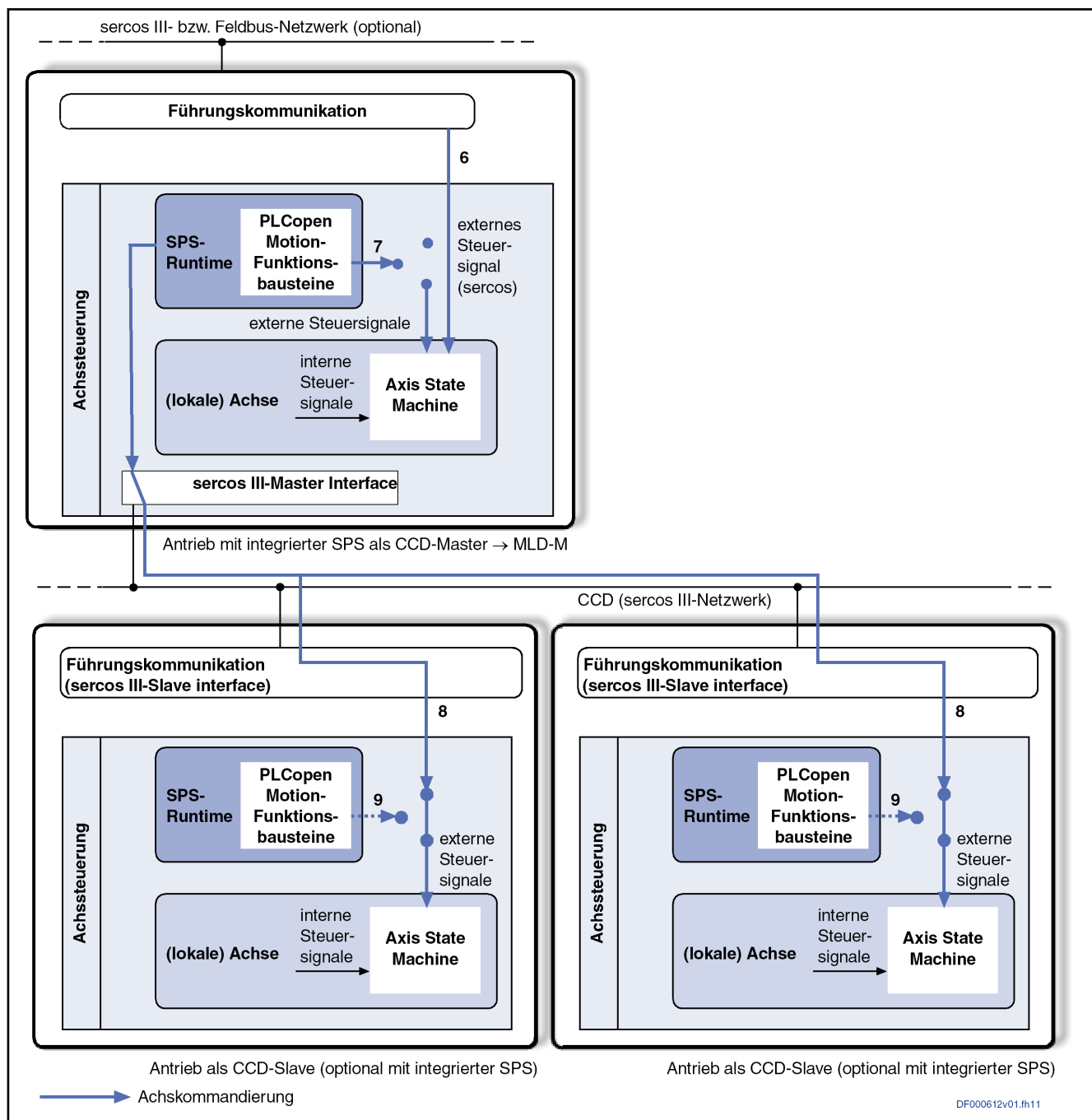


Abb. 4-8: Achskommandierung MLD-M als "Autarke Mehrachs-"Motion Logic Control"

4.4.2 Umsetzung der Motionkommandierung in der Achse

Die Achs-Kommandierung durch die MLD erfolgt durch den Aufruf von PLCopen-konformen Motion-Funktionsbausteinen im SPS-Anwenderprogramm. Die Motion-Funktionsbausteine bewirken über die Vorgabe von Steuersignalen sowie Sollwertparametern die Aktivierung von Sercos basierten Betriebsarten in der Achse.

Dabei sind die folgenden Verwendungen zu unterscheiden:

Anwendungsfall	Betriebsarten zur Realisierung der PLCopen-konformen Motion-Funktionsbausteine
MLD-S als "Intelligente Servoachse"	Verwendung der internen Betriebsarten "Momentenregelung", "Geschwindigkeitsregelung" und "Antriebsgeführtes Positionieren" als unabhängige Instanzen der äquivalenten Sercos Betriebsarten mit den Parametern P-0-1450 bis P-0-1465 Dies stellt sicher, dass bei temporärer Kontrolle die Betriebsarten-Einstellungen der externen Steuerung nicht unzulässig manipuliert werden.
MLD-S als "Autarke Einzelachs-"Motion Logic Control" oder MLD-M als "Autarke Mehrachs-"Motion Logic Control"	Verwendung der Hauptbetriebsart und der 1. bis 7. Nebenbetriebsart (S-0-0032 bis S-0-0035, S-0-0284) mit den vorbesetzten Betriebsarten "Geschwindigkeitsregelung", "Momentenregelung", "Lagesynchronisation", "Antriebsgeführtes Positionieren" und "Geschwindigkeitssynchronisation" und somit den dort verwendeten Parametern (Details siehe "Motion-Kommandokanal")

4.4.3 Grundsätzliche Funktionsweise von PLCopen-konformen Motion-Funktionsbausteinen

Timing

Am Timingdiagramm ist die Arbeitsweise der Ein- und Ausgänge dargestellt. "Done" wird hier als "Execute" des nächsten Bausteins verwendet. Im zweiten Fall wird das Signal noch logisch verknüpft.

Grundfunktionen von Rexroth IndraMotion MLD

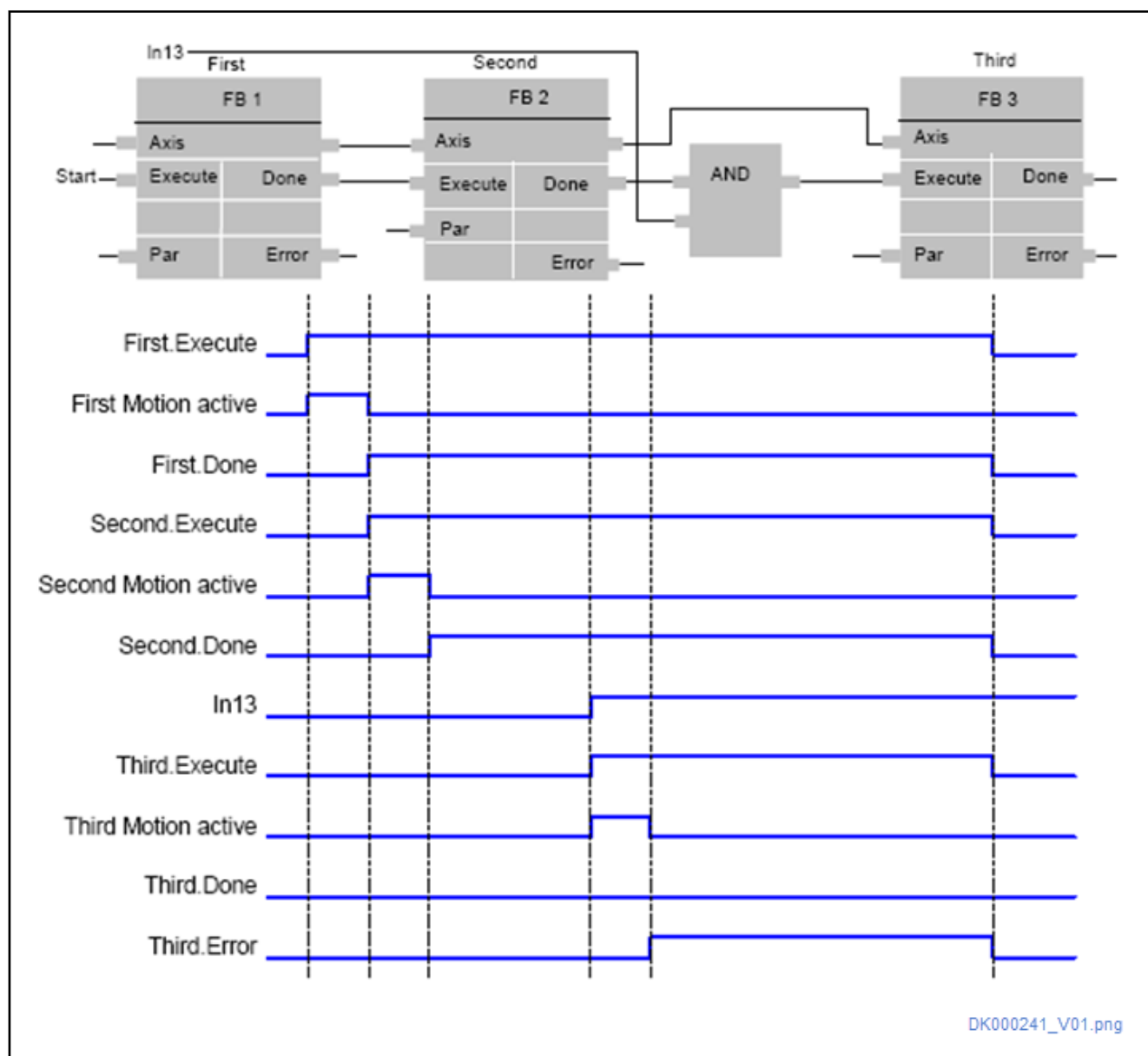


Abb. 4-9: Timingverhalten der Bausteine (Flankensteuerung)

Instanziierung

Bei der selben Instanz kann mit "Execute"-Flanken der Baustein mit geänderten Eingangswerten neu getriggert werden.

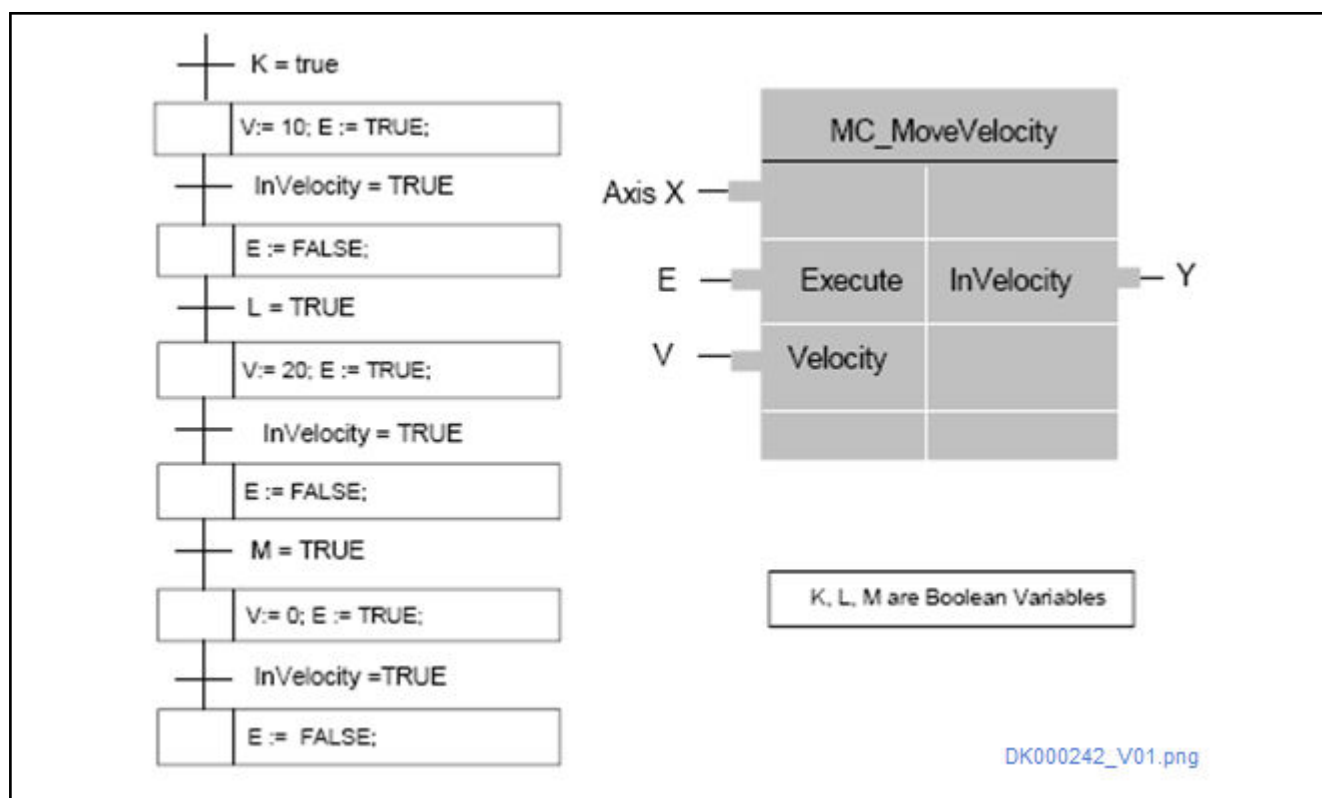


Abb. 4-10: Instanziierung von Bausteinen (selbe Instanz)

Nachfolgende Grafik zeigt die Instanziierung unterschiedlicher Instanzen. Jede Instanz hat somit einen eigenen Status, der über die Ausgänge "InVelocity" etc. sichtbar ist.

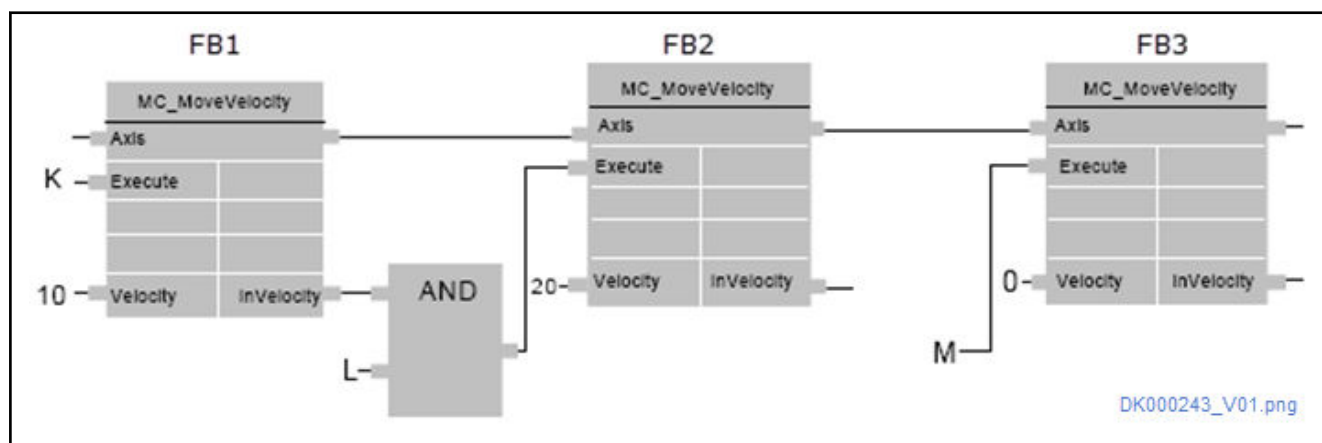


Abb. 4-11: Instanziierung von Bausteinen (unterschiedliche Instanzen)

4.4.4 Achsadressierung

Die Funktionsbausteine zur Motionkommandierung, sowie zum Parameterzugriff und der Diagnosebildung sind zum Großteil achsbezogene Funktionen. Die Zielachse muss somit dem Baustein als Information übergeben werden. PLCopen hat dazu einen Ein-/Ausgang mit dem Namen **Axis** (Datentyp "AXIS_REF") definiert. Neben Bausteinen zum diskreten Positionieren und Endlos-Verfahren existieren Motion-Bausteine, die einen Synchronlauf (Gleichlauf, Kurvensteuerung, Bewegungsprofil) einer Slaveachse auf eine Masterachse bewirken. Neben der Slaveachse, die in diesem Fall die Ziel-

Grundfunktionen von Rexroth IndraMotion MLD

achse dieser Motionbausteine darstellt, wird die Masterachse als weitere Achsreferenz an den Baustein übergeben.

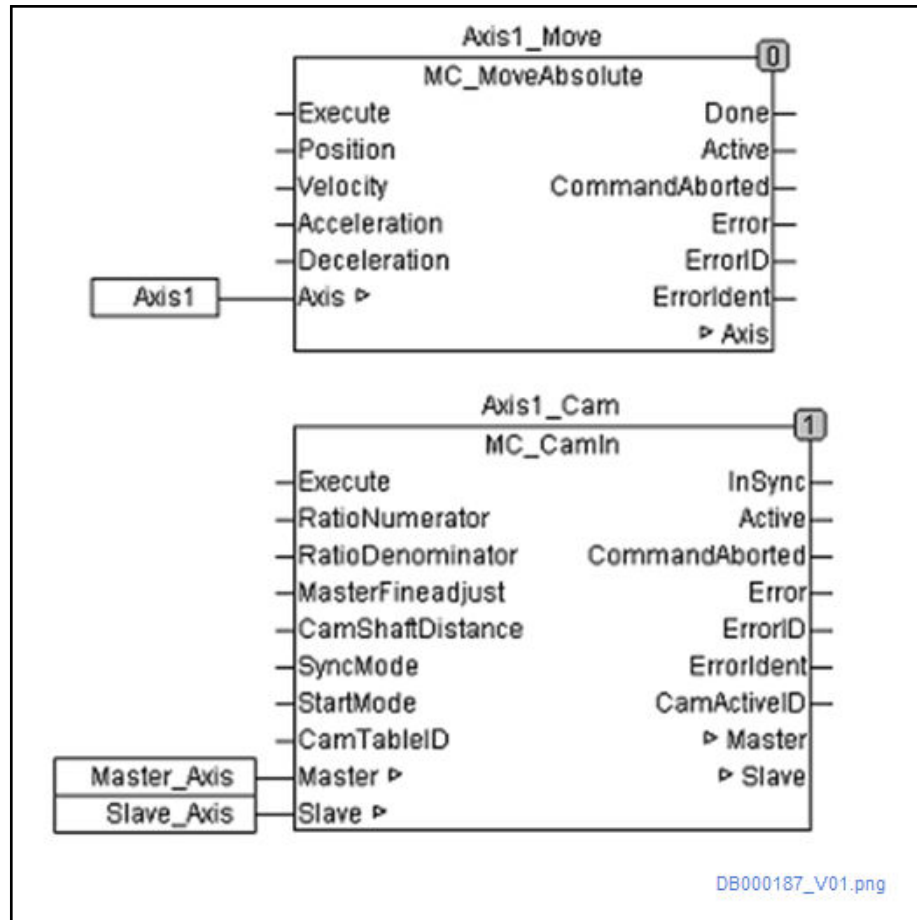


Abb. 4-12: Beispiele für die Achsreferenz an achsbezogenen Funktionsbausteinen

Nach PLCopen sind zu diesem Zweck Variablen des Struktur-Datentyps "AXIS_REF" als "VAR_IN_OUT" zu verwenden, der bei allen IndraMotion-Systemen aus den beiden Elementen "Steuerungsnummer" und "Achsnnummer" besteht. Als Steuerungsnummer ist ausschließlich die Konstante `LOCAL_CNTRL := 0` verwendbar. Als Achsnnummern sind in der Bibliothek "MX_CommonTypes" entsprechende Konstanten für reale Achsen, virtuelle Masterachsen und reale Masterachsen definiert.

Im MLD-System sind Variablen des Typs "AXIS_REF" für alle MLD-relevanten Achsen definiert. Die Variablen können direkt im Anwenderprogramm verwendet werden.

Reale Achsen zur Motionkommandierung / Parameterzugriff / Diagnosebildung

Variablen-Name	AXIS_REF Struktur-Elemente	Bedeutung	Vorhanden MLD-S / MLD-M	Kommandierung als: Ziel- / Slave- achse	AxisData vorhanden
Axis1	CntrlNo := LOCAL_CNTRL AxisNo := AXIS_1	Lokale reale Achse der MLD-S bzw. im CCD-Master von MLD-M Beispiel: Die reale Achse "Axis1" kann mit Hilfe des Bausteins "MC_MoveAbsolute" auf eine Absolutposition verfahren werden. "Axis1" muss am Eingang "Axis" des Bausteins "MC_MoveAbsolute" übergeben werden.	X / X	X	X
Axis2	CntrlNo := LOCAL_CNTRL AxisNo := AXIS_2	Reale Achse in einem CCD-Slave von MLD-M Beispiel: Im MLD-M-Systemmodus können die CCD-Slaveachsen von der SPS des CCD-Masters kommandiert werden. Um beispielsweise die Achse "Axis2" in der Betriebsart Geschwindigkeitsregelung zu verfahren, muss in der SPS des CCD-Masters der Funktionsbaustein "MC_MoveVelocity" aufgerufen und "Axis2" am Eingang "Axis" übergeben werden.	-- / X	X	X
...	...	...	...	...	...
Axis10	CntrlNo := LO- CAL_CNTRL AxisNo := AXIS_10	Reale Achse in einem CCD-Slave von MLD-M	-- / X	X	X

Achsen für Synchronbetriebsarten

Kommandierung des lokalen virtuellen Leitachsgenerators

Der lokale virtuelle Leitachsgenerator "VmAxis1" dient zur Bildung einer Leitachseposition, die als Eingangsvariable für die Lagesynchronisations-Betriebsarten und die Betriebsart "Geschwindigkeitssynchronisation" genutzt werden kann. Der lokale virtuelle Leitachsgenerator ist auf jeder realen Achse einmal vorhanden.



Bei welchen Funktionen und Funktionsbausteinen der lokale virtuelle Leitachsgenerator verfügbar ist, können Sie im Kapitel "[Achsenverfügbarkeit für Funktionen und Funktionsbausteine](#)" nachlesen.

Variablen-Name	AXIS_REF Struktur-Elemente	Bedeutung	Vorhanden MLD-S / M	AxisData vorhanden
VmAxis1	CntrlNo := LOCAL_CNTRL AxisNo := VMA_1	Achse zur Kommandierung des lokalen virtuellen Leit- achsgenerators	X / X	--

Der lokale virtuelle Leitachsgenerator kann sowohl bei MLD-S als auch bei MLD-M genutzt werden. Er kann nur auf der lokalen Achse angesteuert werden; d. h. auch bei MLD-M ist es nicht möglich den Leitachsgenerator auf einer entfernten CCD-Slaveachse zu kommandieren. Damit der virtuelle Leitachsgenerator angesteuert werden kann, müssen folgende Voraussetzungen gegeben sein:

1. Am Eingang "Axis" des jeweiligen Funktionsbausteins muss "VmAxis1" als Achsvariable übergeben werden.

Beispiel:

```
// Positionieren = MC_MoveAbsolute
fbMoAbs( Execute      := bLGExecute,
          Position     := rCmdPosition,
          Velocity     := rCmdVelocity,
          Acceleration := rCmdAcceleration,
          Deceleration := rCmdAcceleration,
          Axis         := VmAxis1);
```

2. Der Leitachsgenerator muss aktiviert werden (P-0-0917, Bit 0="1"):

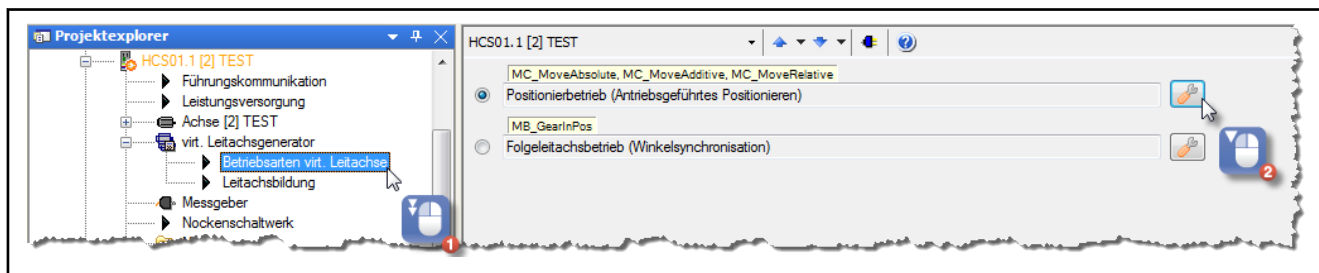


Abb. 4-13: Leitachsgenerator aktivieren (Dialogaufruf)

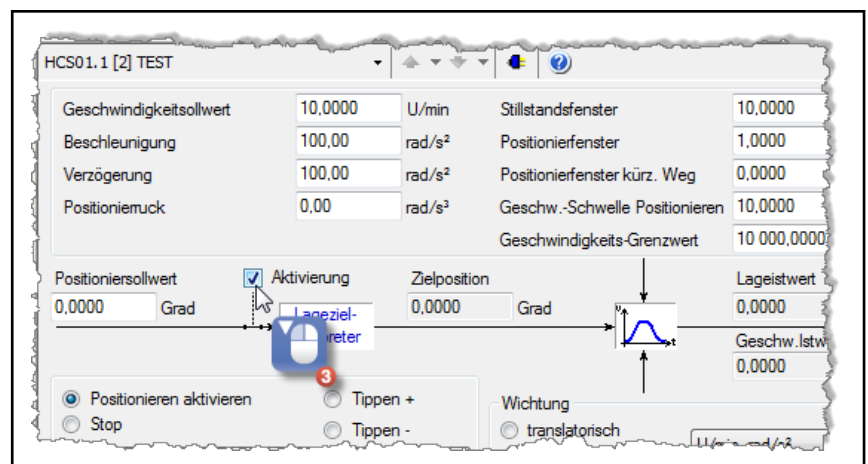


Abb. 4-14: Leitachsgenerator aktivieren

3. "P-0-0758, Virtuelle Leitachse, Lageistwert" muss in "P-0-0916, Leitachsformatwandler Signalauswahl" eingetragen werden:

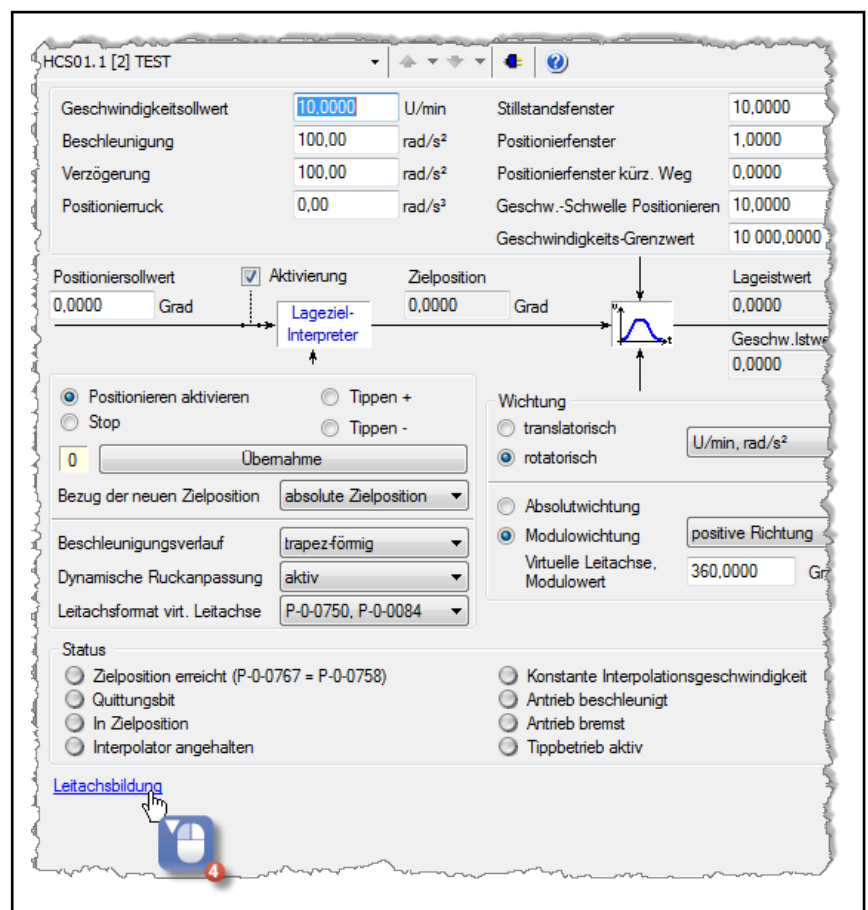


Abb. 4-15: Aufrufen der Leitachsbildung

Grundfunktionen von Rexroth IndraMotion MLD

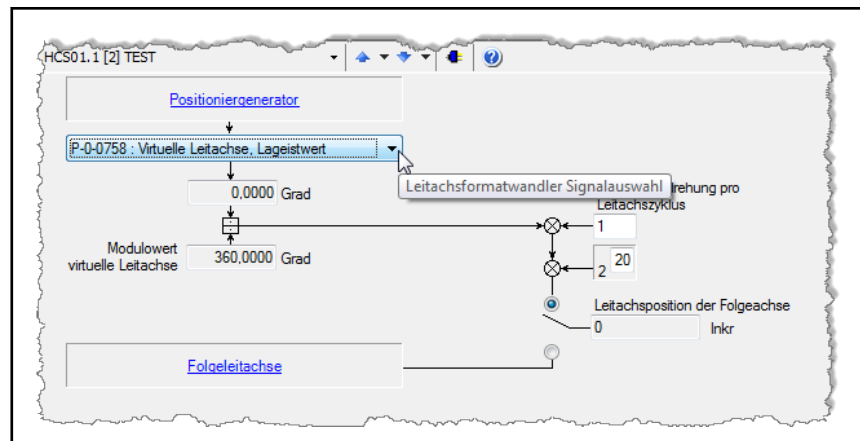


Abb. 4-16: Lageistwert der virtuellen Leitachse als Signal für den Leitachsformatwandler eintragen

Achsvariablen bei Synchronbetriebsarten

Eine Reihe der im MLD-System vordefinierten Variablen vom Typ "AXIS_REF" sind zur Verwendung mit Motion-Bausteinen vorgesehen, die einen Synchronlauf von einer Master- und einer Slaveachse bewirken. Aus der Namensgebung der Achsvariablen lassen sich bereits einige Informationen zur Verwendung der Achsvariablen ablesen:

- Achsen mit der Endung "Int" können nur als Masterachsen verwendet werden.
- Bei Verwendung von Achsen mit der Endung "Int" ist die Leitachseposition stets der lokal auf der Achse vorhandene Parameter. Beispielsweise ist bei Verwendung von "RmAxisInt" der lokal vorhandene P-0-0052 die Leitachseposition und nicht der P-0-0052 auf einer entfernten Achse im CCD-Verbund.
- Achsen mit Endung "1" bis "n" (z. B. "RmAxis1", "Axis3", ...):
 - MLD-M: Bei Verwendung dieser Achsen als Masterachse im MLD-M-Systemmodus müssen diese Achsen in "P-0-1820.0.1, CCD: Konfigurationsliste Leitachsen" als Leitachse konfiguriert werden.
Konfiguration über IndraWorks (Kontextmenü **Sercos (CCD)** ▶ **CCD: Applikationstyp Konfiguration**)

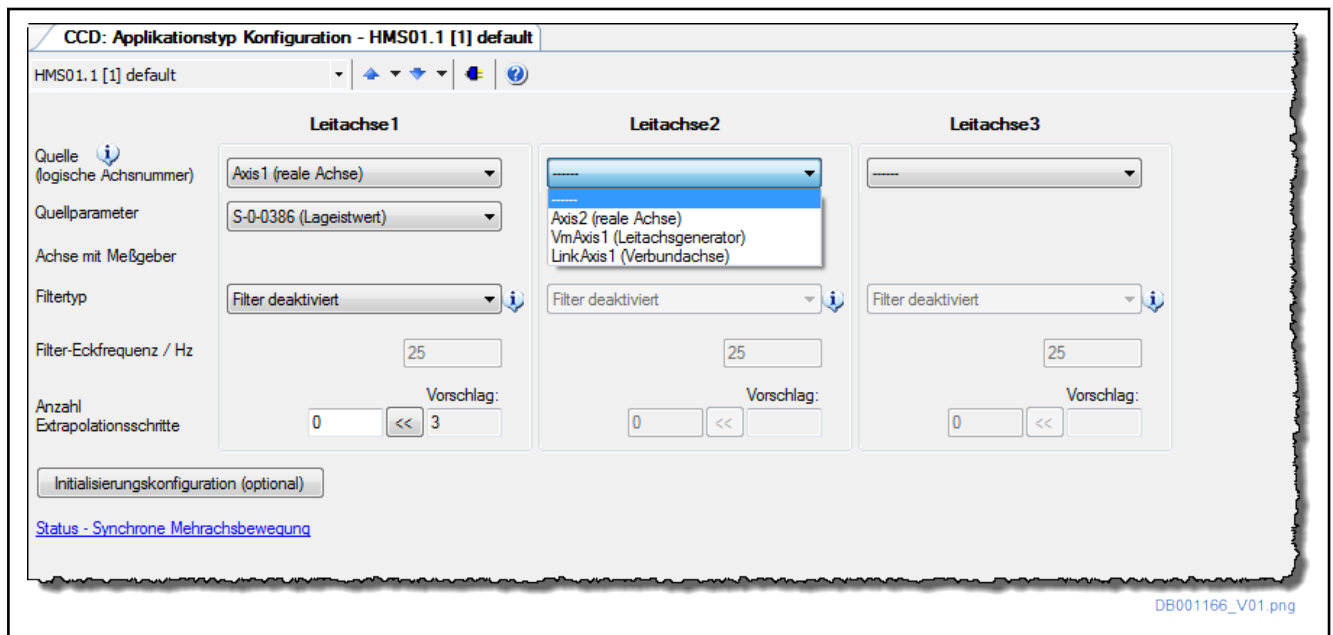


Abb. 4-17: Konfiguration der Leitachse(n)

- MLD-S bis MPx-20: Diese Achsen können bei MLD-S nicht als Masterachse verwendet werden.
- MLD-S ab MPx-20: Die Achsen "VmAxis1", "RmAxis1" und "LinkAxis1" können auch bei MLD-S als Masterachse verwendet werden (gleichbedeutend mit der entsprechenden Achse mit der Endung "Int").

Grundfunktionen von Rexroth IndraMotion MLD

Masterachsen

Variablen-Name	AXIS_REF Struktur-Elemente	Parameter (beinhaltet die Leitachseposition)	Bedeutung / Anwendungsbeispiel	Als Masterachse möglich MLD-S / MLD-M	AxisData vorhanden
VmAxis1	CntrlNo := LO-CAL_CNTRL AxisNo := VMA_1	"P-0-0761, Leitachseposition für Folgeachse"	VmAxis1 ist die virtuelle Leitachseposition im "P-0-0761, Leitachseposition für Folgeachse". <i>Beispiele:</i> - MLD-M Eine reale CCD-Slaveachse soll der virtuellen Leitachse im CCD-Master in einer Synchronbetriebsart folgen. In diesem Fall muss an den Bausteinen für die Synchronbetriebsarten (z. B. MB_GearInPos) die Achse VmAxis1 als Master und Axisn (mit n=2...10) als Slave übergeben werden. Zusätzlich muss im Parameter "P-0-1820.0.1, CCD: Konfigurationsliste Leitachsen" VMA_1 als Leitachse konfiguriert werden, damit "P-0-0761, Leitachseposition für Folgeachse" über CCD an die Folgeachse übertragen wird. - MLD-S - bis MPx-20: VmAxis1 kann auf einer MLD-S nicht als Masterachse für einen Synchronisationsbaustein verwendet werden. - ab MPx-20: Soll die reale Achse der MLD-S ihrem lokalen Leitachsgenerator folgen, dann kann an den Bausteinen für die Synchronbetriebsarten (z. B. MB_GearInPos) die Achse VmAxis1 als Master und Axis1 als Slave übergeben werden.	bis MPx-20: -- / X ab MPx-20: X / X	--
VmAxisInt	CntrlNo := LO-CAL_CNTRL AxisNo := VMA_INT	"P-0-0761, Leitachseposition für Folgeachse"	VmAxisInt ist die interne virtuelle Leitachse in jedem Antrieb, ausschließlich zur lokalen Verwendung. <i>Beispiele:</i> - MLD-M Soll eine reale CCD-Slaveachse ihrem lokalen Leitachsgenerator folgen, dann muss an den Bausteinen für die Synchronbetriebsarten (z. B. MB_GearInPos) die Achse VmAxisInt als Master und Axisn (n = 1...10) als Slave übergeben werden. - MLD-S bis MPx-20 Soll die reale Achse der MLD-S ihrem lokalen Leitachsgenerator folgen, dann muss an den Bausteinen für die Synchronbetriebsarten (z. B. MB_GearInPos) die Achse VmAxisInt als Master und Axis1 als Slave übergeben werden. (Ab MPx-20 kann VmAxis1 als Master verwendet werden.)	X / X	--

Variablen-Name	AXIS_REF Struktur-Elemente	Parameter (beinhaltet die Leitachseposition)	Bedeutung / Anwendungsbeispiel	Als Masterachse möglich MLD-S / MLD-M	AxisData vorhanden
RmAxis1	CntrlNo := LO-CAL_CNTRL AxisNo := RMA_1	"P-0-1820.0.3, CCD: Lageistwert Messgeber"	RmAxis1 ist eine aufbereitete Leitachseposition (Position eines Messgebers). Als Quelle der aufbereiteten Leitachse dient "P-0-0052, Lageistwert Messgeber" eines Messgebers, der an einer realen Achse angeschlossen ist. <i>Beispiele:</i> - MLD-M Die reale CCD-Masterachse soll der Messgeberposition im CCD-Slave Axis2 in einer Synchronbetriebsart folgen. In diesem Fall muss an den Bausteinen für die Synchronbetriebsarten (z. B. MB_GearInPos) die Achse RmAxis1 als Master und Axis1 als Slave übergeben werden. Zusätzlich muss im Parameter "P-0-1820.0.1, CCD: Konfigurationsliste Leitachsen" RMA_1 (= 23) als Leitachse und im "P-0-1820.0.2, CCD: Quelle des Messgeber-Lageistwerts" die logische Achsnummer der CCD-Slaveachse Axis2 konfiguriert werden, damit der "P-0-0052, Lageistwert Messgeber" der CCD-Slaveachse 2 über CCD an die CCD-Masterachse übertragen wird. - MLD-S - bis MPx-20: RmAxis1 kann auf einer MLD-S nicht als Masterachse für einen Synchronisationsbaustein verwendet werden. - ab MPx-20: Soll die reale Achse der MLD-S ihrem lokalen Messgeber folgen, dann kann an den Bausteinen für die Synchronbetriebsarten (z. B. MB_GearInPos) die Achse RmAxis1 als Master und Axis1 als Slave übergeben werden.	bis MPx20: -- / X ab MPx20: X / X	--
RmAxisInt	CntrlNo := LO-CAL_CNTRL AxisNo := RMA_INT	"P-0-0052, Lageistwert Messgeber"	RmAxisInt ist die lokale Messgeberposition in jedem Antrieb, ausschließlich zur lokalen Verwendung. <i>Beispiele:</i> - MLD-M Soll eine reale CCD-Slaveachse ihrem lokalen Messgeber folgen, dann muss an den Bausteinen für die Synchronbetriebsarten (z. B. MB_GearInPos) die Achse RmAxisInt als Master und Axisn (n=1...10) als Slave übergeben werden. - MLD-S bis MPx-20 Soll die reale Achse der MLD-S ihrem lokalen Messgeber folgen, dann muss an den Bausteinen für die Synchronbetriebsarten (z. B. MB_GearInPos) die Achse RmAxisInt als Master und Axis1 als Slave übergeben werden. (Ab MPx-20 kann RmAxis1 als Master verwendet werden.)	X / X	--

Grundfunktionen von Rexroth IndraMotion MLD

Variablen-Name	AXIS_REF Struktur-Elemente	Parameter (beinhaltet die Leitachseposition)	Bedeutung / Anwendungsbeispiel	Als Masterachse möglich MLD-S / MLD-M	AxisData vorhanden
LinkAxis1	CntrlNo := LO-CAL_CNTRL AxisNo := MA_LINK_1	"P-0-0053, Leitachseposition "	LinkAxis1 ist eine aufbereitete Leitachseposition (Position der Verbundachse). Als Quelle der aufbereiteten Leitachse dient der "P-0-0053, Leitachseposition". <i>Beispiele:</i> - MLD-M Die reale CCD-Slaveachse Axisn soll der Verbundachse ("P-0-0053, Leitachseposition" der CCD-Masterachse) in einer Synchronbetriebsart folgen. In diesem Fall muss an den Bausteinen für die Synchronbetriebsarten (z. B. MB_GearInPos) die Achse LinkAxis1 als Master und Axisn (n=2...10) als Slave übergeben werden. Zusätzlich muss im Parameter "P-0-1820.0.1, CCD: Konfigurationsliste Leitachsen" MA_LINK_1 (= 26) als Leitachse konfiguriert werden, damit der "P-0-0053, Leitachseposition" der CCD-Masterachse über den CCD-Verbund in den "P-0-0053, Leitachseposition" der CCD-Slaveachse übertragen wird. - MLD-S - bis MPx-20: LinkAxis1 kann auf einer MLD-S nicht als Masterachse für einen Synchronisationsbaustein verwendet werden. - ab MPx-20: Soll die reale Achse der MLD-S ihrer lokalen Verbundachseposition ("P-0-0053, Leitachseposition") folgen, dann kann an den Bausteinen für die Synchronbetriebsarten (z. B. MB_GearInPos) die Achse LinkAxis1 als Master und Axis1 als Slave übergeben werden.	bis MPx-20: -- / X ab MPx-20: X / X	--

Variablen-Name	AXIS_REF Struktur-Elemente	Parameter (beinhaltet die Leitachseposition)	Bedeutung / Anwendungsbeispiel	Als Masterachse möglich MLD-S / MLD-M	AxisData vorhanden
LinkAxisInt	CntrlNo := LO-CAL_CNTRL AxisNo := MA_LINK_INT	"P-0-0053, Leitachseposition" bei Slaveachse Axis1, "P-0-0787, Verbundachseposition-1" der Slaveachse bei Axis2...10 als Slaveachse	LinkAxisInt ist die lokale Verbundachseposition. Die Verbundachseposition wird von der übergeordneten Steuerung (z. B. MLD) über die Führungskommunikation zyklisch beschrieben. Abhängig von der Slaveachse muss die Verbundachseposition von der übergeordneten Steuerung in verschiedene Parameter geschrieben werden: - Wird Axis1 als Slaveachse an einen Synchronisationsbaustein übergeben (und LinkAxisInt als Masterachse), dann folgt Axis1 der Leitachseposition in "P-0-0053, Leitachseposition". D. h., die übergeordnete Steuerung muss die Leitachseposition in "P-0-0053, Leitachseposition" von Axis1 schreiben. - Wird Axis2 bis Axis10 als Slaveachse an einen Synchronisationsbaustein übergeben, dann folgen diese Achsen der Leitachseposition in ihrem jeweiligen "P-0-0787, Verbundachseposition-1". D. h., die übergeordnete Steuerung muss die Leitachseposition in den jeweiligen "P-0-0787, Verbundachseposition-1" von Axis2 bis Axis10 schreiben. <i>Beispiele:</i> - MLD-M Soll eine reale CCD-Slaveachse ihrer lokalen Verbundachseposition folgen, dann muss an den Bausteinen für die Synchronbetriebsarten (z. B. MB_GearInPos) die Achse LinkAxisInt als Master und Axisn (n=1...10) als Slave übergeben werden. - MLD-S bis MPx-20 Soll die reale Achse der MLD-S ihrer lokalen Verbundachseposition folgen, dann muss an den Bausteinen für die Synchronbetriebsarten (z. B. MB_GearInPos) die Achse LinkAxisInt als Master und Axis1 als Slave übergeben werden. (Ab MPx-20 kann LinkAxis1 als Master verwendet werden.)	X / X	--
Axisn (n=1...10)	CntrlNo := LO-CAL_CNTRL AxisNo := AXIS_2 ... AxisNo := AXIS_10	"P-0-1808.x.10, CCD: Aktiver Lageistwert" oder "P-0-1821.x.1, CCD: Lagesollwert für Leitachse n" (Auswahl in "P-0-1820.0.4, CCD: Auswahl Achseposition") Hinweis: P-0-1808.x.10 beinhaltet den "S-0-0386, Aktiver Lageistwert" der Achse x, P-0-1821.x.1 beinhaltet den "P-0-0434, Lagesollwert Regler" der Achse x.	"Axisn" sind die realen Achsen. "Axisn" kann nur mit MLD-M als Masterachse für einen Synchronisationsbaustein verwendet werden. "Axisn" wird bei MLD-S nicht als Masterachse für einen Synchronisationsbaustein unterstützt. <i>Beispiele:</i> - MLD-M Die reale CCD-Slaveachse Axis3 soll der realen CCD-Slaveachse Axis2 in einer Synchronbetriebsart folgen. In diesem Fall muss an den Bausteinen für die Synchronbetriebsarten (z. B. MB_GearInPos) die Achse Axis2 als Master und Axis3 als Slave übergeben werden. Zusätzlich muss im "P-0-1820.0.1, CCD: Konfigurationsliste Leitachsen" AXIS_2 als Leitachse konfiguriert werden, damit der "S-0-0386, Aktiver Lageistwert" oder wahlweise auch der P-0-0434, Lagesollwert Regler über CCD an die Folgeachse übertragen wird. - MLD-S Axisn kann auf einer MLD-S nicht als Masterachse für einen Synchronisationsbaustein verwendet werden.	-- / X	X

Tab. 4-2: Unterstützte Achsvariablen am Funktionsbaustein-Eingang "Master"

Grundfunktionen von Rexroth IndraMotion MLD

Slaveachsen

Variablen-Name	Bedeutung / Anwendungsbeispiel
VmAxis1	VmAxis1 ist die virtuelle Leitachseposition in "P-0-0761, Leitachseposition für Folgeachse". Wird VmAxis1 als Slaveachse an einem Synchronisationsbaustein (z. B. "MB_GearInPos") übergeben, dann befindet sich die lokale virtuelle Leitachse im Folgeleitachsbetrieb. Im Folgeleitachsbetrieb kann die lokale virtuelle Leitachse zwei verschiedenen Leitachsen folgen (reale Leitachse RmAxisInt="P-0-0052, Lageistwert Messgeber") oder externe virtuelle Leitachse LinkAxisInt="P-0-0053, Leitachseposition"). Im Folgeleitachsbetrieb wird nur die Betriebsart Winkelsynchronisation unterstützt. Daher kann VmAxis1 nur am Baustein MB_GearInPos als Slaveachse übergeben werden (nicht an den Bausteinen MC_GearIn, MB_MotionProfile und MC_CamIn). Der Folgeleitachsbetrieb kann sowohl bei MLD-S als auch bei MLD-M genutzt werden. Er kann nur auf der lokalen Achse aktiviert werden; d. h. auch bei MLD-M ist es nicht möglich, den Folgeleitachsbetrieb auf einer entfernten CCD-Slaveachse zu aktivieren. Beispiel: Folgeleitachsbetrieb mit MLD-S/MLD-M: Die lokale virtuelle Leitachse soll der realen Achse (Messgeber) folgen. In diesem Fall muss der Messgeber der MLD-S-Achse bzw. CCD-Masterachse aktiviert werden und am Funktionsbaustein MB_GearInPos die Achsvariable RmAxisInt als Masterachse und VmAxis1 als Slaveachse übergeben werden. Dann folgt die lokale virtuelle Leitachse ("P-0-0761, Leitachseposition für Folgeachse") dem lokalen Messgeber in der Betriebsart Winkelsynchronisation.
Axis1	Axis1 ist die reale lokale Achse. Axis1 existiert immer, sowohl bei MLD-S als auch bei MLD-M. <i>Beispiele:</i> - MLD-M Die CCD-Materachse (Axis1) soll in einer Synchronbetriebsart einer Masterachse folgen. Als Masterachse können folgende Achsen dienen: - Lokaler virtueller Leitachsgenerator ("P-0-0761, Leitachseposition für Folgeachse" von Axis1 = VmAxisInt oder VmAxis1, wenn VMA_1 als Leitachse im "P-0-1820.0.1, CCD: Konfigurationsliste Leitachsen" konfiguriert ist.) - Messgeber der lokalen Achse ("P-0-0052, Lageistwert Messgeber" = RmAxisInt) / einer entfernten Achse (P-0-0052 der entfernten Achse = RmAxis1), wenn RMA_1 als Leitachse im "P-0-1820.0.1, CCD: Konfigurationsliste Leitachsen" konfiguriert ist. (Parametrierung der Messgeber-Quellen-Achse über Parameter "P-0-1820.0.2, CCD: Quelle des Messgeber-Lageistwertes". P-0-1820.0.3 beinhaltet den Lageistwert des Messgebers.) - Verbundachse ("P-0-0053, Leitachseposition" = LinkAxisInt oder LinkAxis1, wenn MA_LINK_1 als Leitachse im "P-0-1820.0.1, CCD: Konfigurationsliste Leitachsen" konfiguriert ist.) - Reale Achse (CCD-Slaveachse, z.B. Axis 3, wenn AXIS_3 als Leitachse im "P-0-1820.0.1, CCD: Konfigurationsliste Leitachsen" konfiguriert ist) - MLD-S Die lokale Achse (Axis1) soll in einer Synchronbetriebsart einer Masterachse folgen. Als Masterachse können folgenden Achsen dienen: - Virtueller Leitachsgenerator ("P-0-0761, Leitachseposition für Folgeachse" = VmAxisInt bis MPx20, VmAxis1 ab MPx20) - Verbundachse ("P-0-0053, Leitachseposition" = LinkAxisInt bis MPx20, LinkAxis1 ab MPx20) - Messgeber ("P-0-0052, Lageistwert Messgeber" = RmAxisInt bis MPx20, RmAxis1 ab MPx20)

Variablen-Name	Bedeutung / Anwendungsbeispiel
Axis2...Axis10	Axis2 bis Axis10 sind die realen Achsen in einem CCD-Verbund. Axis2 bis Axis10 existieren nur mit CCD und MLD-M, nicht bei MLD-S. 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,...) Beispiel: Die CCD-Slaveachse Axis2 soll in einer Synchronbetriebsart einer Masterachse folgen. Als Masterachse können folgende Achsen dienen: • Virtueller Leitachsgenerator von Axis1 ("P-0-0761, Leitachseposition für Folgeachse" der CCD-Masterachse = VmAxis1). VMA_1 muss als Leitachse im P-0-1820.0.1 konfiguriert werden. • Virtueller lokaler Leitachsgenerator ("P-0-0761, Leitachseposition für Folgeachse" von Axis2 = VmAxisInt) • Messgeber der lokalen Achse ("P-0-0052, Lageistwert Messgeber" von Axis2 = RmAxisInt) / einer entfernten Achse (P-0-0052 der entfernten Achse = RmAxis1), wenn RMA_1 als Leitachse im "P-0-1820.0.1, CCD: Konfigurationsliste Leitachsen" konfiguriert ist. (Parametrierung der Messgeber-Quellen-Achse über Parameter "P-0-1820.0.2, CCD: Quelle des Messgeber-Lageistwertes". P-0-1820.0.3 beinhaltet den Lageistwert des Messgebers.) • Verbundachse ("P-0-0053, Leitachseposition" = LinkAxis1: Leitachseposition wird vom P-0-0053 der CCD-Masterachse übertragen, wenn MA_LINK_1 als Leitachse im "P-0-1820.0.1, CCD: Konfigurationsliste Leitachsen" konfiguriert ist / "P-0-0787, Verbundachseposition-1" = LinkAxisInt: Der P-0-0787 muss von der übergeordneten Steuerung geschrieben werden) • Reale Achse (CCD-Slaveachse, z.B. Axis 3, wenn AXIS_3 als Leitachse im "P-0-1820.0.1, CCD: Konfigurationsliste Leitachsen" konfiguriert ist)

Tab. 4-3: Unterstützte Achsvariablen am Funktionsbaustein-Eingang "Slave"

Achsverfügbarkeit für Funktionen und Funktionsbausteine

Eine genaue Übersicht aller vorhandenen Funktionsbausteine mit Achsreferenz-Eingängen und der Verfügbarkeit für die jeweilige Achse zeigt nachfolgende Tabelle:

Bezeichnung	Verfügbarkeit für Achse							
	Axis1 (lokale Achse)	"Axis2" bis "Axis10" (entfernte Achsen)	VmAxis1	VmAxisInt	RmAxis1	RmAxisInt	LinkAxis1	LinkAxisInt
MB_Command	X	X	--	--	--	--	--	--
MB_ChangeProfileSet	X	X	--	--	--	--	--	--
MB_ChangeProfileStep	X	X	--	--	--	--	--	--
MB_Home	X	X	--	--	--	--	--	--
MB_MotionProfile	MasterAxis SlaveAxis	MasterAxis SlaveAxis	MasterAxis*1	MasterAxis	MasterAxis*1	MasterAxis	MasterAxis*1	MasterAxis
MB_SetPositionControlMode	X	X	--	--	--	--	--	--
MB_GearInPos	MasterAxis SlaveAxis	MasterAxis SlaveAxis	MasterAxis*1 SlaveAxis	MasterAxis	MasterAxis*1	MasterAxis	MasterAxis*1	MasterAxis
MB_PhasingSlave	X	X	X ²	--	--	--	--	--
MB_PreSetMode	X	X	--	--	--	--	--	--
MB_Stop	X	X	X	--	--	--	--	--

Grundfunktionen von Rexroth IndraMotion MLD

Bezeichnung	Verfügbarkeit für Achse							
	Axis1 (lokale Achse)	"Axis2" bis "Axis10" (entfernte Achsen)	VmAxis1	VmAxisInt	RmAxis1	RmAxisInt	LinkAxis1	LinkAxisInt
MC_CamIn	MasterAxis SlaveAxis	MasterAxis SlaveAxis	MasterAxis*1	MasterAxis	MasterAxis*1	MasterAxis	MasterAxis*1	MasterAxis
MC_CamOut	X	X	X	--	--	--	--	--
MC_GearIn	MasterAxis SlaveAxis	MasterAxis SlaveAxis	MasterAxis*1	MasterAxis	MasterAxis*1	MasterAxis	MasterAxis*1	MasterAxis
MC_GearOut	X	X	--	--	--	--	--	--
MC_Reset	X	X	--	--	--	--	--	--
MC_Jog	X	X	X	--	--	--	--	--
MC_Phasing	X	X	--	--	--	--	--	--
MC_Power	X	X	--	--	--	--	--	--
MC_MoveAbsolute	X	X	X	--	--	--	--	--
MC_MoveAdditive	X	X	X	--	--	--	--	--
MC_MoveRelative	X	X	X	--	--	--	--	--
MC_MoveVelocity	X	X	X	--	--	--	--	--
MC_Stop	X	X	X	--	--	--	--	--
MC_TorqueControl	X	X	--	--	--	--	--	--
MX_MoveAbsolute	X	X	X	--	--	--	--	--
MX_MoveAdditive	X	X	X	--	--	--	--	--
MX_MoveRelative	X	X	X	--	--	--	--	--
MX_SetOpMode	X	X	--	--	--	--	--	--

*1

Mit MLD-S erst ab MPx-20 als Masterachse möglich

*2

Erst ab MPx-20 möglich

Tab. 4-4:

Verfügbare Motion-Funktionsbausteine

Bezeichnung	Verfügbarkeit für Achse							
	Axis1 (lokale Achse)	"Axis2" bis "Axis10" (entfernte Achsen)	VmAxis1	VmAxisInt	RmAxis1	RmAxisInt	LinkAxis1	LinkAxisInt
MX_Command	X	X	--	--	--	--	--	--
MX_SetControl	X	--	--	--	--	--	--	--
MX_SetDeviceMode	X	--	--	--	--	--	--	--
MX_SynchronControl	X	--	--	--	--	--	--	--
MX_fGetDriveWarning	X	--	--	--	--	--	--	--
MX_fSetDriveError	X	--	--	--	--	--	--	--
MX_fSetDriveWarning	X	--	--	--	--	--	--	--

Tab. 4-5:

Allgemeine Funktionsbausteine/Funktionen

Zuordnung der logischen MLD-M-Achsennummer zur Sercos Adresse der CCD-Slaves

Die Achsennummern 2 bis 10 (AxisNo=AXIS_2 bis AXIS_10) eines MLD-M-Systems sind reale Achsen in den CCD-Slaves. Die Achsennummer 1 repräsentiert die lokale Achse im CCD-Master. Die Programmierung im SPS-Anwenderprogramm erfolgt mittels der AXIS_REF-Variablen und der darin ent-

haltenen "AxisNo". Es existieren somit in einem MLD-M-System maximal 10 reale Achsen mit den logischen Nummern 1 bis 10.

Die Zuordnung der Sercos Adressen zu den logischen Achsnummern 1 bis 10 in der MLD-M erfolgt durch die Sercos Adresseinstellung für den CCD-Master und für die CCD-Slaves über die Reihenfolge der Sercos Adressen der entfernten Antriebe im Parameter "P-0-1601, CCD: Adressen projizierte Antriebe" bzw. im IndraWorks-Dialog "Cross Communication Drive : Einstellungen".

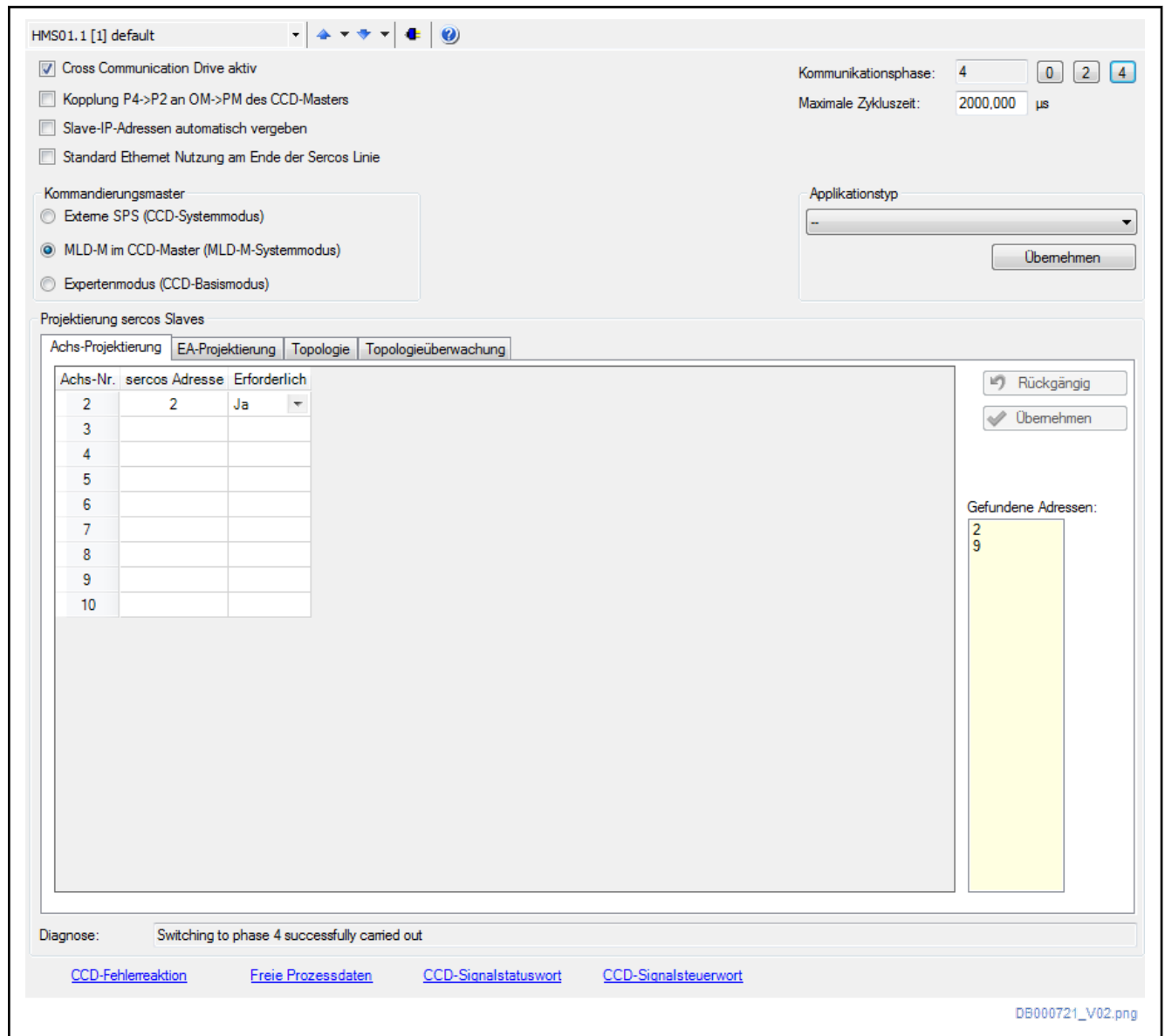


Abb. 4-18: Dialog "Cross Communication Drive : Einstellungen"

Die Gesamtkonstellation aller vorhandenen realen Achsen der MLD-M und deren Sercos Adressen ist im Parameter "P-0-4031, Geräteadressen-Übersicht" sichtbar.

Element-Nr. von P-0-4031	Inhalt Highwort = Sercos Adresse der Achse	Bedeutung
0	1	Sercos Adresse von Axis1 (1. logische MLD-Achse, der lokalen Achse (CCD-Master))
1	3	Sercos Adresse von Axis2 (2. logische MLD-Achse, einer Achse im CCD-Slave)

Tab. 4-6: Inhalt von "P-0-4031, Geräteadressen-Übersicht" für obige Beispiel
Siehe auch Firmware-Funktionsbeschreibung "Querkommunikation (CCD)"

4.4.5 Anwendungs- und Programmierhinweise

Allgemeines

Fallbeispiel A (Projekt mit Verwendung der Achsvariablen aus der Bibliothek "MX_Base")

Fallbeispiel B (Projekt mit anwenderdefinierten Achsvariablen)

Nachfolgend wird anhand von Fallbeispielen die Verwendung von Achsvariablen in Bezug auf PLCopen und Programmportierung erläutert:

```
instMX_MoveAbsolute1(Axis:= Axis3, ...);
instMX_MoveAbsolute2(Axis:= VmAxisInt, ...);
```

VAR

```
PortalX : AXIS_REF:= ( CntrlNo := LOCAL_CNTRL, AxisNo :=
AXIS_1 );
```

```
PortalY : AXIS_REF:= ( CntrlNo := LOCAL_CNTRL, AxisNo :=
AXIS_4 );
```

```
Zustellachse : AXIS_REF:= ( CntrlNo := LOCAL_CNTRL, Axis-
No := AXIS_2 );
```

```
Werkzeug : AXIS_REF:= ( CntrlNo := LOCAL_CNTRL, AxisNo :=
AXIS_3 );
```

```
Leitachse : AXIS_REF:= ( CntrlNo := LOCAL_CNTRL, Axis-
No := VMA_EXT);
```

END_VAR

Damit erfolgt die Adressierung wie folgt:

```
instMX_MoveAbsolute(Axis:= Zustellachse, ...);
```

- oder -

```
instMX_MoveAbsolute(Axis:= Leitachse, ...);
```

MLD-Funktionsbausteine und Parameter zur Achskommandierung

Funktionen und Funktionsbausteine :

- "MX_SetControl": Steuerung der temporären Kontrolle über die lokale Achse bei Kontrolle der lokalen Achse über die Führungskommunikation
- "MX_SetOpMode": Direkte Aktivierung einer Sercos Betriebsart
- "MC_Power", "MB_PresetMode": Aktivierung der Achsreglerfreigabe und evtl. Vorwahl der Betriebsart
- "MC_Stop", "MB_Stop": Anhalten des Antriebs
- "MX_Command"/"MB_Command": Aktivierung eines Antriebskommandos
- "MC_Home": Aktivierung "Antriebsgeführtes Referenzieren"
- "MC_MoveVelocity": Betriebsart "Geschwindigkeitsvorgabe"

- "MC_MoveAbsolute"/"MX_MoveAbsolute", "MC_MoveRelative"/"MX_MoveRelative", "MC_MoveAdditive"/"MX_MoveAdditive": Betriebsart "Positionieren"
- "MC_Jog": Betriebsart "Positionieren (Tippen +/-)"
- "MC_GearIn", "MC_GearOut": Betriebsart "Geschwindigkeitssynchronisation"
- "MB_GearInPos", "MC_CamIn", "MC_CamOut": Betriebsart "Lagesynchronisation"
- "MB_Phasing", "MB_PhasingSlave": additiver Lageoffset bei Lagesynchronisation
- "MB_MotionProfile": Betriebsart "Lagesynchronisation" mit Bewegungsprofilen

Parameter und Diagnosen zur Achskommandierung

- "P-0-1367, SPS Konfiguration", Bit 4: Aktivierung "permanente Kontrolle" der MLD auf die lokale Achse
- "P-0-1800.0.1, CCD: Konfiguration", Bits 4 und 3: Aktivierung MLD-M-Systemmodus
- Parameter der verwendeten Betriebsarten "Geschwindigkeitsregelung", "Momentenregelung", "Lagesynchronisation", "Antriebsgeführtes Positionieren" und "Geschwindigkeitssynchronisation"
- Parameter der "internen Betriebsarten" bei temporärer Kontrolle für den Betriebsmodus als intelligente Servoachse:
 - P-0-1450, SPS/Einrichtbetrieb Positionier-Sollwert
 - P-0-1451, SPS/Einrichtbetrieb Positionier-Geschwindigkeit
 - P-0-1452, SPS/Einrichtbetrieb Positionier-Beschleunigung
 - P-0-1453, SPS/Einrichtbetrieb Positionier-Verzögerung
 - P-0-1454, SPS/Einrichtbetrieb Positioniersollwert-Übernahme
 - P-0-1455, SPS/Einrichtbetrieb Positioniersollwert-Quittung
 - P-0-1460, SPS/Einrichtbetrieb Geschwindigkeits-Sollwert
 - P-0-1461, SPS/Einrichtbetrieb Steigung Rampe
 - P-0-1463, SPS/Einrichtbetrieb Verzögerungs-Rampe
 - P-0-1465, SPS/Einrichtbetrieb Drehmoment-/Kraft-Sollwert

Arbeiten mit Retain-Variablen

Retain-Variablen werden mit dem Schlüsselwort "RETAIN" gekennzeichnet. Diese Variablen behalten ihren Wert nach unkontrolliertem Beenden wie auch nach normalem Aus- und Einschalten der Steuerung (entsprechend Kommando "Online" "Reset"). Bei erneutem Start des Programms wird mit den gespeicherten Werten weitergearbeitet. Ein Anwendungsbeispiel ist ein Stückzähler in einer Fertigungsanlage, der nach einem Stromausfall weiterzählen soll. Alle anderen Variablen werden in diesem Fall neu initialisiert, entweder mit ihren initialisierten Werten oder mit den Standardinitialisierungen. Retain-Variablen werden bei "Reset Kalt", "Reset Ursprung" und - im Gegensatz zu persistenten Variablen - bei einem erneuten Programm-Download neu initialisiert.

Je nach Hardware (Standard- oder Advanced-Bedienteil mit Speicherkarte) stehen 472 Byte bzw. 31704 Byte Speicher für **Retain-Variablen** und **persistente Variablen** zur Verfügung.

Die Retaindaten lassen sich am Block über P-0-1359 auslesen oder beschreiben. Die Listenlänge (MaxLen) des P-0-1359 richtet sich nach der Größe des Retain-Speichers (126 oder 7934 Elemente à 4 Byte).



Das Schreiben darf nicht bei laufendem Programm erfolgen. Wenn P-0-1359 geschrieben wird, müssen die Daten vom gleichen Programm stammen (d. h. es dürfen keine Änderungen an den Retaindaten und an den persistenten Daten vorgenommen worden sein).

Beim Booten wird der Retainspeicher in das RAM übertragen und beim "Power-Fail IRQ" werden die Daten aus dem RAM in den Retainspeicher kopiert.

4.5 Tasksystem

4.5.1 Begriffserklärungen

- **Task:** Eine Task ist ein in sich geschlossener, elementarer Steuerfluss innerhalb eines Prozesses. Jede Task besitzt seinen eigenen Zustand mit Programmzähler, Stapelspeicher usw.
- **Multitasking** (Mehrprozessbetrieb): Multitasking bezeichnet die Fähigkeit, mehrere Aufgaben (Tasks) nebenläufig auszuführen. Dabei werden die verschiedenen Prozesse in so kurzen Abständen immer abwechselnd aktiviert, dass der Eindruck der Gleichzeitigkeit entsteht.
- **Scheduler:** Ein Scheduler (Planer) ist ein wichtiges Element eines Multitasking-Betriebssystems. Er bestimmt nach einer vorgegebenen Strategie die Reihenfolge, in der Tasks, die bereit sind, abgearbeitet werden. Als Scheduling bezeichnet man entsprechende Strategien bzw. Verfahren, mit denen die Abarbeitungsreihenfolge der Task realisiert wird.

4.5.2 Kurzbeschreibung

- Unterstützte Tasktypen** Rexroth IndraMotion MLD unterstützt bis zu vier IEC61131-Anwendertasks der folgende Tasktypen:
- Periodische Task ("zyklisch" ab 1 ms)
 - Freilaufende Task
 - Ereignisgesteuerte Task (Event-Task) (Triggerung im 1 ms-Raster)
 - Extern ereignisgesteuerte Task (externe Event-Task); zwei Systemevents wurden für diesen Tasktyp bereitgestellt:
 - Event "FKM_SYNCHRONIZED_TASK": FKM-synchrone Motion-task
 - Event "CCD_SYNCHRONIZED_TASK": CCD-synchrone Motion-task (nur bei MPC)

Abb. 4-19: (Task-)Konfiguration



Rexroth IndraMotion MLD unterstützt Systemevents.

Tasksystem und Prioritäten

Das interne Timing bzw. die Abfolge der Abarbeitung der einzelnen Tasks wird über das so genannte Tasksystem geregelt:

- Es wird ein preemptives Multitasking unterstützt. Damit können niederpriorere Tasks von höherprioreren Tasks zu jeder Zeit unterbrochen werden.
- Es können maximal vier Tasks gleichzeitig "aktiv" sein.
- Es gibt vier Prioritätsebenen [Priorität 0...3 (0=hochprior)].
 - Die verwendeten Tasks müssen unterschiedliche Priorität haben, damit eine definierte Abarbeitung offensichtlich ist.
 - Hochpriorere SPS-Tasks unterbrechen niederpriorere SPS-Tasks auch während der Bearbeitung (Ausnahmen: Tasks sind jeweils wenige µsec für Übertragung des Prozessabbilds oder Motionkommandierung nicht unterbrechbar).
 - Wenn eine niederpriorere Task gestartet wird, läuft sie ggf. verzögert los.
- Die IEC-Tasks sind keiner Achse oder Aufgabe fest zugeordnet. Somit gibt es keine dedizierte Motions task. Für die unterschiedlichen Aufgaben können die vorhandenen Konfigurationsmöglichkeiten genutzt werden.
- Wenn bei der Taskkonfiguration ein Taskintervall zu groß angegeben wird, wird eine Fehlermeldung ausgegeben.
- Generell sollten so wenige Tasks wie möglich angelegt werden, damit die Programmabläufe übersichtlich bleiben.



Es können bis zu vier unterschiedliche Tasks angelegt werden. (Anlegen von Tasks: siehe Dokumentation "IndraLogic 2G, SPS-Programmiersystem (R911336875)".)

4.5.3 Task-Eigenschaften

Periodische (zyklische) Tasks Periodische ("zyklische") Tasks, zeichnen sich durch die folgenden Merkmale aus:

- Konstante Zykluszeit
- Die zyklische Task wird genau einmal pro angegebener Periode gestartet, d.h. in jedem Zeitintervall wird der Code genau einmal durchlaufen, insofern der vorherige Durchlauf beendet wurde (bei einem Zeitüberlauf und aktiviertem Watchdog wird die SPS mit einem Fehlerstatus angehalten "Exception").



Verbraucht eine periodische Task die zugeteilte Rechenzeit und ist immer noch aktiv, wenn sie bereits neu laufen sollte, so wird sie erst wieder in der darauf folgenden Periode gestartet. Es wird also nicht versucht den ausgefallenen Zyklus "nachzuholen".

Freilaufende Task Eine freilaufende Task zeichnet sich durch folgende Merkmale aus:

- Eine freilaufende Task läuft ständig.
- Es darf nur eine freilaufende Task geben und sie muss mit niedrigster Priorität angelegt werden.
- Sofort nach Taskende wird eine freilaufende Task wieder gestartet.

Ereignisgesteuerte Tasks Ereignisgesteuerte Tasks zeichnen sich durch folgende Merkmale aus:

- Ereignisgesteuerte Tasks werden durch Setzen einer globalen booleschen SPS-Variable gestartet. Die Startbedingungen werden periodisch geprüft, wobei die Reaktionszeiten durch die minimal mögliche SPS-Zykluszeit definiert sind (siehe "[Leistungs- und Performance-Angaben](#)").
- Eine ereignisgesteuerte Task (Eventtask) wird nach Auftreten des Events (Flanke) genau einmal durchlaufen.

Extern ereignisgesteuerte Task Eine extern ereignisgesteuerte Task funktioniert im Prinzip wie eine ereignisgesteuerte Task; der Unterschied besteht in der Aktivierung der Task. Eine extern ereignisgesteuerte Task wird über ein Systemevent aktiviert. Ein Systemevent kann vom Anwender nicht beeinflusst werden.

Folgende zwei Systemevents werden unterstützt:

- FKM_SYNCHRONIZED_TASK
- CCD_SYNCHRONIZED_TASK

Mit Hilfe der extern ereignisgesteuerten Task kann eine synchronisierte Motiontask auf den NC-Zyklus bzw. eine synchronisierte Motiontask auf den CCD-Zyklus realisiert werden.

4.5.4 Motiontask

Allgemeines

Eine Motiontask ist eine zum NC- oder CCD-Zyklus synchrone Task, in der in der Regel die Motion-Funktionsbausteine und Technologiefunktionen verarbeitet werden.

Die Echtzeitdaten der AxisData werden am Anfang und am Ende der Motiontask gerechnet.

FKM-synchrone Motiontask

	Die FKM-synchrone Motiontask ist eine zum NC-Zyklus synchrone Task. Sie wird über eine Task vom Typ "extern ereignisgesteuert" mit dem Systemevent "FKM_SYNCHRONIZED_TASK" angelegt.
Motiontask-Zykluszeit	Die FKM-synchrone Motiontask läuft synchron zum NC-Zyklus. Daraus ergibt sich eine Motiontask-Zykluszeit, die der NC-Zykluszeit entspricht. Die NC-Zykluszeit kann (außer bei Führungskommunikation Sercos III) dem Parameter "S-0-0001, NC-Zykluszeit (TNcyc)" entnommen werden. Bei Führungskommunikation Sercos III entspricht die NC-Zykluszeit der "Producer-Zykluszeit" (S-0-1050.x.10) von der Verbindung, die vom Master im MDT-Telegramm konfiguriert wurde.
	 Die NC-Zykluszeit wird nur bei der Führungskommunikation (FKM) Sercos III vom übergeordneten FKM-Master vorgegeben. Bei allen anderen Führungskommunikationen kann die NC-Zykluszeit über den Parameter "S-0-0001, NC-Zykluszeit (TNcyc)" frei eingestellt werden.
	Bei Festlegung der NC-Zykluszeit in Verbindung mit der Verwendung der FKM-synchronen Motiontask ist folgendes zu beachten: Minimal erlaubte NC-Zykluszeit ist 1 ms. Alle ganzzahligen Vielfachen der minimal erlaubten NC-Zykluszeit sind ebenfalls erlaubt.
Einsatzgebiet	Der Aufrufzeitpunkt der MLD-S soll bei physikalisch getrennten Antrieben synchronisiert werden, damit die jeweilige MLD-S auf Soll- und Istwerte zum gleichen Zeitpunkt reagieren kann. Zu diesem Zweck wird die Task auf den NC-Zyklus synchronisiert.
Beispiel	Eine übergeordnete Steuerung beschreibt zyklisch ein globales SPS-Register mit Leitachspunkten. Die MLD wertet die Leitachspunkte aus und addiert einen Leitachsoffset dazu. Das Ergebnis wird in den Parameter "P-0-0053, Leitachspunkt" geschrieben. Bei dem nächsten Eintreffen des MDT (Master Data Telegram) wird der Wert von P-0-0053 vom Antrieb übernommen. Der Sollwert von der NC wird somit offsetbehaftet einen NC-Takt später wirksam.
Timing	Die FKM-synchrone Motiontask beginnt bei der MLD nach der MDT- und nach der AT-Verarbeitung und endet wieder vor diesen (MDT: Master Data Telegram; AT: Antriebs-Telegramm).

CCD-synchrone Motiontask

Die CCD-synchrone Motiontask ist eine zum CCD-Zyklus (MS-Kanal) synchrone Task.

Sie wird über eine Task vom Typ "extern ereignisgesteuert" mit dem Systemevent "CCD_SYNCHRONIZED_TASK" angelegt.

	Handelt es sich um eine CCD-Masterachse und ist CCD aktiv, ist die CCD-synchrone Motiontask nur im MLD-M-Systemmodus erlaubt. Ist CCD nicht aktiv oder es handelt sich nicht um eine CCD-Masterachse, wird eine im SPS-Projekt konfigurierte CCD-synchrone Motiontask wie eine FKM-synchrone Motiontask behandelt. In diesen Fall gilt die Beschreibung für die FKM-synchrone Motiontask.

Motiontask-Zykluszeit (CCD-Zykluszeit)

Die CCD-Zykluszeit (und somit die Motiontask-Zykluszeit) entspricht bei Führungskommunikation Sercos III der "Producer-Zykluszeit" (S-0-1050.x.10) von der Verbindung, die vom Master im MDT-Telegramm konfiguriert ist, wenn diese Zeit kleiner als der Wert von "P-0-1800.0.10, CCD: Zykluszeit" ist. Bei allen anderen Führungskommunikationen entspricht die Motiontask-Zykluszeit sowie die CCD-Zykluszeit der NC-Zykluszeit [S-0-0001, NC-Zykluszeit (TNcyc)], wenn die NC-Zykluszeit kleiner als der Wert von "P-0-1800.0.10, CCD: Zykluszeit" ist. Wenn nicht, entsprechen sie dem Wert von "P-0-1800.0.10, CCD: Zykluszeit". Die aktive CCD-Zykluszeit kann dem Parameter "P-0-1810.0.3, CCD: Timing-Einstellungen" entnommen werden.

Bei der Festlegung der CCD-Zykluszeit in Verbindung mit der Verwendung der CCD-synchronen Motiontask ist folgendes zu beachten:

- Minimal erlaubte CCD-Zykluszeit ist 1ms. Alle ganzzahligen Vielfachen der minimal erlaubten CCD-Zykluszeit sind für die CCD-synchrone Motiontask ebenfalls erlaubt.



Zurzeit werden von CCD im MLD-M-Systemmodus CCD-Zykluszeiten von 1 ms, 2 ms und 4 ms unterstützt.

Einsatzgebiet

Der Aufrufzeitpunkt einer periodischen SPS-Task (Motiontask) in der CCD-Masterachse soll an den CCD-Zyklus (MS-Kanal) gekoppelt werden. Dazu muss in der CCD-Masterachse der MLD-M-Systemmodus und "permanente Kontrolle der MLD" aktiviert werden.

Die CCD-synchrone Motiontask kommt nur in Verbindung mit dem MLD-M-Systemmodus zum Einsatz. Im MLD-M-Systemmodus steuert die MLD-M der CCD-Masterachse die Masterachse (lokaler Slave) selbst, sowie die entfernten Achsen. Dazu ist es oft notwendig Istwerte von allen Achsen zu verarbeiten, welche zum gleichen Zeitpunkt gelatcht wurden und Sollwerte zu schreiben, die in allen Achsen zum gleichen Zeitpunkt wirksam werden. Dieser konsistente Datenaustausch wird über die CCD-synchrone Motiontask in Verbindung mit der Achsdatenstruktur "AxisData" erreicht.

Beispiel

Mit Hilfe der MLD-M soll ein Fräskopf dreidimensional bewegt werden. Um ein einwandfreies Fräsergebnis zu bekommen, muss der Steuerung (MLD-M) von allen Achsen, welche an der Bewegung teilnehmen, die Lageistwertposition bezogen auf den gleichen Zeitpunkt (T4) bekannt sein, um daraus die entsprechenden Sollwerte berechnen zu können. Diese berechneten Sollwerte müssen dann zur gleichen Zeit an alle Achsen übertragen werden, damit sie zum gleichen Zeitpunkt wirksam werden. Um dies zu gewährleisten, wird eine zum CCD-Takt synchrone Task benötigt. Die Echtzeitdaten werden über die Achsdatenstruktur "AxisData" ausgetauscht.

Timing

Die CCD-synchrone Motiontask läuft bei der MLD zwischen der CCD-AT-Verarbeitung und der CCD-MDT-Verarbeitung.

Die Echtzeitdaten der Achsdatenstruktur "AxisData" werden vor Taskbeginn und nach dem Taskende aktualisiert. Die Istwerte werden nach der CCD-AT-Verarbeitung gelesen und die Sollwerte werden vor der CCD-MDT-Verarbeitung geschrieben. Somit kann innerhalb der CCD-synchronen Motiontask über die Achsdatenstruktur "AxisData" immer auf konsistente Daten zugegriffen bzw. geschrieben werden. Die über die Achsdatenstruktur geschriebenen Werte werden in den Achsen zum gleichen Zeitpunkt wirksam.



Es darf immer nur eine Motiontask vom Typ "extern ereignisgesteuert" verwendet werden.

Diagnosen und Fehlermeldungen im Zusammenhang mit einer Moti- ontask



Beide Motiontask-Arten laufen immer. Synchronisiert sind sie aber erst ab dem Zeitpunkt an dem zyklische Daten über die Führungskommunikation bzw. über CCD ausgetauscht werden.

- **SPS-Fehler**
In P-0-1365 und in IndraWorks (Registerkarte "Log" des Geräte-Editors) wird u. A. angezeigt, wenn das Laden des Bootprojekts fehlerhaft war.
- **Antriebsfehler: F6010 SPS Laufzeitfehler**
Der Fehler F6010 wird unter anderem dann ausgelöst, wenn beim Laden des Bootprojekts ein Fehler auftritt. Mögliche Fehler in Verbindung mit einer Motiontask sind die eben beschriebenen SPS-Fehler (siehe "F6010" in der Dokumentation "Hinweise zur Störungsbeseitigung").
- **Umschaltkommandofehler**
 - C0241 Parametrierung Motiontask fehlerhaft
 - C0266 fehlerhafte CCD-Phasenumschaltung

Mögliche Ursachen und Abhilfen für den Umschaltkommandofehler C0266 entnehmen Sie bitte der Dokumentation "Hinweise zur Störungsbeseitigung".

In Verbindung mit einer CCD-synchronen Motiontask tritt der Fehler dann auf, wenn festgestellt wird, dass die CCD-synchrone Motiontask mit der eingestellten CCD-Zykluszeit (P-0-1810.0.3, CCD: Timing-Einstellungen) nicht betrieben werden kann. Dem Parameter "P-0-1810.0.10, CCD: Diagnose-Text" kann bei Auftreten des Umschaltkommandofehlers C0266 die aufgeschlüsselte Fehlermeldung entnommen werden.

Freilaufende Task

Eine weitere Motiontask-Variante ist die freilaufende Task:

- Eine freilaufende Task läuft ständig.
- Es darf nur eine freilaufende Task geben und sie muss mit niedrigster Priorität angelegt werden.
- Sofort nach Taskende wird eine freilaufende Task wieder gestartet.



Damit der Anwender geeignete Taskkonzepte erstellen kann, liefert Bosch Rexroth Beispielkonfigurationen für die unterschiedlichen Ausprägungen.

4.5.5 Task-Überwachung (Watchdog)

Generell kann die Überwachung jeder Task individuell eingestellt werden. Damit ist es möglich z.B. den Watchdog einer Task auf ein Vielfaches der Periodendauer zu setzen. Dies ist insbesondere für Motiontasks sinnvoll, da diese schnell reagieren sollen, aber teilweise auch rechenintensive Aufgaben wahrnehmen müssen.

Verwendung von Netzwerkvariablen

Wenn Netzwerkvariablen verwendet werden, wird die Task mit zusätzlicher Laufzeit belastet, die von der gesamten Kommunikation über die Ethernet-Schnittstelle abhängt. Je nach Anzahl der Netzwerkkanäle und sonstiger Kommunikation kann die Laufzeit um einige Millisekunden ansteigen.

⇒ Schnell laufende Tasks mit kurz eingestelltem Watchdog sind nicht für die Übertragung von Netzwerkvariablen geeignet. Wir empfehlen Netzwerkvariablen in langsamen oder eigenen Tasks mit tolerant eingestelltem Watchdog einzusetzen.

Watchdog-Funktion Bei Rexroth IndraDrive wird folgende Watchdog-Funktion unterstützt:

- Zeit
- Empfindlichkeit (Anzahl der Überschreitungen bis zum Auslösen)
- *Reaktion:*
 - Es wird eine SPS-Exception erzeugt.
 - Alle Ausgänge werden auf "0" gesetzt und die SPS geht in STOP.
 - Auf dem Antriebsregelgerät wird der Fehler "F6010" erzeugt; dies führt bei den Achsen zu einer bestmöglichen Stillsetzung.



Die SPS muss nach Auftreten eines Watchdogs mit RESET (vgl. P-0-1350) zurückgesetzt werden.

Watchdog konfigurieren

Für jede Task kann eine Zeitüberwachung (Watchdog) konfiguriert werden.

Der Watchdog ist aktiv, wenn die Option aktiviert ist. Die Task wird dann mit Fehlerstatus ("Ausnahmefehler") beendet, sobald die gesetzte Watchdog-Zeit überschritten wird, wobei die angegebene Empfindlichkeit mit eingerechnet wird. Ist in den "SPS-Einstellungen" die Option **Im Stop-Status E/A aktualisieren** aktiviert, so werden die Ausgänge auf die definierten Standardwerte zurückgesetzt. Die folgenden Fälle sind möglich:

Mehrere aufeinanderfolgende Zeitüberschreitungen; folgendes gilt:

Empfindlichkeit	Ausnahmefehler in Zyklus...
0, 1, 2	1
3	2
...	...
n	n-1

Einmalige Zeitüberschreitung:

Ausnahmefehler, wenn die Zykluszeit des aktuellen Zyklus länger als "Zeit × Empfindlichkeit" ist.

Beispiel:

Zeit="t#10ms", Empfindlichkeit="5"

⇒ Ausnahmefehler, sobald die Task (einmalig) länger als 50 ms läuft.

Beachten Sie, dass ein Watchdog mit Hilfe der Bibliotheksfunktionen "CmplecTask.library" für bestimmte SPS-Zyklen abgeschaltet werden kann; das ist für Zyklen nützlich, die aufgrund von Initialisierungen mehr Zeit beanspruchen können.

Um einen Watchdog ab-/einschalten zu können, gehen Sie wie folgt vor:

1. Deklarieren Sie eine geeignete Variable für das Taskhandle (vom Typ `RTS_IEC_HANDLE`):

Programm:

```
hIecTask : RTS_IEC_HANDLE;
```

2. Nutzen Sie die Schnittstellenfunktionen, um einen Watchdog abzuschalten (und anschließend wieder einzuschalten):

Programm:

```
hIecTask := IecTaskGetCurrent(0);
IecTaskDisableWatchdog(hIecTask);
... // Code that is protected against watchdog
IecTaskEnableWatchdog(hIecTask);
```

4.5.6 Task-Stacküberprüfung

Ab MPx-19 wird bei einer Task-Stacküberprüfung der Speicherbedarf der Task durchgeführt. Bei MPB können 8 kB und bei MPC 10 kB Stack genutzt werden. Das bedeutet, dass den (verschachtelten) Funktionsaufrufen innerhalb des IEC-SPS-Programms in jeder Task eine feste Stackgröße zur Verfügung steht (betrifft Aufrufe von "Functions").

Für den Aufruf von Betriebssystemfunktionen (Firmwarefunktionen und -Bausteine) aus einer Function sind 4 kB Speicher reserviert.

Eine Überschreitung des verfügbaren Stackspeichers wird beim Übersetzen der SPS-Applikation als Fehler gemeldet.

Werden dynamische Konstrukte verwendet (objektorientierte Programmierung), so kann der Speicherbedarf der Task nicht berechnet werden. In diesem Fall wird eine Warnung angezeigt.



In Funktionen sollten keine großen Datenfelder als Parameter oder Variablen bzw. Rückgabewerte verwendet werden.

4.5.7 Laufzeitmessungen

Messung der Tasklaufzeiten in Millisekunden

Bei der MLD existieren verschiedene Möglichkeiten zur Ermittlung des Zeitbedarfs. Die Tasklaufzeiten werden vom System automatisch gemessen und statistisch ausgewertet. Weiterhin wird in jedem SPS-Zeitschlitz die noch verbleibende Zeit gemessen.

Die Auswertungen der Tasklaufzeiten können per Funktion im SPS-Programm ermittelt werden. Hierfür wird in der Bibliothek "SysTask" die Funktion "SysTaskGetInfo" mit der Struktur "SYS_TASK_INFO" bereitgestellt. Es wird dabei die Laufzeit der gewünschten Task als Struktur mit Statistik geliefert.

Bei der Beurteilung sind die maximale und durchschnittliche Laufzeit zu betrachten:

- Wenn eine einmalige Zeitüberschreitung bereits zu einem Problem führt und somit überwacht werden soll, so ist die maximale Laufzeit "dwCycleTimeMax" für die Beurteilung heranzuziehen (entspricht der Watchdog-Empfindlichkeit "1"). Am Verhältnis der Maximalzeit zur Watchdogzeit kann man erkennen, wie stark die entsprechende Task ausgelastet wurde. Für spätere Updates der Firmware, welche eventuell geringfügig mehr Eigenbedarf an Rechenleistung hat und vor allem für zukünftige Programmiererweiterungen sollte man hier darauf achten, dass die Auslastung nicht zu hoch liegt (z.B. nicht über 50%).
- Wenn eine einmalige Zeitüberschreitung toleriert wird, kann man der Task höhere Maximallaufzeiten zugestehen. In diesem Fall wird eher das Verhältnis der durchschnittlichen Rechenzeit zur Watchdogzeit zur Beurteilung maßgeblich. Die Maximallaufzeiten von Tasks treten meist in einem Zyklus oder in wenigen Zyklen auf.
- Bei hohen Anforderungen mit einer tolerierten Maximalzeit von nur einer oder zwei Millisekunden ist die Aussagekraft dieser Zeitmessung durch

Grundfunktionen von Rexroth IndraMotion MLD

die Auflösung in Millisekunden relativ ungenau. Hierfür gibt es eine weitere Messung der freien Restzeit.

Beispiel einer Multitasking-SPS-Applikation

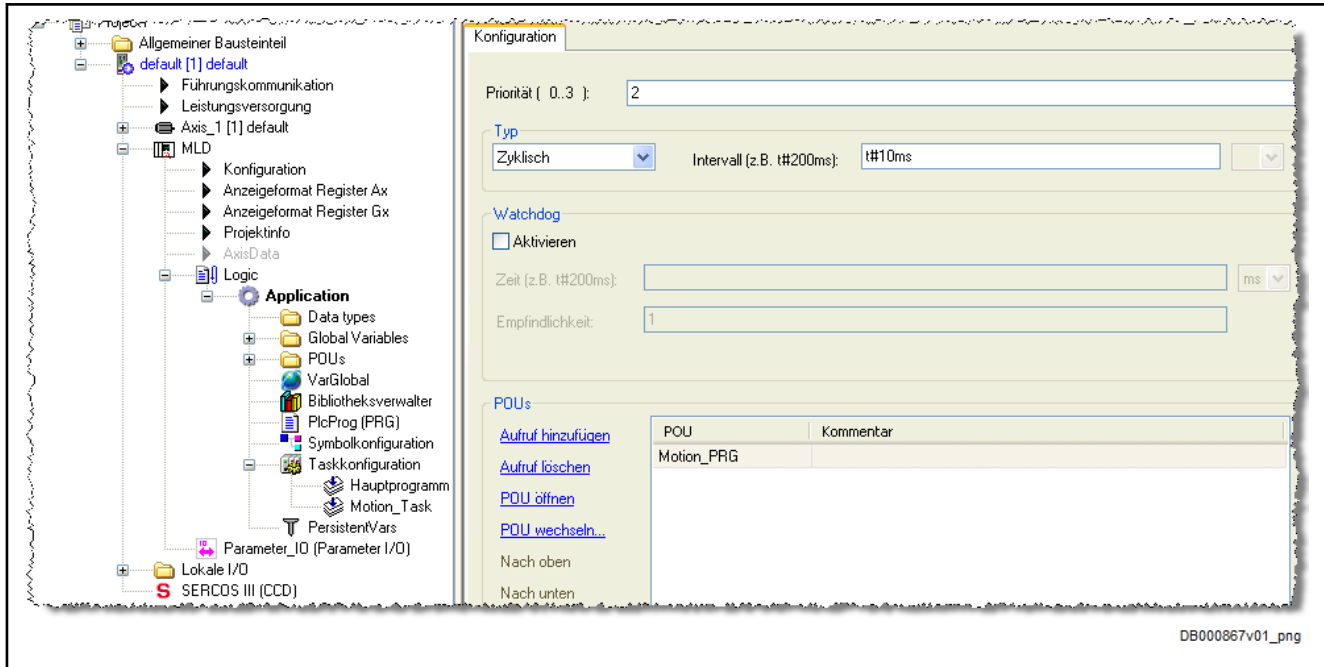


Abb. 4-20: Beispiel einer Multitasking-SPS-Applikation

In dem oben gezeigten Beispiel werden die Laufzeiten dreier Tasks mit 1 ms, 10 ms und 50 ms Zykluszeit dargestellt. Die 1 ms-Task hatte eine maximale Laufzeit von 1ms. Bei einem Watchdog von z. B. 2 ms hat sie eine Reserve von $\geq 50\%$. Es ist hier nicht genau ersichtlich, welche Auslastung die 1ms-Task hat.

Bei der 10ms-Task liegt die Auslastung bei maximal 2 ms. Bei einem Watchdog von z. B. 10 ms hat man eine Reserve von ca. 80%.

Bei der 50ms-Task liegt die Auslastung bei maximal 27 ms. Bei einem Watchdog von z. B. 50 ms hat man eine Reserve von ca. 45%.

Messung der Leerlaufzeit (P-0-1364)

Über die SPS-Funktion "MX_fGetFreeTicks" oder über den Parameter P-0-1364 kann die SPS-Leerlaufzeit ausgelesen werden. Die SPS-Leerlaufzeit zeigt die im SPS-Zeitschlitz noch unverbrauchte Rechenzeit an.

In einer typischen Konstellation von schnellen und langsameren zeitgesteuerten Tasks (ohne freilaufende Task) bringt eine zyklische Anzeige der Leerlaufzeit (z. B. mit dem Oszilloskop) einen groben Überblick über die Auslastung aller Tasks.

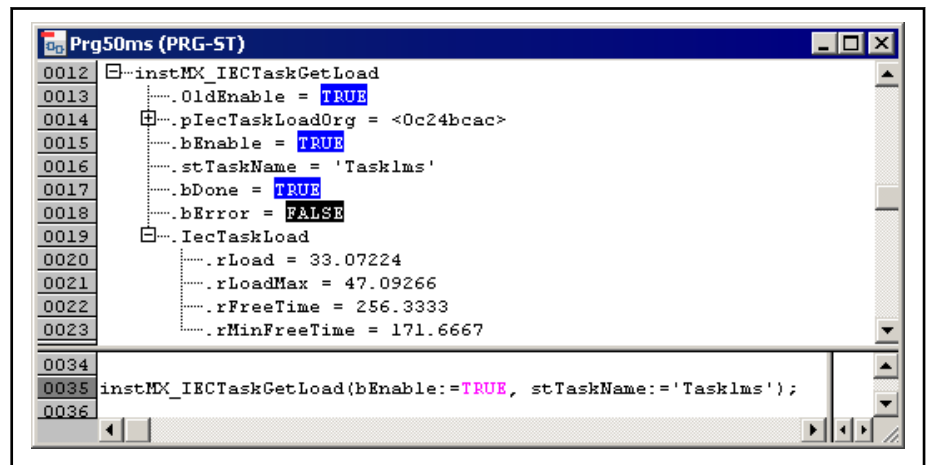
Erweiterte Laufzeitmessung

Bei Bedarf kann die Laufzeit der SPS-Tasks zusätzlich genauer gemessen werden. Dies ist besonders bei schnellen Tasks von Interesse (im Bereich von einer bis wenigen Millisekunden), welche immer in ihrer Periodendauer abgearbeitet sein müssen.

Bei der Messung wird das Zeitschlitzverfahren berücksichtigt. Die Laufzeit der Task wird mit der für sie maximal möglichen Laufzeit (gemäß Periodendauer) in Relation gesetzt. Damit erhält man für jede Task ihre Auslastung in Prozent geliefert. Wenn die Rechenzeit einer hochpriorigen Task steigt, vergrößert sich neben ihrer Auslastung auch die Auslastung der niederpriorigen Tasks, da ihre verfügbare Rechenzeit sinkt. Nicht betrachtet werden Interrupts, wie Stromregler oder Geschwindigkeitsregler.

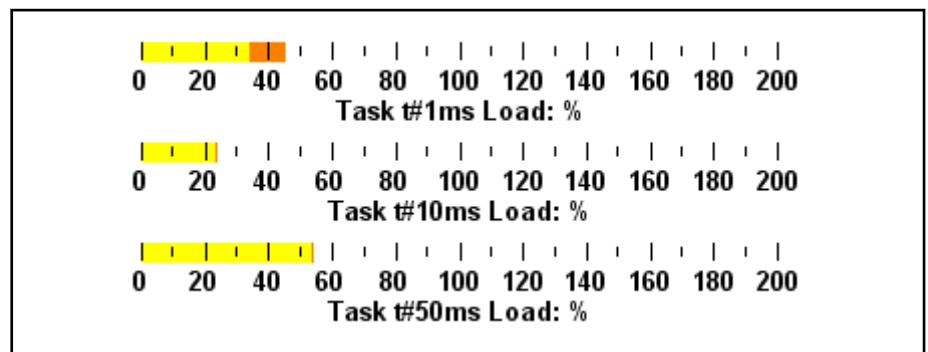
Die Messung erfolgt nicht immer, sondern kann durch Verwendung des entsprechenden Funktionsbausteins "MX_IECTaskGetLoad" aus der Bibliothek "MX_PLCopen" im SPS-Projekt gesteuert werden. Der Baustein liefert die Werte der angegebenen Task. Man kann den Baustein prinzipiell in jeder beliebigen Task aufrufen. (Sinnvoll ist der Aufruf in einer niederpriorigen Task, es sei denn man möchte jeden Zyklus beobachten.) Die Aktivierung der Messungen erfolgt also nach dem Laden eines Projekts, welches den Baustein enthält. Lädt man ein anderes Projekt, welches den Baustein nicht enthält, so wird die Funktion wieder deaktiviert.

In den folgenden Abbildungen wurden die Laufzeiten und die maximalen Laufzeiten von drei Tasks (wie oben) über den Funktionsbaustein "MX_IECTaskGetLoad" gemessen und als Balken dargestellt (gelb: aktuelle Auslastung, rot darunter: maximale Auslastung).



Grafikname DB000346

Abb. 4-21: Beispiel einer Laufzeitmessung

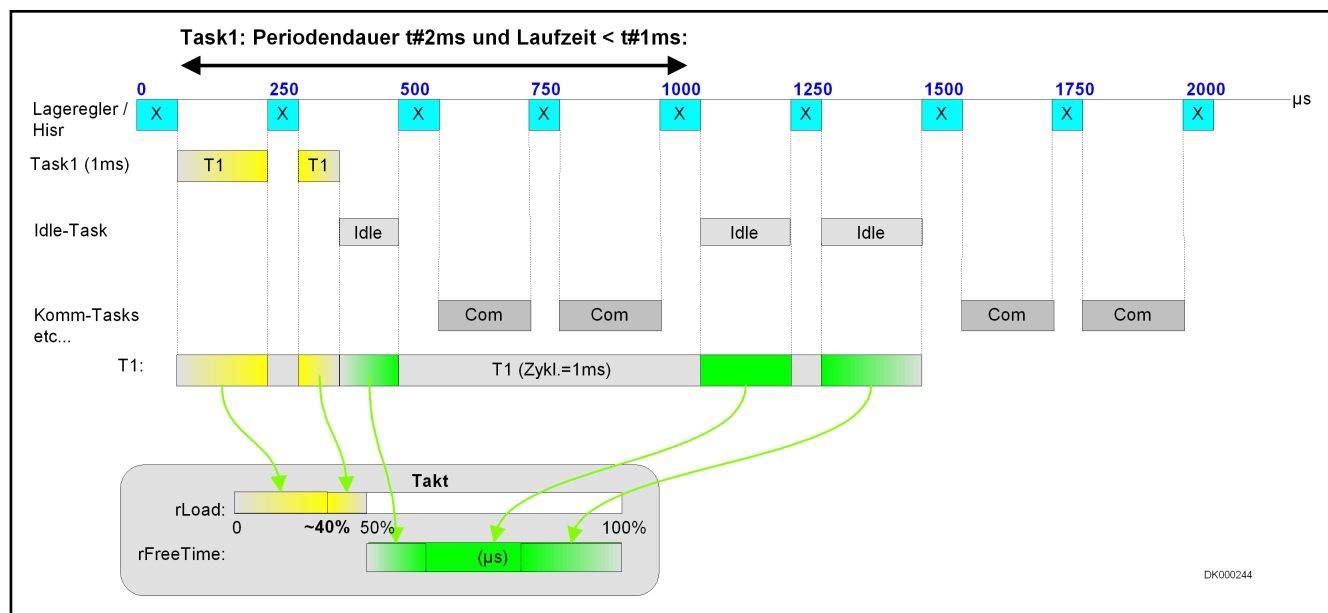


Grafikname DB000347

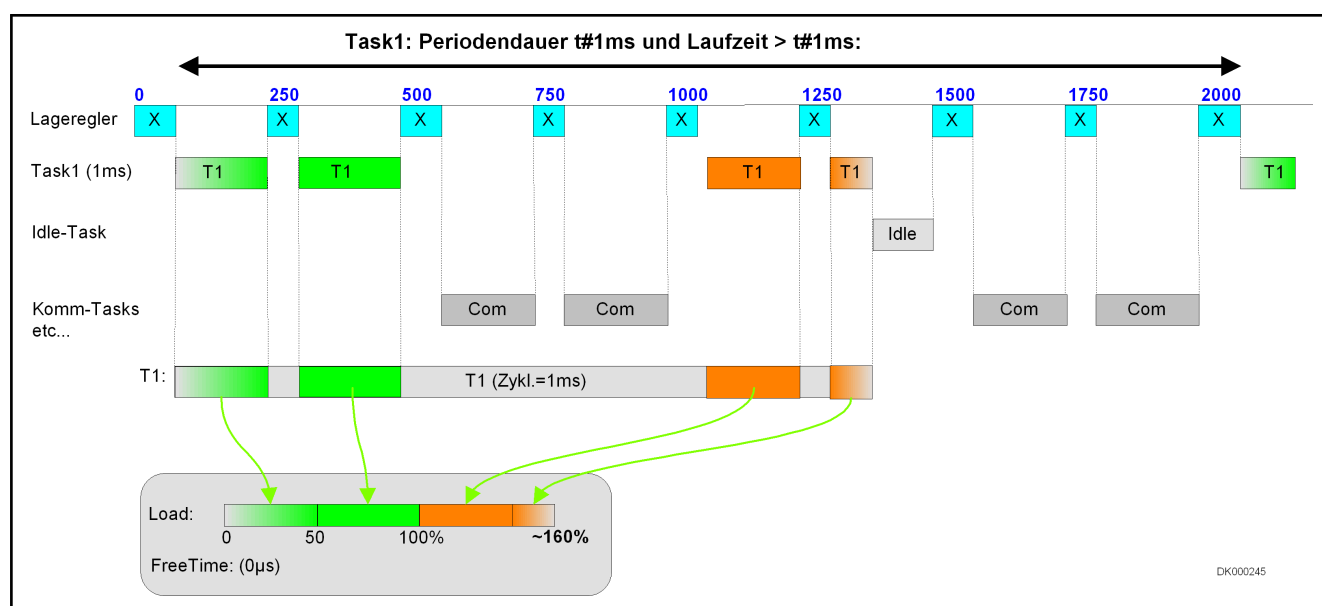
Abb. 4-22: Balkendiagramm zur Laufzeitmessung

In den folgenden beiden Beispielen wird aufgezeigt, wie die Auslastungsanzeige bestimmt wird. Dabei arbeitet die Antriebsregelung in Advanced-Performance mit einer Lageregler-Taktrate (Zykluszeit) von 250µs.

Grundfunktionen von Rexroth IndraMotion MLD



In der obigen Grafik ist eine Messung einer Task mit einer Periodendauer von $t\#2\text{ms}$ dargestellt. Dabei verbraucht die Task nur einen geringen Teil der Rechenzeit. Es ist zu erkennen, dass die 100%-Marke der gesamten für diese Task verfügbaren Rechenzeit entspricht. Die tatsächlich verbrauchte Rechenzeit wird nun in Relation zur 100%-Zeit gestellt und als Prozentwert am Ausgang "rLoad" angezeigt. Die restliche noch nicht verbrauchte Rechenzeit wird als Zeitwert am Ausgang "rFreeTime" in μs angezeigt.



X Lageregler
Com SPS-Kommunikationstask
rLoad verbrauchte Rechenzeit
rFreeTime restliche noch nicht verbrauchte Rechenzeit

Abb. 4-24: Tasklaufzeit-Messung, "overload"

In der obigen Grafik ist eine Messung einer Task mit einer Periodendauer von $t_{\#1ms}$ dargestellt. Dabei verbraucht die Task hier mehr als die vorgegebene Intervallzeit. Die 100%-Marke entspricht wieder der für die Task vorgesehenen Nettozeit innerhalb einer Millisekunde. Die Task benötigt hier jedoch mehr Zeit, so dass der gemessene Prozentwert am Ausgang "rLoad" mehr als 100% angezeigt. Der Ausgang "rFreeTime" zeigt in diesem Fall 0 (μs) an.

4.5.8 Task-Konfiguration

Es muss eine zyklische oder freilaufende Task angelegt werden, damit eine Aktualisierung der E/As möglich ist. Ist keine solche Task definiert, so erscheint eine Fehlermeldung beim Übersetzen des MLD-Projekts und es ist kein Download auf die Steuerung möglich. In einer angelegten Task muss ein entsprechendes Programm aufgerufen werden, sonst erscheint eine Warnung beim Übersetzen des MLD-Projekts.

4.5.9 Task-Zykluszeiten und Timing

SPS-Zykluszeit Die SPS-Zykluszeit T_{PLC} der MLD, definiert die Periodendauer für zyklische Tasks. Eine T_{PLC} setzt sich aus einer oder mehreren SPS-Zeitscheiben zusammen.

SPS-Zeitscheibe Die integrierte SPS (IndraMotion MLD) arbeitet zyklisch in Zeitscheiben, egal welcher Tasktyp verwendet wird. Da die integrierte SPS vom Antriebsprozessor gerechnet wird, muss sie die Ressourcen mit dem übrigen System [Lageregler, Geschwindigkeitsregler, Stromregler, verschiedene Hintergrund-Task (wie z.B. Kommandotask),...] teilen. Pro SPS-Zeitscheibe steht der MLD je nach Systemauslastung eine bestimmte Rechenzeit zur Verfügung. Im Kontext der SPS-Zeitscheiben laufen alle SPS-IEC-Tasks.

Minimale SPS-Zykluszeit Die minimale SPS-Zykluszeit ($T_{PLC,min}$) besteht aus einer SPS-Zeitscheibe; sie definiert auch die Reaktionszeit der Eventtasks. Die minimale Zykluszeit ($T_{PLC,min}$) ist steuerteilunabhängig, sie beträgt 1 ms.

Grundfunktionen von Rexroth IndraMotion MLD



Die maximal einstellbare Zykluszeit beträgt 2000 s (entspricht 33 Minuten)!

Reaktionszeit von Eventtasks

Siehe auch Firmware-Funktionsbeschreibung "Performance-Angaben"

Auch eventgesteuerte Tasks werden innerhalb der SPS-Zeitscheiben gestartet; d.h. deren Reaktionszeit hängt ebenfalls von der SPS-Zykluszeit ab.

Timing

Die erzielbare Programmzykluszeit bzw. das Timing ist abhängig von der Einbindung der SPS in das Tasksystem des Antriebs.

Die Performance der Geräte mit MPB-18- und MPC-18-Firmware kann durch die Einstellung in P-0-0556 geändert werden:

- P-0-0556, Bit 2="0" ⇒ Basic-Performance
- P-0-0556, Bit 2="1" ⇒ Advanced-Performance

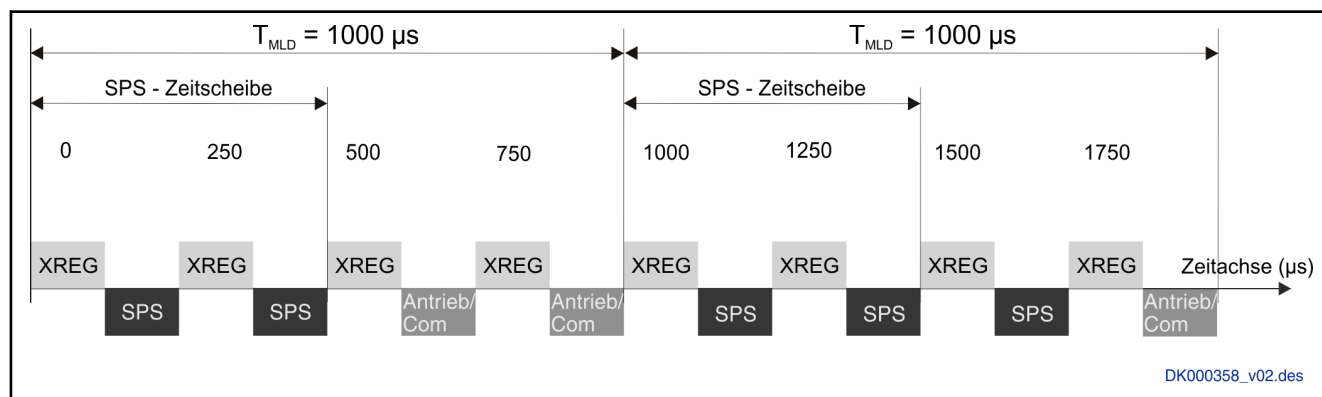


Ist bei MPC-Firmware der "MLD-M-Systemmodus" konfiguriert (P-0-1800.0.1), kann nicht auf Performance-Level "Advanced" umgestellt werden.

Performance [μs]	Stromreglertakt (T_{A_Strom})	Geschwindigkeitsreglertakt (T_{A_Geschw})	Lagereglertakt (T_{A_Lage})	Zykluszeit der SPS (IndraMotion MLD) [T_{MLD}]	Zykluszeit der Führungskommunikation (T_{FKM})
Basic	125	250	500	1000	500
Advanced	62,5	125	250	1000	250

Tab. 4-7: Performance-Einstellungen (Performance-Level)

Die nachfolgenden Grafiken veranschaulichen das Funktionsprinzip (Unterbrechungen durch Geschwindigkeitsregler, Stromregler etc. sind nicht dargestellt):



XREG

Lageregler

SPS

Alle SPS-Tasks (IEC-Tasks) (frei laufende Tasks, periodische Tasks, Event-Tasks)

Antrieb/Com

Kommunikations- und Verwaltungstask

Abb. 4-25:

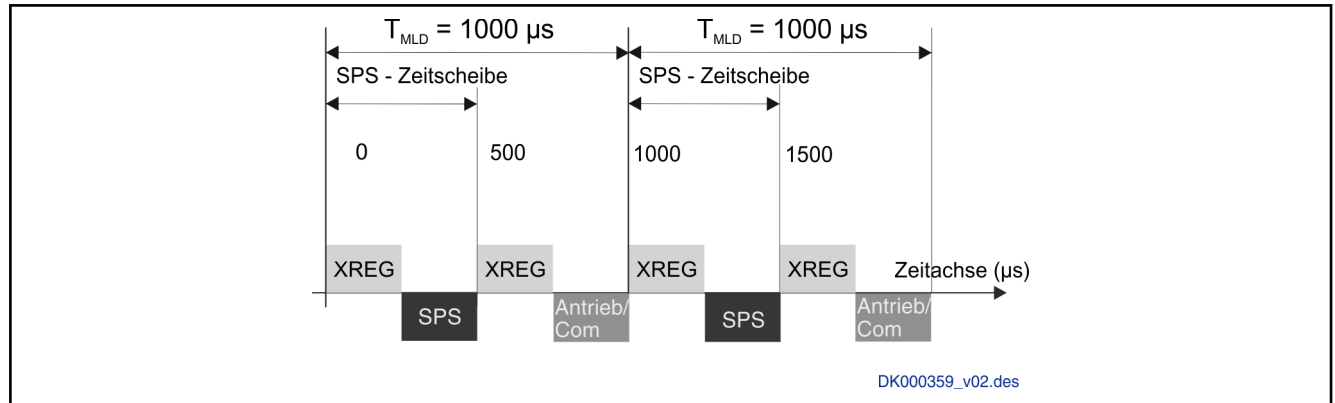
Advanced-Performance (Lageregler alle 250 μs)



Wenn der Antrieb in Advanced-Regelung betrieben wird, dann wird die SPS-Zeitscheibe auch durch den Lageregler unterbrochen und nicht nur durch den Geschwindigkeits- und Stromregler.



Ab der Firmware MPB-18V08 ist die Nutzung der Performance-Stufe "Advanced-Performance" (P-0-0556, Bit 2) nicht möglich, wenn "IndraMotion MLD" aktiv ist [Erweiterungspakete "ML", "MA" oder "TF" (ab MPx-20)].



XREG Lageregler
SPS Alle SPS-Tasks (IEC-Tasks) (frei laufende Tasks, periodische Tasks, Event-Tasks)
Antrieb/Com Kommunikations- und Verwaltungstask
Abb. 4-26: Basic-Performance (Lageregler alle 500µs)

4.6 Fehlerbehandlung der IndraMotion MLD

4.6.1 Betriebsverhalten der IndraMotion MLD

Verhalten bei Programm-Änderung bzw. Download

- Verhalten der MLD(-S)** Ein SPS-Programm kann auch bei einer laufenden Achse geladen werden. Zu diesem Zweck wird bei Programm-Änderung bzw. Download in der MLD(-S) neben dem Setzen der Ausgänge in einen sicheren Zustand (FALSE) auch die Achse in "AH" (bei STOP) bzw. "AB" ohne Reglerfreigabe (bei Reset) geschaltet.
- Verhalten der MLD(-M)** Bei Programm-Änderung bzw. Download wird auch bei Verwendung der MLD-M die lokale Achse in "AH" bzw. "AB" geschaltet, sofern die SPS Kontrolle hat. Entfernte Achsen werden bei Kontrolle (durch MLD-M-Systemmodus) ebenfalls in "AH" bzw. "AB" geschaltet. Zusätzlich wird vor dem Download-Dialog eine Hinweis-Box am Bildschirm angezeigt, falls eine Achse sich in Regelung unter Kontrolle der MLD befindet. Diese Meldung zeigt an, dass bei einem folgenden Download (nicht Online-Change) die Achse(n) angehalten wird (werden).
- Meldung beim Einloggen** Wurde ein Programm geändert, wird beim Einloggen ein Dialog eingeblendet, der Ihnen verschiedene Möglichkeiten bietet:

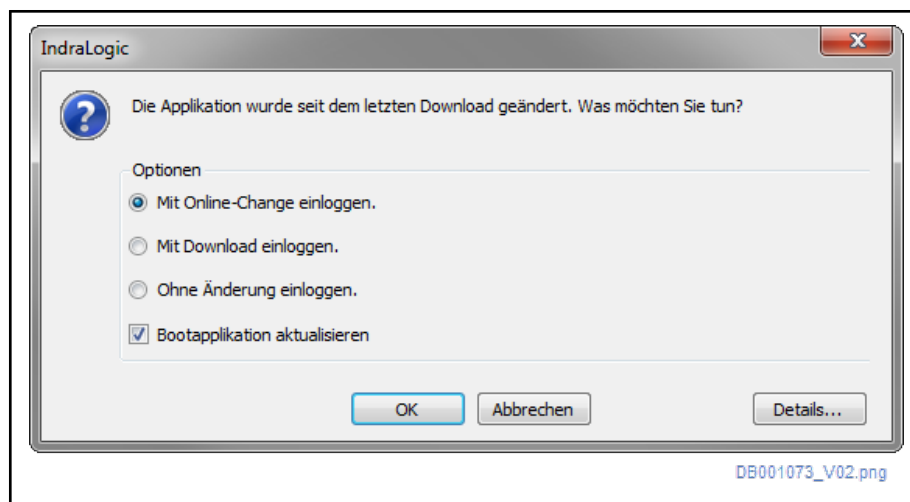


Abb. 4-27: Meldung beim Einloggen

4.6.2 Allgemein (für MLD-S und MLD-M)

Allgemeines

In diesem Abschnitt werden alle allgemeingültigen Funktionen bzgl. der Fehlerreaktion für MLD-S und MLD-M beschrieben. Die spezifischen Themen der MLD-S und MLD-M werden jeweils in einem eigenen Abschnitt dokumentiert.

Kurzbeschreibung und Übersicht



Eine Übersicht aller möglichen Fehler von SPS-Bausteinen wird in der Bibliotheksbeschreibung zu MLD-2G, Abschnitt "Fehler-Referenzlisten" gegeben.

Funktionsmerkmale

Die MLD ist in die Fehlerbehandlung der Antriebsfirmware eingebunden. Fatale Systemfehler werden von der Antriebsfirmware behandelt. In diesem Fall führt der Antrieb eine festgelegte Behandlung durch. Andere Systemfehler, die durch Programmieren der SPS auftreten, werden in der SPS behandelt.

Im Antrieb behandelte Fehler

- Eine Reaktion auf fatale Ausnahmefehler wie fehlerhafter Adresszugriff oder Floating-Point-Exception findet im Antrieb statt. Hier gibt es eine fest definierte Reaktion (siehe hierzu Dokumentation "Hinweise zur Störungsbeseitigung", Kapitel "Verhalten bei fatalem Systemfehler").
- Antriebsfehler und Antriebswarnungen sind jederzeit auch über die Antriebsdiagnose erkennbar (vgl. S-0-0390, S-0-0095,...).
- Generierung einer Warnung bzw. eines Fehlers aus einem SPS-Programm heraus, welche über einfache Funktionsaufrufe ausgelöst werden können:
 - 7 frei definierbare SPS-Warnungen (E2011 .. E2017)
 - 7 frei definierbare SPS-Fehler (F2011 .. F2017)

Beteiligte Bausteine/Funktionen

Folgende Bausteine werden im Zusammenhang mit der Fehlerbehandlung verwendet:

- MX_fSetDriveWarning
- MX_fSetDriveWarningText
- MX_fGetDriveWarning

	• MX_fSetDriveError • MX_fSetDriveErrorText
Zugehörige Diagnosen	Folgende Diagnosen werden im Zusammenhang mit der Fehlerbehandlung verwendet: • E2011 SPS - Warnung Nr. 1 • E2012 SPS - Warnung Nr. 2 • E2013 SPS - Warnung Nr. 3 • E2014 SPS - Warnung Nr. 4 • E2015 SPS - Warnung Nr. 5 • E2016 SPS - Warnung Nr. 6 • E2017 SPS - Warnung Nr. 7 • F2011 SPS - Fehler Nr. 1 • F2012 SPS - Fehler Nr. 2 • F2013 SPS - Fehler Nr. 3 • F2014 SPS - Fehler Nr. 4 • F2015 SPS - Fehler Nr. 5 • F2016 SPS - Fehler Nr. 6 • F2017 SPS - Fehler Nr. 7

In der SPS behandelte Fehler

Fehler, die durch fehlerhafte Programmierung auftreten können

Fehler, die durch fehlerhafte Programmierung auftreten können, sind so genannte SPS-Ausnahmefehler ("Exceptions"). Folgende Fehler werden behandelt:

- Feldzugriffe außerhalb der Grenzen bei Arrays
- Fehlerhafte Zugriffe auf Unterbereichstypen
- Nulldivision
- Watchdog (siehe hierzu "Taskkonfiguration" in IndraLogic)
- Fehlerhafter Pointerzugriff

Die Reaktion auf diese Fehler ist fest definiert:

- Es wird ein Antriebsfehler F6010 erzeugt; die Achse wird gemäß Parametrierung stillgelegt.
- Anzeige durch ein Fehlerbit im SPS-Status (P-0-1351). Eine übergeordnete Steuerung kann damit auf einen MLD-S-Fehler gezielt reagieren.
- Wenn in **SPS-Einstellungen ► Verhalten der Ausgänge bei Stop** so definiert (Default-Einstellung), dann gehen alle Ausgänge in den sicheren Zustand und die SPS geht in STOP.



Alternativ kann "Werte beibehalten" oder "Programm ausführen" konfiguriert werden.

- Die betroffene Task wird sofort abgebrochen.
- Alle SPS-Variablen der betroffenen Task bleiben eingefroren; das erleichtert die Fehlersuche!
- Weitere Tasks laufen bis an ihr Ende und werden nicht mehr aufgerufen (wie "SPS-STOP"). Kommt eine Task in ihrer zweifachen Zykluszeit nicht an ihr natürliches Ende, so wird sie danach abgebrochen.

- Wenn die SPS Kontrolle über den Antrieb hat, wird der Antrieb ausgeschaltet (Systemsteuerung internes EIN).
- Der Fehler wird auch beim nachträglichen Einloggen angezeigt.
- Die SPS lässt sich nicht mehr in RUN versetzen -> meldet beim ersten Versuch einen Fehler. Der Run-Zustand der SPS ist dann nur nach einem SPS-Warmstart möglich.

Fehlersuche bei Übersetzungsfehlern

Am einfachsten ist die Fehlersuche bei Übersetzungsfehlern mit einem (auch nachträglich) angeschlossenen Programmiersystem. Fehler (und auch Warnungen und Informationen) werden in dem Fenster "Meldungen" mit einer kurzen Beschreibung angezeigt. Durch Doppelklick auf eine Meldung gelangt man an die Stelle des Quelltextes, durch die die Meldung verursacht wurde.

Fehlersuche bei Runtime-Fehlern

Bei Runtime-Fehlern ist der Fehler im Parameter "P-0-1365, SPS Fehlermeldung" als Klartext lesbar.

Wenn kein Programmiersystem zur Verfügung steht, so kann der Fehler auch offline eingegrenzt werden, indem der Text des "P-0-1365, SPS Fehlermeldung" ausgelesen wird.

Siehe auch "[Nachverfolgung von SPS-Exceptions](#)"

Fehler, die von Funktionen oder Funktionsbausteinen erkannt werden

Die einzelnen Funktionsbausteine besitzen Ausgänge, welche zur Fehlerbehandlung verwendet werden können:

- "Error": ein Fehlerbit (BOOL)
- "ErrorID": eine Fehlernummer (ENUM-Typ, wenige Werte)
- "ErrorIdent": ein Fehlercode mit erweiterter Fehlerinformation

"Error"

Der Ausgang "Error" ist vom Typ BOOL. Die steigende Flanke von "Error" informiert darüber, dass ein Fehler während der Bearbeitung des Funktionsbausteins aufgetreten ist.

"ErrorID"

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



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

Nachfolgende Tabelle zeigt die möglichen Fehlercodes (ErrorID) und die zugehörigen Fehlerursachen:

ErrorID	Fehlerursache
NONE	Kein Fehler
INPUT_INVALID	Wert außerhalb des programmierbaren Wertebereichs, z. B. Achsnummer 30000
INPUT_RANGE	Wert außerhalb des aktuell möglichen Wertebereichs, z. B. Achse 5 nicht projektiert
RESOURCE	Funktion nicht freigeschaltet "MoveRelative" bei "Open Loop" oder die Hardware ist nicht vorhanden
CALCULATION	Berechnungsfehler, z. B. Overflow in PID-Regler
COMMUNICATION	Kommunikationsfehler, z. B. Störung in Kommunikation zu externen Geräten
STATE_MACHINE	Der aktuelle Zustand der State-Machine verhindert oder verbietet die weitere Abarbeitung

ErrorID	Fehlerursache
ACCESS	Zugriff nicht erlaubt: Schreiben auf spezielle Parameter in Phase 4 oder falsche Phase, keine Freigabe etc.
OTHER	Fehlerausgang ist TRUE, aber Fehlernummer nicht gesetzt
SYSTEM	Systemfehler, z. B. Betriebssystem: Speicherverwaltung. Firmwarefehler, z. B. spezieller Baustein mehrfach instanziiert oder es werden zu viele Echtzeitvariablen verwendet

Tab. 4-8: Standard-Bausteinfehlercodes und Fehlerursachen

"ErrorIdent"

Der Ausgang "ErrorIdent" ermöglicht eine detaillierte Fehlerdiagnose. Die Werte des Ausgangs sind vom Typ ERROR_STRUCT, welcher aus folgenden Elementen besteht:

- "Error.**Table**": Enumerator, 15 Bit mit Reservierungen pro System und Userbereich; kennzeichnet die "Fehlertabelle", aus der Fehlernummern in "ErrorAdditional" eingetragen werden
- "Error.**Additional1**": DWORD, je nach "ErrorTable" unterschiedlich belegt, z. B. Sercos Fehler
- "Error.**Additional2**": DWORD, je nach "ErrorTable" ggf. als zusätzliche Fehlerinformation

Element	Elementdatentyp	Beschreibung
Table	ERROR_TABLE	Der Wert des Elements "Table" beinhaltet die Information, mit Hilfe welcher Fehlertabelle (IndraDrive, Sercos oder MLD-S) der Fehlercode des Elements "Additional1" zu interpretieren ist
Additional1	DWORD	Das Element "Additional1" beinhaltet den Fehlercode zur genauen Spezifizierung des aufgetretenen Fehlers
Additional2	DWORD	"Additional2" beinhaltet Zusatzinformationen (wenn vorhanden)

Tab. 4-9: Aufbau der Ausgabestruktur "ErrorIdent" vom Typ ERROR_STRUCT



Die Struktur und deren Elemente sind in der "CommonTypes"-Bibliothek definiert und per Default mit "0" vorbelegt.

Anwendungs- und Programmierhinweise

Fehler Rücksetzen in der MLD

Fehler können in der MLD wie folgt zurückgesetzt werden:

- Über den Funktionsbaustein "MC_Reset" an allen Achsen, für jede Achse einzeln.
- Über den Funktionsbaustein "MB_ClearAllError" an allen Achsen und E/A-Modulen
- oder -
- über SPS-Reset (vgl. P-0-1350); es werden alle Achsfehler zurückgesetzt, entweder nur das Fehlerflag oder per C05-Kommando.



Dieses Verhalten ist kompatibel zu IndraMotion MLC.

Anzeige der Fehler

Werden beim Kommandieren von Slaves mit Motion-Funktionsbausteinen Fehler erkannt, so erfolgt die Anzeige geräteorientiert in dem kommandierenden Gerät, dem MLD-M-Master. Tritt im Slave ein Fehler auf, so wird dieser primär auch dort angezeigt.

Grundfunktionen von Rexroth IndraMotion MLD

Unterdrückung von SPS-Fehlermeldungen beim Booten

Um ungewollte SPS-Fehlermeldungen zu vermeiden, kann das automatische Starten eines SPS-Bootprojekts verhindert werden, indem folgende Aktion durchgeführt wird:

- Bevor die Anzeige "Boot 2.9" auf dem Display des Bedienfelds erscheint, müssen an dem Bedienfeld gleichzeitig die beiden Tasten <ESC> und <ENTER> gedrückt und gehalten werden.
- Ist das Funktionspaket "Motion Logic" (Antriebs-SPS und Technologiefunktionen) freigeschaltet, erscheint die Anzeige "PLC ?".
- Durch Betätigen der Pfeiltasten an dem Bedienfeld (Pfeil nach unten bzw. Pfeil nach oben) wechselt die Anzeige zwischen "Run PLC" und "Stop PLC".
- Durch Betätigen der <ENTER>-Taste an dem Bedienfeld bei der Anzeige "Stop PLC" wird das Starten eines SPS-Bootprojekts verhindert.

Auslösen von SPS-Warnungen

Durch Aufruf der Funktion "MX_fSetDriveWarning" können die folgenden SPS-Warnungen im Antrieb ausgelöst und zurückgesetzt werden:

- E2011 SPS - Warnung Nr. 1
- E2012 SPS - Warnung Nr. 2
- E2013 SPS - Warnung Nr. 3
- E2014 SPS - Warnung Nr. 4
- E2015 SPS - Warnung Nr. 5
- E2016 SPS - Warnung Nr. 6
- E2017 SPS - Warnung Nr. 7

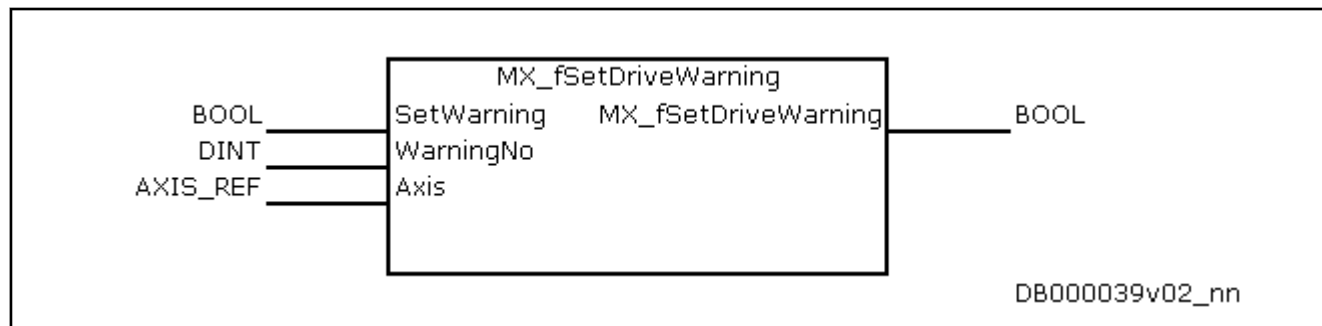


Abb. 4-28: Aufbau des "MX_fSetDriveWarning"

Durch den Aufruf der Funktion "MX_fGetDriveWarning" lassen sich die aktiven SPS-Warnungen rücklesen.

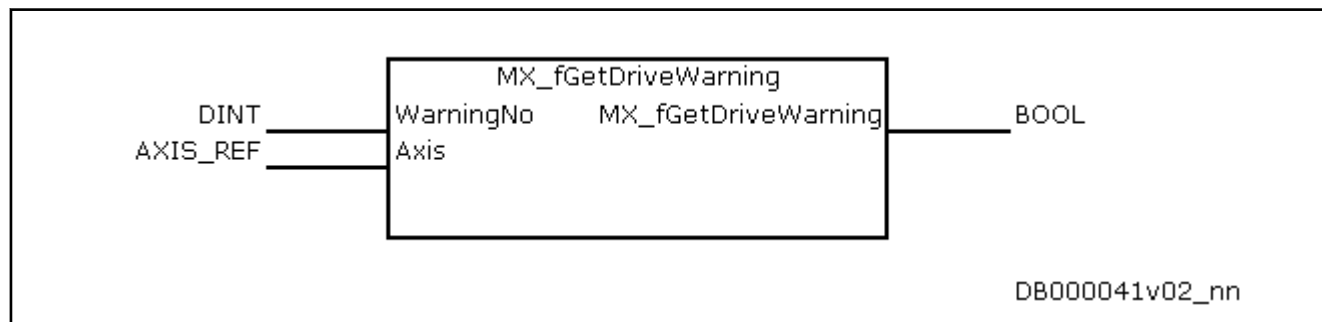


Abb. 4-29: Aufbau des "MX_fGetDriveWarning"

Vorgabe eines Textes für SPS-Warnungen

Durch Aufruf der Funktion "MX_fSetDriveWarningText" kann ein frei wählbarer Text für die übergebene SPS-Warnungen vorgegeben werden. Durch Ausschalten des Antriebs werden die Diagnosetexte wieder auf ihre Defaultwerte gesetzt.

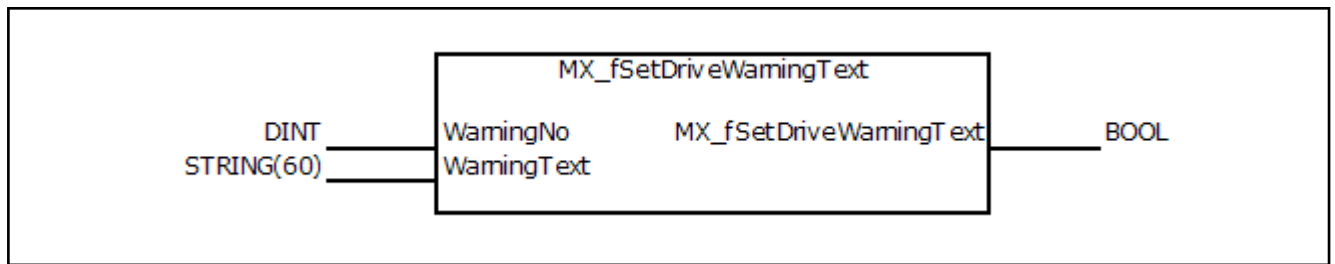


Abb. 4-30: Aufbau des "MX_fSetDriveWarningText"

Auslösen von SPS-Fehlern

Durch den Aufruf der Funktion "MX_fSetDriveError" können die folgenden Antriebsfehler ausgelöst werden:

- F2011 SPS - Fehler Nr. 1
- F2012 SPS - Fehler Nr. 2
- F2013 SPS - Fehler Nr. 3
- F2014 SPS - Fehler Nr. 4
- F2015 SPS - Fehler Nr. 5
- F2016 SPS - Fehler Nr. 6
- F2017 SPS - Fehler Nr. 7

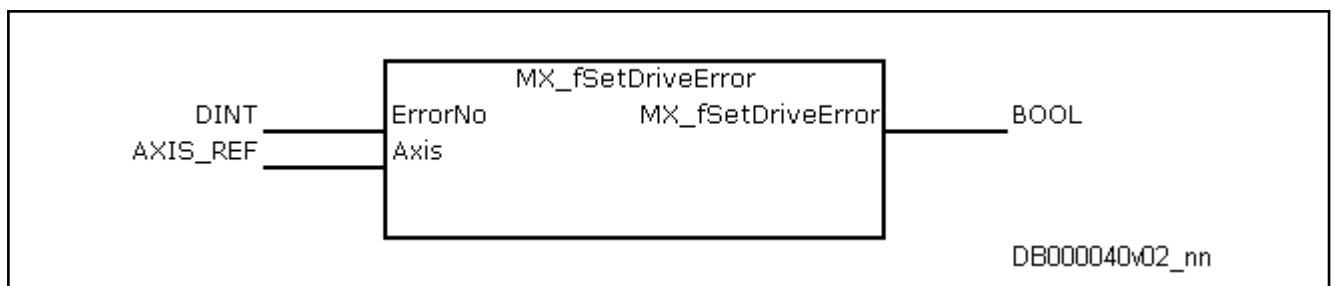


Abb. 4-31: Aufbau des "MX_fSetDriveError"

Vorgabe eines Textes für SPS-Fehler

Durch Aufruf der Funktion "MX_fSetDriveErrorText" kann ein frei wählbarer Text für den übergebenen SPS-Fehler vorgegeben werden. Durch Ausschalten des Antriebs werden die Diagnosetexte wieder auf ihre Defaultwerte gesetzt.

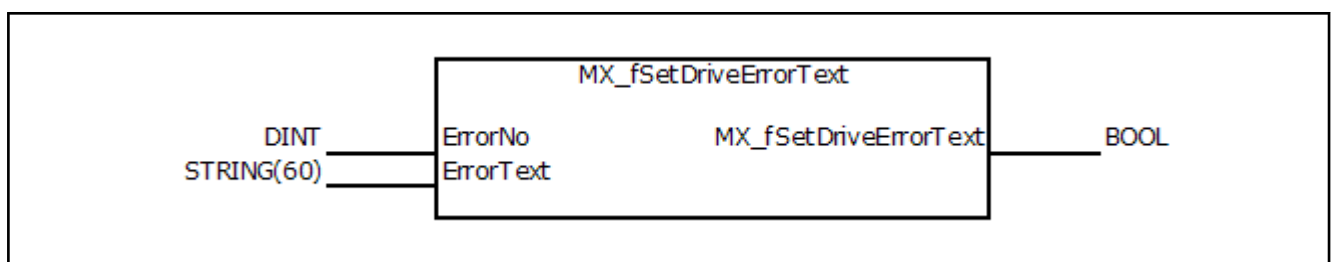


Abb. 4-32: Aufbau des "MX_fSetDriveErrorText"

4.6.3 Konfiguration der Fehlerreaktion bei MLD-S

Kurzbeschreibung

Auf Fehler, die von Motionbausteinen erkannt werden, kann eine automatische Fehlerreaktion aktiviert werden.

Konfiguration der Fehlerreaktion für MLD-S

In der SPS-Konfiguration kann die automatische Reaktion auf "Funktionsbaustein-Fehler" aktiviert / deaktiviert werden:

Grundfunktionen von Rexroth IndraMotion MLD

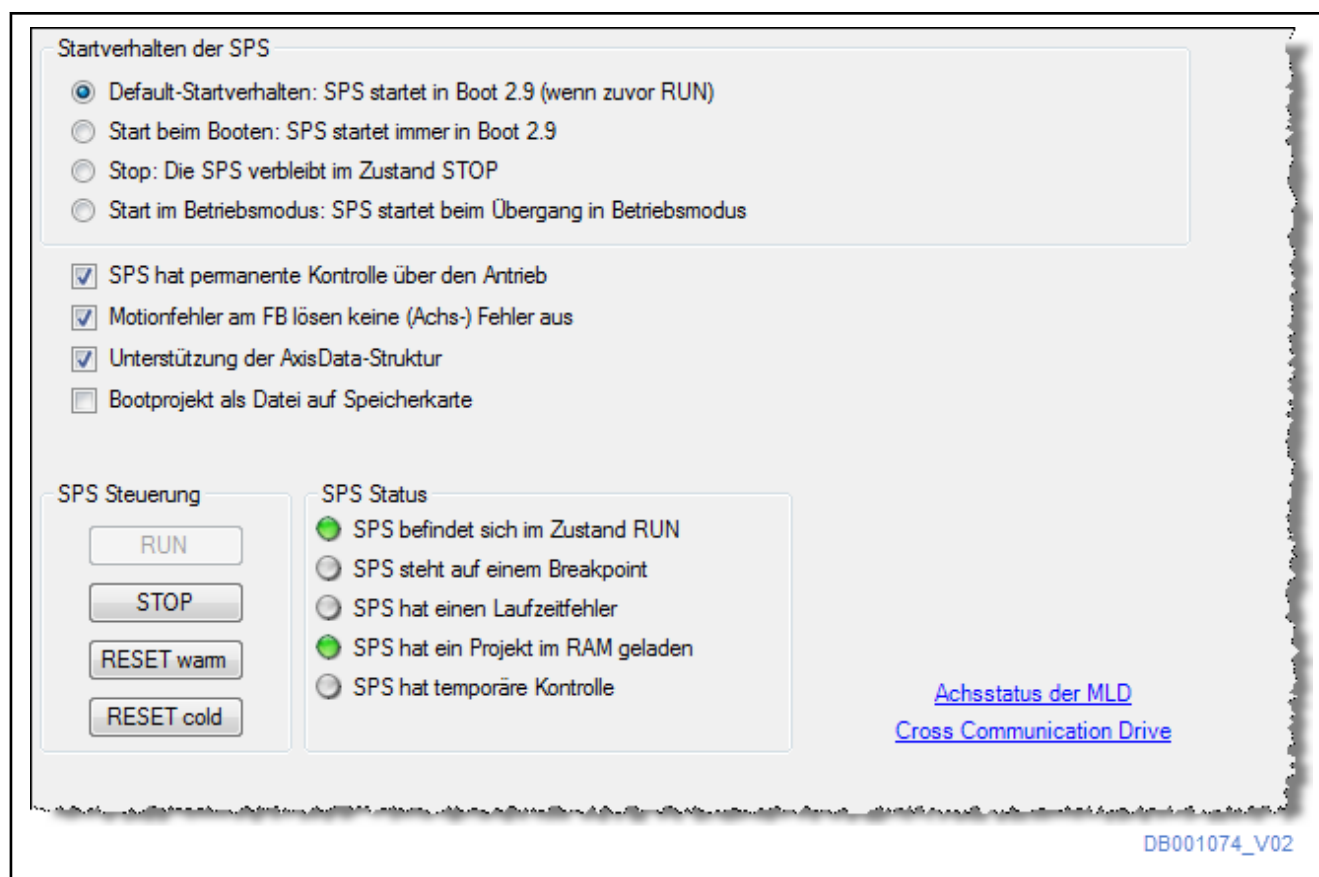


Abb. 4-33: Automatische Reaktion auf "Funktionsbaustein-Fehler" aktivieren / deaktivieren

Fehler	Reaktion (P-0-1367, Bit7="0")
Funktionsbaustein-Fehler	F2150

Tab. 4-10: MLD-S-Fehlerreaktion

Ist die Fehlerreaktion aktiviert, so wird bei einem als fehlerhaft erkannten Baustein aufruf der Fehler F2150 erzeugt. Die SPS läuft weiter und der Baustein meldet einen Fehler am Error-Ausgang.

4.6.4 Fehlerbehandlung und -Reaktion bei MLD-M

Kurzbeschreibung

Bei der Fehlerreaktion eines MLD-M-Antriebssystems ist zu beachten, dass neben einer ggf. vorhandenen Modulbusverbindung eine digitale Verbindung der Achsen (Querkommunikation über Sercos III) besteht. Für die Reaktion des Verbunds im Fehlerfall existieren somit verschiedene Möglichkeiten, die gezielt ausgewählt und aufeinander abgestimmt werden müssen. Man unterscheidet generell folgende Fälle:

- Bei der **autarken achsbezogenen Fehlerreaktion** führen alle Achsen im Verbund eine voneinander unabhängige Fehlerreaktion durch, wenn keine Paketreaktion aktiviert ist und auch die CCD-Fehlerreaktion nicht aktiviert wurde.
- **Paketreaktion des Achsverbunds** (DC-Kopplung): Über den Modulbus werden die Achsen, die im Achsverbund an einem Zwischenkreis betrieben werden, miteinander verbunden und führen im Falle eines Fehlers eine gemeinsame koordinierte Fehlerreaktion (Paketreaktion) aus.

	• CCD-/MLD-M-Fehlerreaktion: Für bestimmte Applikationen (z.B. Gantry-Achse) kann es sinnvoll sein, beim Auftreten eines Fehlers in einem CCD-Slave oder im CCD-Master den kompletten CCD-Verbund kontrolliert oder zumindest in allen Achsen gleichmäßig stillzusetzen. Bei Bedarf kann daher im CCD-Master eine CCD-Fehlerreaktion ausgewählt werden.
Merkmale der CCD-/MLD-M-Fehlerreaktion	Die CCD-/MLD-M-Fehlerbehandlung und -reaktion zeichnet sich durch folgende Merkmale aus: • Konfigurierbare globale Fehlerreaktion mehrerer oder aller Achsen • Optionale automatische Verbundreaktion, um im Fehlerfall alle Achsen auch ohne SPS-Programm stillzusetzen • Umfassende und transparente Verbundreaktion mittels zentraler und einheitlicher Parametrierung der Fehlerreaktion
Verwendete Parameter	Folgende Parameter werden im Zusammenhang mit der MLD-M-/CCD-Fehlerreaktion verwendet: • "P-0-1367, SPS Konfiguration" • "P-0-1800.0.1, CCD: Konfiguration" • "P-0-1810.0.16, CCD: Achsfehler" [Bitleiste mit je einem Fehlerbit (ZKL1) pro Achse (Master und Slaves)]
Zugehörige Diagnosemeldungen	Folgende Diagnosen können im Zusammenhang mit der MLD-M-/CCD-Fehlerreaktion auftreten: • "F2150 MLD Motion FB-Fehler": Dieser Fehler wird auf dem Master (bzw. bei MLD-S) erzeugt, wenn beim Kommandieren der Masterachse ein Fehler festgestellt wird. • "E2140 CCD Teilnehmer-Fehler": Diese Warnung wird im Master erzeugt, wenn für eine Achse ein Funktionsbaustein-Fehler festgestellt wird oder wenn eine Achse in Fehlerzustand ist.

Konfiguration der Fehlerreaktion für MLD-M

Folgende Ursachen führen zu einem Funktionsbaustein-Kommandierungsfehler:

- **Motioninterpreter / Motionhandler**
 - Achse ist in "Stopping" und soll kommandiert werden
 - "MX_SetOpMode" versucht auf einen ungültigen Betriebsartindex umzuschalten (freies Profil)
 - Kommandieren ohne Freischaltpaket
 - Kommandieren ohne Moment
 - Kommandieren ohne Kontrolle
- **Motion-Funktionsbaustein**
 - Konkurrierender Taskzugriff auf eine Instanz
 - Überprüfungen von Eingangswerten (Beschleunigung etc.)

Konfiguration der MLD-M-/CCD-Fehlerreaktion

Über P-0-1800.0.1, Bit 8,7 kann das Verhalten des gesamten CCD-Achsverbundes (lokale und entfernte Achse) im Fehlerfall konfiguriert werden. Man muss hierbei jeweils unterscheiden ob ein Fehler in einem CCD-Slave oder im Master der Auslöser für die Verbundreaktion ist. In der folgenden Übersicht sind die möglichen Fälle dargestellt:

- P-0-1800.0.1, Bit8="0", Bit7="0": **Keine Fehlerreaktion bei Funktionsbaustein-Fehlern**

Grundfunktionen von Rexroth IndraMotion MLD

- Bei einem Funktionsbaustein-Fehler erfolgt keine Reaktion in der betroffenen und allen anderen Achsen.
- Es erfolgt bei einem Fehler im Slave keine Anzeige im CCD-Master.



Bei Bedarf kann unter Verwendung der Bitleiste "P-0-1810.0.16, CCD: Achsfehler" oder "AxisData" eine spezielle Fehlerreaktion in der MLD programmiert werden.

- P-0-1800.0.1, Bit8="0", Bit7="1": **Fehlerreaktion bei Funktionsbaustein-Fehlern**
 - Bei Auftreten eines Achsfehlers oder Funktionsbaustein-Kommandierungsfehlers führt nur die jeweils betroffene Achse ihre selektive Fehlerreaktion durch und alle weiteren Achsen bleiben in Regelung.
 - Im CCD-Master wird die Warnung E2140 angezeigt.
 - Bei Funktionsbaustein-Kommandierungsfehlern wird die betroffene Slaveachse mit "RfAus¹⁾" stillgesetzt!
- P-0-1800.0.1, Bit8="1", Bit7="0": **Mastergeführte synchrone Fehlerreaktion des gesamten CCD-/MLD-Verbundes**
 - Bei Auftreten eines Achsfehlers oder Funktionsbaustein-Kommandierungsfehlers in einer Achse wird im Master eine synchrone und koordinierte Fehlerreaktion für den gesamten Verbund durchgeführt (NC-Reaktion).
 - Im CCD-Master wird der Fehler F2140 und in der jeweiligen Slaveachse der Fehler Fxxxx angezeigt.
 - Alle Achsen werden mit der im Master parametrisierten Rampe (vgl. P-0-0119) stillgesetzt.



Wenn der Master nicht in Regelung ist (z.B. "Ab"), dann wird dort kein F2140 generiert, sondern eine Warnung E2140!

- P-0-1800.0.1, Bit8="1", Bit7="1": **Sofortige Fehlerreaktion des CCD-/MLD-Verbundes**
 - Bei Auftreten eines Achsfehlers oder Funktionsbaustein-Kommandierungsfehlers in einer Achse wird im Master für alle Achsen (lokal oder entfernt) die bestmögliche Stillsetzung (vgl. P-0-0119) eingeleitet, indem Bit 15 im Steuerwort (vgl. S-0-0134) gelöscht wird.
 - Im CCD-Master wird der Fehler F2140 und in der jeweiligen Slaveachse der Fehler Fxxxx angezeigt.
 - Alle Achsen werden mit der jeweils **lokal parametrisierten Rampe** (vgl. P-0-0119) stillgesetzt.



Wenn der Master nicht in Regelung ist (z. B. "Ab"), dann wird dort kein F2140 generiert, sondern eine Warnung E2140!

Anwendungs- und Programmierhinweise



MLD- und CCD-Fehlerreaktion sind miteinander verkoppelt, siehe daher auch Firmware-Funktionsbeschreibung "Querkommunikation (CCD)".

¹⁾ "RfAus": Reglerfreigabe wird abgeschaltet

CCD-Fehlerreaktion Bzgl. der CCD-Fehlerreaktion ist folgendes zu beachten:

- Die Fehlerreaktion bei Funktionsbausteinfehlern wird nur aktiviert, wenn die beiden folgenden Bedingungen erfüllt sind:
 - in P-0-1367 ist die Fehlerreaktion bei Funktionsbausteinfehlern konfiguriert (Bit 7="0")
 - und**
 - in P-0-1800.0.1 ist die Fehlerreaktion bei Funktionsbausteinfehlern konfiguriert (Bit 8="0", Bit 7="1") .
- Im MLD-M-Systemmodus wird die Diagnosenummer (vgl. S-0-0390) automatisch im AT aller CCD-Slaves konfiguriert; somit ist keine selektive Auswahl der fehlerauslösenden Achsen möglich.
- Im CCD-Basismodus bzw. CCD-Systemmodus kann die Fehlernummer selektiv im AT konfiguriert werden, um damit gezielt Achsen aus der automatischen Fehlergenerierung herauszunehmen.



Bei der MLD-M sind neben der Fehlerbehandlung der lokalen Achse auch die entfernten Achsen zu betrachten.

Konfiguration in IndraWorks

Die Konfiguration der Fehlerreaktion erfolgt über folgenden Dialog in IndraWorks:

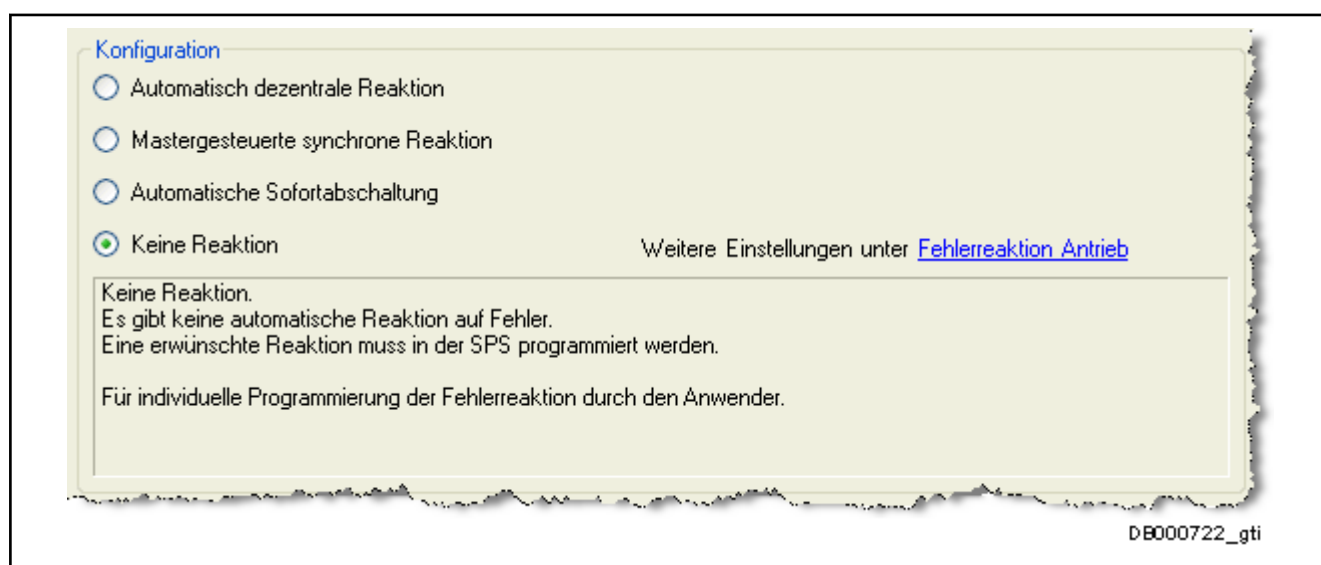


Abb. 4-34: IndraWorks-Dialog: CCD-/ MLD-Fehlerreaktion

Konfiguration über direkten Parameterzugriff

Unabhängig vom Dialog kann auch direkt der "P-0-1367, SPS Konfiguration" editiert werden und über Bit 7 "Funktionsbaustein-Fehlerreaktion deaktivieren" ausgewählt werden, wie auf einen Kommandierungsfehler reagiert wird:

- Bit7="1": Kommandierungsfehler bei Funktionsbausteinen werden ignoriert
- Bit7="0": Kommandierungsfehler bei Funktionsbausteinen werden gemäß nachfolgender Tabelle behandelt

Grundfunktionen von Rexroth IndraMotion MLD

P-0-1800.0.1	P-0-1367, Bit7="0"	Fehler-Ort	Fehler-Ereignis	Master	Slave1	Slave2	
Bit7="0" Bit8="0"	nur MLD	mit Master	FB-Fehler	AF	AF	AF	Keine Reaktion Es gibt keine automatische Reaktion auf Fehler. Eine erwünschte Reaktion muss in der SPS programmiert werden. Für individuelle Programmierung der Fehlerreaktion durch den Anwender.
		Slave1	FB-Fehler	AF	AF	AF	
		Slave2	FB-Fehler	AF	AF	AF	
	mit MLD oder CCD	Master	FnXXX	FnXXX	AF	AF	
		Slave1	FnXXX	AF	FnXXX	AF	
		Slave2	FnXXX	AF	AF	FnXXX	
Bit7="1" Bit8="0"	nur MLD	mit Master	FB-Fehler	F2150	AF	AF	Automatische dezentrale Reaktion Auf F-Fehler in der Achse wird im Master eine Warnung angezeigt. Bei Funktionsbaustein-Kommandierungsfehlern wird die Achse mit "RfAus" ¹⁾ stillgesetzt und im Master eine Warnung angezeigt. Für einfache Anwendungen, bei denen nur die betroffene Achse stillgelegt werden soll.
		Slave1	FB-Fehler	E2140	RfAus	AF	
		Slave2	FB-Fehler	E2140	AF	RfAus	
	mit MLD oder CCD	Master	FnXXX	FnXXX, (E2140)	AF	AF	
		Slave1	FnXXX	E2140	FnXXX	AF	
		Slave2	FnXXX	E2140	AF	FnXXX	
Bit7="0" Bit8="1"	nur MLD	mit Master	FB-Fehler	F2150, (E2140)	AF	AF	Mastergesteuerte synchrone Reaktion Der Master reagiert auf alle Fehler (auch Funktionsbaustein-Fehler) mit Fehler und mit der eingestellten Fehlerreaktion. Alle Slaves (ohne F-Fehler) bleiben in Regelung, um dem Master weiter zu folgen. Für Synchronverbund mit mastergesteuerter synchroner Fehlerreaktion
		Slave1	FB-Fehler	F2140, (E2140)	AF	AF	
		Slave2	FB-Fehler	F2140, (E2140)	AF	AF	
	mit MLD oder CCD	Master	FnXXX	FnXXX, (E2140)	AF	AF	
		Slave1	FnXXX	Fn140, (E2140)	FnXXX	AF	
		Slave2	FnXXX	Fn140, (E2140)	AF	FnXXX	
Bit7="1" Bit8="1"	nur MLD	mit Master	FB-Fehler	F2150, (E2140)	RfAus	RfAus	Automatische Sofortabschaltung Alle Achsen werden bei allen Fehlern mit "RfAus" ¹⁾ stillgesetzt (mit Ausnahme von F-Achsfehlern). Jede Achse führt also direkt ihre Fehlerreaktion durch. Im Master wird der Fehler durch einen Warnung angezeigt. Für nichtsynchrone Mehrachsenanwendungen mit automatischer vollständiger Fehlerreaktion.
		Slave1	FB-Fehler	F2140, (E2140)	RfAus	RfAus	
		Slave2	FB-Fehler	F2140, (E2140)	RfAus	RfAus	
	mit MLD oder CCD	Master	FnXXX	FnXXX, (E2140)	RfAus	RfAus	
		Slave1	FnXXX	F2140, (E2140)	FnXXX	RfAus	
		Slave2	FnXXX	F2140, (E2140)	RfAus	FnXXX	

1) "RfAus": Reglerfreigabe wird abgeschaltet

Tab. 4-11: MLD-M-Reaktion auf Kommandierungsfehler bei Funktionsbausteinen

Sonderfälle bei der MLD-M-Fehlerreaktion

Bzgl. der MLD-M-Fehlerreaktion sind folgende Sonderfälle zu beachten:

- Wenn eine Achse bei einem Funktionsbaustein-Fehler keine Fehlerreaktion durchführt, bleibt der Zustand in der PLCopen State Machine unberührt.
Die Erkennung eines Funktionsbaustein-Fehlers ist mit P-0-1810.0.16 möglich.
- Mittels "P-0-1367, SPS-Konfiguration", Bit7 kann die Erkennung von Funktionsbaustein-Fehlern komplett deaktiviert werden, so dass diese ausschließlich am Funktionsbaustein-Ausgang sichtbar sind.

- E2140 wird im Master immer generiert, wenn eine Achse in Fehler ist (P-0-1810.0.16 != "0") und die CCD-Konfiguration nicht "Keine Reaktion" ist. Die Warnung E2140 wird ggf. durch einen Fehler F2140 überdeckt, die Information ist aber in jedem Fall in dem SPS-Statusregister ersichtlich.
- Wenn der Master nicht in Regelung ist ("Ab"), dann wird dort nicht der Fehler F2140, sondern die Warnung E2140 generiert.

Bei der Auswahl der Fehlerreaktion (vgl. P-0-1800.0.1, Bit8,7) ist folgendes zu beachten:

P-0-1800.0.1 , Bit8,7	Funktion	Verwendungshinweise	Parametrierhinweise
Bit7="0" Bit8="0"	Keine Fehlerreaktion bei Funktionsbaustein-Fehlern	Sinnvoll bei durch Motionsteuerung programmierter Fehlerreaktion (z.B. alle Achsen fahren synchron am virtuellen Leitachsgenerator runter, der im Fehlerfall von der Steuerung angehalten wird).	Bei Bedarf kann unter Verwendung der Bitleiste "P-0-1810.0.16, CCD: Achsfehler" oder "AxisData" eine spezielle Fehlerreaktion in der MLD programmiert werden.
Bit7="0" Bit8="1"	Fehlerreaktion bei Funktionsbaustein-Fehlern	Bei Auftreten eines Achsfehlers oder Funktionsbaustein-Kommandierungsfehlers führt nur die jeweils betroffene Achse ihre selektive Fehlerreaktion durch und alle weiteren Achsen bleiben in Regelung.	
Bit7="1" Bit8="0"	Mastergeführte synchrone Fehlerreaktion des gesamten CCD-/MLD-Verbundes	Sinnvoll für automatisierte Fehlerreaktion bei Synchronverbund von CCD-Slaves auf Leitbewegung des CCD-Masters (z.B. Gantry-Achsen mit CCD-Slave folgt Lageistwert des Masters etc.)	1. Ist diese Fehlerreaktion aktiv, wird zyklisch der Parameter "P-0-170x, CCD Diagnosenummer Slave" ausgewertet. Hierzu muss der P-0-170x für jeden Slave in P-0-1805.x.2 konfiguriert werden. In P-0-1805.x.4 muss an der entsprechenden Stelle der S-0-0390 eingetragen sein. 2. Mit Bit 10 kann ausgewählt werden, ob die Konfiguration von P-0-179x und S-0-0390 für alle Slaves automatisch erfolgt. 3. Um die synchrone Reaktion auch beim Auftreten eines Fehlers im Slave zu gewährleisten, ist in allen Slaves die NC-Reaktion zu konfigurieren (vgl. P-0-0117), damit diese auch im Fehlerfall noch in Regelung bleiben und dem Mastersollwert folgen können.
Bit7="1" Bit8="1"	Sofortige Fehlerreaktion des CCD-/MLD-Verbundes	Sinnvoll für automatisierte Fehlerreaktion ohne aufwändige Ausprogrammierung in der Motionsteuerung (Defaultzustand)	Im CCD-Master werden zyklisch die ZKL1-Fehlerbits (vgl. S-0-1135) der am CCD-Verbund beteiligten Achsen überwacht.

Tab. 4-12: Hinweise zur Auswahl der Fehlerreaktion

5 Kommunikationsschnittstellen und Datenkanäle der MLD

5.1 Einleitung und Übersicht

Kommunikationsschnittstellen der MLD

Die im Antrieb integrierte SPS (Rexroth IndraMotion MLD) besitzt Kommunikationsschnittstellen zu ...

- ... der geräteinternen (lokalen) Achse (über Sercos III angekoppelte Antriebe) und deren Ein-/Ausgängen bzw. Parametern.
- ... den E/A-Modulen von "Rexroth Inline" (Sercos III-**Block**-E/A-Module; Sercos III-**Modular**-E/A-Module).
- ... einer externen Steuerung (z. B. Feldbus-SPS) über eine Führungskommunikations-Schnittstelle (Sercos, Feldbus, ..).
- ... einem externen Bedienteil (HMI).
- ... einem Fremdgerät über Ethernet durch Programmierung von Socket-Kommunikation.

Kommunikationsschnittstellen und Datenkanäle der MLD

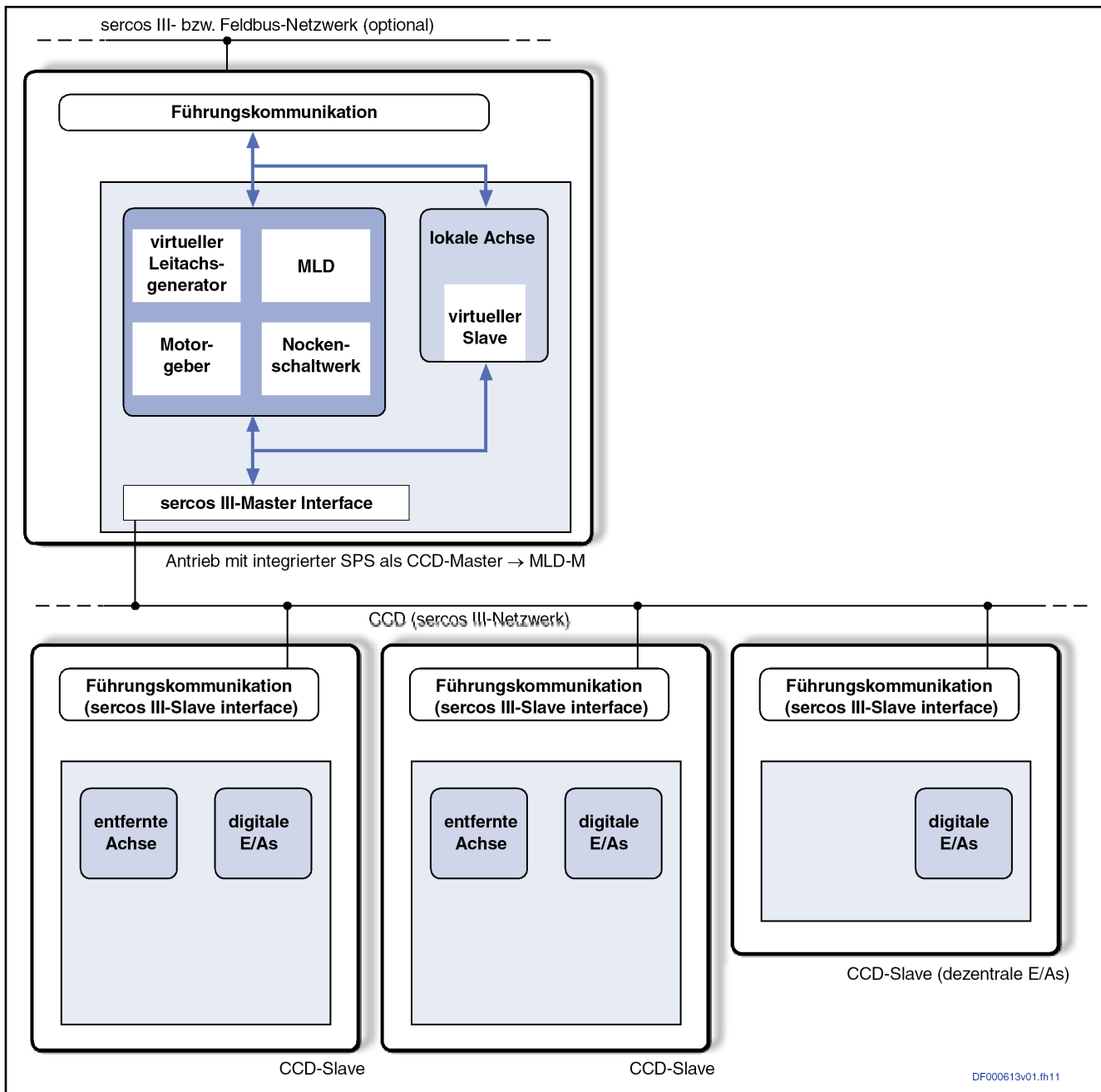


Abb. 5-1: Logische Struktur und Kommunikationsverbindungen von MLD-M

Datenkanäle der IndraMotion MLD

IndraMotion MLD bietet eine Vielzahl von Datenkanälen:

- zum **zyklischen Datenaustausch** zwischen IndraMotion MLD und Achsen
- zur **azyklischen Parameterkommunikation** zwischen IndraMotion MLD und Achsen
- zur **Anbindung** der IndraMotion MLD an ein **Fremdgerät über ein definiertes Kommunikationsprotokoll**, **externe Bedienteile (HMIs)** oder an **externe Steuerungen**

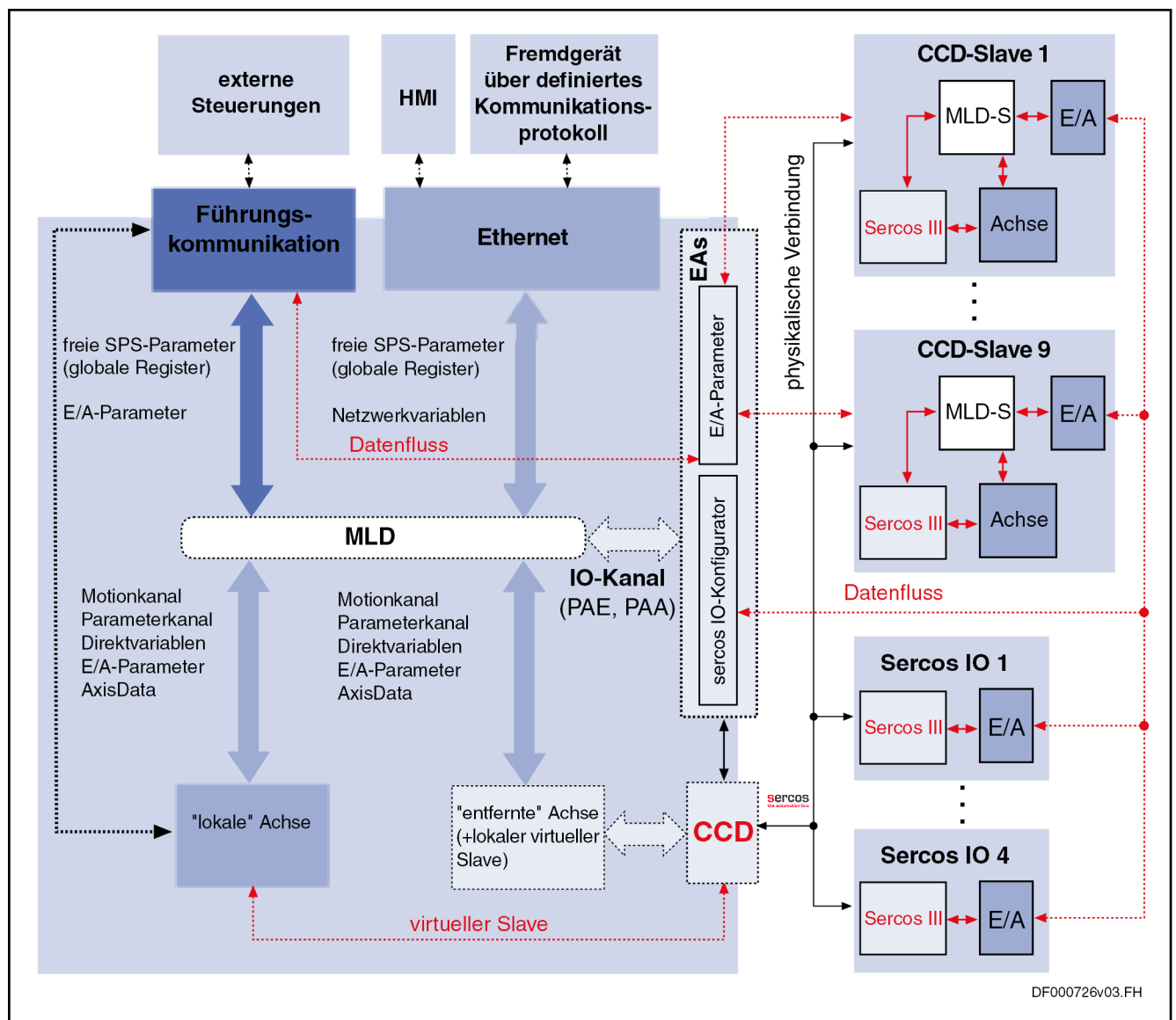


Abb. 5-2: Datenkanäle von IndraMotion MLD für den Zugriff auf lokale und entfernte Achse

Das abgebildete Schema zeigt die Datenkanäle ausgehend von der MLD (SPS) bezogen auf ein vereinfachtes Gerätemodell.

5.2 Datenkanäle der IndraMotion MLD

5.2.1 Zyklische Datenkanäle

Einleitung

Um aus dem SPS-Programm auf antriebsinterne Variablen und Parameter oder vom Antrieb ausgewertete Ein-/Ausgänge und Sensoren bzw. sonstige Schnittstellen des Antriebs zugreifen zu können, existiert eine Vielzahl von Datenkanälen. Rexroth IndraMotion MLD unterstützt folgende Datenkanäle:

- **I/O-Kanal (PAE, PAA)**

Zugriff auf analoge und digitale Ein-/Ausgänge des Antriebs mit Hilfe eines Prozessabbilds

- **Zyklische Achsdaten "AxisData"**

Zyklische Ist- und Sollwerte werden über eine Datenstruktur zur Verfügung gestellt. Diese eignet sich besonders für den Einsatz bei Ein- und Mehrachssteuerung.



Der lokale Echtzeitkanal wird in MLD-2G nicht mehr zur Verfügung gestellt, stattdessen können in Applikationen die AxisData-Schnittstelle und synchronisierte Tasks genutzt werden (siehe auch Dokumentation "Rexroth IndraMotion MLD (2G), Inbetriebnahme ab MPx-18", Stichwort "Echtzeitkanal").

I/O-Kanal (PAE, PAA)

Der I/O-Kanal stellt den Kontakt der IndraMotion MLD zur Außenwelt dar, indem digitale und analoge Ein-/Ausgänge ausgewertet bzw. angesprochen werden können.

Aus dem Programm der MLD kann über Variablen mit entsprechender Adressangabe auf die Prozessabbilder der Ein-/Ausgänge zugegriffen werden.

• Zugriff auf die Eingänge

- Die digitalen und analogen Eingänge werden von der jeweiligen Antriebsfunktion zyklisch im Lagereglertakt ausgewertet (siehe auch Firmware-Funktionsbeschreibung "Digitale Ein-/Ausgänge" und "Analoge Eingänge").
- Vor Task-Beginn kopiert das Task-System das Abbild aus den "SPS Eingang"-Parametern (P-0-1390, P-0-1391,...) in das Prozessabbild-Eingabe (PAE).
- Das Prozessabbild der Eingänge (PAE) wird in jedem Taskzyklus am Anfang ausgelesen; d. h. das PAE wird in die MLD eingelesen. Siehe auch "[Mengengerüst \(Speicher\)](#)" / "[Leistungs- und Performance-Angaben](#)"

• Zugriff auf die Ausgänge

- Das Prozessabbild der Ausgänge (PAA) wird in jedem Taskzyklus am Ende geschrieben; d. h. die "SPS Ausgangs"-Parameter (P-0-1410, P-0-1411,...) werden von der SPS verändert.
- Ein Beschreiben der Parameter der Prozessabbilder ist nicht erlaubt bzw. nicht sinnvoll.
- Die digitalen und analogen Ausgänge werden von der jeweiligen Antriebsfunktion zyklisch im Lagereglertakt bedient (siehe auch Firmware-Funktionsbeschreibung "Digitale Ein-/Ausgänge" und "Analoge Eingänge").

Für die Verknüpfungslogik gilt:

- Eine Änderung der Verknüpfungslogik wird beim Schalten in den Betriebsmodus aktiv.
- Nach dem Booten wirkt die Änderung der Verknüpfungslogik bereits beim Start der SPS.



Wenn die SPS in "STOP" ist, werden trotzdem die Ein- und Ausgänge bedient!

Ein- und Ausgänge, die nicht in einer Task (im Codebereich) verwendet werden, bleiben unverändert und werden nicht bedient.

Eingangsvariablen, die nicht im Programmcode verwendet werden, werden nicht aktualisiert.

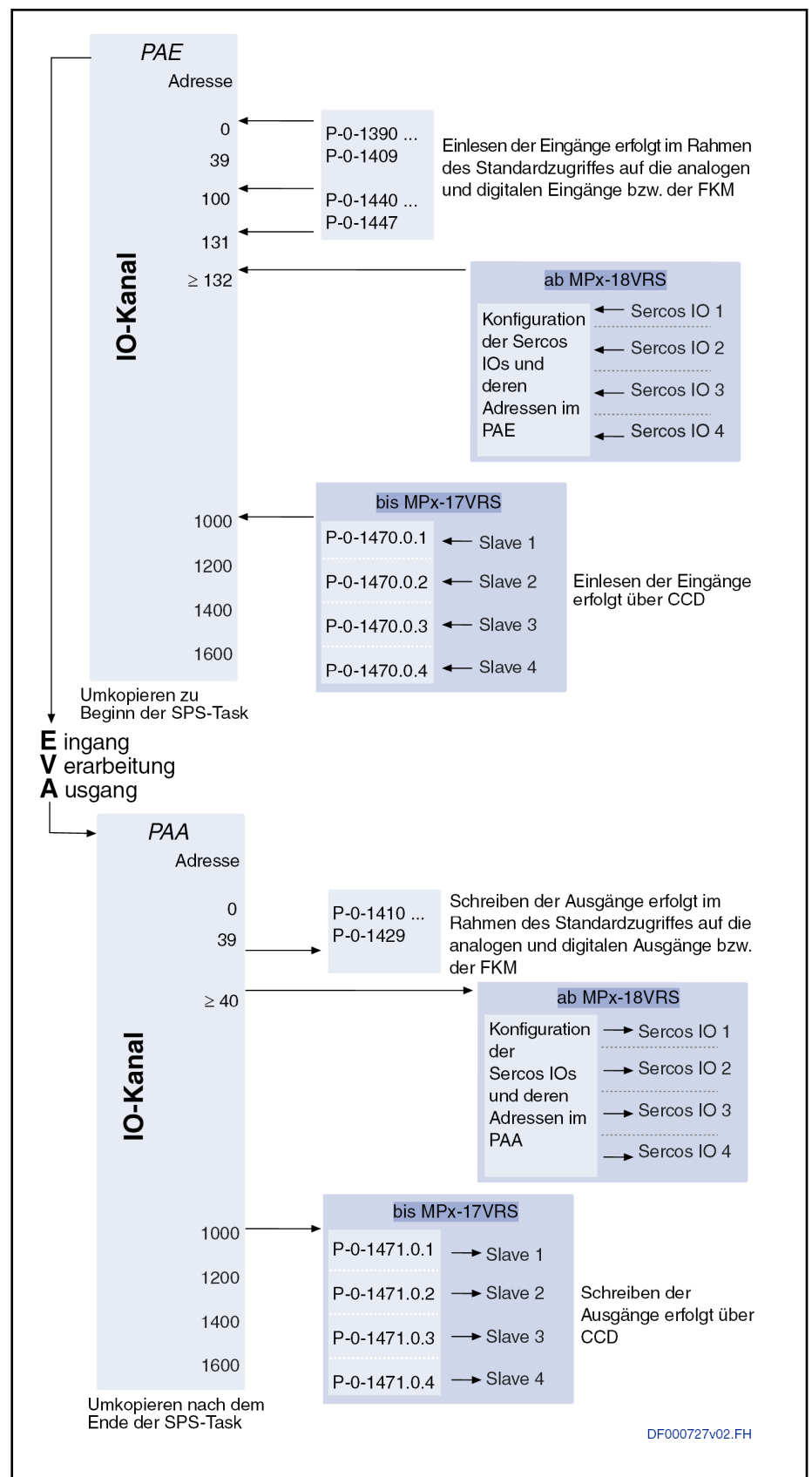


Abb. 5-3: I/O-Kanal (PAE, PAA)

Bei IndraMotion MLD ist es möglich, auf alle lokalen Ein- und Ausgänge des Antriebsregelgeräts zuzugreifen.

Bei IndraMotion MLD können zyklische Sollwerte und Istwerte einer Leitsteuerung als Ein- und Ausgänge verwendet werden.



"Rexroth Inline"-E/As können auch im Einachsbetrieb im MLD-M-Systemmodus (ohne weitere Achsen) genutzt werden. Es können bis zu vier "Rexroth Inline"-E/A-Module betrieben werden.

Merkmale

Der I/O-Kanal zeichnet sich durch folgende Merkmale aus:

- Ein- und Ausgänge liegen im jeweiligen Prozessabbild.
- Es gibt ein gemeinsames Prozessabbild für alle Tasks.
- Die SPS arbeitet byteorientiert nach IEC61131.
- Aufteilung der physikalischen Ein-/Ausgänge zwischen Antrieb und SPS über Identnummern (Parameter) möglich.
- Wirksamkeit einer geänderten I/O-Konfiguration erfolgt nach Umschaltung vom Parametriermodus in den Betriebsmodus.
- Die Konfiguration des I/O-Kanals erfolgt über IndraWorks-Dialoge.
- Nur im Programm benutzte Ein- und Ausgänge werden aktualisiert.
- Sichere Werte bei Verbindungsunterbrechung oder in "STOP"
 - Nullschaltung aller Ausgänge im Zustand "STOP", "RESET" oder gegebenenfalls "Watchdog"
 - Nullschaltung aller Eingänge bei Kommunikationsunterbrechung
- **Update der Prozessabbild-Eingänge**
(PAE) max. $T_{PLC}/2$ (500 μ s bei ADVANCED) vor der SPS-Task. Bei Eingängen, die über die Führungskommunikation oder über CCD gelesen werden, verzögert sich das Update der Prozessabbild-Eingänge zusätzlich um die entsprechende Bus-Zykluszeit.
- **Update der Prozessabbild-Ausgänge**
(PAA) max. $T_{PLC}/2$ (500 μ s bei ADVANCED) nach SPS-Task. Bei Ausgängen, die über die Führungskommunikation oder über CCD geschrieben werden, verzögert sich das Update der Prozessabbild-Ausgänge zusätzlich um die entsprechende Bus-Zykluszeit.

Beteiligte Parameter

Folgende Parameter werden im Zusammenhang mit dem I/O-Kanal (Prozessabbild) verwendet:

- Prozessabbild-Eingabe (PAE):
P-0-1390 bis P-0-1409, P-0-1440 bis P-0-1447 (beachten Sie bitte die in der Parameterbeschreibung angegebene Verfügbarkeit)
- Prozessabbild-Ausgabe (PAA):
P-0-1410 bis P-0-1429 (beachten Sie bitte die in der Parameterbeschreibung angegebene Verfügbarkeit)



Für Anwender, die von MLD-1G auf MLD-2G umsteigen:

In MLD-2G sind in den Prozessabbildern die Listenparameter P-0-1470.0.x / P-0-1471.0.x nicht mehr enthalten.

Werden einem Antrieb, der MLD-2G unterstützt, Sercos I/O-Module zugeordnet (Projektexplorer von IndraWorks, Knoten "Sercos IO"), werden die Sercos I/O-Module und deren Adressen in den Prozessabbildern der MLD automatisch angelegt. Es ist dann möglich, die Sercos I/O-Module den Anforderungen der Anlage anzupassen.

Speicherbereiche und Adressierung

Die Parameter der Prozessabbilder sind wortorientiert angelegt (Eingänge zusätzlich langwortorientiert). Damit ist es möglich, alle im Antrieb vorhandenen Ein- und Ausgänge für die Nutzung in der SPS zuzuweisen.

Die folgenden Adressen sind für den I/O-Kanal reserviert:

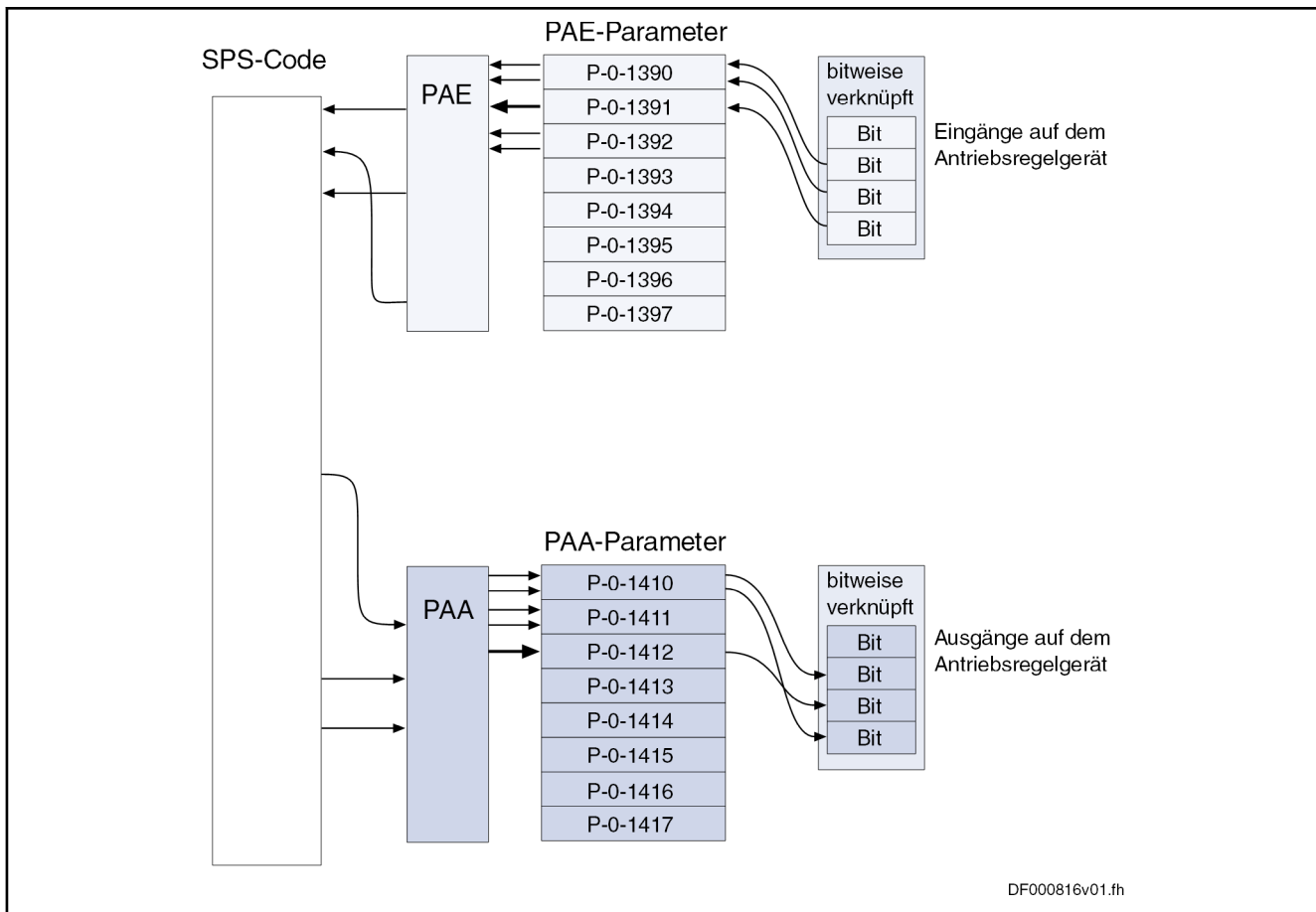
- Prozessabbild-Eingabe (PAE):
 - Der Adressbereich %IB0 bis %IB131 ist für das PAE der lokalen Ein-/Ausgänge reserviert. Der Adressbereich ist nicht änderbar und immer vorhanden.
[Parameter P-0-1390 bis P-0-1409, P-0-1440 bis P-0-1447 (beachten Sie bitte die in der Parameterbeschreibung angegebene Verfügbarkeit)]
 - Der Adressbereich ab %IB132 für "Rexroth Inline"-E/As (Sercos IO)
- Prozessabbild-Ausgabe (PAA):
 - Der Adressbereich %QB0 bis %QB38 ist für das PAA der lokalen Ein-/Ausgänge reserviert. Der Adressbereich ist nicht änderbar und immer vorhanden.
[Parameter P-0-1410 bis P-0-1429 (beachten Sie bitte die in der Parameterbeschreibung angegebene Verfügbarkeit)]
 - Der Adressbereich ab %QB40 für "Rexroth Inline"-E/As

Insgesamt jeweils 16 kByte.

Nachfolgende Beispiele erläutern die Adressierung der Ein- und Ausgänge im Prozessabbild:

Deklaration	Format	Prozessabbild	Parameter
Var AT %IX0.2 : BOOL	Bit	Bit#2 von Wort auf Adresse 0	P-0-1390, Bit 2
Var AT %IX1.9 : BOOL	Bit	Bit#9 von Wort auf Adresse 0	P-0-1391, Bit 9
Var AT %IW3 : WORD	2 Byte	Wort auf Adresse 6	P-0-1393
Var AT %QB2 : BYTE	1 Byte	Byte auf Adresse 2	P-0-1411 (Lowbyte)
Var AT %QD2 : WORD	2 Byte	Wort auf Adresse 8	P-0-1414
Var AT %QB8 : DWORD	4 Byte	Langwort auf Adresse 8	P-0-1414 und P-0-1415

Tab. 5-1: Beispiele für die Adressierung der Ein- und Ausgänge im Prozessabbild



PAE: Prozessabbild-Eingabe

PAA: Prozessabbild-Ausgabe

Abb. 5-4: Beispiel für die Verwendung der Geräte-Ein- und Ausgänge

Verfügbare Geräte-Ein-/Ausgänge

Folgende Ein-/Ausgänge stehen bei den Antriebsregelgeräten "IndraDrive Cs" zur Verfügung:

Analoge Eingänge

- 1 Analogeingang (12 Bit, +/-10 V, $T_A = T_{Lage}$)

Digitale Ein-/Ausgänge

- 7 digitale Eingänge (24 V, $T_A = T_{Lage}$)
- 1 umschaltbarer digitaler Ein-/Ausgang (24 V, $T_A = T_{Lage}$)

Die SPS kann mit der Prozessabbild-Eingabe (PAE) und der Prozessabbild-Ausgabe (PAA) auf die analogen und digitalen Ein-/ Ausgänge des Antriebs zugreifen. Hierfür sind die Ein- und Ausgänge entsprechend zu konfigurieren.

Siehe Firmware-Funktionsbeschreibung "Digitale Ein-/Ausgänge", "Analoge Eingänge" und "Analoge Ausgänge"



Bei Zugriffen über die MLD werden die physikalischen Ein- und Ausgänge über die Verknüpfungslogik des Antriebs mit den Prozessabbildern (PAE und PAA) verbunden.

Gemeinsam benutzte Ein-/Ausgangsvariablen

Für die Ein- und Ausgänge existiert jeweils ein Prozessabbild (PAE und PAA). Dies wirkt sich bei gemeinsam benutzten Variablen in sich unterbrechenden Tasks aus.



Auswirkung bei Multitasking "read - modify - write".

Wird ein Eingang in einer Task benutzt und diese dann durch eine höherprioritäre Task unterbrochen, die denselben Eingang nutzt, so wird das Prozessabbild an dieser Stelle für die hochprioritäre Task aktualisiert. Damit liest die dann wieder weiterlaufende niederprioritäre Task den eventuell neuen Wert. Dies sollte bei Verwendung von mehreren Tasks unterschiedlicher Priorität beachtet werden.



Die selben Ausgänge sollten nicht in mehreren Tasks beschrieben werden!

Startwerte

Bei den Startwerten hängt es von der Deklaration der Variablen ab, welchen Wert sie haben:

- Ohne Vorinitialisierung sind alle Variablen auf "0" gesetzt
- Mit Vorinitialisierung werden alle Variablen mit dem entsprechenden Initialisierungswert vorbesetzt. Beispiel: "MyOutput AT %QX2.4 : BOOL := TRUE;"

Sichere Werte in "STOP"

Im Zustand "STOP", "RESET" oder gegebenenfalls "Watchdog" sind alle Ausgänge auf "0", die Ausgangsvariablen im Prozessabbild der Ausgänge bleiben unverändert.

Sichere Werte bei Verbindungsunterbrechung

- Bei einer Unterbrechung der Verbindung zur **Leitsteuerung** werden die zugeordneten Eingänge auf Null gesetzt.
- Bei einer Unterbrechung der Verbindung zu den **Slaveachsen** werden die zugeordneten Eingänge auf Null gesetzt.
- Bei einer Unterbrechung der Verbindung zu den **Sercos III-E/A-Geräten** werden die zugeordneten Eingänge auf Null gesetzt.

Verarbeitung digitaler Werte

Bei der Verarbeitung digitaler Werte besteht auch die Möglichkeit einzelne Bits statt ganzen boolschen Variablen zu verwenden; bezüglich der Verarbeitungsgeschwindigkeit sind solche Bitzugriffe ungünstig.

Die bitweise Verarbeitung ist mit allen digitalen Ein-/Ausgängen möglich, die byteweise Verarbeitung ist nur mit den digitalen Ein-/Ausgängen auf dem Steuerteil und den digitalen E/A-Erweiterungen möglich.

Multitasking

Wenn auf Bits zugegriffen wird, die einem Byte angehören, kann trotz präemptivem Multitasking sicher auf einzelne Bits zugegriffen werden (read - modify - write).



Das Setzen bzw. Löschen von Bits durch die SPS kann nicht unterbrochen werden und ist somit "tasksicher"! Dies gilt jedoch nicht bei externen Zugriffen (z.B. über das Programmiersystem)!

Zyklische Achsdaten "AxisData"

Kurzbeschreibung

Für den zyklischen Zugriff auf die lokale und auf entfernte Achsen stehen vordefinierte Achsdatenstrukturen ("AxisData") mit einigen Soll- und Istwerten zur Verfügung.

Merkmale

- Bei "AxisData" handelt es sich um eine IEC61131-Datenstrukturdefinition, welche einige wichtige Istwerte der Achse enthält. Weiter sind noch konfigurierbare Ist- und Sollwerte enthalten, mit denen der Anwender eigene zyklische Daten lesen und ggf. schreiben kann. Dies ermöglicht somit einen einfachen Zugriff auf die wichtigsten Achsdaten.



Bei Verwendung einer FKM-synchronen Motiontask bzw. einer CCD-synchronen Motiontask (siehe dazu Kapitel "Grundfunktionen von Rexroth IndraMotion MLD" Abschnitt [Task-system](#)) erfolgt die Verarbeitung der Achsdatenstruktur "AxisData" synchron zur Taskverarbeitung. D. h. die Istwerte der Struktur "AxisData" werden am Anfang der Task aktualisiert und die Sollwerte werden am Ende der Task geschrieben. Bei allen anderen Tasks erfolgt die Verarbeitung der Achsdaten nicht synchron zur Taskverarbeitung, dies muss durch entsprechende Zugriffe im Programm sichergestellt werden.

Zugehörige Bausteine



Der Zugriff auf die Datenstruktur "AxisData" erfolgt nicht über Funktionsbausteine, sondern durch direkten Zugriff auf die Datenstrukturen.

Aufbau der AxisData

Der Strukturtyp "MX_AXISDATA" ist in der "MX_Base" definiert.

Elementname	Datentyp	Richtung	Parameter in MLD Master	Parameter in Achse	Bedeutung
Konfigurierbare Sollwertbits					
wUserCmdDataBitA_q	WORD	Sollwert	P-0-172x.12	(konfigurierbar)	Frei verwendbares Sollwertbit
wUserCmdDataBitB_q	WORD	Sollwert	P-0-172x.13	(konfigurierbar)	Frei verwendbares Sollwertbit
wUserCmdDataBitC_q	WORD	Sollwert	P-0-172x.14	(konfigurierbar)	Frei verwendbares Sollwertbit
wUserCmdDataBitD_q	WORD	Sollwert	P-0-172x.15	(konfigurierbar)	Frei verwendbares Sollwertbit
Konfigurierbare Sollwerte					
dwUserCmdDataA_q	SM_TYPES	Sollwert	32 Bit Zielparameter: P-0-173x 16 Bit Zielparameter: P-0-182x	(konfigurierbar)	Frei verwendbarer Sollwert
dwUserCmdDataB_q	SM_TYPES	Sollwert	32 Bit Zielparameter: P-0-174x 16 Bit Zielparameter: P-0-183x	(konfigurierbar)	Frei verwendbarer Sollwert
dwUserCmdDataC_q	SM_TYPES	Sollwert	32 Bit Zielparameter: P-0-175x 16 Bit Zielparameter: P-0-184x	(konfigurierbar)	Frei verwendbarer Sollwert
dwUserCmdDataD_q	SM_TYPES	Sollwert	32 Bit Zielparameter: P-0-176x 16 Bit Zielparameter: P-0-185x	(konfigurierbar)	Frei verwendbarer Sollwert
Konfigurierbare Istwertbits					
wUserActualDataBitA_i	WORD	Istwert	P-0-171x.12	(konfigurierbar)	Frei verwendbares Istwertbit
wUserActualDataBitB_i	WORD	Istwert	P-0-171x.13	(konfigurierbar)	Frei verwendbares Istwertbit
wUserActualDataBitC_i	WORD	Istwert	P-0-171x.14	(konfigurierbar)	Frei verwendbares Istwertbit
wUserActualDataBitD_i	WORD	Istwert	P-0-171x.15	(konfigurierbar)	Frei verwendbares Istwertbit

Kommunikationsschnittstellen und Datenkanäle der MLD

Elementname	Datentyp	Richtung	Parameter in MLD Master	Parameter in Achse	Bedeutung
Konfigurierbare Istwerte					
dwUserActualDataA_i	SM_TYPES	Istwert	32 Bit Quellparameter: P-0-177x 16 Bit Quellparameter: P-0-186x	(konfigurierbar)	Frei verwendbarer Istwert
dwUserActualDataB_i	SM_TYPES	Istwert	32 Bit Quellparameter: P-0-178x 16 Bit Quellparameter: P-0-187x	(konfigurierbar)	Frei verwendbarer Istwert
dwUserActualDataC_i	SM_TYPES	Istwert	32 Bit Quellparameter: P-0-179x 16 Bit Quellparameter: P-0-188x	(konfigurierbar)	Frei verwendbarer Istwert
dwUserActualDataD_i	SM_TYPES	Istwert	32 Bit Quellparameter: P-0-180x 16 Bit Quellparameter: P-0-189x	(konfigurierbar)	Frei verwendbarer Istwert
Achsstatusinformationen [Zugriff erfolgt i.d.R. über die Zustandsbits bzw. Achsstatus (PLCopen state machine)]					
wDriveStatus_i	WORD	Istwert	P-0-166x	S-0-0135	Achsstatus - die Bits sind einzeln verfügbar
wDriveExtStatus_i	WORD	Istwert	-	-	Achsstatus - die Bits sind einzeln verfügbar
wSignalStatus_i	WORD	Istwert	P-0-171x	S-0-0144	Signalstatus - die Bits sind einzeln verfügbar
wSyncStatus_i	WORD	Istwert	P-0-181x	P-0-0089	Synchstatus - die Bits sind einzeln verfügbar
dwDeviceStatus_i	DWORD	Istwert	P-0-1630.x.2	S-0-1045	Devicestatus - die Bits sind einzeln verfügbar
dwPLCopenStatus_i	DWORD	Istwert	-	-	Status nach PLCopen - die Bits sind einzeln verfügbar
dwDiagNumber_i	DWORD	Istwert	P-0-170x	S-0-0390	Diagnose Nummer
Feste Istwerte					
rActualPosition_i	REAL	Istwert	P-0-167x	S-0-0386	Aktiver Lageistwert
rActualVelocity_i	REAL	Istwert	P-0-168x	S-0-0040	Geschwindigkeits-Istwert
rActualTorqueForce_i	REAL	Istwert	P-0-169x	S-0-0084	Drehmoment-/Kraft-Istwert
Feste Zustandsbits					
Axis_CamTab_0	BIT	Istwert	P-0-181x.0	P-0-0089.0	Aktive Kurvenscheibe Bit#0
Axis_CamTab_1	BIT	Istwert	P-0-181x.1	P-0-0089.1	Aktive Kurvenscheibe Bit#1
Axis_CamTab_2	BIT	Istwert	P-0-181x.2	P-0-0089.2	Aktive Kurvenscheibe Bit#2
Axis_CamSwitching	BIT	Istwert	P-0-181x.5	P-0-0089.5	Status Hubumschaltung
Axis_CmdValueReached	BIT	Istwert	P-0-171x.0	P-0-0115.12	Sollwert erreicht
Axis_Standstill	BIT	Istwert	P-0-171x.1	S-0-0331.0	Meldung Stillstand
Axis_InSynchron	BIT	Istwert	P-0-181x.8	P-0-0089.8	Folgeachse ist synchronisiert
Axis_Homed	BIT	Istwert	P-0-171x.2	S-0-0403.0	Status des Referenzgebers Lage-Istwert

Elementname	Datentyp	Richtung	Parameter in MLD Master	Parameter in Achse	Bedeutung
Axis_Inbb	BIT	Istwert	P-0-166x, Bit (!15 & 14)	S-0-0135 (Bit15 Bit14)	Steuerteil betriebsbereit
Axis_InAb	BIT	Istwert	P-0-166x, Bit (15 & !14)	S-0-0135 (Bit15=1)	Steuer- und Leistungsteil betriebsbereit
Axis_Power	BIT	Istwert	P-0-166x, Bits (15 & 14)	S-0-0135 (Bit15=1 & Bit14=1)	Antrieb momentan behaftet
Axis_FollowingCommand	BIT	Istwert	P-0-166x, Bit 3=1	S-0-0135 (Bit3=1)	Antrieb folgt der Sollwertvorgabe durch die MLD
Axis_Interrupted	BIT	Istwert	P-0-166x (Bit15=1 & Bit14=1 & Bit13=0 & Bit3=0) und P-0-165x (Bit13=1)	S-0-0135 (Bit15=1 & Bit14=1 & Bit13=0 & Bit3=0) und S-0-0134 (Bit13=1)	Motion-Kommandierung wurde durch übergeordnete Antriebsfunktion unterbrochen
Axis_Warning	BIT	Istwert	P-0-166x.12	S-0-0135.12	Änderungsbit Zustandsklasse 2 - Warnung
Axis_Error	BIT	Istwert	P-0-166x.13	S-0-0135 (Bit13=1)	Antriebsfehler, Fehler in Zustandsklasse 1
Achsstatus (PLCopen state machine)					
Errorstop	BIT	Istwert	-	-	PLCopen Status: Errorstop
Stopping	BIT	Istwert	-	-	PLCopen Status: Stopping
Homing	BIT	Istwert	-	-	PLCopen Status: Homing
DiscreteMotion	BIT	Istwert	-	-	PLCopen Status: DiscreteMotion
ContinuousMotion	BIT	Istwert	-	-	PLCopen Status: ContinuousMotion
SynchronizedMotion	BIT	Istwert	-	-	PLCopen Status: SynchronizedMotion
StandStill	BIT	Istwert	-	-	PLCopen Status: StandStill
Disabled	BIT	Istwert	-	-	PLCopen Status: Disabled
PreSetMode	BIT	Istwert	-	-	PLCopen Status: PreSet-Mode

Elementname

IEC-Name des Datenelements. Mit diesem Namen wird das Strukturelement im Programm angesprochen

Datentyp

IEC-Datentyp des Datenelements

Richtung

Sollwerte werden vom Anwender geschrieben und vom System übertragen

Parameter in MLD Master

Im Master existiert für dieses Element ein Diagnose-Parameter. Dieser zeigt den Wert an, ist aber nicht beschreibbar

Parameter in Achse

Quell- bzw. Zielparame-ter auf der jeweiligen Achse. Hier kann ein entsprechender fester Parameter stehen. Bei den konfigurierbaren Elementen ist der Parameter einstellbar. Bei den anderen Elementen gibt es keinen Parameter auf der Achse.

Tab. 5-2: Aufbau der AxisData

Konfigurierbare Inhalte

Die folgenden Elemente können per Konfiguration verschiedenen Parametern der Achsen zugeordnet werden. Die Konfiguration erfolgt über den Dialog für AxisData (siehe auch "5-2: Aufbau der AxisData").

- **Konfigurierbare Sollwertbits**

Mit den Elementen "wUserCmdDataBitA_q" bis "wUserCmdDataBitD_q" können bis zu vier frei konfigurierbare Parameter-Bits für jede reale Achse kommandiert werden.

- **Konfigurierbare Sollwerte**

Mit den Elementen "dwUserCmdDataA_q" bis "dwUserCmdDataD_q" können bis zu vier frei konfigurierbare Parameter für jede reale Achse kommandiert werden.

- **Konfigurierbare Istwertbits**

Mit den Elementen "wUserActualDataBitA_i" bis "wUserActualDataBitD_i" können bis zu vier frei konfigurierbare Parameter-Bits aus jeder realen Achse eingelesen werden.

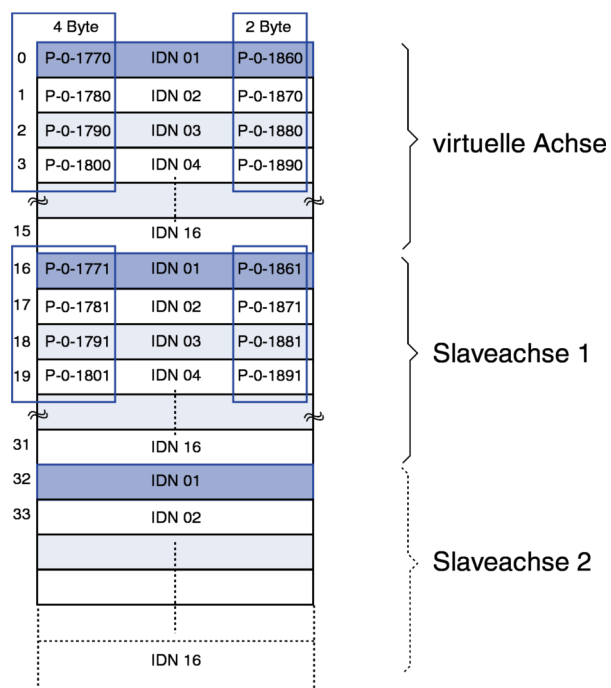
- **Konfigurierbare Istwerte**

Mit den Elementen "dwUserActualDataA_i" bis "dwUserActualDataD_i" können bis zu vier frei konfigurierbare Parameter aus jeder realen Achse eingelesen werden.



Werden die konfigurierbaren Istwerte "manuell"; d.h. nicht über IndraWorks konfiguriert, so muss folgendes beachtet werden (hier am Beispiel der Parameter P-0-1624 und P-0-1626):

- Bei den Parametern P-0-1624 und P-0-1626 sind die ersten vier Identnummern (IDN) jeder Achse für die konfigurierbaren Istwerte zu verwenden. Dabei ist die Reihenfolge (IDN01=dwUserActualDataA_i, IDN02=dwUserActualDataB_i, ...) zwingend einzuhalten (siehe Grafik).
- Im Parameter P-0-1624 wird das Mapping für die konfigurierbaren Istwerte parametrisiert. Die zu den Datencontainer A-D zugehörigen Parameter sind in "5-2: Aufbau der AxisData" aufgelistet.
- Im Parameter P-0-1626 werden in gleicher Reihenfolge die gewünschten/interessanten Achsinformationen parametrisiert.



DF000556v02

SM_TYPES

Bei den konfigurierbaren Ist- und Sollwerten wird die Struktur "SM_TYPES" (Bibliothek "MX_Base") verwendet. Je nach Format des konfigurierten Parameters wird das entsprechende Element aus "SM_TYPES" verwendet. Dort gibt es für den entsprechenden Datentyp des Parameters unterschiedliche Komponenten:

- Parameter in Festkommaformat und Nachkommastellen $\neq 0$ liegen in der Komponente "REAL_" im gewichteten Format.
- Vorzeichenlose 16-Bit- bzw. 32-Bit-Parameter liegen in der Komponente "UINT_" bzw. "UDINT_".
- 16-Bit- bzw. 32-Bit-Parameter mit Vorzeichen liegen in "INT_" bzw. "DINT_".

Struktur "SM_TYPES":

Programm:

```

TYPE SM_TYPES :
STRUCT
  REAL_ :REAL;      (* REAL (32-Bit) *)
  DINT_ :DINT;      (* DINT (32-Bit) *)
  UDINT_ :UDINT;    (* UDINT (32-Bit) *)
  UINT_ :UINT;      (* USINT (16-Bit) *)
  INT_ :INT;        (* SINT (16-Bit) *)
END_STRUCT
END_TYPE

```

Feste Inhalte

Feste Istwerte In der Achsdatenstruktur "AxisData" stehen einige wichtige Achsdaten als REAL-Werte in gewichteter Darstellung fest zur Verfügung.

Elementname	Datentyp	Richtung	Parameter in MLD Master	Parameter in Achse	Bedeutung
rActualPosition_i	REAL	Istwert	P-0-167x	S-0-0386	Aktiver Lageistwert
rActualVelocity_i	REAL	Istwert	P-0-168x	S-0-0040	Geschwindigkeits-Istwert
rActualTorqueForce_i	REAL	Istwert	P-0-169x	S-0-0084	Drehmoment-/Kraft-Istwert

Elementname	IEC-Name des Datenelements. Mit diesem Namen wird das Strukturelement im Programm angesprochen
Datentyp	IEC-Datentyp des Datenelements
Richtung	Sollwerte werden vom Anwender geschrieben und vom System übertragen
Parameter in MLD Master	Im Master existiert für dieses Element ein Diagnose-Parameter. Dieser zeigt den Wert an, ist aber nicht beschreibbar
Parameter in Achse	Quell- bzw. Zielparameter auf der jeweiligen Achse. Hier kann ein entsprechender fester Parameter stehen. Bei den konfigurierbaren Elementen ist der Parameter einstellbar. Bei den anderen Elementen gibt es keinen Parameter auf der Achse.

Achsstatusinformationen • wDriveStatus_i

Der Istwert "wDriveStatus_i" enthält die wesentlichen Informationen aus dem Parameter "S-0-0135, Antriebs-Status" bzw. aus dem Parameter "P-0-0115, Gerätesteuerung: Statuswort".

Die folgende Tabelle zeigt die vorhandenen Bits.

Bit	wDriveStatus_i	S-0-0135 / P-0-0115
2-0	nicht verwendet	Steuerinformation für Service-Kanal
3	Status der Sollwertverarbeitung	Status der Sollwertverarbeitung
4	nicht verwendet	--
5	Sercos III Status Regelungsgeber	Sercos III Status Regelungsgeber
5	nicht Führungskommunikation Sercos III Änderungsbit Kommandostatus	nicht Führungskommunikation Sercos III Änderungsbit Kommandostatus
7/6	nicht verwendet	Echtzeitstatusbits 1 und 2

Bit	wDriveStatus_i	S-0-0135 /P-0-0115
10-8	Betriebsart	Ist-Betriebsart 000: Hauptbetriebsart aktiv 001: Nebenbetriebsart 1 aktiv 010: Nebenbetriebsart 2 aktiv, usw.
11	nicht verwendet	--
12	nicht verwendet	--
13	Antriebsfehler	Antriebsfehler, Fehler in Zustandsklasse 1
15/14	Betriebsbereit 00: Antrieb nicht bereit zur Leistungszuschaltung, da interne Überprüfungen nicht positiv abgeschlossen sind 01: Bereit zur Leistungszuschaltung 10: Steuer- und Leistungsteil betriebsbereit, momentanfrei 11: In Betrieb, momentenbehaftet	Betriebsbereit 00: Antrieb nicht bereit zur Leistungszuschaltung, da interne Überprüfungen nicht positiv abgeschlossen sind 01: Bereit zur Leistungszuschaltung 10: Steuer- und Leistungsteil betriebsbereit, momentanfrei 11: In Betrieb, momentenbehaftet

Tab. 5-3: Aufbau des "wDriveStatus_i"

- **wDriveExtStatus_i**

Der Istwert "wDriveExtStatus_i" enthält Bitkombinationen aus den Parametern "S-0-0135, Antriebs-Status" und "S-0-0134, Master-Steuerwort" bzw. aus den Parametern "P-0-0115, Gerätesteuerung: Statuswort" und "P-0-0116, Gerätesteuerung: Steuerwort".

Die folgende Tabelle zeigt die vorhandenen Bits.

Bit	dwDriveExtStatus_i	
10-0	nicht verwendet	-
11	Motion-Kommandierung ist unterbrochen ["Axis_Interrupted"] ^{*1)}	P-0-166x (Bit 15 = 1 & Bit 14 = 1 & Bit13 = 0 & Bit 3 = 0) und P-0-165x (Bit 13 = 1)
12	Antrieb folgt der Sollwertvorgabe durch die MLD ["Axis_FollowingCommand"]	P-0-166x: permanente Kontrolle, Bit 3 (Status der externen Sollwertvorgabe) temporäre Kontrolle (nur lokale Achse); Bit 6 von P-0-0115 (Status der internen Sollwertvorgabe)
13	Steuerteil betriebsbereit ["Axis_Inbb"]	P-0-166x: Bit 15 Bit 14: Steuerteil betriebsbereit
14	Steuer- und Leistungsteil betriebsbereit ["Axis_InAb"]	P-0-166x: Bit 15=1: Steuer- und Leistungsteil betriebsbereit
15	Antrieb momentenbehaftet ["Axis_Power"]	P-0-166x: Bit 15=1 und Bit 14=1: Antrieb momentenbehaftet

Tab. 5-4: Aufbau des "dwDriveExtStatus_i"

***1) Bitzugriff "Axis_Interrupted"**

Die IndraDrive-Firmware stellt übergeordnete Antriebsfunktionen zur Verfügung, welche eine Motion-Kommandierung durch eine übergeordnete SPS unterbrechen können, aber die Kommandierung in der Regel nicht abbrechen.

Der Bitzugriff "Axis_Interrupted" soll helfen Unterbrechungen von Motion-Kommandierungen leicht zu erkennen, um anschließend die gewünschte Reaktion durchführen zu können. Die Funktionsbausteine "MC_Stop" und "MB_Stop" sind davon nicht betroffen.

Beispiel:

Beispiele für übergeordnete Antriebsfunktionen

- Schnellhalt Messtaster
- SBH (Sicherer Betriebshalt)
- – E8029 Lagegrenzwert positiv überschritten
- – E8030 Lagegrenzwert negativ überschritten
- – E8043 Fahrbereichsendschalter positiv betätigt
- – E8044 Fahrbereichsendschalter negativ betätigt
- Antriebsgeführtes Positionieren
- E2053 Zielposition außerhalb des Verfahrbereichs
- Antriebssteuerkommandos
- ...

Erkennt der aktive Motion-Funktionsbaustein (DiscreteMotion, ContinuousMotion bzw. SynchronizedMotion) auf dem Zielsystem MLD, dass die Motion-Kommandierung durch eine übergeordnete Antriebsfunktion unterbrochen wurde, bricht der Funktionsbaustein seine Abarbeitung mit "Command-Aborted"=TRUE ab.

- Nach Beendigung der übergeordneten Antriebsfunktion bleibt der Motion-Funktionsbaustein im Zustand "CommandAborted"
- Nach Beendigung der übergeordneten Antriebsfunktion wird die letzte kommandierte Betriebsart wieder aktiv, falls nicht zwischenzeitlich eine neue Betriebsart aktiviert wurde:
 - "DiscreteMotion"-Betriebsarten fahren die letzte Zielposition nicht mehr an
 - "ContinuousMotion"- und "SynchronizedMotion"-Betriebsarten folgen wieder der Sollwertvorgabe
- **wSignalStatus_i**
Der Istwert "wSignalStatus_i" entspricht dem Parameter "S-0-0144, Signal-Statuswort" der spezifizierten Achse.
- **wSyncStatus_i**
Der Istwert "wSyncStatus" entspricht dem Parameter "P-0-0089, Statuswort Synchronbetriebsarten" der spezifizierten Achse. Der Parameter liefert wertvolle Statusinformationen bzgl. der Ausführung der Synchronisationsbetriebsarten (Gleichlauf-Funktionsbausteine wie "MC_GearIn", "MC_CamIn", "MB_GearInPos", "MB_MotionProfile" usw.).
- **dwPLCopenStatus**
Die Bits der Achsstatusinformation "dwPLCopenStatus" stellen den Zustand der "state machine" nach PLCopen dar (siehe dazu Abbildung ["5-11: Motion State Diagramm"](#)). Es kann hier immer nur ein Bit aktiv sein, welches den aktuellen Zustand anzeigt. Die einzelnen Bits stehen als feste Zustandsbits in der Achsdatenstruktur "AxisData" zur Verfügung (siehe dazu ["5-2: Aufbau der AxisData"](#)).
- **dwDiagNumber_i**

Der Istwert "dwDiagNumber_i" entspricht dem Parameter "S-0-0390, Diagnose Nummer" der spezifizierten Achse. Der Parameter liefert die aktuelle Antriebsdiagnose.

- **dwDeviceStatus_i**

Der Istwert "dwDeviceStatus_i" enthält die wesentlichen Informationen aus dem Parameter "S-0-1045, Sercos III: Device-Status (S-Dev)"

Feste Zustandsbits

Es sind einige wichtige Achsstatus sowie alle PLCopen-Status als BOOL-Werte verfügbar (Siehe dazu "[5-2: Aufbau der AxisData](#)").

Daten der Achsdatenstruktur "AxisData"

Hinweise zur Verwendung der Achsdatenstruktur "AxisData"



Bei einer selbst programmierten Fehlerreaktion, welche durch einen F4-Kommunikationsfehler ausgelöst wurde, darf die Achsdatenstruktur "AxisData" nicht verwendet werden.



FKM-synchrone-Motiontask: Die Sollwerte der Achsdatenstruktur "AxisData" werden auch dann geschrieben, wenn die FKM-synchrone-Motiontask innerhalb des NC-Zyklus nicht fertig abgearbeitet wurde. In diesem Fall können die Sollwerte inkonsistent sein.

CCD-synchrone-Motiontask: Die Sollwerte der Achsdatenstruktur "AxisData" werden auch dann geschrieben, wenn die CCD-synchrone-Motiontask innerhalb des CCD-Zyklus nicht fertig abgearbeitet wurde. In diesem Fall können die Sollwerte inkonsistent sein.



Die Daten der Achsdatenstruktur "AxisData" stehen nur komplett zur Verfügung, wenn sich der Antrieb in Phase4 befindet. Ansonsten wird nur das Element "Axis_Error" zur Fehlerdiagnose gesetzt, sofern ein Fehler an der entsprechenden Achse ansteht.



Bei Verwendung einer FKM-synchronen-Motiontask in Verbindung mit dem MLD-M-Systemmodus ist folgendes zu beachten: Da die FKM-synchrone-Motiontask synchron zum NC-Zyklus läuft und in diesem Fall nicht synchron zum CCD-Zyklus, aber die Achsdatenstruktur "AxisData" synchron zum CCD-Zyklus verarbeitet wird, ist kein konsistenter Datenzugriff auf die Achsdatenstruktur "AxisData" möglich.

Gültigkeit und Konsistenz der Daten der Achsdatenstruktur "AxisData"

Die Daten (Soll- und Istwerte) liegen jeweils zentral in einem globalen Array.

FKM-synchrone Motiontask ohne MLD-M-Systemmodus

Bei Verwendung einer **FKM-synchronen Motiontask** (siehe dazu Kapitel "Grundfunktionen von Rexroth IndraMotion MLD", Abschnitt "[Tasksystem](#)") wird die Achsdatenstruktur "AxisData" synchron zum NC-Takt und somit synchron zur FKM-synchronen Motiontask verarbeitet; d. h. die Istwerte werden im Anschluss an den Zeitpunkt T4 aktualisiert [T4 ist der Zeitpunkt an dem die Istwerte für das Antriebstelegramm (AT) gelatched werden]. Somit stehen die Istwerte zum Motiontask-Beginn zur Verfügung.

Die Sollwerte der Achsdatenstruktur "AxisData" werden nach Taskende direkt vor der Masterdatentelegramm-Auswertung geschrieben; d.h. die Sollwerte aus der Achsdatenstruktur "AxisData" werden zum gleichen Zeit-

punkt gültig wie die Sollwerte aus dem MDT vom evtl. übergeordneten synchronisierfähigen Sercos III-Master. Ist der Antrieb mit einer anderen Führungskommunikationen als Sercos III verbunden, dann synchronisiert sich der Antrieb selbst im NC-Zyklusintervall. Dabei wird die NC-Zykluszeit über den Parameter S-0-0001 festgelegt. An der Bildung der Achsdatenstruktur ändert sich dadurch nichts. Innerhalb der FKM-synchronen-Motiontask stehen somit konsistente Daten aus der Achsdatenstruktur "AxisData" zur Verfügung.

Wird noch in weiteren Tasks auf die Achsdatenstruktur zugegriffen, sollten - wenn möglich - am Taskanfang die Ist- und Sollwerte gelesen bzw. geschrieben werden. Eine Konsistenz der Daten der "AxisData"-Struktur kann hier nicht gewährleistet werden.

Keine FKM-synchrone-Motiontask

Wird keine FKM-synchrone-Motiontask verwendet, wird die Achsdatenstruktur "AxisData" am Anfang und am Ende einer SPS-Zeitscheibe (siehe dazu Kapitel "Grundfunktionen von Rexroth IndraMotion MLD", Abschnitt "[Task-system](#)") verarbeitet; d.h. die Istwerte werden direkt vor Beginn einer neuen Zeitscheibe aktualisiert und die Sollwerte werden direkt nach dem Ende der SPS-Zeitscheibe geschrieben. Konsistente Datenzugriffe über die Achsdatenstruktur "AxisData" können nur erreicht werden, wenn die "AxisData"-Struktur in einer zyklischen Task mit der kleinsten Periodendauer verwendet wird, in allen anderen Fällen nicht.

FKM-synchrone Motiontask mit MLD-M-Systemmodus

Bei Verwendung einer **FKM-synchronen Motiontask** (siehe dazu Kapitel "Grundfunktionen von Rexroth IndraMotion MLD", Abschnitt "[Tasksystem](#)") **mit aktivem MLD-M-Systemmodus** wird die Achsdatenstruktur "AxisData" auf den CCD-Zyklus synchronisiert; d.h. die Istwerte der Achsdatenstruktur "AxisData" werden nach jeder CCD-AT-Verarbeitung aktualisiert. Die Istwerte der Achsdatenstruktur "AxisData" werden also aktualisiert, wenn alle Istwerte der CCD-Slaveachsen, welche über das jeweilige Antriebstelegramm übertragen wurden, der CCD-Masterachse bekannt sind.

Die Sollwerte der Achsdatenstruktur "AxisData" werden direkt vor dem Zeitpunkt geschrieben, wo das Masterdatentelegramm (MDT¹⁾) an alle CCD-Slaveachsen übertragen wird und sind somit Teil des MDT.

CCD-synchrone-Motiontask

Die CCD-synchrone-Motiontask (siehe dazu Kapitel "Grundfunktionen von Rexroth IndraMotion MLD", Abschnitt "[Tasksystem](#)") läuft synchron zum CCD-Zyklus; d.h. die Motiontask beginnt nach der CCD-AT-Verarbeitung und endet vor der nächsten CCD-MDT-Verarbeitung. Innerhalb der CCD-synchronen-Motiontask kann konsistent auf alle Daten der Achsdatenstruktur "AxisData" zugegriffen werden. Wird noch in weiteren Tasks auf die Achsdatenstruktur zugegriffen, sollten - wenn möglich - am Taskanfang die Ist- und Sollwerte gelesen bzw. geschrieben werden. Eine Konsistenz der Daten der "AxisData"-Struktur kann hier nicht gewährleistet werden.

Keine CCD-synchrone-Motiontask

Wird keine CCD-synchrone-Motiontask verwendet, kann ein konsistenter Datenzugriff auf die Achsdatenstruktur "AxisData" nicht gewährleistet werden. Hier empfiehlt es sich am Taskanfang die Ist- und Sollwerte zu lesen bzw. zu schreiben.



Eine Konsistenz der einzelnen Strukturelemente wird durch die Hardwarearchitektur erreicht (max. 32Bit).

Zugriff auf Elemente in "AxisData"

Hier werden einige Codebeispiele für den Zugriff auf die Elemente von "AxisData" gezeigt.

Deklaration in der "MX_Base":

¹⁾ Sollwerte für die einzelnen CCD-Slaves

Programm:

```
AxisData : ARRAY[1..10] OF
MX_AXISDATA;
```

Beispiele für die Verwendung im SPS-Programm:

Programm:

```
bMyStandstill :=
AxisData[MyAxis.AxisNo].Axis_Standstill;
AxisData[MyAxis.AxisNo].dwUserCmdDataA_q.REAL :=
rMyValue;
rTorque :=
AxisData[Axis3.AxisNo].rActualTorqueForce_i;
```

Aktivierung der "AxisData"

Die Aktivierung der Achsdatenstruktur "AxisData" kann über IndraWorks menügeführt erfolgen (siehe nachfolgende Abbildung). Dazu muss im Dialog "SPS-Konfiguration" die Option "Unterstützung der AxisData-Struktur" gewählt werden (entspricht "P-0-1367, SPS Konfiguration", Bit6=1).

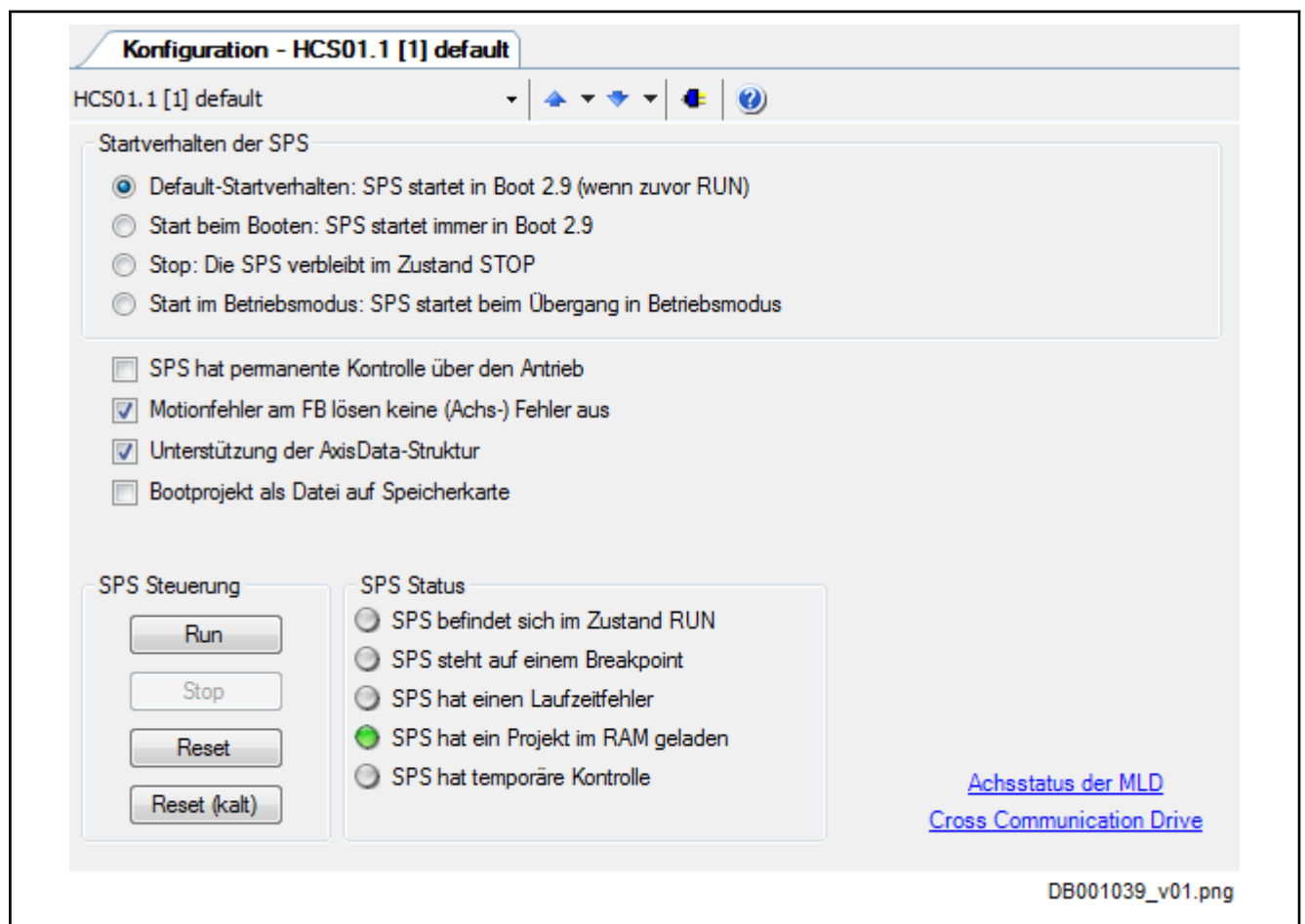


Abb. 5-6: Aktivierung der "AxisData"



Bitte beachten Sie folgende Hinweise zur Aktivierung der "Axis-Data":

- Wenn die Achsdatenstruktur "AxisData" verwendet wird, muss für die lokale Achse die "permanente Kontrolle" aktiviert sein.
- Bei realen Achsen stehen für alle Elemente der Achsdatenstruktur "AxisData" Werte zur Verfügung; für die interne virtuelle Achse stehen nur für einen Teil der Elemente Werte zur Verfügung.

Konfiguration von Anwenderdaten in "AxisData"

Wenn zusätzlich zu den vorgegebenen Istwerten auch noch anwenderdefinierte Soll- oder Istwerte übertragen werden sollen, dann muss die Konfiguration von "AxisData" in dem folgenden Dialog vorgenommen werden:

Abb. 5-7: Dialog zur Einstellung der konfigurierbaren Elemente der AxisData

Die im Dialog eingegebenen Parameter werden automatisch in die CCD-Konfiguration eingetragen. Die gesamte resultierende Liste der zyklischen Werte über CCD kann in den Dialogen der CCD-Konfiguration angezeigt werden.

5.2.2 Azyklische Datenkanäle / Schnittstellen

Einleitung

Für den azyklischen Zugriff auf Achsparameter existieren bei IndraMotion MLD folgende Alternativen:

- **Direktvariablen-Kanal**

Einfacher und schneller funktionaler Zugriff aus dem SPS-Programm auf zyklisch konfigurierbare S- und P-Parameter im Antrieb. (Es sind nur

Zugriffe auf die lokale Achse möglich. Für zyklische Zugriffe auf Parameter entfernter Achsen siehe "[Zyklische Daten im MLD-M-Systemmodus](#)".)

- **Parameterkanal** für den azyklischen Zugriff (Lesen und Schreiben) auf Achsparameter der lokalen und entfernten Achsen mittels vorgefertigter Funktionen bzw. Funktionsbausteine; ermöglicht den Zugriff auf alle S- und P-Parameter im Antrieb inklusive den "SPS-Register"-Parametern über Funktionsbausteine und Funktionen.

- **Freie SPS-Parameter (SPS-Register)**

Im SPS-Programm frei verwendbare Antriebsparameter zur Datenhaltung und/oder Kommunikation von IndraMotion MLD mit externen Geräten, Ein-/Ausgängen, Sensoren, usw. Es handelt sich hier nicht direkt um einen Datenkanal, sondern um Antriebsparameter, die zur Kommunikation oder Datenhaltung verwendet werden können.

- **Motion-Kommandokanal**

Interner Datenkanal zur Übertragung konsistenter Vorgaben von Motion-Funktionsbausteinen. Dieser Kanal ist für den Anwender nicht sichtbar bzw. nicht direkt zugreifbar, sondern wird von der Motion-Steuerung der IndraMotion MLD genutzt.

- **Socket-Kommunikation**

Freie Programmierung eines Datenkanals zur Übertragung von beliebigen Informationen über Ethernet (TCP oder UDP).

Möglichkeiten

- Zur Ankopplung eines Fremdgerätes über ein dokumentiertes Legacy-Kommunikationsprotokoll im Fremdgerät und Ansprechen des Protokolls aus der MLD.
- Freie Definition/Programmierung eines Kommunikationsprotokolls nach eigenem Bedarf im Fremdgerät und im Antriebsregelgerät.
- oder -
- Ansprechen des Antriebsregelgeräts über Sercos Internet Protocol (S/IP) zum Zugriff auf Antriebsparameter durch Implementierung der Schnittstelle im Fremdgerät.

Zugriff auf lokale Parameter über Direktvariablen

Kurzbeschreibung

Eine Möglichkeit Parameter einfach und schnell zu verarbeiten besteht durch den Zugriff über Direktvariablen. Hier kann mit einfacher Syntax, ohne Funktionsaufruf oder Funktionsbausteine und ohne den Umweg über das Prozessabbild, im SPS-Programm lesend und schreibend auf Parameter zugegriffen werden. Der SPS-Quellcode ist damit sehr einfach und übersichtlich.



Wie bei der Verarbeitung der zyklischen Daten der Führungskommunikation wird auch hier keine Speicherung und keine Grenzwertprüfung bzw. Fehlerbehandlung durchgeführt.

Merkmale

Der Zugriff über Direktvariablen ermöglicht den direkten Zugriff auf alle zyklisch konfigurierbaren Parameter im Antrieb (vgl. "S-0-0187, Liste der konfigur. Daten im zykl. Istwert-Datenkanal" und "S-0-0188, Liste der konfigur. Daten im zykl. Sollwert-Datenkanal").

Die Parameter der Ein- und Ausgänge (P-0-1390 bis P-0-1429, P-0-1440 bis P-0-1447, P-0-1450 bis P-0-1455 und P-0-1460 bis P-0-1466) sind nicht als

Direktvariable verwendbar, weil der Zugriff auf deren Werte über I/O-Zugriffe erfolgt.



Direktvariablen sind als globales Array `DV_Axis[ ]` des Funktionsbausteins "MX_DirectVarAxis" deklariert. Dies ist Bestandteil der globalen Variablenliste "GVL_Base" der Bibliothek "MX_Base".

Beispiele für einen Zugriff:

- IDN:
`DV_Axis[AXIS_1].S_0_0092`
- EIDN:
`DV_Axis[AXIS_1].S_0_1101_000_001`

Der Zugriff auf Parameter über Direktvariablen bietet die folgenden **Vorteile**:

- schneller Parameterzugriff
- einfache und übersichtliche Programmierung in SPS

Der Zugriff auf Parameter über Direktvariablen hat aber auch **Einschränkungen**:

- nur Zugriff auf zyklisch konfigurierbare Parameter
- kein Zugriff auf Parameter entfernter Achsen
- Direktvariablen können keine Auslöser für eine SPS-Eventtask sein
- kein Bitzugriff möglich (z. B. `DV_Axis[AXIS_1].P_0_0115.3`)

Beteiligte Parameter

Folgende Parameter werden im Zusammenhang mit dem Zugriff über Direktvariablen verwendet:

- S-0-0187, Liste der konfigur. Daten im zykl. Istwert-Datenkanal
- S-0-0188, Liste der konfigur. Daten im zykl. Sollwert-Datenkanal

Funktion

Verfügbare Direktvariablen

Alle zyklisch konfigurierbaren Parameter (vgl. "S-0-0187, Liste der konfigur. Daten im zykl. Istwert-Datenkanal" und "S-0-0188, Liste der konfigur. Daten im zykl. Sollwert-Datenkanal") sind als globale Direktvariablen in der "Base"-Bibliothek bereits vorhanden.

Wichtige Hinweise

Beim Zugriff über Direktvariablen ist folgendes zu beachten:

- Es ist nicht möglich eigene Direktvariablen zu definieren
- Listenparameter können nicht mit Direktvariablen adressiert werden
- Bei dem Zugriff über Direktvariablen gibt es keine Fehlerbehandlung, daher erfordern diese Zugriffe gewisse Kenntnisse bezüglich der zugehörigen Parameter (z. B. Grenzwerte, ...)

Parameterkanal

Kurzbeschreibung

Der Parameterkanal ermöglicht den azyklischen Zugriff (lesen und schreiben) auf alle S- und P-Parameter des Antriebs.



Der Zugriff auf die Parameter erfolgt direkt, d. h. die Zeitdauer für einen Lese- bzw. Schreibvorgang ist bei der lokalen Achse sehr kurz.

Zugriffe auf Parameter entfernter Achsen sind möglich und nutzen den Servicekanal des CCD in Sercos III. Dies hat dort zur Folge, dass die Bausteine mehrfach aufgerufen werden müssen und nach der Übertragung "Done" melden.

Zugriff über Funktion oder Funktionsbaustein

Beim Zugriff auf Parameter werden mehrere Möglichkeiten zur Verfügung gestellt.

- Merkmale bei dem Zugriff auf Parameter über eine Funktion:
 - schneller als ein Funktionsbaustein
 - keine Instanzen
 - keine Fehlerkontrolle
 - keine Zugriffe auf entfernte Achsen
- Merkmale bei dem Zugriff auf Parameter über einen Funktionsbaustein:
 - Übertragung mit Fehlerkontrolle
 - Instanz notwendig
 - Aktion geschieht auf steigende "Execute"-Flanke
 - Zugriff auch auf E/A-Module "Rexroth Inline" (Sercos III) möglich

Merkmale des Parameterkanals

Der Parameterkanal besitzt folgende Eigenschaften:

- Die Bausteine und Funktionen zum Bearbeiten der Parameter sind in der Bibliothek "MX_PLCOpen" enthalten.
- Bei einem Einachssystem muss immer die Achsnummer "Axis1" angegeben werden.



Als Schnittstelle von der SPS-Programmierungsumgebung zu den Sercos III-Teilnehmern steht zusätzlich die Bibliothek "RIL_SercosIII" zur Verfügung. Es können z. B. E/A-Module "Rexroth Inline" diagnostiziert werden oder azyklisch Sercos Parameter geschrieben bzw. gelesen werden.

Die Sercos III-Teilnehmer müssen hierbei über ihre Sercos Adresse angesprochen werden.

Funktionsweise

In diesem Abschnitt wird die Detailfunktion bezüglich der Zugriffe auf Antriebsparameter aus einem SPS-Programm heraus beschrieben.

Adressierung

Die Adressierung der Parameter erfolgt über Konstanten aus der Base-Bibliothek (FP_....). Die dort verwendeten Konstanten beinhalten die Sercos compatible Adresse für P-/S-Parameter. Bei den Funktionsbausteinen der Base-Bibliothek wird der Eingang "Ident" bzw. "ParameterNumber" jeweils mit diesen Konstanten versorgt.

Kommunikationsschnittstellen und Datenkanäle der MLD

The screenshot shows the IndraDrive software interface. At the top, there are tabs for 'Bibliothek hinzufügen', 'Bibliothek löschen', 'Eigenschaften', 'Details', 'Platzhalter', and 'Bibliotheksrepository'. Below these is a table listing various libraries and their versions.

Name	Namensraum	Effektive Version
BASELIB = MX_Base, 20.12.0.0 (System)	MX_Base	20.12.0.0
CHECKLIB = MX_CheckRtv, 20.1.2.0 (System)	MX_CheckRTV	20.1.2.0
IecVarAccess = IecVarAccess, 3.5.5.0 (System)	IecVarAccessLibrary	3.5.5.0
IoStandard = IoStandard, 3.5.7.0 (System)	IoStandard	3.5.7.0
PLCOPENLIB = MX_PLCopen, 20.8.0.0 (System)	MX_PLCopen	20.8.0.0
RIL_COMMONTYPES = MX_CommonTypes, 20.10.0.0 (System)	MX_CommonTypes	20.10.0.0
RIL_FIELDBUSTYPES = RIL_FieldbusTypes, 14.6.0.0 (System)	RIL_FieldbusTypes	14.6.0.0
RIL_SERCOSIII = RIL_SercosIII, 14.16.0.0 (System)	RIL_SercosIII	14.16.0.0
RIL_UTILITIES = RIL_Uilities, 14.16.0.0 (System)	RIL_Uilities	14.16.0.0
Standard = Standard, 3.5.5.0 (System)	Standard	3.5.5.0

Below the table, there is a tree view on the left showing the project structure. The 'MX_Base, 20.12.0.0 (System)' folder is selected, and the 'Parameter ID' folder is expanded, showing 'GVL_P_Param_ID' selected.

On the right, the 'Eingänge/Ausgänge' tab is active, showing a table of parameters:

Name	Datentyp	Geerbt von	Adresse	Initialwert	Kommentar
FP_P_0_0001	MB_IDN			32769	Switching frequency of
FP_P_0_0002	MB_IDN			32770	SPI flash aging counter
FP_P_0_0003	MB_IDN			32771	Status of parameter bu
FP_P_0_0004	MB_IDN			32772	Velocity loop smoothing
FP_P_0_0005	MB_IDN			32773	SPI flash aging counter
FP_P_0_0006	MB_IDN			32774	Diagnostic message cor
FP_P_0_0008	MB_IDN			32776	Activation E-Stop functi
FP_P_0_0010	MB_IDN			32778	Excessive position comm
FP_P_0_0011	MB_IDN			32779	Last valid position comm
FP_P_0_0013	MB_IDN			32781	Last valid position comm

The bottom right corner of the screenshot shows the text 'DB001022_V02'.

Abb. 5-8: Adressierung der Parameter über Konstanten



Der Eingang akzeptiert aufgrund seiner Breite von 16 Bit keine weiteren Parametertypen (wie "A" oder "Y"). Somit wird der Wertebereich automatisch gegen falsche Werte gesichert.

Parametrier- und Programmierhinweise

Read-/Write-Parameter in DINT

Mit den Funktionsbausteinen "MX_ReadParamDINT"/"MX_WriteParamDINT" werden 16-Bit und 32-Bit Parameter im Sercos Wichtungsformat transferiert. (Beispiel: bei Standardwichtung entspricht 1 Grad dem Wert 10000). Die Werte werden also beim Lesen als vorzeichenbehafteter 32-Bit -Wert "DINT" geliefert.



Ist ein SPS-Register-Parameter (z. B. P-0-1370) auf das Datenformat FLOAT eingestellt, muss der Funktionsbaustein "MB_ReadRealParameter"/"MB_WriteRealParameter" verwendet werden, um den Wert (Datum) des Parameters zu lesen/zuschreiben.



Bei den Bausteinen, die auf Parameter zugreifen, wurde ab MPx06 der Eingang "ParameterNumber" von "DINT" auf "MB_IDN" umgestellt. "MB_IDN" ist bei allen Varianten von IndraMotion verfügbar. Bei "IndraMotion MLD" entspricht "MB_IDN" der Sercos III-Definition und hat somit eine Datenbreite von 32 Bit. An dem Eingang "ParameterNumber" können sowohl Konstanten des Typs "DINT" als auch "MB_IDN" verwendet werden. SPS-Programme, die vor der MPx06 erstellt wurden, sind weiterhin lauffähig.

Read-/Write-Textparameter

Zur Verarbeitung von Textparametern können die Funktionen "MX_fReadStringParam"/"MX_fWriteStringParam" verwendet werden.

Mit dem Eingang "WriteBuffered" kann der jeweilige Parameter speichernd geschrieben werden.

Parameterkanal-Bibliotheken

Nachfolgend wird ein Beispiel aus der "PLCopen"-Bibliothek aufgeführt:

Name	Namensraum	Effektive Version
BASELIB = MX_Base, 20.12.0.0 (System)	MX_Base	20.12.0.0
CHECKLIB = MX_CheckRtv, 20.1.2.0 (System)	MX_CheckRTV	20.1.2.0
IecVarAccess = IecVarAccess, 3.5.5.0 (System)	IecVarAccessLibrary	3.5.5.0
IoStandard = IoStandard, 3.5.7.0 (System)	IoStandard	3.5.7.0
PLCOPENLIB = MX_PLcOpen, 20.8.0.0 (System)	MX_PLcOpen	20.8.0.0
BASELIB = MX_Base, 20.12.0.0 (System)	MX_Base	20.12.0.0
RIL_COMMONTYPES = MX_CommonTypes, 20.10.0.0 (System)	MX_CommonTypes	20.10.0.0
Standard = Standard, 3.5.5.0 (System)	Standard	3.5.5.0
RIL_COMMONTYPES = MX_CommonTypes, 20.10.0.0 (System)	MX_CommonTypes	20.10.0.0
RIL_FIELDBUSTYPES = RIL_FieldbusTypes, 14.6.0.0 (System)	RIL_FieldbusTypes	14.6.0.0

MX_PLcOpen, 20.8.0.0 (System)		
Enable	BOOL	BOOL Done
ParameterNumber	MB_IDN	BOOL Active
Axis	AXIS_REF	BOOL Error
		ERROR_CODE ErrorID
		ERROR_STRUCT ErrorIdent
		DINT Value

Abb. 5-9: Funktionen und Funktionsbausteine für den Zugriff auf Parameter

Freie SPS-Parameter (globale Register)

Kurzbeschreibung

Der Zugriff über "freie Parameter" (globale Register) kann zum Datenaustausch der MLD mit einem benachbarten Antrieb (ab MP*04VRS durch CCD)

bzw. einer übergeordneten Steuerung oder einem externen Bedienterminal (z. B. BTV) verwendet werden.



Die globalen Register haben keinen direkten Einfluss auf den Antrieb, sondern wirken nur im Zusammenhang mit der MLD.

Merkmale Das Anzeigeformat der globalen Register lässt sich frei definieren. Es stehen gepufferte und ungepufferte Register zur Verfügung. Zur Übergabe großer Datenmengen sind Register mit Listenstruktur vorhanden.

- 32 ungepufferte globale Register zur Parametrierung von SPS-Funktionen bzw. Bausteinen
- 32 gepufferte globale Register zur spannungsausfallsicheren Konfiguration von SPS-Funktionen bzw. Bausteinen
- 2 globale Textregister als frei verwendbare Textparameter zur Anzeige von Diagnosetexten
- 3 gepufferte Listenregister mit 1024 4-Byte Werten.
- 1 ungepuffertes Listenregister mit 8192 4-Byte Werten

Zugehörige Parameter Die Register werden durch die folgenden Parameter repräsentiert:

- **Globale SPS-Register, ungepuffert:**
 - P-0-1270, SPS Globales Register A0
 - P-0-1271, SPS Globales Register A1
 - P-0-1272, SPS Globales Register A2
 - ...
 - P-0-1301, SPS Globales Register A31
- **Globale Textregister, ungepuffert:**
 - P-0-1387, SPS Globales Textregister AT0
 - P-0-1388, SPS Globales Textregister AT1
- **Globales Listenregister, ungepuffert:**
 - P-0-1368, SPS Globales Register AL0
- **Globale SPS-Register, gepuffert:**
 - P-0-1370, SPS Globales Register G0
 - P-0-1371, SPS Globales Register G1
 - P-0-1372, SPS Globales Register G2
 - ...
 - P-0-1385, SPS Globales Register G15
 - P-0-1316, SPS Globales Register G16
 - P-0-1317, SPS Globales Register G17
 - P-0-1318, SPS Globales Register G18
 - ...
 - P-0-1331, SPS Globales Register G31
- **Globale Listenregister, gepuffert:**
 - P-0-1389, SPS Globales Register GL0
 - P-0-1311, SPS Globales Register GL1
 - P-0-1312, SPS Globales Register GL2
- **Zur Konfiguration des Anzeigeformats der Register:**

- P-0-1386, SPS Anzeigeformat Globaler Register

Funktion

Formatanpassung für globale Register

Die Anzeigeformate der globalen Register können mit "P-0-1386, SPS Anzeigeformat Globaler Register" individuell angepasst werden. Es können Anzeigeformat (z. B. BIN, Dez, Hex,...) und die Anzahl der Nachkommastellen definiert werden.

Der Inhalt der globalen Register "Gxx" und "GLx" wird bei Spannungsausfall gepuffert, d. h. es erfolgt eine residente Speicherung der Registerinhalte, so dass die Parameterinhalte bei einem Spannungsausfall nicht verlorengehen.

Anwendungsfälle für globale Register Gxx

Die globalen Register G0 .. G31 bzw. GL0 bis GL2 lassen sich für folgende Anwendungsfälle einsetzen:

- Konfiguration von SPS-Funktionen bzw. -Bausteinen
- Kommunikation mit der externen Steuerung über die Führungskommunikations-Schnittstelle



Bei Verwendung der globalen Register als Sollwerte (Sollwerte von der übergeordneten Steuerung) ist zu beachten, dass bei einem Ausfall der Kommunikation die Werte auf "0" gesetzt werden (wie bei Eingängen üblich).

Anwendungsfälle für globale Register Axx

- Nutzung als resistenter (permanenter) Speicher für die MLD-S, da die Inhalte bei einem Spannungsausfall erhalten bleiben

Die globalen Register A0 .. A31 bzw. AL0 lassen sich für folgende Anwendungsfälle einsetzen:

- Online-Parametrierung von SPS-Funktionen bzw. -Bausteinen
- Kommunikation mit der externen Steuerung über die Führungskommunikations-Schnittstelle

Anwendungsfälle für globale Register ATx

Die globalen Textregister "P-0-1387, SPS Globales Register AT0" und "P-0-1387, SPS Register AT1" stehen als frei verwendbare Textparameter mit maximal 255 Zeichen plus abschließendem "0"-Zeichen zur Verfügung.

Die globalen Register (AT0 und AT1) lassen sich für folgende Anwendungsfälle einsetzen:

- Kommunikation mit übergeordneter Steuerung bzw. HMI
- Definition von frei definierbaren Diagnosetexten

Parametrier- und Inbetriebnahmehinweise

Das Anzeigeformat für die freien Parameter im Parameterkanal kann mit "P-0-1386, SPS Anzeigeformat Globaler Register" definiert werden. P-0-1386 ist ein Listenparameter mit 68 Elementen und folgender Zuordnung:

Element-Nr.	Beeinflusster Parameter
1..16	P-0-1370 bis P-0-1385 (Globales Register G0 bis G15)
17	P-0-1389, Globales Register GL0
18	P-0-1368, Globales Register AL0
19..34	P-0-1316 bis P-0-1331 (Globales Register G16 bis G31)
35..66	P-0-1270 bis P-0-1301 (Globales Register A0 bis A31)
67..68	P-0-1311 bis P-0-1312 (Globales Register GL1 bis GL2)

Tab. 5-5: P-0-1386, SPS Anzeigeformat Globaler Register

Für die freien Parameter können Grenzwerte, Name und Einheit über das SPS-Programm frei definiert werden. Dazu dienen die folgenden Funktionen aus der Bibliothek "MX_Base":

- MX_fSetParamLimits
- MX_fSetParamName
- MX_fSetParamUnit

Motion-Kommandokanal

Kurzbeschreibung

Bei dem Motion-Kommandokanal handelt es sich um einen internen Datenkanal zur Übertragung konsistenter Vorgaben von fertigen Motion-Funktionsbausteinen. Diese fertigen Motion-Funktionsbausteine können direkt aus dem SPS-Programm aufgerufen werden. Der Motion-Kommandokanal wird intern vor allem zur Realisierung von "Motion Control"-Aufgaben bzw. für Technologiefunktionen verwendet. Die verfügbaren Motion-Funktionsbausteine sind PLCopen-konform; d.h. mit den Motion-Funktionsbausteinen erstellte Programme können auf andere Zielsysteme übertragen werden.

Merkmale	• Die Motion-Funktionsbausteine besitzen eine logische Achsadresse zur Auswahl der Achse. Hiermit können im Einachssystem MLD-S die lokale Achse und der Leitachsgenerator angesteuert werden. • Die wichtigsten Vorgaben bei der Bewegungssteuerung erfolgen über entsprechende Eingänge an den Motion-Funktionsbausteinen. Weitere spezielle Einstellungen wie Wichtung oder Ruck etc. müssen parametrisiert werden. Dies geschieht i.d.R. bei der Inbetriebnahme der Achsen mit der Inbetriebnahmesoftware IndraWorks. Es ist jedoch auch möglich mittels Parameterzugriff weitere Einstellungen zu verändern (z.B. den Ruck).
Aktivierung	Für die Bewegungssteuerung der MLD ist bei der lokalen Achse "Permanente Kontrolle" zu aktivieren. Damit folgt die lokale Achse den Vorgaben der MLD und kann mit den Motion-Funktionsbausteinen gesteuert werden.

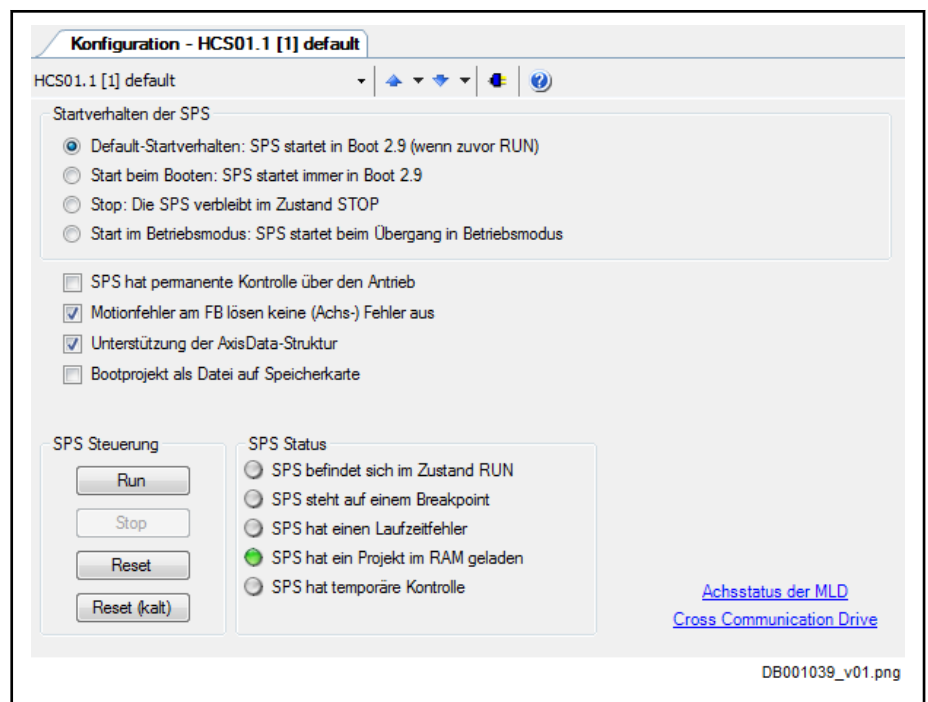


Abb. 5-10: SPS-Konfiguration; "Permanente Kontrolle"

PLCopen

Die Motion-Steuerung über das SPS-Programm wird mittels Funktionsbausteinen nach PLCopen durchgeführt. Die PLCopen hat hierfür eine Reihe von IEC61131-Bausteine definiert mit deren Hilfe man die Achsen ansteuern kann. Neben den durch die PLCopen bereits definierten Bausteine existieren noch zusätzliche an die Norm angelehnte Bausteine, die weitere Funktionalität unserer Antriebe anbieten.

Nach PLCopen verhalten sich alle Achsen nach einem definierten Statusmodell (PLCopen state machine). Rexroth unterstützt dieses Statusmodell; für weitergehende Funktionalitäten wurden zusätzliche Status hinzugefügt (z. B. "PreSetMode").

Der aktuelle Status nach PLCopen wird in "AxisData" über Statusbits dargestellt.

Kommunikationsschnittstellen und Datenkanäle der MLD

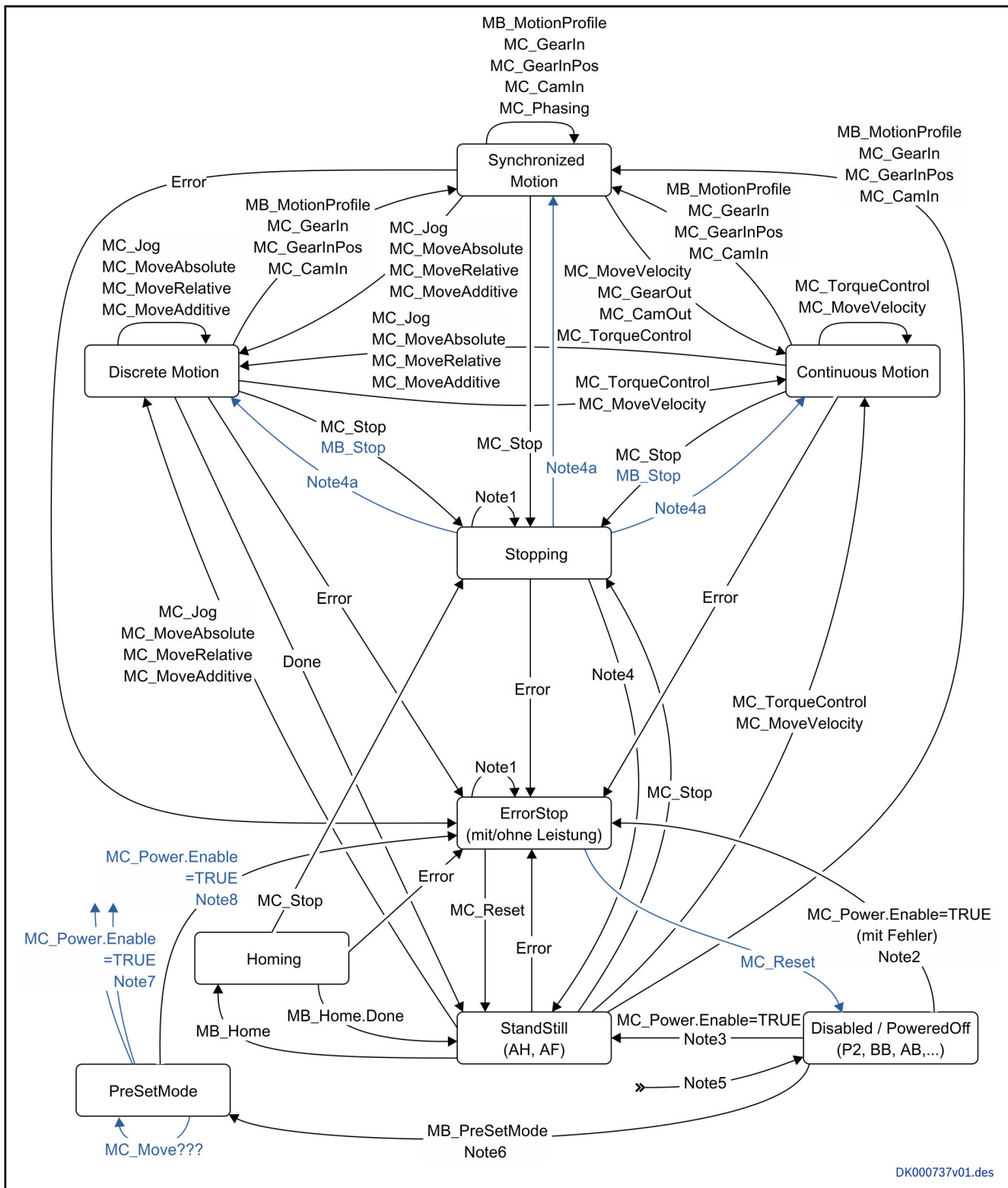
**Note x**

Abb. 5-11:

Erläuterung der Notes: siehe nachfolgende Tabelle
Motion State Diagramm

Note 1	In diesem Status ("ErrorStop" oder "Stopping") können alle Funktionsbausteine aufgerufen werden, obwohl sie nicht ausgeführt werden; Ausnahmen sind "MC_Reset" und "MB_ClearAllError", sie bewirken jeweils den Übergang zu "StandStill" oder "ErrorStop"
Note 2	Power.Enable = TRUE und ein Fehler liegt in der Achse vor
Note 3	Power.Enable = TRUE und kein Fehler liegt in der Achse vor (Standard ohne MB_PresetMode)
Note 4	MC_Stop.Done UND NICHT MC_Stop.Execute
Note 4a	Neues Motion-Kommando während NICHT MB_Stop.Execute (ohne MC_Stop)
Note 5	Wenn in einem beliebigen Status "Enable" von "MC_Power" weggenommen wird, geht der Antrieb in "PowerOff"
Note 6	Mit "MB_PreSetMode" wird der Zustand "PreSetMode" erreicht. Dort kann kommandiert werden, ohne dass Leistung vorhanden ist. Ein anschließendes Einschalten mit "Power" führt zum Start des vorgewählten Modus
Note 7	Vom Zustand "PreSetMode" wird bei steigender Flanke an "Enable" von "MC_Power" ein vorwählbarer Modus gestartet
Note 8	Eine Flanke an "MC_Power" ohne vorige Kommandierung im Zustand "PreSetMode" führt zu einem Fehler und damit wieder zu "PowerOff"

Tab. 5-6: *Legende zu Abbildung "Motion State Diagramm"*

Programmierung	Die Programmierung erfolgt mit den Funktionsbausteinen der Bibliothek "MX_PLCCopen". Die Bausteine sind so ausgelegt, dass sie zyklisch aufgerufen werden. Sie werden i.d.R. über eine Eingangsflanke aktiviert und liefern Informationen über Statusausgänge. Die genaue Funktionsweise ist in der jeweiligen Bibliotheksdokumentation beschrieben.
Bibliotheken	Die Bibliothek "MX_PLCCopen" enthält Bausteine zur Bewegungssteuerung wie z. B. "MC_MoveAbsolute" oder "MC_Stop", sowie Bausteine zur Achskontrolle, wie z. B. "MC_Power" und "MC_Reset", um die Achse in Regelung zu bringen oder einen Fehler zu löschen.
Einstellungen	Nach der Aktivierung der permanenten Kontrolle bei der lokalen Achse, werden die notwendigen Betriebsarten der Achsen automatisch parametrisiert. Damit wird die Betriebsartenauswahl automatisch vorgenommen. Es ist im Normalfall nicht notwendig, weitere Einstellungen vorzunehmen.
Wichtung	Bei der Inbetriebnahme muss die Wichtung der Achsen in den entsprechenden Dialogen der Inbetriebnahmesoftware IndraWorks vorgenommen werden. An den Motion-Funktionsbausteinen werden dann die Vorgaben über die Bausteineingänge als physikalische Werte angegeben.
Sonstige Einstellungen	Sonstige Einstellungen, wie z.B. der Ruck können nicht an den Funktionsbausteinen programmiert werden, sondern müssen in der Inbetriebnahmesoftware IndraWorks eingestellt werden. Sollen solche Werte verändert werden, können sie über den Parameterkanal (z. B. "MB_WriteParameter") auch zur Laufzeit eingestellt werden.

Leitachsgenerator

Der Leitachsgenerator stellt sich dem Anwender bezüglich der Kommandierung als eigenständige Achse dar. Die Achse kann mit einigen Motion-Funktionsbausteinen verfahren und mit "MB_Stop" / "MC_Stop" angehalten werden.

Um den Leitachsgenerator nutzen zu können, muss er einmalig aktiviert werden.

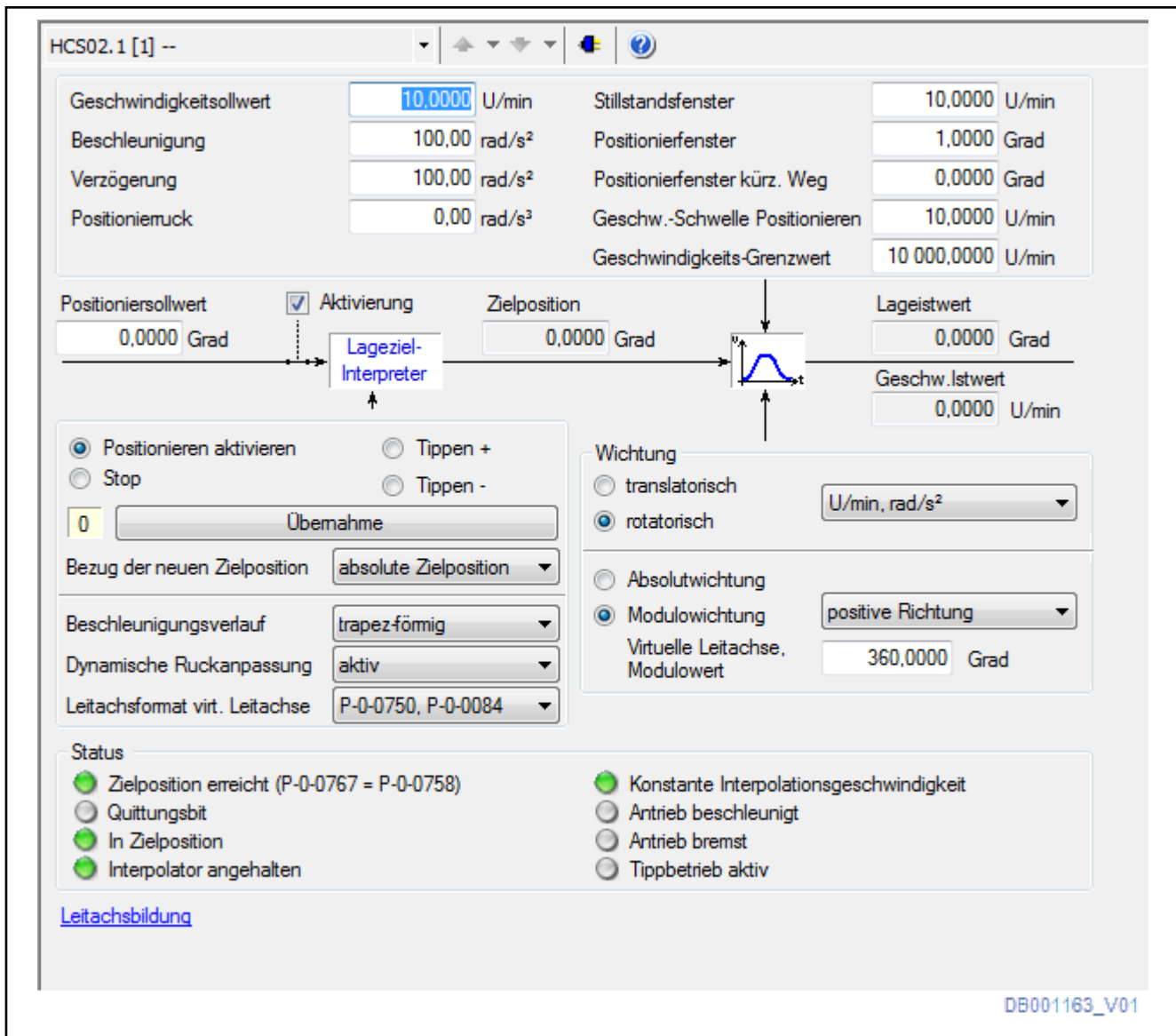


Abb. 5-12: Aktivierung des Leitachsgenerators in IndraWorks



Die virtuelle Leitachse kann über den Axis-Eingang "VmAxisint" angesteuert werden.

Mögliche Funktionsbausteine

Für die Verwendung des Leitachsgenerators stehen folgende Funktionsbausteine zur Verfügung:

- "MC_MoveRelative" oder "MX_MoveRelative"
- "MC_MoveAbsolute" oder "MX_MoveAbsolute"
- "MC_MoveAdditive" oder "MX_MoveAdditive"
- "MC_MoveVelocity"
- "MC_Stop" oder "MB_Stop"

Interne Umsetzung des Motionkanals



Die nachfolgenden Informationen über die interne Umsetzung des Motionkanals sind für die Benutzung der Bausteine im Allgemeinen nicht erforderlich. Sie liefern vielmehr Zusatzinformationen, wie die Bausteine den Antrieb mit seinen Betriebsarten und Parametern nutzen.

Mit diesen Informationen lassen sich die Vorgaben der Motion-Funktionsbausteine antriebsseitig diagnostizieren.

Benutzte Betriebsarten

Folgende Betriebsarten werden bei der Aktivierung des Motionkanals eingestellt:

Betriebsart	Einstellung
Hauptbetriebsart	Geschwindigkeitsregelung
Nebenbetriebsart 1	Drehmoment-/Kraftregelung ⁽¹⁾
Nebenbetriebsart 2	Lagesynchronisation ⁽²⁾
Nebenbetriebsart 3	Antriebsgeführtes Positionieren ⁽¹⁾
Nebenbetriebsart 4	Geschwindigkeitssynchronisation ⁽²⁾

1 nur im Grundpaket mit der Ausprägung "CL" (Closed-Loop); benötigt außerdem die Freischaltung eines beliebigen Funktionspakets

2 benötigt die Freischaltung des Funktionspakets "Synchronisation"

Tab. 5-7: Betriebsarten für die "Motion Control"-Kommandoschnittstelle



Wenn die notwendigen Paketfreischaltungen für die Betriebsarten nicht vorhanden sind, wird die betreffende Betriebsart auf "Geschwindigkeitsregelung" eingestellt. Damit sind die betroffenen Motion-Funktionsbausteine nicht lauffähig und melden einen entsprechenden Fehler.

Verarbeitung in IndraMotion MLD-S

Bei IndraMotion MLD-S werden die Vorgabeparameter für die Betriebsarten vom SPS-Motion-System direkt geschrieben. Die betroffenen Parameter sind in der Bibliotheksbeschreibung bei den Bausteinen aufgelistet. Beim Aktivieren eines Motion-Kommandos wird die zugehörige Betriebsart angewählt und alle zugehörigen Vorgabe-Parameter werden synchron an den Antrieb übergeben.

Allgemeine Verwendung

Bei der Verwendung des Motionkanals ist folgendes zu beachten:

- Achsen können aus mehreren präemptiven Tasks kommandiert werden.
- In einem Motionzyklus (z. B. in 1 ms) können alle Achsen gleichzeitig kommandiert werden.
- In einem Motionzyklus können parallele Motionvorgaben (Haupt- und Nebensollwerte) pro Achse gleichzeitig kommandiert werden (bis zu 3 Kanäle).
- Auch bei schnell hintereinander abgesetzten Kommandos wird gewährleistet, dass das letzte Kommando bearbeitet wird.
 - Mehrere Kommandos, die im gleichen Zyklus für den selben Antrieb abgesetzt wurden, werden zunächst in einem FIFO-Speicher abgelegt und anschließend vom Kommandointerpreter abgearbeitet.

tet. Es können 3 Kommandos zwischengespeichert werden. Werden zu viele Kommandos abgesetzt, und es kommt zu einem Überlauf, wird ein Fehler gemeldet

- Wenn eine Task ein Kommando absetzt, überschreibt dieses ein gegebenenfalls noch nicht vom Interpreter abgeholtes Kommando. Ein so überschriebenes Kommando meldet in seiner Instanz ein "CommandAborted".
- Wurde das Kommando von der selben Instanz ausgelöst, ist keine Meldung möglich.
- Ein neues Kommando mit einem Hauptsollwert bricht auch laufende Kommandos mit Nebensollwerten ab.
- Nebensollwerte können nur im korrekten Antriebszustand beauftragt werden: Ein neues Kommando mit einem Nebensollwert wird nur akzeptiert, wenn auch ein entsprechender Port kommandiert wurde. Es wird die entsprechende Betriebsart geprüft.
- Die virtuelle Leitachse verhält sich bezüglich der Kommandierung wie eine reale Achse, d.h. sie hat eine eigene Achsnummer!

Nachfolgend werden Hinweise zur Verwendung einiger spezieller Motion-Funktionsbausteine gegeben:

Referenzieren Der Funktionsbaustein "MB_Home" - zum Referenzieren einer Achse - kann nur gestartet werden, wenn sonst kein anderer Baustein gestartet ist. Dies muss vom SPS-Programm sichergestellt werden.

Baustein für Absolutmaß Das Setzen des Absolutmaßes ist auch im Zustand "AH" möglich. Somit kann hierzu direkt das Kommando abgesetzt werden. Bei einer laufenden Achsbewegung arbeitet das Kommando nicht. Erst wenn der Antrieb in "AH" geht oder ohne Leistung ist, kann das Kommando gestartet werden.

"MB_Stop" Bei der Verwendung des Funktionsbausteins "MB_Stop" ist folgendes zu beachten:

- Der Ansteuer-Eingang nennt sich "ExecuteLock"
- Solange "ExecuteLock"="TRUE" ist, kann kein anderer Funktionsbaustein den Antrieb in Bewegung setzen; es sollte aber auch im SPS-Programm ausgeschlossen werden, dass ein anderer Funktionsbaustein den Antrieb in Bewegung setzen kann.
- Ein "MB_Stop" kann im Zustand "Active" bei "ExecuteLock"=FALSE durch einen Motion-Funktionsbaustein abgebrochen werden.
- Der Anwender sollte nur eine Instanz des "MB_Stop" einsetzen, um das Programm übersichtlich und sicher zu gestalten.
- Bei mehreren Instanzen von "MB_Stop" setzt sich automatisch die letzte Instanz durch; die letzte Instanz hat höhere Priorität.

"MC_Stop" Bei der Verwendung des "MC_Stop" ist folgendes zu beachten:

- Solange "Execute"="TRUE" ist, kann kein anderer Funktionsbaustein den Antrieb in Bewegung setzen; es sollte aber auch im SPS-Programm ausgeschlossen werden, dass ein anderer Funktionsbaustein den Antrieb in Bewegung setzen kann.
- Der Anwender sollte nur eine Instanz des "MC_Stop" einsetzen, um das Programm übersichtlich und sicher zu gestalten.
- Bei mehreren Instanzen von "MC_Stop" setzt sich automatisch die letzte Instanz durch; die letzte Instanz hat höhere Priorität.

"MC_Power" Bei der Verwendung des "MC_Power" ist folgendes zu beachten:

- Beim Einschalten geht die Achse auf "AH" sofern vorher keine Bewegung vorkommandiert wurde (mit "MB_PreSetMode" und Motion-Funktionsbaustein).
 - Im Falle einer Vorkommandierung geht die Achse direkt in die vorgeählte Betriebsart.
- Geberumschaltung, Umschaltung Schleppfehler** Für die Geberumschaltung und die Umschaltung mit/ohne Schleppfehler verfügen die Funktionsbausteine über eigene Eingänge.

Inbetriebnahmehinweise

GEFAHR

Tödliche Verletzungen und/oder Sachschäden durch ungewollte Achsbewegungen!

⇒ Bei autarkem Betrieb (interne SPS hat die Kontrolle über den Antrieb) kann der Antrieb über die MC-Kommandoschnittstelle frei bewegt werden; daher unbedingt darauf achten, dass keine ungewollten Achsbewegungen ausgelöst werden.

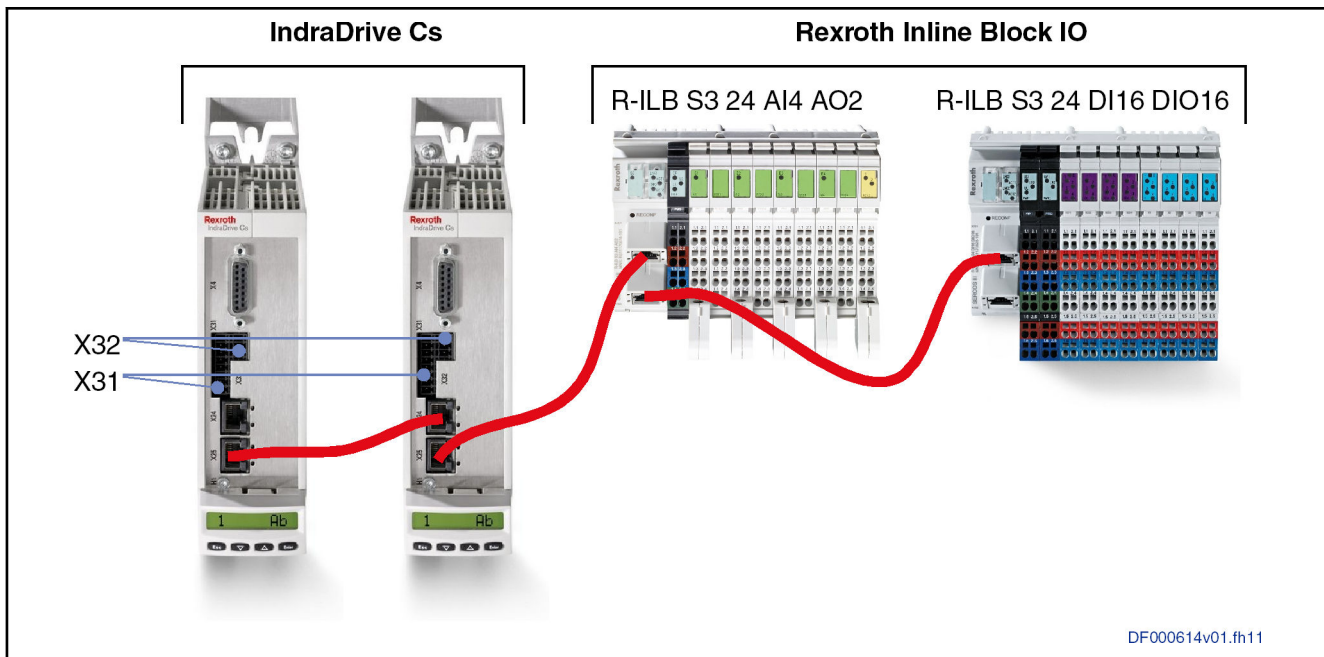
5.2.3 Konfiguration der Ein- und Ausgänge (E/A-Konfiguration der MLD)

Einleitung

Mit IndraMotion MLD kann theoretisch auf alle Ein- und Ausgänge mit Sercos Schnittstelle zugegriffen werden, z. B.:

- auf alle Ein- und Ausgänge des Steuerteils (auch auf entfernte Achsen)
- auf alle Ein- und Ausgänge des Optionsmoduls "DA"
- auf die Ein- und Ausgänge von "Rexroth Inline Block IO"
- auf die Ein- und Ausgänge von "Rexroth Inline Modular IO"

Kommunikationsschnittstellen und Datenkanäle der MLD



DF000614v01.fh11

X31

digitale Eingänge, digitaler Ausgang

X32

analoger Eingang

R-ILB S3 24 AI4 AO2

Inline Block IO mit 4 analogen Eingängen und 2 analogen Ausgängen

R-ILB S3 24 DI16 DIO16

Inline Block IO mit 16 digitalen Eingängen und 16 konfigurierbare digitale Ein- oder Ausgänge

*Abb. 5-13: Übersicht der Ein-/Ausgänge (exemplarisch für "Rexroth Inline" die Block IOs)***Konfiguration der digitalen und analogen Ein-/Ausgänge**

Für die Zuordnung der lokalen Ein- und Ausgänge zur SPS existieren in IndraWorks entsprechende Dialoge. Die Konfigurationsparameter werden dabei automatisch vom jeweiligen Dialog gefüllt. Für die Konfiguration der lokalen Ein- und Ausgänge stehen baugruppenspezifische Dialoge zur Verfügung.

In den folgenden Tabellen sind alle Ein- und Ausgänge aufgelistet.

Funktion	Hardware	Anzahl
Digitale Eingänge	Rexroth Inline Block IO, digital	min. 16 (max. 32) ^{1), 2)}
	Rexroth Inline Modular IO, digital	8 (On-Board) ³⁾ erweiterbar
Digitale Ausgänge	Rexroth Inline Block IO, digital	min. 0 (max. 16) ^{1), 2)}
	Rexroth Inline Modular IO, digital	4 (On-Board) ³⁾ erweiterbar
Analoge Eingänge	Rexroth Inline Block IO, analog	4 ²⁾
	Rexroth Inline Modular IO, analog	je nach Erweiterung ³⁾
Analoge Ausgänge	Rexroth Inline Block IO, analog	2 ²⁾
	Rexroth Inline Modular IO, analog	je nach Erweiterung ³⁾

- 1 16 konfigurierbare digitale Ein- oder Ausgänge
 2 weitere Informationen unter dem Stichwort "Rexroth Inline Block IO"
 3 weitere Informationen unter dem Stichwort "Rexroth Inline Modular IO"

Tab. 5-8: Übersicht der Ein-/Ausgänge der "Rexroth Inline Block IO" / "Rexroth Inline Modular IO"

Antriebssystem/Steuerenteil		Grundgerät (on board)		Optionsmodul "DA"		
		Standard-Eingänge (davon Messtaster-Eingänge)	Umschaltbare Ein-/Ausgänge	Eingänge	Ausgänge	Umschaltbare Ein-/Ausgänge
HCS01.1		7 (2)	1	6	6	2
HCQ02.1, HCT02.1		16 (-)	-	-	-	-
KSM02.1, KMS02.1		-	4	-	-	-
CSE02.1A, CSB02.1A		7 (2)	1	-	-	-
CDB02.1B		14 (4)	8	6	6	2
CSB02.1B, CSH02.1B		11 (2)	5	6	6	2
Digitale Eingänge	Konfigurationsparameter	P-0-0300, P-0-0301, P-0-0306, P-0-0307				
	Eingangsabbild	P-0-0303				
Digitale Ausgänge	Konfigurationsparameter	P-0-0310, P-0-0311, P-0-0312, P-0-0313, P-0-0316				
	Ausgangsabbild	P-0-0304				

Tab. 5-9: Übersicht der digitalen Ein-/Ausgänge des Antriebssystems/Steuer-teils

Kommunikationsschnittstellen und Datenkanäle der MLD

Funktion	Hardware	Anzahl	Konfigurationsparameter	Register ²⁾
Analoge Eingänge	Antriebssystem/Steuerteil	HCS01.1: 1 CSE02.1A, CSB02.1A: 1 CDB02.1B: 2 CSB02.1B, CSH02.1B: 3	P-0-0213 .. P-0-0220 P-0-0231 / P-0-0232 P-0-0255	P-0-0210 P-0-0211 P-0-0228
	Optionsmodul "DA"	2 ¹⁾	P-0-0233 P-0-0234 P-0-0255	P-0-0229 P-0-0208
Analoge Ausgänge	Antriebssystem/Steuerteil	CDB02.1B, CSB02.1B, CSH02.1B: 2	P-0-0419 / P-0-0423 / P-0-0425 P-0-0418 / P-0-0420 / P-0-0422 P-0-0427	P-0-0139 P-0-0140
	Optionsmodul "DA"	HCS01.1: 2 ¹⁾ CDB02.1B, CSB02.1B, CSH02.1B: 2	P-0-0428 / P-0-0459 / P-0-0463 P-0-0429 / P-0-0462 / P-0-0464	P-0-0414 P-0-0415

1 nicht bei HCS01 Economy
 2 Parameter zum Anzeigen / Auslesen des Analogwertes
 Tab. 5-10: Übersicht der analogen Ein-/Ausgänge

Lokale Ein-/Ausgänge (Master/Axis1)

Im Folgenden werden anhand von IndraWorks-Dialogen Konfigurationsbeispiele für die Zuweisung von Prozessabbild-Ein-/Ausgängen (SPS-Parametern) auf digitale und analoge Ein-/Ausgänge geschrieben.

Konfiguration von digitalen Ein-/Ausgängen des Steuerteils

Jeder Ein- und Ausgang des Steuerteils kann einzeln im Antrieb verwendet oder der SPS zugeordnet werden. Die Zuordnung von digitalen Ein-/Ausgängen des Steuerteils auf SPS-Parameter erfolgt durch Zuweisung der jeweils gewünschten Bits eines SPS-Parameters auf einen digitalen Ein- oder Ausgang. In IndraWorks steht dafür ein Dialog zur Verfügung. Der Dialog kann im Gerätebaum unter "Lokale E/A - X31/X32" aufgerufen werden.

Siehe auch Firmware-Funktionsbeschreibung "Digitale Ein-/Ausgänge"

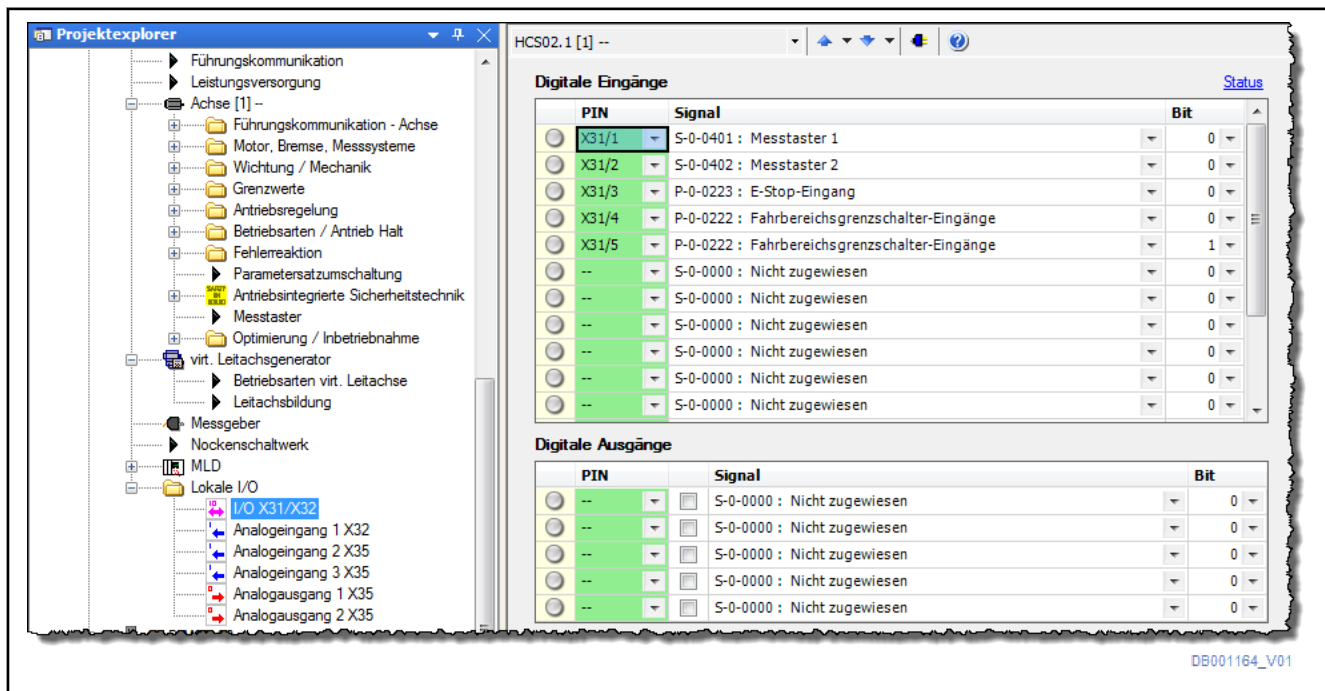


Abb. 5-14: Konfiguration der digitalen Ein-/Ausgänge in IndraWorks

Konfiguration des analogen Eingangs des Steuerteils

Zum Einlesen von analogen Spannungswerten ist die Konfiguration der jeweiligen SPS-Parameter (P-0-1390,...) für den Analogeingang erforderlich. In dem nachfolgenden Beispiel wird die Zuordnung des SPS-Parameters "P-0-1391, SPS Eingang WORD1 AT %IB2" auf den Analogeingang 1 gezeigt (siehe auch Firmware-Funktionsbeschreibung "Analoge Eingänge").

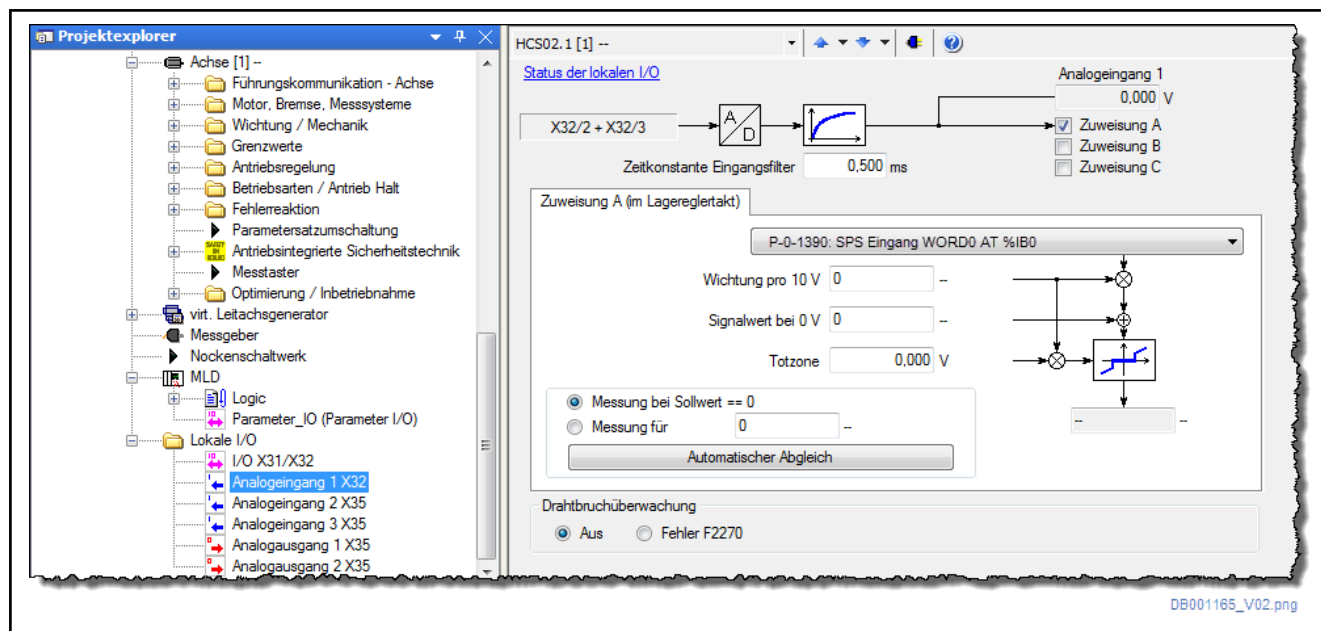


Abb. 5-15: Konfiguration eines analogen Eingangs in IndraWorks

Ein-/Ausgänge des Optionsmoduls "DA"

Der Zugriff auf die digitalen und analogen Ein-/Ausgänge erfolgt über definierte Parameter. Jeder digitale Ein-/Ausgang wird über ein Konfigurationsmenü entsprechend voreingestellt.

Ein-/Ausgänge von einer Leitsteuerung

In dem Dialog "Führungskommunikation Einstellungen" können bei den Prozessdaten auch die MLD-Prozessabbildparameter eingetragen werden, um so eine Verbindung von der Leitsteuerung in die I/O der MLD herzustellen.



MLD-Eingänge, die von einer Leitsteuerung kommen, werden bei Verbindungsunterbrechung auf Null gesetzt.

Kommunikationsschnittstellen und Datenkanäle der MLD

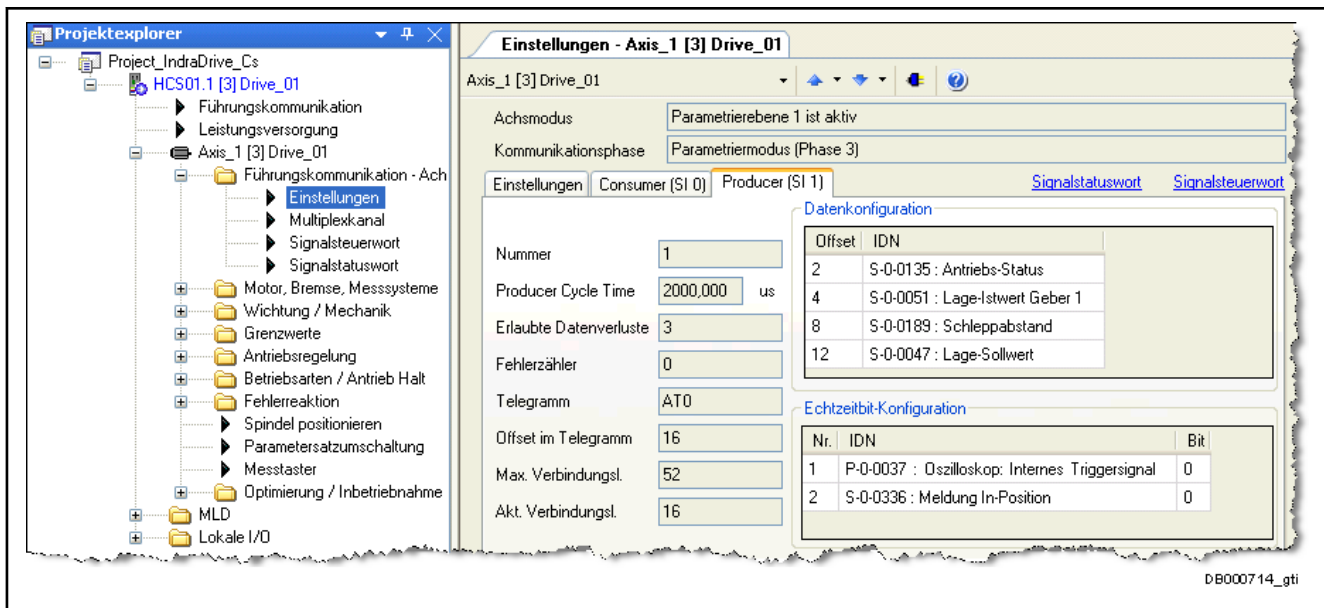


Abb. 5-16: "Führungskommunikation Einstellungen", "Prozessdaten-Ein (AT)"

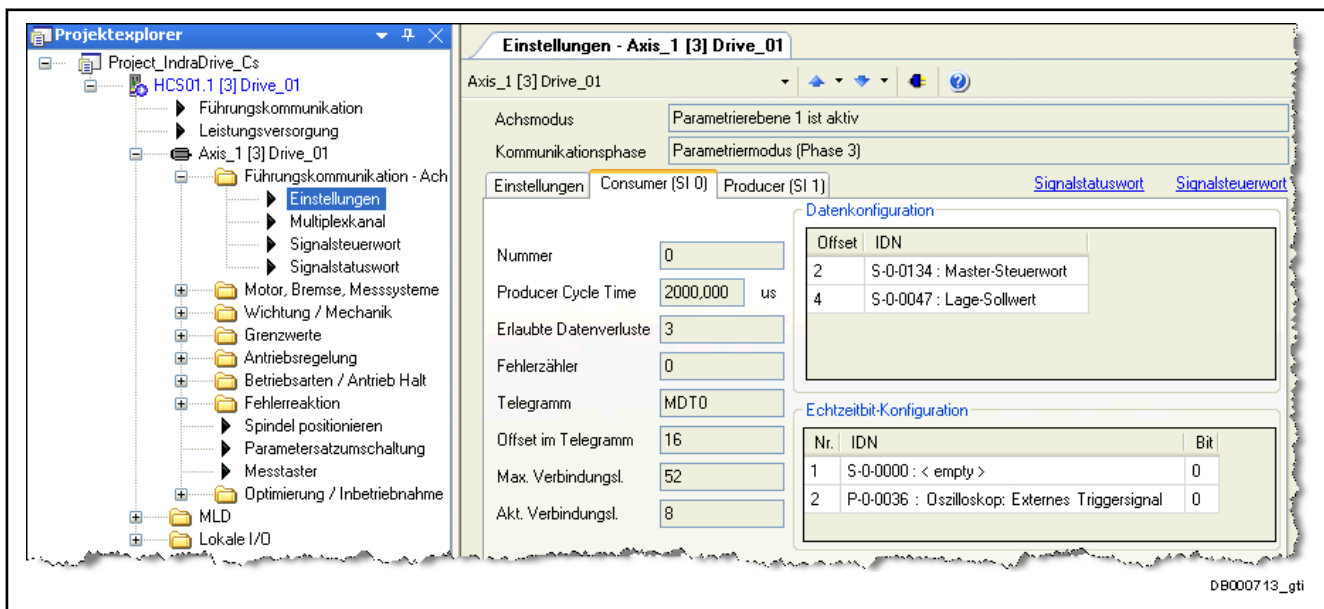


Abb. 5-17: "Führungskommunikation Einstellungen", "Prozessdaten-Aus (MDT)"

Entfernte Achsen und E/A-Module "Rexroth Inline"

Antriebe bzw. Sercos III-Geräte in den Bus aufnehmen

Antriebe bzw. Sercos III-Geräte (hier E/A-Module "Rexroth Inline") können online oder offline in die Sercos Konfiguration "CCD Basiseinstellungen" aufgenommen werden. Dazu muss eine Sercos Adresse und eine logische Antriebsnummer bzw. die Nummer des Sercos III-Geräts vergeben werden.

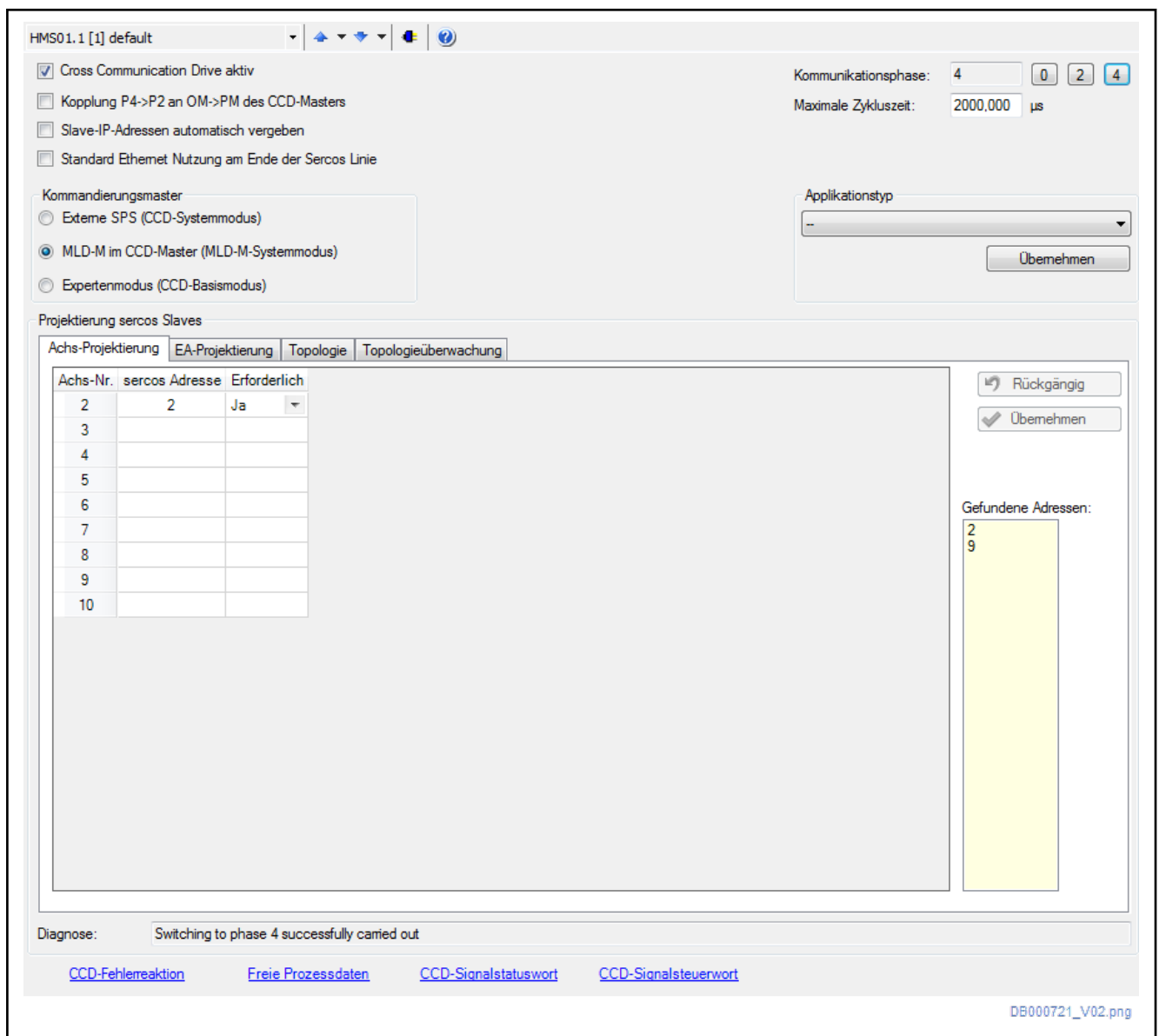


Abb. 5-18: Dialog "CCD: Basis-Einstellungen"

Eine im Dialog neu eingestellte oder geänderte Sercos Adresse (im Dialog-Abschnitt "Projektierung Sercos Slaves") wird im Gerät gespeichert, wenn die Schaltfläche "Übernehmen" betätigt wird.

Die Antriebe und die E/A-Module "Rexroth Inline" werden danach im Projektbaum unter dem Knoten "SERCOS III (CCD)" als Antriebe bzw. "EA" angezeigt.

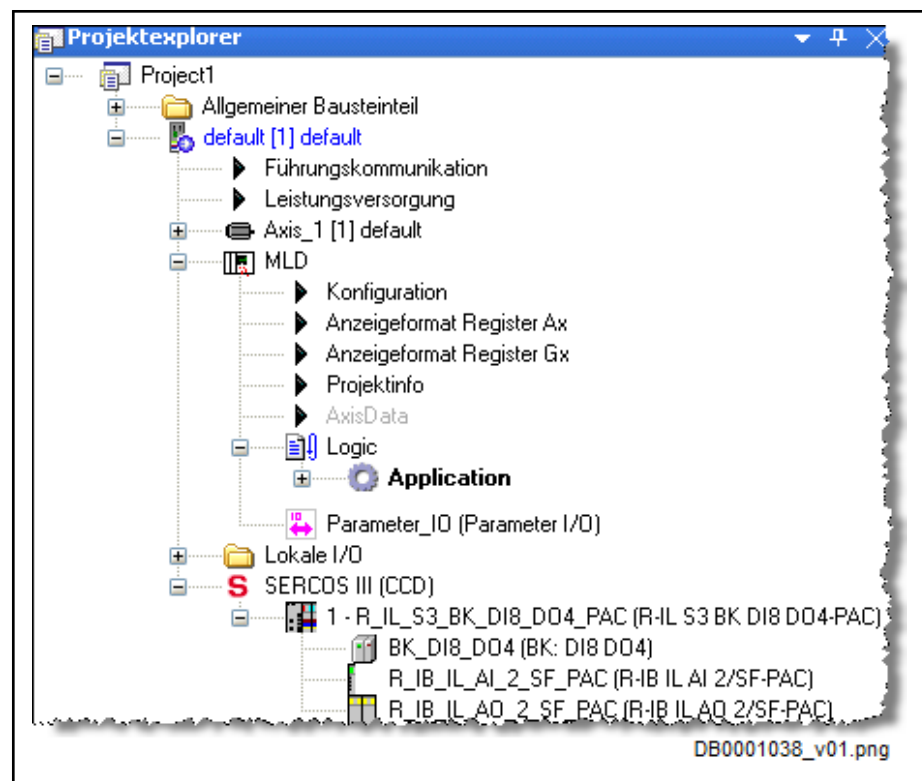


Abb. 5-19: Projektexplorer mit Sercos III-Geräten



MLD-Eingänge, die von einem Sercos III-Slave kommen, werden bei Verbindungsunterbrechung auf Null gesetzt.

Ein-/Ausgänge entfernter Achsen (Slave/Axis2..10)

Für die Parametrierung von Ein-/Ausgängen auf entfernten Achsen steht in IndraWorks MLD der Dialog "MLD-M E/A-Konfiguration" (im Projektexplorer im Ordner "MLD" innerhalb der Baumstruktur zum MLD-M-Antrieb) zur Verfügung.

Der Dialog stellt in der linken Hälfte eine Auswahl der Prozessabbild-Parameter zur Auswahl; der gewünschte Parameter des Prozessabbildes ist einzutragen. In der rechten Hälfte erscheint eine Auswahl aller auf dem Slave verfügbaren Ein- bzw. Ausgänge; das entsprechende Register der E/A-Baugruppe ist einzutragen.



Die hier eingestellten Parameter werden automatisch den zyklischen Soll- bzw. Istwerten der jeweiligen Achse hinzugefügt. Eine Liste aller zyklischen Parameter kann im CCD-Dialog eingesehen werden.

Im Folgenden wird anhand von Beispielen gezeigt, wie entfernte Ein-/Ausgänge für die MLD-M konfiguriert werden.

Beispiel 1 In dem ersten Beispiel wurde das Eingangsdoppelwort 25 (P-0-1440) über P-0-0303 auf die digitalen Eingänge des Slaves #2 zugewiesen.



Bitte beachten Sie dabei in welchen Bits die entsprechenden Klemmen wirken.



P-0-0303 enthält 32 Bit und muss daher einem 32-Bit Prozessabbildregister, wie z.B. P-0-1440, zugewiesen werden.

Weiter wurde das Eingangswort 2 (P-0-1392) über P-0-0210 auf den Analogeingang 1 des Slaves #2 zugewiesen.

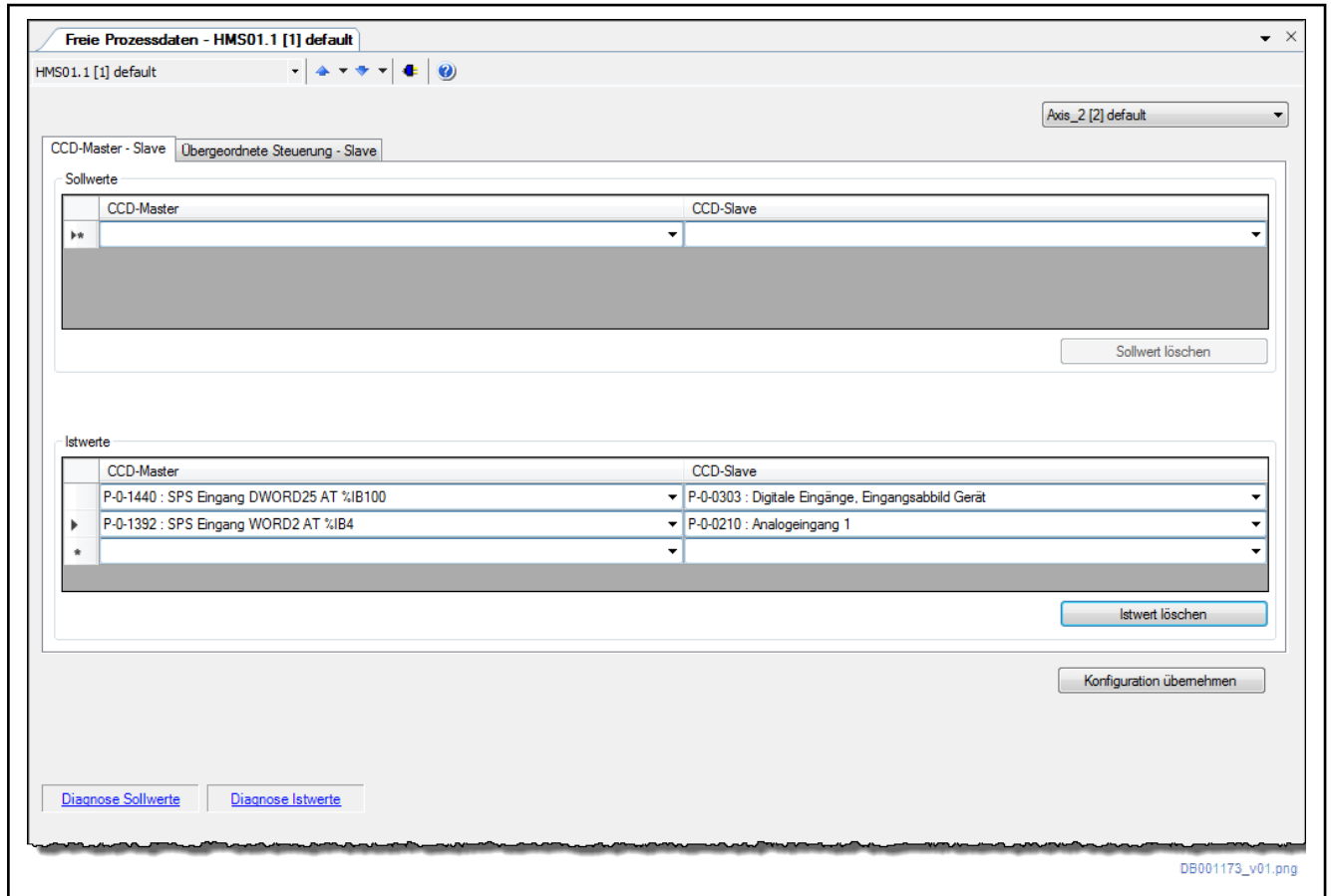


Abb. 5-20: Zuweisung von Eingängen

Beispiel 2

In dem zweiten Beispiel wurde das Ausgangswort 0 (P-0-1410) über P-0-0304 den digitalen Ausgängen des Slaves #2 zugewiesen. Die einzelnen Bits %QX2.0 und folgende steuern die Ausgänge des Slaves.



Bitte beachten Sie dabei in welchen Bits die entsprechenden Klemmen wirken.



Es ist möglich, dass digitale Ausgänge des Slave-Steuerteils von der SPS im Master und andere vom Slaveantrieb selbst angesteuert werden.

Weiter wurde das Ausgangswort 2 (P-0-1412) über P-0-0139 auf den Analogausgang 1 des Slaves geschrieben.

Kommunikationsschnittstellen und Datenkanäle der MLD

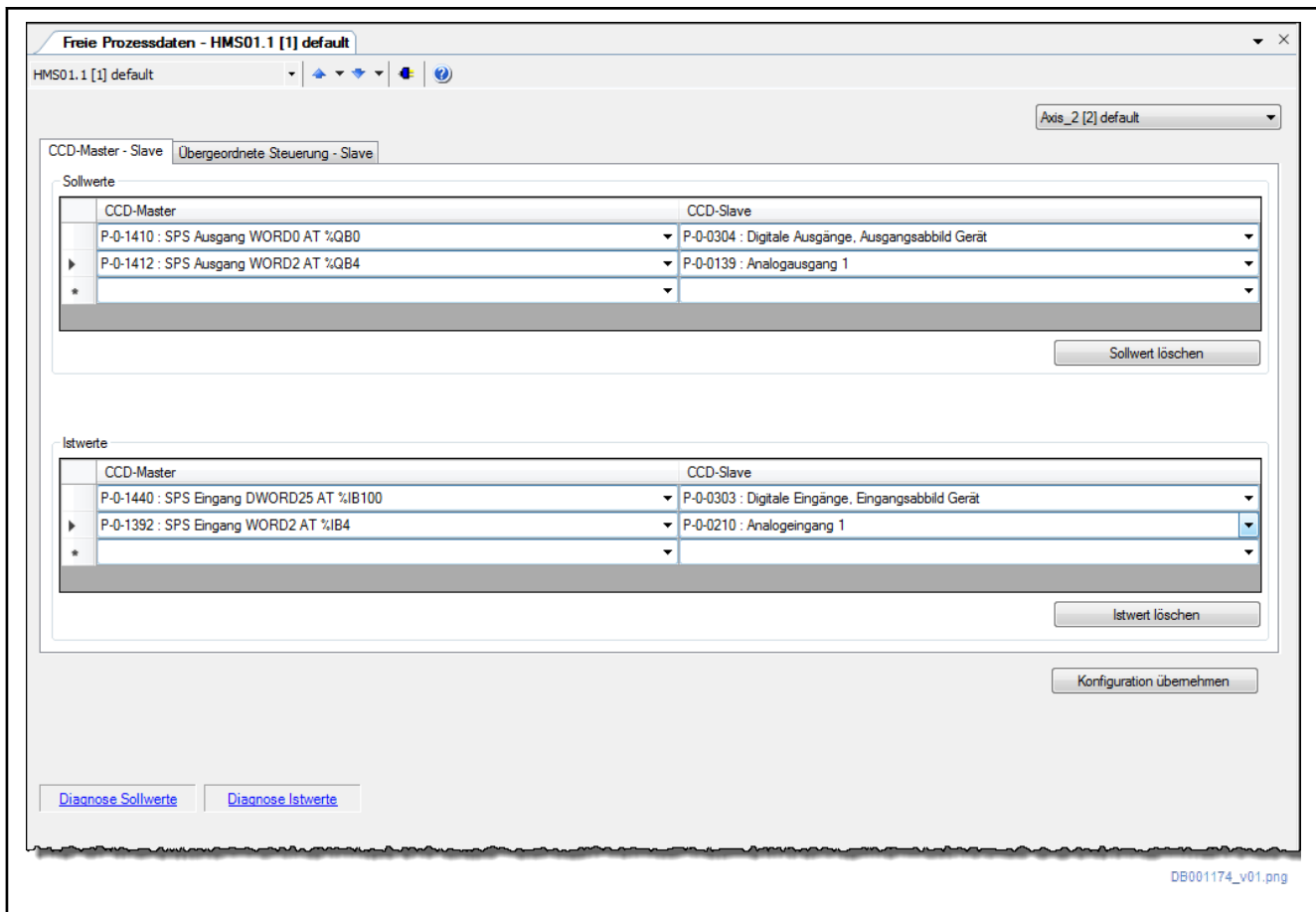


Abb. 5-21: Zuweisung von Ausgängen

Beispiel 3

In dem dritten Beispiel werden die digitalen Eingänge der lokalen Achse über CCD mittels P-0-1441 eingelesen und die digitalen Ausgänge mittels P-0-1413 ausgelesen. Das Beispiel soll zeigen, dass es bei der Verwendung der IndraMotion MLD-M möglich ist, auch auf die lokalen E/As über den CCD-Mechanismus ("virtueller Slave") zuzugreifen. Wird der CCD-Master als virtueller Slave konfiguriert, werden die lokalen E/As und die E/As entfernter Achsen zeitgleich aktualisiert; d.h. die digitalen Ein-/Ausgänge der lokalen Achse werden über Sercos III zeitgleich mit den Ein-/Ausgängen von entfernten Achsen ein- bzw. ausgelesen.



Das Gleiche gilt sinngemäß für Ausgänge auf der lokalen Achse, welche über den CCD-Mechanismus konfiguriert sind.

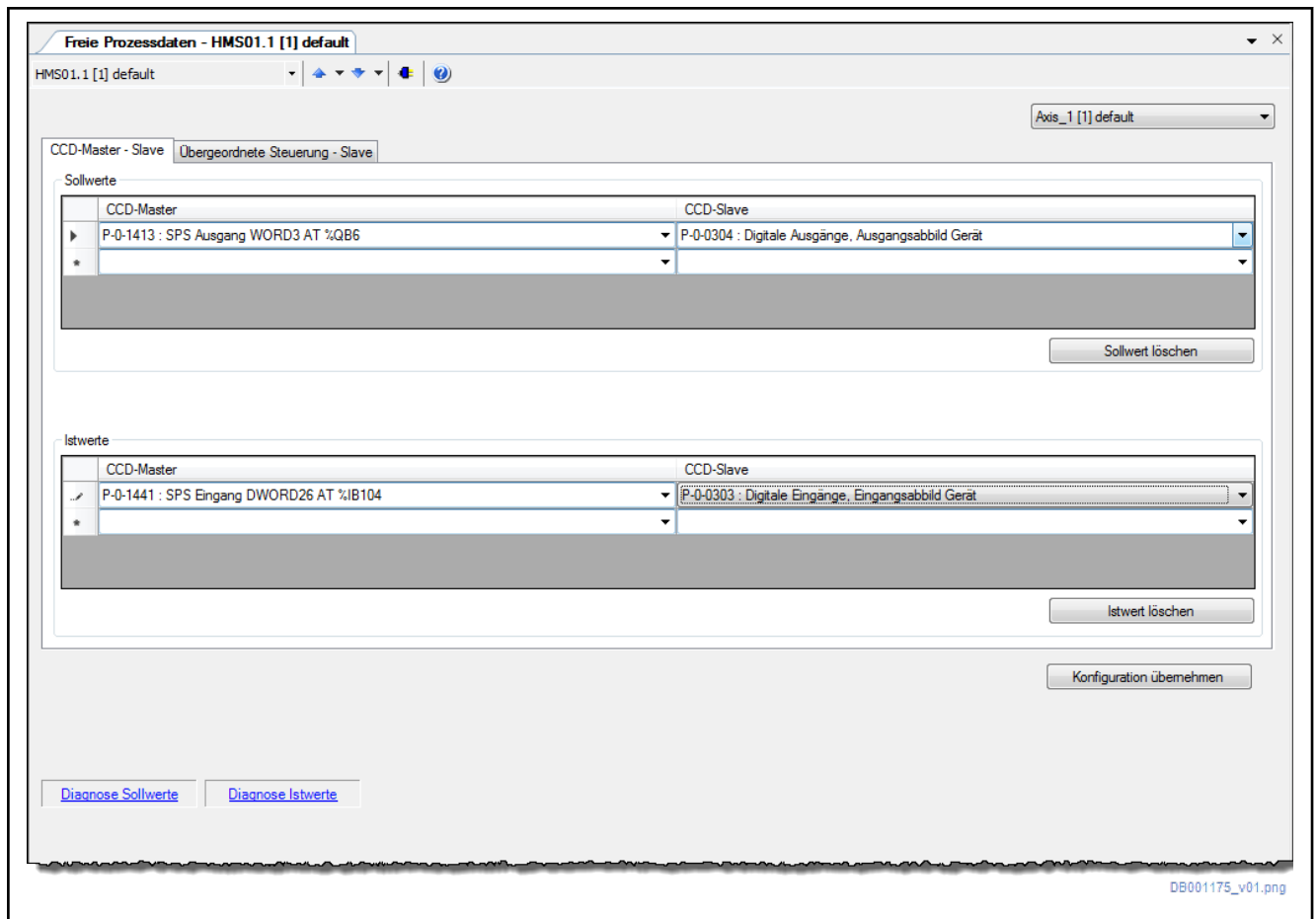


Abb. 5-22: Zugriff auf lokale Ein-/Ausgänge über CCD (virtueller Slave)

Ein-/Ausgänge von "Rexroth Inline"

Für die Parametrierung von Ein-/Ausgängen auf "Rexroth Inline"-IO-Modulen steht im Projektextplorer das Kontextmenü **Sercos (CCD-Master) ► Konfiguration Sercos Teilnehmer** zur Verfügung (Reiter "Sercos Module I/O Abbild" im Dialog).

5.2.4 Zyklische Daten im MLD-M-Systemmodus

Im MLD-M-Systemmodus werden die zyklischen Parameter der Achsen automatisch aus den festen Parametern für den Motionkanal und den Einstellungen für die Anwenderdaten in "AxisData", der E/A-Konfiguration und den freien Prozessdaten zusammengestellt. Die resultierenden zyklischen Parameter werden hier nur noch dargestellt, aber nicht parametrierbar. Für die Parametrierung der zyklischen Daten gibt es aufgeteilt nach ihrer Funktion unterschiedliche Dialoge.

Zusammenfassung aller zyklischen numerischen Daten:

- **Motiondaten**
Die Parameter des Motionkanals werden automatisch zu den zyklischen Parametern hinzugefügt. Hierzu gibt es keine Eingaben.
- **E/A-Konfiguration**
In dem Dialog "E/A-Konfiguration" können die notwendigen Parameter zum Austausch entfernter I/Os eingetragen werden.
- **AxisData**

Die notwendigen Parameter für die Anwenderdaten können in dem Dialog "AxisData" definiert werden.

- **Freie Prozessdaten**

Weitere frei wählbare zyklische Parameter für spezielle Anforderungen können in dem Dialog "Freie Prozessdaten" eingegeben werden.

- **Signalstatus- und Steuerbits**

In den Motiondaten wird das Signalsteuer- bzw. Signalstatuswort fest zyklisch konfiguriert. Damit sind von den je möglichen 16 Bits 12 Bits definiert bzw. reserviert. Die verbleibenden 4 Bits (Bit 12 bis Bit 15) können über "AxisData" als Istwert- bzw. Sollwertbits verwendet werden. Dies muss im Dialog der "AxisData" eingestellt werden. In der CCD-Konfiguration ist ein Dialog zur Sichtung aller zyklischen Daten enthalten; hier sind alle zyklischen Parameter zusammengefasst dargestellt.

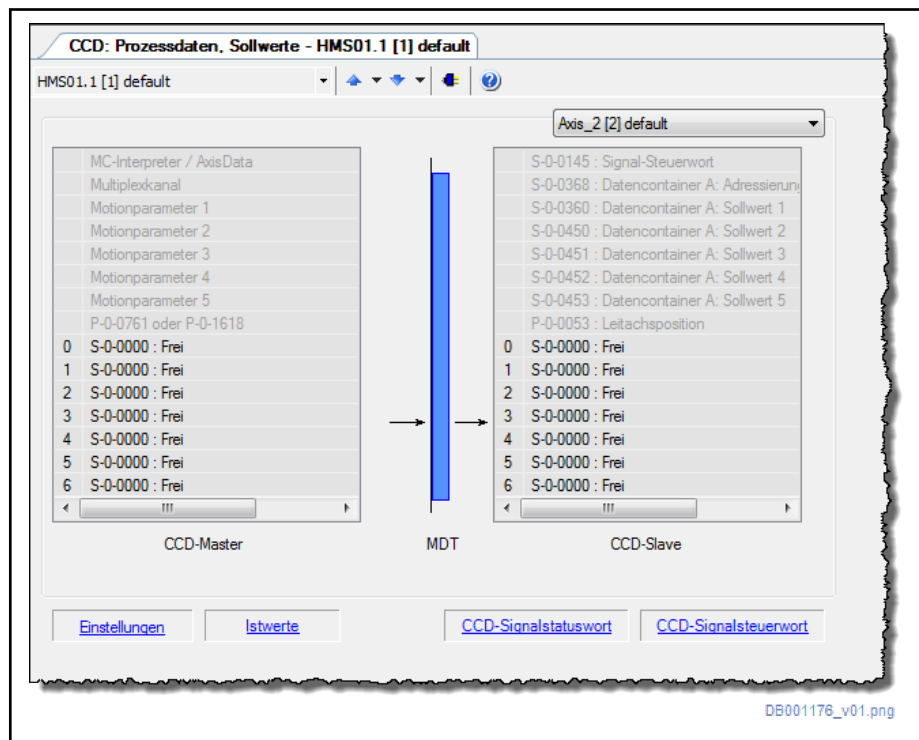


Abb. 5-23: Dialog zur Sichtung aller zyklischen Daten

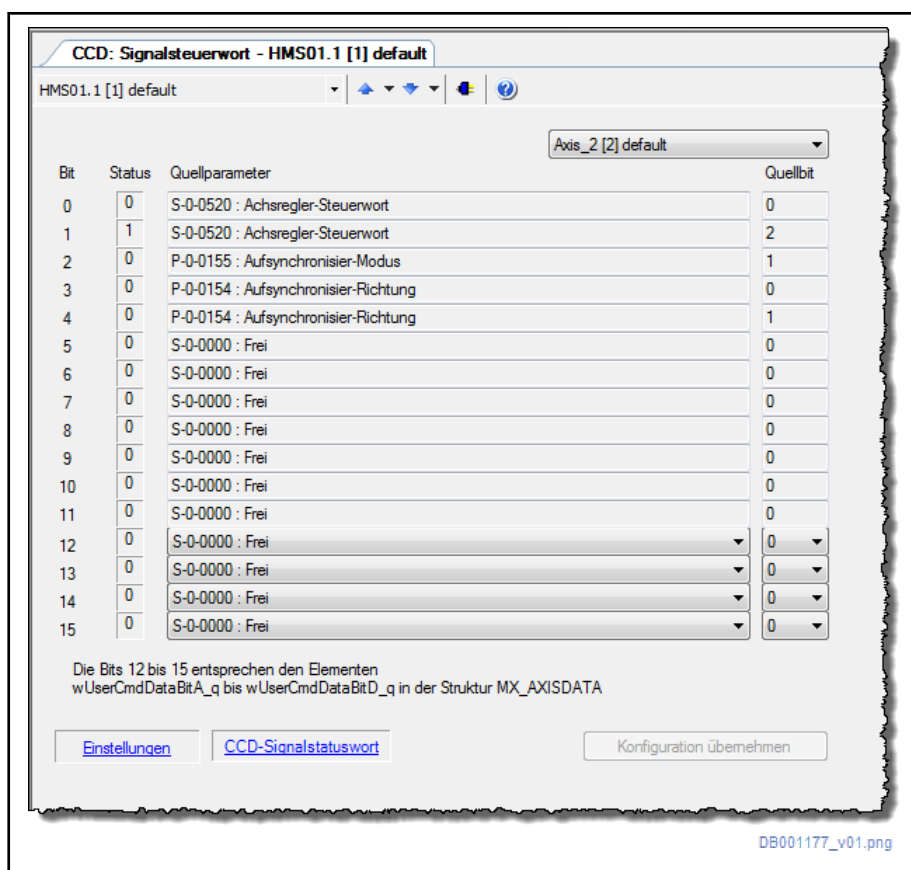


Abb. 5-24: Dialog zur Sichtung der resultierenden Konfiguration des Signalsteuerworts

Kommunikationsschnittstellen und Datenkanäle der MLD

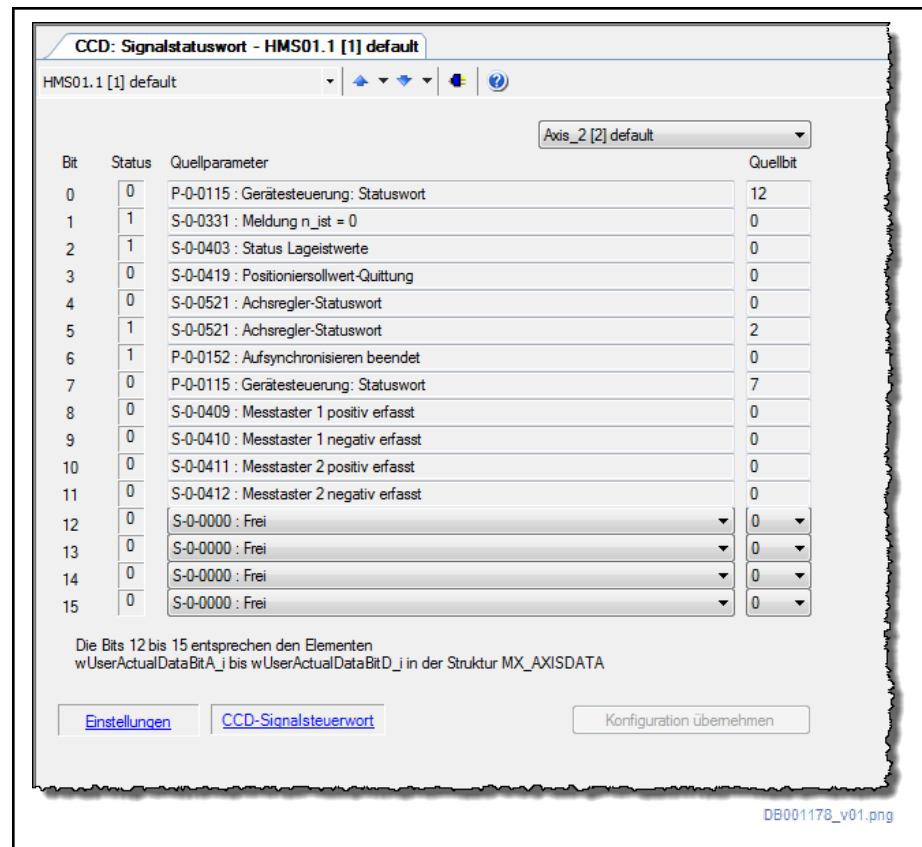


Abb. 5-25: Dialog zur Sichtung des Signalstatusworts

Zugehörige Parameter

Folgende Parameter werden im Zusammenhang mit dem Datenaustausch zwischen MLD und dem Antrieb verwendet.

Für die Konfiguration werden per Dialog folgende Parameter beschrieben:

- S-0-0187, Liste der konfig. Daten im zykl. Istwert-Datenkanal
- S-0-0188, Liste der konfig. Daten im zykl. Sollwert-Datenkanal
- P-0-1611, CCD: Konfigurationsliste Signal-Statuswort
- P-0-1612, CCD: Konfigurationsliste Signal-Steuerwort
- P-0-1613, CCD: Zuweisungsliste Signal-Statuswort
- P-0-1614, CCD: Zuweisungsliste Signal-Steuerwort
- P-0-1623, CCD: Konfigurationsliste Master-Sollwerte
- P-0-1624, CCD: Konfigurationsliste Master-Istwerte
- P-0-1625, CCD: Konfigurationsliste Slave-Sollwerte
- P-0-1626, CCD: Konfigurationsliste Slave-Istwerte

Im MLD-M-Master sind folgende Parameter als Datencontainer für die "Axis-Data" vorhanden (diese Parameter können optional für Diagnosezwecke verwendet werden):

- "P-0-1660, CCD: Statuswort Kompakt-E/As"
- P-0-1660.x.y
 - P-0-1660.x.2, CCD: Device-Steuerwort (C-Res)
 - P-0-1660.x.3, CCD: Connection-Control #1 (C-Con)
 - P-0-1660.x.20, CCD: Resource-Control (C-Res)

- "P-0-1661, CCD: Antriebs-Statuswort, Slave 1" bis "P-0-1669, CCD: Antriebs-Statuswort, Slave 9"
- P-0-1670, CCD: Aktiver Lageistwert, Master
- "P-0-1671, CCD: Aktiver Lageistwert, Slave 1" bis "P-0-1679, CCD: Aktiver Lageistwert, Slave 9"
- P-0-1680, CCD: Geschwindigkeits-Istwert, Master
- "P-0-1681, CCD: Geschwindigkeits-Istwert, Slave 1" bis "P-0-1689, CCD: Geschwindigkeits-Istwert, Slave 9"
- P-0-1690, CCD: Drehmoment-/Kraft-Istwert, Master
- "P-0-1691, CCD: Drehmoment-/Kraft-Istwert, Slave 1" bis "P-0-1699, CCD: Drehmoment-/Kraft-Istwert, Slave 9"
- P-0-1710, CCD: Signal-Statuswort, Master
- "P-0-1711, CCD: Signal-Statuswort, Slave 1" bis "P-0-1719, , CCD: Signal-Statuswort, Slave 9"
- P-0-1720, CCD: Signal-Steuerwort, Master
- "P-0-1721, CCD: Signal-Steuerwort, Slave 1" bis "P-0-1729, CCD: Signal-Steuerwort, Slave 9"
- P-0-1810, CCD: Statuswort Synchronbetriebsarten, Master
- "P-0-1811, CCD: Statuswort Synchronbetriebsarten, Slave 1" bis "P-0-1819, CCD: Statuswort Synchronbetriebsarten, Slave 9"
- "P-0-1730, CCD: Sollwert Datencontainer 1.0 4Byte" bis "P-0-1739, CCD: Sollwert Datencontainer 1.9 4Byte"
- "P-0-1740, CCD: Sollwert Datencontainer 2.0 4Byte" bis "P-0-1749, CCD: Sollwert Datencontainer 2.9 4Byte"
- "P-0-1750, CCD: Sollwert Datencontainer 3.0 4Byte" bis "P-0-1759, CCD: Sollwert Datencontainer 3.9 4Byte"
- "P-0-1760, CCD: Sollwert Datencontainer 4.0 4Byte" bis "P-0-1769, CCD: Sollwert Datencontainer 4.9 4Byte"
- "P-0-1770, CCD: Istwert Datencontainer 1.0 4Byte" bis "P-0-1779, CCD: Istwert Datencontainer 1.9 4Byte"
- "P-0-1780, CCD: Istwert Datencontainer 2.0 4Byte" bis "P-0-1789, CCD: Istwert Datencontainer 2.9 4Byte"
- "P-0-1790, CCD: Istwert Datencontainer 3.0 4Byte" bis "P-0-1799, CCD: Istwert Datencontainer 3.9 4Byte"
- "P-0-1800, CCD: Istwert Datencontainer 4.0 4Byte" bis "P-0-1809, CCD: Istwert Datencontainer 4.9 4Byte"
- "P-0-1820, CCD: Sollwert Datencontainer 1.0 2Byte" bis "P-0-1829, CCD: Sollwert Datencontainer 1.9 2Byte"
- "P-0-1830, CCD: Sollwert Datencontainer 2.0 2Byte" bis "P-0-1839, CCD: Sollwert Datencontainer 2.9 2Byte"
- "P-0-1840, CCD: Sollwert Datencontainer 3.0 2Byte" bis "P-0-1849, CCD: Sollwert Datencontainer 3.9 2Byte"
- "P-0-1850, CCD: Sollwert Datencontainer 4.0 2Byte" bis "P-0-1859, CCD: Sollwert Datencontainer 4.9 2Byte"
- "P-0-1860, CCD: Istwert Datencontainer 1.0 2Byte" bis "P-0-1869, CCD: Istwert Datencontainer 1.9 2Byte"
- "P-0-1870, CCD: Istwert Datencontainer 2.0 2Byte" bis "P-0-1879, CCD: Istwert Datencontainer 2.9 2Byte"

- "P-0-1880, CCD: Istwert Datencontainer 3.0 2Byte" bis "P-0-1889, CCD: Istwert Datencontainer 3.9 2Byte"
- "P-0-1890, CCD: Istwert Datencontainer 4.0 2Byte" bis "P-0-1899, CCD: Istwert Datencontainer 4.9 2Byte"

Die folgenden Parameter werden automatisch aus den Achsen gelesen und in der Struktur "AxisData" abgelegt:

- S-0-0040, Geschwindigkeits-Istwert
- S-0-0084, Drehmoment-/Kraft-Istwert
- S-0-0144, Signal-Statuswort
- S-0-0386, Aktiver Lageistwert
- S-0-0135/P-0-0115 (siehe "wDriveStatus_i")
- S-0-1045, Sercos III: Device-Statuswort (S-Dev)
- P-0-0089, Statuswort Synchronbetriebsarten

5.3 Sercos I/O

Die Konfiguration und Nutzung von Sercos I/O-Modulen wird durch einen in IndraMotion MLD-2G integrierten Sercos III-Master vollständig unterstützt; d. h. es werden die Teilnehmer (I/O-Module) sowohl konfiguriert als auch in die Prozessabbilder der Ein-/Ausgänge automatisch integriert. Daraus ergeben sich wesentliche Änderungen im Vergleich zur Nutzung mit der MLD-1G.

Der integrierte Sercos III-Master ist eine Komponente, die in allen Steuerungssystemen zum Einsatz kommt; d. h. vom Steuerungssystem IndraMotion MLD-2G werden grundsätzlich alle "Inline Block I/O" und "Inline Modular I/O" unterstützt, die auch von den Systemen MLC und MTX unterstützt werden. Bedingt durch das Datenformat "Big Endian" können sich jedoch Unterschiede ergeben.



Die vom Antrieb aktuell unterstützte Anzahl der projektierbaren I/Os ist zu berücksichtigen. Der Antrieb unterstützt max. 4 Buskoppler mit bis zu 63 Scheiben. In IndraWorks MLD ist die Anzahl entsprechend eingeschränkt.

Konfiguration der Sercos I/O

Der in IndraMotion MLD-2G integrierte Sercos III-Master (MLD-Master) ergänzt den im Antriebssystem realisierten CCD-Master. Während über den MLD-Master die I/O-Module konfiguriert werden, umfasst der CCD-Master alle Teilnehmer im CCD-Ring.

Die Konfiguration über den CCD-Master umfasst lediglich die Adressierung der Buskoppler. Die Konfiguration der Sercos I/O-Module (Scheiben) und die Konfiguration der Funktion der Sercos I/O-Module erfolgt unabhängig vom CCD-Master in Dialogen, die von IndraWorks zur Verfügung gestellt werden, oder durch direktes Beschreiben der entsprechenden Parameter der I/O-Module.

Wechselwirkung der Sercos Konfiguration in der MLD (IO) und der CCD-Konfiguration:

- Der MLD-Master wird von IndraWorks automatisch mit den Daten des CCD-Masters konfiguriert. Dadurch wird eine konsistente Adressierung der Teilnehmer sichergestellt.
- Vom CCD-Master wird beim Phasenschalten geprüft, ob die Konfiguration der Sercos I/O-Module korrekt ist. Hierbei prüft der CCD-Master die Adressierung der Sercos I/O. Der MLD-Master prüft, ob die Sercos I/O

Prozessabbilder der Ein-/Ausgänge	im Typ und der Bestückung korrekt ist. Letzteres kann nur erfolgen, wenn die Kommunikation über den Sercos Ring möglich ist. Im Gegensatz zu den lokalen Ein-/Ausgängen, sind die Adressen in den Prozessabbildern der Ein-/Ausgänge, die den Sercos I/O zugeordnet werden, nicht festgelegt. Da die Adressbereiche, beginnend bei %IB0 und %QB0, durch die lokalen Ein-/Ausgänge reserviert sind, werden beim Anlegen der Sercos I/O-Module die Adressen aus dem freien Bereich hinter den lokalen I/O fortlaufend vergeben.
	Mit dem "I/O Mapping" können z. B. Projektvariablen, die von der Applikation verwendet werden, den Eingangs-, Ausgangs- und Speicheradressen des Steuerungsgeräts zugewiesen werden. Des Weiteren können Adresswerte geändert und fixiert werden.
	Für die Prozessabbilder der Ein-/Ausgänge der Sercos I/O stehen in der Firmware keine Parameter zur Verfügung. Die Sercos I/O können daher nicht durch Parameterverknüpfungen im Antrieb innerhalb der Führungskommunikation oder bei den Ein- und Ausgängen entfernter und lokaler Achsen genutzt werden.
	Die Prozessabbilder der Ein-/Ausgänge der Sercos I/O werden nur dann aktualisiert, wenn sich der CCD-Ring in Kommunikationsphase 4 befindet. Andernfalls werden alle Eingänge der Sercos I/O genullt.

6 Visualisierung, Engineering und Anbindung über OCI

6.1 Übersicht

Unterstützte Funktionen Zur Visualisierung und Bedienung der mit der antriebsintegrierten SPS realisierten Funktionen werden folgende Möglichkeiten angeboten:

- Lesen und Schreiben von Parametern
- Lesen und Schreiben von SPS-Variablen

**Schnittstellen zur Anbindung von
Bedienterminals**



An IndraDrive-Antriebsregelgeräte können neben Bedienterminals (HMIs) von Bosch Rexroth auch HMIs anderer Hersteller angebunden werden; diese Dokumentation beschränkt sich darauf, die Anbindung der HMIs von Bosch Rexroth zu beschreiben.

Nachfolgende Grafik zeigt in einer Übersicht die grundsätzlichen Möglichkeiten zur Anbindung von HMIs:

- **Ethernet-Schnittstelle** [Anmerkung: TCP/IP, FTP (in Vorbereitung)]
- **Führungskommunikations-Schnittstelle** (bei Feldbus- und Sercos Antrieben)

Für die Anbindung eines Bedienterminals (HMI) sind folgende **Softwarevoraussetzungen** erforderlich: Für die Programmierung eines Bedienterminals und das Erzeugen der Symbolkonfiguration (für Parameter- und Variablenzugriff) wird IndraWorks benötigt.

Zur Inbetriebnahme der HMIs von Bosch Rexroth muss bei der Installation von IndraWorks die Komponente "WinStudio" installiert werden. (Eine nachträgliche Installation einer Komponente ist möglich, indem die Installation erneut gestartet wird; wählen Sie dann in den erweiterten Optionen "Ändern einer Installation".

6.2 Schnittstellen zur Anbindung von Kommunikationspartnern

6.2.1 Ethernet Schnittstelle (TCP/IP)

Kurzbeschreibung

Die antriebsintegrierte SPS und die Antriebsparameter sind über die Standard-Ethernet-Schnittstelle mit dem TCP/IP-Protokoll erreichbar.



Die TCP/IP-Kommunikation arbeitet nach dem CSMA/CD-Verfahren (Carrier Sense Multiple Access/Collision Detection). Aus diesem Grund ist das Zeitverhalten nicht deterministisch. Die Antwortzeiten hängen von der Netzwerkstruktur und dem Datenaufkommen ab. Für eine Abschätzung der Geschwindigkeiten setzen Sie sich bitte mit Ihrem Netzwerkadministrator in Verbindung.

Beteiligte Parameter

Folgende Parameter werden im Zusammenhang mit der TCP/IP-Kommunikation verwendet:

- P-0-1530, Engineering: MAC-Adresse
- P-0-1531, Engineering: IP-Adresse
- P-0-1532, Engineering: Netzwerkmaske
- P-0-1533, Engineering: Gatewayadresse
- ...

Einstellen der IP-Adresse

Um eine Verbindung zwischen dem Antriebsregelgerät und einem PC bzw. einem Bedienterminal aufzubauen, müssen die beteiligten Antriebsparameter passend zu den anzuschließenden Geräten eingestellt werden. D. h. die IP-Adressen der einzelnen Geräte müssen entsprechend den Netzwerkmasken so vorgegeben werden, dass die Teilnehmer einem gemeinsamen Netzwerk angehören.

Das Einstellen der IP-Adresse kann erfolgen über

- das Standard-Bedienteil am Antriebsregelgerät oder
- IndraWorks MLD

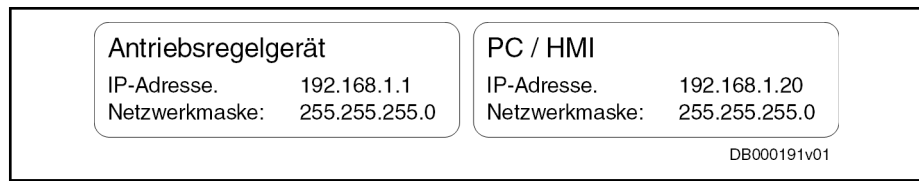


Abb. 6-2: Vergabe der IP-Adressen der einzelnen Geräte entsprechend den Netzwerkmasken

IP-Adresse, Netzwerkmaske und Standard-Gateway über das Standard-Bedienteil einstellen

Die Parametrierung von IP-Adresse, Netzwerkmaske und Standard-Gateway kann über das Bedienteil am Antriebsregelgerät vorgenommen werden; dies ist nur im Parametriermodus möglich. Das evtl. notwendige Umschalten in den Parametriermodus über das Bedienteil ist nachfolgend beschrieben.

1. Schalten Sie die Steuerspannung des Antriebsregelgeräts ein.
2. Drücken Sie für ca. 8 s gemeinsam die "ESC"- und "Enter"-Taste auf dem Bedienfeld.
3. Wählen Sie mit den "Aufwärts"/"Abwärts"-Tasten "2. Comand" aus und bestätigen Sie die Auswahl mit der "Enter"-Taste.
4. Wenn Sie noch in den Parametriermodus umschalten müssen, wählen Sie jetzt mit den "Aufwärts"/"Abwärts"-Tasten "2.11 > PM" aus und starten Sie das Kommando mit der "Enter"-Taste.
5. Wählen Sie jetzt "2.3 Kom. Ethernet" aus und bestätigen Sie die Auswahl mit der "Enter"-Taste.
6. Unter "2.3.1 P-0-1531" können Sie jetzt die IP-Adresse einstellen.



Die Übernahme der einzelnen Oktetts (Bytes) erfolgt jeweils durch Drücken der "Enter"-Taste. Mit der "Esc"-Taste kann die Übernahme abgebrochen werden.

7. Unter "2.3.2 P-0-1532" können Sie die Netzwerkmaske einstellen.
8. Unter "2.3.3 P-0-1533" können Sie das Standard-Gateway einstellen.



Damit die Parametrierung der gewünschten IP-Adresse, Netzwerkmaske und Standard Gateway wirksam wird, muss die Steuerspannung des Antriebsregelgeräts aus- und wieder eingeschaltet werden.

IP-Adresse, Netzwerkmaske und Standard-Gateway über IndraWorks einstellen

Die Parametrierung von IP-Adresse, Netzwerkmaske und Standard-Gateway kann über IndraWorks MLD vorgenommen werden.

1. IndraWorks MLD starten; beim Start die Verbindungsart "IndraDrive (Ethernet)" auswählen.
2. In IndraWorks MLD den Verzeichnisbaum im Projekt-Explorer-Fenster öffnen.
3. Klicken Sie mit der rechten Maustaste im Projekt-Explorer-Fenster auf das Antriebsregelgerät und wählen Sie "IP-Einstellungen" im Kontextmenü.

Im Arbeitsbereich wird das Fenster "IP-Einstellungen" geöffnet.

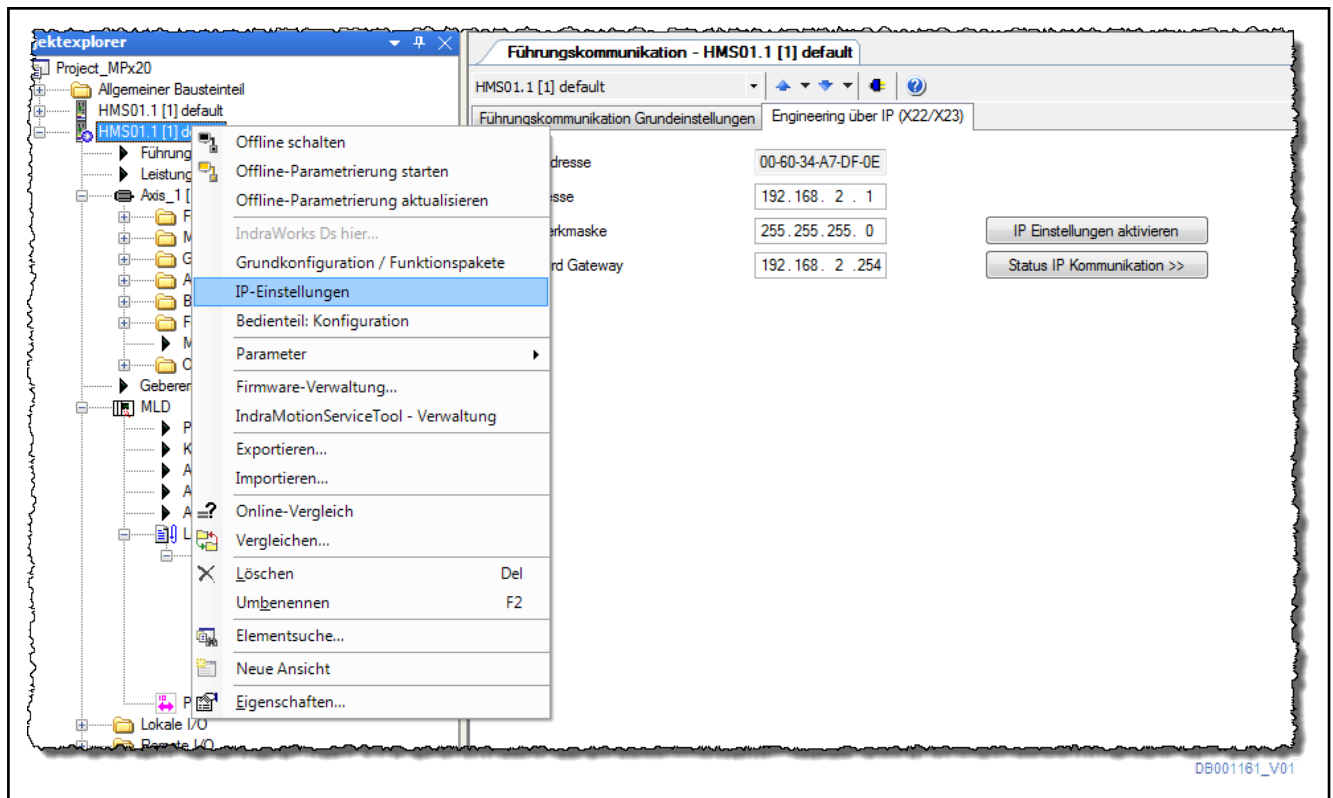


Abb. 6-3: IP-Einstellungen mit IndraWorks

4. Parametrieren Sie jetzt in der Registerkarte "Engineering Port" die gewünschte IP-Adresse, Netzwerkmaske und das Standard-Gateway.
5. Übernehmen Sie die IP-Einstellungen durch Klicken auf **IP Einstellungen aktivieren**.

6.2.2 Führungskommunikations-Schnittstelle

Kurzbeschreibung

Rexroth IndraMotion MLD nutzt die selben Führungskommunikations-Schnittstellen, welche auch für den Antrieb ohne integrierte SPS zur Verfügung stehen.

Es ist grundsätzlich zu unterscheiden, welche Ausprägung der SPS verwendet wird:

- **IndraMotion MLD als intelligente Servoachse** (-> externe Steuerung hat die Kontrolle)
- **IndraMotion MLD als autarke Motion Logic Control** (-> interne SPS übernimmt die Kontrolle)

Beteiligte Parameter

Folgende Parameter werden im Zusammenhang mit der Führungskommunikation verwendet:

- P-0-1368, SPS Globales Register AL0
- P-0-1389, SPS Globales Register GL0
- P-0-1370, SPS Globales Register G0
- P-0-1371, SPS Globales Register G1
- ...
- P-0-1385, SPS Globales Register G15

- P-0-1390, SPS Eingang WORD0 AT %IB0
- P-0-1391, SPS Eingang WORD1 AT %IB2
- ...
- P-0-1397, SPS Eingang WORD7 AT %IB14

Funktion

Der Zugriff über die Führungskommunikations-Schnittstelle ist prinzipiell identisch mit dem Zugriff über digitale Ein-/Ausgänge auf dem Steuerteil oder dem Zugriff über ein Optionsmodul für digitale Ein-/Ausgänge. Lediglich die Update-Rate ist unterschiedlich, da die Ein- und Ausgangsdaten der Schnittstelle im Takt der Führungskommunikation verarbeitet werden (vgl. S-0-0001) und alle übrigen Ein-/Ausgänge im Lagereglertakt bearbeitet.

Verwendungshinweise

Bei der Inbetriebnahme muss über "P-0-1367, SPS Konfiguration" ausgewählt werden, ob der externe Master (z. B. SPS oder NC) oder die interne SPS die Kontrolle über den Antrieb erhalten soll.

6.3 Anbindung von IndraControl V

6.3.1 Einleitung und Übersicht

Produktvorstellung

VR 21 Die Kleinbedienterminals "IndraControl VR 21" sind Bedien- und Visualisierungsterminals zum Bedienen und zum Beobachten von Maschinen. Je nach Applikation, können mit "IndraControl VR 21" über Bedienbilder Aktionen an der Maschine ausgelöst bzw. Zustände angezeigt werden (Parameter schreiben / Parameter lesen und Zugriff auf SPS-Variablen). [Aus der Betriebsanleitung "Rexroth IndraControl VR 21 Bedienpanel" (Mat.-Nr.: R911339475) können detaillierte Informationen über "IndraControl VR21" entnommen werden.]

Die Projektierung der Bedienbilder erfolgt mit Hilfe von Rexroth WinStudio (zugehörige Dokumentation: "IndraWorks 14VRS, WinStudio 7.3+SP4", Mat.-Nr.: R911341584).

Die Geräte "IndraControl VR 21" sind in verschiedenen Bildschirmdiagonalen als Singletouch- oder Multitouch-Bildschirme verfügbar. Durch die kompakte Bauweise sind die Geräte besonders vielseitig einsetzbar.

Die Spannungsversorgung erfolgt über ein externes 24-V-Netzteil, wie beispielsweise das "IndraControl VAP 01.1" (zugehörige Dokumentation: "Rexroth IndraControl VAP 01.1, Netzteil", Mat.-Nr.: R911339612)

Die Kommunikation zu einer übergeordneten Steuerung erfolgt über eine Ethernet-Schnittstelle. "IndraControl VR 21" verfügt außerdem über zwei USB-Host-Schnittstellen und einen Steckplatz für eine SD-Speicherkarte.

VEH / VEP Die Embedded Bedienterminals "IndraControl VEP" sind **PC-basierte Maschinen-Bedienterminals**, die - je nach Applikation - Aktionen an der Maschine auslösen bzw. Zustände anzeigen kann (Parameter schreiben / Parameter lesen und Zugriff auf SPS-Variablen). Außerdem können die Bedienterminals "IndraControl VEP" Steuerungsfunktionalitäten übernehmen.

- Die Embedded Terminals "IndraControl VEP" / "IndraControl VEH" sind PC-basierende Maschinen-Bedienterminals, die je nach Applikation bzw. Konfiguration auch Steuerungsfunktionalitäten erfüllen können. Bei "IndraControl VEP" stehen je nach Typ bis zu zwei PC/104-Steckplätze für Einsteckkarten zur Verfügung.

- Alle Geräte des Typs "IndraControl VEP" sind für die Lebensmittelindustrie geeignet.
- Die Embedded Terminals sind in unterschiedlichen Varianten verfügbar. Sie unterscheiden sich im Wesentlichen in der Displaygröße.
- Die Geräte des Typs "IndraControl VEP 30.2", 40.2 und 50.2 besitzen eine Touch-Screen-Bedienung. Das "IndraControl VEH 30.2" verfügt über eine Touch-Screen- und eine Tastenbedienung.
- Verfügbare Schnittstellen bei Geräten des Typs "IndraControl VEP":
 - 1 externer VGA-Anschluss (15-polig, HD-Sub)
 - 2 USB-Anschlüsse (Typ A)
 - 2 Ethernet-Anschlüsse (RJ 45, 10/100 Base-T)
 - 1 PS/2-Anschluss (serielle Schnittstelle für Tastatur / Maus)
 - 1 serielle Standardschnittstelle RS232 (9-polig; D)

Weitere Angaben zu "IndraControl VEH" / "IndraControl VEP" können der Projektierungsanleitung "Rexroth IndraControl VEP / VEH", Mat.-Nr.: R911309681 entnommen werden.

VH2110 Die Handbediengeräte "IndraControl VH21" sind tragbare Bedien- und Visualisierungsgerät zum Bedienen, Einrichten, Parametrieren und Diagnostizieren einer über Ethernet angeschlossenen Steuerung. Als Sicherheitsfunktionen sind ein 3-stufiger Freigabetaster sowie ein zweikreisig ausgeführter NOT-HALT-Taster installiert. Über eine USB-OTG-Schnittstelle können ein USB-Speicherstick, Maus oder Tastatur angeschlossen werden. [Aus der Betriebsanleitung "IndraControl VH 2110.01 Handbediengerät" (Mat.-Nr.: R911346749) können detaillierte Informationen über "IndraControl VH21" entnommen werden.]

Das Handbediengerät wird über ein Anschlussmodul angeschlossen (zugehörige Dokumentation: "IndraControl VAC 30.2, VAC 31.1, VAC 05.1 Anschlussmodul", Mat.-Nr.: R911338408)

Auf allen Geräten des Typs "IndraControl V" laufen spezifische Betriebssysteme; die Beschreibung der Betriebssysteme entnehmen Sie bitte der Projektierungsanleitung "Rexroth IndraControl V-Geräte Betriebssysteme", Mat.-Nr.: R911343900

Voraussetzungen

Antriebsfirmware Zum Anzeigen von SPS-Variablen der antriebsintegrierten SPS (Rexroth IndraMotion MLD) und Parametern muss mindestens eine IndraDrive-Antriebsfirmware FWA-INDRV*-MPx-17VRS-MS verwendet werden.

Geräteversion Es muss die zur Antriebsfirmware passende Geräteversion verwendet werden.

Für den Betrieb der HMI-Geräte "IndraControl VR21", "IndraControl VH21" ist die Antriebsfirmware MPx-20 notwendig.

Symbolkonfiguration Sollen Variablen der Antriebs-SPS angezeigt werden, so muss eine Symbolkonfiguration im Programmiersystem erzeugt werden.

Um Parameter (S- und P-Parameter) anzuzeigen, liefert die Antriebsfirmware bereits vorgefertigte Symbolkonfigurationen. In diesen Symbolkonfigurationen sind alle möglichen Parameter mit Datentyp eingetragen.

Mit Hilfe dieser vorgefertigten Symbolkonfigurationen wird folgendes ermöglicht:

- Anzeigen und Editieren von Nutzdaten
- Anzeigen der Datenstatus

- Anzeigen von Maximalwerten
- Anzeigen von Parameter-Namen
- Anzeigen von Parameter-Einheiten



Die Symbolkonfiguration für den Zugriff auf Antriebsparameter (S- und P-Parameter) ist firmwareabhängig!

Ethernet-Kommunikation

Für die Ethernet-Kommunikation wird ein Steuerteil mit Ethernet-Engineering-Port benötigt. Die Kommunikation über Ethernet mit dem TCP/IP-Protokoll ist ab Antriebsfirmware FWA-INDRV*-MPx-17VRS-MS möglich.

Sonstige Voraussetzungen

- Installation von IndraWorks MLD mit der Komponente "WinStudio" in der aktuellen Version
- Ein Gerät der IndraControl V-Serie mit aktueller Firmware

6.3.2 Konfiguration der Symbole für Variablen-/ Parameterzugriff

Kurzbeschreibung

Funktion

Die Symbole werden benötigt, um mit einem Bedienterminal (HMI) SPS-Variablen und Antriebsparameter anzuzeigen bzw. zu editieren.

- Die **Symbolkonfiguration für den Zugriff auf Antriebsparameter** (S- und P-Parameter) ist firmwareabhängig und wird von Bosch Rexroth bereitgestellt.
- Eine **Symbolkonfiguration für den Zugriff auf SPS-Variablen der MLD** muss durch den Anwender mit Hilfe der Symbolkonfiguration erzeugt werden.

Mit Hilfe der Symbolkonfiguration ist der Zugriff auf die Variablen der IndraMotion MLD und der Antriebsparameter unter Verwendung des Variablen-Namens bzw. der Direktvariablen möglich.

Erzeugen der Symbolkonfiguration für den Zugriff auf SPS-Variablen der MLD

Um eine Symbolkonfiguration für den Zugriff auf SPS-Variablen der MLD zu erzeugen, rufen Sie in IndraWorks MLD die zugehörige Funktion wie folgt auf:

1. Wählen Sie den Knoten **MLD ► Logic ► Application ► Symbolkonfiguration** aus.
2. Klicken Sie doppelt auf "Symbolkonfiguration".
⇒ Das Fenster "Symbolkonfiguration" wird geöffnet.

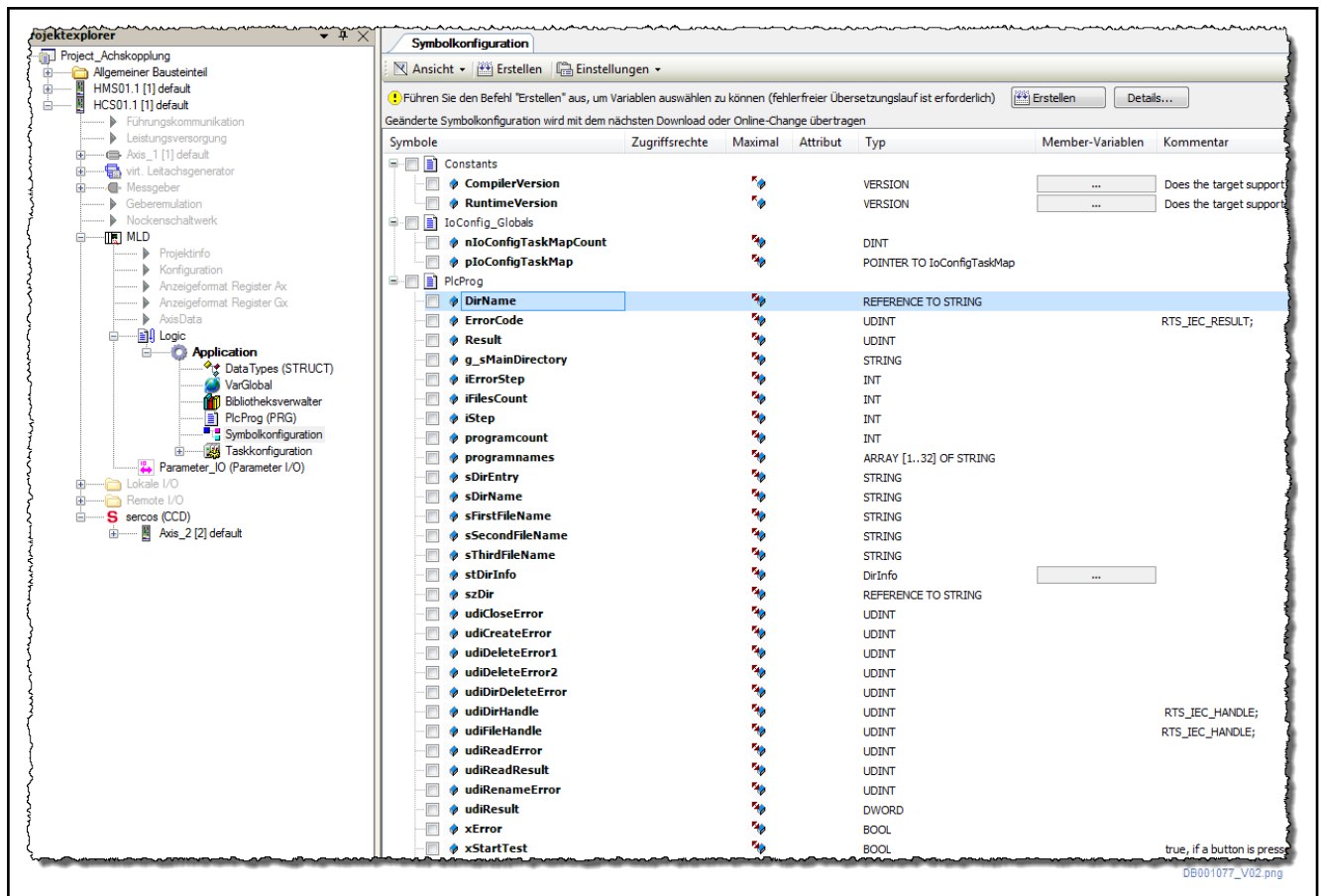


Abb. 6-4: Erzeugen der Symbolkonfiguration für den Zugriff auf SPS-Variablen der MLD

6.3.3 Anbindung eines IndraControl VE* über Ethernet

Zum Herstellen der Verbindung zwischen einem Antrieb und einem IndraControl VE* sind folgende Schritte erforderlich:

1. Stellen Sie die IP-Adresse des Antriebsregelgeräts ein (siehe "[Einstellen der IP-Adresse](#)").
2. Fügen Sie ein IndraControl VE* in das Projekt ein.
3. Rufen Sie (wie im Screenshot gezeigt) das Kontextmenü auf und wählen **Treiber einfügen**

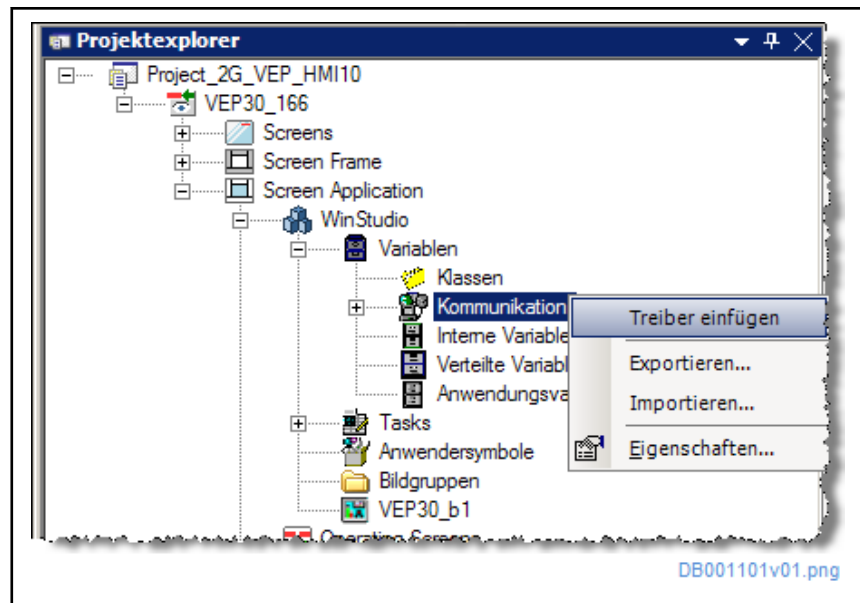


Abb. 6-5: Einfügen eines Kommunikationstreibers
Der Dialog "Kommunikationstreiber" öffnet sich.

4. Wählen Sie "BR_WS" aus, um die Com-Serverdienste zu nutzen.

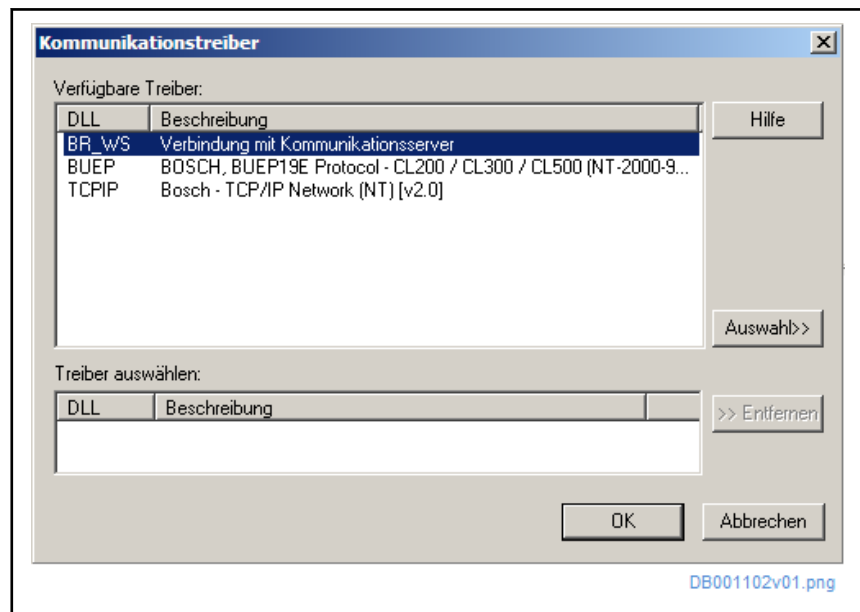


Abb. 6-6: "Kommunikationstreiber" "BR_WS" auswählen

5. Rufen Sie über das Kontextmenü des soeben eingefügten Treibers den Eigenschafts-Dialog auf.

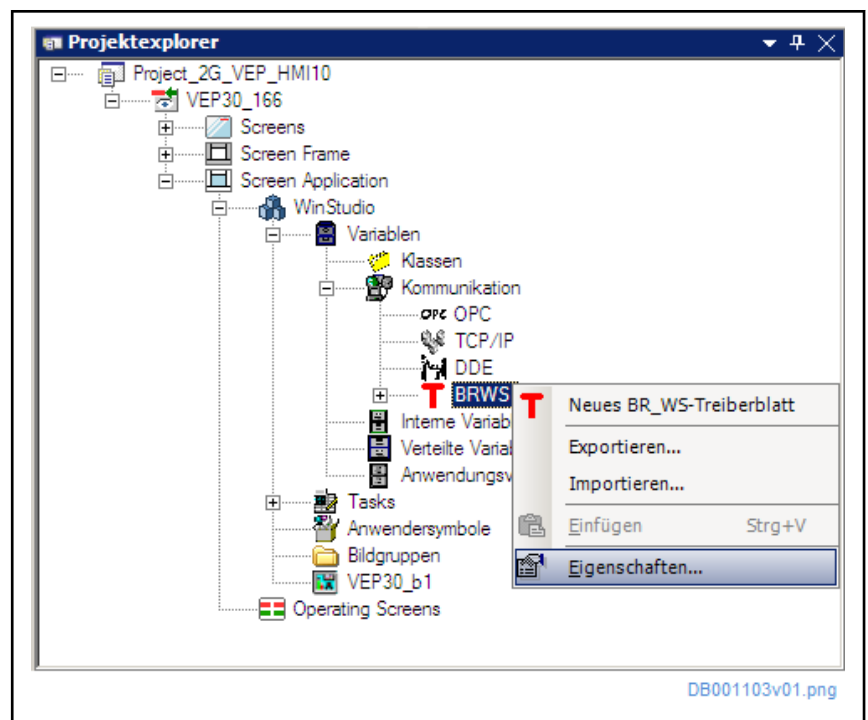
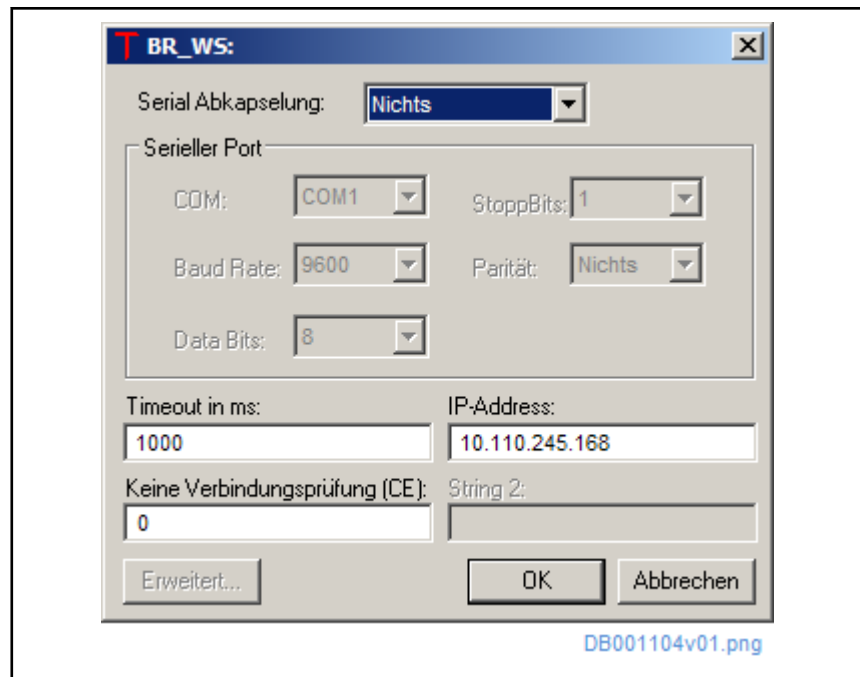


Abb. 6-7: *Eigenschaften-Dialog von "BR_WS" aufrufen*

6. Nehmen Sie in dem Eigenschaften-Dialog von "BR_WS" folgende Einstellungen vor:

- "Serial Abkapselung": "Nichts"
- "Timeout in ms": Es ist eine Timerüberwachung der Kommunikationsverbindung anzugeben. Empfohlener Wert bei einer Kommunikationsverbindung zum Antriebsregelgerät: ≥ 1000 ms.
- "IP Address": TCP/IP-Adresse der Zielsteuerung
- "Keine Verbindungsprüfung (CE)": "0" (die Kommunikationsverbindung soll auch bei Geräten überwacht werden, die eine CE-Kennzeichnung tragen).

Abb. 6-8: *Eigenschaften-Dialog von "BR_WS"*

In der Online-Hilfe von IndraWorks finden Sie weiterführende Informationen zum Kommunikationstreiber "BR_WS" (Bosch-Rexroth-WinStudio Treiber) (z. B. Projektierung der Treiberblätter, unterstützte Datentypen,...).

6.4 Open Core Interface für Antriebe

Mit "Open Core Interface für Antriebe" stellt Bosch Rexroth Schnittstellen zur Verfügung, die eine einfache Anbindung der Maschinenautomatisierung an die übergeordnete IT-Automation ermöglicht.

Für verschiedene Programmierumgebungen werden die jeweils passenden Funktionsbibliotheken in einem SDK (Software Development Kit) bereitgestellt. In diesen Bibliotheken befinden sich Funktionen für den direkten Zugriff auf alle Daten und Funktionen des Antriebs.

"Open Core Interface für Antriebe" bietet die folgenden Möglichkeiten:

- **Smart Devices**

Durch die Verwendung von Smart Devices und darauf installierten Apps können neue Bedienkonzepte für Anlagen entwickelt werden.

Die Apps haben Zugriff auf alle Hardware-Funktionen der Smart Devices, wie die Kamera oder die Beschleunigungssensoren. Damit erschließen sie den gesamten Bedienkomfort von Smart Devices für die Inbetriebnahme, Bedienung und Diagnose von IndraDrive- und HydraulicDrive-Antrieben.

Open Core Interface für Antriebe unterstützt derzeit die Betriebssysteme Android, iOS und Windows Phone.

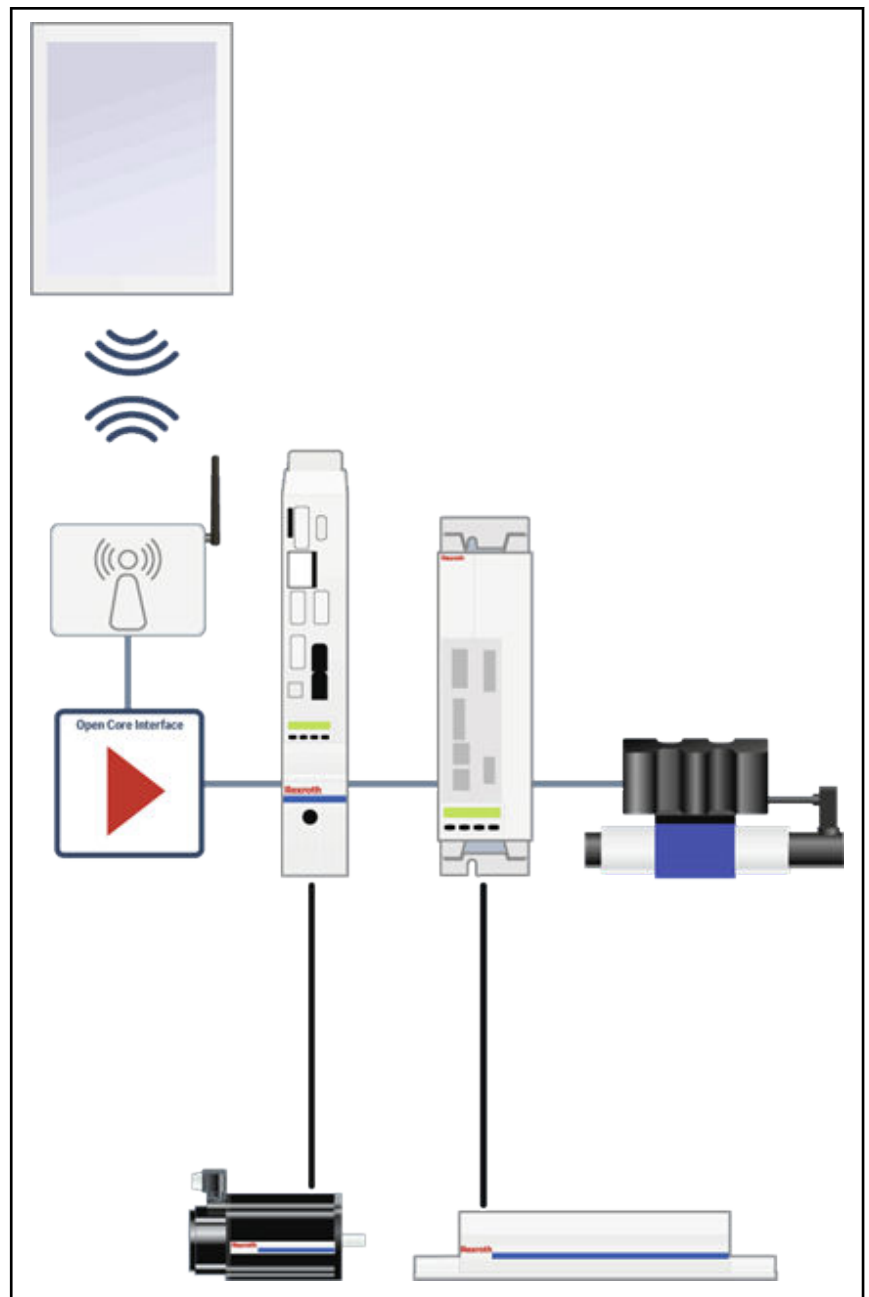


Abb. 6-9: Inbetriebnahme, Bedienung und Diagnose von IndraDrive- und HydraulicDrive-Antrieben über Smart Devices

- **IT-Automation**

Open Core Interface im Bereich der IT-Automation bezeichnet die Verwendung von PC-basierten Lösungen im Automationsumfeld einer Produktionsmaschine.

Für den schnellen Datenaustausch werden Anbindungen an Windows- und Linux-basierte Entwicklungsumgebungen zur Verfügung gestellt, in denen mit Hochsprachen wie Java, C, C++ und C# programmiert wird.

- **Rapid Control Prototyping**

Open Core Interface bietet alle Möglichkeiten für Rapid Control Prototyping, einer Entwurfsmethode zur Regelungs- und Steuerungsentwicklung. Es dient zur frühzeitigen Entwicklung von Prozessen ohne reale

Maschine und trägt somit wesentlich zu einer kostenoptimierten und risikominimierten Entwicklung bei; unterstützt wird Rapid Control Prototyping mit LabVIEW, Simulink™, MATLAB®.



Informationen und Unterstützung zu "Open Core Interface" erhalten Sie im [Forum](#); hier können Sie auch Fragen an Mitglieder und die Experten von Bosch Rexroth stellen.

6.4.1 Zielplattformen

Mit "eal_sdk" ("Easy Automation Library"- "Software Development Kit") stellt Bosch Rexroth Programmierschnittstellen für folgende Zielplattformen zur Verfügung:

- PC mit Windows / PC-basierte Steuerung mit LabVIEW
- PC mit Linux als Betriebssystem
- (Mac OS auf Anfrage)
- Smart Device mit Android, iOS oder Windows Phone als Betriebssystem
- Linux-basierte Controller (z. B. BeagleBone Black®, Raspberry pi®,...)

6.4.2 Unterstützte Geräte

Mit Hilfe von "eal_sdk" können Sie Anwendungen erstellen, um Antriebsregelgeräte und Frequenzumrichter von Bosch Rexroth zu konfigurieren und zu steuern. Unterstützt werden Geräte, die das "Sercos Internet Protocol" (S/IP) unterstützen (Geräte mit Multi-Ethernet-Schnittstelle):

- elektrische Antriebsregelgeräte der Produktfamilien IndraDrive Cs, IndraDrive C, IndraDrive M, IndraDrive Mi, IndraDrive ML [mit allen in der Produktfamilie verfügbaren Steuerteilen (ECONOMY, BASIC und ADVANCED, Single- und Doppelachsgeräte)]
- hydraulische Antriebsregelgeräte "Integrated Axis Controller (IAC)", "Hydraulic Motion Control (HMC)"
- Frequenzumrichter des Typs EFC x610

6.4.3 Anwendungsmöglichkeiten

Über die Führungskommunikations-Schnittstelle, die Sercos Master-Schnittstelle oder den Engineering Port können ein oder mehrere Antriebe in Stern- oder Linientopologie konfiguriert und gesteuert werden.

Mit "Open Core Interface für Antriebe" sind folgende Anwendungen möglich:

- Mit "Open Core Interface für Antriebe" kann eine konventionelle Steuerung ersetzt werden; sowohl die Führungskommunikation als auch die Kommandierung des Antriebsregelgeräts wird von "Open Core Interface für Antriebe" übernommen.



Die Führungskommunikation muss für diesen Anwendungsfall deaktiviert werden.

- "Open Core Interface für Antriebe" kann parallel zu einer aktiven Führungskommunikation (Sercos, PROFINET, EtherCAT mit EoE, EtherNet/IP, Ethernet POWERLINK) eingesetzt werden. Es sind dann folgende Anwendungen möglich:
 - Als HMI-Schnittstelle; die Kommandierung erfolgt über eine externe SPS über das Führungskommunikationsprotokoll.

- Als Inbetriebnahmetool zum Laden einer Parameterdatei, Tausch der Antriebsfirmware,...
- Als Diagnose- und Debug-Schnittstelle, indem das Firmware-Oszilloskop genutzt wird.
- Als Subsystem zur antriebsinternen SPS (IndraMotion MLD).



Ist die Führungskommunikation aktiv, dann ist es über die OCI-Schnittstelle nicht möglich, den Antrieb in Regelung (AF) zu schalten.

6.4.4 SDK: Voraussetzungen und Installation

Um das "eal_sdk" ("Easy Automation Library"- "Software Development Kit") nutzen zu können, führen Sie bitte die folgenden Schritte aus:

1. Stellen Sie sicher, dass Sie eines der folgenden Systeme einsetzen:
 - elektrisches Antriebsregelgerät mit IndraDrive-Firmware MPx-18 (oder höher)
 - hydraulisches Antriebsregelgerät mit HydraulikDrive-Firmware HDx-20 (oder höher)
 - Frequenzumrichter des Typs EFC x610
 - IndraMotion MLD-Steuerungssystem in der Version 13VRS oder höher
2. Für Inbetriebnahme und Engineering der Antriebe müssen Sie eine IndraWorks DS/IndraWorks MLD in der Version 13VRS oder höher installiert haben; alternativ kann auch die Anwendung "Drive tool" bzw. "Drive tool EFC" verwendet werden, die in dem SDK enthalten ist.
3. Registrieren für das **Engineering Network**

Um sich das SDK "eal_sdk" von der Internetseite von Bosch Rexroth herunterladen zu können, müssen Sie sich für das **Engineering Network** registrieren:

1. Gehen Sie auf die Internetseite von [Bosch Rexroth](#).
 2. Registrieren Sie sich über "myRexroth". (Eine Beschreibung des Registrierungsprozesses finden Sie im [Engineering Network-Forum](#).)
 3. Rufen Sie die Internet-Seite zu "Open Core Engineering" auf.
 4. Nach der Registrierung steht im [Downloadbereich](#) des **Engineering Network** das SDK "eal_sdk" zur Verfügung.
4. Abhängig von Anwendung, Geräte-Plattform und Betriebssystem des Zielgeräts, müssen Sie die in dem SDK enthaltene "Easy Automation Library" in Ihre Entwicklungsumgebung integrieren.

6.4.5 SDK: Inhalte

Das SDK "eal_sdk" ist in verschiedene "Toolboxes" unterteilt. Jede Toolbox hat in dem SDK ein eigenes Unterverzeichnis. Die von den "Toolboxes" unterstützten Funktionen sind im Wesentlichen identisch, sie unterscheiden sich bzgl. der Entwicklungsumgebung und des Betriebssystems, in dem die Entwicklungsumgebung eingesetzt wird.



In jeder Toolbox befinden sich nicht nur Bibliotheken, sondern auch Dokumentationen und [Anwendungsbeispiele](#).

Unterstützte Funktionen der Toolboxes:

- **System**
 - Verbindung zum Antriebsregelgerät bzw. Frequenzumrichter aufnehmen
 - Systeminformationen, wie Diagnosedaten und Firmware-Version lesen
 - Firmware-Up- und Download
- **Parameter**

- Parameter lesen und schreiben
- Parameter-Name, -Attribute, -Einheit und -Status lesen
- Kommandos stoppen und ausführen
- **SPS** (ab Firmware-Version 20)
 - SPS starten, stoppen und zurücksetzen
 - SPS-Status lesen und schreiben
 - Symbolvariablen durchsuchen, lesen und schreiben
- **Bewegung**
 - Allgemeine Funktionen (Reglerfreigabe aktivieren, Antrieb in den Zustand STOP bringen, antriebsgeführt referenzieren)
 - Achsbewegungen (Geschwindigkeitsvorgabe, Verfahren auf Absolutposition, Verfahren um einen Weg zusätzlich zur Zielposition, Verfahren um einen Weg ab der aktuellen Istposition)
- **Oszilloskop**
 - Oszilloskop-Kanäle konfigurieren
 - Oszilloskop-Trigger konfigurieren
 - Oszilloskop-Daten lesen

Das SDK wird ständig weiterentwickelt; in den nachfolgenden Kapiteln eine Übersicht des SDK "eal_sdk" in der Version 2.0



Da sich die Entwicklungsumgebungen und Betriebssysteme ändern können, übernimmt Bosch Rexroth keine Gewähr für eine einwandfreie Funktion des SDK.

EAL4Android

"EAL4Android" enthält eine Android-Bibliothek und eine "Mono for Android"-App, um Antriebsregelgeräte und Frequenzumrichter von Bosch Rexroth zu konfigurieren und zu steuern, die das "Sercos Internet Protocol" (S/IP) unterstützen.

Systemvoraussetzungen: Java™ 32/64-Bit, Android Studio

EAL4DotNet

Mit Hilfe der Toolbox "EAL4DotNet" wird .NET-Bibliotheken der Zugriff auf Funktionen von Antriebsregelgeräten und Frequenzumrichtern von Bosch Rexroth ermöglicht.

Unterstützte Betriebssysteme des Zielgeräts: Windows XP/Vista/7/8, Android¹⁾, iOS¹⁾, Mac OS¹⁾, Linux¹⁾ [¹⁾: mit Hilfe von Mono]

Unterstützte Entwicklungsumgebung: Microsoft Visual Studio 2005 oder neuer, Visual Studio mit Xamarin-Erweiterung, Xamarin Studio, RAD studio®

Unterstützte Programmiersprachen: .NET (C#, VB, F#), VBA (für MS-Excel, MS-Access, MS-PowerPoint, MS-Word), Delphi

EAL4C

Mit Hilfe der Toolbox "EAL4C" werden die Funktionen von "EAL4DotNet" in Strukturfunktionen mit Mono umgewandelt.

Unterstützte Betriebssysteme des Zielgeräts: Windows XP/Vista/7/8, Linux mit Mono-Erweiterung

Unterstützte Entwicklungsumgebung: Eclipse, Microsoft Visual Studio, Qt etc.

Unterstützte Programmiersprachen: C/C++

EAL4Java

Die Toolbox "EAL4Java" verpackt die "EAL4C"-Funktionalität als objektorientierte Java-Klassen, indem sie eine native Java NDK-Bibliothek verwendet. Mit EAL4Java können EAL-Funktionen aus der Java-Laufzeitumgebung heraus aufgerufen werden.



Anwendungen, die die Toolbox "EAL4Java" verwenden, können nur auf der Java-Laufzeitumgebung unter Windows und Linux laufen.

Unterstützte Betriebssysteme des Zielgeräts: Windows XP/Vista/7/8, Linux

Unterstützte Entwicklungsumgebung: Eclipse

Unterstützte Programmiersprachen: Java

EAL4LabVIEW

Mit Hilfe der Toolbox "EAL4LabVIEW" wird unter LabVIEW der Zugriff auf Funktionen von Antriebsregelgeräten und Frequenzumrichtern von Bosch Rexroth ermöglicht.

Unterstützte Betriebssysteme des Zielgeräts: Windows XP/Vista/7

Unterstützte Entwicklungsumgebung: National Instruments LabVIEW

Unterstützte Programmiersprachen: G (LabVIEW VI language)

6.4.6 Anwendungsbeispiele

Im "Engineering Network" [[Download→Apps](#) ([Registrierung und Login erforderlich](#))] und in dem SDK stellen wir verschiedene Demoprogramme und Apps zur Verfügung.

Zu allen Demoprogramme und Apps erhalten Sie den kompletten Quellcode, teilweise die entsprechenden Projektdateien, Informationen über den Anwendungsfall, die Voraussetzungen für die Ausführung und die Erstellung.

In den nachfolgenden Kapiteln einige Beispiele.

Windows-Anwendung mit Microsoft Visual Studio zum Steuern und Überwachen von "IndraDrive"-Antriebsregelgeräten

In der Entwicklungsumgebung Microsoft Visual Studio wurde die Windows-Anwendung "DriveTool" entwickelt.



In dem SDK "eal_sdk" finden Sie unter `..\eal4DotNET\Samples\Sample - DriveTool` die Windows-Anwendung "DriveTool" und auch die Quelldateien.

Mit "DriveTool" ist der Zugriff auf folgende Funktionen von Antriebsregelgeräten der Produktfamilie "IndraDrive" möglich:

- Suchen nach verfügbaren Antrieben
- Verbindung mit einem Antrieb über die Ethernet-IP-Adresse herstellen
- Leistungsstatus lesen und Leistung zuschalten
- Neustart des Antriebs
- Initialisieren des Antriebs
- Lesen des Firmware-Typenschlüssels
- Schalten zwischen Betriebs- und Parametriermodus
- Referenzieren des Antriebs
- Löschen von Antriebsfehlern
- Lesen und Auswählen der Betriebsart
- Lesen und Schreiben von Motor-Geschwindigkeit, -Position und -Moment

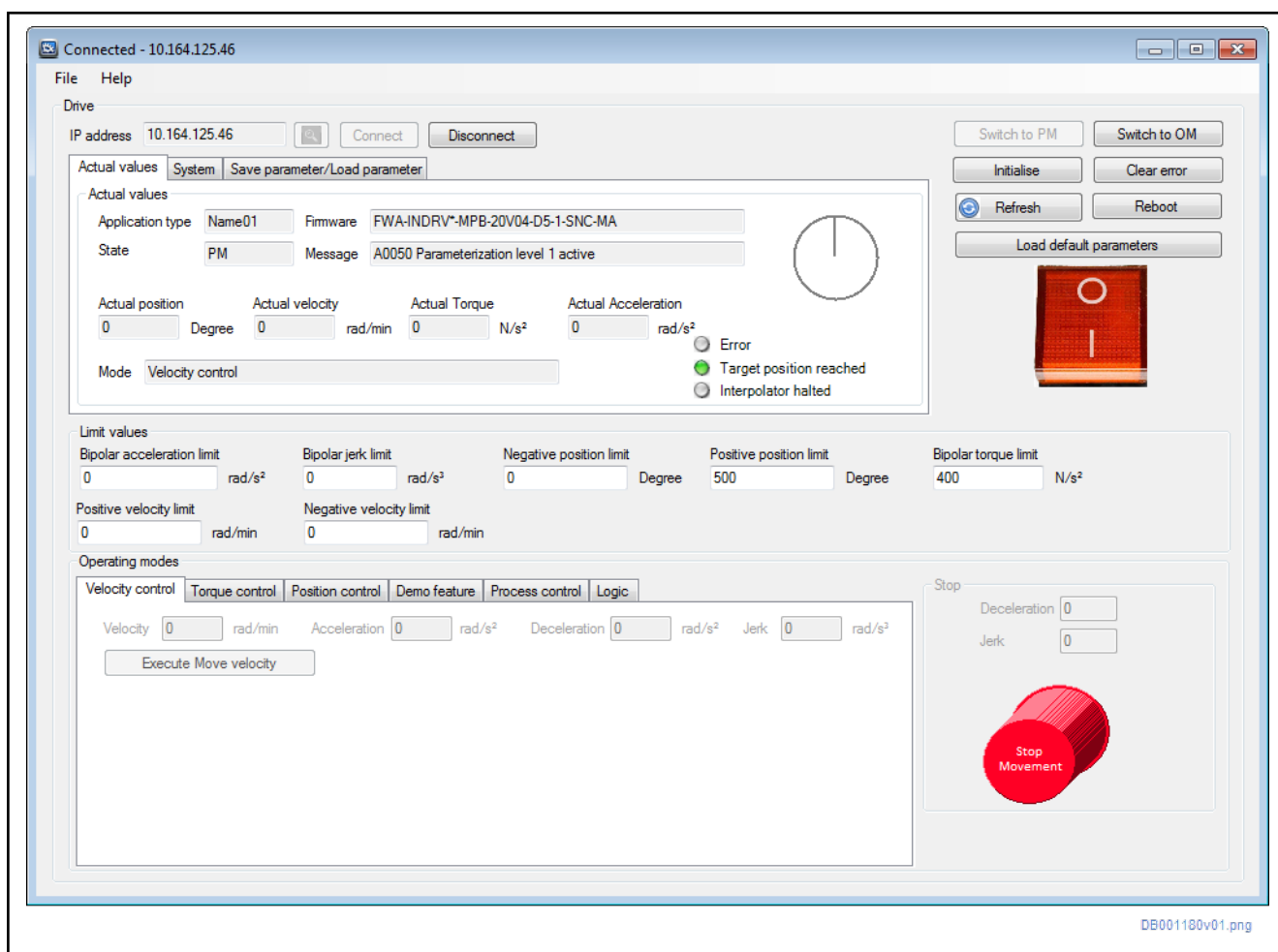


Abb. 6-10: "DriveTool"

Microsoft Excel-Datei als Benutzeroberfläche

Das folgende Beispiel zeigt die Integration von "Open Core Interface für Antriebe" in die Microsoft-Office-Anwendung Excel.

Mit der Excel-Benutzeroberfläche ist der Zugriff auf folgende Funktionen möglich:

- Verbindung mit einem Antrieb über die Ethernet-IP-Adresse herstellen
- Leistung zuschalten
- Schalten zwischen Betriebs- und Parametriermodus
- Schreiben von Motor-Geschwindigkeit, -Beschleunigung, -Verzögerung und -Ruck

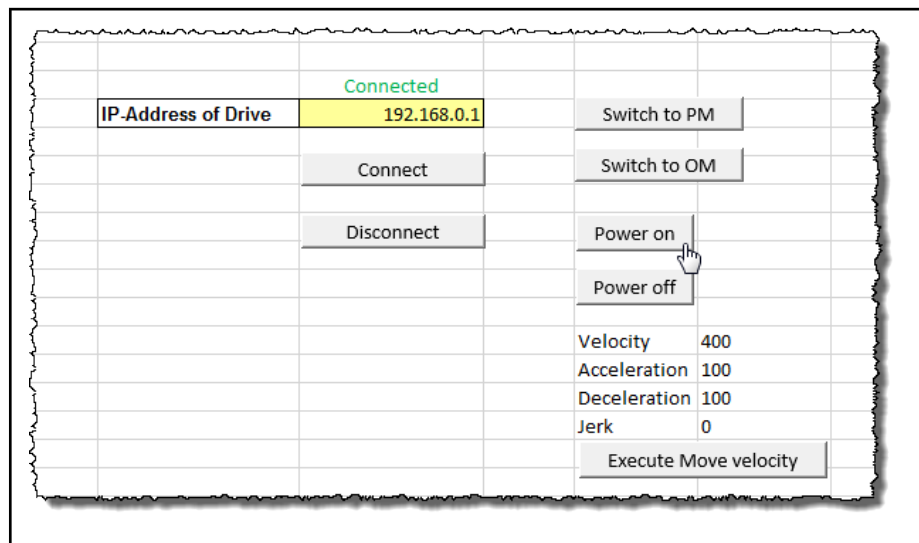


Abb. 6-11: Excel-Benutzeroberfläche

Windows-Anwendung mit Microsoft Visual Studio zum Aufzeichnen von Geräte-Signalen

In der Entwicklungsumgebung Microsoft Visual Studio wurde die Windows-Anwendung "Oscilloscope" entwickelt.

In der Anwendung können Sie unter anderem das aufzuzeichnende Signal, die Abtastrate und den Trigger konfigurieren, um so für Inbetriebnahme, Service und Test von Antriebsregelgeräten zu dienen.

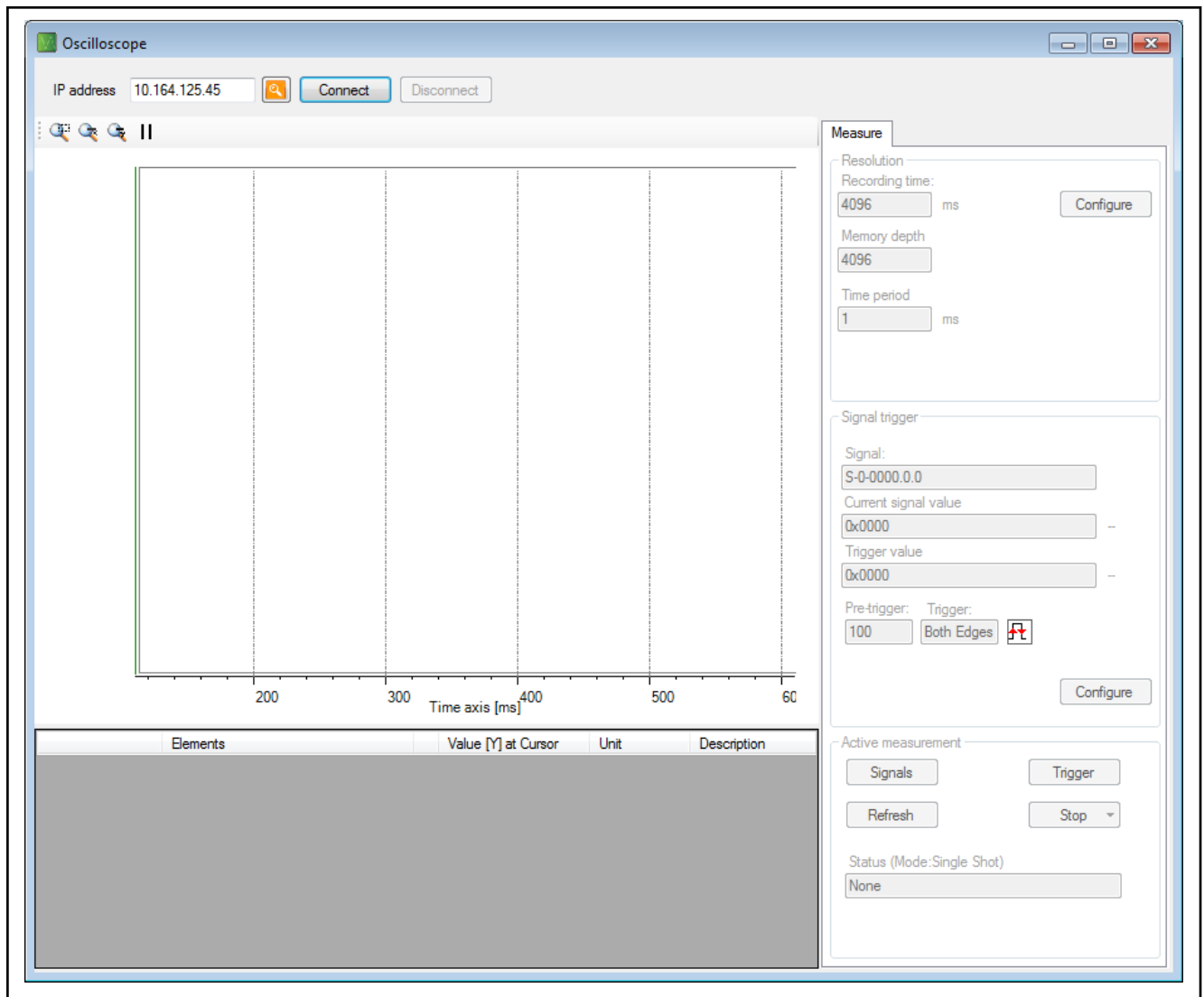


Abb. 6-12: Windows-Anwendung "Oscilloscope"

Windows-Anwendung mit Microsoft Visual Studio zum Archivieren und Wiederherstellen von Firmware- und Parameterdateien

In der Entwicklungsumgebung Microsoft Visual Studio wurde die Windows-Anwendung "Download tool" entwickelt.

Mit der Anwendung "Download tool" können Sie Firmware- und Parameterdateien von Antriebsregelgeräten der Produktfamilie "IndraDrive" archivieren und wiederherstellen.

Das "Download tool" kann im Konsolenmodus und im GUI-Modus mit grafischer Benutzeroberfläche betrieben werden.

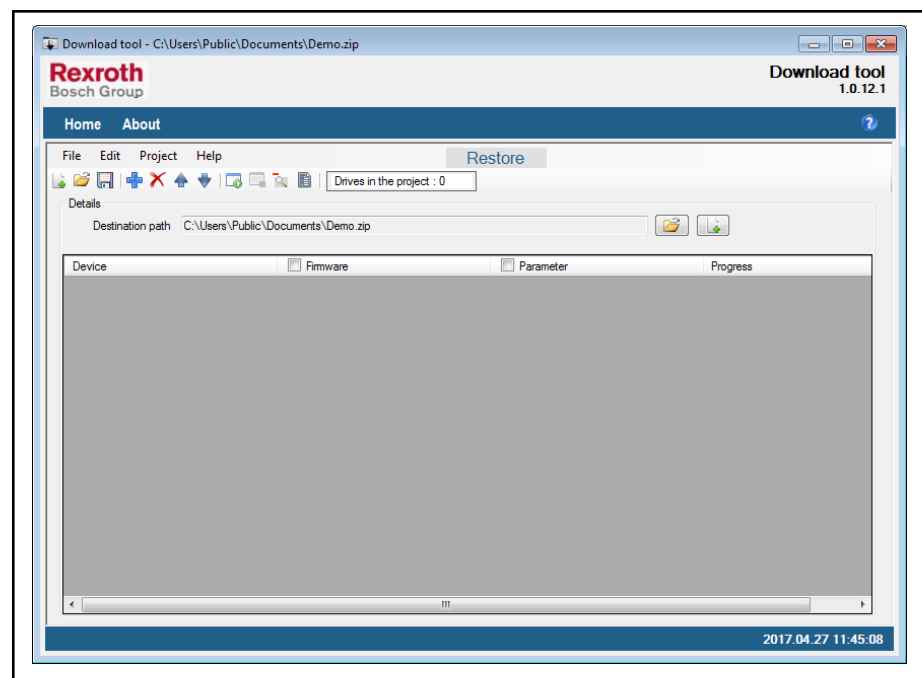


Abb. 6-13: *Windows-Anwendung "Download tool" im GUI-Modus mit grafischer Benutzeroberfläche*

Android-App

Für Android werden mit dem SDK verschiedene Demo-Apps mit verschiedenen Funktionalitäten zur Verfügung gestellt:

- Verfahrbewegung
- Oszilloskopfunktion
- Parametersicherung
- Firmware-Update
- ...

Die Demo-Apps werden als APK (Android Package File) zur Verfügung gestellt.

Um die Demo-Apps installieren zu können, müssen Sie die Installation von Apps erlauben, die nicht aus dem Google Play Store kommen.

1. Öffnen Sie auf Ihrem Smartphone die Einstellungen und setzen bei **Anwendungen** ► **Unbekannte Quellen** ein Häkchen (bei manchen Geräten unter **Sicherheit** ► **Unbekannte Quellen**).

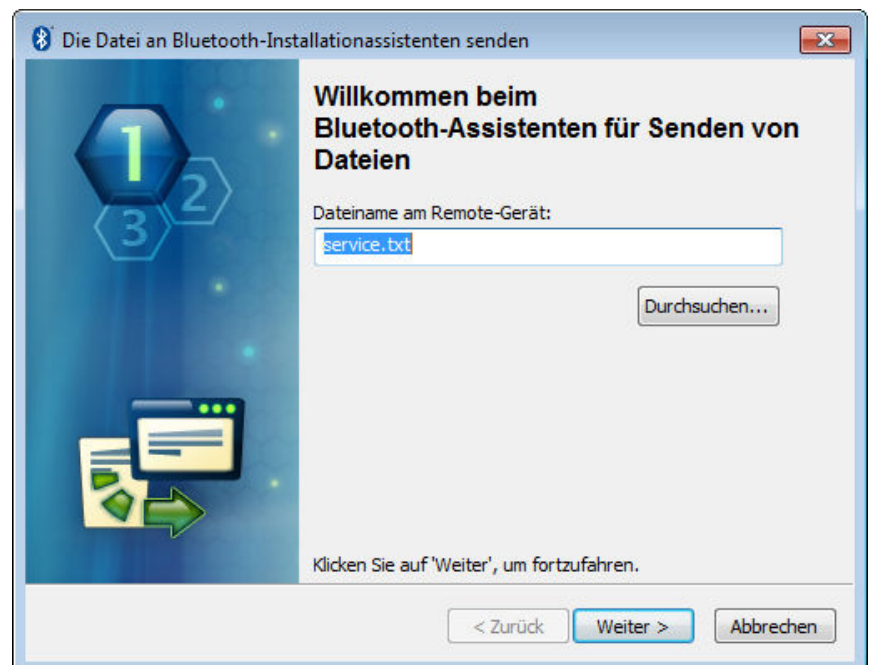
VORSICHT: Das Häkchen stellt auch ein Sicherheitsrisiko dar, da Apps aus unbekannten Quellen Viren enthalten können.

⇒ Installieren Sie nur Apps aus vertrauenswürdigen Quellen.

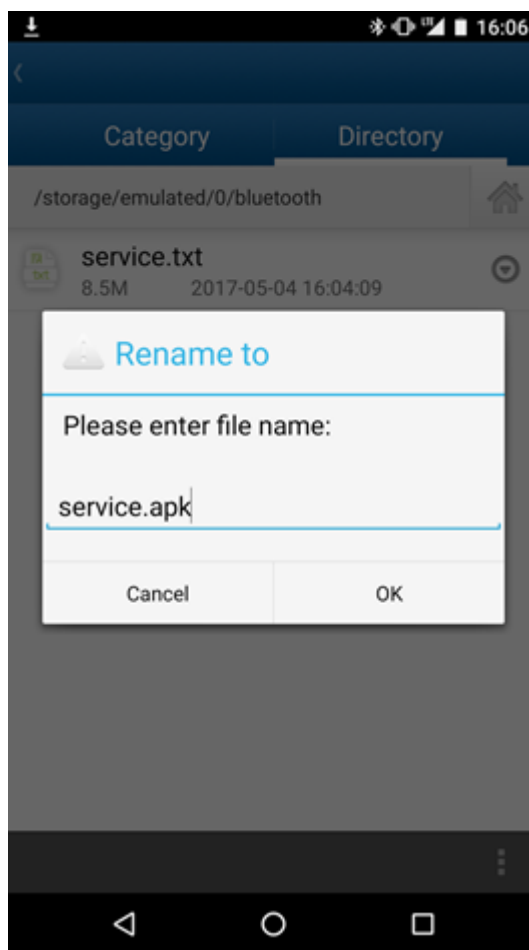
⇒ Standardmäßig sollte das Häkchen deaktiviert sein.

2. Die einfachste Möglichkeit, ein APK von einem Computer auf Ihr Smartphone zu übertragen, ist über USB-Kabel.

Ebenfalls möglich ist die Übertragung per Bluetooth, dafür müssen Sie die Dateiendung *.apk vor der Übertragung umbenennen (beispielsweise in *.txt).



3. Nach der Übertragung muss die Datei auf dem Smartphone wieder in *.apk umbenannt werden.



4. Installieren Sie die APK auf Ihrem Smartphone.



Um "Open Core Interface für Antriebe" auf dem Smartphone nutzen zu können, **müssen Sie in jedem Fall zu jeder Applikations-APK das APK "service.apk" installieren.** "service.apk" finden Sie in dem SDK in dem Unterverzeichnis "lib" des EAL4Android-Verzeichnisses.

Nach der Installation von "service.apk" finden Sie die App unter **Einstellungen ► Anwendungen** als "EAL Service".

Verfahrbewegung über Eingabe von Sollwerten

In der "Demo App" (demoapp.apk) werden die Verfahrbewegungen über die Eingabe von Sollwerten (**MOVE VELOCITY/MOVE FREQUENCY**) ausgeführt. Außerdem können Grenzwerte festgelegt werden.

Hier die "Demo App" (demoapp.apk) mit den Schritten "Verbindung zum Antriebsregelgerät herstellen" und "Aufrufen der Eingabemaske für Geschwindigkeitsparameter":

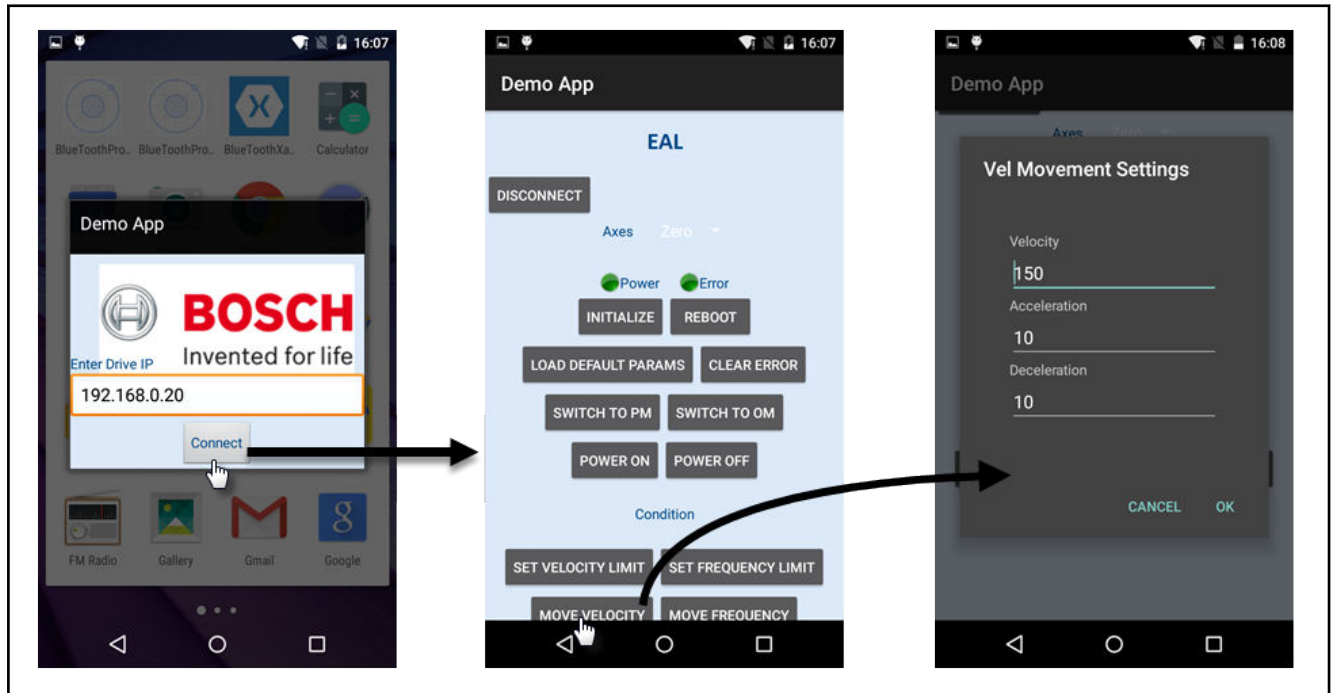


Abb. 6-14: "Verbindung zum Antriebsregelgerät herstellen" und "Aufrufen der Eingabemaske für Geschwindigkeitsparameter" in der "Demo App"

Verfahrbewegung über Dreh- und Kippbewegung des Smartphones

Um eine Verfahrbewegung des Antriebs auszuführen, können auch die Beschleunigungs- und Lage-Sensoren des Smartphones genutzt werden. Die nachfolgende Abbildung zeigt eine App, bei der der Antrieb durch Dreh- und Kippbewegungen des Smartphones verfahren wird:

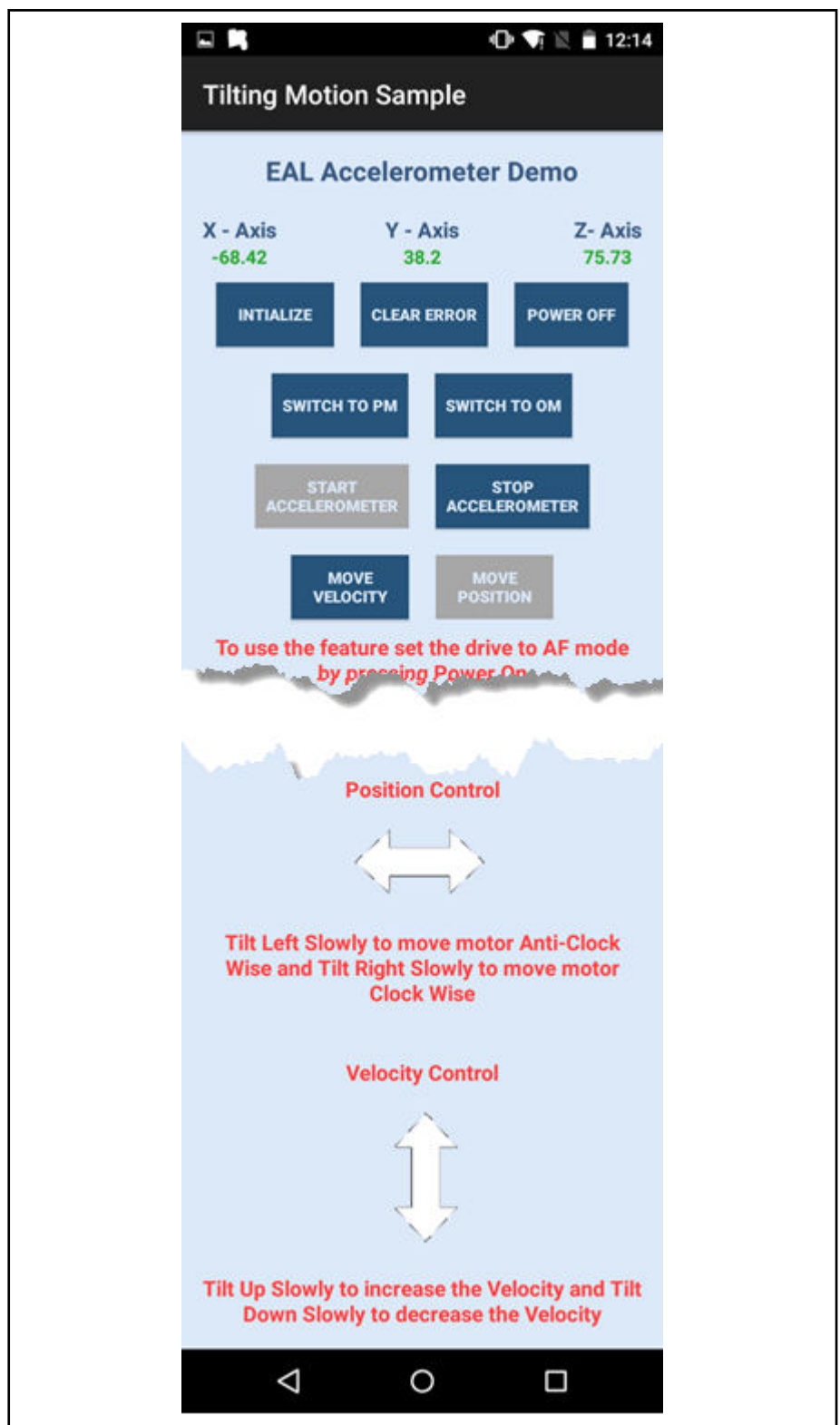


Abb. 6-15:

Demo-App: Verfahrenbewegungen über Dreh- und Kippbewegung des Smartphones

7 Inbetriebnahme- und Anwendungshinweise

7.1 Voraussetzungen für die Nutzung von Rexroth IndraMotion MLD

7.1.1 Firmware- und Hardware-Voraussetzungen

Siehe "Systemübersicht", "[Firmware-Voraussetzungen](#)" bzw. "[Hardware-Voraussetzungen](#)".

7.1.2 Software-Voraussetzungen

Mit der Installation der Inbetriebnahmesoftware IndraWorks MLD wird automatisch das Entwicklungssystem für speicherprogrammierbare Steuerungen (z. B. IndraMotion MLD) zur Verfügung gestellt.

Im Detail besteht das Entwicklungssystem aus folgenden Software-Komponenten:

- **Logic-Programmiersoberfläche**
Die Logic-Programmiersoberfläche enthält die SPS-Editoren und Debugger.
- **IndraLogic OPC Server**
Der OPC-Server wird i. d. R. von 3rd Party-Visualisierungen benötigt.
- **IndraWorks-Packages**
IndraWorks-Packages enthalten Informationsdateien und alle notwendigen Bibliotheken für die antriebsintegrierte SPS Rexroth IndraMotion MLD. Alle notwendigen Geräteunterstützungen (IW-Packages) samt Bibliotheken werden automatisch installiert, so dass keine speziellen Softwarekomponenten ausgewählt werden müssen.
- **IndraLogic Gateway Server**
Der "IndraLogic Gateway Server" stellt die Kommunikation für das Programmiersystem und ggf. 3th Party-Applikationen zur Verfügung.
- **ENI-Server**
Der ENI-Server dient zur Kommunikation zwischen IndraWorks und IndraLogic bzw. fungiert als Server für eine Quellcodeverwaltung.

Die Geräteunterstützungen (IW-Packages) können auch separat nachinstalliert werden. Dies kann während der Einführung neuer Produktversionen in deren Prototypphase notwendig werden. Der entsprechende Dialog kann im Projektextplorer über das Kontextmenü **MLD ► IndraLogic Geräteversion ändern** aufgerufen werden.

In dem Dialog "IndraLogic Geräteversion ändern" können IndraWorks-Packages durch Anklicken des Links "IW-Package hinzufügen" nachinstalliert werden.



Bei einem Update von IndraWorks bleiben alle vorhandenen Bibliotheken auf dem Rechner verfügbar. Es ist nicht notwendig ein "älteres" IndraWorks-Package erneut zu installieren.

7.1.3 Voraussetzungen für die Kommunikation

Wenn mit der antriebsintegrierten SPS (MLD) ausschließlich fertige Technologiefunktionen genutzt werden, dann ist es nicht notwendig eine Verbindung zwischen IndraLogic und dem Antrieb herzustellen. Lediglich zur Projektierung des Antriebs ist eine Kommunikation mit IndraWorks MLD notwendig.

Sollen eigene Funktionen genutzt werden, so ist bei der Antriebsfamilie "IndraDrive Cs" für die Programmierung der antriebsintegrierten SPS (MLD) eine Ethernetverbindung notwendig.

Die Verbindungsaufnahme mit angeschlossenen Geräten wird von IndraWorks MLD unterstützt. Die Auswahl erfolgt über die Menüleiste **Projekt ► Nach Geräten suchen....**

Nach der Verbindungsaufnahme mit IndraWorks MLD kann in IndraLogic direkt online gegangen werden (**Online ► Einloggen**). Die Kommunikationseinstellungen werden automatisch vorgegeben; es sind keine weiteren Einstellungen notwendig.

7.2 Programmerstellung in IndraWorks MLD

In IndraWorks MLD wird die Antriebsfunktionalität in einem Gerätebaum dargestellt. Die notwendigen Dialoge zur Parametrierung des Antriebs und der MLD sind über den Knoten **MLD** erreichbar.

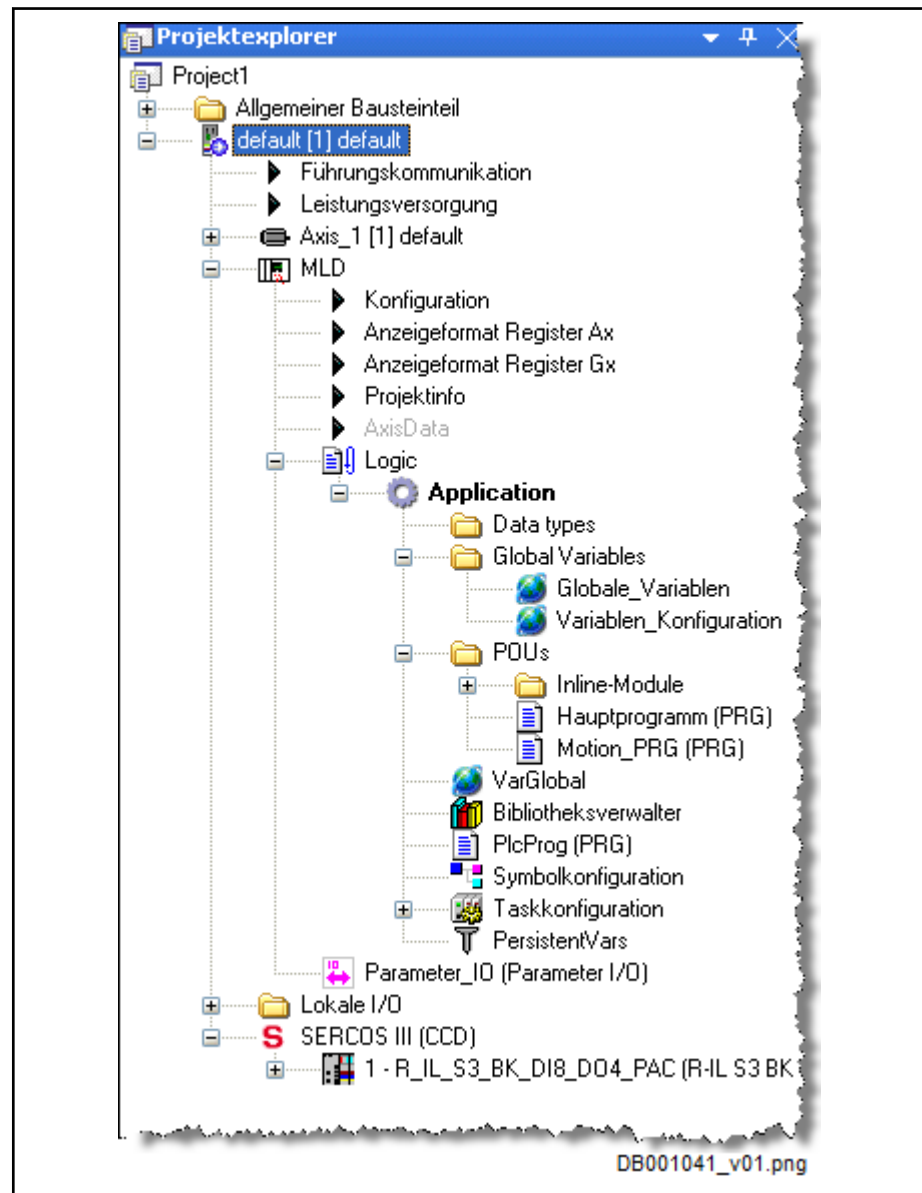


Abb. 7-1: Projektexplorer mit freigeschalteter MLD



Die MLD-spezifischen Knoten im Projektbaum sind nur bei einer Freischaltung des Funktionspakets sichtbar!

Durch einen Doppelklick auf den Logikknoten wird ein Dialog geöffnet, in dem ausgewählt werden kann, wie das Programm in IndraWorks MLD erstellt werden soll:

- **Freie Programmierung**
 - Globale Variablenliste, Symbolkonfiguration und Datentypen, Defaultprogramm (PlcProg)
 - Globale Variablenliste, Symbolkonfiguration und Datentypen
- **Bestehende Projekte verwenden**
 - Aus IndraLogic 1G-Projekt erstellen
 - Aus Exportdatei oder CoDeSys 3.x-Projekt erstellen

- **AxisInterface Beispielprojekt**

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 (siehe "[Nutzung des "AxisInterface" für die Programmierung von Rexroth IndraMotion MLD](#)").

- **GAT compact Beispielprojekt**

GAT compact (Generic Application Template) ist ein einschaltfertiger Programmierrahmen, in dem der Applikationscode an vordefinierten Stellen eingetragen wird. Durch die dialogbasierte Codegenerierung des GAT-Wizards werden professionell gut strukturierte Applikationsprogramme mit vorbereiteten Elementen erstellt. GATcompact ist ab IndraWorks 14V14 mit MPx20 verfügbar (siehe "[Automatisierte Programmerstellung mit GAT compact](#)").

Außerdem ist die Nutzung von Technologiefunktion möglich. Es handelt sich hierbei um ein komplettes (übersetztes) SPS-Projekt, welches sich aus mehreren Bausteinen (IEC-konform) zusammensetzen kann (siehe "[Nutzung von Technologiefunktionen](#)").

7.2.1 Nutzung von Technologiefunktionen

Übersicht

Ein komplettes (übersetztes) SPS-Projekt, welches sich aus mehreren Bausteinen (IEC-konform) zusammensetzen kann, wird als Technologiefunktion bezeichnet.

Ein Projekt wird durch Kombination von IEC- oder Firmwarebausteinen erstellt und bietet meist eine komplexe und umfangreiche Funktionalität.



Für einige Technologiefunktionen gibt es spezielle Dialoge in der Inbetriebnahmesoftware IndraWorks. Die Dialoge unterstützen die Inbetriebnahme und die Bedienung der Technologiefunktionen. Zusätzlich existiert zu jeder Technologiefunktion jeweils eine Anwendungsbeschreibung.

Um eine Technologiefunktion nutzen zu können, existieren folgende Möglichkeiten:

Die Technologiefunktion ist über Inbetriebnahmesoftware (z.B. IndraWorks) ladbar als Parameterfile

- Die Technologiefunktion wird als Paket geliefert, welches das SPS-Projekt und ggf. noch zusätzliche Parameter enthält (Parameterfile ohne Quellcode).
- Es ist kein Zugriff auf den SPS-Quellcode über IndraLogic möglich.
- Es gibt keine Debug-Funktion und keine Programmiermöglichkeit.
- Die Handhabung ist vergleichbar mit einer fest programmierten Firmwarefunktion.

Technologiefunktion als Bibliothek (einzelne Funktionsbausteine)

- Die Bibliothek enthält keinen Quellcode (der Zugriff ist passwortgeschützt).
- Die einzelnen Funktionsbausteine sind in Gruppen zusammengefasst (Bibliotheken) [z.B. variable Fehlerreaktionen (Rückzugsbewegung,...)].
- Die volle Debug-Funktion steht zur Verfügung und es gibt die Möglichkeit zu programmieren (ähnlich wie "DISC"-Makros bei der Antriebsfamilie EcoDrive).

Beteiligte Parameter Folgende Parameter werden im Zusammenhang mit Technologiefunktionen verwendet:

- P-0-1350, SPS Steuerwort
- P-0-1351, SPS Statuswort
- P-0-1352, SPS Anwenderprogramm Verwaltungsdaten
- P-0-1353, SPS Anwenderprogramm Bereich 0
- P-0-1354, SPS Anwenderprogramm Bereich 1
- P-0-1355, SPS Anwenderprogramm Bereich 2
- P-0-1356, SPS Anwenderprogramm Bereich 3
- P-0-1357, SPS Anwenderprogramm Bereich 4
- P-0-1358, SPS Anwenderprogramm Bereich 5
- P-0-1360, SPS Programmkennung
- P-0-1361, SPS Programmname
- P-0-1367, SPS Konfiguration
- (evtl. noch weitere Registerparameter der SPS)

Inbetriebnahme und Programmierhinweise

Technologiefunktionen laden und aktivieren

Das Laden einer Technologiefunktion kann auf zwei Arten erfolgen:

- Laden als Parameterfile über die Inbetriebnahmesoftware (z.B. IndraWorks. Das Menü zur Verwaltung der Technologiefunktionen wird durch Doppelklick auf "Technologiefunktionen" im Ordner MLD geöffnet) (vgl. "P-0-1353, SPS Anwenderprogramm Bereich 0" bis "P-0-1358, SPS Anwenderprogramm Bereich 5"). In diesem Fall muss lediglich die gewünschte Technologiefunktion geladen und anschließend parametrisiert werden (die Parametrierung muss entsprechend den fest programmierten Funktionen erfolgen).
- Laden des Projektes über IndraLogic

Technologiefunktionen

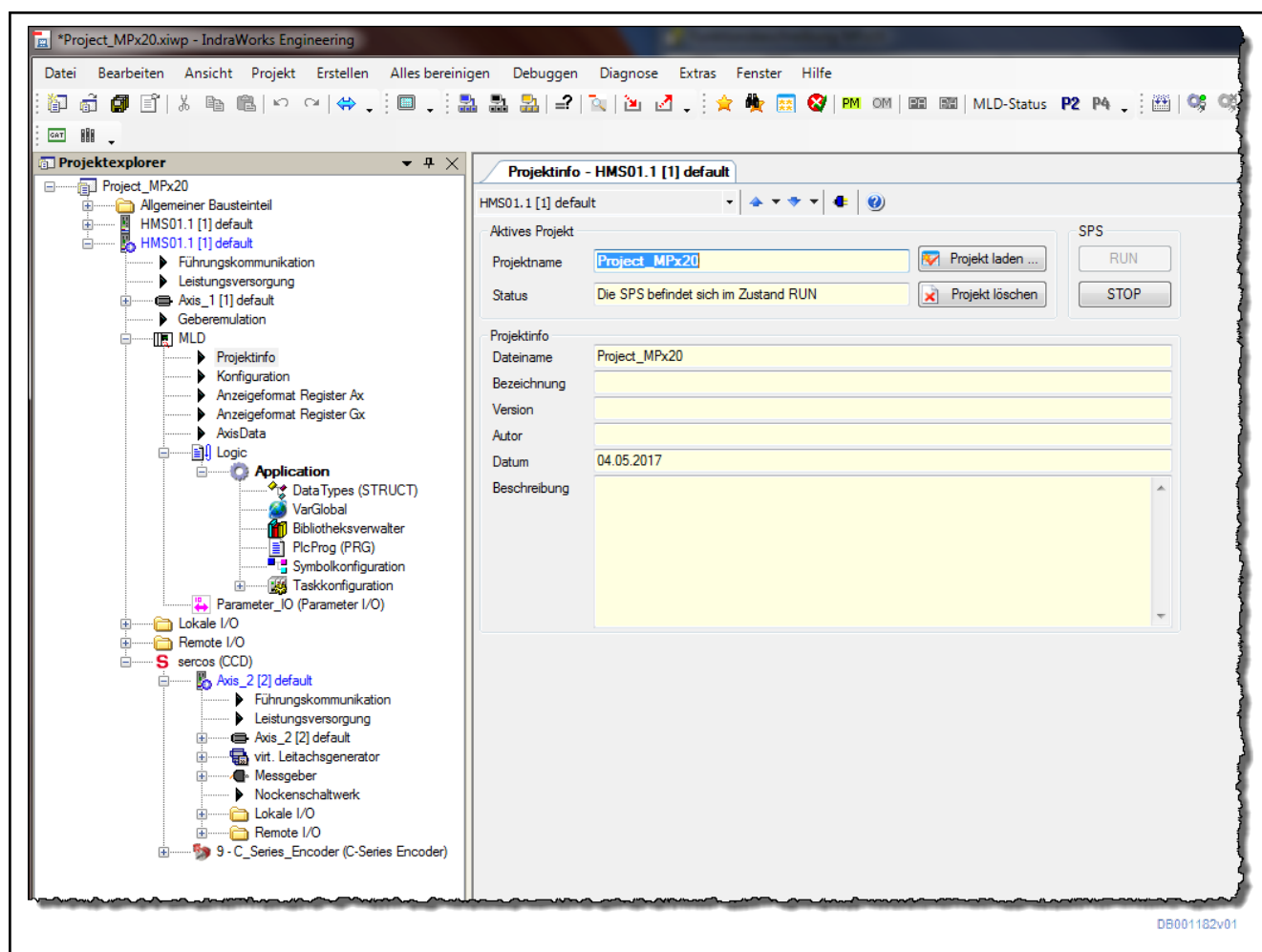


Abb. 7-2: Technologiefunktionen

Die Technologiefunktionen werden als vorbereitetes Paket geliefert. Dieses besteht im Wesentlichen aus einer Datei mit der Endung "*.spj". Die Datei enthält ein komplettes SPS-Projekt in übersetzter Form; Quellcode oder SPS-Variablen sind nicht sichtbar. Mit obigem Dialog kann eine Technologiefunktion in den Antrieb geladen werden.



Unabhängig davon, wie die Technologiefunktion in den Antrieb geladen wird (über IndraWorks oder IndraLogic), wird sie als Projekt in den Parametern (P-0-1352, P-0-1353, P-0-1354, P-0-1355, P-0-1356,... und ggf. P-0-1367) gespeichert, so dass sie bei einer Parametersicherung enthalten ist.



Technologiefunktionen sind überwiegend so erstellt, dass sie beim Booten des Antriebs automatisch gestartet werden (vgl. "P-0-1367, SPS Konfiguration"). Ist dies nicht der Fall, so können sie nach dem Laden des Projekts über "P-0-1350, SPS Steuerwort" aktiviert werden.

Parametrierung bzw. Inbetriebnahme

Die Parametrierung der Technologiefunktion kann auf verschiedene Arten erfolgen:

- über eine Inbetriebnahmesoftware (z. B. IndraWorks MLD) mittels eigens für die Technologiefunktion erstellten Dialogen (z. B. "Präventive Diagnose")
- über die Möglichkeit der Visualisierung in IndraLogic (siehe auch "[Visualisierung, Engineering und Anbindung über OCI](#)")
- über einen Screen des eigens dafür programmierten Bedien- und Visualisierungsgeräts, welches die Zugriffe auf die SPS-Parameter ermöglicht
- über eine eventuell vorhandene übergeordnete Steuerung



Die für die Technologiefunktion spezifischen Einstellungen und Parametrierungen sind der jeweiligen Dokumentation zu entnehmen.

Diagnose- und Status-Informationen

Zur Diagnose von SPS-Projekten (Technologiefunktion) stehen folgende Parameter zur Verfügung:

- P-0-1360, SPS Programmkenennung
- P-0-1361, SPS Programmname
- P-0-1351, SPS Statuswort

7.2.2 Freie Programmierung von Rexroth IndraMotion MLD

Verbindung mit dem Antrieb

Allgemeines Die Kommunikation zwischen der SPS-Programmieroberfläche IndraLogic und dem Antrieb wird über die Inbetriebnahmesoftware IndraWorks MLD bereitgestellt.

Die Einstellung der Verbindung erfolgt in IndraWorks MLD. Hier kann z.B. mit "Nach Geräten suchen" die Verbindung automatisch eingestellt werden.

IndraLogic wird aus IndraWorks MLD über Doppelklick auf den Knoten "Logic" in dem Projektexplorer gestartet. IndraLogic übernimmt dabei automatisch die Kommunikationseinstellung aus IndraWorks MLD; der IndraLogic-Dialog zur Einstellung der Kommunikationsparameter wird nicht benötigt.



IndraLogic soll nur aus IndraWorks MLD heraus gestartet werden; ein Betrieb von IndraLogic ohne die Einbettung in IndraWorks MLD wird nicht unterstützt.

Ethernet Schnittstelle (TCP/IP) Auf dem Antrieb wird der Ethernet-Engineering-Port (HCS01) verwendet. Zur Kommunikation wird eine Ethernetschnittstelle mit TCP/IP-Protokoll benötigt. Die Einstellungen für die TCP/IP-Kommunikation können in IndraWorks über das Kontextmenü des Antriebs vorgenommen werden (beteiligte Parameter: P-0-1531, P-0-1532 und P-0-1533).

MLD-Einstellungen in IndraWorks

Voraussetzungen Vor der Programmierung der MLD sollte der Antrieb bereits fertig in Betrieb genommen sein.

Einstellungen Die Einstellungen der Funktion "IndraMotion MLD" können unter dem Knoten "MLD" durchgeführt werden. Nachfolgend die Möglichkeiten, die IndraWorks im Projektexplorer zur Verfügung stellt:

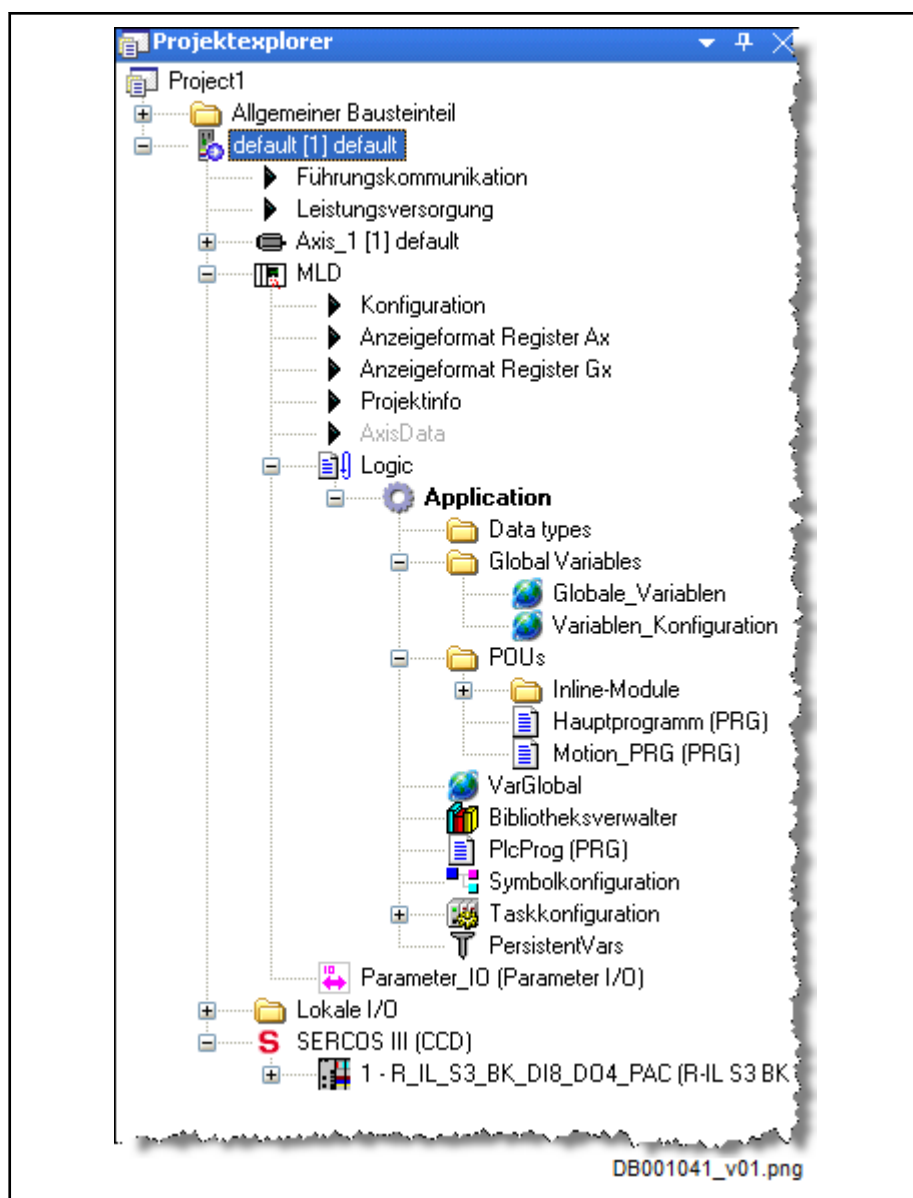


Abb. 7-3: Ausschnitt aus dem Projektexplorer von IndraWorks



Der Dialog "Konfiguration" kann nur aufgerufen werden, wenn eine Verbindung zum Antrieb aufgebaut, oder in IndraLogic eine Offline-Simulation gestartet wurde.

Kontextmenü "MLD"

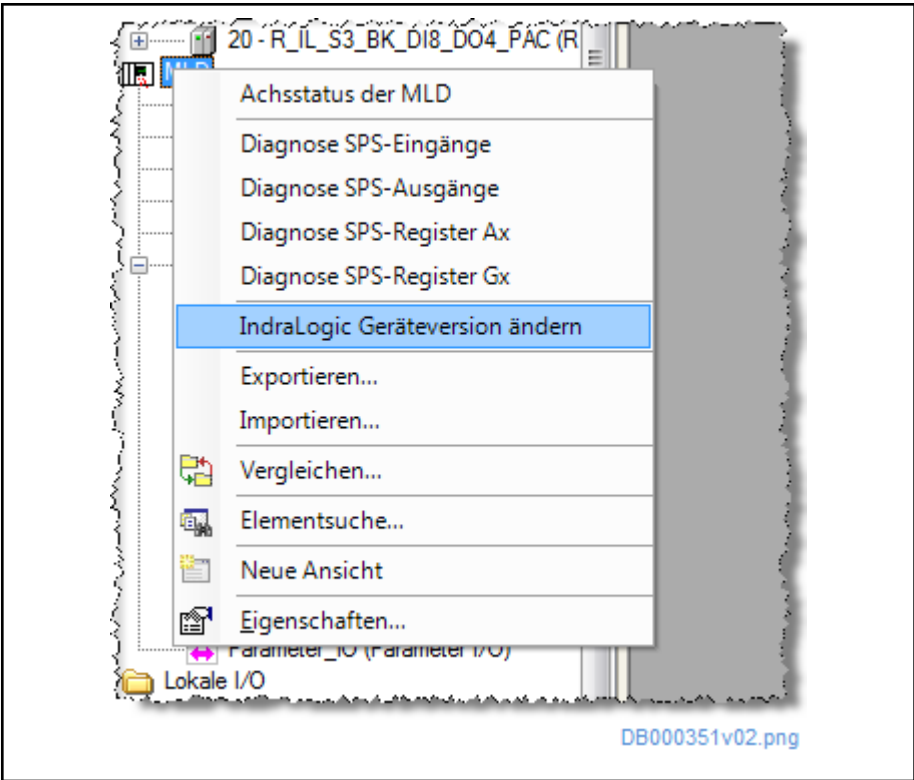


Abb. 7-4: Kontextmenü "MLD"

Achsstatus der MLD

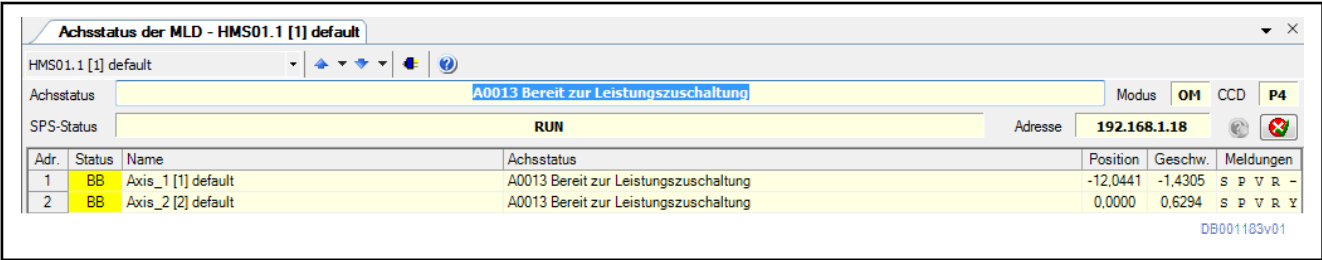


Abb. 7-5: Achsstatus eines MLD-M CCD-Verbunds

Diagnose SPS-Eingänge

In diesem Dialog kann der Zustand des Prozessabbilds der Eingänge kontrolliert werden. Eingänge, die der SPS zugewiesen sind, werden hier angezeigt.

i **Optimierung der Eingänge in IndraLogic:** Eingänge, die im SPS-Programm nicht verwendet werden, werden in IndraLogic nicht angezeigt. In diesem Dialog wird das gesamte Prozessabbild der MLD unabhängig von der Nutzung im Programm angezeigt.

Diagnose SPS-Ausgänge

In diesem Dialog kann der Zustand des Prozessabbilds der Ausgänge kontrolliert werden. Ausgänge, die der SPS zugewiesen sind, werden hier angezeigt.

Diagnose SPS-Register Ax

In diesem Dialog sind die Inhalte der SPS-Register A0...A31, sowie der SPS-Register AT0 und AT1 dargestellt. Der jeweilige Registerinhalt des SPS-Registers A0...A31 wird in dem Format dargestellt, wie er unter "Anzeigeformat

Register Ax" eingestellt wurde. Die Inhalte der SPS-Register können mit Hilfe des Dialogs geändert werden.

Die SPS-Register AT0 und AT1 sind Textregister und können je max. 255 Zeichen beinhalten.

Das SPS-Register AL0 ist ein Listenregister mit 8192 Elementen (je 4 Byte). Das in "Anzeigeformat Register Ax" eingestellte Format gilt für alle 8192 Elemente.



Alle SPS-Register Ax sind nicht gepuffert; d. h. bei Spannungsausfall gehen die Registerinhalte verloren.

Diagnose SPS-Register Gx

In diesem Dialog sind die Inhalte der SPS-Register G0...G31, sowie der SPS-Register GL0...GL2 dargestellt. Der jeweilige Registerinhalt des SPS Registers G0..G31 wird in dem Format dargestellt wie er unter "Anzeigeformat Register Gx" eingestellt wurde. Die Inhalte der SPS-Register können mit Hilfe des Dialogs geändert werden.

Die SPS-Register GL0...GL2 sind Listenregister mit 1024 Elementen (je 4 Byte). Das in "Anzeigeformat Register Gx" eingestellte Format gilt jeweils für alle 1024 Elemente.



Alle SPS-Register Gx sind gepuffert; d. h. die Registerinhalte bleiben bei Spannungsausfall erhalten.

IndraLogic Geräteversion ändern

Mit der Funktion "IndraLogic Geräteversion ändern" kann die Geräteversionen (IndraWorks-Package) geändert oder eine neue Geräteversionen hinzugefügt werden. Geräteversionen (IndraWorks-Packages) müssen nur dann geändert werden, wenn schwerwiegende Fehler behoben worden sind. Im Allgemeinen werden geänderte Geräteversionen zur Releaseänderung des IndraWorks mitinstalliert.

Exportieren / Importieren

Hier werden SPS-Programmteile exportiert bzw. importiert. Die Endung der Standardexport-Datei ist "*.iwx". Es können beim Import nur entsprechende Dateien importiert werden.

Vergleichen

Die Funktion "Vergleichen" ermöglicht es, zwei Projektdateien miteinander zu vergleichen. Es müssen die Quell- und die Zieldatei angegeben werden.

Elementsuche

Die Funktion "Elementsuche" ermöglicht es, in einem SPS-Projekt nach Wörtern / Ausdrücken zu suchen.

Konfiguration

Durch Doppelklick auf "Konfiguration" wird ein Dialog aufgerufen, in dem das Startverhalten der MLD und einige Einstellungen zur Motionsteuerung vorgenommen werden können.



Abb. 7-6: Dialog zur Konfiguration

- Bei "Startverhalten der SPS" muss definiert werden wie sich die antriebsintegrierte SPS in der Einschaltsequenz verhält:
 - "Default-Startverhalten"

Die SPS behält ihren Zustand bei. Das bedeutet, sie startet beim Booten des Antriebs, wenn sie vor dem letzten Ausschalten im Zustand RUN war. Ansonsten verbleibt sie im Zustand STOP.
 - "Start beim Booten"

Die SPS startet unabhängig vom letzten Zustand (STOP/RUN) immer beim Booten.
 - "Stop"

Die SPS bleibt unabhängig vom letzten Zustand in STOP.
 - "Start im Betriebsmodus"

Die SPS startet erst nachdem der Antrieb sich im Betriebsmodus befindet.
- Mit "SPS hat permanente Kontrolle über den Antrieb" kann der Antrieb mit MLD als autarker Motion Controller für Steuerungsaufgaben (Kontrollkästchen angewählt) oder als intelligente Servoachse zur Erweiterung der Antriebsfunktionalität (Kontrollkästchen **nicht** angewählt) konfiguriert werden.



Mit der Konfiguration "SPS hat permanente Kontrolle über den Antrieb" wird auch automatisch der Speicherungsmodus so eingestellt, dass Parameteränderungen nach dem Ausschalten der Steuerspannung verloren gehen (S-0-0269="1").

- Mit "Motionfehler am FB lösen keine (Achs-)Fehler aus" kann konfiguriert werden, dass bei Fehlern, die bei der Kommandierung der Achsen von den zugehörigen SPS-Bausteinen erkannt werden, ein Antriebsfehler F2150 generiert wird (Kontrollkästchen angewählt). Soll keine automatische Reaktion des Antriebs erfolgen, dann darf das Kontrollkästchen nicht angewählt sein.
- Kontrollkästchen "Unterstützung der AxisData-Struktur": Für Motionaufgaben gibt es eine optionale globale Struktur "AxisData", welche den Zugriff auf zyklische Daten der Achsen vereinfacht. Die Unterstützung dieser Datenstruktur über alle Achsen verbraucht permanent Rechenzeit und kann bei Bedarf ein- oder ausgeschaltet werden.
- Mit dem Link "Cross Communication Drive" gelangt man auf den Dialog zu den Einstellungen für Mehrachsanwendungen. Die Funktionalität "Mehrachsanwendung" steht mit der Firmwareversion MPC18 VRS zur Verfügung.
- Anhand der Anzeige in dem Bereich "SPS Status" kann erkannt werden, ob ein SPS-Projekt geladen ist und in welchem Zustand es sich befindet.
- Über "SPS Steuerung" kann das SPS Programm gestartet ("Run") und gestoppt werden ("Stop").



Das Starten und Stoppen der SPS ist hier nur in Ausnahmefällen sinnvoll; in der Regel wird man dies im Programmiersystem IndraLogic durchführen.

Anzeigeformat Register Ax

Durch Doppelklick auf "Anzeigeformat Register Ax" wird der gleichnamige Dialog aufgerufen. Mit Hilfe dieses Dialogs können Anzeigeformat und Nachkommastellen der globalen SPS-Register A0..A31 (P-0-1270..P-0-1301) und des Listenparameters "P-0-1368, Globales SPS-Register AL0" eingestellt werden.

Bei der Auswahl der Formate SIGNED_DEC und UNSIGNED_DEC kann die Anzahl der Nachkommastellen zwischen 0..7 gewählt werden.

Anzeigeformat Register Gx

Durch Doppelklick auf "Anzeigeformat Register Gx" wird der gleichnamige Dialog aufgerufen. Mit Hilfe dieses Dialogs können Anzeigeformat und Nachkommastellen der Globalen SPS Register G0..G31 (P-0-1370..P-0-1385, P-0-1316..P-0-1331) und der Listenparameter Globales SPS-Register GL0..GL2 (P-0-1389, P-0-1311, P-0-1312) eingestellt werden.

Bei der Auswahl der Formate SIGNED_DEC und UNSIGNED_DEC kann die Anzahl der Nachkommastellen zwischen 0..7 gewählt werden.

Projektinfo

Durch Doppelklick auf "Projektinfo" wird der gleichnamige Dialog aufgerufen. Mit Hilfe dieses Dialogs werden Information (Status, Projektname/Dateiname, Erstelldatum, usw.) zum Projekt angezeigt. Mit dieser Funktion kann das SPS-Programm gestartet und gestoppt werden.

AxisData



Der Menüpunkt "AxisData" ist möglicherweise ausgegraut, er muss ggf. zuvor in "Konfiguration" freigeschaltet werden.

Durch Doppelklick auf "AxisData" wird der gleichnamige Dialog aufgerufen. Mit Hilfe dieses Dialogs können globale zyklische Daten für Motion-Programme parametrierbar werden.

"AxisData" ist bei der MLD-2G der Ersatz für den lokalen Echtzeitkanal RtcR / RtcW. Es sind 5 frei definierte Sollwerte und 6 Istwerte möglich.

Application

Der Knoten "Application" beinhaltet verschiedene Funktionen, die zum Bedienen (Online / Offline) und Editieren (Datenübernahme / Hinzufügen / Exportieren / Importieren / Vergleichen) des SPS-Programms dienen. Über einen Rechtsklick auf den Application-Knoten kann das folgende Kontextmenü geöffnet werden. Das Kontextmenü des "Logic"-Knotens ist mit dem des "MLD"-Knotens vergleichbar. Im Folgenden wird das Kontextmenü unter dem "Application"-Knoten beschrieben.

Das Kontextmenü ist unterschiedlich, je nachdem ob IndraLogic "offline" oder "online" ist:

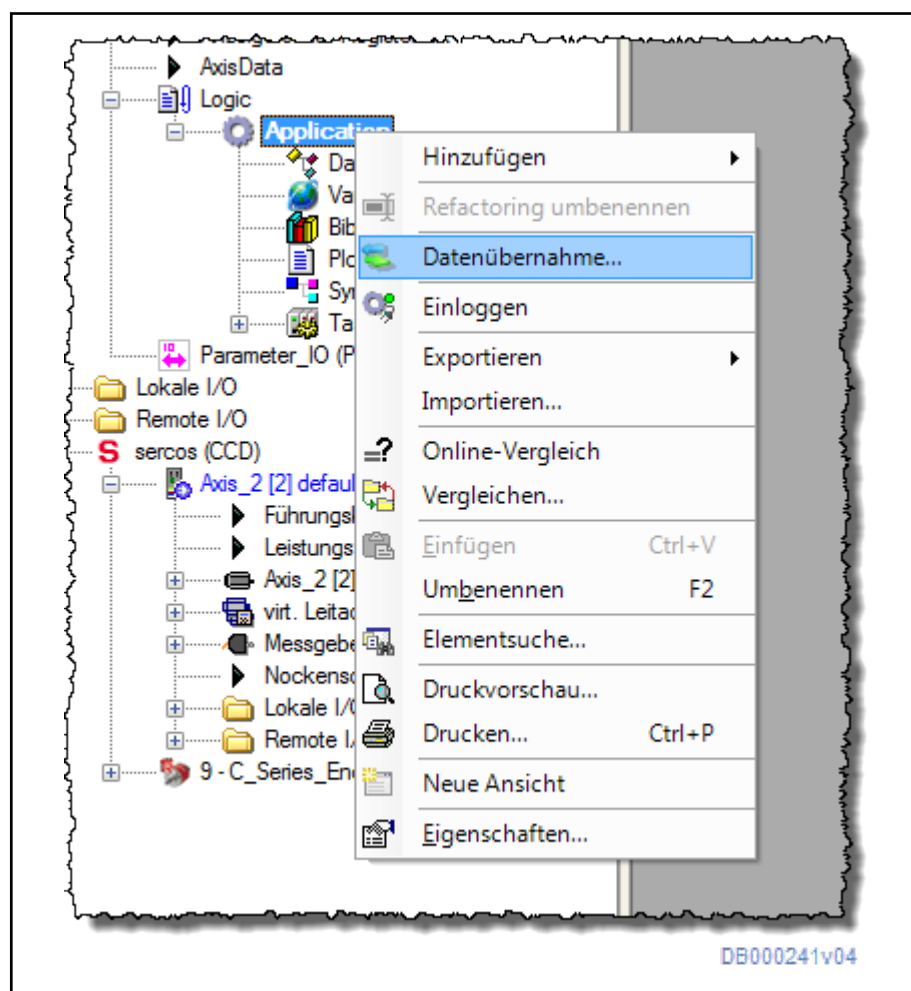


Abb. 7-7: Kontextmenü des Application-Knotens (IndraLogic ist offline)

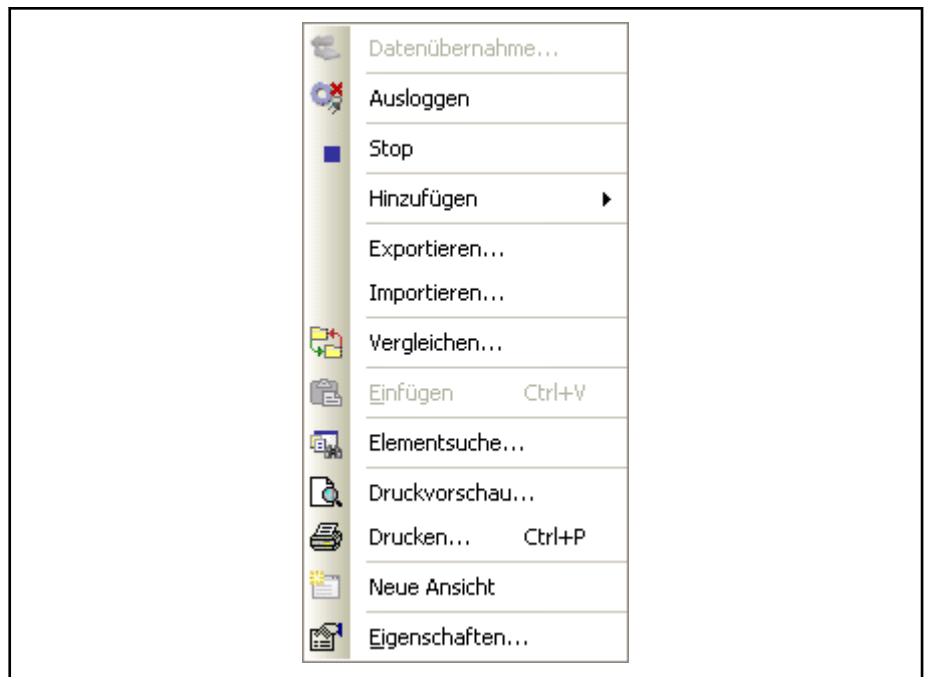


Abb. 7-8: Kontextmenü des Application-Knotens (IndraLogic ist online)

- Mit **Datenübernahme** wird ein IndraLogic-Projekt in dieses Projekt eingefügt. Es können auch Projekte, die in MLD-1G erstellt wurden, in MLD-2G eingefügt werden.
- Der 2. und 3. Menüeintrag des Kontextmenüs wechselt dynamisch. Abhängig vom Zustand des geöffneten IndraLogic-Projekts wird **Einloggen/ Ausloggen** oder **Start/ Stop** eingetragen.
 - Mit **Einloggen** wird eine Verbindung zwischen IndraLogic und dem Antrieb hergestellt (oder das Simulationsprogramm gestartet) und in den Online-Modus gewechselt. Ist zum Zeitpunkt des Kommandos "Einloggen" noch nicht alles übersetzt oder noch kein Programm in den Antrieb geladen, so werden diese Schritte automatisch ausgeführt.
Nur im eingeloggten Zustand können SPS-Projekte verändert und auf den Antrieb geladen werden.
Mit **Ausloggen** wird die Verbindung zum Antrieb beendet.
 - Mit **Start** wird die SPS in "RUN-Mode" geschaltet, mit **Stop** wird die SPS in "STOP-Mode" geschaltet.
- Mit **Hinzufügen** können dem SPS-Projekt weitere Bausteine, POU's (SPS-Teil-Programme), Tasks, Datenserver, Datentypen, usw. hinzugefügt werden.
- Mit **Exportieren / Importieren** werden SPS-Programme, Funktionsbausteine, Funktionen, usw. exportiert bzw. importiert. Die Dateierweiterung dieser Exportdateien ist "*.iwx".
- Mit **Vergleichen** können zwei SPS-Projekte miteinander verglichen werden.
- Mit **Umbenennen** können die Bezeichnungen der Elemente unter dem Knoten "Application" geändert werden.
- **Elementsuche** ermöglicht es in einem SPS-Projekt nach Wörtern / Ausdrücken zu suchen.

- Mit **Druckvorschau** kann der Ausdruck der markierten Datei überprüft werden.
- Mit **Drucken** wird eine Auswahlmöglichkeit aller im IndraLogic-Projekt enthaltenen Komponenten angeboten. Nach der Auswahl der Komponente gelangen Sie über **Weiter >>** in den nächsten Dialog, in dem die Druckseite eingerichtet werden kann. Von hier aus wird über **Fertigstellen** sofort der Druckvorgang auf dem als Standarddrucker definierten Drucker gestartet.



Weitere Druckeinstellungen, wie z. B. die Auswahl eines anderen Druckers, können Sie vornehmen, wenn Sie **Datei ▶ Druckeinstellungen...** aufrufen.

- Mit **Neue Ansicht** wird der markierte Programmteil in einem eigenen Fenster dargestellt.
 - Mit **Eigenschaften...** können Projekteinstellungen angesehen und teilweise geändert werden.
- Lokale I/O** Mit Hilfe der über den Knoten "Lokale I/O" angebotenen Dialoge können die digitalen und analogen Ein- und Ausgänge zwischen Antriebsfunktionen und MLD aufgeteilt werden. Bei der Zuteilung zur MLD wird dabei die Konfiguration in das Prozessabbild vorgenommen (siehe "[Kommunikationsschnittstellen und Datenkanäle der MLD](#)").
- Sercos** Über das Kontextmenü kann die Konfiguration der MLD als Mehrachssteuerung (MLD-M) vorgenommen werden. Die Mehrachssteuerung MLD-M ist mit der Firmwareversion MPC-18VRS verfügbar:

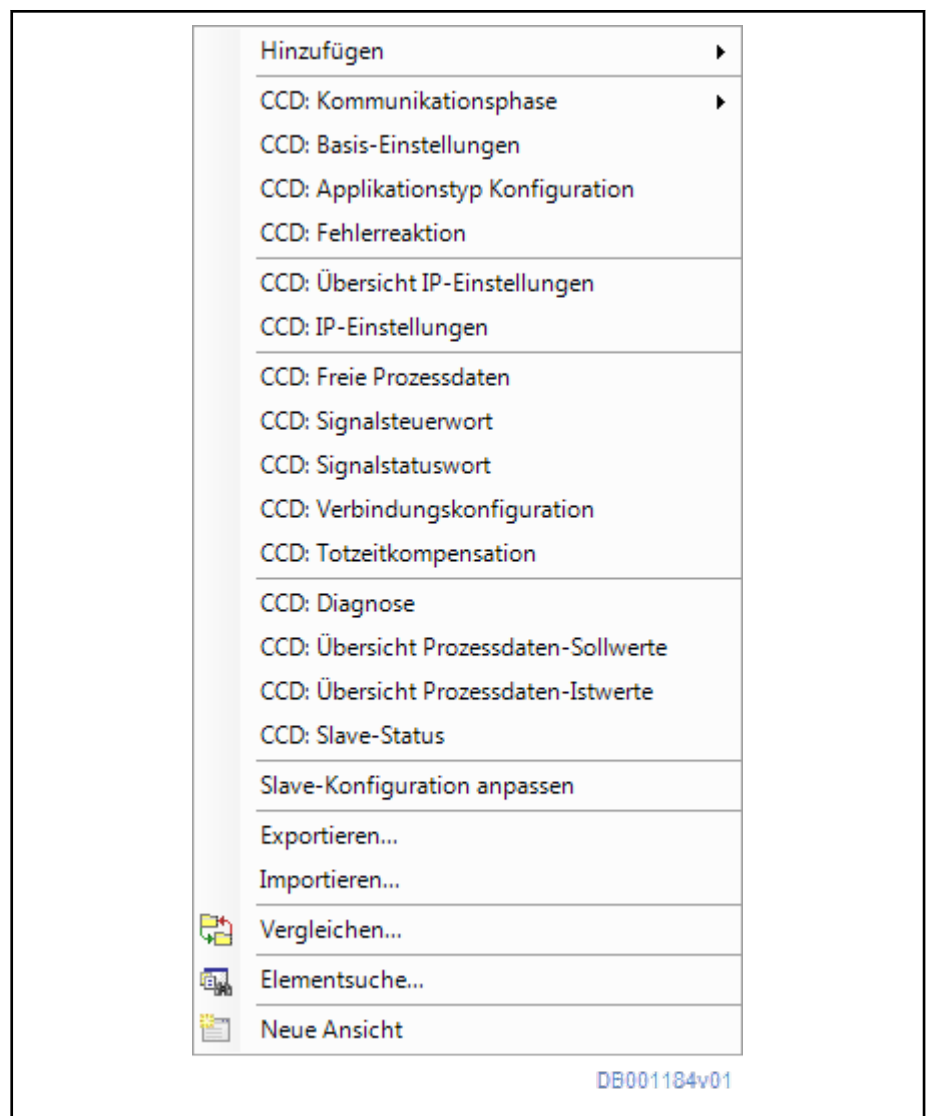


Abb. 7-9: Kontextmenü des Sercos Knotens

- Mit "CCD: **Basis-Einstellungen**" wird der gleichnamige Dialog aufgerufen:

Inbetriebnahme- und Anwendungshinweise

HMS01.1 [1] default

☒ Cross Communication Drive aktiv

☐ Kopplung P4->P2 an OM->PM des CCD-Masters

☐ Slave-IP-Adressen automatisch vergeben

☐ Standard Ethernet Nutzung am Ende der Sercos Linie

Kommunikationsphase: 4 0 2 4

Maximale Zykluszeit: 2000,000 µs

Kommandierungsmaster

☐ Externe SPS (CCD-Systemmodus)

☒ MLD-M im CCD-Master (MLD-M-Systemmodus)

☐ Expertenmodus (CCD-Basismodus)

Applikationstyp

Übernehmen

Projektierung sercos Slaves

Achs-Projektierung EA-Projektierung Topologie Topologieüberwachung

Achs-Nr.	sercos Adresse	Erforderlich
2	2	Ja
3		
4		
5		
6		
7		
8		
9		
10		

Rückgängig

Übernehmen

Gefundene Adressen:

2
9

Diagnose: Switching to phase 4 successfully carried out

[CCD-Fehlerreaktion](#) [Freie Prozessdaten](#) [CCD-Signalstatuswort](#) [CCD-Signalsteuerwort](#)

DB000721_V02.png

Abb. 7-10: Dialog "CCD: Basis-Einstellungen"

- "Cross Communication Drive aktiv": Hier kann die Querkommunikation aktiviert/deaktiviert werden.
- "Kommandierungsmaster": Durch die Anwahl "MLD-M im CCD-Master (MLD-M-Systemmodus)" kann die MLD als Mehrachssteuerung betrieben werden.
- "Kommunikationsphase": Zeigt die aktuelle Sercos Phase an.
- "Maximale Zykluszeit": Hier kann die Sercos Zykluszeit eingestellt werden.
- "Projektierung Sercos Slaves": Hier können die von der MLD-M kontrollierten Slaveachsen projektiert werden.
 - In der Registerkarte "Achsprjektierung" ist die maximale Anzahl der projektierten Slaveadressen in einer Tabelle gelistet; die gefundenen Antriebsadressen können konfiguriert werden. Es werden nur die konfigurierten Antriebsadressen in den CCD-Ring ein-

gefügt. [Die Geräteadressen der E/As (Inline-I/O, SPS-Antriebe,...) können im Projektbaum konfiguriert werden.] Es werden die Adressen der projektierten Antriebe angezeigt (P-0-1801.0.10). In der Spalte "Erforderlich" können mit Auswahl von "Nein" die konfigurierten Antriebsadressen temporär aus dem CCD-Ring ausgeschlossen werden.

"Gefundene Adressen": Hier werden alle über Sercos gefundenen Adressen der CCD-Slaves angezeigt (P-0-1803.0.3).

- In der Registerkarte "EA-Projektierung" werden alle Geräteadressen angezeigt, die als IO-Gerät angelegt und verwendet werden.
- In der Registerkarte "Topologie" wird die Topologie der konfigurierten Geräteadressen angezeigt.
- In der Registerkarte "Topologieüberwachung" kann die Topologieüberwachung aktiviert werden und der CCD-Ring wiederhergestellt werden, wenn ein Ringbruch erkannt wurde.



Auf alle hier aufgeführten Achsen kann über die MLD zugegriffen werden. Die in der SPS anzugebende Adresse beruht auf der Reihenfolge der projektierten Slaves. Die Adressierung geschieht dabei wie folgt:

- "Axis1" adressiert immer die lokale Achse (welche hier nicht angezeigt wird)
- "Axis2" adressiert den ersten projektierten Slave.
- "Axis3" adressiert den zweiten projektierten Slave.
- ...
- "Axis10" adressiert den neunten projektierten Slave.

Bedienfeld

Mit Hilfe des Bedienfelds besteht die Möglichkeit zu prüfen, ob eines der optionalen Erweiterungspakete "ML", "MA" oder "TF" (ab MPx-20) im Antrieb freigeschaltet ist und somit die SPS-Funktionalität genutzt werden kann.

Nach dem Zuschalten der Steuerspannung bootet der Antrieb. Während des Bootvorgangs erscheinen Meldungen auf dem Display. Bevor die Meldung "Boot 2.9" erscheint, müssen die beiden Tasten "Esc" und "Enter" gleichzeitig gedrückt und gehalten werden. Bei freigeschalteter SPS (Firmwarepaket "ML" bzw. "MA" lizenziert) erscheint dann die Meldung "PLC ?" im Display.

Mit den Pfeiltasten des Bedienfelds kann zwischen "Run PLC" und "Stop PLC" gewechselt werden. Falls ein lauffähiges SPS-Projekt im Antrieb gespeichert ist, kann es mit "Run PLC" gestartet werden. Die Option, die auf dem Display angezeigt wird, kann durch betätigen von "Enter" angewählt werden.

Wichtige Hinweise für bestimmte Funktionalitäten von IndraLogic 2G

IndraLogic 2G ist die in IndraWorks MLD integrierte SPS-Programmierschnittfläche für verschiedene SPS-Systeme.

Folgendes ist zu beachten:

- Systemereignisse werden nicht unterstützt.
- Der Parametermanager wird nicht unterstützt. Netzwerkvariablen werden ab der Version MPx-18VRS unterstützt.
- Die Funktionalität "Nicht verwendete I/O's aktualisieren" wird nicht unterstützt. Statt dessen gibt es in IndraWorks MLD eine Anzeige der I/O-Bereiche.

- Die Funktionalität "Forcen beibehalten" wird nicht unterstützt.
- Die Tasklaufzeiten werden in IndraLogic 2G nicht angezeigt. Stattdessen können mit der Funktion "MX_IECTaskGetLoad" aus der Bibliothek "MX_PLCOpen.library" die Laufzeiten im Programm ermittelt werden.
- Die Funktionalität "Persistente Variablen" ist nur bei Verwendung einer Speicherkarte möglich. Systembedingt ist diese Funktion jedoch auch bei allen Projekten in der SPS-Variablendeklaration anwählbar.

Sichere Programmierung

Verwendung von Pointern

Pointerzugriffe werden zur Laufzeit überwacht. Dabei wird verhindert, dass ein Programm auf Speicher außerhalb der SPS-Datenbereiche zugreift. Ein falscher Zugriffsversuch erzeugt eine SPS-Exception mit F6010.

GEFAHR

Tödliche Verletzung oder Sachschäden durch ungewollte Bewegungen oder Absturz hängender Achsen!

⇒ Bei der Verwendung von Pointern können fehlerhafte Zugriffe unvorhersehbare Folgen haben, daher ist hier äußerste Vorsicht geboten.

Stacküberprüfung

Der Stackbedarf wird beim Compilieren geprüft. Dabei wird verhindert, dass ein Programm mit zu großem Stackbedarf übersetzt werden kann.

GEFAHR

Tödliche Verletzung oder Sachschäden durch ungewollte Bewegungen oder Absturz hängender Achsen!

⇒ Bei der Verwendung von Funktionen mit größeren Datenmengen können unvorhersehbare Folgen entstehen. Bei Funktionsbausteinen ist der Stackbedarf minimal, da ihre Daten in den SPS-Daten abgelegt werden. Funktionen haben jedoch keine Instanz und legen alle Daten im Stack ab. Hier sollt man auf größere Datenstrukturen bzw. Felder verzichten oder Funktionsbausteine erstellen.

Arrayzugriffe (Bereichsüberschreitung)

GEFAHR

Tödliche Verletzung oder Sachschäden durch ungewollte Bewegungen oder Absturz hängender Achsen!

⇒ Wenn im Projekt keine Funktion mit dem Namen "CheckBounds" vorhanden ist, erfolgt keine Prüfung von indizierten Zugriffen auf Arrays (siehe auch Dokumentation "SPS-Programmentwicklung mit Rexroth IndraLogic 1.0"). Wenn sich der Index außerhalb des erlaubten Bereiches befindet, werden fremde Daten gelesen bzw. überschrieben. Dies birgt die Gefahr, dass fehlerhafte Zugriffe unvorhersehbare Folgen haben können.



Bei Projekten, die mit der Version MPx-18 neu erstellt werden, ist in der automatisch geladenen Bibliothek "CheckRtv" eine entsprechende Checkfunktion enthalten.

Unterbereichstypen (Bereichsüberschreitung)

⚠ GEFAHR

Tödliche Verletzung oder Sachschäden durch ungewollte Bewegungen oder Absturz hängender Achsen!

⇒ Wenn im Projekt keine Funktion mit dem Namen "CheckRangeSigned" bzw. "CheckRangeUnsigned" vorhanden ist, erfolgt keine Prüfung von Zugriffen auf Unterbereichstypen (siehe auch Dokumentation "SPS-Programmentwicklung mit Rexroth IndraLogic 1.0"). Wenn sich der Wert außerhalb des erlaubten Bereichs befindet, kann es im SPS-Programm evtl. zu unerwünschten Nebenwirkungen kommen. Dies birgt die Gefahr, dass bei fehlerhaften Zugriffen unvorhersehbare Folgen entstehen können.



Bei Projekten, die mit der Version MPx-18 neu erstellt werden ist in der automatisch geladenen Bibliothek "CheckRtv" eine entsprechende Checkfunktion enthalten.

Nulldivision

⚠ GEFAHR

Tödliche Verletzung oder Sachschäden durch ungewollte Bewegungen oder Absturz hängender Achsen!

⇒ Wenn im Projekt keine Funktionen enthalten sind, mit denen eine Prüfung auf Division durch Null durchgeführt wird (siehe auch Dokumentation "IndraLogic 2G Programmieranleitung"), kann es im SPS-Programm evtl. zu unerwünschten Nebenwirkungen kommen. Wenn der Divisor 0 ist, birgt dies die Gefahr, dass bei fehlerhaften Zugriffen unvorhersehbare Folgen entstehen können.



Bei Projekten, die mit der Version MPx18 neu erstellt werden ist in der automatisch geladenen Bibliothek "CheckRtv" eine entsprechende Checkfunktion enthalten.

Zugriffe auf Variablen über Programmiersystem oder eine Schnittstelle sind nur bytekonsistent

Wenn eine 2- oder 4-Byte-Variable gelesen wird, kann es sein, dass das Lesen durch eine Änderung der Variable in einer SPS-Task unterbrochen wird. So könnte bei gleichzeitiger Änderung mehrerer enthaltener Bytes kurzzeitig ein falscher Wert übertragen werden (z.B. ein Byte alt, ein Byte neu).

7.2.3 Nutzung des "AxisInterface" für die Programmierung von Rexroth IndraMotion MLD

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 Funktion "AxisInterface" kann durch einen Doppelklick auf den Logikknoten aktiviert werden.



Die Bibliothek "MX_TechInterface" wird in der Online-Hilfe von IndraWorks und in der gedruckten Dokumentation "Rexroth IndraMotion MLC 14VRS, Technologie-Bibliotheken" beschrieben.

Weitere Besonderheiten, die bei der Anwendung des Achs-Interface im System IndraMotion MLD zu beachten sind, sind in der Bibliotheksbeschreibung zu IndraMotion MLD beschrieben.

7.2.4 Automatisierte Programmerstellung mit "GAT compact"

"GAT compact" (**G**eneric **A**pplication **T**emplate) ist ein Beispielprojekt, basierend auf IEC 61131-3. Es wird mit dem IndraWorks Engineering Framework "IndraMotion MLD" ab der Version 14V14 zur Verfügung gestellt und lässt sich Wizard-unterstützt ab der Firmware-Version MPx-20 in den Antrieb laden und in Betrieb nehmen. "GAT compact" enthält folgende Teilfunktionen:

- Schrittkette mit Schritten für Initialisierung, Manuell- und Automatikbetrieb und Fehlerzustand
- Integriertes Fehlerhandling
- Integriertes Achshandling
- Inbetriebnahme-Visualisierungen
- Vorbereitete HMIs

"GAT compact"-Beispielprojekt

Einleitung

Mit der nachfolgenden Beschreibung erstellen Sie in wenigen Minuten eine Motion-Logic-Applikation mit einer Achse auf Basis des "GAT compact"-Beispielprojektes.

Das Beispielprojekt ist einschaltfertig, Sie können es frei erweitern und modifizieren. Das GAT compact verwendet das Achs-Interface und enthält eine Maschinenbetriebsarten-Schrittkette, sowie eine Fehlerbehandlung. Weiterhin enthält es vorbereitete Inbetriebnahme-Visualisierungen und HMIs.

Mit dem Beispielprojekt lässt sich die lokale Achse im Automatikbetrieb in Geschwindigkeitsregelung mit einer einstellbaren Geschwindigkeit bewegen. Im Manuellbetrieb kann die Achse beispielsweise über eine Visualisierung bewegt werden.

Voraussetzungen

"GAT compact" ist in dem IndraWorks Engineering Framework "IndraMotion MLD" ab der Version 14V14 verfügbar.

Es wird ein Antriebsregelgerät benötigt, auf dem sich mindestens die Firmware-Version MPx-20 befindet. Außerdem muss das Funktionspaket "IndraMotion MLD" freigeschaltet sein.

Damit das "GAT compact"-Beispielprojekt funktioniert, muss die SPS wie folgt konfiguriert sein:

- SPS hat permanente Kontrolle über den Antrieb (P-0-1367, Bit 4)
- Unterstützung der AxisData-Struktur (P-0-1367, Bit 6)

In IndraWorks kann der Dialog zur Konfiguration der SPS durch Doppelklick auf "Konfiguration" aufgerufen werden, um die notwendigen Einstellungen vorzunehmen:

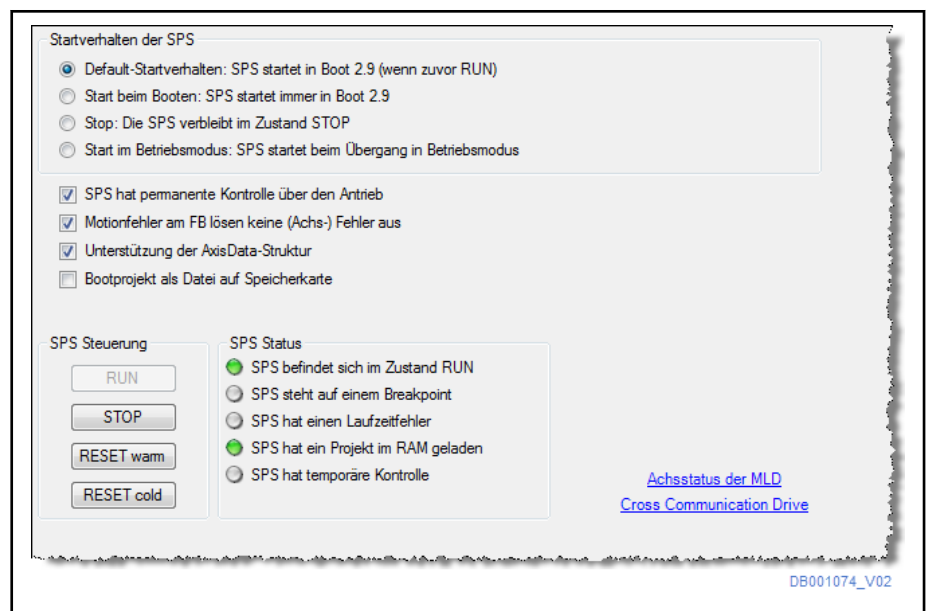


Abb. 7-11: Dialog zur Konfiguration

Erste Schritte mit dem "GAT compact"-Beispielprojekt

"GAT compact" einfügen

1. Fügen Sie ein GAT-geeignetes Gerät (z. B. Antriebsregelgerät "IndraDrive") mit MLD-Projekt in ein IndraWorks-Projekt ein.

Der Dialog "Logic anlegen" erscheint.

2. Wählen Sie im Dialog "Logic anlegen" "GAT compact Beispielprojekt" und bestätigen Sie Ihre Auswahl mit **OK**.



Ist noch keine Logic angelegt, dann können Sie den Dialog "Logic anlegen" aufrufen, indem Sie im Projektexplorer auf den Logic-Knoten einen Doppelklick ausführen.

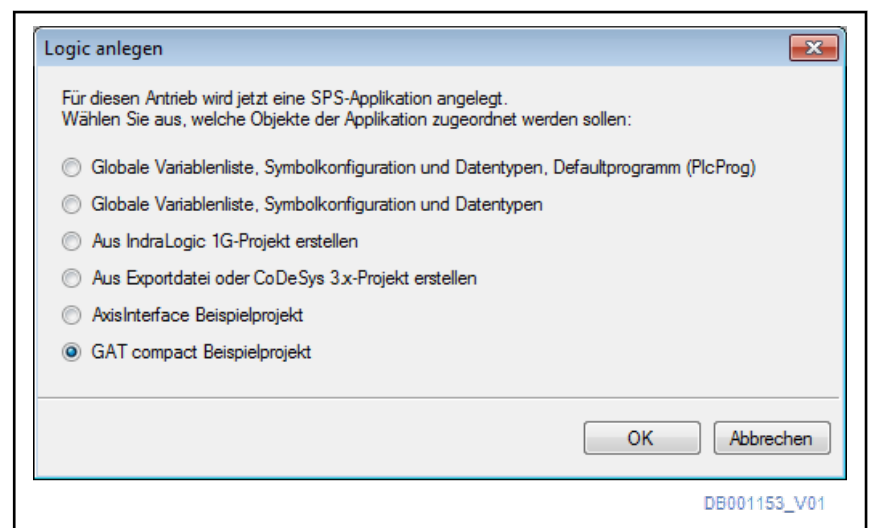


Abb. 7-12: Logic anlegen

Das "GAT compact Beispielprojekt" wird automatisch in die SPS-Applikation eingefügt.

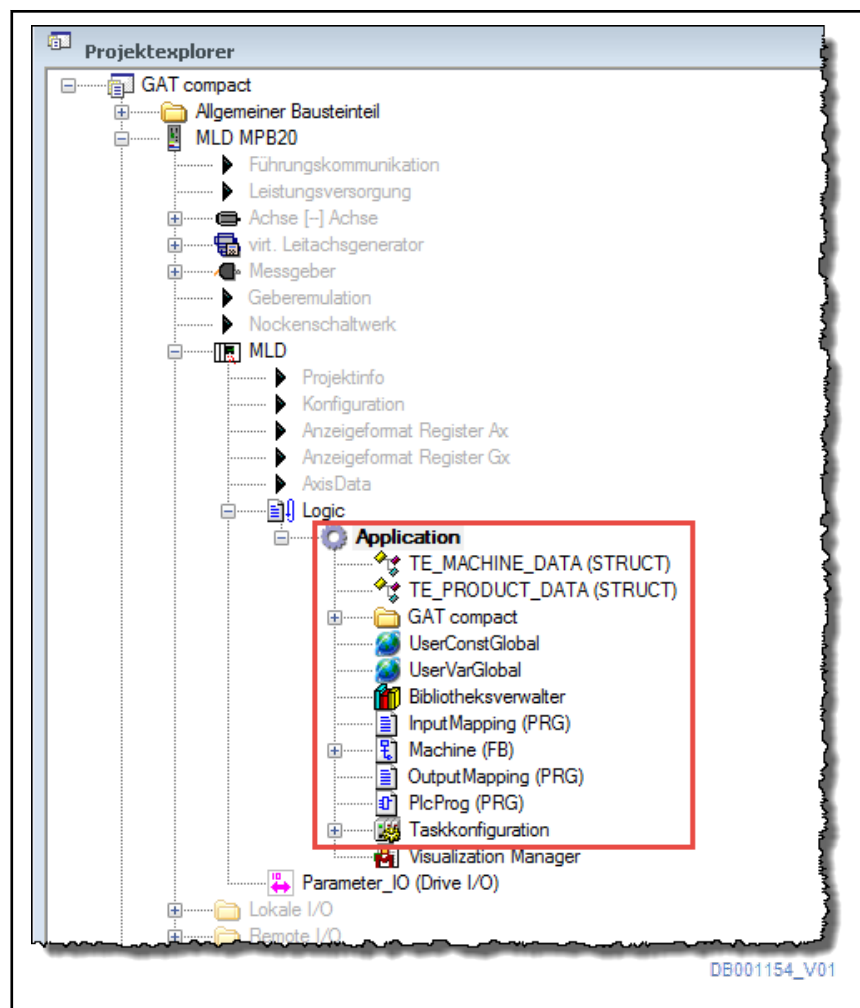


Abb. 7-13: "GAT compact"-Beispielprojekt

Projekt laden und starten

Laden und starten Sie das "GAT compact"-Beispielprojekt mit folgenden Schritten:

1. Einloggen
2. SPS-Applikation auf die Steuerung laden
3. SPS-Applikation starten

Bedienung über Visualisierung

Über die im "GAT compact"-Beispielprojekt vorhandene Inbetriebnahmevisualisierung "System_Overview_All_Axis" kann das "GAT compact"-Beispielprojekt einfach bedient werden.

1. Öffnen Sie die Visualisierung, indem Sie in dem Fenster "GAT compact" einen Doppelklick auf "System_Overview_All_Axis" ausführen.

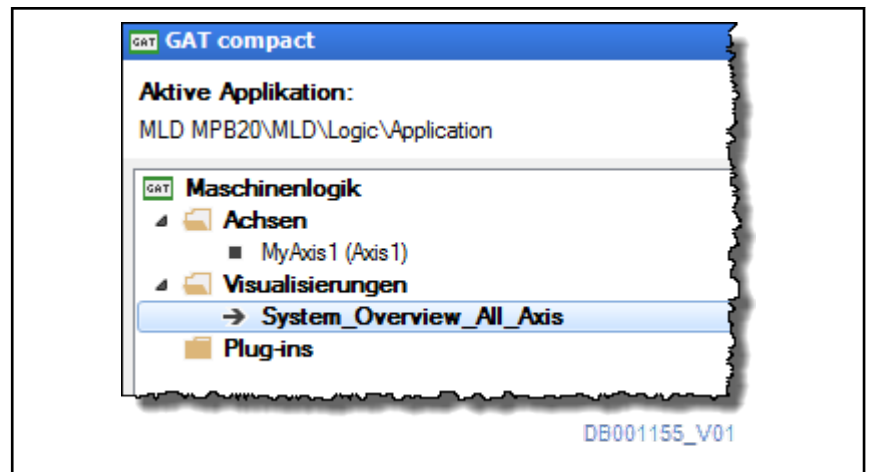


Abb. 7-14: Visualisierung öffnen



Das Fenster "GAT compact" kann über **Ansicht ► Andere Fenster ► GAT compact** geöffnet werden.

Die Visualisierung "System_Overview_All_Axis" wird geöffnet.

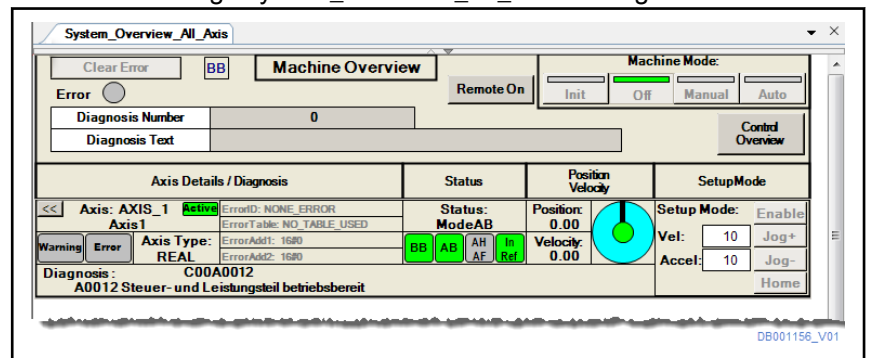


Abb. 7-15: Visualisierung "System_Overview_All_Axis"

- Wählen Sie in der Visualisierung **Remote On** aus, damit die Visualisierung die Kontrolle über das "GAT compact"-Beispielprojekt übernimmt.

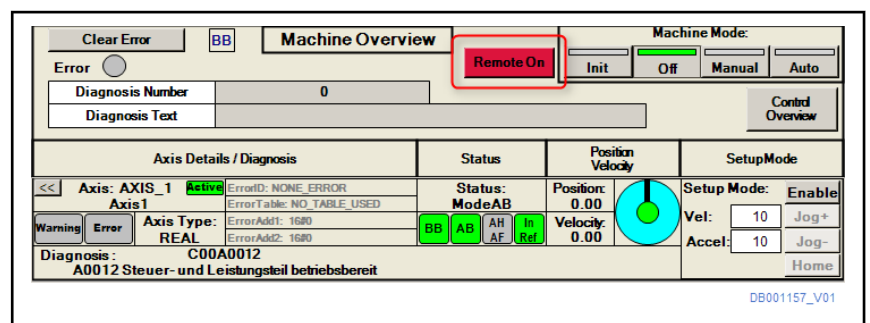


Abb. 7-16: Die Kontrolle über das "GAT compact"-Beispielprojekt mit der Visualisierung übernehmen

- Wählen Sie **Manual** und anschließend **Enable** aus.
Über **Jog+** und **Jog-** können Sie die Achse bewegen.

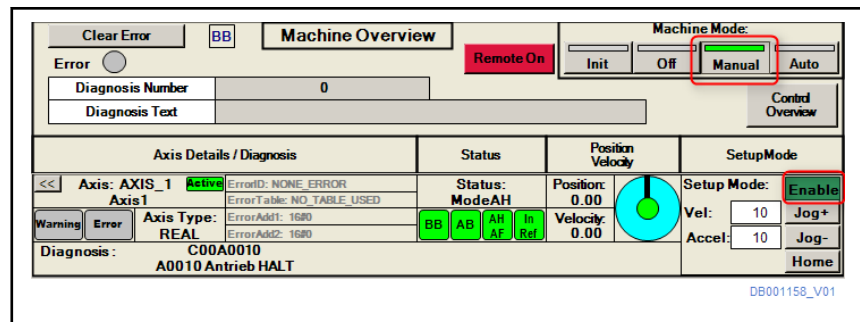


Abb. 7-17: Bewegen der Achse im Manuellbetrieb

4. Aktivieren Sie über **Auto** den Automatikbetrieb.

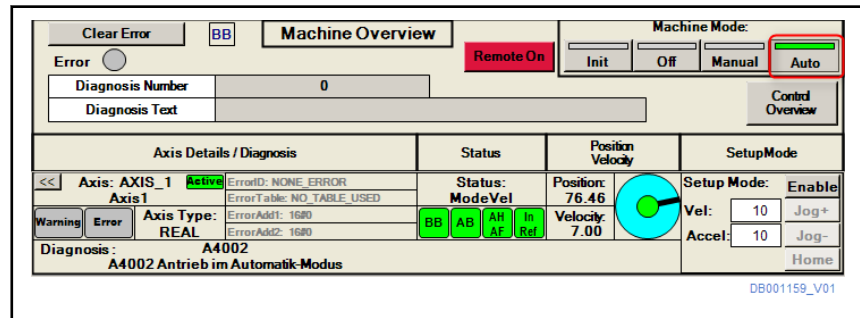


Abb. 7-18: Bewegen der Achse im Automatikbetrieb

Die Achse bewegt sich kontinuierlich mit 7 U/min, da in der Betriebsart "Auto" eine Bewegung mit kontinuierlicher Geschwindigkeit programmiert ist.

Betriebsarten-Steuerung durch den Funktionsbaustein "Machine"

Der Funktionsbaustein "Machine" ist das Hauptelement des "GAT compact"-Beispielprojekts. Der Funktionsbaustein enthält eine Schrittkette und die Implementierung der Betriebsarten.

Durch einen Doppelklick auf den Funktionsbaustein "Machine" öffnen Sie die Betriebsarten-Schrittkette.

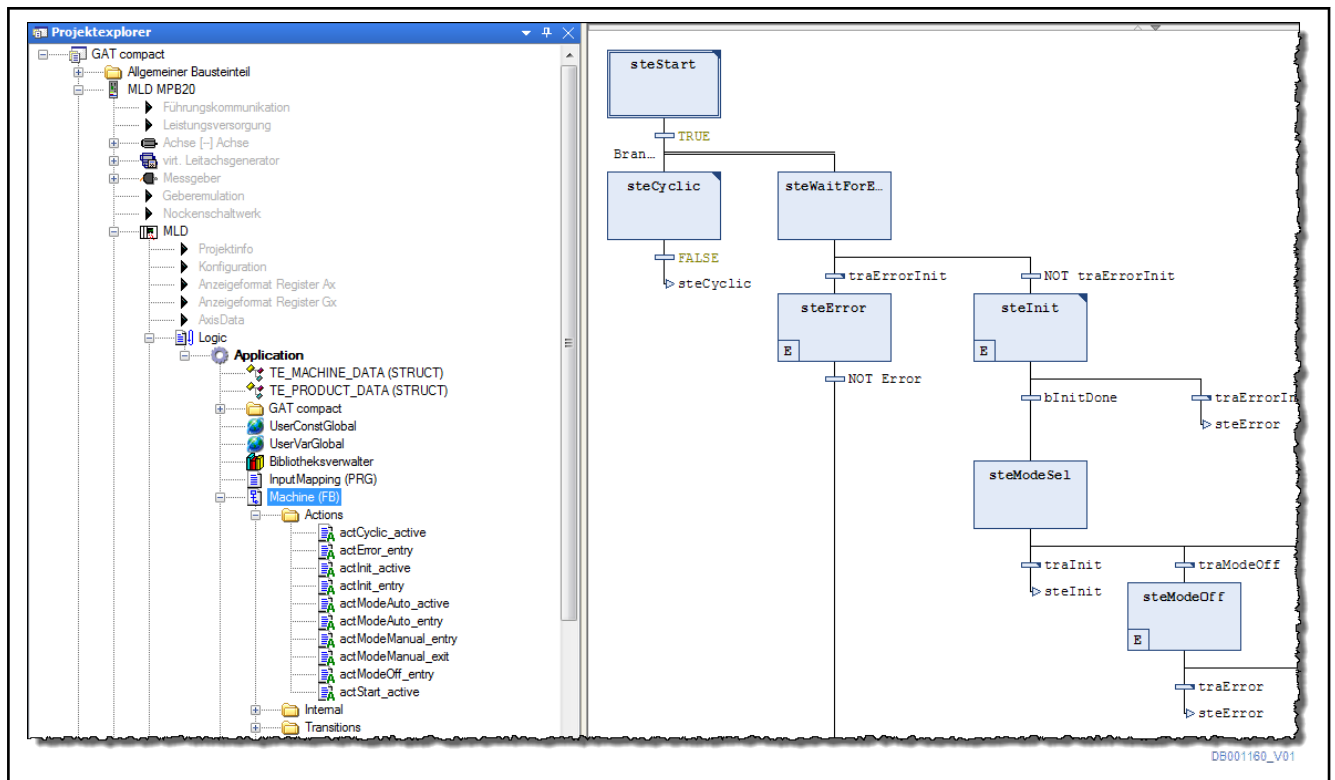


Abb. 7-19: Funktionsbaustein "Machine"

Die Eingangsaktion "actModeAuto_entry" des Funktionsbausteins "Machine" wird einmalig nach Auswahl der Betriebsart "Auto" bearbeitet. Diese Aktion startet die Achsbewegung. Die Achse wird in den Geschwindigkeitsmode (ModeVel) geschaltet.

Die Aktion "actModeAuto_active" wird permanent zyklisch bearbeitet, wenn die Betriebsart "Auto" aktiv ist. Hierbei werden die Sollgeschwindigkeit der Achse und die angezeigte Geschwindigkeit zyklisch kopiert. Dadurch werden Geschwindigkeitsänderungen in der Betriebsart "Auto" sofort wirksam.

Die Eingangsaktion "actModeManual_entry" stoppt alle Achsen durch die FOR-Schleife über alle Achsen und Vorgabe der Betriebsart "ModeAb".

Die Aktion "actCyclic_active" wird, unabhängig von der Maschinenbetriebsart, permanent zyklisch aufgerufen. Die Aktion enthält den Aufruf des Achs-Interface und das Fehlerhandling.

Die Eingangsaktion "actError_entry" wird einmalig nach Fehlerauftritt bearbeitet. Sie enthält die Fehlerreaktion. Im Beispiel werden alle Achsen des Achscontainers in den Stillstand verzögert. Weitere Stillsetzungsbeispiele sind auskommentiert enthalten.

7.3 SPS-Projektaufbau

7.3.1 Allgemeines

Das SPS-Projekt besteht aus mehreren Dateien und ist Teil eines IndraWorks-Projekts:

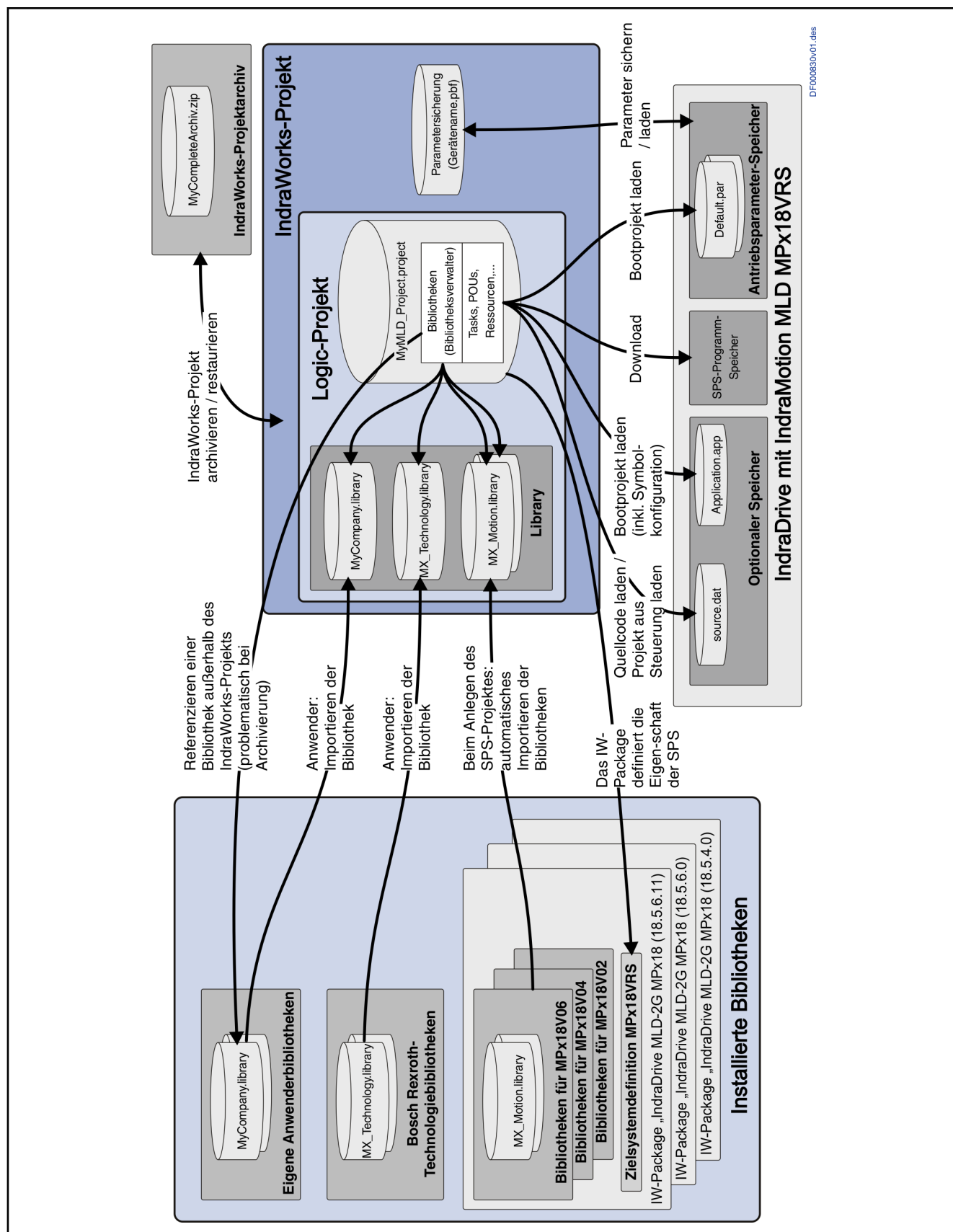


Abb. 7-20: Die wichtigsten SPS-Projektdateien im Kontext des IndraWorks-Projekts

Auf der linken Seite der obigen Abbildung sind die auf dem PC installierten IW-Packages und Bibliotheken dargestellt. Es sind auch optionale Bibliotheken dargestellt, die durch die Installation von Technologiepaketen oder durch die Installation des Anwenders erzeugt wurden.

In dem unteren Bereich der Abbildung ist der Antrieb mit den relevanten Datenablagen dargestellt. Wenn die externe Speicherkarte vorhanden ist, kann der SPS-Quellcode geladen werden bzw. es können Symbolinformationen für HMIs geladen werden. In diesem Fall liegt auch der Parameterspeicher auf der Speicherkarte (in der Grafik nicht sichtbar).

In dem oberen Bereich der Abbildung ist ein IndraWorks-Projektarchiv dargestellt. Das gesamte Projekt wird beim Archivieren in eine einzige gepackte Datei überführt. Das IndraWorks-Projekt selbst besteht aus Strukturinformationen, einer optionalen Parametersicherung und einem optionalen SPS-Projekt. Das SPS-Projekt liegt in einem Unterverzeichnis und enthält alle von IndraLogic erzeugten Dateien. Die wichtigste davon ist die Datei mit der Endung *.project, welche den SPS-Quellcode und Projekteinstellungen enthält. Im SPS-Verzeichnis gibt es ein Unterverzeichnis "LIB". In dieses Unterverzeichnis kopiert das System beim Anlegen eines neuen Projekts alle Bibliotheken des passenden Release des aktuellen Zielsystems.

Die vom Benutzer ausführbaren Aktionen sind mit Pfeilen gekennzeichnet und werden in den folgenden Kapiteln näher erläutert.

7.3.2 Projektablage im Antrieb

Beim Erstellen und Laden eines SPS-Projekts werden auf dem Zielsystem (Antrieb) unterschiedliche Dateien erzeugt.

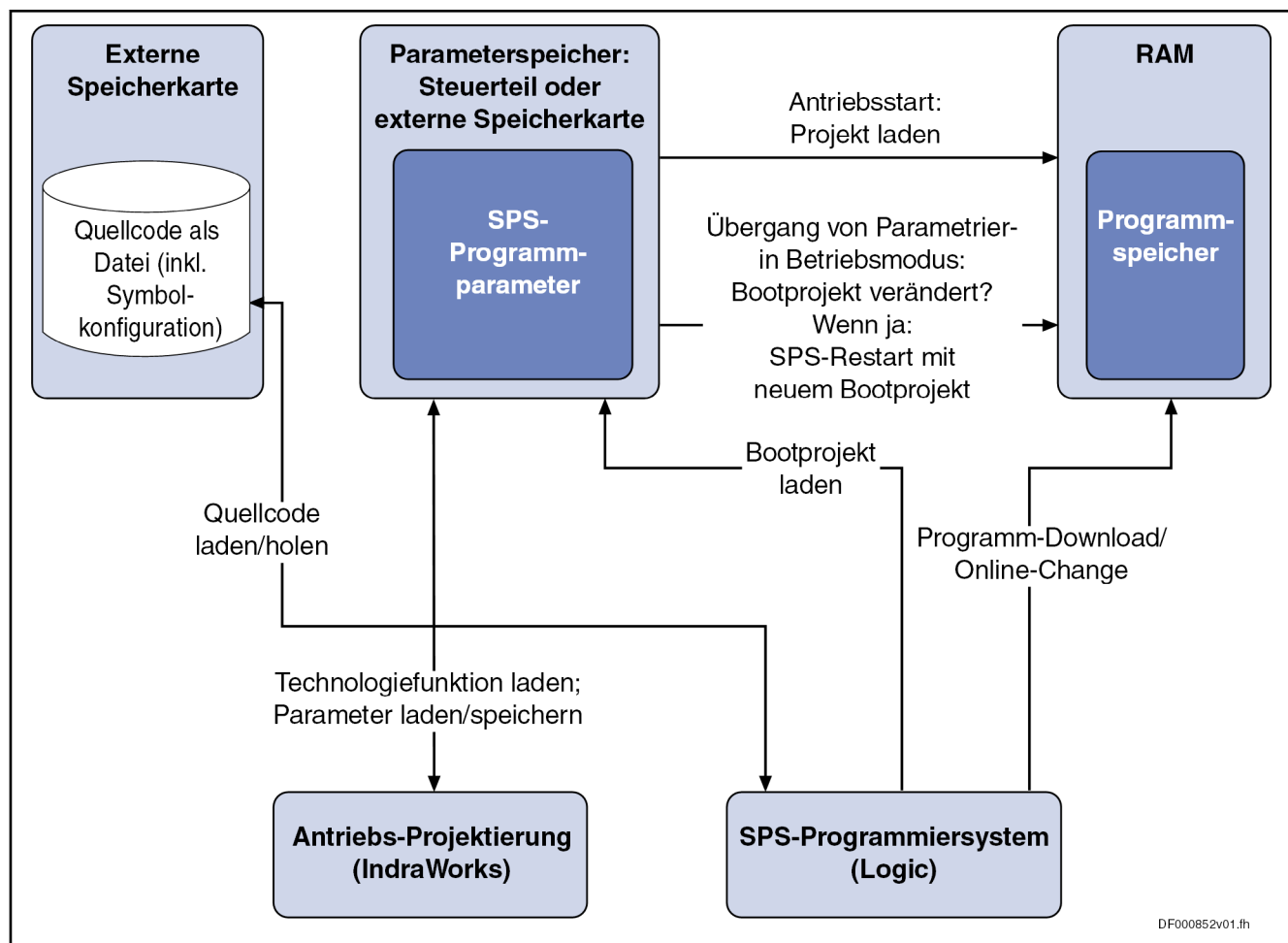


Abb. 7-21: Projektablage im Antrieb

Der "Programm-Download" lädt das SPS-Programm lediglich in den RAM-Speicher des Antriebs. Wird das Bootprojekt mit dem Menüpunkt **Bootprojekt laden** in den Antrieb übertragen, so überträgt IndraLogic zunächst einige Dateien, die im Standardfall kodiert und komprimiert als Satz von einigen Parametern abgelegt werden.

- **Betrieb ohne externe Speicherkarte**

Wenn keine externe Speicherkarte vorhanden ist, wird der Quellcode in Parametern abgelegt.

- **Externe Speicherkarte als aktiver Speicher**

Ist eine externe Speicherkarte vorhanden, so kann zusätzlich das SPS-Projekt als Quellcode und die Symbolkonfiguration auf dem Antrieb abgelegt werden. In diesem Fall liegt das Bootprojekt und die Symbolkonfiguration in der Parameterdatei auf der externen Speicherkarte.

7.3.3 Bootprojekt

Mit IndraWorks MLD kann das Bootprojekt in Form von Parametern (P-0-1352 .. P-0-1358) geladen und gespeichert werden (eine Parametersicherung enthält somit das Bootprojekt, also den Binärcode, aber nicht den Quellcode).

Das Bootprojekt der SPS ist durch folgende Eigenschaften gekennzeichnet:

- Es besteht aus mehreren Systemdateien. Diese enthalten das Binärprojekt ohne symbolische Informationen. Die Systemdateien werden beim **Bootprojekt erzeugen** (nur online) automatisch in den Antrieb geladen.
- Im Antrieb werden die Systemdateien ...
 - ... in die Listenparameter P-0-1352 bis P-0-1358 aufgeteilt und abgelegt.
 - ... auf der externen Speicherkarte als Datei abgelegt, Voraussetzungen dafür ist der Einsatz einer Speicherkarte und entsprechende "MLD-Konfiguration" in IndraWorks bzw. die Parametrierung von P-0-1367.
- Das Bootprojekt hat zunächst keinen Einfluss auf das aktuell im RAM laufende Programm. Erst nach dem Wiedereinschalten des Antriebs wird das Bootprojekt in den RAM geladen und ggf. gestartet.
- Alternativ wird ein geändertes Bootprojekt beim Schalten in den Betriebsmodus (P3 nach P4) geladen.

7.3.4 Dateisystem im Antrieb

Die nachfolgende Tabelle zeigt alle im Antrieb vorhandenen Dateien. Viele der Dateien sind für den Anwender nicht direkt sichtbar bzw. werden automatisch verwaltet. Alle Dateien des Typs "Systemdatei" werden von der Antriebsfirmware direkt verwaltet. Die SPS-Systemdateien werden durch IndraLogic verwaltet.

Es gibt also Dateien, die intern genutzt werden und nur im Speicher existieren. Weiter gibt es Dateien, die unabhängig von der externen Speicherkarte in Form von Parametern vorliegen. Alle weiteren Dateien befinden sich auf der Speicherkarte.

In dem Verzeichnis "Dokumentation" kann der Anwender eigene Dateien zur Dokumentation seiner Anlage/Maschine hinterlegen.

Medium	Verzeichnis	Datei	Typ	Inhalt	Zugriff
Parameter P-0-1352	-	Application.sts	SPS-Systemdatei	SPS-Status	Logic-Kontextmenü: Start/Stop
	-	Application.crc	SPS-Systemdatei (P-0-1367.0.8=0)	Bootapplikation Checksumme	Logic-Kontextmenü: Bootprojekt erzeugen
Parameter P-0-1353 bis P-0-1358 (nur IndraDrive)	-	Application.app	SPS-Systemdatei (P-0-1367.0.8=0)	Bootapplikation	
Internes Flash (nur HydraulicDrive)	-	Application.app	SPS-Systemdatei (P-0-1367.0.8=0)	Bootapplikation	
Externe Speicher- karte	Plc	Application.crc	SPS-Systemdatei (P-0-1367.0.8=1)	Bootapplikation Checksumme	
		Application.app	SPS-Systemdatei (P-0-1367.0.8=1)	Bootapplikation	
		Archive.prj	SPS-Systemdatei	Quellcode-Archiv	Logic-Kontextmenü: Quellcode laden
	Documentation	*.*	Anwenderdateien	Anlagen-Dokumentation	Extern
	Tools	*.*	Anwenderdateien	Tools	Extern
	User	*.*	Anwenderdateien	Anwenderdateien	SPS-Programm mit "Sys-File", FTP oder IndraWorks-Explorer
	Backup	*.*	SPS-Systemdatei	Retaindaten- und Parameter-Backup	Logic: Kommando "Sicherung"

Tab. 7-1: Dateisystem im Antrieb

7.3.5 Externe Speicherkarte als Speichermedium

Bei Verwendung der MLD kommt der externen Speicherkarte eine besondere Bedeutung zu. Hierbei ist die Speicherkarte permanent gesteckt und als aktiver Parameterspeicher wirksam. (Die antriebsintegrierte SPS kann auch ohne Speicherkarte genutzt werden.)

Ablage des Quellcodes einer SPS-Applikation

Die Funktion "Quellcode laden" dient zur kompletten Ablage des SPS-Programms auf dem Antrieb / der externen Speicherkarte; d. h. es wird nicht nur der Binärcode, sondern auch der gesamte Quellcode abgelegt. So ist es möglich das SPS-Projekt zu einem späteren Zeitpunkt mit Hilfe der externen Speicherkarte wieder herzustellen.



Es ist sinnvoll das SPS-Projekt zusätzlich auf dem PC zu archivieren. Hierzu gibt es eine menügeführte Archivierung im Programmiersystem. Wir empfehlen das gesamte IndraWorks-Projekt mit IndraWorks MLD zu archivieren.

Wiederherstellen eines SPS-Projektes aus dem geladenem Quellcode

Ein auf der externen Speicherkarte abgelegter Quellcode kann per Menü **Datei ► Öffnen...** aus dem Antrieb geladen werden. Dabei ist zu beachten, dass beim Laden eines Projekts aus der Steuerung dieses nicht mit der gleichen Dateiablage restauriert wird. Es gibt ein so genanntes Upload-Verzeichnis, in welches alle zusätzlichen Dateien wie z. B. Bibliotheken abgelegt werden. Für die Einbindung der Bibliotheken wird dazu der Bibliothekspfad so eingestellt, dass er auf das Uploadverzeichnis zeigt. Soll ein so restauriertes Projekt wieder auf dem PC gespeichert werden, so müssen die Dateien aus dem Uploadverzeichnis in das Bibliotheksverzeichnis des Projekts kopiert werden. Für weitere Informationen siehe IndraLogic 2G-Hilfe.

Symbolkonfiguration für HMI (z.B. VCP)

Wenn eine externe Speicherkarte vorhanden ist, wird die Symbolkonfiguration mit dem Programmcode auf die externe Speicherkarte (µSD-Card) gela-

den. Mit Hilfe der Symbolkonfiguration kann über die Kommunikationsschnittstelle auf SPS-Variablen symbolisch zugegriffen werden. Dies ist für die Darstellung von SPS-Projektvariablen auf dem Komfort-Bedienfeld bzw. einem externen Bediengerät notwendig.

Über die Symbolkonfiguration kann eingestellt werden, ob und welche Symbole gespeichert werden sollen.

Generell sollten nur die benötigten Symbole ausgewählt werden, um die Ladezeiten (Download) zu verringern und eine hohe Verarbeitungsgeschwindigkeit der Symbolzugriffe durch ein HMI oder Komfortdisplay zu erreichen.

Die Symbole werden nach dem Übertragen im Antrieb geladen. Dies benötigt dynamischen Speicher, der jedoch begrenzt ist. Falls der Speicher nicht für die Symbole ausreicht, müssen weniger Symbole ausgewählt werden.

Der tatsächlich benötigte Speicher hängt von der Anzahl der Symbole, deren Namen und den Datentypen ab.



Der verfügbare Speicher ist abhängig von der Ablage des Bootprojekts.

Ablage von eigenen Daten auf der externen Speicherkarte mittels SPS

Es gibt die Möglichkeit aus dem SPS-Programm auf Anwenderdateien zuzugreifen. Dazu werden die Bibliotheken "SysFile"/"SysFileAsync" und "SysDir"/"SysDirAsync" bereitgestellt.

Ablage von eigenen Daten auf der externen Speicherkarte mittels FTP-Server

Es gibt die Möglichkeit mittels FTP-Server auf die externe Speicherkarte zuzugreifen.

7.3.6 IndraWorks-Package

Die unterschiedlichen Firmware-Versionen und Firmware-Releases werden von den zugehörigen IndraWorks-Packages unterstützt. Jede Firmware-Version bringt ein eigenes IndraWorks-Package mit. Die IndraWorks-Packages werden bei der Installation von IndraWorks MLD installiert oder können über den Dialog "IndraLogic Geräteversion ändern" (Projektexplorer: Kontextmenü **MLD ▶ IndraLogic Geräteversion ändern**) durch Anklicken des Links "IW-Package hinzufügen" nachinstalliert werden.

7.3.7 Bibliotheken

Über Bibliotheken werden Funktionalitäten und Ressourcen (Variablen, Datentypen,...) bereitgestellt. Dabei gibt es unterschiedliche Kategorien von Bibliotheken:

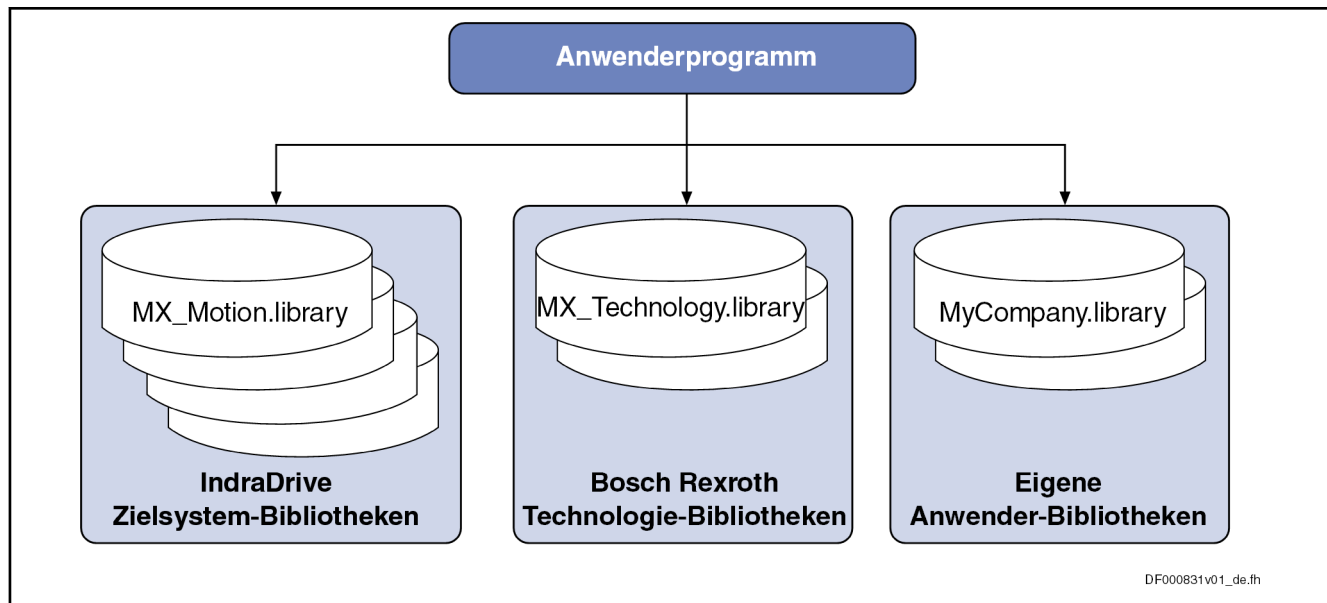


Abb. 7-22: Bibliotheks-Kategorien

IndraWorks bringt eine Sammlung an Bibliotheken mit.

Es sind außerdem Bibliotheken mit Technologiebausteinen erhältlich. Diese setzen auf die installierten Bibliotheken auf und stellen eine erweiterte Funktionalität zur Verfügung.

Zusätzlich hat der Anwender die Möglichkeit, weitere z.B. firmenspezifische Bibliotheken einzubinden.



Zielsystemfremde Bibliotheken (Bibliotheken anderer Zielsysteme oder Bibliotheken aus fremden Quellen) dürfen in der MLD nicht verwendet werden.

7.4 Arbeitsschritte während der Programmierung und Inbetriebnahme

7.4.1 Neues Projekt anlegen

Da ein SPS-Projekt stets eine im IndraWorks-Projekt eingebettete Komponente ist, muss im ersten Schritt ein neues Projekt in IndraWorks MLD angelegt werden, das zunächst leer ist. Es existieren zwei Möglichkeiten, um einen Antrieb in das Projekt einzubinden:

- **Um offline einen Antrieb in das Projekt einzubinden**, muss in der Gerätebibliothek mit der linken Maustaste ein Antrieb (IndraDrive) ausgewählt werden (Maustaste gedrückt halten) und auf den obersten Knoten im Projektexplorer gezogen werden. Beim Loslassen der Maustaste startet der Konfigurationsassistent für den Antrieb.
- **Wenn der Antrieb bereits zur Verfügung steht**, so kann er direkt über eine serielle Verbindung oder eine Ethernetverbindung angeschlossen werden. Mit dem Befehl "Antriebe suchen" wird der Antrieb dann automatisch identifiziert und kann in das Projekt übernommen werden. Die Grundkonfiguration entfällt.

Falls die SPS-Funktionalität in den Funktionspaketen noch nicht aktiviert wurde, ist dies nun durchzuführen (siehe Firmware-Funktionsbeschreibung "Funktionspaketfreischaltung").

Durch einen Doppelklick auf den Logic-Knoten wird nun ein leeres SPS-Projekt erstellt. Anschließend kann die SPS-Projektierung durch einen weiteren Doppelklick oder über das Kontextmenü gestartet werden.

7.4.2 SPS-Projekt importieren

Zielsetzung Um bereits vorhandene SPS-Projekte zu verwenden, die mit einer früheren IndraLogic-Version ohne IndraWorks MLD erstellt wurden, gehen Sie wie folgt vor:

- Erforderliche Arbeitsschritte**
1. Ein IndraWorks-Projekt anlegen.
 2. Steuerung im Projekttexplorer durch Doppelklick auf den Logic-Knoten initialisieren.
 3. Im Kontextmenü von **Application** ► **Datenübernahme...** auswählen.

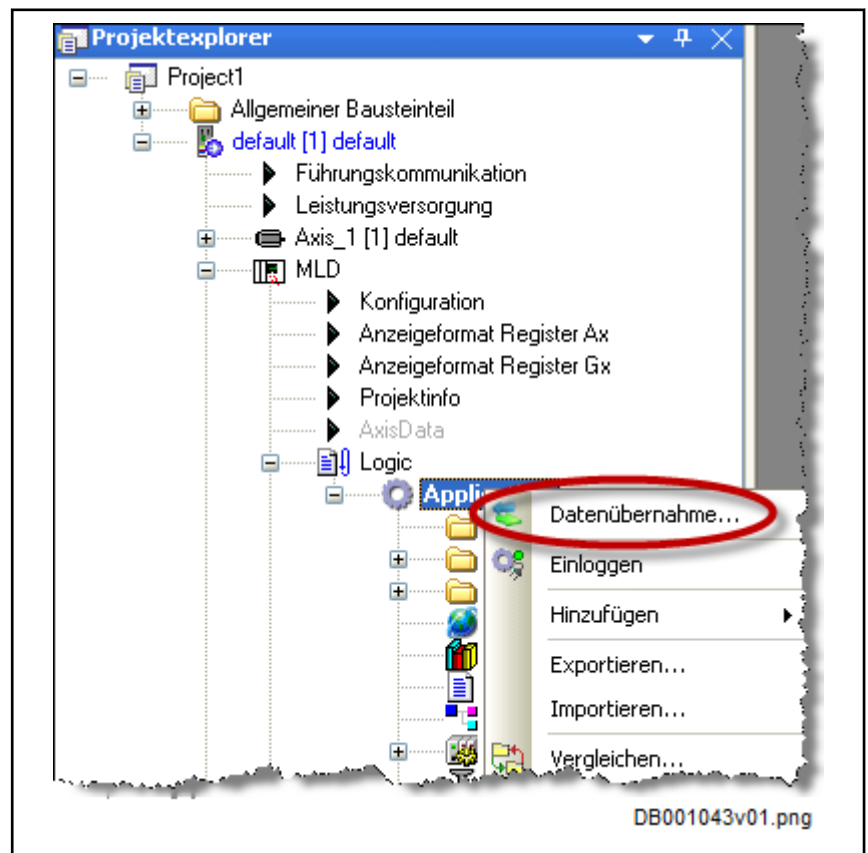


Abb. 7-23: Projektbaum mit Logic-Knoten

4. Im Dialogfenster "Projekt selektieren" die gewünschte Projektdatei auswählen.
5. Durch Mausklick auf die Schaltfläche "Öffnen" wird der Importvorgang gestartet.



Beachten Sie folgendes bei dem Import von SPS-Projekten:

- Die im Quellprojekt enthaltene Tracekonfiguration und Traceaufzeichnung werden nicht importiert. Workaround: Die Traceaufzeichnung in eine *.trc-Datei und die Tracekonfiguration in eine *.tcf-Datei speichern. Beide Dateien können nach dem Import von IndraLogic geladen werden.
- Netzwerkvariablen können nicht importiert werden (Fehlermeldung).
Workaround: Die globalen Netzwerkvariablen im Quellprojekt über die IndraLogic-Funktion **Projekt exportieren...** in eine *.iwx-Datei schreiben. Im Zielprojekt diese *.iwx-Datei über die IndraLogic-Funktion **Projekt importieren...** in das Projekt importieren.
- Die Zielsystemeinstellungen (Registerkarte Ressourcen) und die Einstellungen unter **Projekt ▶ Optionen** werden nicht importiert, sondern auf Defaultwerte gesetzt. Diese Einstellungen müssen überprüft und gegebenenfalls angepasst werden.
- Beim Import eigener in CFC geschriebener Funktionsbausteine gehen die graphischen Verbindungen und die Funktionsbaustein-Bezeichnungen verloren.

7.4.3 IndraLogic-Projekt (*.project-Datei) aus IndraWorks MLD exportieren

Zielsetzung Um ein IndraLogic-Projekt außerhalb der IndraWorks-Umgebung verwenden zu können, gehen Sie wie folgt vor:

- Erforderliche Arbeitsschritte**
1. Klicken Sie im Projektexplorer mit rechter Maustaste auf den Logic-Knoten.
Das Kontextmenü wird geöffnet.
 2. Wählen Sie **Speichern unter....** Dieser Menüpunkt im Kontextmenü wird nur angeboten, wenn das IndraLogic-Projekt bereits im IndraWorks-Projekt gespeichert ist.
 3. Vergeben Sie einen Dateinamen für das IndraLogic-Projekt.



Die Geräteversion wird nicht mit dem IndraLogic-Projekt zusammen exportiert! Stellen Sie sicher, dass die richtige Geräteversion zur Verfügung steht, wenn das exportierte IndraLogic-Projekt verwendet werden soll.

7.4.4 IndraLogic-Projekt in den Antrieb laden

Während der Programmerstellung gibt es unterschiedliche Ausprägungen, wie ein IndraLogic-Projekt in den Antrieb geladen werden kann:

- "Download"
- "Online-Change"
- "Bootprojekt laden"

Download Bei einem "Download" wird nur das aktuell laufende Projekt geändert; das Bootprojekt, welches beim nächsten Start geladen wird, ist davon nicht berührt. Wenn ein Projekt dauerhaft, also beim nächsten Start im Antrieb laufen soll, muss dazu ein Bootprojekt geladen werden.

Online-Change Während der Inbetriebnahme ist es möglich über den so genannten "Online-Change" das Projekt ohne Anhalten der Steuerung in Grenzen zu modifizieren; dies betrifft nicht das eventuell geladene Bootprojekt.

Beim Online-Change ist jedoch zu beachten, dass die Programmbearbeitung je nach Programmgröße kurzzeitig beeinträchtigt werden kann. Zeitkritische Tasks können dabei ggf. zu einem Watchdog führen.



Auch Programme, die den Echtzeitkanal benutzen, können Online geändert werden. Der zyklische Datenaustausch wird automatisch vom System umkonfiguriert.

Bootprojekt laden

Damit ein Projekt auch nach dem Wiedereinschalten läuft, muss es als Bootprojekt geladen werden. Bei "Bootprojekt laden" wird das aktuelle Projekt in den Antrieb geladen.



Zum Laden des Bootprojekts müssen Sie "Online" sein (am Antrieb eingeloggt). Andernfalls werden die Bootdateien nur auf dem PC erzeugt aber nicht in den Antrieb übertragen.



Bei Verwendung des Steuerteils mit externer Speicherkarte ist es möglich das Bootprojekt alternativ als Datei auf der Speicherkarte zu speichern (spezielle "MLD-Konfiguration" in IndraWorks bzw. entsprechende Parametrierung von P-0-1367).

Das binär übersetzte Programm wird in Parametern abgelegt und dabei komprimiert. Dazu wird ein speziell abgestimmtes Komprimierverfahren eingesetzt, welches eine Komprimierung auf eine Restgröße von typisch 35% bis 50% erreicht.

Aktuell wird das Bootprojekt in den folgenden Parametern gespeichert (die Parameter werden vom System verwaltet und dürfen vom Anwender nicht direkt geändert werden):

- P-0-1352, SPS Anwenderprogramm Verwaltungsdaten
- P-0-1353, SPS Anwenderprogramm Bereich 0
- P-0-1354, SPS Anwenderprogramm Bereich 1
- P-0-1355, SPS Anwenderprogramm Bereich 2
- P-0-1356, SPS Anwenderprogramm Bereich 3



Für die Größe des Bootprojekts stehen ca. 650 kB in Parametern und 8 MB auf der externen Speicherkarte zur Verfügung, um auch größere Programme abspeichern zu können und auch die SPS-Quellen und Anwenderdateien in Parametern ablegen zu können.

Ist ein Bootprojekt zu groß, kann es nicht gespeichert werden und es kommt während des Ladevorgangs zu einer Fehlermeldung.

Das Laden des Bootprojekts wird jedoch nicht sofort abgebrochen; nach einer Weile kommt noch eine Fehlermeldung, dass das Bootprojekt nicht erzeugt werden konnte.

Automatisches Laden des Bootprojekts

Bei einem neuen Projekt ist das automatische Erzeugen des Bootprojekts nach jedem Download aktiviert. Damit wird vermieden, dass Änderungen am Projekt zwar geladen werden, es aber versäumt wird, ein Bootprojekt zu erzeugen.



Falls gerade bei größeren Projekten und einer seriellen Verbindung die Übertragungszeiten stören, kann das automatische Laden des Bootprojektes in den Zielsystemeinstellungen (IndraLogic) abgestellt werden. Achten Sie aber dann darauf, vor dem Ausloggen das Bootprojekt zu laden!

7.4.5 IndraWorks-Projekt archivieren / wiederherstellen

Sie haben in IndraWorks die Möglichkeit, **ganze Projekte** sowohl auf dem lokalen Dateisystem als auch auf einem per Netzwerk angebundenen FTP-Server (Gerät oder Computer) zu archivieren. Diese Archive können auf dem Dateisystem des lokalen Rechners wiederhergestellt werden.

Das Archivieren und Wiederherstellen eines IndraWorks-Projekts ist in der allgemeinen Dokumentation zu IndraWorks beschrieben, Kapitel "Projekte archivieren und wiederherstellen".

Im Gegensatz zum "Archivieren", haben Sie in IndraWorks außerdem die Möglichkeit, **nur Teile von Projekten zu exportieren** (siehe "[MLD-Projekt archivieren](#)").

Weitere Unterschiede zwischen "Archivieren" und "Exportieren" sind in der nachfolgenden Tabelle ersichtlich:

Eigenschaft	Archivieren	Exportieren
ganzes Projekt	✓	✓
Teil eines Projekts	–	✓
komprimiert	✓	–
passwortgeschützt	✓	–
Ablage auf dem Dateisystem	✓	✓
Ablage auf einem FTP-Server	✓	–

Tab. 7-2: Unterschiede zwischen "Archivieren" und "Exportieren"

7.4.6 Quellcode laden

In IndraLogic kann in eingeloggtem Zustand mit **Online ► Quellcode laden** die gesamte SPS-Projektinformation in eine Datei gepackt werden. Ist im Antrieb eine externe Speicherkarte gesteckt, dann kann diese Datei (gepackter Quellcode) auf die Speicherkarte geschrieben werden.



Es wird empfohlen diese Funktion nicht zu nutzen.

7.4.7 Projekt in der Steuerung öffnen

In IndraLogic ist in dem über **Datei ► Öffnen** geöffneten Dialog eine Schaltfläche "SPS..." zu finden. Besteht eine Verbindung zu einem Antriebsregelgerät und wird diese Schaltfläche betätigt, dann wird nach der Auswahl eines Targets und der Einstellung der Kommunikationsparameter ein Projekt in der Steuerung geöffnet.



Es wird empfohlen diese Funktion nicht zu nutzen.

7.4.8 MLD-Projekt archivieren

Zielsetzung

Um ein MLD-Projekt an einer baugleichen Anlage in Betrieb zu nehmen, besteht die Möglichkeit das Projekt zu archivieren. Ein archiviertes Projekt beinhaltet alle Daten und Einstellungen, so dass es möglich ist, sich an einer bereits laufenden Anlage einzuloggen (eine Online-Verbindung herzustellen) ohne erneut das MLD-Projekt kompilieren zu müssen.



Es empfiehlt sich ein Paket mit allen benötigten Bestandteilen zusammenzustellen (z. B. in Form einer CD oder DVD), um aus jedem beliebigen PC, der über die entsprechenden Systemvoraussetzungen verfügt, ein Programmiersystem für IndraMotion MLD zu machen. Das Paket sollte folgendes enthalten:

- Die gleiche oder höhere Version einer IndraWorks MLD-Version, wie sie bei der Archivierung vorhanden war.
- Das Archiv (.zip) des MLD-Projekts.

MLD-Projekt zum Archivieren vorbereiten

Um sicherzustellen, dass es möglich ist, sich nach dem Wiederherstellen des archivierten Projekts an dem laufenden MLD-Projekt ohne erneutes Kompilieren einloggen zu können, müssen in IndraWorks vor dem Archivieren folgende Schritte durchgeführt werden:

1. **Projekt ► SPS-Compilerversion...** aufrufen und sicherstellen, dass die aktuellste SPS-Compilerversion ausgewählt ist.

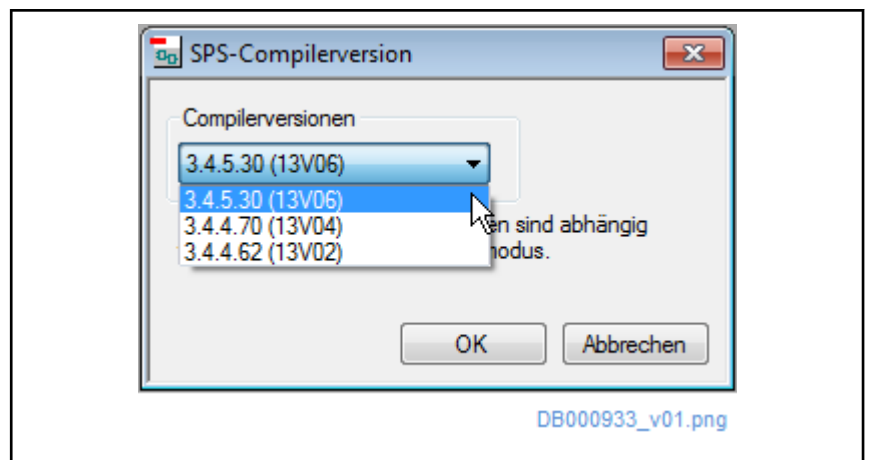


Abb. 7-24: SPS-Compilerversion prüfen

2. **Erstellen ► Alles bereinigen** ausführen.
3. **Erstellen ► Code erzeugen** ausführen.
4. **Debuggen ► Einloggen** ausführen.
5. **Debuggen ► Start** ausführen.
6. Wenn in den Eigenschaften der Applikation die Bootapplikation nicht implizit beim Download erzeugt wird, dann muss dies noch über **Debuggen ► Bootapplikation erzeugen** gemacht werden.
7. **Debuggen ► Ausloggen** ausführen.

Archivieren des MLD-Projekts

Nachdem das MLD-Projekt zum Archivieren vorbereitet wurde, kann es nun archiviert werden.



Das Archivieren kann online und offline durchgeführt werden.

Offline wird das MLD-Projekt mit den Offlinedaten archiviert. Bevor die Archivierung begonnen wird, wird im Dialog "Archivierung der Antriebsparameter" die Möglichkeit angeboten die Offlinedaten zu aktualisieren.

Folgende Schritte zum Archivieren des IndraWorks MLD-Projekts müssen durchgeführt werden:

1. IndraWorks MLD starten.
2. **Projekt ► Archivieren...** auswählen.
3. Archivname und Kommentar für das Archiv festlegen.

Abb. 7-25: Archiv-Eigenschaften vergeben



Das Archiv kann optional mit einem Passwort geschützt werden.

4. **Weiter>>** betätigen.
5. In dem folgenden Dialog müssen die Archiv-Zieleinstellungen festgelegt werden [Ablage auf dem Dateisystem und/oder auf einem per Netzwerk angeschlossenen FTP-Server (Gerät oder Computer)].

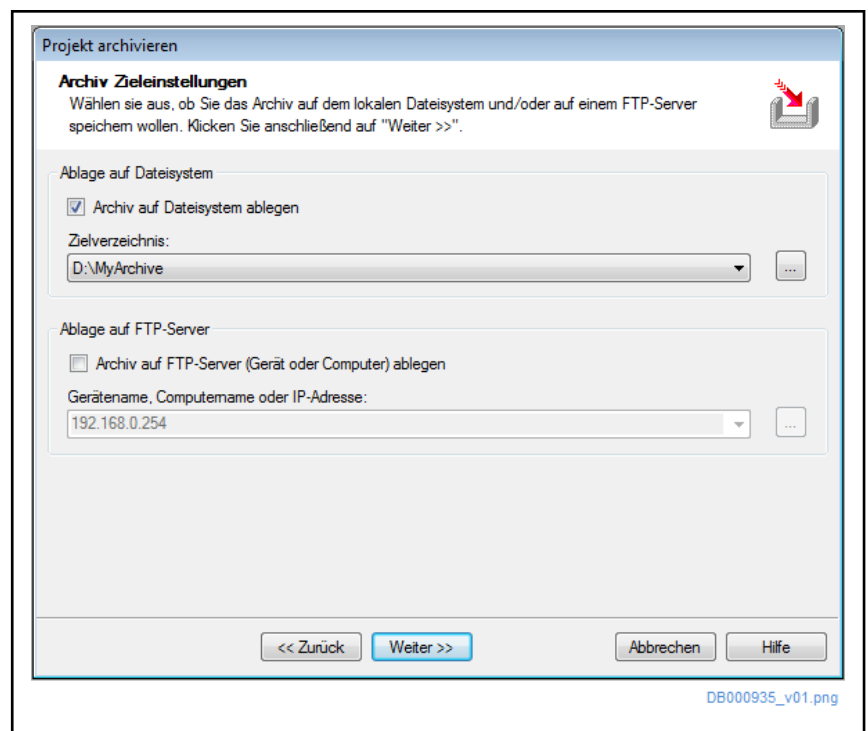


Abb. 7-26: Archiv-Zieleinstellungen festlegen

- **Ablage auf Dateisystem**

Wenn Sie das Archiv auf dem Dateisystem ablegen möchten, dann setzen Sie einen Haken bei **Archiv auf Dateisystem ablegen**; das Eingabefeld **Zielverzeichnis** wird beschreibbar. Legen Sie hier fest, in welchem Verzeichnis des Dateisystems das Archiv abgelegt werden soll.

- **Ablage auf FTP-Server**

Wenn Sie das Archiv auf einem FTP-Server ablegen möchten, dann setzen Sie einen Haken bei **Archiv auf FTP-Server (Gerät oder Computer) ablegen**; das Eingabefeld **Gerätename, Computename oder IP-Adresse** wird beschreibbar. Legen Sie hier fest, auf welchem Zielgerät das Archiv gespeichert werden soll. Dazu gibt es vier Möglichkeiten:

1. IP-Adresse
ODER
2. Computernamen des Zielgerätes
ODER
3. Auswahl über die Auswahlliste. Diese enthält alle FTP-fähigen Geräte des aktiven Projektes sowie die letzten fünf bei der Archivierung verwendeten Zielgeräte (Gerätename, IP-Adresse oder Computernamen).
ODER
4. Mit der Schaltfläche ... kann das Zielgerät aus einer Liste aller FTP-fähigen Geräte des aktiven Projektes ausgewählt werden.

6. **Weiter>>** betätigen.

⇒ Es wird automatisch eine Verbindung zum eingestellten Gerät aufgebaut. Verbindungsstörungen zum Zielgerät werden per Fehlermeldung angezeigt.

7. In dem folgenden Dialog müssen die Archivbestandteile fest gelegt werden:

Wählen Sie zu jedem im linken Navigationsbereich angezeigten, archivierbaren Element bzw. Gerät, im rechten Bereich den Umfang der durchzuführenden Archivierung aus.



Im Standard-Fall sollten alle Elemente ausgewählt werden!

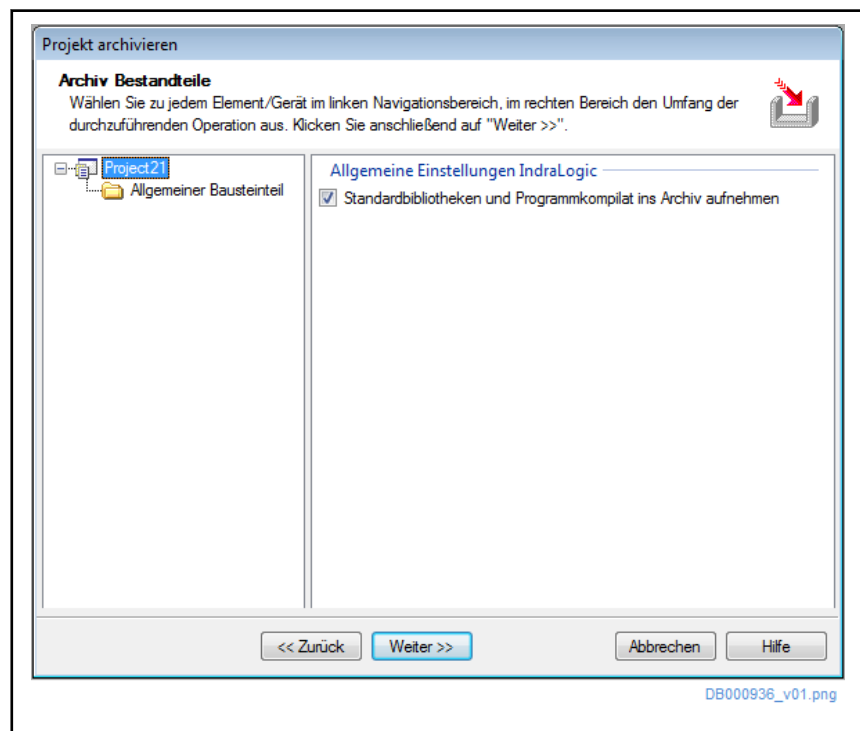


Abb. 7-27: Archiv-Bestandteile festlegen

8. **Weiter>>** betätigen.
9. In dem folgenden Dialog können die Eingaben überprüft werden.

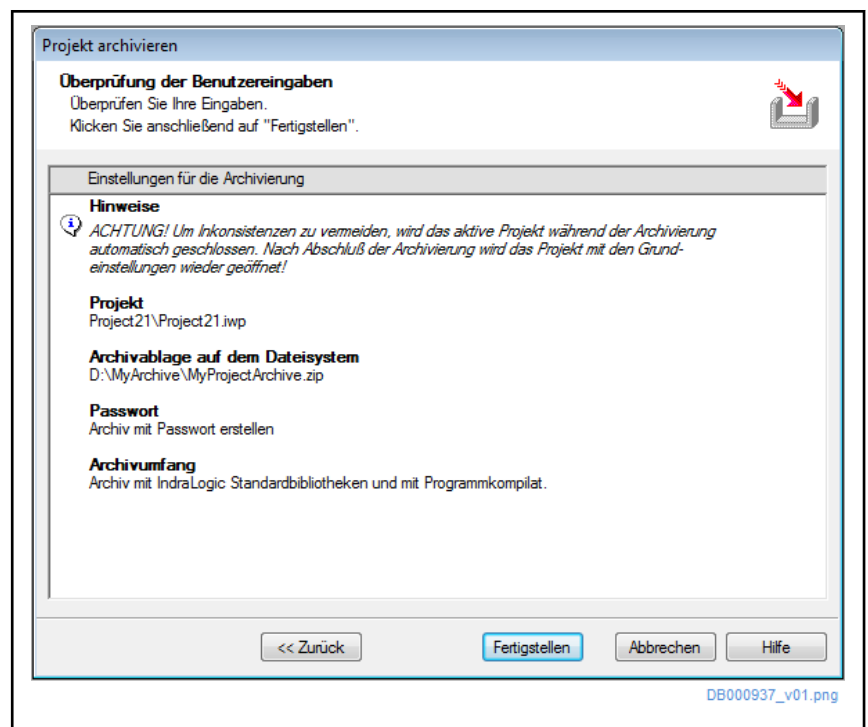


Abb. 7-28: Benutzereingaben der Archivierung prüfen

10. Zur Erstellung des Archivs auf **Fertigstellen** klicken.

Archiviertes MLD-Projekt wiederherstellen

Nachfolgend wird beschrieben, was zu tun ist, um ein archiviertes MLD-Projekt wieder herzustellen.

Stellen Sie sicher, dass auf dem PC, mit dem das archivierte MLD-Projekt wieder hergestellt wird, die gleiche oder eine höhere Version von IndraWorks MLD installiert ist, wie sie bei der Archiverzeugung vorhanden war.

Wenn das archivierte MLD-Projekt in einem CCD-Verbund eingesetzt werden soll, muss der CCD-Verbund vor der Wiederherstellung des MLD-Projekts in Betrieb genommen werden:

1. Folgende Parameter müssen entsprechend der Anlagenkonfiguration eingestellt werden:
 - CCD-Slave: Antriebsadresse (P-0-4089.0.3) und eventuell Auswahl der Funktionspakete (P-0-2003).
 - CCD-Master: Eventuell Auswahl der Funktionspakete (P-0-2003) und Einstellen der TCP/IP-Kommunikation (P-0-1531 ... P-0-1533).



Änderungen der Funktionspaketfreischaltung und der TCP/IP-Kommunikationseinstellungen werden erst beim nächsten Neustart des Antriebs wirksam.

2. In IndraWorks MLD die gewünschte Kommunikationsart (Seriell oder TCP/IP) einstellen und mit dem Antrieb (CCD-Master) online gehen ("Online schalten" im Kontextmenü des Antriebsknotens auswählen).

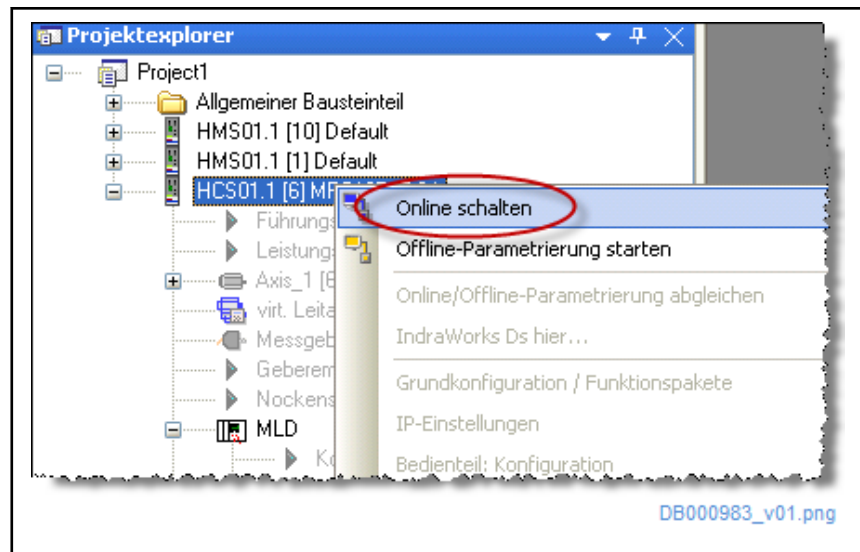


Abb. 7-29: Antrieb "Online schalten"

3. In dem Funktionsbaum das Kontextmenü von **Sercos** aufrufen, **CCD: Basis-Einstellungen** wählen und den CCD-Verbund konfigurieren.

Das archivierte IndraWorks MLD-Projekt kann wie folgt wieder hergestellt werden:

1. **Projekt ► Wiederherstellen...** auswählen.

⇒ Der Dialog "Projekt aus Archiv wiederherstellen", "Auswahl Wiederherstellungstyp" wird geöffnet.

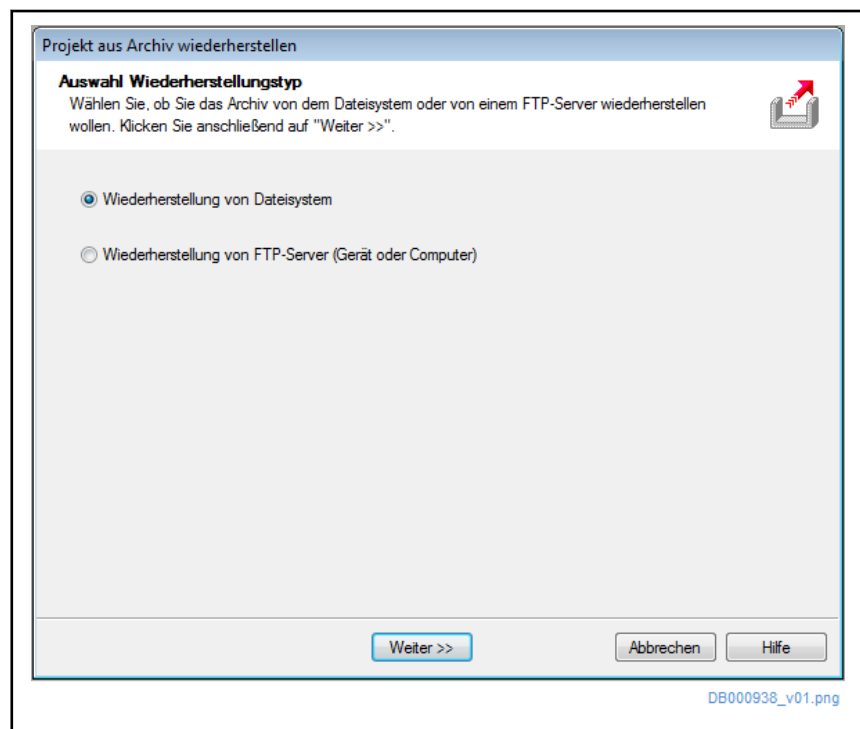


Abb. 7-30: Auswahl des Wiederherstellungstyps

2. "Wiederherstellung von Dateisystem" oder "Wiederherstellung von FTP-Server (Gerät oder Computer)" wählen.
3. **Weiter>>** betätigen.

4. a.) Wenn "Wiederherstellung von Dateisystem" gewählt wurde, dann muss im folgenden Dialog das Archiv ausgewählt werden.

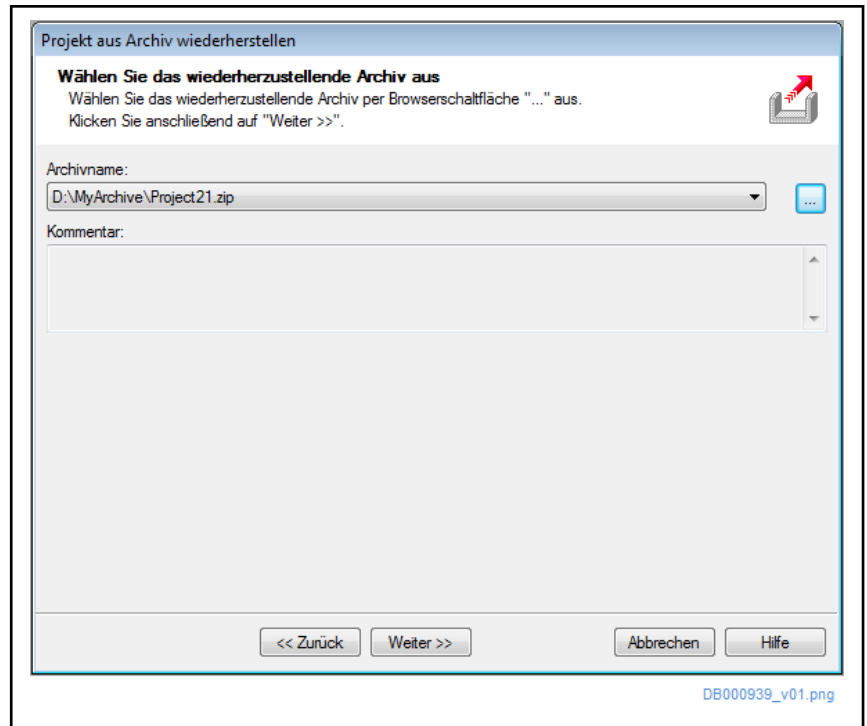


Abb. 7-31: Wiederherstellen eines Archivs von dem Dateisystem

- b.) Wenn "Wiederherstellung von FTP-Server (Gerät oder Computer)" gewählt wurde, muss eine Verbindung mit dem jeweiligen Gerät hergestellt werden.

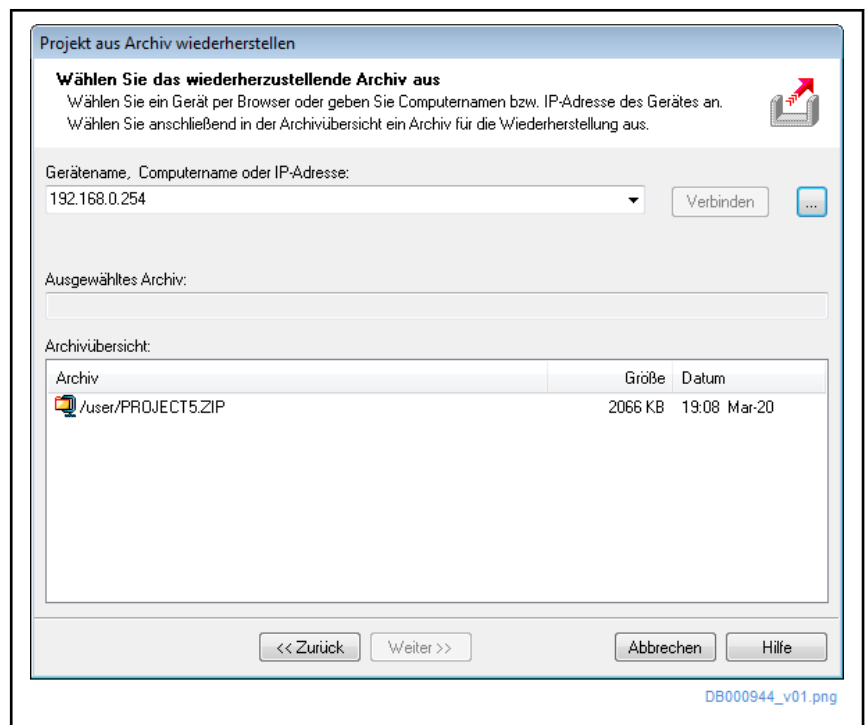


Abb. 7-32: Wiederherstellen eines Archivs von einem FTP-Server

Dazu gibt es vier Möglichkeiten:

4.1 IP-Adresse

ODER

4.2 Computername des Zielgerätes

ODER

4.3 Auswahl über die Auswahlliste. Diese enthält alle FTP-fähigen Geräte des aktiven Projektes sowie die letzten fünf bei der Wiederherstellung verwendeten Zielgeräte (Gerätename, IP-Adresse oder Computernamen).

ODER

4.4 Mit der Schaltfläche

Bei Geräteauswahl über die Auswahlliste oder über den Gerätebrowser wird automatisch eine Verbindung zu dem ausgewählten Gerät hergestellt. Bei Eingabe von IP-Adresse bzw. Computernamen wird die Verbindung zum Zielgerät über **Verbinden** hergestellt.

5. Nach dem Verbindungsaufbau werden alle auf dem Gerät verfügbaren Archive in der Liste "Archivübersicht" angezeigt.

Das wiederherzustellende Archiv auswählen.

6. **Weiter>>** betätigen.

7. Zielverzeichnis auswählen, in dem das Projekt wiederhergestellt werden soll.

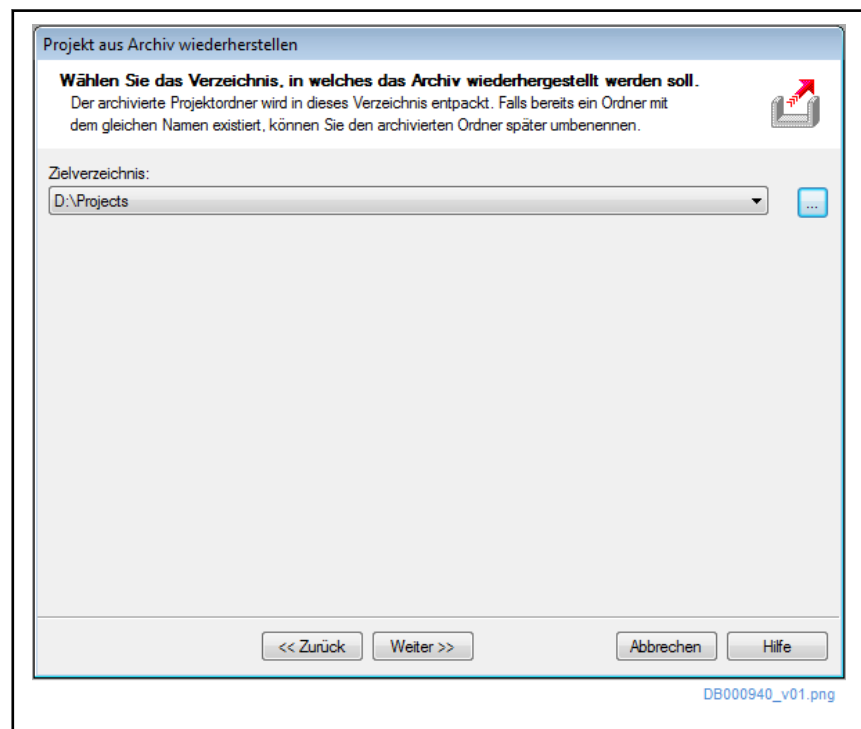


Abb. 7-33: Zielverzeichnis auswählen

8. **Weiter>>** betätigen.

9. In dem folgenden Dialog können die Eingaben überprüft werden.

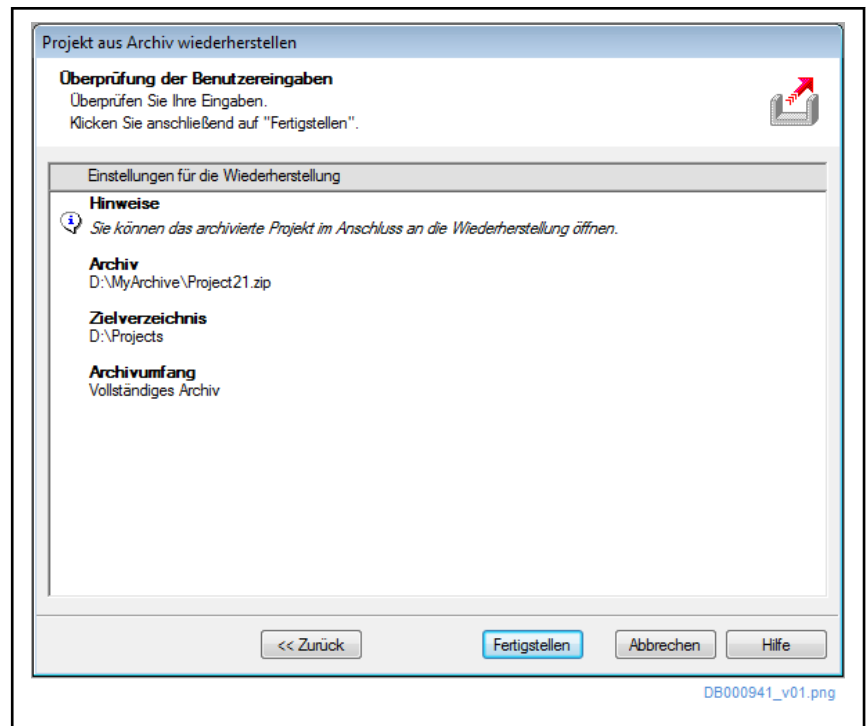


Abb. 7-34: Eingaben überprüfen

10. Zur Wiederherstellung des Archivs auf **Fertigstellen** klicken.
11. Wurde das Archiv mit einem Passwort geschützt, so muss dieses jetzt eingegeben werden.

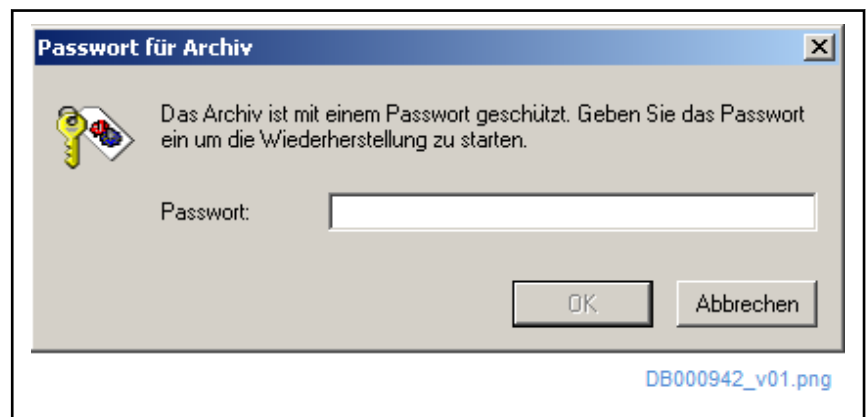


Abb. 7-35: Passwort für das Archiv eingeben

12. In dem Dialog "Zusammenfassung" besteht die Möglichkeit **Projekt beim Verlassen des Assistenten öffnen** anzuwählen. Bei Anwahl wird nach Klick auf **Schließen** das wiederhergestellte Projekt geöffnet.

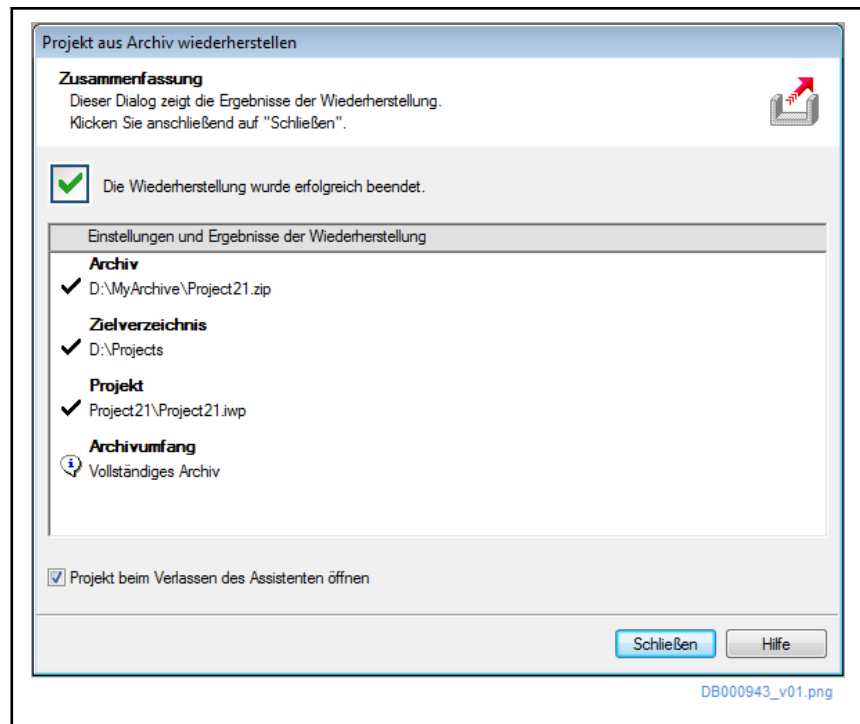


Abb. 7-36: Zusammenfassung des wieder hergestellten MLD-Projekts



Eventuelle Änderungen der Funktionspaketfreischaltung werden erst bei dem nächsten Neustart des Antriebs wirksam.

13. Ist die Firmware-Version im wiederhergestellten MLD-Projekt gleich der Firmware-Version des angeschlossenen Antriebs, kann mit dem Antrieb online gegangen werden.

7.4.9 Exportiertes MLD-Projekt importieren

Nachfolgend wird beschrieben, was zu tun ist, um ein exportiertes MLD-Projekt zu importieren.

Stellen Sie sicher, dass auf dem PC, mit dem das exportierte MLD-Projekt importiert wird, die gleiche oder eine höhere Version von IndraWorks MLD installiert ist, wie sie bei der Exporterzeugung vorhanden war.

Wenn das exportierte MLD-Projekt in einem CCD-Verbund eingesetzt werden soll, muss der CCD-Verbund vor der Wiederherstellung des MLD-Projekts in Betrieb genommen werden:

1. Folgende Parameter müssen entsprechend der Anlagenkonfiguration eingestellt werden:
 - CCD-Slave: Antriebsadresse (P-0-4089.0.3) und eventuell Auswahl der Funktionspakete (P-0-2003).
 - CCD-Master: Eventuell Auswahl der Funktionspakete (P-0-2003) und Einstellen der TCP/IP-Kommunikation (P-0-1531 ... P-0-1533).



Änderungen der Funktionspaketfreischaltung und der TCP/IP-Kommunikationseinstellungen werden erst beim nächsten Neustart des Antriebs wirksam.

2. In IndraWorks MLD die gewünschte Kommunikationsart (Seriell oder TCP/IP) einstellen und mit dem Antrieb (CCD-Master) online gehen ("Online schalten" im Kontextmenü des Antriebsknotens auswählen).

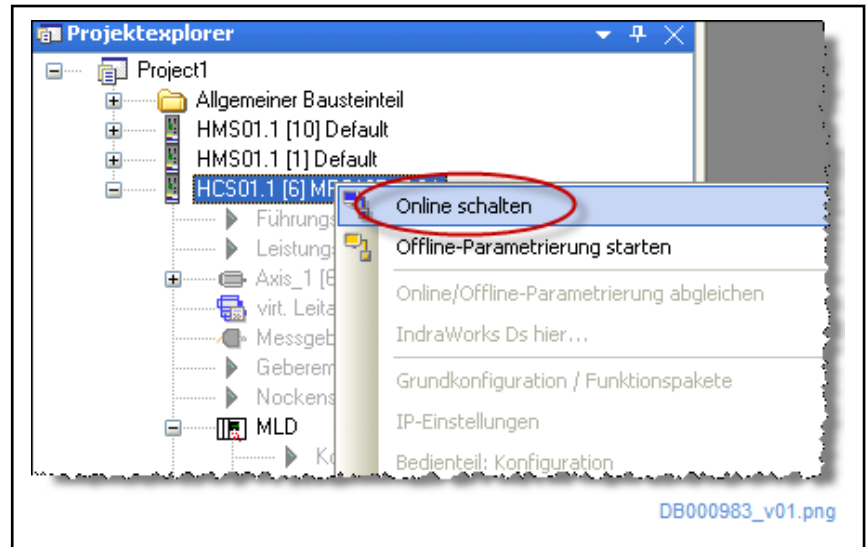


Abb. 7-37: Antrieb "Online schalten"

3. In dem Funktionsbaum das Kontextmenü von **Sercos** aufrufen, **CCD: Basis-Einstellungen** wählen und den CCD-Verbund konfigurieren.

Das exportierte IndraWorks MLD-Projekt kann wie folgt importiert werden:

1. **Projekt ► Importieren...** auswählen.
⇒ Der Standard-Windows-Dialog "Datei öffnen" wird geöffnet.
2. MLD-Projekt auswählen und **Öffnen** betätigen.
⇒ Der Import-Assistent wird geöffnet.
3. Im Projekt auswählen welche Teile des Projekts importiert werden sollen.

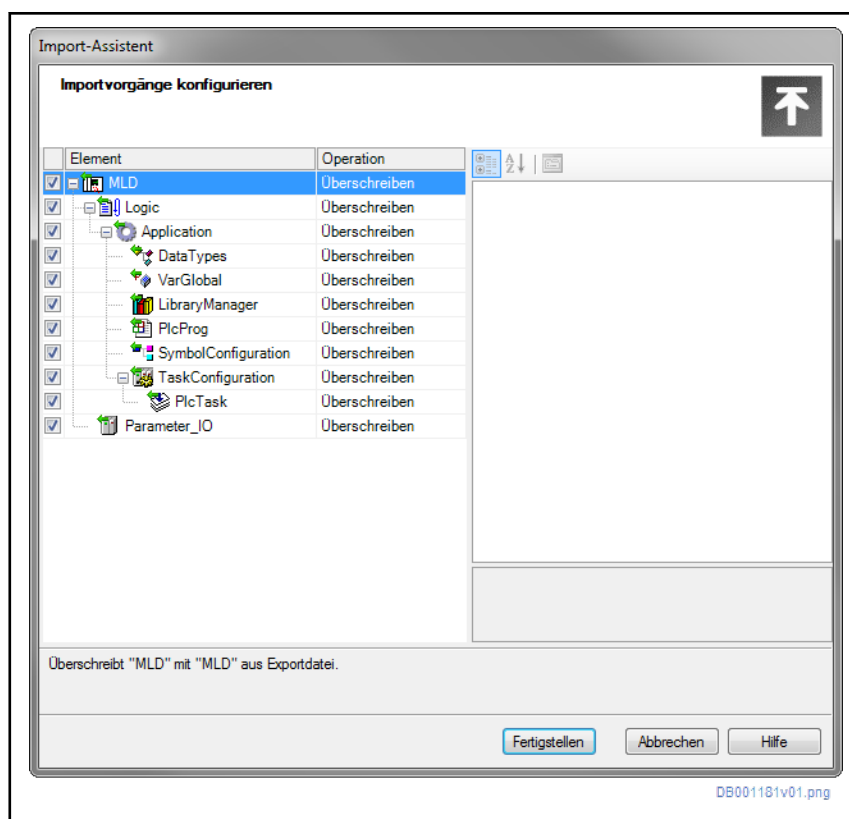


Abb. 7-38: Auswahl der Teile des Projekts, die importiert werden sollen

4. Starten Sie den Importvorgang, indem Sie auf **Fertigstellen** klicken.
 ⇒ Nach dem Importvorgang, wird eine Zusammenfassung angezeigt.

7.4.10 Update der IndraLogic-Geräteversion

Aufgrund von Fehlerbehebungen, zusätzlichen Bibliotheken,... kann es notwendig werden, eine IndraLogic-Geräteversion (IndraWorks-Packages) nachzuinstallieren. In der Regel werden IndraWorks-Packages über den Support von Bosch Rexroth zur Verfügung gestellt oder mit einer neuen IndraWorks MLD-Version geliefert.

Der Dialog zur Nachinstallation von IndraWorks-Packages kann im Projekttexplorer über das Kontextmenü **MLD ► IndraLogic Geräteversion ändern** aufgerufen werden.

In dem Dialog "IndraLogic Geräteversion ändern" können IndraWorks-Packages durch Anklicken des Links "IW-Package hinzufügen" nachinstalliert werden.



Bei einem Update von IndraWorks bleiben alle vorhandenen Bibliotheken auf dem Rechner verfügbar. Es ist nicht notwendig ein "älteres" IndraWorks-Package erneut zu installieren.

7.4.11 Ändern der für ein SPS-Projekt gültigen Bibliotheken

Beim Anlegen eines SPS-Projekts werden die zu diesem Zeitpunkt aktuellen Bibliotheken automatisch in das Projektverzeichnis kopiert.

Bibliotheken eines vorhandenen SPS-Projekts aktualisieren

Sollen in einem SPS-Projekt die Bibliotheken auf den neuesten Stand gebracht werden, ist die Installation eines IndraWorks-Package (IW-Package) notwendig. Das IndraWorks-Package enthält Informationen zur Geräteversion und allen benötigten Bibliotheken.

IW-Package installieren

1. Rufen Sie im Projektextplorer am Knoten **MLD** das Kontextmenü auf und wählen Sie **IndraLogic Geräteversion ändern** aus.
Der Dialog "IndraLogic Geräteversion ändern" wird aufgerufen.
2. Klicken Sie auf den Link "IW-Package installieren":

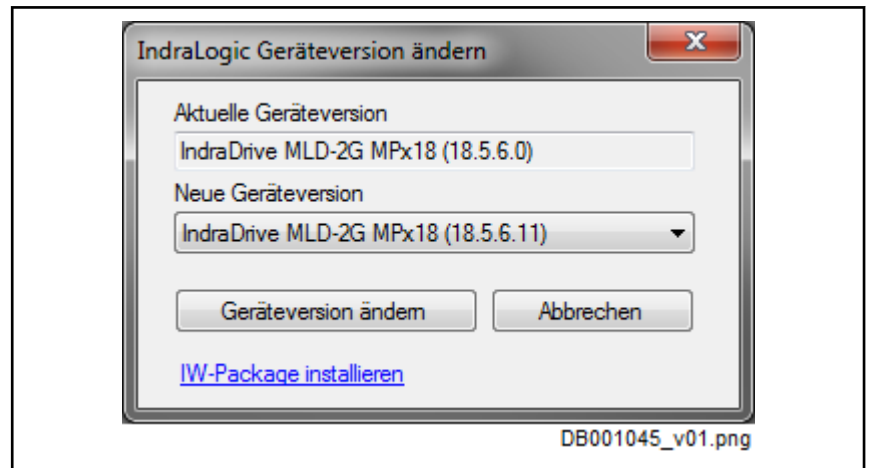


Abb. 7-39: IW-Package installieren

Der Windows-Dialog "Datei öffnen" wird geöffnet.

3. Wählen Sie das zu installierende Package aus. (Es dürfen nur Dateien mit der Endung ".iwpackage" ausgewählt werden!)



Die Installation kann einige Zeit in Anspruch nehmen.

Nach der Installation steht eine neue Geräteversion in der Klappliste zur Auswahl zu Verfügung.

4. Wählen Sie über die Klappliste die neue Geräteversion aus.
5. Betätigen Sie die Schaltfläche **Geräteversion ändern**.
6. Nach dem Umstellen der Geräteversion sollte das Projekt bereinigt werden:
Im Hauptmenü **Erstellen ► Alles bereinigen** ausführen.
7. Das bereinigte Projekt muss neu übersetzt werden:
Im Hauptmenü **Erstellen ► Alles übersetzen** ausführen.



Das bereinigte und übersetzte Projekt wird beim nächsten "Einloggen" auf die Steuerung geladen.

SPS-Projekt auf eine andere IndraLogic-Geräteversion umstellen

Soll ein SPS-Projekt auf eine andere Firmware-Release umgestellt werden oder eine andere Firmware ausgewählt werden, muss die Geräteversion umgestellt werden; das Vorgehen ist unter "[Kompatibilität zwischen Versionen](#)", Arbeitsablauf "Geräteversion umstellen" beschrieben.

7.4.12 Import einer externen Bibliotheksdatei

Vorgehensweise, um eine externe Bibliotheksdatei zu importieren:

1. IndraWorks MLD starten.
2. Im Projektextplorer das Kontextmenü (Mausklick rechts) des Logic-Knotens öffnen.

3. Bibliotheken importieren... anwählen.

Der Dialog "Importieren einer IndraLogic-Bibliotheksdatei" wird geöffnet.

4. Die zu importierende Bibliothek auswählen und mit "Öffnen" bestätigen



Bei mehrfachem Import von Bibliotheken die im gleichen Verzeichnis liegen empfiehlt es sich die Navigation mit z.B. dem Explorer zu machen. Den Pfad im Explorer aus "Adresse" kopieren, im Import-Dialog in "Dateiname" einfügen und mit Eingabe bestätigen.

7.5 Antrieb als SPS-Gerät

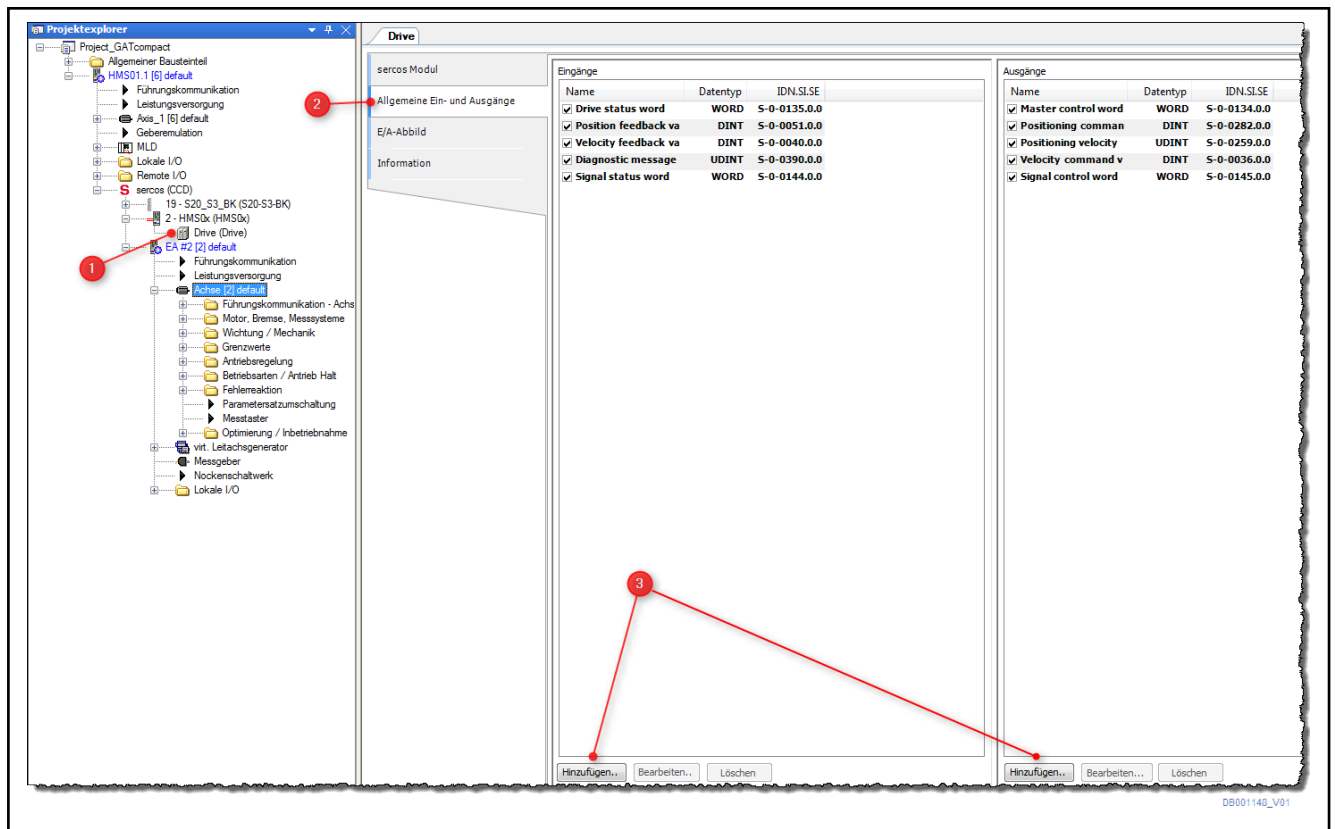
7.5.1 Allgemeines

Ein Antrieb als SPS-Gerät hat, im Gegensatz zu einer Achse, keine Motionfunktionalität, d. h. die Funktionbausteine der Bibliothek "PLCopen.library" können nicht angewendet werden. Der Antrieb wird von der SPS kommandiert und ist frei programmierbar.

Im IndraWorks-Projekt kann der Antrieb durch Ziehen des Gerätes aus dem Bibliotheksfenster (Ordner "Peripherie / Sercos") auf den Sercos Knoten angelegt werden. IndraDrive-Geräte sind ab MPx18VRS in der Bibliothek der Peripheriegeräte "Sercos" zu finden. Beispiel: IndraDrive Cs Einzelachsgerät als HCS0x.

Antriebe beliebiger Hersteller können in der Gerätedatenbank nachinstalliert werden.

Zur Konfiguration der zyklischen Kommunikation und Diagnose der SPS-Geräte stehen in IndraWorks-Dialoge zur Verfügung.



- 1 Doppelklick auf das Sercos Gerät "HCS0x" unterhalb des Sercos Knotens
- 2 Reiter "Allgemeine Ein- und Ausgänge" anwählen
- 3 Durch "Hinzufügen" können Parameter angewählt werden bzw. gelöscht werden

Abb. 7-40: Konfiguration der zyklischen Kommunikation

Die Konfiguration der Kommunikation und die Parametrierung der Antriebe erfolgt im Falle von IndraDrive über IndraWorks Ds.

7.5.2 Sercos III Antrieb als SPS-Gerät

Übersicht

Nachstehend wird beispielhaft die Umsetzung des folgenden Bewegungsablaufes dargestellt:

- Positionieren auf eine absolute Position
- Relatives Positionieren (Zielposition = aktuelle Position + Positionsvorgabe)
- Additives Positionieren (Zielposition = letzte Zielposition + Positionsvorgabe)
- Geschwindigkeitsregelung
- Antrieb Halt

Es können beliebige weitere Bewegungsabläufe und antriebsbezogene Funktionen implementiert werden. Hier in diesem Beispiel wird nur ein kleiner Teil der Funktionalität eines IndraDrive Antriebs genutzt.

Die Schritte der Antriebsinbetriebnahme können anhand dieses Abschnitts nachvollzogen werden; am Ende sollte mit Hilfe des Programmlistings ein Bewegen des Antriebs mit dem beschriebenen Bewegungsablauf möglich

sein (siehe [Kap. "SPS-Programmierung" auf Seite 245](#)). Am Ende des jeweiligen Abschnittes werden die notwendigen Arbeitsschritte kurz beschrieben.



Die Nutzung als reale Achse (Slaveachse im MLD-M-System) hat viele Vorteile gegenüber der hier beschriebenen Nutzung als SPS-Gerät mit freier Programmierung. Zur Nutzung als SPS-Gerät sind erweiterte Kenntnisse der benutzten Sercos Parameter und zusätzlicher Programmieraufwand notwendig.

Anlegen des SPS-Antriebs

Ein IndraWorks-Projekt mit einem MLD-Master ist angelegt. Das Funktionspaket "MLD" ist aktiviert. Die zugehörige Gerätebeschreibungsdatei des Antriebs, der als SPS-Gerät eingebunden werden soll, ist installiert.

1. Antrieb aus dem Bibliotheksfenster (Ordner "Peripherie", Ordner "Sercos") auf den Sercos Knoten ziehen und Sercos Adresse konfigurieren.
2. SPS-Programm einloggen (SPS-Programm übersetzen und auf den Antrieb laden).
3. SPS-Antrieb wird automatisch erkannt, der CCD-Verbund wird in Sercos Phase "P2" geschaltet.
4. Der Masterantrieb und der konfigurierte Slaveantrieb kann jetzt in P4 (Operation Mode) geschaltet werden.

Parametrierung

Die Parametrierung der Antriebe erfolgt im Falle von IndraDrive über IndraWorks bzw. für andere Antriebe herstellerspezifisch mit dem jeweiligen Tool.

Alternativ zu "IndraWorks Ds hier..." kann die Kommunikation direkt zwischen PC und Antrieb erfolgen. Der PC kann an einem freien Sercos Port angeschlossen werden oder über einen "Sercos III NetSwitch" (Bestellnr.: R911328254 HAWA HGS NS-S3-1NRT) kommunizieren. Alternativ kann mit dem Windows Befehl "route" die IP-Kommunikation über die Steuerung geroutet werden.



Alternativ zur Nutzung von IndraWorks Ds kann auch in das IndraWorks-Projekt der Steuerung ein IndraDrive aus der Bibliothek (Drive and Control) auf den Hauptknoten gezogen werden (parallel zur Steuerung). Ein vorher mit IndraWorks Ds erzeugtes Parameterfile ist hilfreich zur Konfiguration.

Die Konfiguration der zyklisch zu übertragenden Parameter erfolgt bei Sercos, im Gegensatz zu anderen Feldbussen, in IndraWorks unterhalb des Sercos Knotens.

Um die in [Kap. "Übersicht" auf Seite 239](#) genannten Bewegungsabläufe umzusetzen, ist mit IndraWorks mindestens die folgende Parametrierung auszuführen:

- Hauptbetriebsart "Geschwindigkeitsregelung"
- 1. Nebenbetriebsart "Momentenregelung"

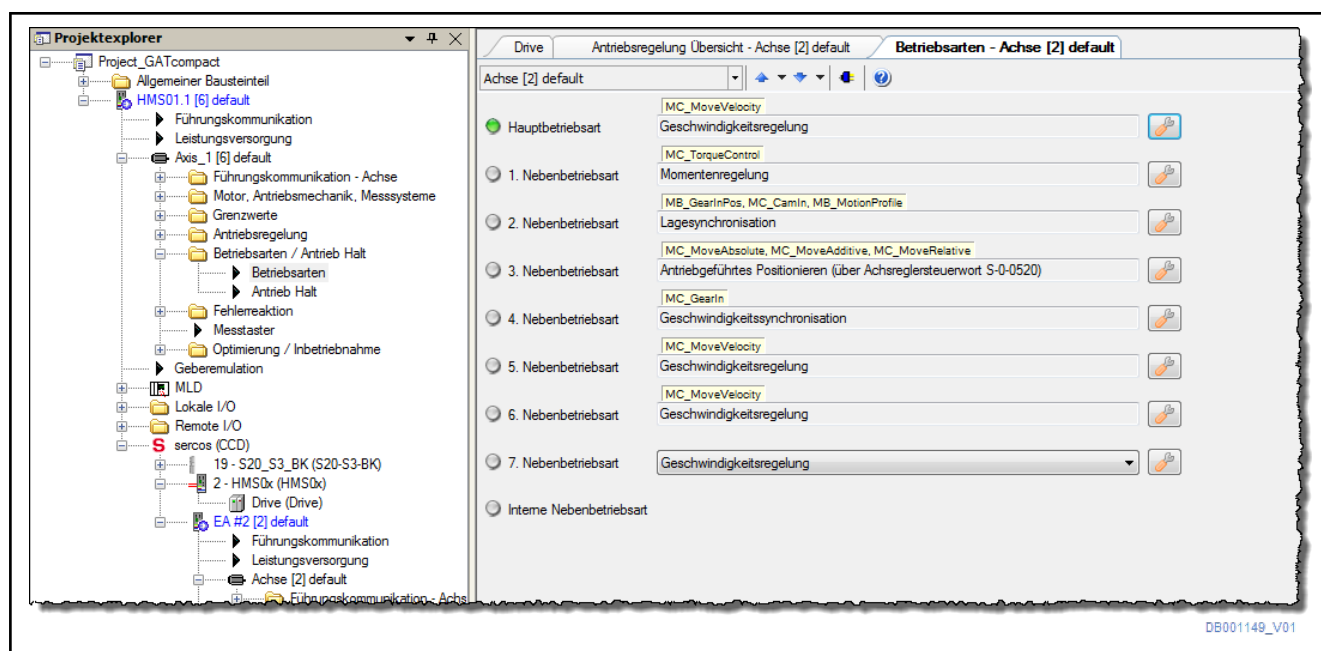


Abb. 7-41: Parametrierung der Betriebsarten mit IndraWorks Ds

- Signalsteuerwort
 - Bit 0: S-0-0346, Bit 0 "Übernahme Positioniersollwert durch Toggeln"
 - Bit 1: S-0-0346, Bit 3 "Art des Positioniersollwerts, 0: absolut, 1: relativ (abhängig von Bit 4)"
 - Bit 2: S-0-0346, Bit 4 "Bezugspunkt für Positioniersollwerte, 0: letzte wirksame Zielposition (S-0-0430), 1: aktiver Lageistwert (S-0-0386)"
- Signalstatuswort:
 - Bit 0: S-0-0331, Bit 0 "Meldung n_ist = 0"
 - Bit 1: S-0-0342, Bit 0 "Zielposition erreicht"
 - Bit 2: S-0-0330, Bit 0 "Meldung n_ist = n_soll"
 - Bit 3: S-0-0403, Bit 0 "Status des Referenzgebers Lage-Istwert (Geber 1 bzw. 2), 0: relativ, 1: referenziert"

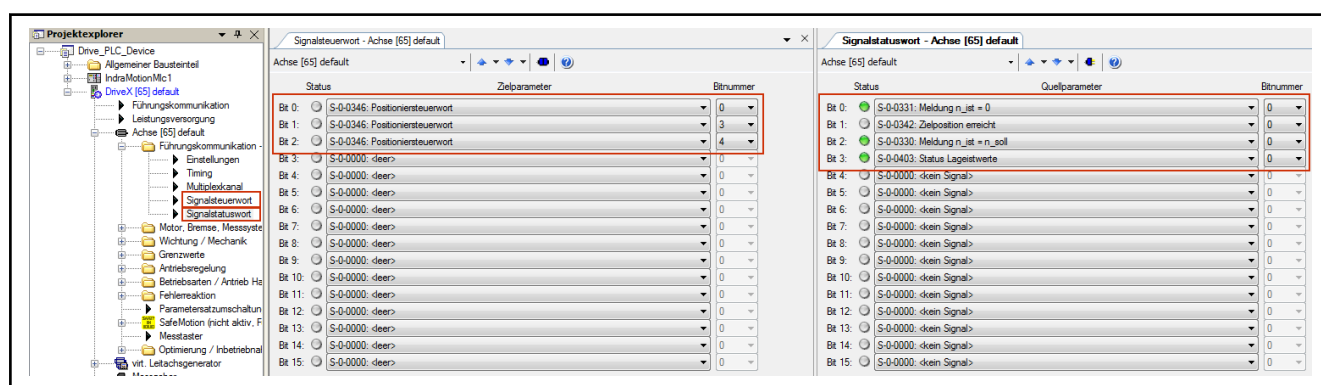


Abb. 7-42: Parametrierung Signalsteuer- und Signalstatuswort mit IndraWorks

Die weitere Parametrierung des Antriebs, z. B. Bewegungsgrenzwerte, ist ebenfalls vorzunehmen.



Um Parameter eines SPS-Gerätes aus dem SPS-Programm heraus zu lesen oder zu schreiben, können Funktionsbausteine aus der Bibliothek RIL_SERCOSIII.library, Ordner "AcyclicCommunication", benutzt werden.

Parametrierung des Antriebs mit Direktverbindung

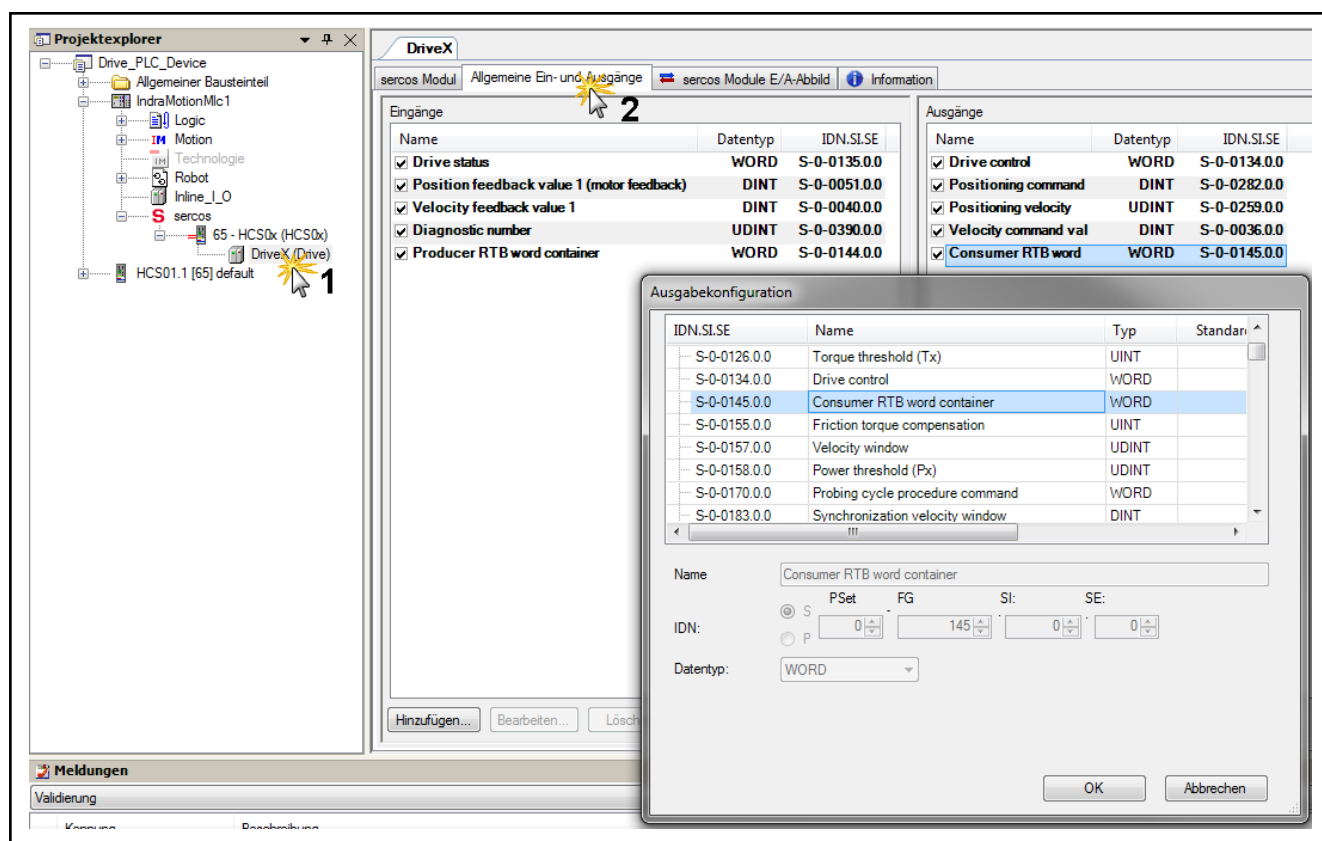
Bei Nutzung von "IndraWorks Ds hier..." (rechte Maustaste auf das Sercos Gerät) entfallen Arbeitsschritt 1. und 2.

Nur bei Direktverbindung: PC mit einem offenen Sercos Port oder über einen "Sercos III NetSwitch" verbunden. IP-Adresse des PCs mit einer zur Steuerungs- und Antriebs-IP-Adresse passenden Adresse konfiguriert.

1. Im IndraWorks-Projekt die in ["Anlegen des SPS-Antriebs"](#), Punkt 4, notierte Bridge-Adresse als IP-Adresse der Steuerung eintragen. (Online schalten sollte wieder möglich sein.)
2. IndraWorks Ds starten. "Verbindungsauswahl" → "IP-Adress-Suche". IP-Adresse des Antriebs (in ["Anlegen des SPS-Antriebs"](#), Punkt 5, notiert) eingeben und suchen. Verbinden mit dem Antrieb.
3. In IndraWorks Ds Betriebsarten parametrieren, wie in ["Parametrierung der Betriebsarten mit IndraWorks Ds"](#) gezeigt.
4. In IndraWorks Ds Signalsteuerwort und Signalstatuswort parametrieren, wie in ["Parametrierung Signalsteuer- und Signalstatuswort mit IndraWorks"](#) gezeigt.
5. In IndraWorks Ds die weitere Parametrierung des Antriebes ausführen; Bewegungsgrenzwerte, Schnellhalt-Verzögerung, Reglerparameter, Statusmeldungen usw.

Konfiguration der zyklischen Daten

Die zyklisch zu übertragenden Daten können in IndraWorks unterhalb des Sercos Knotens im Dialog "Allgemeine Ein- und Ausgänge" konfiguriert werden.



- 1 Doppelklick auf den Antriebsknoten unterhalb des Sercos Gerätes
- 2 Allgemeine Ein- und Ausgänge anwählen

Abb. 7-43: Beispielkonfiguration der zyklischen Parameter im Antriebstelegramm (AT) (Eingänge) und Master-Daten-Telegramm (MDT) (Ausgänge)

Beispielhaft sind hier die notwendigen Parameter für die Betriebsarten "Positionieren" und "Geschwindigkeitsregelung" konfiguriert.



Wenn die Gerätebeschreibungsdatei die notwendigen Informationen (z. B. welche Parameter zyklisch konfigurierbar sind, Name und Attribut der Parameter) enthält, bietet der Konfigurator eine Auswahlliste an und schlägt die notwendigen Einstellungen vor. Sind die Informationen nicht verfügbar, wird keine Auswahlliste angeboten und die Einstellungen sind vom Anwender vorzunehmen.

Der eingegebene Name wird später in dem E/A-Abbild in der Spalte "Kanal" angezeigt (siehe Abbildung "Zuordnung der SPS-Variablen zu den konfigurierten Parametern").

Abhängig vom Parameter-Attribut, muss in der SPS der richtige Datentyp (mit passender Länge; mit oder ohne Vorzeichen) gewählt werden:

Datenlänge im Attribut	Format im Attribut	Datentyp in der SPS
2 Byte	BIN oder HEX	WORD
4 Byte	BIN oder HEX	DWORD
2 Byte	DEC_MV	INT

Datenlänge im Attribut	Format im Attribut	Datentyp in der SPS
4 Byte	DEC_MV	DINT
2 Byte	DEC_OV	UINT
4 Byte	DEC_OV	UDINT

Tab. 7-3: Zuordnung Parameter-Attribut ↔ Datentyp in der SPS

Die Zuordnung der Parameter zu SPS-Variablen kann im Reiter "E/A-Abbild" erfolgen. Es können neue Variablen erzeugt werden oder die Ein- und Ausgangsparameter auf bestehende SPS-Variablen gemappt werden.

In dem folgenden Beispiel werden die Variablen neu erzeugt:

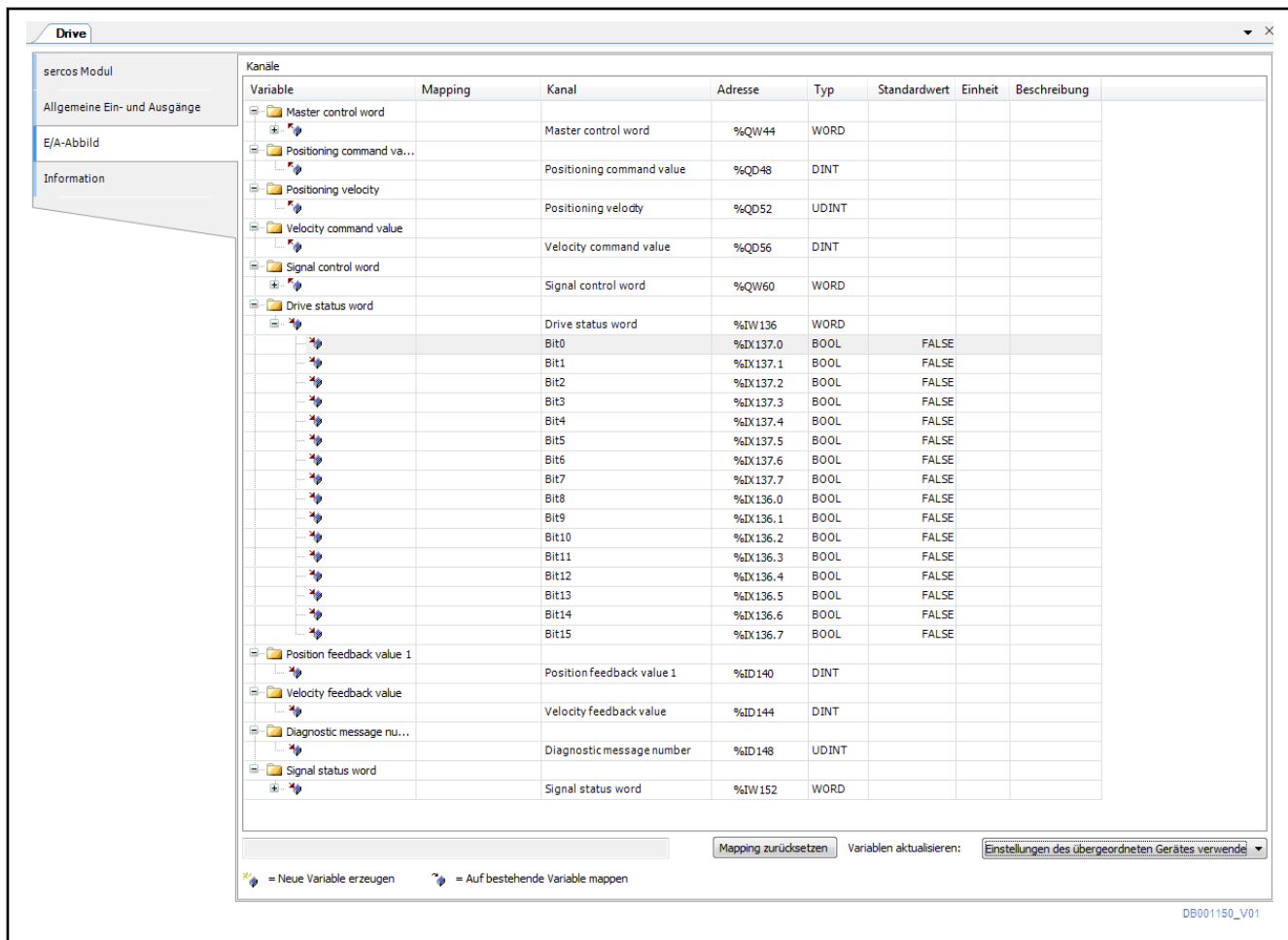


Abb. 7-44: Zuordnung der SPS-Variablen zu den konfigurierten Parametern

Konfiguration der zyklischen Daten

- In dem IndraWorks-Projekt Doppelklick auf den "Drive"-Knoten des Antriebs. Reiter "Allgemeine Ein- und Ausgänge" wählen.
- Parameter "S-0-0135, Antriebs-Status" soll konfiguriert werden.
In der Hilfe zu S-0-0135 das Attribut prüfen:
 - Länge: 2 Byte
 - Format: BIN
 - Bei den Eingängen "Hinzufügen" wählen
- Parameter "S-0-0051, Lage-Istwert Geber 1" soll konfiguriert werden.

In der Hilfe zu S-0-0051 das Attribut prüfen:

- Länge: 4 Byte
 - Format: DEC_MV
 - Bei den Eingängen "**Hinzufügen**" wählen
4. Die weiteren Parameter eingeben, bis alle Parameter wie in Abbildung "[Beispielkonfiguration der zyklischen Parameter im Antriebstelegramm \(AT\) \(Eingänge\) und Master-Daten-Telegramm \(MDT\) \(Ausgänge\)](#)" gezeigt konfiguriert sind.
 5. Wechseln zum Reiter "E/A-Abbild" und die SPS-Variablen in der Spalte "Variable" eintragen, wie in Abbildung "[Zuordnung der SPS-Variablen zu den konfigurierten Parametern](#)" gezeigt.
 6. SPS-Programm laden; jetzt sollte über IndraWorks das Schalten in "P4" möglich sein.

SPS-Programmierung

Die konfigurierten SPS-Variablen der zyklisch konfigurierten Parameter können nun in der SPS beliebig verschaltet werden.

Die Programmierung muss standardmäßig nicht in einer speziellen Task erfolgen. Nur bei besonderen Anforderungen, wie z. B. bei Gantryachsen, müssen die Vorgaben in der motionsynchronen oder Sercos synchronen Task erfolgen.

Programmbeispiel

Im folgenden Beispiel wurde dieser Ablauf umgesetzt:

- Start im MODE_OFF = AB. Die Variable "OpMode" mit 20 (ABS_POS_MODE) beschreiben, um den Ablauf zu starten
- ABS_POS_MODE = AF, absolutes Positionieren mit 4 U/min auf 42 Grad
- REL_POS_MODE = AF, relatives Positionieren mit 2 U/min um 142 Grad im Uhrzeigersinn
- ADD_POS_MODE = AF, relatives Positionieren mit 3 U/min um 42 Grad gegen den Uhrzeigersinn
- VEL_MODE = AF, Geschwindigkeitsregelung mit 42 U/min für 5 Sekunden
- MODE_AH = AH, stoppen mit der parametrisierten "Verzögerung Schnell-Halt" S-0-0372
- Ende wieder im MODE_OFF = AB. Abschalten der Reglerfreigabe sobald Stillstand erreicht ist

Durch Schreiben der Variable "OpMode" auf einen der Werte REL_POS_MODE, ADD_POS_MODE oder VEL_MODE kann der Ablauf auch an diesen Stellen gestartet werden.

SPS-Programmierung

1. Im IndraWorks-Projekt das "PlcProg" öffnen.
2. Das Programmlisting "[Deklaration](#)" in den Deklarationsteil kopieren.
3. Das Programmlisting "[Implementation](#)" in den Implementationsteil kopieren. (Seitenumbruch nicht mit kopieren bzw. hinterher wieder löschen.)
4. Übersetzen, Laden und Starten des SPS-Programmes.
5. Am Antrieb Leistung zuschalten "bb"→"Ab". Referenz prüfen (boDriveX_Homed im SPS-Projekt).

Inbetriebnahme- und Anwendungshinweise

6. Breakpoints an den Übergängen zur nächsten Betriebsart setzen (Kommentar "// set breakpoint here to check"), um die korrekte Abarbeitung zu prüfen.
7. Variable "OpMode" auf den Wert 20 schreiben. Damit sollte der Ablauf gestartet und die erste Bewegung bis zum Breakpoint ausgeführt werden.
8. SPS-Programm wieder starten bis zum nächsten Breakpoint und Bewegungsablauf prüfen. Wiederholen bis alle Schritte abgearbeitet sind.
9. Variable "OpMode" auf die Werte 30, 40 oder 50 schreiben. Damit sollte der Ablauf mit der jeweiligen Betriebsart gestartet werden.
10. Aus diesem Programm können jetzt für die endgültige Programmumsetzung Teile kopiert und weiterverwendet werden.

Deklaration

```

PROGRAM PlcProg
VAR
  OpMode:INT:=MODE_DRV_OFF; // selection operation mode and state machine
  fbTON: TON;
  rActPosition: REAL; // actual position as real
  rActVelocity: REAL; // actual velocity as real
  boError: BOOL; // error flag from status word
  boWarning: BOOL; // warning flag from status word
END_VAR

VAR CONSTANT
  // operation modes and state machine
  MODE_DRV_OFF :INT:=10;
  ABS_POS_MODE :INT:=20; ABS_POS_MODE_1 :INT:=21; ABS_POS_MODE_2 :INT:=22;
  REL_POS_MODE :INT:=30; REL_POS_MODE_1 :INT:=31; REL_POS_MODE_2 :INT:=32;
  ADD_POS_MODE :INT:=40; ADD_POS_MODE_1 :INT:=41; ADD_POS_MODE_2 :INT:=42;
  VEL_MODE :INT:=50; VEL_MODE_1 :INT:=51;
  MODE_DRV_AH :INT:=60;
  // control word S-0-0134
  CTRL_WORD_OFF:WORD :=16#0000;
  CTRL_WORD_AH:WORD :=16#C000;
  CTRL_AF_PRIMARY_OP_MODE:WORD :=16#E000;
  CTRL_AF_SECONDARY_OP_MODE:WORD:=16#E100;
  // status word S-0-0135
  MASK_STATUS_AF_OP_MODE:WORD :=16#C700;
  STATUS_AF_PRIMARY_OP_MODE:WORD :=16#C000;
  STATUS_AF_SECONDARY_OP_MODE:WORD:=16#C100;
  // Weighting
  WEIGHTING_FACTOR_POS: DINT:=10000; // S-0-0078 = -4
  WEIGHTING_FACTOR_VEL: DINT:=10000; // S-0-0046 = -4
END_VAR

```

Implementation

```

CASE OpMode OF
  MODE_DRV_OFF:
    // Ab, switched OFF
    wDriveX_ControlWord := CTRL_WORD_OFF; // all OFF...

  ABS_POS_MODE:
    // start absolute positioning to 42°
    boDriveX_PosRelNotAbs:=FALSE; // S-0-0346.3 0=absolute Pos, 1=relative Pos
    //boDriveX_PosRelNotAdd; // S-0-0346.4 0=additive (last target), 1=rel. (actual pos.)
    udiDriveX_PositionVelocity:=DINT TO UDINT(4*WEIGHTING_FACTOR_VEL); // 4 rpm * 10000
    diDriveX_PositionCmdValue:=42*WEIGHTING_FACTOR_POS; // 42 degree * 10000
    wDriveX_ControlWord := CTRL_AF_PRIMARY_OP_MODE; // AF and primary operation mode
    boDriveX_PosToggle := NOT boDriveX_PosToggle; // start positioning
    OpMode:=ABS_POS_MODE_1;

  ABS_POS_MODE_1:
    // check acknowledge power AND homed
    IF (wDriveX_StatusWord AND MASK_STATUS_AF_OP_MODE) = STATUS_AF_PRIMARY_OP_MODE
    AND boDriveX_Homed = TRUE
    THEN
      IF boDriveX_InPosition = FALSE THEN // in position can be set from last command
        OpMode:=ABS_POS_MODE_2; // next
      END_IF
    END_IF
END_CASE

```

```

ABS_POS_MODE_2:
    // check in position
    IF boDriveX_InPosition THEN
        // here we should be at 42°
        OpMode := REL_POS_MODE; // set breakpoint here to check
    END_IF

REL_POS_MODE:
    // start relative positioning 142° clockwise
    boDriveX_PosRelNotAbs:=TRUE; // S-0-0346.3 0=absolute Pos, 1=relative Pos
    boDriveX_PosRelNotAdd:=TRUE; // S-0-0346.4 0=additive (last target), 1=rel. (actual pos.)
    udiDriveX_PositionVelocity:=DINT TO UDINT(2*WEIGHTING_FACTOR_POS); // 2 rpm
    diDriveX_PositionCmdValue:=142*WEIGHTING_FACTOR_VEL; // 142 degree * 10000
    wDriveX_ControlWord := CTRL_AF_PRIMARY_OP_MODE; // AF and primary operation mode
    boDriveX_PosToggel := NOT boDriveX_PosToggel; // start positioning
    OpMode:=REL_POS_MODE_1;

REL_POS_MODE_1:
    // check acknowledge power
    IF (wDriveX_StatusWord AND MASK_STATUS_AF_OP_MODE) = STATUS_AF_PRIMARY_OP_MODE THEN
        IF boDriveX_InPosition = FALSE THEN // in position can be set from last command
            OpMode:=REL_POS_MODE_2; // next
        END_IF
    END_IF

REL_POS_MODE_2:
    // check in position
    IF boDriveX_InPosition THEN
        // here we should be at 42°+142°=184°
        OpMode := ADD_POS_MODE; // set breakpoint here to check
    END_IF

ADD_POS_MODE:
    // start additive positioning 42° counterclockwise
    boDriveX_PosRelNotAbs:=TRUE; // S-0-0346.3 0=absolute Pos, 1=relative Pos
    boDriveX_PosRelNotAdd:=FALSE; // S-0-0346.4 0=additive (last target), 1=rel. (actual pos.)
    udiDriveX_PositionVelocity:=DINT TO UDINT(3*WEIGHTING_FACTOR_VEL); // 3 rpm
    diDriveX_PositionCmdValue:=-42*WEIGHTING_FACTOR_POS; // -42 degree * 10000
    wDriveX_ControlWord := CTRL_AF_PRIMARY_OP_MODE; // AF and primary operation mode
    boDriveX_PosToggel := NOT boDriveX_PosToggel; // start positioning
    OpMode:=ADD_POS_MODE_1;

ADD_POS_MODE_1:
    // check acknowledge power
    IF (wDriveX_StatusWord AND MASK_STATUS_AF_OP_MODE) = STATUS_AF_PRIMARY_OP_MODE THEN
        IF boDriveX_InPosition = FALSE THEN // in position can be set from last command
            OpMode:=ADD_POS_MODE_2; // next
        END_IF
    END_IF

ADD_POS_MODE_2:
    // check in position
    IF boDriveX_InPosition THEN
        // here we should be at 42°+142°-42°=142°
        OpMode := VEL_MODE; // set breakpoint here to check
    END_IF

VEL_MODE:
    // start velocity control with 42 rpm
    diDriveX_VelocityCmdValue:=42*WEIGHTING_FACTOR_VEL; // 42 rpm * 10000
    wDriveX_ControlWord := CTRL_AF_SECONDARY_OP_MODE; // AF and secondary operation mode 1
    OpMode := VEL_MODE_1;

VEL_MODE_1:
    // check acknowledge power
    IF (wDriveX_StatusWord AND MASK_STATUS_AF_OP_MODE) = STATUS_AF_SECONDARY_OP_MODE THEN
        fbTON(IN := TRUE, PT:= T#5S);
        // 5 seconds in velocity control
        IF boDriveX_InVelocity AND fbTON.Q=TRUE THEN
            // here we should rotate with 42rpm
            OpMode := MODE_DRV_AH; // set breakpoint here to check
            fbTON(IN := FALSE);
        END_IF
    END_IF

MODE_DRV_AH:
    // stop the drive until StandStill is reached
    wDriveX_ControlWord := CTRL_WORD_AH; // AH
    IF boDriveX_StandStill THEN
        OpMode := MODE_DRV_OFF; // set breakpoint here to check
    END_IF

```

```

END_IF

END_CASE

//actual values
rActPosition := DINT_TO_REAL(diDriveX_ActualPosition)/DINT_TO_REAL(WEIGHTING_FACTOR_POS);
rActVelocity := DINT_TO_REAL(diDriveX_ActualVelocity)/DINT_TO_REAL(WEIGHTING_FACTOR_VEL);

boError:= wDriveX_StatusWord.13;
boWarning:=wDriveX_StatusWord.12;
// dwDriveX_DiagnosisNumber; // diagnosis number
// Clear error: write S-0-0099 to b0011 with IL_SIIISvcWrite()

```

7.6 Versorgungsgerät als SPS-Gerät

7.6.1 Allgemeines

Mit den Versorgungsgeräten des Typs HMV05¹⁾ (Produktfamilie IndraDrive ML) und KMV03 (Produktfamilie IndraDrive Mi) stehen Versorgungsgeräte mit separater Führungskommunikation zur Verfügung. Diese Versorgungsgeräte können mit der Firmware PSB-2x betrieben werden.

Versorgungsgeräte dieses Typs können als "intelligente" Versorgungsgeräte bezeichnet werden; sie verfügen über eine Busadresse, verschiedene Parameter, um Funktionen zu parametrieren, ein Gerätesteuerswort und ein Gerätestatuswort.



Informationen zu den "intelligenten" Versorgungsgeräten finden Sie in der jeweiligen Projektierungsbeschreibung:

- KMV03: Projektierungsbeschreibung mit der Materialnummer R911335702
- HMV05: Projektierungsbeschreibung mit der Materialnummer R911344278

Versorgungsgeräte mit separater Führungskommunikation werden von IndraLogic XLC / IndraMotion MLC / IndraMotion MLD als SPS-Geräte unterstützt.

7.6.2 Parametrierung

Im IndraWorks-Projekt kann das Versorgungsgerät angelegt werden, indem es aus dem Bibliotheksfenster (Ordner "Peripherie", Ordner für den jeweiligen Feldbus) auf den Feldbusknoten gezogen wird. Für Sercos ist das jeweilige Versorgungsgerät in der Bibliothek der Peripheriegeräte "Sercos" zu finden.

Eine dialoggeführte Parametrierung des Versorgungsgeräts kann über IndraWorks Ds erfolgen. (Eine Parametrierung über Einzelparameter ist über IndraLogic XLC / IndraMotion MLC möglich.)



IndraLogic XLC / IndraMotion MLC: Mit rechter Maustaste auf das Sercos Gerät und den Eintrag "IndraWorks Ds hier..." kann IndraWorks Ds direkt aufgerufen werden.

Alternativ zu "IndraWorks Ds hier..." kann die Kommunikation zwischen PC und Versorgungsgerät direkt erfolgen. Hierzu kann der PC an einen freien

¹⁾ HMV05 ist die Bezeichnung für ein Versorgungsgerät bestehend aus Netzanschaltmodul HNA05, Universal-Wechselrichter HMU05 und Steuerteil CSB02.5 mit Firmware für Versorgungsgeräte (FWA-INDRV*-PSB-...). HMV05 ist keine offizielle Produktbezeichnung und wird ausschließlich für Dokumentationszwecke verwendet.

Port des Sercos Geräts angeschlossen werden oder über einen "Sercos III NetSwitch" (Bestellnr.: R911328254, HAWA HGS NS-S3-1NRT) kommunizieren. Alternativ kann mit dem Windows-Befehl "route" die IP-Kommunikation über die Steuerung geroutet werden.

Die Konfiguration der zyklisch zu übertragenden Parameter kann in IndraWorks unterhalb des Sercos Knotens erfolgen.

Verbindungsaufbau und Parametrierung des Versorgungsgeräts

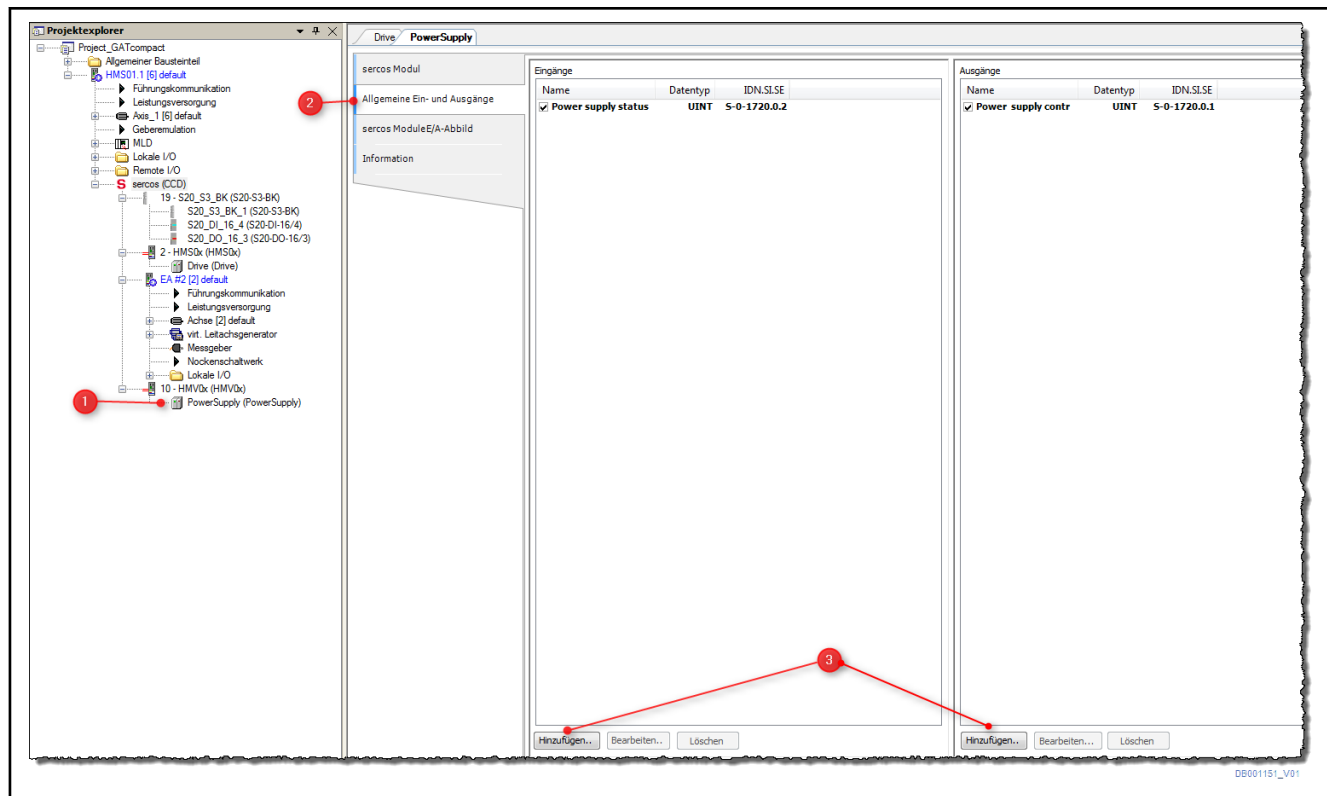
Bei Nutzung von "IndraWorks Ds hier..." bei IndraLogic XLC / IndraMotion MLC (rechte Maustaste auf das Sercos Gerät) entfallen Arbeitsschritt 1 und 2.

1. Versorgungsgerät an einen PC mit freiem Ethernet-Netzwerkanschluss bzw. über einen USB-Anschluss mit Ethernetadapter anschließen. IP-Adresse des Ethernet-Netzwerkanschlusses bzw. des Ethernetadapters auf den Bereich des Versorgungsgeräts einstellen 172.31.x.x.
2. IndraWorks Ds starten. **Verbindungsauswahl ► IP-Adress-Suche**. Verbinden mit dem Gerät.
3. In IndraWorks Ds ggf. eine weitere Parametrierung des Versorgungsgeräts ausführen.

7.6.3 Konfiguration der zyklischen Daten

Die zyklisch zu übertragenden Daten können in IndraWorks unterhalb des Sercos Knotens im Dialog "Allgemeine Ein- und Ausgänge" konfiguriert werden.

Inbetriebnahme- und Anwendungshinweise



- 1 Doppelklick auf den PowerSupply-Knoten unterhalb des Sercos Geräts
- 2 Reiter "Allgemeine Ein- und Ausgänge" anwählen
- 3 Durch "Hinzufügen" können Parameter angewählt werden bzw. gelöscht werden

Abb. 7-45: Beispielkonfiguration der zyklischen Parameter im Antriebstelegramm AT (Eingänge) und Masterdatentelegramm MDT (Ausgänge)

Die gezeigte Beispielkonfiguration ist die Minimalkonfiguration. "S-0-1720.0.1, Versorger-Steuerwort" als SPS-Ausgang wird benötigt, um das Versorgungsgerät zu kommandieren und "S-0-1720.0.2, Versorger-Statuswort" als SPS-Eingang, um den entsprechenden Status auszulesen.



Wenn die Gerätebeschreibungsdatei die notwendigen Informationen enthält (z. B. welche Parameter zyklisch konfigurierbar sind, Name und Attribut der Parameter), bietet der Konfigurator eine Auswahlliste an und schlägt die notwendigen Einstellungen vor. Sind die Informationen nicht verfügbar, wird keine Auswahlliste angeboten und die Einstellungen sind manuell vorzunehmen.

Konfiguration der zyklischen Daten

1. Im IndraWorks Projekt Doppelklick auf den Knoten des Versorgungsgeräts. Reiter "Allgemeine Ein- und Ausgänge" wählen.
2. "S-0-1720.0.2, Versorger-Statuswort" muss konfiguriert werden.

In der Hilfe zu S-0-1720.0.2 das Attribut prüfen:

- Länge: 2 Byte
- Format: BIN
- Bei den Eingängen "Hinzufügen" wählen

3. Parameter "S-0-1720.0.1, Versorger-Steuerwort" muss konfiguriert werden.

In der Hilfe zu S-0-1720.0.1 das Attribut prüfen:

- Länge: 2 Byte
- Format: BIN
- Bei den Ausgängen "**Hinzufügen**" wählen

4. Zuordnung Variable ↔ Parameter

Die Zuordnung der Parameter zu SPS-Variablen kann im Reiter "E/A-Abbild" erfolgen. Es können neue Variablen erzeugt werden oder die Ein- und Ausgangsparameter auf bestehende SPS-Variablen gemappt werden.

Wechseln zum Reiter "E/A-Abbild". Tragen Sie die SPS-Variablen in der Spalte "Variable", wie in [Beispielkonfiguration der zyklischen Parameter im Antriebstelegramm AT \(Eingänge\) und Masterdatentelegramm MDT \(Ausgänge\)](#) gezeigt, ein.

Der eingegebene Name wird später in [Zuordnung der SPS-Variablen zu den konfigurierten Parametern](#) in der Spalte "Kanal" angezeigt.

Abhängig vom Parameter-Attribut, muss in der SPS der richtige Datentyp (mit passender Länge; mit oder ohne Vorzeichen) gewählt werden:

Datenlänge im Attribut	Format im Attribut	Datentyp in der SPS
2 Byte	BIN oder HEX	WORD
4 Byte	BIN oder HEX	DWORD
2 Byte	DEC_MV	INT
4 Byte	DEC_MV	DINT
2 Byte	DEC_OV	UINT
4 Byte	DEC_OV	UDINT

Tab. 7-4: Zuordnung Parameter-Attribut ↔ Datentyp in der SPS

In diesem Beispiel werden die Variablen neu erzeugt:

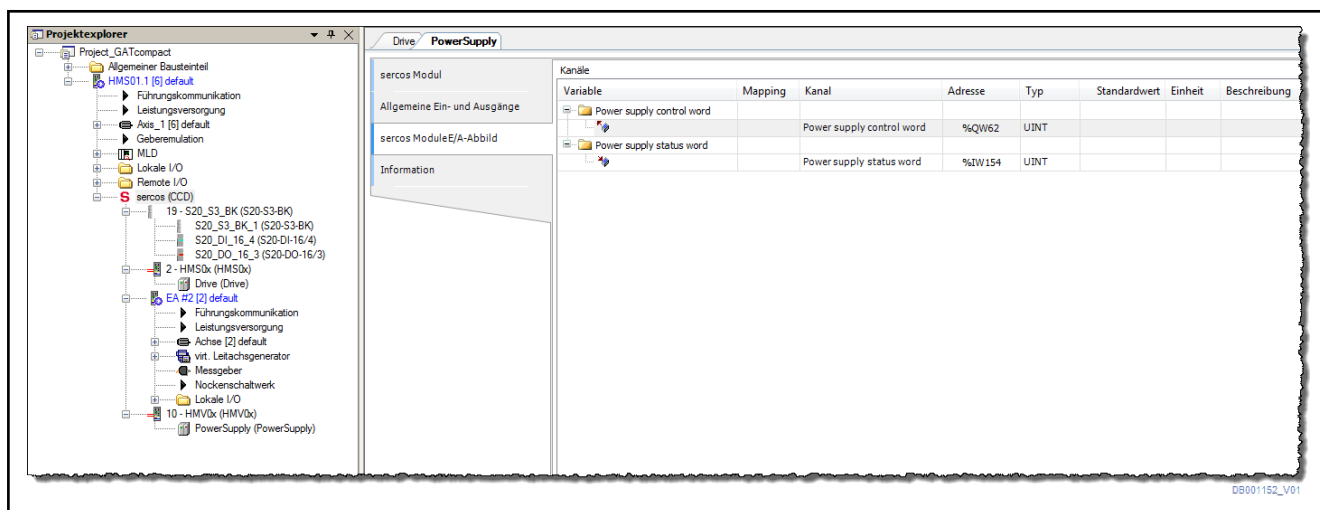


Abb. 7-46: Zuordnung der SPS-Variablen zu den konfigurierten Parametern

7.6.4 SPS-Programmierung

Die konfigurierten SPS-Variablen der zyklisch konfigurierten Parameter für Steuer- und Statuswort können im SPS-Programm verwendet werden.



Alternativ zu einem SPS-Programm, können die Versorgungsgeräte über Funktionen gesteuert werden, die in einer IndraWorks-Bibliothek zur Verfügung gestellt werden:

- **Library:** RMB_SercosIII_Util
- **Group:** PowerSupply
- **Target Systems:**
 IndraMotion MTX 14VRS
 IndraMotion MLC 14VRS, Function package PLC
 IndraMotion MLD with MPx20

Die Bitbelegung der Versorgungsgeräte-Parameter (S-0-1720.0.1, Versorger-Steuerwort / S-0-1720.0.2, Versorger-Statuswort) entnehmen Sie bitte der Parameterbeschreibung.

Die Kommandierung des Versorgungsgerätes kann in einer beliebigen zyklischen Task erfolgen (z. B. "PlcTask" und "PlcProg").

Programmbeispiel

Im folgenden Beispiel wurde die Leistungszuschaltung und -abschaltung umgesetzt:

- **1. Zustand "Hauptschütz einschalten"**
 Steuerwort: Bit 12 setzen (16#1000)
 Statuswort: Bits 4, 5, 7 abfragen (16#00B0), dann Zustand weiterschalten
- **2. Zustand "Netzschütz einschalten (Gleichspannung laden)"**
 Steuerwort: Bits 12, 14 (16#5000) setzen
 Statuswort: Bits 4, 5, 6, 7 abfragen (16#40F0), dann Zustand weiterschalten
- **3. Zustand "Sollbetriebsart aktivieren"**
 Steuerwort: Bits 12, 14, 15 setzen; Bit 8, Bit 9 sind nicht gesetzt für Auswahl der Hauptbetriebsart (16#D000)
 Statuswort: Bits 4, 5, 6, 7, 14, 15 abfragen (16#C0F0)
 Angeschlossene Achsen können kommandiert werden
- **4. Zustand "Leistung ausschalten"**
 Steuerwort: alle Bits löschen (16#0000)

SPS-Programmierung

1. Im IndraWorks-Projekt das Programm "PlcProg" öffnen.
2. Das Programmlisting "**Deklaration**" in den Deklarationsteil kopieren.
3. Das Programmlisting "**Implementation**" in den Implementationsteil kopieren. (Seitenumbruch nicht mit kopieren bzw. nach dem Kopieren und Einfügen wieder löschen.)
4. Übersetzen, Laden und Starten des SPS-Programmes.
5. Setzen der Leistungsanforderung "bSetPower".

6. Überprüfen, ob Leistungszuschaltung erfolgt ist (Statuswort: Bits 4, 5, 6, 7, 14, 15).

⇒ Aus diesem Programm können für die individuelle Lösung einer Aufgabe Teile kopiert und weiterverwendet werden.

Deklaration

```

PROGRAM PlcProg
VAR
    iStatePowerSupply: INT;          (* state for power supply *)
    bSetPower: BOOL;                 (* set power *)
END_VAR

VAR CONSTANT
    STATE_ENABLE_MAINS_CONTACTOR : INT := 1;      (* state enable mains contactor *)
    STATE_ENABLE_POWER_SUPPLY     : INT := 2;      (* state enable power supply, load dc voltage *)
    STATE_ACTIVATE_POWER_SUPPLY   : INT := 3;      (* state activate power supply *)
    STATE_POWER_OFF               : INT := 4;      (* state power off *)
END_VAR

(* S-0-1720.0.1 Power supply control *)
(* Bit 15: power supply activation *)
(* Bit 14: power supply enable *)
(* Bit 13: reserved *)
(* Bit 12: mains contactor *)
(* Bit 11: reserved *)
(* Bit 10: DC discharging *)
(* Bit 09: selection of OpMode *)
(* Bit 08: selection of OpMode *)
(* Bit 07: reserved *)
(* Bit 06: reserved *)
(* Bit 05: reserved *)
(* Bit 04: reserved *)
(* Bit 03: reserved *)
(* Bit 02: reserved *)
(* Bit 01: reserved *)

(* S-0-01720.0.2 Power supply status *)
(* Bit 15: voltage boost ON/OFF *)
(* Bit 14: DC bus okay *)
(* Bit 13: error *)
(* Bit 12: warning *)
(* Bit 11: brake resistor *)
(* Bit 10: DC discharging *)
(* Bit 09: selection of OpMode *)
(* Bit 08: selection of OpMode *)
(* Bit 07: feedback signal of mains contactor *)
(* Bit 06: mains barrage *)
(* Bit 05: hardware enable *)
(* Bit 04: mains status *)
(* Bit 03: reserved *)
(* Bit 02: reserved *)
(* Bit 01: reserved *)

```

Implementation

```

(* set power request and check status *)
IF bSetPower AND wStatusWord = 16#00A0 OR wStatusWord = 16#0020 THEN
    iStatePowerSupply := STATE_ENABLE_MAINS_CONTACTOR;
ELSIF bSetPower AND wStatusWord = 16#00B0 THEN
    iStatePowerSupply := STATE_ENABLE_POWER_SUPPLY;
ELSIF bSetPower AND wStatusWord = 16#40F0 THEN
    iStatePowerSupply := STATE_ACTIVATE_POWER_SUPPLY;
ELSIF bSetPower AND wStatusWord = 16#C0F0 THEN
    iStatePowerSupply := STATE_POWER_OFF;
END_IF

(* state machine for power supply *)
CASE iStatePowerSupply OF

    STATE_ENABLE_MAINS_CONTACTOR: (* enable mains contactor *)

        wControlWord := 16#1000;    (* set bit 12 *)

        IF wStatusWord = 16#00B0 THEN (* check bit 4, 5, 7 *)
            iStatePowerSupply := STATE_ENABLE_POWER_SUPPLY;
        ELSE
            iStatePowerSupply := STATE_ENABLE_MAINS_CONTACTOR;
        END_IF

    STATE_ENABLE_POWER_SUPPLY: (* enable power supply, load DC voltage *)

        wControlWord := 16#5000;    (* set bit 12, 14 *)

        IF wStatusWord = 16#40F0 THEN (* check bit 4, 5, 6, 7 *)
            iStatePowerSupply := STATE_ACTIVATE_POWER_SUPPLY;
        ELSE
            iStatePowerSupply := STATE_ENABLE_POWER_SUPPLY;
        END_IF

    STATE_ACTIVATE_POWER_SUPPLY: (* enable power supply, operation mode constant DC voltage *)

        wControlWord := 16#D000;    (* set bit 12, 14, 15 *)

        IF wStatusWord = 16#C0F0 THEN (* check bit 4, 5, 6, 7, 14, 15 *)
            iStatePowerSupply := STATE_POWER_OFF;
        ELSE

```

```

    iStatePowerSupply := STATE_ACTIVATE_POWER_SUPPLY;
END_IF

STATE_POWER_OFF: (* power off *)

    IF NOT bSetPower THEN
        wControlWord := 16#0000;    (* reset bits *)
    END_IF

    IF NOT wStatuswort = 16#0040 THEN    (* check bit 6 *)
        iStatePowerSupply := STATE_POWER_OFF;
    END_IF

END_CASE

```

7.7 Vergleich der Antriebsfamilie "IndraDrive Cs" mit der Antriebsfamilie "IndraDrive C"/"IndraDrive M"

7.7.1 Einleitung

Die antriebsinterne SPS (IndraMotion MLD) ist bei der Antriebsfamilie "IndraDrive Cs" gegenüber der Antriebsfamilie "IndraDrive C"/"IndraDrive M" mit einigen funktionalen und hardwaretechnischen Änderungen verbunden. Dies hat unter anderem Auswirkungen auf die Applikation von SPS-Programmen. Welche Änderungen im Einzelnen betroffen sind, und welche Punkte bei der Portierung von SPS-Programmen zu beachten sind, wird in den folgenden Kapiteln vorgestellt.

7.7.2 Physikalische Schnittstellen

- **Ein- und Ausgänge**

Bei "IndraDrive" steht ein analoger **Eingang** zur Verfügung. Ein analoger **Ausgang** existiert nicht.

"IndraDrive" verfügt über digitale Eingänge und konfigurierbare digitale Ein-/Ausgänge.

Mit IO-Modulen der Produktfamilie "Rexroth Inline" besteht die Möglichkeit, weitere analoge und digitale Ein-/Ausgänge anzubinden.

- **Schnittstelle für Engineering**

"IndraDrive Cs" verfügt über eine Ethernet-Schnittstelle. Programmierung und Download erfolgt über diese Schnittstelle.

- **Speicherkarte**

Das Standard-Bedienteil besitzt keine Möglichkeit, um eine Speicherkarte zu verwenden.

Das Advanced-Bedienteil besitzt einen Steckplatz für eine microSD-Speicherkarte.

7.7.3 Byte-Reihenfolge

"Big Endian" / "Little Endian"

Bei der Firmware bis MPx08 wurde die Byte-Reihenfolge "Little Endian" verwendet; ab MPx17 wird die Byte-Reihenfolge "Big Endian" verwendet.

I/O-Kanal (PAE, PAA)

Einleitung

Der I/O-Kanal stellt den Kontakt der IndraMotion MLD zur Außenwelt dar, indem digitale und analoge Ein-/Ausgänge ausgewertet bzw. angesprochen werden können. Um den I/O-Kanal im SPS-Anwenderprogramm nutzen zu können, müssen Variablen auf bestimmte Speicherzellen (Prozessabbild) zu-

gewiesen werden. ["PAE" (Prozessabbild der Eingänge) ab Parameter P-0-1390ff, "PAA" (Prozessabbild der Ausgänge) ab Parameter P-0-1410ff.]

Die Zuweisung von Variablen auf bestimmte Speicherzellen erfolgt mittels Prozentzeichen "%" und einer Zeichenreihe, die festlegt, ob ein Ein- oder Ausgang angesprochen werden soll und welcher Datentyp verwendet wird.

%QX7.5 und %Q7.5: Ausgangsbit 7.5
 %IW215: Eingangswort 215
 %QB7: Ausgangsbyte 7
 %MD48: Doppelwort an der Speicherstelle 48 im Merker

Tab. 7-5: Beispiele

Der folgende Programmausschnitt zeigt ein Beispiel für die Deklaration von Variablen mit unterschiedlichen Datentypen, die dem Prozessabbild zugewiesen werden.

Programm:

```
PROGRAM PLC_PRG
VAR
  Dig_IN_DWORD AT %ID0: DWORD;
  Dig_IN_WORD AT %IW2: WORD;
  Dig_IN_BYTE AT %IB5: BYTE;

  Dig_OUT_DWORD AT %QD2: DWORD;
  Dig_OUT_WORD AT %QW0: WORD;
  Dig_OUT_BYTE AT %QB7: BYTE;
END_VAR
```

Parameter/Adressierungs-Relationstabellen

Die folgenden Tabellen zeigen die Relation zwischen den Parametern für das Prozessabbild der Ein-/Ausgänge und den entsprechenden Adressierungstypen. In den Tabellen sind lediglich drei Parameter aufgeführt, um das Prinzip zu erklären. Die dargestellte Aufteilung gilt für alle Parameter der Eingänge (P-0-1390 bis P-0-1409) und Ausgänge (P-0-1410 bis P-0-1429).

P-0-1390		P-0-1391		P-0-1392		...
IB0 Bit15...Bit8	IB1 Bit7...Bit0	IB2 Bit15...Bit8	IB3 Bit7...Bit0	IB4 Bit15...Bit8	IB5 Bit7...Bit0	...
IW0		IW1		IW2		...
ID0				...		

Tab. 7-6: Prozessabbild Eingänge (Big Endian: ab MPx17)

P-0-1410		P-0-1411		P-0-1412		...
QB0 Bit15...Bit8	QB1 Bit7...Bit0	QB2 Bit15...Bit8	QB3 Bit7...Bit0	QB4 Bit15...Bit8	QB5 Bit7...Bit0	...
QW0		QW1		QW2		...
QD0				...		

Tab. 7-7: Prozessabbild Ausgänge (Big Endian: ab MPx17)

Netzwerkkommunikation

Wenn mit der MLD eine Netzwerk-Kommunikation zu anderen Geräten hergestellt werden soll (z. B. über TCP/IP), so sollten stets die Konvertierungsfunktionen für die jeweils verwendete Byte-Reihenfolge genutzt werden, damit die Anwenderprogramme firmwareübergreifend lauffähig sind.

Hierfür stellen in der Regel alle Kommunikations-Bibliotheken Funktionen zur Verfügung, die diese Konvertierung durchführen. Bei der Bibliothek "SysLibSockets" sind z. B. folgende Konvertierungs-Funktionen verfügbar:

- SysSockHtons (Host to network): Steuerung zu Netzwerk, 16 Bit
- SysSockNtohs (Network to host): Netzwerk zu Steuerung, 16 Bit
- SysSockHtonl (Host to network): Steuerung zu Netzwerk, 32 Bit
- SysSockNtohl (Network to host): Netzwerk zu Steuerung, 32 Bit

Die folgenden Datentypen müssen zur Übertragung über ein Netzwerk konvertiert werden:

Datentyp	Steuerung zu Netzwerk	Netzwerk zu Steuerung
WORD, INT, UINT	SysSockHtons()	SysSockNtohs()
DWORD, DINT, UDINT, TIME	SysSockHtonl()	SysSockNtohl()

Die Datentypen BYTE, CHAR, STRING müssen nicht konvertiert werden. Bei den Datentypen REAL und LREAL ist die Byte-Reihenfolge nicht vorgeschrieben. Hier kann die richtige Einstellung nur experimentell ermittelt werden.

Dateizugriffe bei der MLD

Dateizugriffe sind nur in Verbindung mit einem ADVANCED-Bedienteil möglich. Bei Dateizugriffen mit Hilfe der Bibliothek "MX_SysLibFile" ist ebenfalls die Byte-Reihenfolge zu beachten. In der unten zu sehenden Abbildung werden über die Variable "ardiBuffer" (32-Bit-Werte) ASCII-Zeichen in eine Textdatei ("Test.txt") geschrieben. Auf Systemen mit unterschiedlicher Byte-Reihenfolge werden die Zeichen unterschiedlich abgelegt. Speziell bei der Portierung von Anwenderprogrammen zwischen Systemen mit unterschiedlicher Byte-Reihenfolge muss geprüft werden, ob die Daten nach der Portierung noch konsistent sind. Sollte dies nicht der Fall sein, muss entweder die Reihenfolge händisch getauscht werden, oder auf 8-Bit-Werte umgestellt werden.

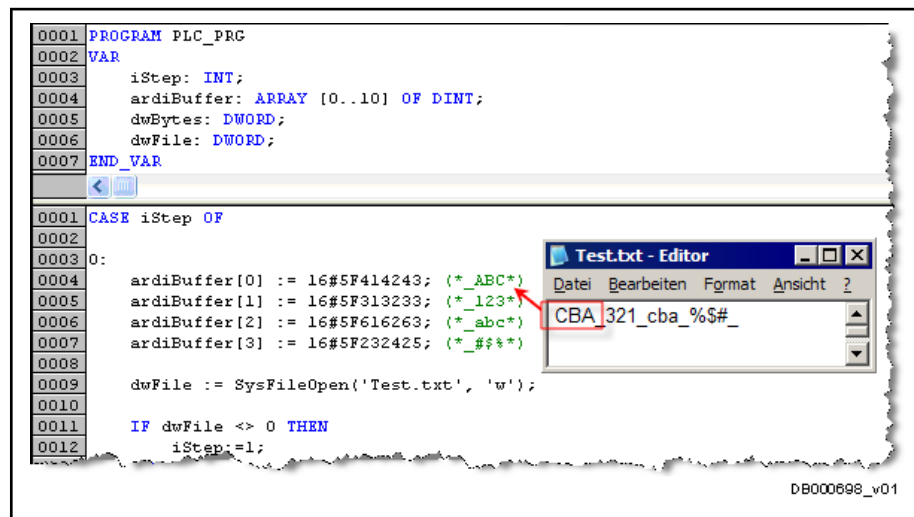


Abb. 7-47: Überprüfung der Byte-Reihenfolge

Zwei Möglichkeiten das SPS-Programm anzupassen:

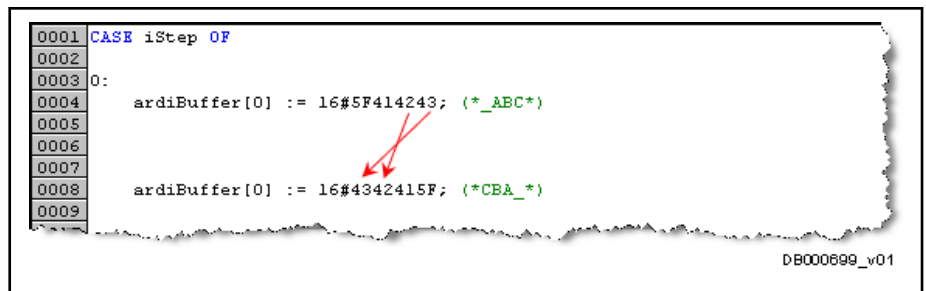


Abb. 7-48: Byte-Reihenfolge tauschen

- oder -

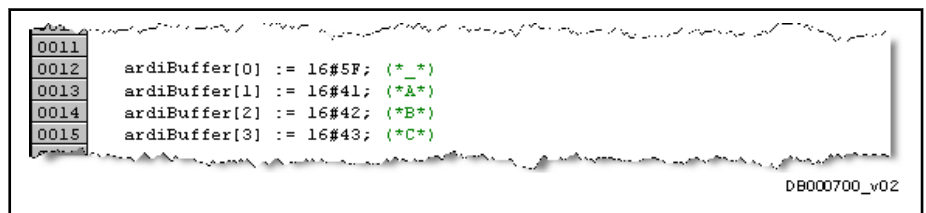


Abb. 7-49: Byteweise schreiben

Adresszugriffe mit unterschiedlichen Datentypen

Zugriff über Merker

Über die Merker "Merker_LByte" und "Merker_HByte" werden die Bytes Nr. 200 und 201 beschrieben. Der Merker "Merker_WORD" mit der Adresse "%MW100" ($100 * 2\text{Byte} = 200$) soll einen 16-Bit-Wert auslesen. "Merker_WORD" hat in diesem Beispiel die gleiche Adresse wie "Merker_LByte".

Bei Systemen mit unterschiedlicher Byte-Reihenfolge werden über "Merker_WORD" verschiedene Werte ausgelesen.

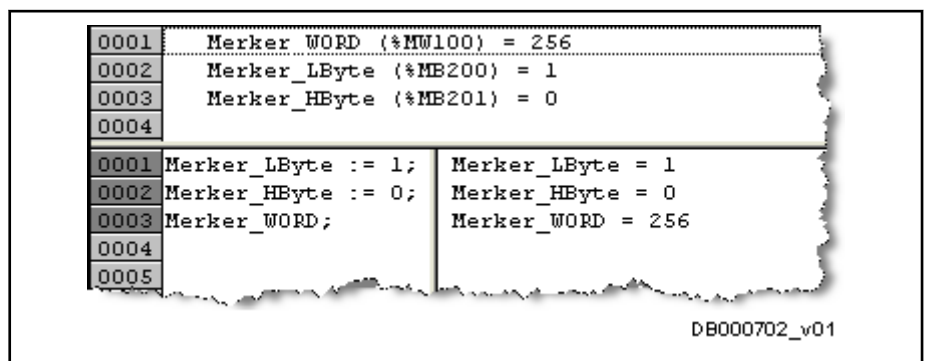


Abb. 7-50: Wert bei Big Endian (ab MPx17): "Merker_WORD = 256"

Um die Byte-Reihenfolge im SPS-Programm zu kompensieren, muss lediglich die Adressierung der Byte-Merker in der Variablen-Deklaration angepasst werden.

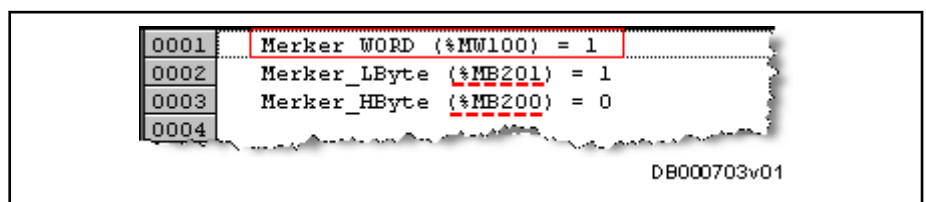


Abb. 7-51: Adressierung der Byte-Merker in der Variablen-Deklaration anpassen

Zugriff über Pointer

In einem Test-Programm wird das Feld "arWORD" vom Typ "WORD" mit Werten belegt. Mit einem Pointer, der mit der Startadresse des Feldes initialisiert wird, soll ein Wert vom Typ "DWORD" ausgelesen werden. Dieser Wert wird in einer IF-Abfrage geprüft und je nach Ergebnis eine Variable "bDummy" gesetzt. In der ersten Abbildung ist die IF-Abfrage erfolgreich und die Variable "bDummy" wird auf TRUE gesetzt.

0001	arWord[0]:=3;	arWord[0] = 3
0002	arWord[1]:=1002;	arWord[1] = 1002
0003		
0004	ptDWORD:=ADR(arWord);	ptDWORD = <0cd189a0>
0005	ptDWORD^;	ptDWORD^ = 65667075
0006		
0007	IF {ptDWORD^ = 65667075} THEN	ptDWORD^ = 65667075
0008	bDummy := TRUE;	bDummy = TRUE
0009	ELSE	
0010	bDummy := FALSE;	bDummy = TRUE
0011	END_IF	
0012		

DB000704_v01

Abb. 7-52: Wert bei Little Endian (bis MPx08)

Mit der Portierung des Beispiel-Programms auf die Antriebsfamilie "IndraDrive Cs", erhält man über den Pointer wegen der "Big Endian"-Umstellung einen anderen Wert, wodurch die IF-Abfrage inkonsistent wird. In diesem Beispiel müsste die IF-Abfrage angepasst werden.

0001	arWord[0]:=3;	arWord[0] = 3
0002	arWord[1]:=1002;	arWord[1] = 1002
0003		
0004	ptDWORD:=ADR(arWord);	ptDWORD = <0c418838>
0005	ptDWORD^;	ptDWORD^ = 197610
0006		
0007	IF {ptDWORD^ = 65667075} THEN	ptDWORD^ = 197610
0008	bDummy := TRUE;	bDummy = FALSE
0009	ELSE	
0010	bDummy := FALSE;	bDummy = FALSE
0011	END_IF	
0012		

DB000705_v01

Abb. 7-53: Wert bei Big Endian (ab MPx17)



Bei der Verwendung von Funktionsbausteinen zum Schreiben von Listen-Parametern, z. B. "MB_WriteListParameter", sollte das Feld mit den zu schreibenden Werten stets die gleiche Datenlänge besitzen, wie der Listen-Parameter. Ansonsten treten die oben beschriebenen Probleme auf.

7.7.4 MLD-Leistungsvergleich

Es sollte eine Leistungsvergleichs-Messungen bezüglich der MLD durchgeführt werden.

Für die Leistungsvergleich-Messung sollte der Funktionsbaustein "MX_IECTaskGetLoad" der "MX_Base" verwendet werden. Der Funktionsbaustein "MX_IECTaskGetLoad" dient dazu, die erweiterte Laufzeitmessung zu aktivieren. Er zeigt Informationen über die Taskauslastung an.

Die Taskauslastung ist das prozentuale Verhältnis von der genutzten Rechenzeit (t_{genutzt}) für das SPS-Programm zur maximal verfügbaren Rechenzeit in einem Taskzyklus (t_{Zyklus}):

$$\text{Taskauslastung} = (t_{\text{genutzt}} / t_{\text{Zyklus}}) [\%]$$

Es ist empfehlenswert, die Taskauslastung der Anwenderprogramme zu ermitteln. Ist die Auslastung recht hoch (über 85 %), so muss eventuell die Taskzykluszeit angepasst oder ein "IndraDrive"-Antriebsregelgerät mit Advanced-Performance eingesetzt werden.

8 Programmierhinweise

8.1 Industrie-Standards für die Programmierung

8.1.1 Allgemeines

Um SPS-Programme möglichst einfach und schnell von einem Zielsystem auf das andere übertragen zu können, wurden Industrie-Standards für die Programmierung eingeführt.



Alle von Bosch Rexroth zur Verfügung gestellten Bausteine werden gemäß der Standards erstellt.

8.1.2 IEC 61131

Durch die Definition des Industrie-Standards für die Programmierung in der Automatisierung in Form der **IEC 61131-3** sind die Grundlagen für eine kombinierte Logik- und Bewegungssteuerung gelegt. Im Wesentlichen sind dies:

- Definition der Ablaufsprache **SFC (Sequential Function Chart)**, als leistungsfähiges Programmierwerkzeug für datenflussorientierte Programme wie z. B. Regler
- Definition der pascalähnlichen Hochsprache **ST (Structured Text)**, als leistungsfähiges Instrument - auch zur Programmierung von komplexen mathematischen Funktionen
- Definition von Basis-Datentypen von **BOOL** bis **DINT** (32 Bit Integer) oder **REAL** (32 Bit Float) sowie Felder und Strukturen
- Definition instanzierbarer Funktionsbausteine
- Definition von **Standard-Funktionsbausteinen** (Zähler, Zeitfunktionen, Flankenerkennung, ..) sowie **Standard-Funktionen** (mathematische Funktionen, String-Verarbeitung, Vergleiche, ..)

Durch die Einhaltung der IEC 61131 wird die zukünftige Kompatibilität der Systeme unterschiedlicher Anbieter gewährleistet; normgerechte Programme sind an allen Automatisierungsgeräten lauffähig.

Die in der Norm festgelegten Programmiersprachen führen zum besseren Austausch von Steuerungsprogrammen unterschiedlicher Hersteller. Demnach können Steuerungsprogramme in folgenden Sprachen geschrieben werden:

- Anweisungsliste [Instruction List (IL)]
- Kontaktplan [Ladder Diagram (LD)]
- Funktionsbausteinen [Function Block Diagram (FBD)]
- Strukturierter Text [Structured Text (ST)] und
- Ablaufsteuerung [Sequential Function Chart (SFC)]

Die Norm enthält unter anderem Begriffsbestimmungen und Funktionsmerkmale für die SPS und die Verbindungen zu anderen relevanten Standards der ISO/IEC. Sie hat den Status "Internationaler Standard" (IS).

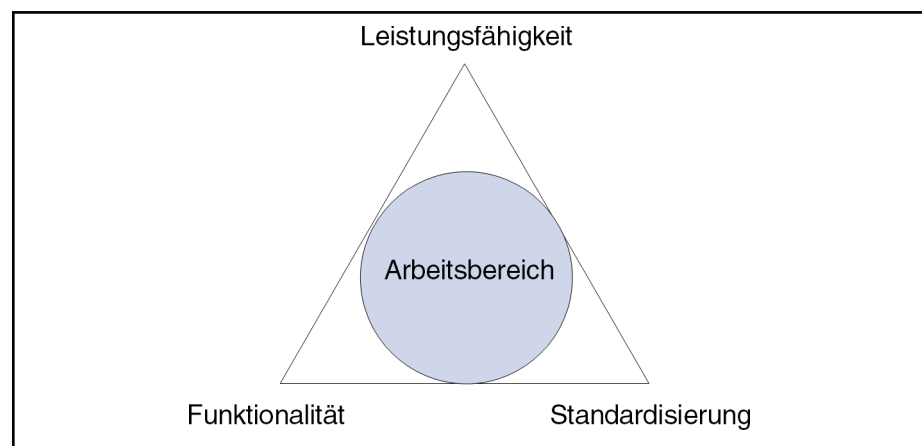
8.1.3 PLCopen

Auf dem Motion-Control-Markt existiert eine breite Vielfalt von inkompatiblen Systemen und Lösungen. In Unternehmen, in denen verschiedene Systeme verwendet werden, verursacht diese Inkompatibilität erhebliche Kosten für die Endnutzer, die Einarbeitung ist verwirrend, die Entwicklung gestaltet sich schwierig und das Marktwachstum verlangsamt sich.

Eine Standardisierung würde diesen Negativfaktoren zweifellos entgegenwirken. Standardisierung beinhaltet nicht nur die Programmiersprachen selbst (wie es in dem weltweiten Standard IEC 61131-3 der Fall ist), sondern auch eine Standardisierung der Schnittstelle zu verschiedenen Motion-Control-Lösungen. Somit ist die Programmierung dieser Motion-Control-Lösungen weniger hardwareabhängig. Dies erhöht die Wiederverwendbarkeit der Anwendungs-Software und reduziert den Aufwand für die Schulung und die Projektierung.

Die PLCopen hat folgende Zielsetzungen:

- **Leistungsfähigkeit:** Anwender schreiben ihre Programme mit bestimmten Funktionen sehr nahe an der Hardware, um die größtmögliche Leistungsfähigkeit, wie von ihrer Umgebung vorgeschrieben, zu erzielen. Der Anwender ist deshalb in seinen Möglichkeiten bezüglich der Zielhardware und der Wiederverwendbarkeit der Steuerungssoftware beschränkt und die Schulungskosten steigen.
- **Funktionalität:** Die zweite Anwenderoption berücksichtigt einen sehr breiten Spielraum an verfügbarer Software-Funktionalität. Das kann für den Anwender sehr hilfreich sein, führt jedoch selten zu einer hohen Leistungsfähigkeit. Außerdem entstehen hohe Schulungskosten.
- **Standardisierung:** Der dritte Eckpfeiler, die Standardisierung, konzentriert sich vor allem auf die Wiederverwendbarkeit über verschiedene Systeme unterschiedlicher Hersteller, einschließlich integrierter, dezentraler und vernetzter Systeme, und auch auf die Reduzierung der Schulungskosten. Aufgrund des allgemeinen Charakters dieser Definition kann die Leistungsfähigkeit auf verschiedenen Architekturen weniger optimal sein als feste Kodierung. Deshalb sollte man von der Standardisierung nicht erwarten, dass sie maximale Leistungsfähigkeit bietet; sie kann vielmehr der maximalen Funktionalität sehr nahe kommen, d. h. die Grundlinie des Dreiecks ist sehr kurz.



Grafikname DF000146

Abb. 8-1: Zielsetzung der PLCopen

Verstärkt wird der Trend, SPS-Programme möglichst einfach und schnell von einem Zielsystem auf das andere übertragen zu können, durch die offizielle Freigabe der Spezifikation der Motion Task Force der PLCopen. Darin sind Funktionen und Funktionsbausteine zur Einzelachspositionierung, sowie für elektronische Getriebefunktionen definiert. Außerdem wurde eine grundsätzliche Verfahrensweise zur Implementierung von bewegungsauslösenden Funktionsbausteinen festgelegt.

8.2 Bibliotheken für Rexroth IndraMotion MLD

8.2.1 Allgemeine Eigenschaften der Bibliotheken



Im Folgenden wird unter anderem ein Überblick zu den Bibliotheken für Rexroth IndraMotion MLD gegeben. Details entnehmen Sie bitte der Bibliotheksbeschreibung ("Rexroth IndraMotion, MLD Bibliotheken ab MPx-18").

Die in den Bibliotheken für Rexroth IndraMotion MLD enthaltenen Bausteine sind an die PLCopen-Spezifikation angelehnt und verhalten sich prinzipiell alle gleich.

Weitere Bausteine, die benötigt, aber nicht in PLCopen spezifiziert sind, haben die gleiche Syntax und Semantik und grundsätzliche Eigenschaften:

- Die Bausteine werden in der Regel zyklisch aufgerufen und liefern an ihren Ausgängen Informationen, wie z. B. "Done", "Error", "CommandAborted".
- Mit einer steigenden Flanke an dem "Execute"-Eingang werden die Vorgänge i. d. R. gestartet.
- Nach Beendigung oder Abbruch wird ein entsprechender Ausgang ("Done", "Error", "CommandAborted") gesetzt. Dieser Ausgang bleibt solange auf TRUE, bis "Execute" zurückgenommen wird. Ist "Execute" bereits FALSE, so bleibt der jeweilige Ausgang für einen Zyklus lang (Aufruf) auf seinem Status und fällt danach auf FALSE zurück.
- Der Antrieb verhält sich gemäß der in der PLCopen definierten Zustandsmaschine (State Machine). Der jeweilige Status ist jedoch vom Anwender nicht lesbar.
- Alle Motion-Bausteine arbeiten ebenfalls in einer State Machine, die anzeigt, in welchem Status sich der Baustein befindet. Damit ist z. B. erkennbar, ob ein Verfahr-Kommando bereits vom Antrieb "abgeholt" wurde.

Weitere Merkmale der Bibliotheken bzw. der Anwenderschnittstelle:

- Instanzverhalten in Norm beschrieben
- Parallelität von Bewegungsvorgaben
- Sollwerte: Lage, Geschwindigkeit, Beschleunigung
- Achsadressierung AXIS_REF (vordefinierte Variable in Bibliothek)
- Fehlerbehandlung [Ausgänge "Error", "ErrorID" und "ErrorIdent"]
- Offene Eingänge
- Kaskadierung von Bausteinen

8.2.2 Bibliotheken

Unterstützte Bibliotheken

HINWEIS

Sachschäden durch das Einbinden von nicht unterstützten Bibliotheken.

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

Folgende Funktionen, Datentypen und Strukturen stehen dem Anwender, gegliedert in Bibliotheken, zur Verfügung.

Programmierhinweise

Einige Bibliotheken werden beim Anlegen eines Projekts mit einem IndraDrive-Zielsystem automatisch geladen, weitere Bibliotheken können vom Anwender in das Projekt eingebunden werden.

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_PLCopen	- 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_Utilityies	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-Programmierumgebung und Sercos III-Teilnehmern ermöglicht wird
RIL_SocketComm	Funktionen und Funktionsbausteine, um die TCP-/UDP-Kommunikation nutzen zu können
RIL_Utilityies	Funktionen und Funktionsbausteine für Konvertierung und Beeinflussung verschiedener Datentypen
RMB_SercosIII_Util (ab Firmware-Version 20)	Funktionsbausteine für die Achsansteuerung über den Sercos Bus

Bibliothek	Beschreibung
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

Hierarchischer Aufbau der Bibliotheken

Nachfolgende Grafik erläutert anhand des Zielsystems "IndraDrive MP18", wie die Bibliotheken aufeinander aufbauen. Prinzipiell muss der Anwender in der Regel selbst keine Bibliotheken einbinden, sondern kann sich auf die automatisch eingebundenen Bibliotheken stützen.



Die Bibliothek "MX_PLCOpen" zieht automatisch ihre untergeordneten Bibliotheken mit ("MX_Base", "MX_CommonTypes").

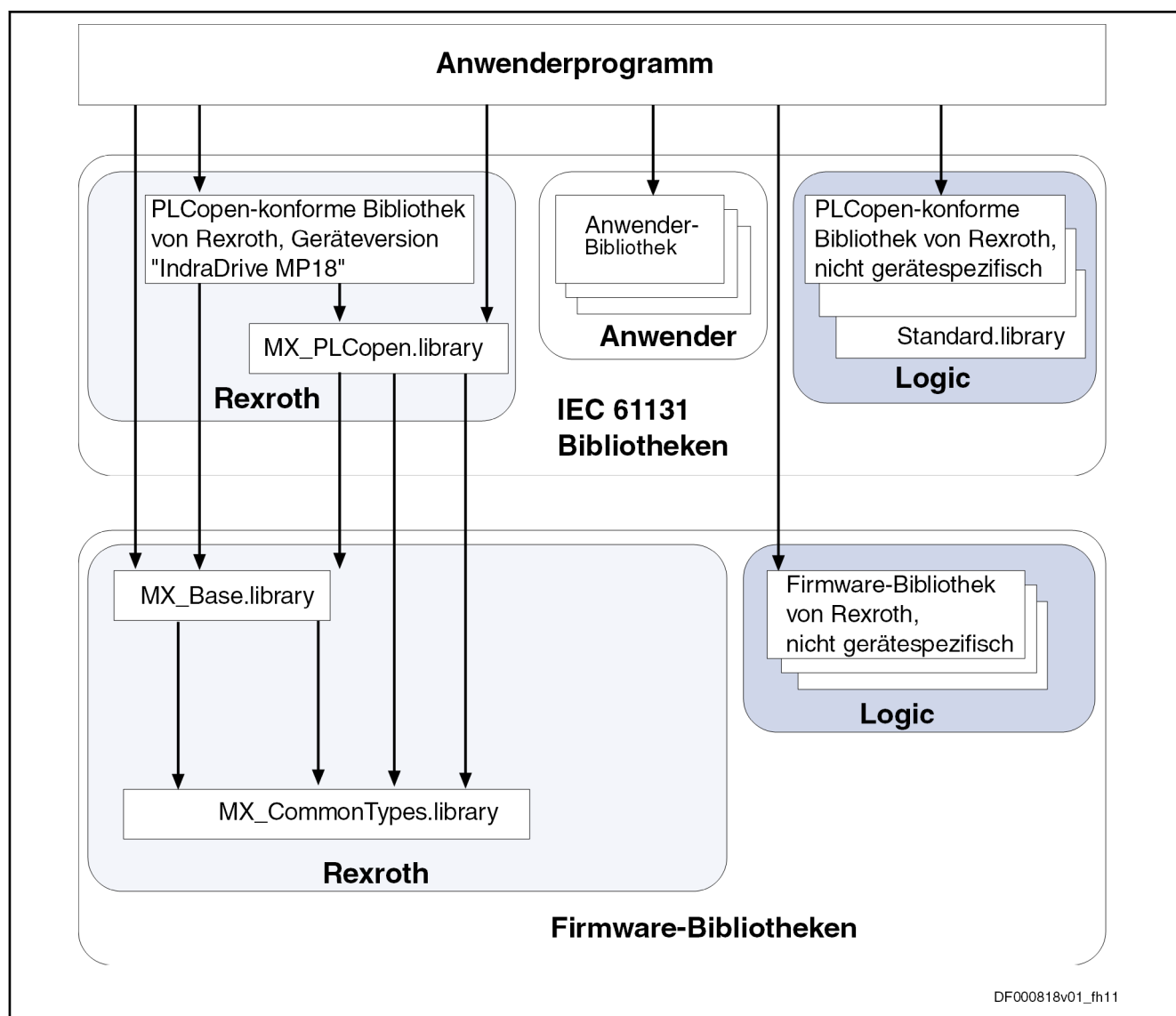


Abb. 8-2: Übersicht - Hierarchischer Aufbau der Bibliotheken

8.3 Retaindaten sichern und wieder herstellen

Retaindaten können gesichert und wieder hergestellt werden. Dies kann manuell oder automatisiert erfolgen, etwa bei Systemereignissen oder zyklisch zeitgesteuert.

Hier nur zwei der Anwendungsmöglichkeiten:

- Backup für den Fall eines unbeabsichtigten Verlustes der Retaindaten ([Retaindaten manuell sichern und wieder herstellen](#))
- [Persistente Variablen bei Reset Ursprung sichern und bei SPS-Neustart mit gesicherten persistenten Daten initialisieren](#)

8.3.1 Retaindaten manuell sichern und wieder herstellen

Retaindaten können manuell gesichert und wieder hergestellt werden, indem die Werte der persistenten Variablen (Retaindaten), wie in dem folgenden Beispiel gezeigt, in einer Rezeptur gespeichert und anschließend wieder geladen werden.

In dem Beispiel wird vorausgesetzt, dass eine persistente Variablenliste mit folgendem Inhalt angelegt wurde:

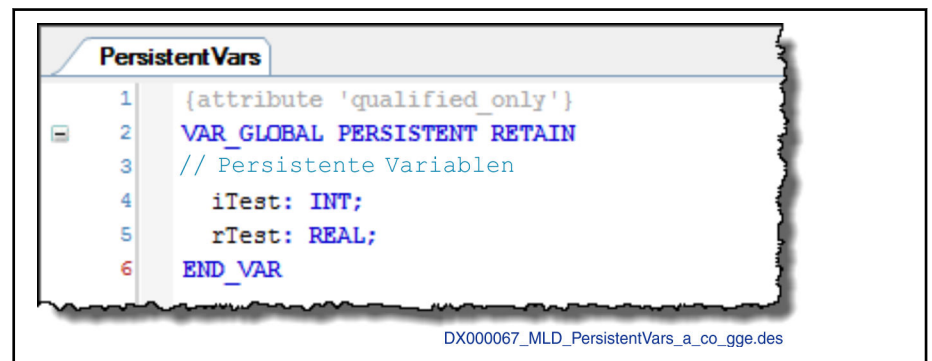


Abb. 8-3: Persistente Variablenliste

1. Starten Sie IndraWorks und loggen Sie sich in die SPS ein.
2. Machen Sie einen Doppelklick auf die persistente Variablenliste.
⇒ Der Editor zur persistenten Variablenliste wird geöffnet.
3. Öffnen Sie das Kontextmenü des Editors zur persistenten Variablenliste und wählen Sie das Menü **Aktuelle Werte in Rezeptur speichern** aus.

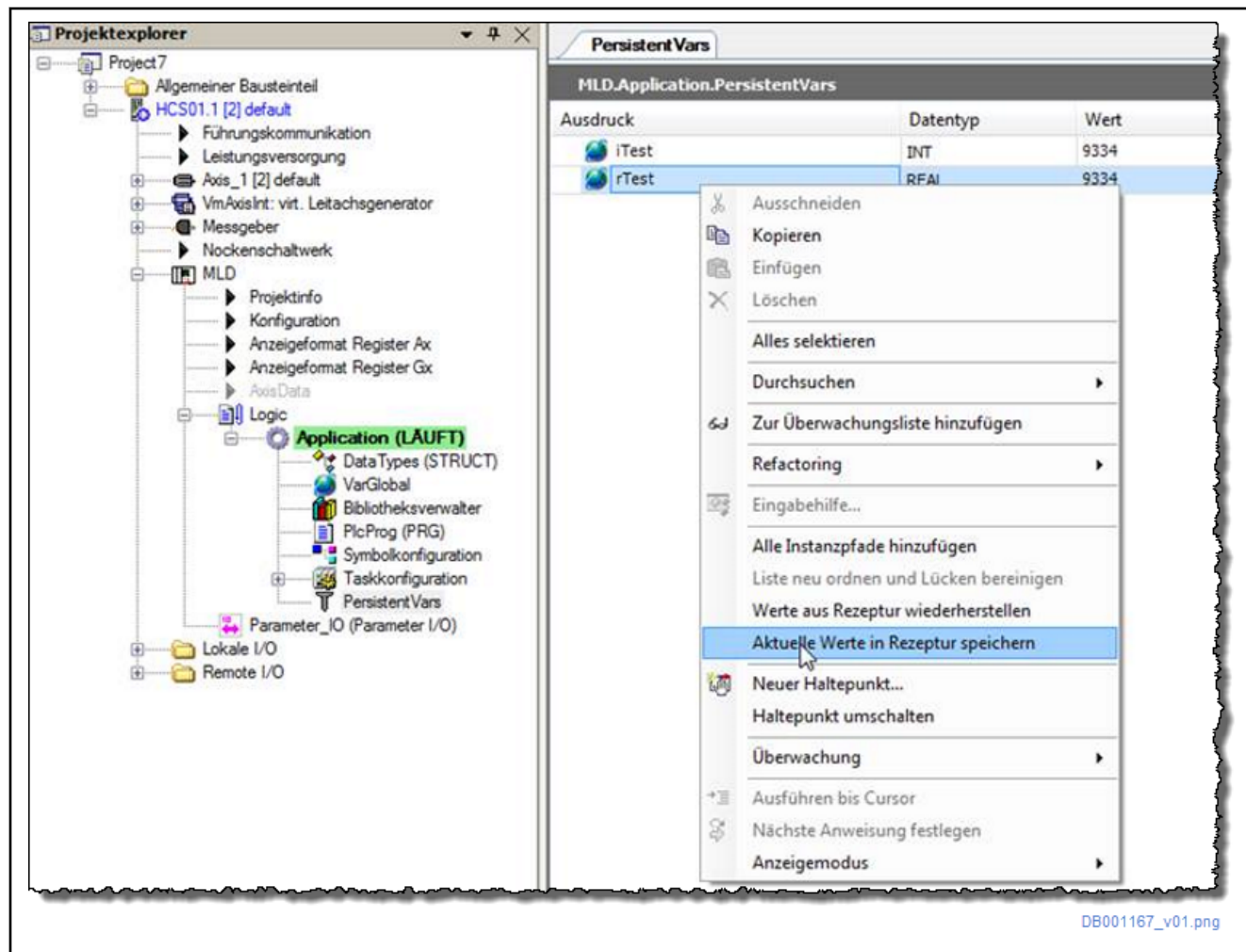


Abb. 8-4: Kontextmenü des Editors zur persistenten Variablenliste

⇒ Im Projektextplorer unterhalb des Application-Knotens wird der Rezepturverwalter und eine Rezepturdefinition für die persistente Variablenliste angelegt (hier Rezepturdefinition "PersistentVariables"):

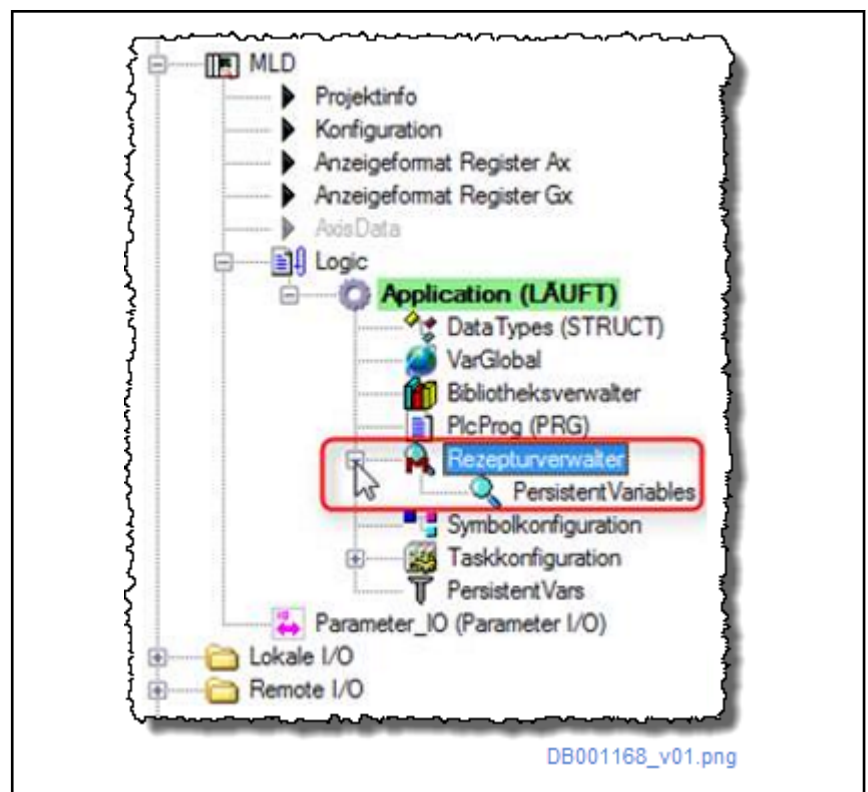


Abb. 8-5: Rezepturverwalter

4. Öffnen Sie "PersistentVariables" (Doppelklick).

PersistentVariables							
Variable	Typ	Name	Kommentar	Minimalwert	Maximalwert	Aktueller Wert	PersVars
PersistentVars.iTest	INT					32278	13216
PersistentVars.rTest	REAL					32278	13216

DB001169_v01.png

Abb. 8-6: Dialog "PersistentVariables"

5. Neben den aktuellen Werten der persistenten Variablen wird die Rezeptur "PersVars" angezeigt. Zeigen Sie mit der Maus auf eine Zelle in der Spalte "PersVars" und öffnen Sie das Kontextmenü.
 ⇒ Es werden die verfügbaren Kommandos für die Rezeptur "PersVars" angezeigt:

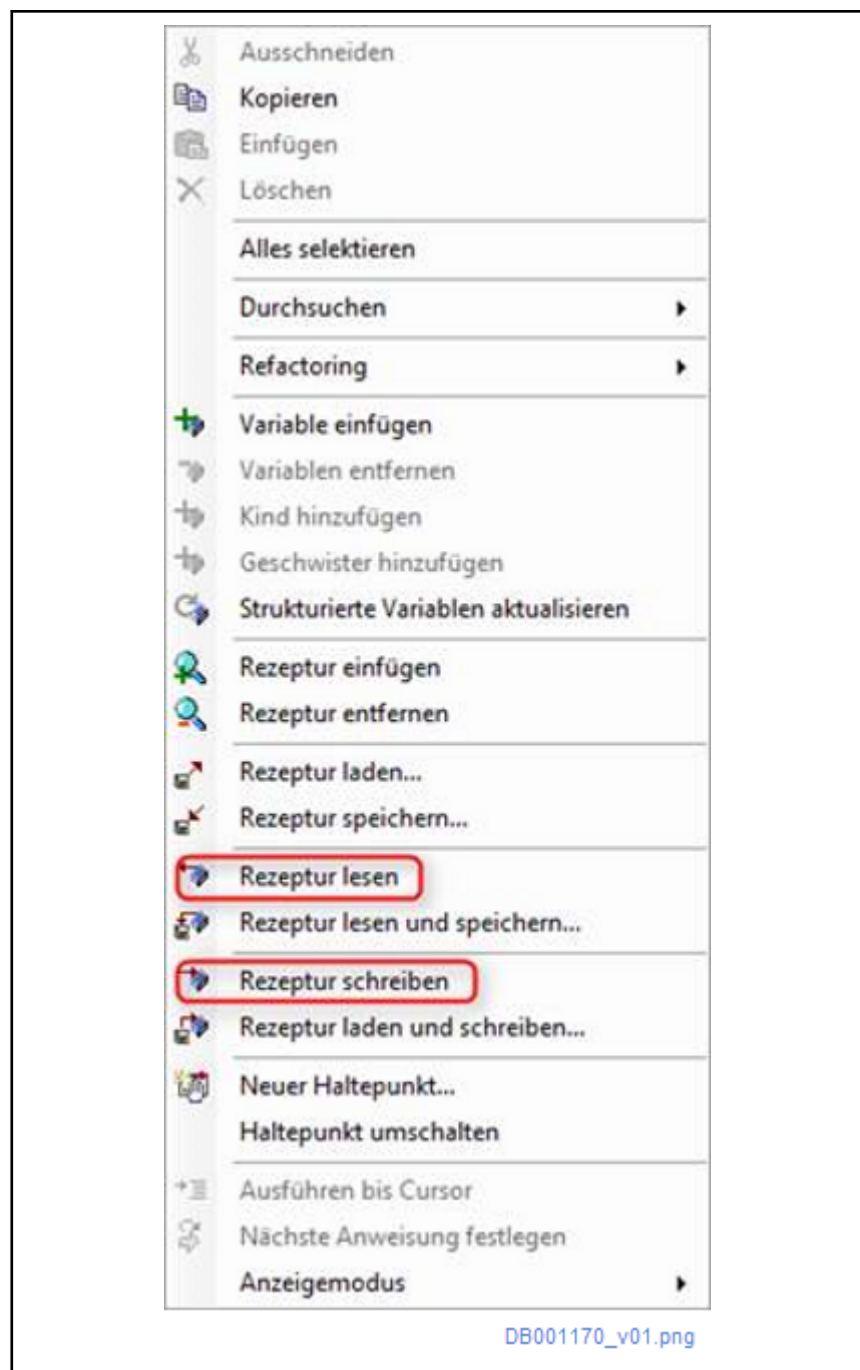


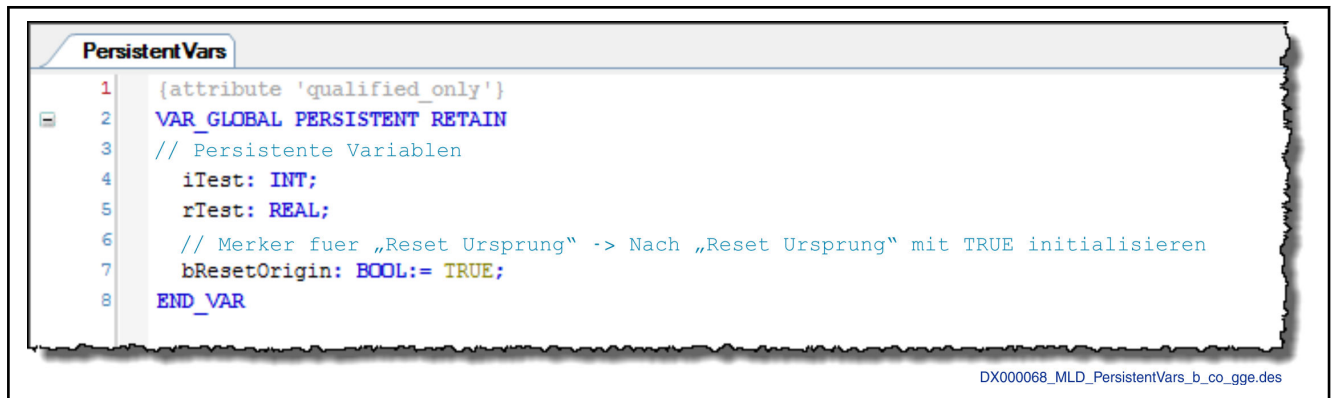
Abb. 8-7: Kontextmenü einer Rezeptur

- Mit "Rezeptur lesen" werden die aktuellen Werte der persistenten Variablen in der SPS in die entsprechenden Zellen der Spalte "PersVars" übernommen.
- Mit "Rezeptur schreiben" werden die aktuellen Werte der persistenten Variablen in der SPS mit den Werten der entsprechenden Zellen der Spalte "PersVars" überschrieben. Die Werte werden also von der Rezeptur in die SPS geschrieben.
- Mit "Rezeptur speichern..." kann die Rezeptur als Datei (*.txtrecipe) auf dem PC gespeichert und zu einem späteren Zeitpunkt mit "Rezeptur laden..." wieder geladen werden.

8.3.2 Persistente Variablen bei "Reset Ursprung" sichern und bei SPS-Neustart mit gesicherten persistenten Daten initialisieren

In dem folgenden Beispiel wird gezeigt, wie persistente Variablen bei "Reset Ursprung¹⁾" automatisch gesichert und nach Neuladen des SPS-Programms und Start der SPS mit den gesicherten Werten initialisiert werden können.

In dem Beispiel wird vorausgesetzt, dass eine persistente Variablenliste mit folgendem Inhalt angelegt wurde:



```
1 {attribute 'qualified_only'}
2 VAR_GLOBAL PERSISTENT RETAIN
3 // Persistente Variablen
4 iTest: INT;
5 rTest: REAL;
6 // Merker fuer „Reset Ursprung“ -> Nach „Reset Ursprung“ mit TRUE initialisieren
7 bResetOrigin: BOOL:= TRUE;
8 END_VAR
```

DX000068_MLD_PersistentVars_b_co_gge.des

Abb. 8-8: Persistente Variablenliste

1. Starten Sie IndraWorks und loggen Sie sich in die SPS ein.
2. Machen Sie einen Doppelklick auf die persistente Variablenliste.
⇒ Der Editor zur persistenten Variablenliste wird geöffnet.
3. Öffnen Sie das Kontextmenü des Editors zur persistenten Variablenliste und wählen Sie das Menü **Aktuelle Werte in Rezeptur speichern** aus.

¹⁾ Der Befehl "Reset Ursprung" setzt alle Variablen der aktiven Applikation auf ihre Initialisierungswerte zurück und löscht das Applikationsprogramm auf der Steuerung

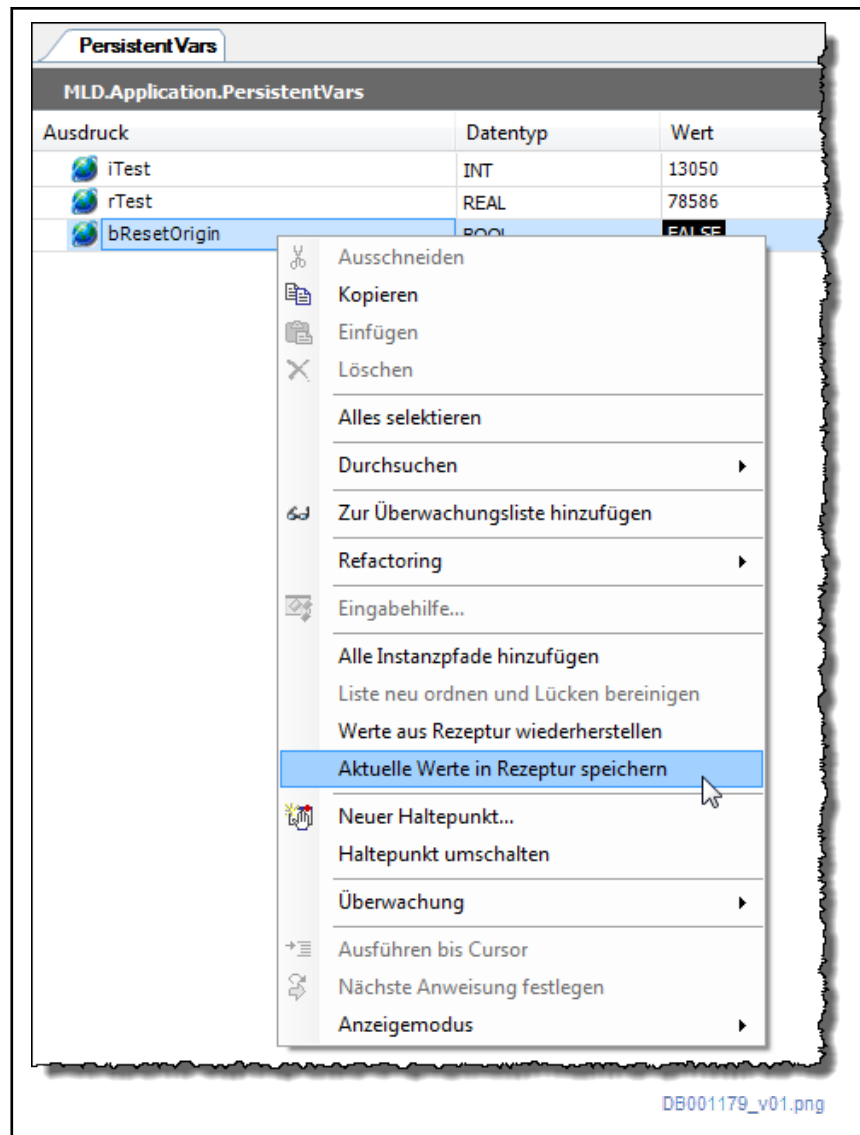


Abb. 8-9: Kontextmenü des Editors zur persistenten Variablenliste

⇒ Im Projektextplorer unterhalb des Application-Knotens wird der Rezepturverwalter und eine Rezepturdefinition für die persistente Variablenliste angelegt (hier Rezepturdefinition "PersistentVariables"):

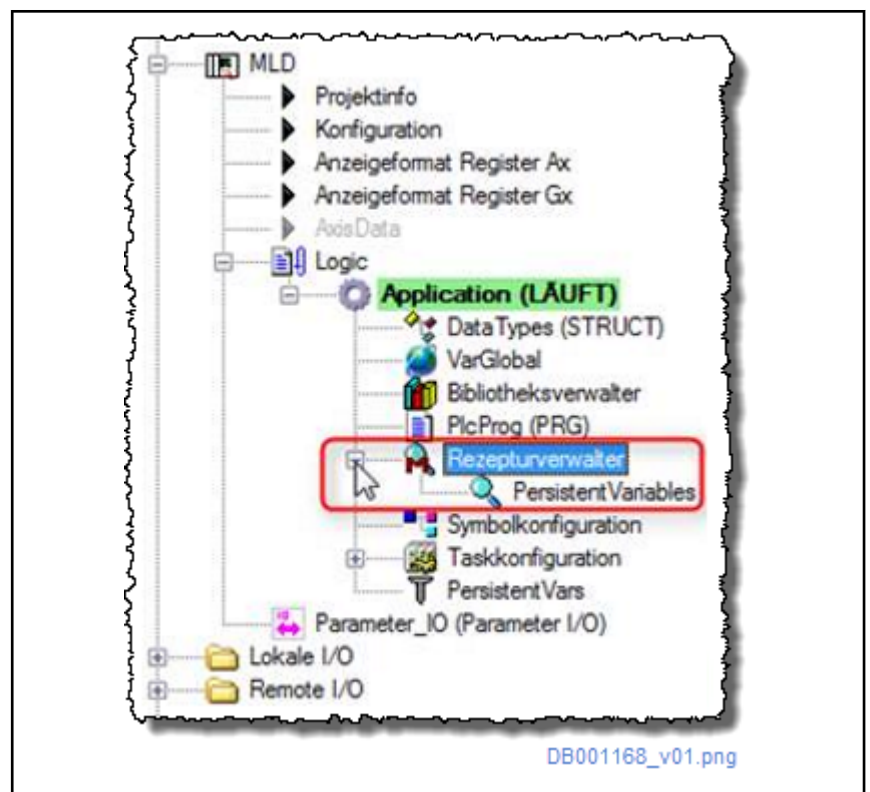


Abb. 8-10: Rezepturverwalter

4. Öffnen Sie den Rezepturverwalter (Doppelklick auf "Rezepturverwalter" im Projektexplorer) und wechseln Sie auf die Registerkarte "Allgemein".
5. Aktivieren Sie die Kontrollkästchen **Rezepturverwaltung in der SPS** und **Rezepturänderungen automatisch in Rezepturdateien speichern**.

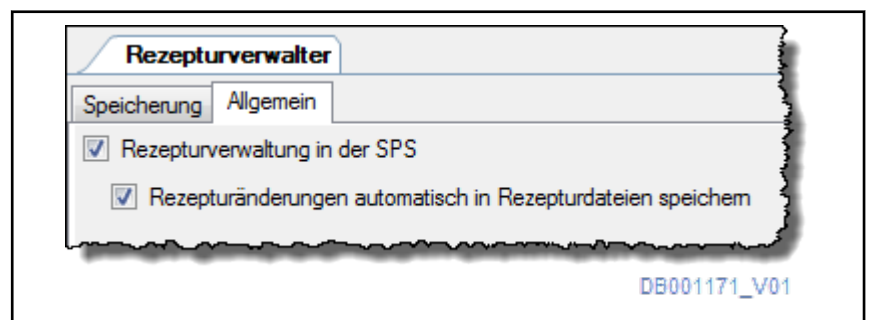


Abb. 8-11: Rezepturverwalter konfigurieren

6. Öffnen Sie die Taskkonfiguration und wechseln Sie auf die Registerkarte "Systemereignisse"
7. Fügen Sie über die Schaltfläche **Event-Handler hinzufügen...** die beiden Ereignisse "PrepareReset" und "PrepareStart" hinzu. Geben Sie jeweils eine Funktion an, die bei Eintreffen der Events aufgerufen werden soll. In dem Beispiel wurden die Funktionen "MyReset" und "MyStart" genannt.

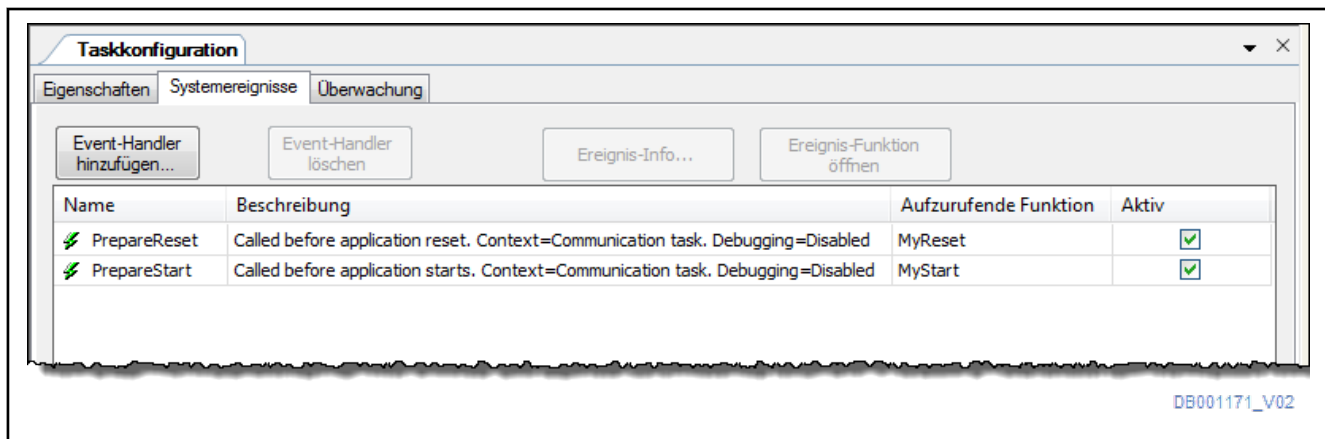


Abb. 8-12: Event-Handler hinzufügen

8. Die Funktion "MyReset" wird von IndraWorks beim Hinzufügen des EventHandlers "PrepareReset" und Angabe der auszuführenden Funktion (MyReset) automatisch angelegt.

In der Funktion "MyReset" muss programmiert werden, was bei einem Reset der SPS erfolgen soll.

In dem nachfolgenden Beispiel werden bei "Reset Ursprung" die Werte der persistenten Variablen in die Rezeptur "PersVars" gelesen und als Datei ("MyPersVars.txtrecipe") gespeichert:



Abb. 8-13: Mögliche Umsetzung der Anforderung in der Funktion "MyReset"

9. Die Funktion "MyStart" wird von IndraWorks beim Hinzufügen des EventHandlers "PrepareStart" und Angabe der auszuführenden Funktion ("MyStart") automatisch angelegt.

In der Funktion "MyStart" muss programmiert werden, was beim Start der SPS erfolgen soll.

In dem nachfolgenden Beispiel werden die Werte aus der Rezepturdatei "MyPersVars.txtrecipe" in die Rezeptur "PersVars" geladen und in die persistenten SPS-Variablen übernommen. Nach dem einmaligen Einlesen der persistenten Variablen wird die Variable .PersistenVars.bResetOrigin auf FALSE gesetzt, um zu ver-

hindern, dass die persistenten Variablen bei jedem Start der SPS mit den Werten aus der Rezeptur befüllt werden, sondern nur nach einem "Reset Ursprung":



Abb. 8-14: Mögliche Umsetzung der Anforderung in der Funktion "MyStart"

8.4 Dateizugriffe auf die optionale Speicherkarte

Es stehen synchrone und asynchrone Zugriffe auf das Dateisystem zur Verfügung.

Synchrone Zugriffe auf das Dateisystem

Bei den synchronen Zugriffen auf das Dateisystem wird eine Funktion aufgerufen, die erst nach Beendigung der Ausgabe zurückkehrt. Dies ist für eine typische SPS-Task ungünstig, da diese Task in dieser Zeit auf die Funktion "wartet". Daher ist es je nach Anforderung empfehlenswert, diese Aufgaben in einer niederpriorigen Task ohne Watchdog oder mit sehr großem Watchdog auszuführen.

Asynchrone Zugriffe auf das Dateisystem

Bei den asynchronen Zugriffen auf das Dateisystem werden zyklisch orientierte Funktionsbausteine verwendet. Wie bei den Motion-Funktionsbausteinen werden die Aufträge mit einer Flanke am Eingang aktiviert. Der Status des Auftrags wird in weiteren Aufrufen geliefert. Die Funktionsbausteine müssen zyklisch aufgerufen werden, bis sie "bDone" melden. Je nach Funktion steckt in einem weiteren Ausgang die Returninformation.

Die Laufzeiten der Bausteine sind relativ gering, so dass sie in einer zyklischen Task eingesetzt werden können.

Die eigentliche Aufgabe wird in einer eigenen niederpriorigen Systemtask erledigt. Diese Task wird auch bei voller Auslastung der SPS nicht behindert. Ihre Geschwindigkeit ist jedoch von der Auslastung des Antriebs abhängig.

Die Bibliothek mit Bausteinen für den asynchronen Zugriff auf das Dateisystem eignet sich zum Einbau in eine typische zyklische SPS-Task mit Watchdog.

Bibliotheken für Zugriffe auf das Dateisystem

Für Zugriffe auf das Dateisystem stehen die Systembibliotheken "SysFile", "SysFileAsync" und "SysDir" zur Verfügung.

Ordnerstruktur für Zugriffe auf das Dateisystem

Die Dateien werden in einem speziellen Verzeichnis "User" auf der Speicherkarte abgelegt. Die Funktionsbausteine und Funktionen werden alle mit dem reinen Dateinamen und optional einem führenden Unterverzeichnis (Anwender-Unterverzeichnis) aufgerufen.

Mit der SPS kann nur auf den Ordner "User" zugegriffen werden; auf alle anderen Ordner und Dateien kann mit der SPS aus Sicherheitsgründen nicht zugegriffen werden.

Beim Listen der Verzeichnisse im Windows-Explorer wird nur der Inhalt des "User"-Ordners 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.

Dateizugriffe Die Open-Funktionen öffnen Dateien nicht exklusiv, so dass andere Zugriffe auf die selbe Datei möglich sind.

Solange das SPS-Programm läuft, können Dateien offen gehalten werden. Sie werden jedoch zwangsweise geschlossen, wenn das SPS-Programm mit "Reset" zurückgesetzt wird oder die Speicherkarte abgezogen wird. In letzterem Fall wird ein Fehler F2006 erzeugt.

Es können maximal 8 Dateien gleichzeitig geöffnet werden. Weiteres Öffnen von Dateien meldet einen Fehler.

"SysDirOpen" darf maximal 4 mal ohne Leerlesen mit "SysDirRead" gestartet werden. Ein weiterer Aufruf meldet Fehler.

Bei einem **Reset der SPS** werden offene Dateien und Directory-Suchhandles geschlossen und alle Ressourcen freigegeben.

GEFAHR

Wenn während des Schreibens auf die Speicherkarte ein Spannungsausfall auftritt, kann es zu Datenverlust kommen. Im schlimmsten Fall ist die Speicherkarte anschließend nicht mehr lesbar.

Um Datenverlust vorzubeugen, sollte in regelmäßigen Abständen eine Sicherung der gesamten Speicherkarte gemacht werden.

Wenn eine Speicherkarte im Antrieb nicht mehr lesbar ist, ist es unter Umständen möglich, sie mit einem PC und einem Cardreader durch die Windows-Funktion "Überprüfen eines Datenträgers" zu reparieren. Im Fehlerfall werden "Lost Clusters" in Dateien zusammengefasst

8.5 Startverhalten und Bootprojekt

8.5.1 Allgemeines

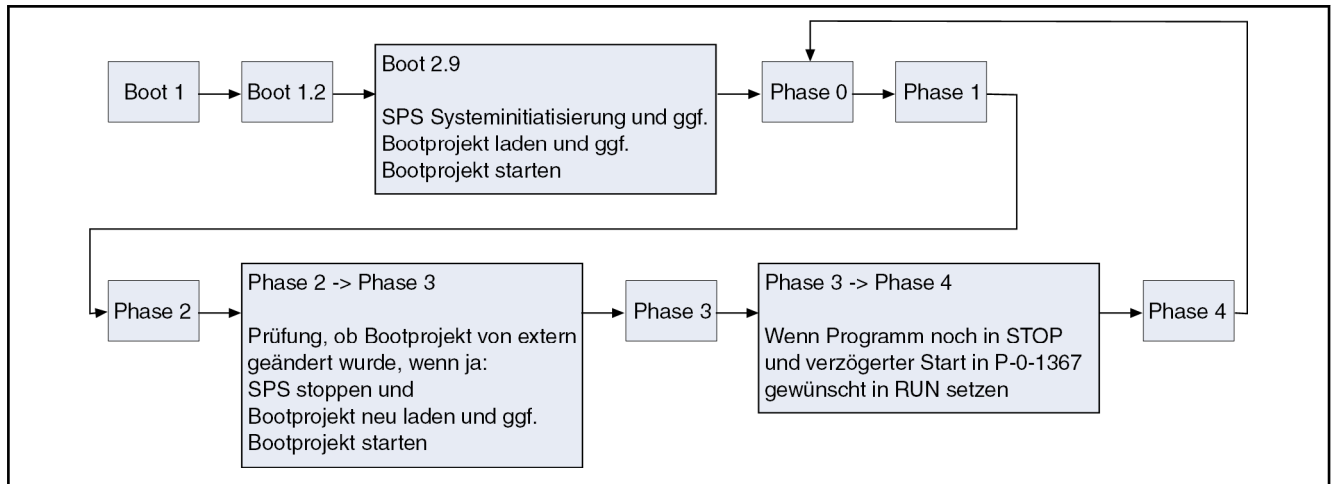
Da die antriebsintegrierte Steuerung als Task mitläuft, wird sie zusammen mit dem Antrieb initialisiert. Sie unterliegt auch bezüglich der verschiedenen Kommunikationsphasen den Konventionen die für die Antriebsfirmware getroffen wurden.

Siehe auch Firmware-Funktionsbeschreibung "Geräteinterne Zustandsmaschine".

8.5.2 Kurzbeschreibung

Die antriebsintegrierte SPS (Rexroth IndraMotion MLD) wird im Bootvorgang (Boot 2.9) aktiviert. Sie initialisiert sich dabei zunächst und prüft, ob ein korrektes Bootprojekt vorliegt. Wenn ja, wird es geladen. Je nach Konfiguration wird die antriebsintegrierte SPS in RUN gesetzt, verbleibt in STOP oder wird erst in Phase 4 (Betriebsmodus) gestartet (Parametrierung über "P-0-1367, SPS Konfiguration").

Nachfolgende Grafik zeigt den Bootvorgang und das Durchlaufen der Kommunikationsphasen der Rexroth IndraMotion MLD:



Grafikname DF000133

Abb. 8-15: Hochlauf der Rexroth IndraMotion MLD

Zugehörige Parameter

Folgende Parameter werden in diesem Zusammenhang verwendet:

- P-0-1367, SPS Konfiguration
- S-0-0127, C0100 Umschaltvorbereitung auf Komm.-Phase 3
- S-0-0128, C0200 Umschaltvorbereitung auf Komm.-Phase 4

Beteiligte Diagnosen

Folgende Diagnosen treten während der Hochlaufphase auf:

- C0100 Umschaltvorbereitung Phase 2 nach 3
- C0200 Umschaltvorbereitung Phase 3 nach 4

Kommt es zu einem Fehler bei den Umschaltvorbereitungen, so wird dies mit einer Diagnose nach dem Muster C01xx bzw. C02xx gemeldet.

Bootvorgang

Beim Booten wird

- das SPS-Betriebssystem initialisiert (wenn nicht vor Boot 2.9 verhindert)
- ggf. ein Bootprojekt geladen (wenn vorhanden)
- ggf. das Bootprojekt gestartet (wenn nicht über "P-0-1367, SPS Konfiguration" verhindert)

C0100 Umschaltvorbereitung Phase 2 nach 3

Im Umschaltkommando von Kommunikationsphase 2 nach Kommunikationsphase 3 (siehe auch "C0100 Umschaltvorbereitung Phase 2 nach 3") werden die folgenden Schritte durchlaufen:

- Es wird überprüft, ob das Bootprojekt geändert wurde. Wurde das Bootprojekt geändert, dann wird die SPS gestoppt und das Bootprojekt neu geladen.
- Gegebenenfalls wird das neu geladene Programm gestartet (wenn nicht über "P-0-1367, SPS Konfiguration" verhindert).

C0200 Umschaltvorbereitung Phase 3 nach 4

Im Umschaltkommando von Kommunikationsphase 3 nach Kommunikationsphase 4 (vergleiche "C0200 Umschaltvorbereitung Phase 3 nach 4") werden die folgenden Schritte durchlaufen:

- Wenn ein vorhandenes Bootprojekt nicht gestartet wurde, wird es jetzt gestartet.
- Die Systemkontrolle, d.h. der Zugriff auf die Gerätesteuerung des Antriebes, wird gestartet.

8.5.3 Anwendungs- und Programmierhinweise

Nachfolgend werden einige Hinweise bzgl. der Konfiguration (Parametrierung) und Handhabung der MLD bei der Programmerstellung und Anwendung gegeben.

Konfiguration des Startverhaltens	Die Konfiguration des Startverhaltens (Booten) erfolgt über die Parametrierung von "P-0-1367, SPS Konfiguration" (Bit 0 und 1).
Konfiguration der Gerätekontrolle	Über Bit 4 von "P-0-1367, SPS Konfiguration" wird die Kontrolle über den Antrieb, d. h. der Zugriff auf die interne Gerätesteuerung, konfiguriert.
Bootprojekt	Beim Wechsel von Phase 2 in Phase 3 wird geprüft, ob das Bootprojekt als letztes über Parameter-Write (also nicht von IndraLogic) geändert wurde. Wenn ja, dann wird die SPS deinitialisiert und neu initialisiert. Damit lädt sie das neu eingespielte Bootprojekt.
Wann läuft die SPS?	Die SPS läuft beim Booten nach dem Freischalten der Systemtasks - also auch in Phase 2 - los. Beim Schalten zwischen Phase 2 und 4 läuft sie durch. Sie wird nur bei geändertem SPS-Bootprojekt zwischen Phase 2 und 4 kurz angehalten und mit dem neuen Projekt neu gestartet.
Notfallmechanismus zum Stoppen der SPS	Da beim Hochfahren des Antriebs automatisch ein Bootprojekt geladen wird, wurde ein Mechanismus implementiert, mit dem das Starten von SPS-Programmen verhindert werden kann: • In der Bootphase müssen, bevor die Anzeige Boot 2.9 auf dem Display des Bedienfelds angezeigt wird, an dem Bedienfeld gleichzeitig die beiden Tasten "ESC" und "ENTER" gedrückt und gehalten werden. • Das automatische Starten eines SPS-Bootprojekts wird verhindert und es erscheint die Anzeige "PLC ?". • Durch Betätigen der Pfeiltasten (Pfeil nach unten bzw. Pfeil nach oben) wechselt die Anzeige zwischen "Run PLC" und "Stop PLC". Wird "Stop PLC" mit "ENTER" bestätigt, wird kein Projekt geladen; wird "Run PLC" mit "ENTER" bestätigt, wird ein evtl. gespeichertes Projekt gestartet oder geladen.

8.6 Richtlinien für die Programmierung mit IndraLogic

8.6.1 Übersicht

Ziel dieses Kapitels ist die Vermittlung von Grundlagen für den Aufbau und die Programmierung von SPS-Projekten mit IndraLogic für das Zielsystem "Rexroth IndraMotion MLD".

Es werden Vorschläge und Empfehlungen für die SPS-Programmierung in Form von Basis-Richtlinien gegeben. Diese sollen den einheitlichen Einsatz von Softwarestrukturen ermöglichen.

Grundsätzliche Kenntnisse zum IndraLogic SPS-Programmiersystem werden vorausgesetzt (siehe Dokumentation "IndraLogic 2G Programmieranleitung").



Wünschenswert ist die vollständige Anwendung der Richtlinien. Ist dies nicht realisierbar, sollte eine nahe an den Richtlinien orientierte Umsetzung angestrebt werden.

8.6.2 Strukturierung von SPS-Projekten

Taskverwaltung

Grundsätzlich empfehlen wir eine Taskkonfiguration anzulegen, damit nicht der automatische Singletask-Mechanismus wirkt (max. 4 Tasks möglich).



Wird eine Taskkonfiguration angelegt, kommt das (Single-Task) Hauptprogramm **PLC_PRG** nicht zum Einsatz.

Die Taskkonfiguration bietet durch preemptive zeitgesteuerte und eventgesteuerte sowie eine permanente "freilaufende" Task immer eine optimal auf die Aufgabe abgestimmte Lösung.

Generell sollte man so wenige Tasks wie möglich und so viele wie der Aufgabe angemessen einsetzen. Damit bleiben SPS-Projekte übersichtlich und einfach testbar.

Mehrprogrammtechnik

Jeder Task muss ein Haupt-Programm zugeordnet werden. Dieses Programm kann bedarfsweise weitere Bausteine (Programme und Funktionsbausteine) enthalten (Mehrprogrammtechnik). Dies bietet eine hohe Flexibilität in Bezug auf Aufruf und Steuerung der weiteren Bausteine. Außerdem ist damit die Reihenfolge innerhalb des jeweiligen Haupt-Programms eindeutig.



Es ist möglich jeder Task mehrere Programmen zuzuweisen; die Abarbeitung der Programme entspricht der zugewiesenen Reihenfolge. Eine weitere Steuerung dieser Programme ist jedoch dann nur innerhalb der Programme selbst möglich, daher wird diese Variante nicht favorisiert.

Über die Namensgebung der Bausteine kann die Reihenfolge der Darstellung im Register "Bausteine" (Tree) beeinflusst werden. Es ist vorteilhaft, die Hauptprogramme an den Anfang zu stellen, z. B. "_prMain_<Maschinennummer>".

Die Anordnung der Bausteine kann innerhalb der Baustein-Sicht über weitere Unter-Ordner strukturiert werden. Damit lässt sich eine funktionale Gruppierung vornehmen, die das gesamte Projekt übersichtlicher macht.



Das "Abtauchen" in die Programme/Funktionsbausteine kann über das Register "Bausteine" oder im Programmeditor über <Alt>+<Enter> erfolgen. Allein die Aufrufstelle bestimmt den Ort in der zeitlichen Abarbeitung. Ggf. sollte die Baustein-Bezeichnung so erfolgen, dass die Reihenfolge auch im Tree ersichtlich ist.

8.6.3 Programmiersprachen

Übersicht

Insgesamt stehen in IndraLogic "5+1" Programmiersprachen zur Verfügung:

- Anweisungsliste (AWL)
- Strukturierter Text (ST)
- Kontaktplan (KOP)
- Funktionsplan (FUP)
- Freigrafischer Funktionsplan (CFC)
- + Ablaufsprache (AS)

Welche Basis-Programmiersprache verwendet werden sollte, ist von dem Anwendungsfall abhängig.

Empfohlene Basis-Programmiersprachen

Logik-Verarbeitung	Als Basis-Programmiersprache sollte der Kontaktplan (KOP) eingesetzt werden. Dies betrifft insbesondere die Basislogik, die von dem Maschineneinrichter in Betrieb genommen und vom Betreiber der Anlagen gewartet und ggf. programmiert wird (insbesondere E/A-Handling der Maschinenperipherie).
Motion Control	Welche Programmiersprache für die Verwendung von "Motion Control"-Funktionen eingesetzt wird, richtet sich nach den Festlegungen des Betreibers einer Anlage. Wenn Schnittstellen und Funktionsweisen von Bausteinen exakt dokumentiert sind, dann ist es weniger wichtig welche Programmiersprache gewählt wurde. Bei jedem neuen Baustein kann die Programmiersprache neu gewählt werden.  Auch bei jeder neuen Aktion kann die Programmiersprache neu gewählt werden.  Aktionen können auch innerhalb von Bausteinen und nicht nur innerhalb von Abläufen der Ablaufsprache (AS) verwendet werden. Diese Aktionen benutzen dann die gleiche Deklaration wie der Baustein selbst.

Hinweise zur SFC-Programmierung

Es sollte weitestgehend mit den so genannten IEC-Schritten gearbeitet werden (**Extras ► IEC-Schritte benutzen**). An einem IEC-Schritt können maximal 9 Aktionen und je eine Eingangs- und Ausgangs-Aktion programmiert werden.

 Die Reihenfolge der Abarbeitung von Aktionen je Schritt erfolgt alphanumerisch und nicht anhand der grafischen Reihenfolge in der AS-Kette!

Hierbei gilt:

- Ein "Abtauchen" in die Aktionen ist nur über die Baustein-Sicht möglich.
- Transitionen dienen als Weiterschaltbedingungen und können, wie die Aktionen auch, in einer beliebigen Sprache programmiert werden.

In einem SPS-Zyklus werden zunächst immer die aktiven Schritte/Aktionen ausgeführt und erst danach wird die Transition berechnet. Bei erfüllter Transition wird im nächsten SPS-Zyklus die Schrittkette weitergeschaltet, dann aber noch vor dem neuen Schritt/Aktion(en) die Aktionen des/der bisherigen Schritt(e) nachbearbeitet. Daher ist es hier überlegenswert, ob man eine freilaufende Task benutzt, da hier die Bearbeitung quasi sofort nach dem Wechsel in einen anderen Schritt wieder aufgenommen wird.

Für einfache Abläufe ist die Programmierung mit den vereinfachten Schritten möglich. Schritt und Aktion verschmelzen zu einer Logik-Implementation je Schritt. Das "Abtauchen" in die Logik erfolgt direkt über den Schritt.

8.6.4 Globale Daten

In allen Bausteinen kann auf globale Daten zugegriffen werden.

In der Regel werden globale Daten für die Querkommunikation zwischen Bausteinen verwendet und können somit die Aufrufschnittstelle von Bausteinen entlasten. Außerdem muss die Aufrufschnittstelle bei Änderungen nicht

wiederholt angepasst werden, was zum Vorteil hat, dass das Aussehen beibehalten wird und die Anwenderdokumentation des SPS-Projekts nicht geändert werden muss.

Die Deklaration globaler Daten erfolgt im Register "Ressourcen". Es können dort weitere globale Daten-Listen und Unterordner angelegt werden, um eine bessere Strukturierung zu erzielen.

Insbesondere können Deklarationen globaler Daten auch als zentrale Stelle der Deklaration von E/A-Daten dienen, um eine Gesamtliste an E/A-Daten im SPS-Projekt zu haben (falls gewünscht).

8.6.5 Programm-Header

Ein Header (Programm-Kopf) zeigt dem Anwender schnell und übersichtlich wichtige Informationen über den zugeordneten Programmcode.

Ein vollständig ausgefüllter Header enthält die folgenden Informationen:

- die Funktion / der Verwendungszweck des SPS-Programms in knappen Worten
- aktueller Versionsstand
- Name des Programmierers
- letztes Änderungsdatum
- Unternehmen für welches der Programmierer tätig ist
- auf welchen IndraLogic-Zielsystemen das SPS-Programm lauffähig ist (Target)
- die ausführliche Beschreibung der Funktion / des Verwendungszwecks des SPS-Programms, sowie Hinweise auf spezielle Vorbereitungen oder Bedienungen

Der Header soll in Englisch verfasst werden.



Die Verwendung des Headers wird dringend empfohlen.

Um einen einheitlichen Header anzuwenden, kann die nachfolgende Vorlage verwendet werden. Markieren Sie dazu den Text in die Zwischenablage. Fügen Sie anschließend den Inhalt der Zwischenablage im Deklarationsteil, vor allem anderen Code in IndraLogic ein.



Ist im IndraLogic-Editor **Deklaration als Tabelle** ausgewählt, dann müssen Sie den Inhalt der Zwischenablage unter der Registerkarte "Info" einfügen.

Header-Vorlage

Programm:

```
(*#####*)
(*#####*)
(* General Header *)
(*-----*)
(* Short description      : *)
(* Version                : *)
(* Name                   : *)
(* Date                   : *)
(* Company                : *)
(* Target                 : *)
(* Functional description : *)
(* Handling specials      : *)
(*-----*)
(* Additional Header *)
(*-----*)
(* *)
```

```
( * *)
( * *)
( * *)
( * *)
( *-----*)
( *#####*)
( *#####*)
```

8.6.6 History

In der History werden Code-Änderungen dokumentiert.

Eine vollständig ausgefüllte History enthält die folgenden Informationen:

- Versionsstand
- Name des Programmierers
- Änderungsdatum
- Beschreibung der Änderung

Der Header soll in Englisch verfasst werden.



Die Verwendung der History wird dringend empfohlen.

Um eine einheitliche History anzuwenden, kann die nachfolgende Vorlage verwendet werden. Markieren Sie dazu den Text in die Zwischenablage. Fügen Sie anschließend den Inhalt der Zwischenablage in IndraLogic am Ende der Deklaration ein.



Wurde die History am Ende der Deklaration eingefügt, so ist beim Wechsel in die tabellarische Ansicht und einer anschließenden Änderungen in der Deklaration die History nicht mehr sichtbar!



Ist im IndraLogic-Editor **Deklaration als Tabelle** ausgewählt, dann müssen Sie den Inhalt der Zwischenablage unter der Registerkarte "Info" einfügen, direkt nach dem Header.

History-Vorlage

Programm:

```
(*#####*)
(*Modification - History*)
(*-----*)
(* Version      : *)
(* Name         : *)
(* Date         : *)
(* Comment      : *)
(**)
(* Version      : *)
(* Name         : *)
(* Date         : *)
(* Comment      : *)
(**)
(* Version      : *)
(* Name         : *)
(* Date         : *)
(* Comment      : *)
(*-----*)
(*#####*)
```

8.6.7 Typbezeichner

Allgemeines

Ein einheitliches Vorgehen bei der Vergabe von Namen für Typen erhöht die Lesbarkeit des Programmcodes. Damit wird Dritten die Einarbeitung in den Code wesentlich erleichtert sowie eine eventuelle Fehlersuche verkürzt.

Prinzipiell kann zwischen zwei Typen unterschieden werden:

- Typen, die Programmcode enthalten können. Dazu gehören Programme, Funktionsbausteine und Funktionen.
- Typen, die keinen Programmcode enthalten. Dazu gehören Strukturen, Felder und Enumeratoren sowie alle IEC-Datentypen wie z. B. String, Integer oder Real.

Typbezeichner für Programme, Funktionsbausteine und Funktionen

In Anlehnung an die PLCopen sind die Typbezeichner für Programme, Funktionen und Funktionsbausteine unter Berücksichtigung von Groß-/Kleinschreibung und in englischer Sprache vergeben worden.



Es besteht ein Konflikt zwischen PLCopen und IEC-Norm. Funktionen und Funktionsbausteine nach IEC-Norm sind groß geschrieben (z. B. R_TRIG, TON, usw.). Dagegen muss bei Funktionen und Funktionsbausteinen der PLCopen die Groß-/Kleinschreibung berücksichtigt werden (z. B. "MC_MoveAbsolute").



In den Programmierrichtlinien wird versucht, sich weitestgehend am PLCopen-Standard zu orientieren.

Programme

Typbezeichner für Programme haben keine Unterstriche, keinen Präfix oder Suffix.

Beispiel: "MainProgram"

Funktionen und Funktionsbausteine

Für Bosch Rexroth sind Kürzel für systemspezifische sowie systemunabhängige Funktionen und Funktionsbausteine festgelegt worden. Diese sind als Präfix mit Unterstrich dem Typbezeichner hinzugefügt.

Präfix	System-abhängig-keit	Zuordnung	Beschreibung	Beispiel
MC	Nein	PLCopen	100%-ige PLCopen-Bausteine	MC_MoveAbsolute
MB	Nein	Motionorientiert in Anlehnung an PLCopen	Bausteine mit motionorientierten Funktionalitäten, die nicht PLCopen-zertifiziert sind, aber sich an dieser orientieren	MB_WriteParameter
MH	Ja	MLC	Hydraulische Funktionen, nur für MLC verfügbar	MH_HydrControl
IL	Nein	-	Alle systemunabhängigen Funktionen und Funktionsbausteine, die nicht PLCopen- und nicht motionorientiert sind	IL_SercosAttribute
ML	Ja	MLC	MLC-spezifische Funktionen und Funktionsbausteine	ML_ReadParameter-Bool
MS	Ja	Synax	Synax-spezifische Funktionen und Funktionsbausteine	MS_ReadSingleParameter
MSV	Ja	Synax + VisualMotion	Funktionen und Funktionsbausteine für die Systeme Synax und VisualMotion	MSV_ReadMaxValue
MT	Ja	MTX	MTX-spezifische Funktionen und Funktionsbausteine	MT_NcBlk
MV	Ja	VisualMotion	VisualMotion-spezifische Funktionen und Funktionsbausteine	MV_Hysteresis
MX	Ja	MLD (IndraDrive und HydraulicDrive)	Funktionen und Funktionsbausteine, speziell für die antriebsintegrierte SPS	MX_SetDeviceMode

Tab. 8-1: Übersicht Präfixe für Funktionen und Funktionsbausteine



Funktionsbausteine und Funktionen, die mit "IL", "MC" oder "MB" gekennzeichnet sind, werden nicht zwangsläufig von jedem System (z. B. MTX, SYNAX, usw.) angeboten oder unterstützt. Es ist vielmehr so, dass jedes System, welches einen Baustein mit einem "IL", "MC" oder "MB" im Namen zur Verfügung stellt, diesen so realisieren muss, dass dessen Verhalten nach außen System übergreifend gleich ist. Das bedeutet, dass z. B. der Funktionsbaustein "IL_SercosAttribute" in seinem Aussehen und Verhalten nach außen hin in jedem System, das diesen Funktionsbaustein zur Verfügung stellt, gleich sein muss.

Typbezeichner für Strukturen, Felder, Enumeratoren und IEC-Datentypen

In Anlehnung an die PLCopen werden Typbezeichner für Strukturen, Felder und Enumeratoren nur mit Großbuchstaben gebildet. Zur Verbesserung der Lesbarkeit wurden zum Teil Unterstriche verwendet.



Sind Felder direkt in der Deklaration definiert, wurde kein Typbezeichner vergeben.

Sind Strukturen, Felder oder Enumeratoren an ein spezielles System gebunden, sind in der Regel Präfixe im Namen enthalten:

System	Präfix	Beispiel
MLC	ML	ML_ReadParameterBool
Synax	MS	MS_ReadSingleParameter
Synax + VisualMotion	MSV	MSV_ReadMaxValue
MTX	MT	MT_NcBlk
VisualMotion	MV	MV_Hysteresis
MLD	MX	MX_SetDeviceMode

Tab. 8-2: Präfixe für systemabhängige Typen

8.6.8 Instanzbezeichner

Allgemeines

Von Datentypen können Vervielfältigungen (Instanzen) gebildet werden. Bei Instanzbezeichnern ist Groß-/Kleinschreibung berücksichtigt; sie sind in englischer Sprache verfasst.

Instanzbezeichner komplexer Datentypen

Nachfolgend aufgeführte Präfixe sind für Instanzen komplexer Datentypen verwendet worden. Die Präfixe sind klein und ohne Unterstrich dem Namen hinzugefügt.

Datentyp	Präfix	Instanzbeispiel	Typbeispiel
Funktionsbausteine	Fb	fbJogMode	MT_Jogging
Strukturen	St	stComData	MX_COM_DATA
Felder (Arrays)	Ar	arStateInfo	MV_STATE_INFO
Enumeratoren	En	enDiagData	ML_DIAG_DATA

Tab. 8-3: Instanzbezeichner komplexer Datentypen

Instanzbezeichner einfacher Datentypen

Präfixe für Instanzen einfacher Datentypen

Nachfolgend aufgeführte Präfixe sind für Instanzen einfacher Datentypen verwendet worden. Die Präfixe sind klein und ohne Unterstrich dem Namen hinzugefügt.

Datentyp	Präfix	Beispiel	Speicherbelegung	Datentypbezeichnung	Datentypbeschreibung
BOOL	b	bVar	1 Bit	Bool	bitorientiertes boolsches Format
BYTE	by	byVar	8 Bit	Byte	bitorientiertes Byte-Format
WORD	w	wVar	16 Bit	Word	bitorientiertes Format einfacher Wortlänge
DWORD	dw	dwVar	32 Bit	Double Word	bitorientiertes Format doppelter Wortlänge
LWORD	lw	lwVar	64 Bit	Long Word	bitorientiertes Format vierfacher Wortlänge
SINT	si	siVar	8 Bit	Short Integer	ganzzahliges vorzeichenbehaftetes Format verkürzter Länge

Programmierhinweise

Datentyp	Präfix	Beispiel	Speicherbelegung	Datentypbezeichnung	Datentypbeschreibung
INT	i	iVar	16 Bit	Integer	ganzzahliges vorzeichenbehaftetes Format einfacher Länge
DINT	di	diVar	32 Bit	Double Integer	ganzzahliges vorzeichenbehaftetes Format doppelter Länge
LINT	lii	liVar	64 Bit	Long Integer	ganzzahliges vorzeichenbehaftetes Format vierfacher Länge
USINT	usi	usiVar	8 Bit	UnsignedShort Integer	ganzzahliges nicht vorzeichenbehaftetes Format verkürzter Länge
UINT	ui	uiVar	16 Bit	Unsigned Integer	ganzzahliges nicht vorzeichenbehaftetes Format einfacher Länge
UDINT	udi	udiVar	32 Bit	Unsigned Double Integer	Ganzzahliges nicht vorzeichenbehaftetes Format doppelter Länge
ULINT	uli	uliVar	64 Bit	Unsigned Long Integer	ganzzahliges nicht vorzeichenbehaftetes Format vierfacher Länge
REAL	r	rVar	32 Bit	Real	reelles Zahlenformat einfacher Länge
LREAL	lr	lrVar	64 Bit	Long Real	reelles Zahlenformat doppelter Länge
STRING	str	strVar	8 Bit pro Zeichen	String	Zeichenkette von 1-255 Zeichen (ANSI-Code möglich)
WSTRING	wstr	wstrVar	16 Bit pro Zeichen	Wide String	Zeichenkette von 1-65535 Zeichen (UNI-Code möglich)
TIME	t	tVar	32 Bit	Time	Zeitformat
DATE	d	dVar	32 Bit	Date	Datumsformat
TIME_OF_DAY	tod	todVar	32 Bit	Time Of Day	Uhrzeit-Format
DATE_AND_TIME	dat	datVar	32 Bit	Date And Time	Datum- und Uhrzeit-Format
POINTER TO ???	p???	ptrVar	32 Bit	Pointer To ???	Zeiger / Adresse einer Variablen mit bestimmten Datentyp
POINTER TO DWORD	pdw	pdwVar	32 Bit	Pointer To Double Word	Beispiel: Zeiger auf eine Doppelwort-Variable

Hinweis

Die Datentypen LWORD, LINT, ULINT und WSTRING werden derzeit von IndraLogic nicht unterstützt

???

Datentyp der Variable auf die der Pointer zeigt

Tab. 8-4:

Instanzen einfacher Datentypen

Suffixe für Instanzen einfacher Datentypen

Zusätzlich können zum Zweck der Herkunftskennzeichnung die Instanznamen einfacher Datentypen mit Suffixen versehen sein. Diese sind als Kleinbuchstabe mit einem Unterstrich dem Namen hinzugefügt.

Suffix	Bedeutung	Beispiel
i	Hardware-Eingang	bAxisTravelLimit_i
q	Hardware-Ausgang	bLockDoor_q
gb	Globale Variable	stControlState_gb
m	Lokaler Merker	iComData_m

Tab. 8-5: Beispiele für Suffixe einfacher Datentypen

8.6.9 Definition von Standardschnittstellen an Funktionsbausteinen

Einleitung

Die Mehrzahl der Funktionsbausteine hat einen Eingang zur Aktivierung und einen Ausgang der die fehlerfreie Bearbeitung anzeigt. Zusätzlich wird oft ein Ausgang benötigt, der den Bearbeitungszeitraum kennzeichnet. Zum Anzeigen von Fehlern werden weitere Ausgänge definiert.

Eine einheitliche Namensgebung sowie gleiches Verhalten dieser Standardschnittstellen erhöhen das Verständnis, verkürzen die Einarbeitung und entlasten damit den Support.

Mit Funktionsbausteinen ist es möglich komplexe Aufgaben wiederverwendbar zu kapseln und diese über definierte Schnittstellen anzusprechen. Dabei kann die Bearbeitung entweder zustandsgesteuert oder flankengesteuert erfolgen.

Zustandsgesteuert Wiederholt ein Funktionsbaustein nach dem Einschalten seinen Auftrag immer wieder, wenn er einen definierten Zustand erreicht hat, arbeitet er zustandsgesteuert.

Flankengesteuert Erfüllt ein Funktionsbaustein nach dem Einschalten seine Aufgabe genau einmal, arbeitet er flankengesteuert.

Standard-Eingänge an Funktionsbausteinen

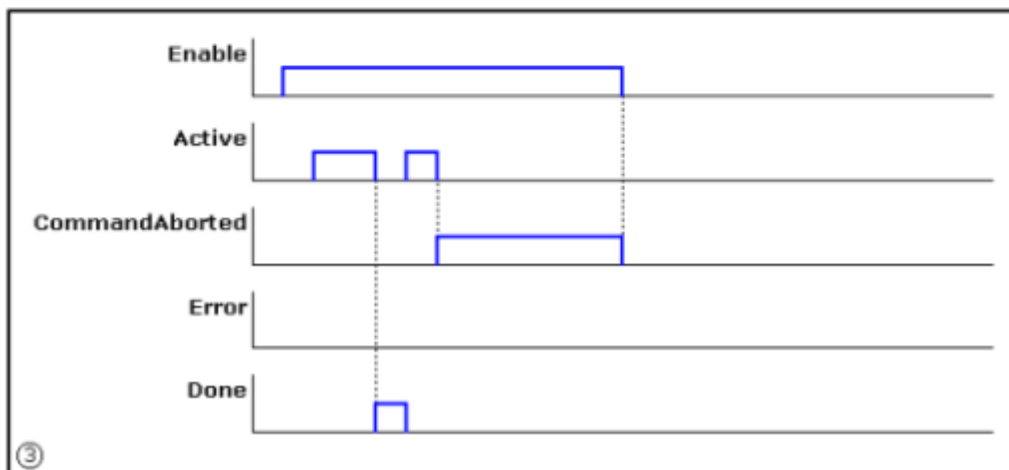
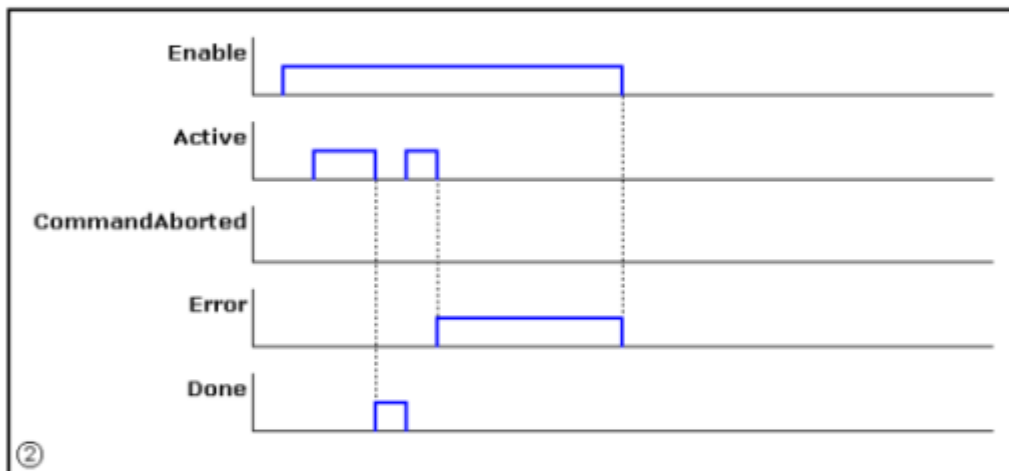
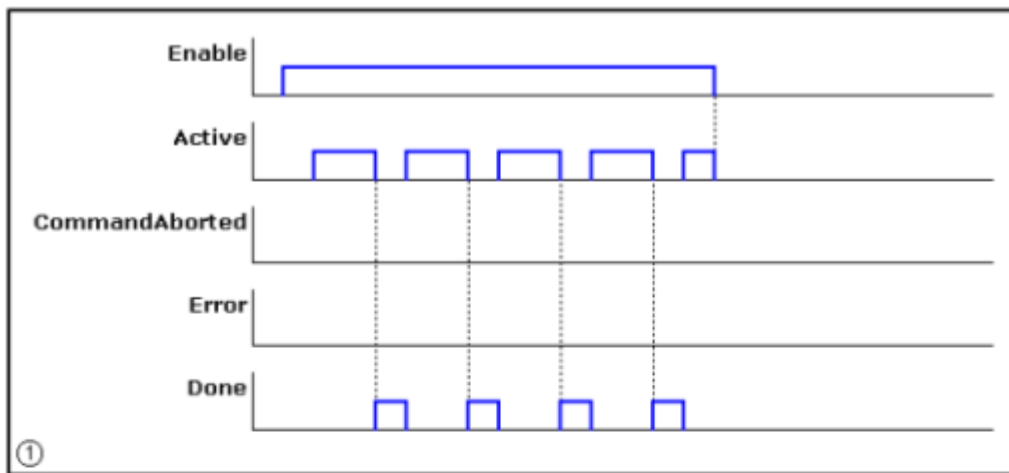
Um nach außen kenntlich zu machen, ob ein Funktionsbaustein zustandsgesteuert oder flankengesteuert arbeitet, werden zwei verschiedene Variablenamen für die Aktivierung des Funktionsbausteins verwendet:

- Der Aktivierungseingang "Enable" soll an Funktionsbausteinen eingesetzt werden, die zustandsgesteuert arbeiten.
- Der Aktivierungseingang "Execute" soll an Funktionsbausteinen eingesetzt werden, die flankengesteuert arbeiten.

Standard-Ausgänge an Funktionsbausteinen

- Dass ein Funktionsbaustein seinen Auftrag erfolgreich ausgeführt hat und eventuell gelieferte Daten gültig sind, wird mit dem Ausgang "Done" signalisiert.
- Jeder Baustein hat ein bestimmtes Ziel, einen Auftrag (z. B. Kommunikation, Regelung, Bewegungssteuerung, usw.). Solange ein Funktionsbaustein an seiner eigentlichen Aufgabe arbeitet, soll dies über den Ausgang "Active" signalisiert werden. Eventuelle Vorverarbeitungen werden nicht mit diesem Ausgang gekennzeichnet.
- Konnte die Bearbeitung eines Auftrages nicht erfolgreich beendet werden, wird dies durch den Ausgang "Error" signalisiert.

Signal-Zeit-Diagramm für zustandsgesteuerte Funktionsbausteine



1

Zustandsgesteuerte Bausteinbearbeitung erfolgreich abgeschlossen

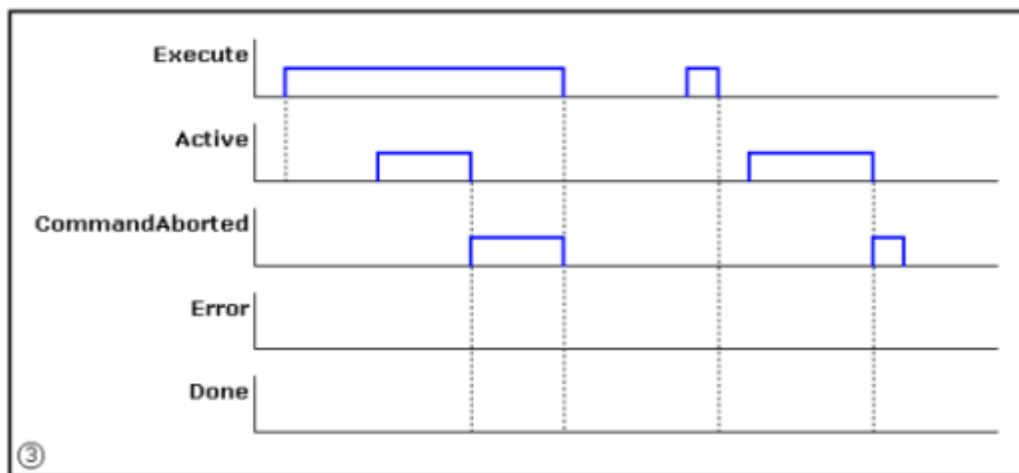
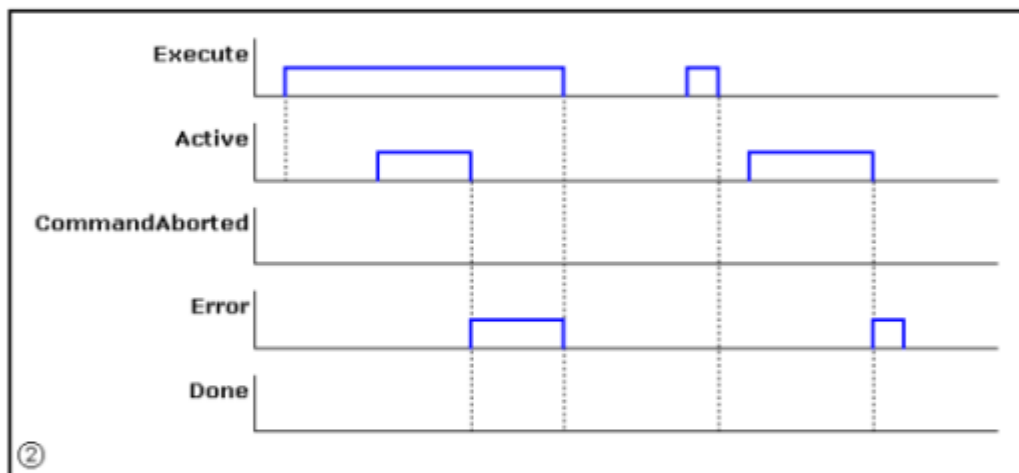
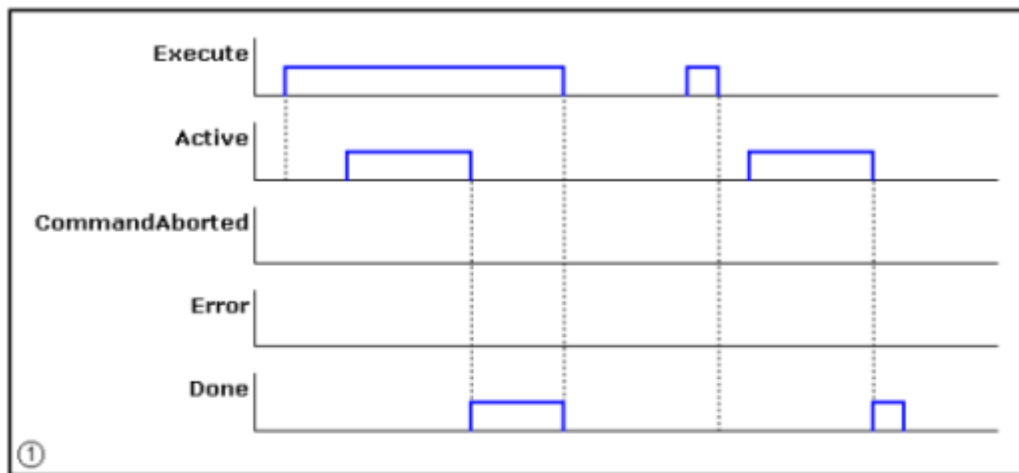
2

Zustandsgesteuerte Bausteinbearbeitung mit Fehler abgebrochen

3*Abb. 8-16:*

Zustandsgesteuerte Bausteinbearbeitung wurde unterbrochen
*Signal-Zeitverhalten von zustandsgesteuerten Funktionsbausteinen
(mit Eingang "Enable"); bei Einzelachsen wird der Ausgang "Active"
nicht gesetzt!*

Signal-Zeit-Diagramm für flankengesteuerte Funktionsbausteine



1

Flankengesteuerte Bausteinbearbeitung erfolgreich abgeschlossen

2

Flankengesteuerte Bausteinbearbeitung mit Fehler abgebrochen

- 3 Flankengesteuerte Bausteinbearbeitung wurde unterbrochen
 Abb. 8-17: *Signal-Zeitverhalten von flankengesteuerten Funktionsbausteinen (mit Eingang "Execute"); bei Einzelachsen wird der Ausgang "Active" nicht gesetzt!*

8.6.10 Fehlerhandling

Das Fehlerhandling ist für Funktionsbausteine von Bosch Rexroth standardisiert; es ist in drei Diagnosestufen eingeteilt:

- Stufe 1: Der Ausgang "Error" zeigt an, dass ein Fehler ansteht, wenn er TRUE ist. "Error" ist vom Typ BOOL.
- Stufe 2: Ein Enumerator ("ErrorID") gibt Grobinformation über den Fehler (grobe Zuordnung)
- Stufe 3: Für Detailinformationen gibt es eine Struktur ("ErrorIdent") in welcher der Fehlerursprung sowie der genaue Fehlercode ausgelesen werden kann.

Siehe auch "[Fehlerbehandlung der IndraMotion MLD](#)"

8.6.11 Versionierung von Bibliotheken

Versionsfunktion IndraMotion MLD bietet die Möglichkeit, interne sowie externe Bibliotheken zu erstellen. Interne Bibliotheken werden vollständig in einer Programmiersprache der **IEC 61131** programmiert, während externe Bibliotheken in der Firmware umgesetzt werden.

Die Tatsache, dass bei einer externen Bibliothek der Programmcode einer POE beim Download über die POE-Bezeichnung gelinkt wird, kann für eine Bindung der POE und somit der Bibliothek an ein bestimmtes IW-Package genutzt werden.

Die Versionierung, die bedingte IW-Package-Bindung und die Führung einer History wird bei Bibliotheken von Bosch Rexroth durch eine standardisierte Versionsfunktion in einer einheitlichen Ordnerstruktur bereitgestellt.

Meldung während des Downloads Wird eine Firmware-Bibliothek in ein Anwenderprojekt eingebunden, so werden während eines Downloads alle in der Firmware-Bibliothek enthaltenen Funktionen / Funktionsbausteine eingebunden. (Die POE-Bindung und somit der Versionsschutz kann unter Umständen nicht auf allen Targets gegeben sein.)

Können eine oder mehrere Funktionen nicht eingebunden werden, so wird der Download mit einer Fehlermeldung abgebrochen.

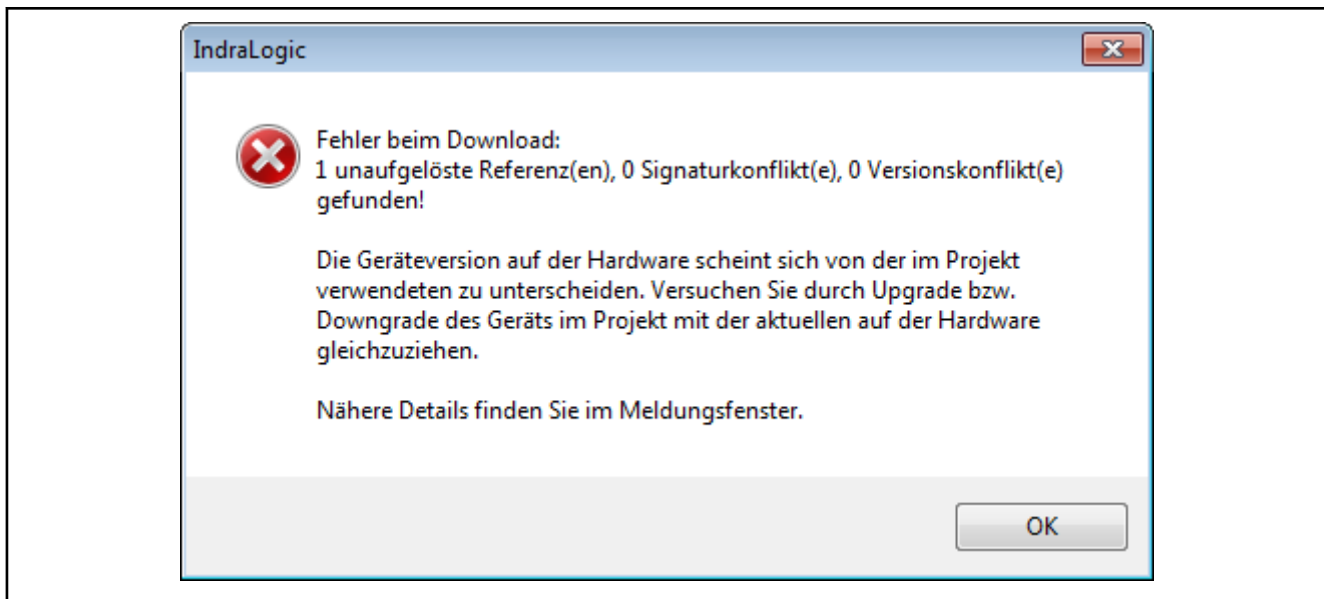


Abb. 8-18: Meldung einer fehlgeschlagenen Versionsprüfung

Die Fehlermeldung enthält eine Auflistung welche Funktionen / Funktionsbausteine aus welcher Bibliothek nicht gebunden werden konnten. Mit Hilfe einer Referenzliste (z. B. aus den Firmware-Release-Notes) kann festgestellt werden, welche Bibliotheken und Funktionen / Funktionsbausteine in welcher Firmware zur Verfügung gestellt werden.

8.6.12 Codeoptimierung

Anforderungen an das Echtzeitverhalten der Applikation

Antriebsregelung und antriebsintegrierte SPS teilen sich die Rechenleistung des Antriebs, wobei die Antriebsregelung einen Großteil der Rechenleistung des Antriebs einnimmt. Um die verfügbare Rechenzeit optimal ausnutzen zu können, ist zunächst zu prüfen, welche Anforderungen an das Echtzeitverhalten der Applikation gestellt werden:

- Welche Funktionalitäten müssen in bestimmten Zeitabständen oder bei bestimmten Ereignissen aufgerufen werden?
- Was geschieht, wenn eine Funktion nicht zum gewünschten Zeitpunkt beendet ist?
- Ist es notwendig eine FKM-synchrone oder CCD-synchrone²⁾ Task einzusetzen?

Tasksystem

Aus den Anforderungen an das Echtzeitverhalten der Applikation ergibt sich ein erster Ansatz an die Taskkonfiguration der Applikation. Generell sollten mehrere Tasks dazu eingesetzt werden, um Funktionen mit großen Laufzeiten von Funktionen, die in Echtzeit laufen müssen, zu trennen. Jede Task benötigt Rechenzeit für die Verwaltung und Steuerung des Tasksystems. Je mehr Tasks konfiguriert sind, desto mehr Rechenzeit muss für die Verwaltung der Tasks aufgebracht werden.

Generell sollte in zeitkritischen Tasks nur der notwendige Code durchlaufen werden. Auf zu kurze Intervalle bei den Tasks sollte verzichtet werden.

Eine Aufteilung in verschiedenen schnelle Tasks ist nur sinnvoll, wenn relativ wenige Aktionen sehr schnell abgearbeitet werden müssen und andere Aktionen in einer langsameren Task laufen (z. B. eine Task mit $t_{\#1ms}$ und eine Task mit $t_{\#10ms}$).

²⁾ CCD-synchrone Task ist nur bei MLD-M möglich

Außerdem ist zu prüfen, ob bei einmaliger Tasküberschreitung bereits ein Watchdog ausgelöst werden muss oder dieser toleranter sein kann.



Alternativ zu einem Multitasksystem können in einer Task zunächst alle schnellen Aktionen durchgeführt werden und dann jeweils nur eine langsamere Aktion (CASE-Anweisung).

Wenn allerdings die langsameren Aktionen mehr Rechenzeit benötigen, ist es notwendig, sie in einer eigenen Task auszuführen.

Stringverarbeitung	Die Verarbeitung von Strings ist relativ langsam. Bei schnellen Taskzyklen möglichst vermeiden.
IEC-Sprachauswahl	Die Programmiersprache "Strukturierter Text" (ST) ist für eine performance-optimierte Programmierung empfehlenswert. Eine übersichtliche Programmierung ist mit "ST" ebenfalls gegeben.
Programmcode	• Bitverarbeitung ist deutlich langsamer als Byteverarbeitung. Dies betrifft adressierte Variablen und Bitzugriffe in Worten (z. B. Eingänge %IX0.0, Ausgänge %QX0.0, Merker oder Bitzugriff). • Division von Integertypen ist deutlich langsamer als REAL-Division.
I/O-Konfiguration	Das I/O-Mapping auf BOOL-Variablen (8 Bit lang) ist deutlich schneller als Bitvariablen (z. B. %IX0.0).
IEC-Konstanten verwenden	Die Verwendung von <code>VAR CONSTANT</code> ist schneller als die Verwendung von konstanten Werten in Form von normalen Variablen. Eine Variablenadressierung entfällt, der Wert wird direkt eingesetzt.
Parameterzugriffe	• Parameterzugriffe nur so häufig wie nötig durchführen. • Stringparameter ggf. in eigener Task ausführen, da sehr viel Rechenzeit benötigt wird. • Zugriffe von Direktvariablen auf Parameter sind deutlich schneller als Funktionen. • Zugriffe von Funktionen auf Parameter sind schneller als Funktionsbausteine. • Zugriffe auf lokale Listenparameter (Axis1) benötigen relativ viel Rechenzeit und sollten entweder in eine langsame Task ausgegliedert oder elementweise mit <code>"MX_fReadParamDINT()"</code> / <code>"WriteParamDINT()"</code> durchgeführt werden.
Funktionen / Funktionsbausteine	• Sollten in einem Taskzyklus mehrmals gleiche Bausteine oder Funktionen genutzt werden, sollten diese möglichst direkt hintereinander aufgerufen werden (Cache-Effekt!). • Funktionsbaustein nur solange aufrufen wie notwendig - speziell zyklische Funktionsbausteine. • Konstante Funktionsbaustein-Parameter können bereits bei der Deklaration belegt werden. • Ausgänge von Funktionsbausteinen können direkt gelesen werden, ohne dass man sie einer Variablen zuweisen muss.
Checkfunktion	Mit der Checkfunktion werden fehlerhafte Zugriffe erkannt und verhindert. Geprüft werden Pointerzugriffe, Arrayzugriffe, Bereichsgrenzen, Division durch 0... Ist der häufig durchlaufene Code ausreichend geprüft, kann diese automatische Prüfung (implizite Checkfunktion) abgeschaltet werden.

Beispiel:

Implizite Checkfunktion abschalten

Hier werden innerhalb der POU keine Laufzeitprüfungen vorgenommen:

```
{attribute 'no check'}
```

```
FUNCTION MyFunc : INT
```

Strukturierte Programmierung

- Die Aufteilung in Unterfunktionen bzw. Unterfunktionsbausteine ist nur performanter, wenn diese Funktionen / Funktionsbausteine mehrfach verwendet werden.
- Über CASE-Anweisung können je nach Status der Funktionalität bzw. der Applikation gezielt unterschiedliche Code-Zweige bearbeitet werden.
- Übergabeparameter / Rückgabewerte an Funktionen und Funktionsbausteine sollten möglichst kompakt sein. Bei größeren Daten global zugreifen oder per VAR_IN_OUT oder REFERENCE übergeben.

8.7 VCS – Version Control System

IndraWorks erlaubt mithilfe des "Version Control System" (VCS) das Arbeiten mit versionierten Projekten, die auf einem Teamserver liegen. An solchen versionierten Projekten können mehrere IndraWorks-Anwender gleichzeitig arbeiten.



Informationen zur Installation und Einrichtung von Teamservern für die Verwendung mit IndraWorks erhalten Sie in der Anwendungsbeschreibung zu IndraWorks Engineering.

Ab IndraWorks 14V14 kann das "Version Control System" auch genutzt werden, um MLD-Projekte zu versionieren und archivieren. Folgende Funktionen können durchgeführt werden:

- Projektteile können exklusiv oder nicht-exklusiv bearbeitet werden
- Teil-Projekte können abgespalten werden
- Lokale Änderungen können koordiniert mit Änderungen im Teamprojekt zusammengeführt werden

Das "Version Control System" bietet weiterhin folgende Möglichkeiten:

- Aufzeichnen aller Änderungen (wer, wann, warum)
- Änderungen können rückgängig gemacht werden
- Projektstände können festgehalten werden
- Unterschiede zwischen Projektständen werden angezeigt
- Alte Projektstände können wiederhergestellt werden

Ein neues Teamprojekt kann angelegt werden, indem aus dem Kontextmenü des Projektknotens der Menüpunkt **Versionsverwaltung ► Teamprojekt erstellen...** aufgerufen wird.



Alle Verzeichnisse und Dateien die mit einem Schloss versehen sind, sind dem "Version Control System" zugeordnet. Bei geschlossenen Schlössern sind die Verzeichnisse und Dateien schreibgeschützt, Verzeichnisse und Dateien mit offenen Schlössern sind beschreibbar.



Weiterführende Informationen zum "Version Control System" (VCS) erhalten Sie in der Anwendungsbeschreibung zu IndraWorks Engineering.

9 Diagnose- und Servicefunktionen

9.1 Übersicht und Einleitung

In diesem Abschnitt werden die Diagnose- und Servicefunktionen beschrieben, die vor allem für die Fehlersuche sowie zur Fehlerbehebung verwendet werden können. Hierbei werden sowohl die Möglichkeiten bei freier SPS-Programmierung, als auch bei der Verwendung von fertigen Technologiefunktionen erläutert.

Generell ist folgende Klassifizierung zu treffen:

- Standard-Diagnosemöglichkeiten des Antriebs
- Diagnose- und Servicefunktionen der MLD
- Allgemeine Servicefunktionen

9.2 Geräte-Editor

9.2.1 Geräte-Editor, Allgemeines

Der Geräte-Editor stellt unter anderem Registerkarten für die Konfiguration eines Gerätes bereit.

Der "**Geräte-Editor**" kann im Projektextplorer über den Kontextmenüeintrag **Öffnen** einer Steuerung [z. B. Antriebsregelgerät mit integrierter SPS (IndraMotion MLD)] geöffnet werden. Bei Verwendung von IndraMotion MLD V14 sind folgende Registerkarten enthalten:

- [Status](#)
- [Dateien](#)
- [Taskaufstellung](#)
- [Log](#)
- [SPS-Einstellungen](#)
- [SPS Shell](#)
- [Information](#)



Die Registerkarten "Log" und "SPS Shell" sind zu- und abschaltbar:

Extras ▶ Optionen ▶ IndraLogic 2G ▶ Allgemeine Einstellungen.

9.2.2 Status

Die Registerkarte "Status" zeigt Statusinformationen des jeweiligen Geräts und des internen Bussystems.



Abb. 9-1: Geräte-Editor: Status

9.2.3 Dateien

Die Registerkarte "Dateien" dient dazu, Dateien zwischen dem PC (Host) und der externen Speicherkarte des Antriebs (IndraLogic-Laufzeitsystem) auszutauschen.

Es ist ausschließlich der Zugriff auf das Verzeichnis "User" der externen Speicherkarte des Antriebs möglich; es können dort Verzeichnisse / Dateien gelöscht werden oder Verzeichnisse angelegt werden, um z. B. zum Projekt gehörige Dateien abzulegen.



Aus einem MLD-Programm heraus, ist der Zugriff auf das Verzeichnis "User" mit Hilfe der Bibliotheken "SysFile"/"SysFileAsync" und "SysDir" möglich.

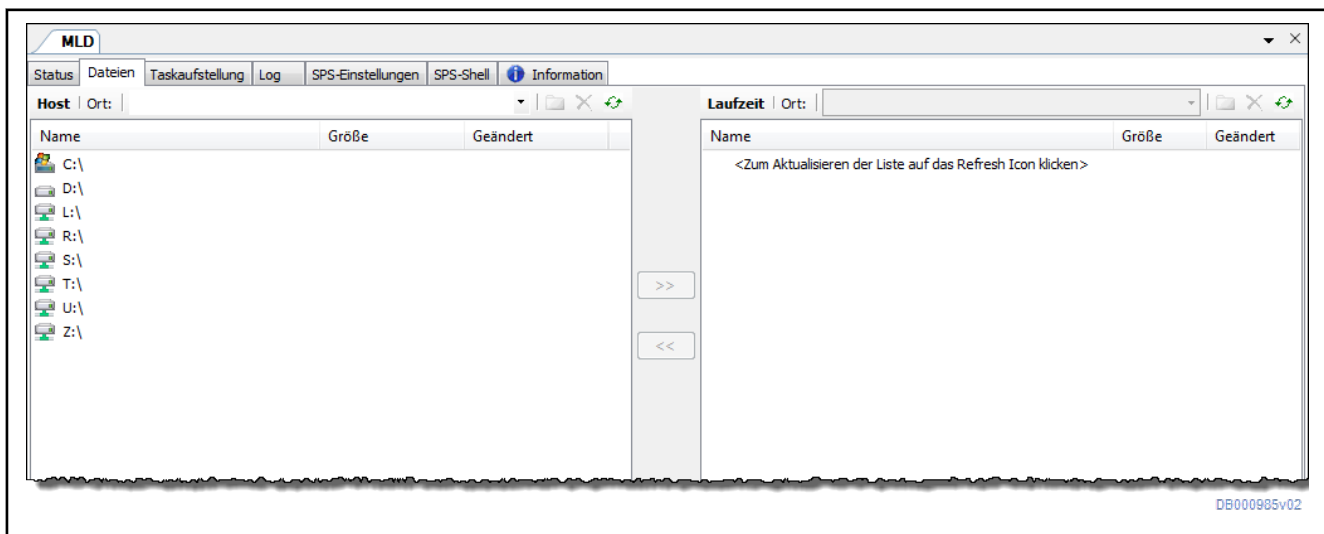


Abb. 9-2: Geräte-Editor: Dateien

9.2.4 Taskaufstellung

Nachdem für die Applikation Code erzeugt wurde, werden in der Registerkarte "Taskaufstellung" die Ein- und Ausgänge mit ihren Zuordnungen zu den definierten Tasks gezeigt.

Die Registerkarte hilft bei der Fehlersuche, weil sie aufzeigt, wo Ein- oder Ausgänge in mehreren Tasks mit unterschiedlichen Prioritäten verwendet werden. Mehrfache Verwendung kann zu undefinierten Werten durch Überschreiben führen.

9.2.5 Log

Die Registerkarte "Log" dient dem Anzeigen des Logbuchs der Steuerung, d. h. dem Anzeigen von Ereignissen, die auf dem Zielsystem aufgezeichnet wurden.



Die Registerkarte ist zu- und abschaltbar über **Extras ▶ Optionen ▶ IndraLogic 2G ▶ Allgemeine Einstellungen ▶ SPS-Logger aktivieren**.

Angezeigt werden:

- Ereignisse beim Start und beim Beenden des Systems (geladene Komponenten mit Version)
- Applikations-Download und Laden des Bootprojekts

- Kundenspezifische Einträge
- Log-Einträge von I/O-Treibern
- Log-Einträge des DataServers

Gewichtung	Zeitstempel	Beschreibung	Komponente
Information	23.01.2014 12:46:37	Application [Application] loaded via [OnlineChange]	CmpApp
Information	23.01.2014 12:42:34	Application [Application] loaded via [Download]	CmpApp
Information	23.01.2014 12:42:34	sercos: No new Configuration.	IoDrvS3Io
Information	23.01.2014 12:42:34	sercos: PAR_ID_CONNECTOR_NAME not found:	IoDrvS3Io
Information	23.01.2014 12:40:50	sercos : SetParamP4 OK	IoDrvS3Io
Information	23.01.2014 12:40:49	sercos : SetParamP3 OK	IoDrvS3Io
Information	23.01.2014 12:40:49	sercos : CheckConfigP3 OK	IoDrvS3Io
Information	23.01.2014 12:40:47	sercos : SetParamP2 OK	IoDrvS3Io
Information	23.01.2014 12:40:47	sercos : ReplaceFunctionGroup OK	IoDrvS3Io
Information	23.01.2014 12:40:47	sercos : CheckIdentityP2 OK	IoDrvS3Io
Information	23.01.2014 12:39:08	sercos PLC Configuration active	IoDrvS3Io
Information	23.01.2014 12:39:06	sercos : LeaveP4 OK	IoDrvS3Io
Information	23.01.2014 12:34:59	sercos : SetParamP4 OK	IoDrvS3Io
Information	23.01.2014 12:34:59	sercos : SetParamP3 OK	IoDrvS3Io

Abb. 9-3: Geräte-Editor: Log

Ein Logbucheintrag wird mit den folgenden Informationen dargestellt:

- **Schwere** (Gewichtung): Für die Schwere des Ereignisses gibt es vier Kategorien: Warnungen, Fehler, Ausnahmefehler, Information. Die Darstellung jeder Kategorie kann ein- oder ausgeblendet werden, indem die entsprechende Schaltfläche in der Leiste oberhalb der Liste betätigt wird. In der Schaltfläche wird jeweils die Anzahl der Log-Einträge der betreffenden Kategorie angezeigt.
- **Zeitstempel**: Datum und Uhrzeit, z. B. "30.06.2012 04:22"
- **Beschreibung**: Beschreibung des Ereignisses, z. B. "Import function failed of <CmpFileTransfer>"
- **Komponente**: Name der betroffenen Komponente.
- **Kom Name** (Komponentenname): Hier kann aus einer Auswahlliste eine einzelne Komponente ausgesucht werden, so dass nur Logbucheinträge, die diese betreffen, angezeigt werden. Die Standardeinstellung ist Alle Komponenten.
- **Logger**: Die Auswahlliste enthält die verfügbaren Aufzeichnungen. Die Standardeinstellung ist der vom Zielsystem vorgegebene "<Default Logger>", derzeit identisch mit "PlcLog" für das IndraLogic-Laufzeitsystem.



- Die Liste der Logbucheinträge kann mit Hilfe der Schaltfläche  aktualisiert werden.
- Der Inhalt der Liste kann in eine XML-Datei exportiert werden. Drücken Sie dazu Schaltfläche , um den Standarddialog zum Speichern einer Datei zu öffnen. Der Dateifilter ist auf "xml-files (*.xml)" eingestellt. Die Liste der Logbucheinträge wird mit dem angegebenen Dateinamen im gewählten Verzeichnis abgelegt. Wenn die Option "Offline-Logging" aktiviert ist, werden auch Aktionen, die sich nicht auf die Verbindung mit der Steuerung beziehen, aufgezeichnet. Dies ist momentan jedoch nur für die Safety-Version des Programmersystems realisiert.

9.2.6 SPS-Einstellungen

Die Registerkarte "SPS-Einstellungen" dient dazu festzulegen, wie sich die SPS im Stoppzustand verhält.

Abb. 9-4: Geräte-Editor, SPS-Einstellungen

Applikation für E/A-Behandlung

Wenn mehrere Applikationen für das Gerät bereitstehen, werden diese hier in einer Auswahlliste angeboten. Zunächst ist hier immer die Standard-Applikation, die mit einem Standardprojekt automatisch angelegt wird, eingetragen.

SPS-Einstellungen:

E/As aktualisieren im Stop: Wenn diese Option aktiviert ist (Default), werden die Werte der Ein- und Ausgangskanäle auch aktualisiert, wenn die SPS in den Stop-Status geht.

Verhalten der Ausgänge bei Stop

Die Auswahlliste bietet folgende Möglichkeiten, wie die Werte der Ausgangskanäle behandelt werden sollen, wenn die Steuerung in den Stop-Status geht:

- **Aktuelle Werte beibehalten:** Die augenblicklichen Werte werden beibehalten.
- **Alle Ausgänge auf Standardwerte setzen:** Die Standardwerte, die aus dem I/O-Abbild resultieren, werden zugewiesen.

- **Programm ausführen:** Die Behandlung der Ausgangswerte kann über ein im Projekt enthaltenes Programm gesteuert werden. Der Name dieses Programms kann hier angegeben werden und es wird dann ausgeführt, wenn die Steuerung in Stop-Status geht. Mit der Schaltfläche "..." kann die Eingabehilfe geöffnet werden, um die Programmauswahl zu erleichtern.

Buszyklus-Optionen

Buszyklus-Task: Die Auswahlliste bietet alle Tasks an, die in der Taskkonfiguration der aktiven Applikation definiert sind (z. B. "MotionTask", "PlcTask" etc.).



Kontrollieren Sie die Daten der Tasks Ihrer Applikation und tragen Sie dann die gewünschte Task ein.

Wählen Sie eine bestimmte Task aus, die den Buszyklus steuern soll, oder wählen Sie die Einstellung "<Unspezifiziert>", wenn die Task mit der kürzesten Zykluszeit, d. h. die schnellste Task, dafür verwendet werden soll.

9.2.7 SPS Shell



Die Funktion "SPS Shell" dient Diagnose- und Debugging-Zwecken. Bei Verwendungsbedarf setzen Sie sich bitte mit dem Kundendienst in Verbindung.

Die Registerkarte ist zu- und abschaltbar über **Extras ▶ Optionen ▶ IndraLogic 2G ▶ Allgemeine Einstellungen ▶ SPS-Logger aktivieren**.

9.2.8 Information

Die Registerkarte "Information" zeigt allgemeine Informationen zu dem Gerät an:

Name, Hersteller, Kategorien, Version, Bestellnummer, Beschreibung,... In der Regel wird auch ein Bild des Gerätes angezeigt.

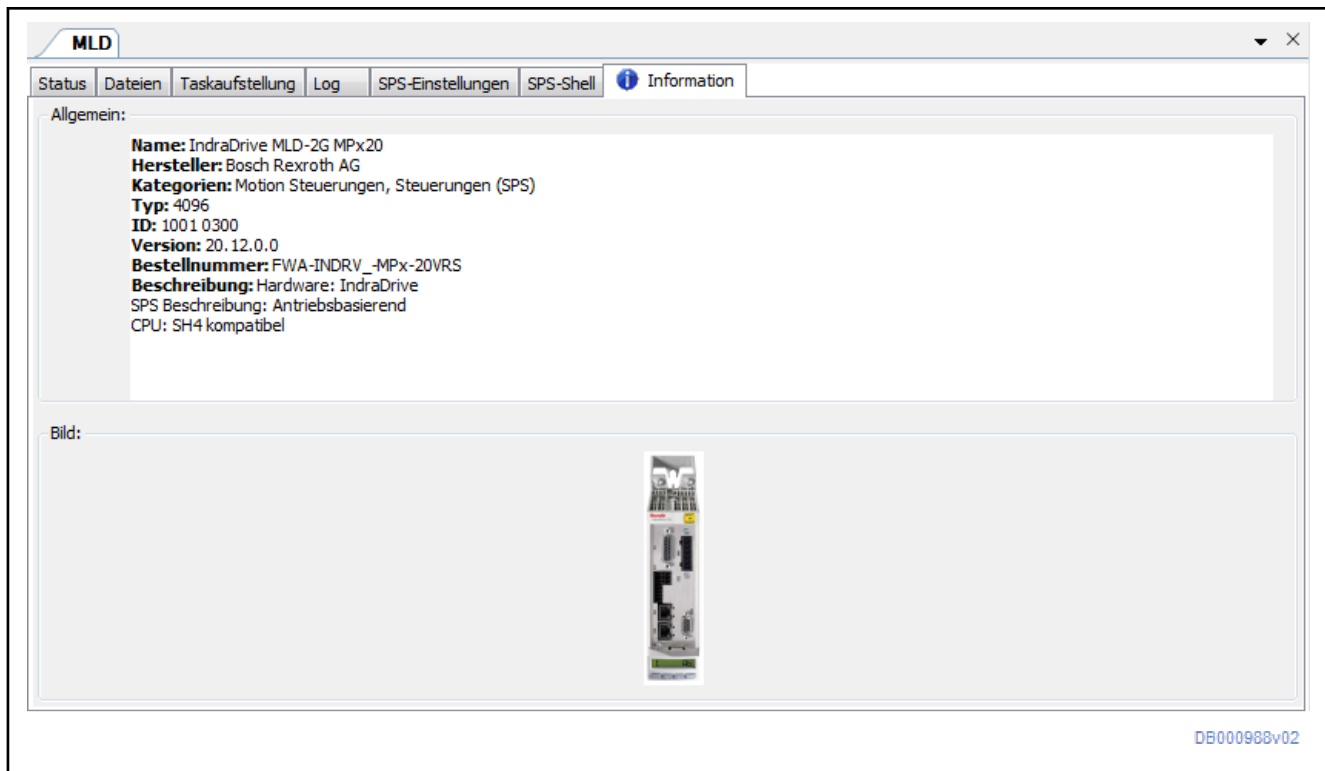


Abb. 9-5: Geräte-Editor, Informationen

9.3 Diagnosefunktionen

9.3.1 Standard-Diagnosemöglichkeiten des Antriebs

Übersicht

Zur Diagnose des Antriebsverhaltens sowie der integrierten SPS stehen im Antrieb die folgenden Funktionen zur Verfügung:

- **Oszilloskopfunktion**

Die Oszilloskopfunktion kann zum Aufzeichnen von antriebsinternen und externen Zustandsgrößen (Parameterinhalten) verwendet werden.

Sie kann sowohl bei der Erstinbetriebnahme als auch zur Fehlersuche effektiv eingesetzt werden. Ihre Funktionalität ist mit der eines 4-Kanal-Oszilloskops vergleichbar.

Die Oszilloskopfunktion unterstützt verschiedene Messmodi, in denen Sie Messungen durchführen können. Es wird zwischen "Single Shot" und "Trending" unterschieden.

Bei "Single Shot" werden die aufgezeichneten Messwerte **im Anschluss an die Messung** zum Oszilloskop übertragen.

Beim Messmodus "Trending" handelt es sich um die klassische Oszilloskop-Betriebsart. In diesem Modus werden die vom Gerät aufgenommenen Messwerte **kontinuierlich** zum Oszilloskop übertragen.

Weiterführende Dokumentationen zur Oszilloskopfunktion:

- Firmware-Funktionsbeschreibung: "Oszilloskopfunktion"
- Online-Hilfe zu IndraWorks: "Oszilloskop - Funktion"
- oder -

gedruckte Ausgabe der Anwendungsbeschreibung zu "Rexroth IndraWorks" [Dokumentations-Type "DOK-IWORKS-ENGINE*V_{xx}-APRS" ("xx" steht für die jeweilige IndraWorks-Version, z. B. "13")]

Außerdem ist mit der Oszilloskopfunktion die Aufzeichnung von SPS-Variablen eines MLD-Anwenderprogramms möglich (siehe "SPS-Variablen mit der Oszilloskopfunktion aufzeichnen").

- **Analogausgabe**

Die Antriebsfunktion "Analoge Ausgänge" ermöglicht die Verwendung analoger Signalwerte zur Inbetriebnahme und Optimierung von Antrieben in Verbindung mit geeigneten Messgeräten (z. B. Oszilloskop, Multimeter) sowie zur Visualisierung der Inhalte von Antriebsparametern (siehe Firmware-Funktionsbeschreibung "Analoge Ausgänge").

- **Patchfunktion**

Die Patchfunktion ermöglicht das Lesen von internen Speicherstellen und von SPS-Variablen (siehe Firmware-Funktionsbeschreibung "Patchfunktion").

- **Fehlerspeicher**

Im Antrieb werden alle aufgetretenen Fehler zusammen mit dem zugehörigen Betriebsstundenzählerstand in einem Fehlerspeicher auf dem Steuerteil erfasst (siehe Firmware-Funktionsbeschreibung "Fehlerspeicher (Leistungs- und Steuerteil)").

9.3.2 Diagnosefunktionen der MLD

Übersicht

In Verbindung mit Rexroth IndraMotion MLD stehen noch weitere SPS-spezifische Diagnosefunktionen zur Verfügung (die detaillierten Beschreibungen der Funktionen können dem jeweiligen Kapitel der Dokumentation "SPS-Programmentwicklung mit Rexroth IndraLogic 1.0" entnommen werden):

- **Ablaufkontrolle**

Ist "Ablaufkontrolle" aktiviert, werden Programmteile markiert, die während des letzten Steuerungszyklus ausgeführt wurden.

- **Traceaufzeichnung**

Die Traceaufzeichnung erlaubt es das zeitliche Verhalten von SPS-Variablen genau zu verfolgen.

- **Watch- und Rezepturverwalter**

Mit dem Watch-/Rezepturverwalter können Variablen zusammengefasst geladen und gespeichert werden.

- **Visualisierung**

Es besteht die Möglichkeit SPS-Variablen grafisch darzustellen oder über die IndraLogic-Oberfläche zu verändern.

- **Überwachungsfunktionen**

In Rexroth IndraMotion MLD werden sowohl die Laufzeit als auch die verfügbaren Ressourcen überwacht.

- **Simulation**

Mit Hilfe der Simulation kann zum Beispiel die SPS-Logik getestet werden.

- **Debugging / Fehlersuche** (Breakpoints, Watch, Einzelschritt, Schreiben, Forcen)

Ablaufkontrolle

Ist "Ablaufkontrolle" aktiviert, werden Programmteile markiert, die während des letzten Steuerungszyklus ausgeführt wurden.



Bei der Verwendung der Ablaufkontrolle ist darauf zu achten, dass die Ablaufkontrolle zusätzlichen SPS-Code erzeugt, der die Verarbeitungsgeschwindigkeit massiv beeinflusst.

Traceaufzeichnung

Funktion Die Traceaufzeichnung erlaubt es, das zeitliche Verhalten von SPS-Variablen genau zu verfolgen. (Direktvariablen werden nicht aufgezeichnet, diese sollten ggf. in einen Merker kopiert werden, der dann aufgezeichnet werden kann.).

Verwendung Bei der Verwendung sind folgende Punkte zu beachten:

- Ist die gewählte Abtastrate in der Trace-Konfiguration kleiner als das SPS-Task-Intervall, dann passt die Skalierung der X-Achse nicht zu den dargestellten Daten!
- Sinnvoll ist eine Abtastrate von "0" (Vorbelegung=1 Abtastvorgang pro SPS-Zyklus) bzw. ganzzahlige Vielfache der Task-Zykluszeit.



Bei Abtastrate "0" erfolgt 1 Abtastvorgang pro SPS-Zyklus! Sinnvolle minimale Abtastrate = SPS-Intervall [ms].

Watch- und Rezepturverwalter

Mit Hilfe des Watch- und Rezepturverwalters können die Werte von ausgesuchten Variablen angezeigt werden. Außerdem ist es möglich, Variablen mit bestimmten Werten vorzubelegen und auf einmal in die Steuerung zu übertragen (siehe auch Dokumentation "SPS-Programmentwicklung mit Rexroth IndraLogic 1.0").

Visualisierung

Während der Inbetriebnahme ist die integrierte Visualisierung gut geeignet, um SPS-Variablen grafisch darzustellen oder über die IndraLogic-Oberfläche zu verändern (z. B. wenn noch nicht alle Eingänge verdrahtet sind). Zur Veranschaulichung von Projektvariablen stellt IndraLogic die Möglichkeit einer Visualisierung zur Verfügung; so lassen sich komfortable Bedien- und Anzeigedialoge für SPS-Projekte erzeugen (siehe auch Dokumentation "SPS-Programmentwicklung mit Rexroth IndraLogic 1.0").

Überwachungsfunktionen

In Rexroth IndraMotion MLD werden sowohl die Laufzeit als auch die verfügbaren Ressourcen überwacht, um unerwünschte Überschreitungen zu erkennen.

Im Falle eines unerlaubten Zugriffs wird eine SPS-Exception erzeugt. Diese Fehler führen zu einem Antriebsfehler F6010 mit Stillsetzung der Achse (siehe ["Fehlerbehandlung der IndraMotion MLD"](#)).

Simulation

Wenn im Simulationsmodus gearbeitet wird, wird keine Verbindung zum Antriebsregelgerät aufgebaut. Beim Einloggen wird die SPS (teilweise) auf dem PC simuliert. Es gibt dann keine Echtzeit und keine Firmware-Funktionsbausteine. Mit Hilfe der Simulation kann zum Beispiel die SPS-Logik getestet werden.

Debugging / Fehlersuche

Für die Fehlersuche und den Test der erstellten SPS-Programme ist in IndraLogic ein Debugger auf Quellcodebasis integriert. Damit ist es möglich, Breakpoints zu setzen oder die Tests im Einzelschritt durchzuführen. Variablen können angezeigt oder auch überschrieben werden.



Sinnvoll ist in jeder Task ein Zähler, so kann einfach überwacht werden, ob der Code durchlaufen wird. Außerdem ist ersichtlich wie oft und wie schnell eine Task läuft.

Folgende Möglichkeiten werden bei der Fehlersuche geboten:

- **FlowControl / Ablaufkontrolle** dient zur Anzeige welche Programmteile durchlaufen werden. "FlowControl" verlangsamt das SPS-Programm stark!
- **Forcelisten** werden verwendet, um Variablen dauerhaft zu überschreiben (z.B. simulieren eines Hardwareschalters). Forcelisten werden bei "Logout", "Reset" und "Download" gelöscht.

Nachverfolgung von SPS-Exceptions

Allgemeines

Bei SPS-Exceptions wie z. B. Division durch Null oder Bereichsüberschreitung ist es vorteilhaft, wenn die entsprechende Stelle im SPS-Quellcode schnell gefunden werden kann.

Fehlersuche nach der Inbetriebnahme

Nach der Inbetriebnahme wird bei einer SPS-Exception der Fehler "F6010 SPS-Laufzeitfehler" generiert. Die Programmieroberfläche springt in einem solchen Fall zu dem fehlerhaften Programmcode und hinterlegt ihn farbig. In P-0-1365 sind Informationen zu der Fehlerursache und zu der betroffenen POU (Programm-Organisationseinheit) hinterlegt (diese Informationen werden auch in IndraWorks in dem Fenster "Meldungen" angezeigt).

Diagnose- und Servicefunktionen

The screenshot shows the IndraLogic software interface. On the left is the 'Projektexplorer' (Project Explorer) showing a project structure with folders like 'Führungskommunikation', 'Leistungsversorgung', 'Axis_1 [5] default', 'MLD', 'Konfiguration', 'Anzeigeformat Register Ax', 'Anzeigeformat Register Gx', 'Projektinfo', 'AxisData', and 'Logic'. The 'Logic' folder is expanded, showing 'Application (STOP)', 'AxisInterface', 'Type', 'Visualizations', 'Global_Axis_Interface', 'TE_AxisInterfaceType', 'TE_MX_AxisInitAllAxis', 'TE_MX_AxisInterface', 'Version_AxisInterface', '_Readme (PRG)', 'UserConstGlobal', 'UserVarGlobal', 'Bibliotheksverwalter', 'PlcProg (PRG)', 'Symbolkonfiguration', 'Taskkonfiguration', 'Visualisation Manager', and 'Parameter_IO (Parameter I/O)'. The 'Parameter_IO' folder is expanded, showing 'Lokale I/O', 'Sercos (CCD)', 'Axis_2 [2] Drive1', and 'Cam Pool'.

At the top, a table lists variables and their values:

Ausdruck	Datentyp	Wert	Vorbereiteter Wert	Kommentar
bVM_Enable	BOOL	FALSE		
rVM_Acc	REAL	10		
rVM_Dec	REAL	10		
rVM_Velocity	REAL	0		
bVM_VelModeAdk	BOOL	FALSE		
bVM_Standstill	BOOL	FALSE		
rVM_actVelocity	REAL	0		
i	INT	0		

The main window displays a ladder logic program for 'MLD.Application.PlcProg'. The program includes comments like 'Call all required programs --> Imc' and 'Example for axis-access with'. It contains several conditional statements and variable assignments. A parameter editor window is open on the right, showing the 'Axis_1 [5] default' parameter with a status of 'OK' and a value of 'SPS Fehlermeldung'.

Abb. 9-6: Anzeige einer SPS-Exception nach der Inbetriebnahme

Vorgehen nach einer Exception ohne IndraLogic am Antrieb

Der Parameter P-0-1365 enthält die Adressinformation (kann über beliebigen Parameterzugriff gelesen werden).

Programm:

```
getp P-0-01365
SPS Fehlermeldung 126 Chars
= 'Id:000007E7 Hex; Time:000027BF s; Par:00000000 Text:WORD zero division; Pou:215; Ofs:232; Task = MyTask1;
```

Die POU-Nummer gibt einen Rückschluss, in welcher POU der Fehler aufgetreten ist. Wird diese POU von mehreren Tasks aufgerufen, so kann man die betroffene Task noch dazu auswerten. Der Offset ("Ofs") kann noch zum Finden der Programmzeile herangezogen werden. Den entsprechenden Funktionsbaustein kann man in IndraLogic mit Hilfe der zweiten Zahl in den Klammern an den POU's wiederfinden.

Um die Codezeile zu finden, öffnet man die Übersetzungsdatei <Projektname>.bpl und sucht die Breakpointliste des betroffenen Funktionsbausteins. Dabei sucht man zunächst den Funktionsbaustein anhand seines Namens oder seiner Nummer (in den eckigen Klammern). In der Liste des Funktionsbausteins sind dann die Zeilennummern mit Adressoffset und POU-Nummer aufgelistet. Man wandelt den Offset (hier 232) in hexadezimale Schreibweise (hier 00E8) um und sucht die Zeile mit dem gleichen bzw. nächst größeren Offset (hier Zeile 10). Mit dieser Zeilennummer findet man die Stelle dann in IndraLogic wieder.

ZeroDiff.bpl (Übersetzungsdatei):

Programm:

MYFB1 (0) (201): 0010-0134

```

Breakpoints:
0001 0010[215]
0002 001C[215]
0003 0028[215]
0004 0034[215]
0005 0040[215]
      0018[217]
0006 0064[215]
      0018[217]
0007 0092[215]
      0018[217]
0008 00BE[215]
0009 00CA[215]
0010 0124[215]
0012 0134[215]

```

-> man erkennt die betroffenen Zeilennummern 9 bzw. 10

9.4 Servicefunktionen

9.4.1 Servicefunktionen der SPS (IndraMotion MLD)

Servicefunktionen in IndraWorks (IndraLogic)

IndraWorks bzw. IndraLogic stellt folgende Servicefunktionen zur Verfügung:

- **"Download" / "Online-Change" / "Bootprojekt laden"**
siehe ["IndraLogic-Projekt in den Antrieb laden"](#)



Vor dem Verlassen der Anlage ist darauf zu achten, dass das Bootprojekt aktualisiert wird, damit nach einem Neustart kein veraltetes Projekt geladen wird.

- **"Alles bereinigen" / "Alles übersetzen"**
"Alles bereinigen" löscht Informationen des letzten Downloads und des letzten Übersetzungsvorgangs.
Mit "Alles übersetzen" wird im Gegensatz zum (inkrementellen) "Übersetzen" das Projekt komplett neu kompiliert. Dabei wird allerdings die Download-Information nicht verworfen, wie es beim Befehl "Alles bereinigen" der Fall ist. Beachten Sie die Möglichkeit, Objekte vom Übersetzen auszuschließen.
- **Reset**
 - "Reset" [entspricht "Reset (warm)"]: Programm rücksetzen; Retain-Variablen behalten ihren Wert
 - "Reset (Kalt)": Programm rücksetzen; Retain-Variablen werden initialisiert
 - "Reset (Ursprung)": Programm und Bootprojekt werden gelöscht

Stoppen der IndraMotion MLD (Notfallmechanismus)

Motivation Da beim Hochfahren des Antriebs automatisch ein Bootprojekt geladen wird und je nach Konfiguration auch startet, wurde ein Mechanismus implementiert, mit dem das Starten von SPS-Programmen verhindert werden kann.

Verwendungshinweise/ Ablauf Um das Starten von SPS-Programmen beim Hochfahren des Antriebs zu verhindern, ist wie folgt vorzugehen:

- In der Bootphase müssen, bevor die Anzeige Boot 2.9 auf dem Display des Bedienfelds angezeigt wird, an dem Bedienfeld gleichzeitig die beiden Tasten "ESC" und "ENTER" gedrückt und gehalten werden.

- Das automatische Starten eines SPS-Bootprojekts wird verhindert und es erscheint die Anzeige "PLC ?".
- Durch Betätigen der Pfeiltasten (Pfeil nach unten bzw. Pfeil nach oben) wechselt die Anzeige zwischen "Run PLC" und "Stop PLC".
Wird "Stop PLC" mit "ENTER" bestätigt, wird kein Projekt geladen; wird "Run PLC" mit "ENTER" bestätigt, wird ein evtl. gespeichertes Projekt gestartet oder geladen.

9.4.2 Firmware-Tausch

Übersicht und Einleitung

Bei einem Firmware-Tausch wird zwischen Firmware-Release-Update und Firmware-Versions-Upgrade unterschieden.

Folgende Punkte sind bei einem Firmware-Tausch zu beachten:

- Bei Firmware-Release-Updates wird die Geräteversion ggf. aktualisiert; sie enthält auch die Bibliotheken alter Releases
- Nicht jede neue Firmware enthält neue oder geänderte Bibliotheken. Liegen keine neuen oder geänderten Bibliotheken vor, so wird auch kein neues IndraWorks-Package bereitgestellt.
- Eine neue Firmware-Version oder eine andere Ausprägung bringt ein neues IndraWorks-Package mit (Beispiele: MPB17→MPC17, MPB17→MPB18)

Firmware-Release-Update

Allgemein

Bei einem Firmware-Release-Update wird eine Firmware der gleichen Version auf einen neuen Releasestand gebracht (z.B. MPB17V08 auf MPB17V10).

Die generelle Vorgehensweise zur Durchführung eines Firmware-Release-Updates wird in der Firmware-Funktionsbeschreibung in dem Kapitel "Firmwaretausch" erläutert.



Nachfolgend werden nur die MLD-spezifischen Verhaltens- und Vorgehensweisen beschrieben.

Verwendung und Inhalt der Firmware Release Notes

Die Dokumentation "Firmware Release Notes" gibt Aufschluss über Firmware-Versionen und deren Kompatibilität.

Die für Rexroth IndraMotion MLD relevanten Fehlerbehebungen befinden sich in dem Unterkapitel "Drive Firmware / Drive PLC" der Firmware Release Notes für die jeweilige Firmware-Version (z. B. MP*18VRS).

Verwendungshinweise

Hierbei sind folgende Punkte zu beachten:

- Bei einem Firmware-Release-Update bleiben alle Parameter erhalten. D. h. nach dem Firmware-Tausch ist das Antriebsregelgerät ohne weitere Maßnahmen wieder einsatzbereit.
- Ein Firmware-Release-Update enthält Fehlerbehebungen oder kompatible Änderungen bzw. Funktionserweiterungen.
- Generell sind die SPS-Programme, die für eine Firmware übersetzt wurden, auch auf einem höheren Firmware-Release lauffähig. Das Projekt selbst ist in den Parametern abgespeichert und bleibt somit erhalten.
- Projekte können unter Verwendung der "alten" passenden Bibliotheken auch für ein altes Release erstellt werden.

Kompatibilität innerhalb einer Version	• Um neu hinzugekommene Funktionen (kompatible Funktionserweiterungen) nutzen zu können, wird ein IndraWorks-Package zur Verfügung gestellt, das im Wesentlichen die neuen Bibliotheken enthält. • IndraWorks-Packages können mit IndraWorks installiert und verwaltet werden (siehe "Geräteupdate mit IndraWorks" auf Seite 310). Mit dem Breitenverkauf einer Firmware-Version wird das IndraWorks-Package mit seinen Bibliotheken für diese Version mit IndraWorks MLD zur Verfügung gestellt. Mit IndraWorks MLD werden die SPS-Projekte automatisch so angelegt, dass sie zur verwendeten Firmware kompatibel sind. Rexroth IndraMotion MLD bietet auch eine Versionsverwaltung; d. h. es werden auch hinsichtlich der Kompatibilität Unterscheidungen getroffen: • Binäre Aufwärtskompatibilität: Sobald sich die Firmware im Breitenverkauf befindet, können die SPS-Programme, die mit einem älteren Release erzeugt wurden, unverändert weiter eingesetzt werden; sie sind binär aufwärtskompatibel. • Abwärtskompatibilität: Innerhalb einer Version sind übersetzte Projekte in der Regel auch binär abwärtskompatibel. Damit ist es möglich ein SPS-Projekt auch auf eine ältere Release der Firmware zu laden.
IndraWorks-Package	 Alle Inkompatibilitäten werden geprüft und abgefangen. Das IndraWorks-Package enthält Informationen zur Geräteversion und allen benötigten Bibliotheken. • Der Target-Name enthält die Information über Gerät, Ausprägung und Version (Beispiel: "MldStdDevice_V18.5.4.0.iwpackage"). • Nicht jede neue Firmware enthält auch Änderungen der Bibliotheken, so dass nicht immer ein neues IndraWorks-Package vorliegt.  Soll ein vorhandenes Projekt auf eine andere Firmware umgestellt werden, ist dies mit IndraWorks möglich.
Firmware-Konflikt	Falls ein Projekt nicht zur Firmware-Release passt, wird eine entsprechende Meldung ausgegeben. Das kann in folgenden Fällen passieren: • Wenn eine alte Firmware mit einem neuen IndraWorks-Package bearbeitet werden soll, dann wird eine Inkompatibilität der Bibliotheken abgefangen und führt zu einer Meldung beim Download bzw. verhindert das automatische Laden des Programms. Letzteres führt zu einem F6010. • Eine interne Überprüfung erkennt den Konflikt, es wird eine Fehlermeldung ausgegeben.
Abhilfemaßnahme	Zur Lösung des Problems sind folgende Schritte erforderlich: • Prüfen der Antriebsfirmware (S-0-0030). • IndraWorks-Package installieren (siehe "Geräteupdate mit IndraWorks" auf Seite 310). • Danach kann das Projekt in den Antrieb geladen werden.

9.4.3 Firmware-Versions-Upgrade

Allgemein

Firmware-Versions-Upgrade	Wenn an einem Antriebsregelgerät ein Firmware-Tausch mit einer Firmware auf einer neuen Version durchgeführt wird, dann spricht man von einem

Firmware-Versions-Upgrade (z. B. FWA-INDRV*-MPB-17V12-D5 soll durch FWA-INDRV*-MPB-18V06-D5 ersetzt werden).

Die generelle Vorgehensweise zur Durchführung eines Firmware-Version-Upgrades wird in der Dokumentation "Firmware Release Notes" und der Firmware-Funktionsbeschreibung in dem Kapitel "Firmwaretausch" erläutert.



Nachfolgend werden nur die MLD-spezifischen Verhaltens- und Vorgehensweisen beschrieben.

Verwendung und Inhalt der Firmware Version Notes

Die Dokumentation "Firmware Version Notes" gibt Aufschluss über Firmware-Versionen und deren Kompatibilität bzw. Unterschiede ggü. der Vorgängerversion.

Die für Rexroth IndraMotion MLD relevanten Änderungen befinden sich in den Unterkapiteln "Applikations- und Anwendungshinweise"- "Inkompatible neue Funktionen und Funktionsänderungen"- "Optionale Gerätefunktionen" bzw. "Applikations- und Anwendungshinweise"- "Kompatible neue Funktionen und Funktionsänderungen"- "Optionale Gerätefunktionen" der Firmware Version Notes für die jeweilige Firmware-Version.

Kompatibilität zwischen Versionen



Ein Firmware-Versions-Upgrade enthält ein neues IndraWorks-Package und neue oder ggf. inkompatible Bausteine. Daher muss ein vorhandenes Projekt vor dem Download neu übersetzt werden. In einigen Fällen sind Anpassungen am SPS-Projekt notwendig!

Verwendungshinweise

Folgende Punkte sind bei einem Firmware-Tausch zu beachten:

- Eine neue Firmware-Version oder andere Ausprägung bringt immer ein neues Package (MLD-2G) bzw. Target (MLD-1G) mit (Beispiel: MPH05→MPH07).
- Ab Breitenverkauf sind die Packages einer Firmware-Version abwärts-kompatibel, so dass z. B. ein Package für MPx18V12-Projekte auch für MPx18V10-Projekte gültig ist. Im Prototypenstatus einer Firmware ist dies nicht zwingend gültig!
- Unter Umständen kann es nötig sein, mehrere Packages zu installieren, wenn Projekte für verschiedene Firmware-Versionen erstellt werden sollen.

Geräteupdate mit IndraWorks

Bei MLD-2G gibt es unterschiedliche Releases der Firmware und von den Geräten mit ihren Bibliotheken.

In der Regel kann mit einer Firmware die neueste Version des Geräts betrieben werden; es müssen keine Einstellungen vorgenommen werden.

Wenn bei einem Download des SPS-Programms von IndraWorks eine Fehlermeldung angezeigt wird, in der auf ein nicht aktuelles Gerät hingewiesen wird, dann muss die Geräteversion umgestellt werden; ggf. ist auch eine Nachinstallation einer Geräteversion (IW-Package) notwendig.

Geräteversion umstellen

1. Rufen Sie im Projektextplorer am Knoten **MLD** das Kontextmenü auf und wählen Sie **IndraLogic Geräteversion ändern** aus.

Der Dialog "IndraLogic Geräteversion ändern" wird aufgerufen.

- Wählen Sie über die Klappliste von "Neue Geräteversion" die neue Geräteversion aus.

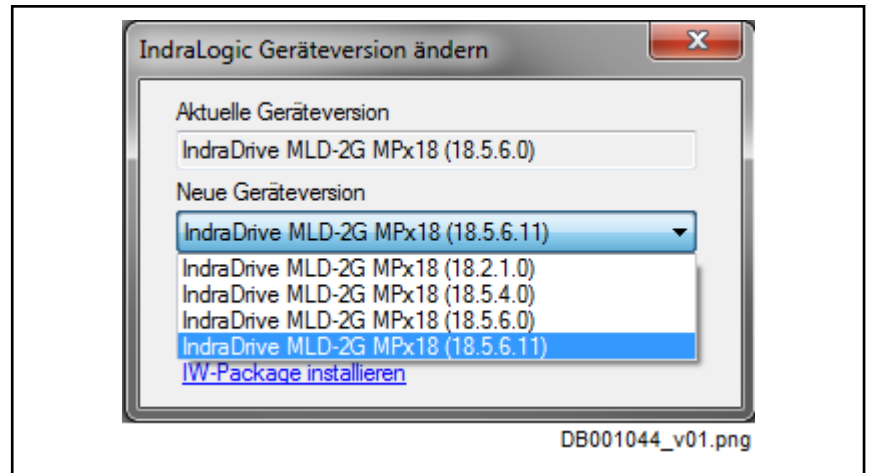


Abb. 9-7: Geräteversion auswählen

- Betätigen Sie die Schaltfläche **Geräteversion ändern**.
- Nach dem Umstellen der Geräteversion sollte das Projekt bereinigt werden:
Im Hauptmenü **Erstellen** ► **Alles bereinigen** ausführen.
- Das bereinigte Projekt muss neu übersetzt werden:
Im Hauptmenü **Erstellen** ► **Alles übersetzen** ausführen.



Das bereinigte und übersetzte Projekt wird beim nächsten "Einloggen" auf die Steuerung geladen.

In der Regel enthält eine IndraWorks-Installation alle notwendigen Geräteversionen für die Projektierung der IndraMotion MLD; d. h. in der Klappliste "Neue Geräteversion" des Dialogs "IndraLogic Geräteversion ändern" ist die Geräteversion enthalten.

Falls ein Gerät nicht vorhanden ist, kann es über die Installation eines IW-Package nachinstalliert werden. Das aktuelle IW-Package wird über die Installation einer neuen IndraWorks-Version bereitgestellt. In Sonderfällen (Fehlerbehebung in einer Version), kann das IW-Package über den Service bezogen werden.

Geräteversion nachinstallieren

Eine neue Geräteversion kann durch die Installation eines so genannten IW-Package hinzugefügt werden.

- Rufen Sie im Projektextplorer am Knoten **MLD** das Kontextmenü auf und wählen Sie **IndraLogic Geräteversion ändern** aus.
Der Dialog "IndraLogic Geräteversion ändern" wird aufgerufen.
- Klicken Sie auf den Link "IW-Package installieren":

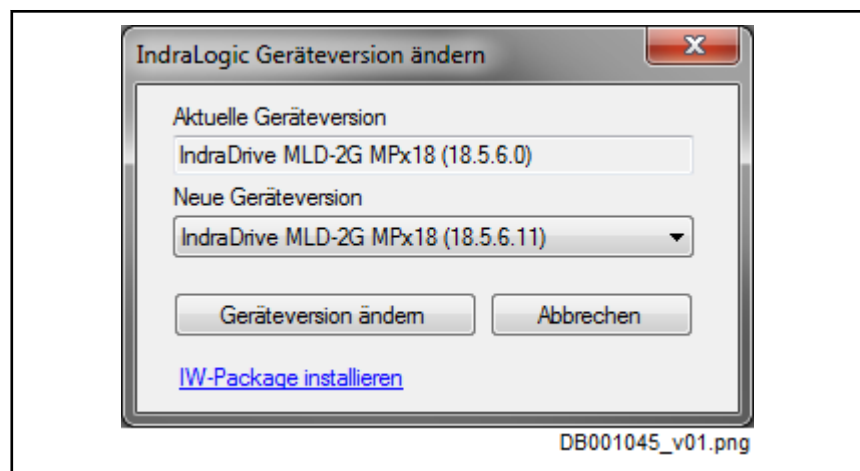


Abb. 9-8: Geräteversion nachinstallieren

Der Windows-Dialog "Datei öffnen" wird geöffnet.

3. Wählen Sie das zu installierende Package aus. (Es dürfen nur Dateien mit der Endung ".iwpacage" ausgewählt werden!)



Die Installation kann einige Zeit in Anspruch nehmen.

Nach der Installation steht die neue Geräteversion in der Klappliste zur Auswahl zu Verfügung.

4. Wählen Sie über die Klappliste die neue Geräteversion aus.
5. Betätigen Sie die Schaltfläche **Geräteversion ändern**.
6. Nach dem Umstellen der Geräteversion sollte das Projekt bereinigt werden:
Im Hauptmenü **Erstellen ► Alles bereinigen** ausführen.
7. Das bereinigte Projekt muss neu übersetzt werden:
Im Hauptmenü **Erstellen ► Alles übersetzen** ausführen.



Das bereinigte und übersetzte Projekt wird beim nächsten "Einloggen" auf die Steuerung geladen.

10 Projektkonvertierung

10.1 Einleitung

In diesem Kapitel wird die Konvertierung von IndraLogic 1.x-Projekten (MLD-1G) in IndraLogic 2G-Projekte (MLD-2G) beschrieben. Es wird insbesondere auf die Änderungen im Bereich "Motion" und "SPS" eingegangen.

Die Konvertierung von IndraLogic 1.x-Projekten (MLD-1G) in IndraLogic 2G-Projekte (MLD-2G) ist nur mit IndraWorks 13VRS oder mit IndraWorks 14VRS möglich.



Überprüfen Sie nach der Konvertierung alle Teile des Projekts auf eventuelle Konvertierungsfehler.

Die Anlage ist nach erfolgter Konvertierung komplett neu in Betrieb zu nehmen.



Die Projekte, die mit der Firmwareversion bis MPx08 entwickelt wurden, und Projekte für die Firmwareversion MPx17 / MPx18 und MPx20 besitzen eine andere Speicherorganisation (Byte-Order). Die Speicherorganisation ist:

bis MPx08: Little Endian

MPx17 / MPx18 / MPx20: Big Endian

Durch die Änderung der Speicherorganisation können folgende Probleme entstehen:

- bei Verwendung von BYTE-Adressierung (%QB oder %IB): E/A-Konfiguration neu parametrieren (%QB0 mit %QB1 vertauschen!)
- bei Verwendung von WORD-Adressierung (%QW oder %IW): Keine Änderung notwendig! Bevorzugte Adressierung!
- Verwendung von DWORD-Adressierung (%QD oder %ID): Wenn möglich durch WORD-Adressierung ersetzen / ggf. E/A-Konfiguration neu parametrieren bzw. Modul neu verdrahten.

10.2 Voraussetzungen in MLD-1G

Um ein bestehendes Projekt aus MLD-1G in MLD-2G übernehmen zu können, müssen folgende Voraussetzungen in MLD-1G überprüft bzw. voreingestellt werden:

**Laufzeiten des Projekts
kontrollieren**

IndraMotion MLD-2G benötigt einen größeren Rechenzeitbedarf, als die Implementierung der IndraMotion MLD-1G. Die Performance von IndraMotion MLD-2G ist bei kleinen Programmen geringer als bei IndraMotion MLD-1G. Bei größeren Programmen wird die geringere Performance durch den kompakteren Programmcode ausgeglichen.

HINWEIS

Nach der Umstellung auf MLD-2G reicht ggf. die freie Rechenzeit nicht aus, um das MLD-Programm abarbeiten zu können.

Kontrollieren Sie **vor der Umstellung** auf MLD-2G die Laufzeiten des Projekts. Die Auslastung bei IndraMotion MLD-1G sollte kleiner als 70 % sein.

Nach der Umstellung auf MLD-2G sollten Sie erneut prüfen, ob die freie Rechenzeit ausreicht, um das MLD-Programm abarbeiten zu können. Die Auslastung bei IndraMotion MLD-2G sollte kleiner als 90 % sein.

Am Einfachsten ist eine Überprüfung mit P-0-1364 durchzuführen. Der Parameter ist in beiden Versionen verfügbar und zeigt die freie Rechenzeit an.

Eine genauere Analyse der verfügbaren Zeit bietet der Funktionsbaustein "MX_IECTaskGetLoad". Er liefert in der Struktur "MX_IECTASKLOAD" Laufzeitinformationen der Task:

- "rLoad": Auslastung der Task im letzten Zyklus in Prozent bezogen auf die Zykluszeit
- "rLoadMax": maximale Auslastung der Task in Prozent bezogen auf die Zykluszeit
- "rFreeTime": ungenutzte Rechenzeit der Task bis zu ihrem nächsten Start vom letzten Zyklus in µs
- "RMinFreeTime": minimale ungenutzte verbleibende Rechenzeit der Task bis zu ihrem nächsten Start in µs



MLD-1G: Der Funktionsbaustein "MX_IECTaskGetLoad" ist in der Bibliothek "MX_Base" unter **Tools** zu finden.

MLD-2G: Der Funktionsbaustein "MX_IECTaskGetLoad" ist in der Bibliothek "MX_PLCOpen" unter **POUs ► Tools** zu finden.

Datentypen überprüfen

Um Übersetzungsfehler im Vorfeld der Projektkonvertierung zu umgehen, sollte die Verwendung der Datentypen überprüft werden. Die Datentypen "INT" / "WORD" und "DINT" / "REAL" sollten auf ihre richtige Verwendung hin überprüft werden. Der Compiler des Laufzeitsystems der MLD-2G prüft sehr streng die richtige Verwendung der Datentypen.

Zielsystem-Einstellungen

Soll ein Projekt verwendet werden, das bereits in MLD-1G entwickelt wurde, ist zu beachten, dass die "Trace-Aufzeichnung" in "Zielsystem-Einstellungen" deaktiviert wird. Bleibt "Trace-Aufzeichnung" aktiviert, kann das SPS-Programm nicht importiert werden.

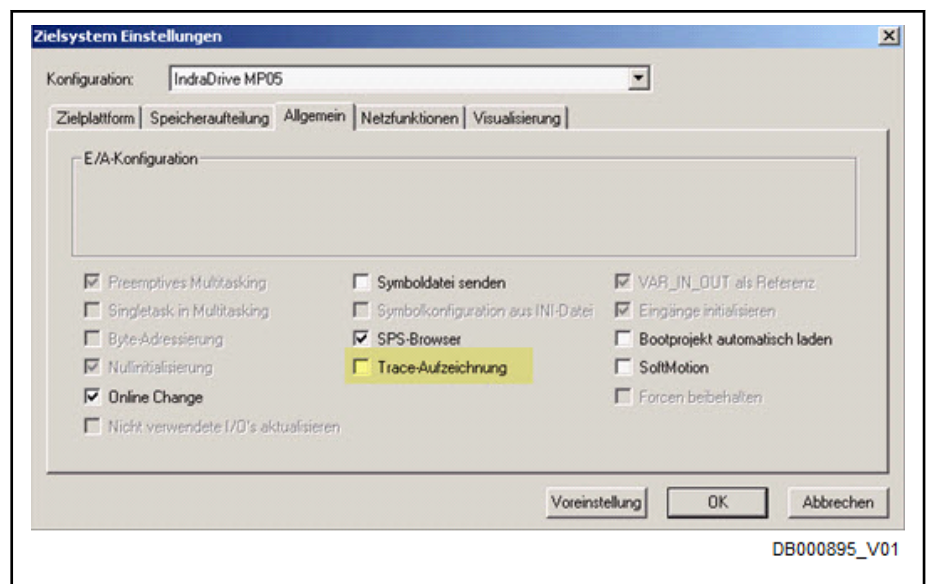


Abb. 10-1: Trace-Aufzeichnung

10.3 Einfügen in MLD-2G

10.3.1 Projekt einfügen

Ab der Version IndraWorks MLD 13VRS ist die Bedienoberfläche für die Programmierung und Visualisierung der MLD-2G in IndraWorks integriert.

Vorhandene Programme, die bereits in MLD-1G entwickelt wurden, können wie folgt importiert werden:

1. Wählen Sie in IndraWorks im Projektextplorer das Projekt aus, in das Sie das MLD-1G-Projekt importieren wollen.
2. Klappen Sie den Projektbaum **MLD ► Logic** auf und klicken mit der rechten Maustaste auf **Application**.
3. Wählen Sie in dem sich öffnenden Kontextmenü **Datenübernahme....**
⇒ Der Standard-Windows-Dialog zum Öffnen einer Datei erscheint.
4. Stellen Sie ggf. den Dateifilter auf "IndraLogic-Projektdatei", um die Auswahl der zu öffnenden Dateien auf Dateien mit der Endung ".pro" zu begrenzen.
Wählen Sie die zur Übernahme bestimmte Projektdatei aus.
⇒ Der Dialog zur Auswahl der zu importierenden Objekte wird geöffnet.
5. Wählen Sie die zu importierenden Objekte aus und bestätigen Sie Ihre Auswahl mit **OK**.

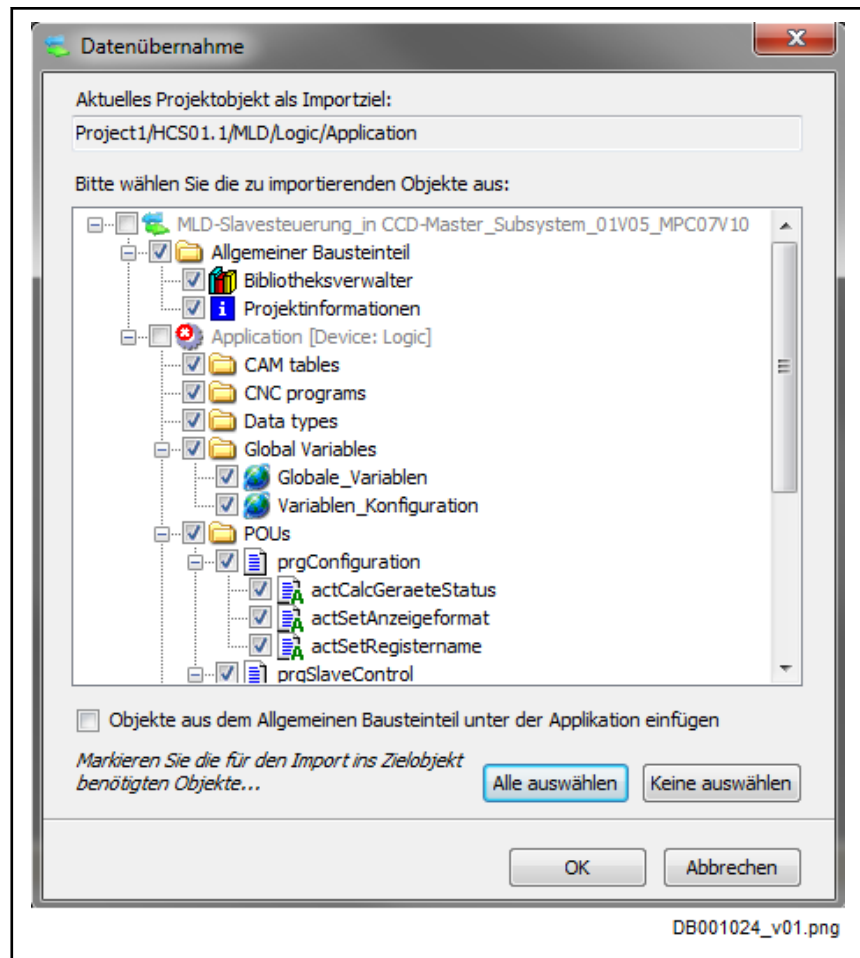


Abb. 10-2: Zu importierenden Objekte auswählen

⇒ Das MLD-1G-Projekt wird konvertiert.

10.3.2 Übersetzungsfehler nach der Projektkonvertierung

Syntax der Direktvariablen Ein Anwenderprogramm kann nach der Portierung in MLD-2G durch globales "Suchen und Ersetzen" umgestellt werden.

Beispiel:

"Suchen und Ersetzen"

MLD-1G: DV_P_0_1370 := DV_P_0_1371;

MLD-2G: DV_Axis[1].P_0_1370 := DV_Axis[1].P_0_1371;

besser:

DV_Axis[AXIS_1].P_0_1370 := DV_Axis[AXIS_1].P_0_1371;

**Schreibschutz der Parameter
P-0-1410 bis P-0-1429**

Falls in einem Projekt der MLD-1G schreibend auf die Parameter P-0-1410 bis P-0-1429 zugegriffen wurde, müssen stattdessen in der MLD-2G SPS-Ausgänge definiert und beschrieben werden.

Beispiel:

MyAusgang0 : WORD AT %QB0;

statt DV_P_0_1410

	<pre>MyAusgang1 : WORD AT %QB2; statt DV_P_0_1411</pre>
Zuweisung von verschiedenen Datentypen	Entsprechende Zugriffe auf das Prozessabbild müssen auf Beschreiben von SPS-Ausgängen umgestellt werden. Werden unterschiedliche Datentypen verwendet, wird beim Übersetzungslauf eine Warnung generiert - das Projekt ist nach der Übersetzung lauffähig. Beispiel: <i>Programm:</i> <pre>diLagesollwert2 :DINT; rLageistwert2 :REAL; rLagedifferenz2 :REAL;</pre> Warnung wird generiert: <pre>rLagedifferenz2 := diLagesollwert2 - rLageistwert2;</pre> richtig: <pre>rLagedifferenz2 := DINT_TO_REAL(diLagesollwert2) - rLageistwert2;</pre> Behebung der Warnung durch die Wandlung von Datentypen, z. B. • INT_TO_WORD • DINT_TO_REAL etc.
Aufruf von Funktionsbausteinen oder ihrer Methoden mit dem Funktionsbaustein-Typ	Werden Funktionsbausteine oder ihre Methoden mit der Angabe des Funktionsbaustein-Typs aufgerufen, so wird beim Übersetzungslauf ein Fehler generiert. Eine Abhilfemaßnahme ist, den Funktionsbaustein zu instanzieren oder den Aufruf innerhalb des Funktionsbausteins ohne Präfix durchzuführen. Beispiel: <pre>MX_PID_ControllerPL.Init_Regler(); (→führt zu Fehler) Init_Regler(); (→fehlerfrei)</pre>
Funktionen mit Dummy-Argumenten	Bei MLD-1G: Funktionen ohne Parameter sind nicht möglich, ein Parameter muss angegeben werden, dieser wird aber nicht benutzt. Bei MLD-2G: Funktionen können auch ohne Parameter programmiert werden.

10.3.3 Anpassungen in MLD-2G

Tasksystem überprüfen	Nach dem Import des Programms muss das Tasksystem überprüft werden. IndraLogic MLD-2G unterstützt verschiedene Tasksysteme: Events: • Zyklische Task • Ereignisgesteuerte Task • Freilaufende Task (Verhalten wie in MLD-1G gewohnt) • Extern ereignisgesteuerte Task – CCD_SYNCHRONIZED_TASK – FKM_SYNCHRONIZED_TASK – PositionLoopEvent

Der "PositionLoopEvent" ermöglicht es, bestimmte SPS-Programme, die in einer Task konfiguriert werden, im Positionsregler-Takt abzuarbeiten.

Die Standard-Installation von IndraWorks MLD beinhaltet das MLD-Package ohne "PositionLoopEvent". Der "PositionLoopEvent" ist mit einem zusätzlichen IW-Package verfügbar (z. B. MldPosLoopDevice_V20.12.0.0.iwpackage). Das IW-Package zum "PositionLoopEvent" kann über den Support bezogen und anschließend nachinstalliert werden.

Beim "PositionLoopEvent" darf nicht auf I/O-Variablen zugegriffen werden. Der indirekte Zugriff auf I/O-Variablen kann auf zwei Arten erfolgen:

- Lesen der I/Os in zyklischer Task und kopieren auf Merker. Im "PositionLoopEvent" auf die Merker zugreifen.
- Direkter Zugriff auf Parameter im "PositionLoopEvent" (wenn es schnell gehen soll!)



Bei MLD-2G ist es nicht möglich ein Projekt nur mit eventgesteuerten Tasks zu erstellen. Es ist notwendig mindestens eine zyklische oder freilaufende Task anzulegen.

System-Ereignisse (Systemevent):

Bei MLD-2G werden verschiedene Systemevents unterstützt:

- StartDone (Aufruf nach Start der Applikation)
- StopDone (Aufruf nach Stop der Applikation)
- PrepareReset (Aufruf vor Reset der Applikation)
- ResetDone (Aufruf nach Reset der Applikation)



- Es dürfen nur die hier beschriebenen Systemevents verwendet werden.
- Bei der Verwendung ist darauf zu achten, dass möglichst wenig Code durchlaufen werden sollte.

10.4 Projektkonvertierung MPx18 --> MPx20

Bei der Umstellung eines SPS-Projekts der Firmware-Version MPx18 auf ein Projekt für die Firmware-Version MPx20 ist darauf zu achten, dass sich die Deklaration der Variablen "AXIS_REF" geändert hat. Die Variablen vom Typ "AXIS_REF" werden für PLCOPEN-Bausteine verwendet. Aus diesem Grund wird hier bei der Fehlermeldung auf PLCOPEN verwiesen. Die Fehlermeldung weist nicht eindeutig auf eine falsche Deklaration hin.

Die Deklaration der Variablen muss angepasst werden. Von bisher "VAR_IN_OUT" muss die Variable jetzt diese Deklaration enthalten: "VAR_IN_OUT CONSTANT"

11 Service und Support

Für Ihre schnelle und optimale Unterstützung verfügen wir über ein dichtes weltweites Servicenetz. Unsere Experten stehen Ihnen mit Rat und Tat zur Seite. Sie erreichen uns täglich **rund um die Uhr - auch an Wochenenden und Feiertagen**.

Service Deutschland Unser technologieorientiertes Competence Center in Lohr deckt alle Belange rund um den Service für elektrische Antriebe und Steuerungen ab.

Sie erreichen unsere **Service-Hotline** und unseren **Service-Helpdesk** unter:

Telefon: **+49 9352 40 5060**
Fax: **+49 9352 18 4941**
E-Mail: service.svc@boschrexroth.de
Internet: <http://www.boschrexroth.com>

Auf unseren Internetseiten finden Sie ergänzende Hinweise zu Service, Reparatur (z. B. Anlieferadressen) und Training.

Service weltweit Außerhalb Deutschlands nehmen Sie bitte zuerst Kontakt mit Ihrem Ansprechpartner auf. Die Hotline-Rufnummern entnehmen Sie bitte den Vertriebsadressen im Internet.

Vorbereitung der Informationen Wir können Ihnen schnell und effizient helfen, wenn Sie folgende Informationen bereithalten:

- Eine detaillierte Beschreibung der Störung und der Umstände
- Angaben auf dem Typenschild der betreffenden Produkte, insbesondere Typenschlüssel und Seriennummern
- Ihre Kontaktdaten (Telefon-, Faxnummer und E-Mail-Adresse)

Index

Symbole

.NET-Bibliotheken..... 176

A

Achs-Kommandierung..... 58
 Achsadressierung..... 61
 Achse..... 49
 Achskommandierung..... 53
 Achskontrolle..... 53
 Achssteuerung..... 49
 Achsverfügbarkeit für Funktionen und Funktionsbausteine..... 73
 Anbindung eines VE* über Ethernet..... 167
 Antrieb als SPS-Gerät..... 238
 Antriebsfehler..... 97
 Antriebssystem..... 35
 Anwenderprogramm..... 28
 Anwendungs- und Programmierhinweise..... 76
 Application (Kontextmenü)..... 200
 archivieren..... 224
 archivieren von MLD-Projekte..... 294
 AT..... 48
 Aufbau der AxisData..... 115
 autarke Einzelachs-Motion Logic Control..... 56
 autarke Motion Logic Control..... 27, 57
 AxisInterface..... 48, 207

B

Backup von Retaindaten..... 267
 Bausteinverhalten..... 59
 Instanziierung..... 60
 Timing..... 59
 Bedienfeld..... 205
 Begriffe..... 16
 Bestellnummer Gerät..... 301
 bestimmungsgemäßer Gebrauch..... 31
 Einsatzfälle..... 31
 Bibliotheken..... 19
 allgemeine Eigenschaften..... 263
 hierarchischer Aufbau..... 265
 unterstützte Bibliotheken..... 263
 Bootprojekt..... 18, 276

C

C0241..... 83
 C0266..... 83
 CCD..... 48
 CCD-Datencontainer..... 154
 CCD-Master /CCD-Slave..... 48
 CCD-synchrone Motions task..... 81
 Einsatzgebiet..... 82
 Motions task-Zykluszeit..... 82
 Timing..... 82

D

Darstellungsmittel
 Hinweise..... 2
 Konventionen der Schreibweise..... 1
 Schreibweisen..... 1
 Datenkanal
 I/O-Kanal..... 110
 Parameterkanal..... 128
 Datenkanäle der IndraMotion MLD..... 107
 Direktvariablen..... 127
 Dokumentation
 Änderungsverlauf..... 1
 mitgeltende Dokumentationen..... 4
 Übersicht..... 4
 zusätzliche Dokumentationen..... 4
 Dokumentationsstruktur..... 4

E

eal_sdk..... 174
 Anwendungsbeispiele..... 174
 Dokumentation..... 174
 EAL4Android..... 176
 EAL4C..... 176
 EAL4DotNet..... 176
 EAL4Java..... 176
 EAL4LabVIEW..... 176
 Easy Automation Library..... 174
 Echtzeitkanal
 siehe Zyklische Achsdaten AxisData..... 108
 Einstellen der IP-Adresse..... 161
 Elektrisches Antriebssystem..... 35
 entfernte Achsen..... 48
 ereignisgesteuerte Tasks..... 80
 Event-Logging..... 298
 Eventtasks..... 90
 extern ereignisgesteuerte Task..... 80

F

Firmware-Voraussetzungen..... 24
 FKM-synchrone Motions task..... 81
 Einsatzgebiet..... 81
 Motions task-Zykluszeit..... 81
 Timing..... 81
 Freie Parameter..... 131
 Freie Restzeitscheibe..... 86
 freilaufende Task..... 80, 83
 Führungskommunikation..... 49
 Funktion..... 17
 Funktionsbaustein..... 17

G

GAT compact..... 28
 Betriebsarten-Steuerung..... 212
 Erste Schritte..... 209
 Konfiguration..... 208

Index

Teilfunktionen.....	208
GAT compact.....	48
Gebrauch	
bestimmungsgemäßer Gebrauch.....	31
nicht-bestimmungsgemäßer Gebrauch.....	32
Geräte-Editor.....	297
Logger.....	298
Taskaufstellung.....	298
Geräte-Kategorie.....	301
Geräte-Konfiguration.....	297
Geräte-Name.....	301
Geräte-Typ.....	301
Geräte-Version.....	301
Gerätesteuerung.....	49
Gerätesteuerung bei MLD-M.....	51
Globale Register.....	131
Grundlagen.....	16

H

Hardware-Voraussetzungen.....	23
Helpdesk.....	319
Herstellername.....	301
Hochlauf-Modus.....	49
Hotline.....	319

I

I/O-Kanal.....	110
I/O-Konfiguration der MLD.....	141
IEC 61131.....	18, 261
IEC 61131.....	18
IndraControl S20.....	16
IndraMotion for Handling.....	17
IndraMotion MLD.....	10
IndraMotion MLD-M.....	13
IndraMotion MLD-S.....	12
IndraMotion, Übersicht.....	9
IndraWorks-Package.....	236, 309
Informationen zum Gerät.....	301
Instanzbezeichner	
einfacher Datentypen.....	285
komplexer Datentypen.....	285
intelligente Servoachse.....	26, 55

J

Java.....	176

K

Konfiguration der Ein- und Ausgänge.....	141
Konfiguration der MLD mit IndraWorks.....	54
Konfiguration Gerät.....	297

L

LabVIEW.....	176
Laufzeitsystem.....	50
Leistungs- und Performance-Angaben.....	22
Logger im Laufzeitsystem.....	298
Logic (Kontextmenü): siehe Application.....	200

lokale Achse.....	48

M

MDT.....	48
mitgeltende Dokumentationen.....	4
MLD-Funktionsbausteine und Parameter zur	
Achskommandierung.....	76
MLD-M-Achsnummer.....	74
MLD-Projekte	
als Team bearbeiten.....	294
archivieren.....	294
versionieren.....	294
MLD-S, MLD-M, MLD.....	47
Motion Control-Kommandoschnittstelle	
Betriebsarten.....	139
Motion-Steuerung.....	48

N

Netzwerkvariablen.....	83
nicht-bestimmungsgemäßer Gebrauch.....	32
Folgen, Haftungsausschluss.....	31

O

OCI@Drives.....	170
Open Core Interface für Antriebe.....	170
Anwendungsbeispiele.....	177
Anwendungsmöglichkeiten.....	172
unterstützte Geräte.....	172
Zielpattformen.....	172

P

PAA.....	49
PAE.....	48
Parameter und Diagnosen zur Achskom-	
mandierung.....	77
Parameterkanal.....	128
PELV.....	40
periodische (zyklische) Tasks.....	80
persistente Variablen sichern und wieder	
herstellen.....	267
persistente Variablen speichern.....	267
PLCopen.....	18, 261
PLCopen-Bausteine.....	48
Programm.....	18
Programmiersprachen.....	19
empfohlene Basis-Programmiersprachen....	280
Hinweise zur SFC-Programmierung.....	280
Übersicht.....	279
Projektkonvertierung.....	313
Anpassungen in MLD-2G.....	317
Projekt aus MLD-1G übernehmen.....	315
Übersetzungsfehler beheben.....	316
Voraussetzungen in MLD-1G.....	313

R

Ressourcen.....	17
Steuerungskonfiguration.....	17

Taskkonfiguration.....	17	Ressourcen.....	17
Traceaufzeichnung.....	17	Task.....	17
Variablenkonfiguration.....	17	Typbezeichner.....	282
Watch- und Rezepturverwalter.....	17	SPS-Projekt importieren.....	221
Retaindaten sichern und wieder herstellen.....	267	SPS-Runtime.....	50
Rexroth IndraControl S20.....	16	SPS-Warnungen.....	96
Rexroth IndraMotion MLD		Stand der Technik.....	31
Einsatzgebiete.....	26	Standard-Ausgänge an Funktionsbausteinen.....	287
Rexroth Inline.....	15, 16, 146	Standard-Eingänge an Funktionsbausteinen...	287
Rexroth Inline Block IO.....	14, 151	Startverhalten.....	276
Rexroth Inline Modular IO.....	15	Steuerung	
Rezeptur als Datei auf dem PC speichern.....	270	Logger.....	298
Rezeptur lesen.....	270	Strukturierung von SPS-Projekten	
Rezeptur schreiben.....	270	Mehrprogrammtechnik.....	279
S		Taskverwaltung.....	278
Schloss-Symbol.....	294	Support.....	319
Schnittstellen und Datenkanäle der MLD.....	105	Systemverhalten der MLD	
Schutzkleinspannung.....	40	Anwendungs- und Programmierhinweise....	278
SDK		T	
Installation.....	174	Task.....	17
Voraussetzungen.....	174	Task-Eigenschaften.....	80
Sequential Motion Control (SMC).....	16, 29	Task-Konfiguration.....	89
Sercos Adresse der CCD-Slaves.....	74	Task-Überwachung.....	83
Sercos I/O.....	156	Task-Zykluszeiten.....	89
Konfiguration.....	156	Taskaufstellung.....	298
Prozessabbilder der Ein-/Ausgänge.....	157	Tasksystem.....	78
Sercos III Antrieb als SPS-Gerät		Tasksystem und Prioritäten.....	79
Konfiguration der zyklischen Daten.....	242	Teamserver.....	294
Parametrierung.....	240	Technische Daten.....	20
SPS-Programmierung.....	245	Technologie-Bausteine.....	48
Übersicht.....	239	Technologie-Bibliotheken.....	28
Sercos Peripherie		Technologiefunktion	16
(Frequenzumrichter) EFC.....	16	Diagnose- und Status-Informationen.....	194
Antriebsregelgerät.....	16	Dokumentation.....	194
RMB_SercosIII_Util.library.....	16	laden und aktivieren.....	192
Service-Hotline.....	319	parametrieren und in Betrieb nehmen.....	193
Sicherheitshinweise für elektrische Antriebe und Steuerungen.....	35	Übersicht.....	191
Signal-Zeit-Diagramm		Technologiepakete.....	28
für flankengesteuerte Funktionsbausteine.....	290	Timing.....	59
für zustandsgesteuerte Funktionsbausteine.....	288	TurnKey-Lösungen.....	28
SMC.....	16	Typbezeichner	
Software-Voraussetzungen.....	25	für Programme, Funktionsbausteine und Funktionen.....	283
SPS Shell.....	301	für Strukturen, Felder, Enumeratoren und IEC-Datentypen.....	284
SPS-Einstellungen.....	300	U	
SPS-Fehler.....	97	unterstützte Tasktypen.....	78
SPS-Programmierung		Update der IndraLogic-Geräteversion.....	236
Bootprojekt.....	18	V	
Funktion.....	17	VCS.....	294
Funktionsbaustein.....	17	Version Control System.....	294
globale Daten.....	280	versionieren von MLD-Projekte.....	294
History.....	282	Versionierung von Bibliotheken.....	291
Programm.....	18		
Programm-Header.....	281		

Versionsfunktion.....	291
Versorgungsgeräte als SPS-Gerät	
Allgemeines.....	248
Konfiguration der zyklischen Daten.....	249
Parametrierung.....	248
SPS-Programmierung.....	252
Verwendungshinweise und Anwendungsgebiete.....	23
virtuelle Leitachse.....	48
W	
Watchdog.....	83
Watchdog-Funktion.....	84
wiederherstellen.....	224
Z	
Zuordnung der logischen MLD-M-Achsnummer zur Sercos Adresse der CCD-Slaves.....	74
zusätzliche Dokumentationen.....	4
Zyklische Achsdaten AxisData.....	113

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



R911338913