

Rexroth IndraMotion MLC 12VRS Die ersten Schritte

R911333845
Ausgabe 01

Inbetriebnahmeanleitung



Titel	Rexroth IndraMotion MLC 12VRS Die ersten Schritte								
Art der Dokumentation	Inbetriebnahmeanleitung								
Dokumentations-Type	DOK-MLC***-F*STEP**V12-CO01-DE-P								
Interner Ablagevermerk	RS-bec1385da4450f050a6846a0005b156d-1-de-DE-12								
Zweck der Dokumentation	Diese Dokumentation beschreibt die "ersten Schritte" mit der IndraMotion MLC und RobotControl. Beschrieben werden die Hard- und Softwarevoraussetzungen und das Erstellen eines Projekts. Sie ist die Grundlage für die Online-Hilfe im System.								
Änderungsverlauf	<table><tr><th>Ausgabe</th><th>Stand</th><th>Bemerkung</th></tr><tr><td>Ausgabe 01</td><td>03.2011</td><td>Erstausgabe</td></tr></table>			Ausgabe	Stand	Bemerkung	Ausgabe 01	03.2011	Erstausgabe
Ausgabe	Stand	Bemerkung							
Ausgabe 01	03.2011	Erstausgabe							
Schutzvermerk	© Bosch Rexroth AG 2011 Weitergabe sowie Vervielfältigung dieser Unterlage, Verwertung und Mitteilung ihres Inhalts wird nicht gestattet, soweit nicht ausdrücklich zugestanden. Zuwiderhandlungen verpflichten zum Schadensersatz. Alle Rechte für den Fall der Patenterteilung oder Gebrauchsmustereintragung vorbehalten (DIN 34-1).								
Verbindlichkeit	Die angegebenen Daten dienen allein der Produktbeschreibung und sind nicht als zugesicherte Eigenschaften im Rechtssinne zu verstehen. Änderungen im Inhalt der Dokumentation und Liefermöglichkeiten der Produkte sind vorbehalten.								
Herausgeber	Bosch Rexroth AG Bgm.-Dr.-Nebel-Str. 2 • D-97816 Lohr a. Main Telefon +49 (0)93 52 / 40-0 • Fax +49 (0)93 52 / 40-48 85 http://www.boschrexroth.com/ System IndraLogic & Basis Motion Logic HB (vha)								
Hinweis	Diese Dokumentation ist auf chlorfrei gebleichtem Papier gedruckt.								

Inhaltsverzeichnis

	Seite
1 Zu dieser Dokumentation.....	5
1.1 Gültigkeit der Dokumentation.....	5
1.2 Erforderliche und ergänzende Dokumentationen.....	5
2 Wichtige Gebrauchshinweise.....	11
2.1 Bestimmungsgemäßer Gebrauch.....	11
2.1.1 Einführung.....	11
2.1.2 Einsatz- und Anwendungsbereiche.....	11
2.2 Nicht-bestimmungsgemäßer Gebrauch.....	12
3 Sicherheitshinweise für elektrische Antriebe und Steuerungen.....	13
3.1 Begriffsdefinitionen.....	13
3.2 Grundsätzliche Hinweise.....	14
3.2.1 Benutzung und Weitergabe der Sicherheitshinweise.....	14
3.2.2 Voraussetzungen für den sicheren Gebrauch.....	14
3.2.3 Gefahren durch falschen Gebrauch.....	16
3.3 Gefahrenbezogene Hinweise.....	17
3.3.1 Schutz gegen Berühren elektrischer Teile und von Gehäusen.....	17
3.3.2 Schutzkleinspannung als Schutz gegen elektrischen Schlag	18
3.3.3 Schutz vor gefährlichen Bewegungen.....	19
3.3.4 Schutz vor magnetischen und elektromagnetischen Feldern bei Betrieb und Montage.....	20
3.3.5 Schutz gegen Berühren heißer Teile.....	21
3.3.6 Schutz bei Handhabung und Montage.....	21
3.3.7 Schutz beim Umgang mit Batterien.....	21
3.3.8 Schutz vor unter Druck stehenden Leitungen.....	22
3.4 Erläuterung der Signalwörter und der Signalgrafik.....	22
4 Erste Schritte - IndraLogic.....	25
4.1 Ziel des Projekts "Erste Schritte - IndraLogic".....	25
4.2 Technische Voraussetzungen.....	25
4.3 Neues IndraWorks-Projekt erstellen.....	26
4.3.1 Steuerung zurücksetzen (CMLx5).....	26
4.3.2 IndraWorks starten.....	26
4.3.3 Neues Projekt erstellen.....	26
4.4 Neues Gerät in ein Projekt einfügen.....	27
4.5 Basiseinstellungen zum Gerät vorgeben.....	28
4.5.1 Geräteeinstellungen.....	28
4.5.2 Kommunikationseinstellungen.....	29
4.5.3 Funktionsmoduleinstellungen.....	30
4.6 SPS-Programm erstellen.....	32
4.6.1 Die Steuerung einer Ampelanlage.....	32
4.6.2 Programme und Funktionsbausteine anlegen.....	32
Programm "PlcProg" in "CFC" oder in "ST" einfügen.....	32

Inhaltsverzeichnis

	Seite
Programm "SEQUENCE" in "AS" einfügen.....	34
Funktionsbaustein "TRAFFICSIGNAL" in "FUP" einfügen.....	35
Funktionsbaustein "WAIT" in "ST" einfügen.....	37
4.6.3 Programme und Funktionsbausteine editieren – erster Teil.....	38
Funktionsbaustein "TRAFFICSIGNAL" in "FUP" editieren.....	38
Funktionsbaustein "WAIT" in "ST" editieren.....	44
Programm "SEQUENCE" in "AS" editieren.....	46
4.6.4 Taskkonfiguration.....	56
4.6.5 Globale Variablen deklarieren.....	57
4.6.6 Testen von "SEQUENCE" im Simulationsmodus.....	58
4.6.7 Programme und Funktionsbausteine editieren – zweiter Teil.....	62
Programm "PlcProg" in "ST" editieren.....	62
Testen der Applikation im Simulationsmodus, PlcProg in ST.....	63
Programm "PlcProg" in "CFC" editieren.....	65
Testen der Applikation im Simulationsmodus, PlcProg in CFC.....	68
4.6.8 Testen der Applikation auf der Steuerung.....	70
4.7 Visualisierung zum SPS-Programm erstellen.....	72
4.7.1 Visualisierung in IndraLogic.....	72
4.7.2 Erstellen einer neuen Visualisierung.....	72
4.7.3 Visualisierung editieren.....	73
Erstes Element "Ampellampe" einfügen und editieren.....	73
Element "Ampellampe" kopieren, einfügen und editieren.....	76
Element "Ampelgehäuse" einfügen und editieren.....	78
Element "Ampelmast" einfügen und editieren.....	78
Element "Ampelfuß" einfügen und editieren.....	79
Zweite Ampel einfügen und editieren.....	79
EIN-Schalter einfügen und editieren.....	80
Beschriftung einfügen und editieren.....	81
Anzeige von Variablenwerten.....	82
4.7.4 Visualisierung testen.....	83
4.8 Projekt wiederherstellen.....	84
5 Erste Schritte - Motion.....	87
5.1 Ziel des Projekts "Erste Schritte - Motion".....	87
5.2 Technische Voraussetzungen.....	87
5.3 Anlegen des Motion-Teilprojektes.....	88
5.3.1 Steuerung zurücksetzen (CMLx5).....	88
5.3.2 IndraWorks starten.....	88
5.3.3 Neues Projekt erstellen.....	88
5.3.4 Neues Gerät in ein Projekt einfügen.....	89
5.3.5 Basiseinstellungen zum Gerät vorgeben.....	90
Geräteeinstellungen.....	90
Kommunikationseinstellungen.....	91
Funktionsmoduleinstellungen.....	92
5.3.6 SERCOS III-Antriebe (reale Achsen) zum Projekt hinzufügen.....	94
5.3.7 Virtuelle Achse zum Projekt hinzufügen.....	95

Inhaltsverzeichnis

	Seite
5.3.8 Download des Motion-Teilprojektes.....	96
5.4 Programmieren des Bewegungsablaufes in Funktionsbausteinsprache (FUP).....	98
5.4.1 Programmieren des Bewegungsablaufes in FUP, Allgemeines.....	98
5.4.2 Programmieren der virtuellen Achse VA1, Programmiersprache FUP.....	99
5.4.3 Download des SPS-Teilprojekts virtuelle Achse VA1 (FUP).....	106
5.4.4 Aufzeichnung der Bewegung von Achse VA1 (Trace), FUP.....	110
5.4.5 Programmieren der realen Achse RA1 (FUP).....	115
5.4.6 Download des kompletten SPS-Projektes (FUP).....	118
5.4.7 Workflow "Virtuelle Achse → Elektronisches Getriebe → Reale Achse".....	119
5.4.8 Aufzeichnung der Bewegung beider Achsen (Trace), FUP.....	120
5.5 Programmieren des Bewegungsablaufes in Strukturiertem Text (ST).....	121
5.5.1 Programmieren des Bewegungsablaufes in ST, Allgemeines.....	121
5.5.2 Programmieren der virtuellen Achse VA1, Programmiersprache ST.....	123
5.5.3 Download des SPS-Teilprojektes virtuelle Achse VA1 (ST).....	129
5.5.4 Aufzeichnung der Bewegung von Achse VA1 (Trace), ST.....	134
5.5.5 Programmieren der realen Achse RA1 (ST).....	139
5.5.6 Download des kompletten SPS-Projektes (ST).....	142
5.5.7 Workflow "Virtuelle Achse → Elektronisches Getriebe → Reale Achse".....	143
5.5.8 Aufzeichnung der Bewegung beider Achsen (Trace), ST.....	144
5.6 Projekt wiederherstellen.....	145
6 Erste Schritte - FlexProfile.....	149
6.1 Ziel des Projekts "Erste Schritte - FlexProfile".....	149
6.2 Technische Voraussetzungen.....	150
6.3 Erstellen des Motion-Teilprojektes.....	150
6.4 Erstellen eines FlexProfiles.....	151
6.4.1 Erstellen eines FlexProfiles, Allgemeines.....	151
6.4.2 Erstellen des FlexProfiles über IndraWorks-Dialoge.....	151
6.4.3 Erstellen des FlexProfiles über die SPS.....	154
6.5 Komplettierung des SPS-Teilprojektes, Ausführen eines FlexProfile-Bewegungsschrittes.....	157
6.6 Download und Test des SPS-Teilprojektes.....	161
6.7 Workflow "Virtuelle Achse → FlexProfile 0 → Reale Achse".....	162
6.8 Aufzeichnung der Bewegung beider Achsen (Trace).....	163
6.9 Projekt wiederherstellen.....	164
7 Erste Schritte - Robot-Control.....	165
7.1 Ziel des Projekts "Erste Schritte - Robot-Control".....	165
7.2 Erforderliche Hard- und Software.....	165
7.3 Vorgehensweise beim Erstellen eines RoCo-Projektes.....	166
7.4 Detail zum Projekt "RoCoTestV12".....	177
7.5 Robot-Control - Applikationsbeispiele.....	179
7.5.1 RoCo-Grundlagen für Pick & Place-Anwendungen.....	179
Allgemein.....	179
Überschleifen.....	179
Bandsynchronisation.....	181

Inhaltsverzeichnis

	Seite
Bandsynchronisation mit überschleifenden Bewegungen.....	183
Hinweise zum Überschleifen bei Bandsynchronisation.....	185
Beispiel-Programm.....	186
8 Service und Support.....	189
Index.....	191

1 Zu dieser Dokumentation

1.1 Gültigkeit der Dokumentation

Zielgruppe Diese Dokumentation richtet sich an Anwender (Neueinsteiger), die sich unter Nutzung einer IndraMotion MLC-Steuerung in die Grundlagen des IndraWorks Engineering-Systems einarbeiten wollen.

Die Dokumentation enthält die Kapitel:

- [Erste Schritte - IndraLogic, Seite 25](#),
Grundlagen der Programmierung mit IndraLogic und der Visualisierung,
Das Kapitel ist auch ohne Steuerung im Simulationsmodus (Soft-SPS) nachvollziehbar.
- [Erste Schritte - Motion, Seite 87](#),
Grundlagen der Steuerung eines Antriebs aus einem SPS-Programm (PLCopen), Aufzeichnung der Bewegung der Achsen mittels Trace.
Das Kapitel ist mit einer realen Steuerung auch ohne Antrieb (mit virtuellen Achsen) nachvollziehbar.
- [Erste Schritte - FlexProfile, Seite 149](#),
Erzeugung und Anwendung von Bewegungsprofilen.
Das Kapitel baut auf [Erste Schritte - Motion](#) auf.
- [Erste Schritte - Robot-Control, Seite 165](#),
Anlegen eines Robot-Control-Projektes bis zum Ausführen des RoCo-Quellcodes.

Allgemeine Hinweise Die nachstehenden Kapitel informieren über

- [Erforderliche und ergänzende Dokumentationen, Seite 5](#),
- [Wichtige Gebrauchshinweise, Seite 11](#), und
- [Sicherheitshinweise für elektrische Antriebe und Steuerungen, Seite 13](#).

1.2 Erforderliche und ergänzende Dokumentationen

Dokumentationstitel mit Typenschlüssel und Materialnummer

IndraWorks		MLC	XLC
/36/	Rexroth IndraWorks 12VRS Software-Installation DOK-IWORKS-SOFTINS*V12-COxx-DE-P, R911334395 Die Dokumentation beschreibt den Installationsprozess von IndraWorks.	X	X
/5/	Rexroth IndraWorks 12VRS Engineering DOK-IWORKS-ENGINEE*V12-APxx-DE-P, R911334387 Diese Dokumentation beschreibt die Verwendung von IndraWorks, in das sich die Rexroth Engineering-Tools integrieren. Beschrieben werden das Arbeiten mit IndraWorks und die Bedienung der Oszilloskop-Funktion.	X	X
/20/	Rexroth IndraMotion MLC 12VRS Funktionsbeschreibung DOK-MLC***-FUNC****V12-APxx-DE-P, R911333847 Die Dokumentation dient der Beschreibung der Wizards, Kontextmenüs, Dialoge, Steuerungsinbetriebnahme, Gerätekonfiguration, Funktionalitäten der IndraMotion MLC.	X	

Zu dieser Dokumentation

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

Motion		MLC	XLC
/23/	Rexroth IndraLogic XLC IndraMotion MLC 12VRS PLCopen-Bibliotheken DOK-XLCMLC-FUNLIB**V12-LIxx-DE-P, R911334181 Diese Dokumentation beschreibt die Funktionsbausteine, Funktionen und Datentypen der Bibliotheken RIL_CommonTypes, ML_Base und ML_PLCopen für die IndraLogic XLC/IndraMotion MLC. Die Dokumentation enthält ferner die Fehlerreaktion der Bausteine.	X	X
/27/	Rexroth IndraLogic XLC IndraMotion MLC 12VRS Generic Application Template DOK-XLCMLC-TF*GAT**V12-APxx-DE-P, R911334190 Dieses Handbuch stellt dem IndraLogic SPS-Programmierer ein strukturiertes Template, welches zum Hinzufügen und Bearbeiten des SPS-Programmiercodes verwendet werden kann, zur Verfügung. Beschrieben werden das Template selbst, der Template Wizard und Beispielapplikationen.	X	X
/31/	Rexroth IndraMotion MLC 12VRS RCL-Programmieranleitung DOK-MLC***-RCL*PRO*V12-APxx-DE-P, R911333851 Das vorliegende Handbuch informiert über die Robot Control. Die Programmiersprache RCL (Robot Control Language) steht dabei im Vordergrund. Beschrieben werden die Programmstruktur, Variablen, Funktionen, Bewegungsanweisungen sowie die notwendigen Parameter des Systems.	X	

Zu dieser Dokumentation

/21/	Rexroth IndraLogic XLC IndraMotion MLC 12VRS Parameter DOK-XLCMLC-PARAM***V12-RExx-DE-P, R911334185 Diese Dokumentation beschreibt die Parameter der Systeme XLC/MLC, sowie das Wechselspiel zwischen Parametrieren und Programmieren. Beschrieben werden die Achs-Parameter, die Steuerungs-Parameter, die Kinematik-Parameter, die Messtaster- und die Nockenschaltwerks-Parameter.	X	X
/10/	Rexroth IndraDrive Firmware für Antriebsregelgeräte MPH-, MPB-, MPD-, MPC-07 DOK-INDRV*-MP*-07VRS**-FKxx-DE-P, R911328669		
/11/	Rexroth IndraDrive MPx-16 Funktionen DOK-INDRV*-MP*-16VRS**-APxx-DE-P, R911326484		

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

HMI		MLC	XLC
/8/	Rexroth IndraWorks 12VRS HMI DOK-IWORKS-HMI*****V12-APxx-DE-P, R911334391 Diese Dokumentation beschreibt die Funktionen, die Konfiguration und die Bedienung der Bedienoberflächen IndraWorks HMI Engineering und IndraWorks HMI Operation.	X	X
/6/	Rexroth IndraWorks 12VRS WinStudio DOK-IWORKS-WINSTUD*V12-APxx-DE-P, R911333843 Diese Dokumentation die Installation der Software, das Arbeiten mit WinStudio und das Erstellen und den Betrieb von Applikationen.	X	X
/50/	Rexroth IndraLogic XLC IndraMotion MLC 12VRS HMI-Anbindung DOK-XLCMLC-HMI*****V12-APxx-DE-P, R911334183 Dieses Dokument beschreibt die von der IndraLogic XLC und IndraMotion MLC unterstützten Visualisierungssysteme sowie deren Anbindung.	X	X

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

Zu dieser Dokumentation

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

Hardware		MLC	XLC
/1/	Rexroth IndraControl L45/L65 DOK-CONTRL-IC*L45*L65*-PRxx-DE-P, R911324660	X	X
/2/	Rexroth IndraControl L25 DOK-CONTRL-IC*L25*****-PRxx-DE-P, R911328473	X	X
/24/	Rexroth IndraControl Lxx 12VRS Funktionsmodule DOK-CONTRL-FM*LXX**V12-APxx-DE-P, R911333832 Diese Dokumentation dient der Beschreibung aller Funktionsmodule der Lxx Steuerungen inkl. Engineering und Diagnose.	X	X
/12/	Rexroth IndraDrive Antriebsregelgeräte MPx-02 bis MPx-07 DOK-INDRV*-GEN-**VRS**-PAxx-DE-P, R911297316		
/13/	Rexroth IndraDrive MPx-02 bis MPx-07 und HMV DOK-INDRV*-GEN-**VRS**-WAxx-DE-P, R911297318		
/35/	Rexroth IndraDrive Antriebsregelgeräte Steuerteile CSB01, CSH01, CDB01 DOK-INDRV*-CSH*****-PR08-DE-P, R911295011		

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

Zu dieser Dokumentation

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

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

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

2 Wichtige Gebrauchshinweise

2.1 Bestimmungsgemäßer Gebrauch

2.1.1 Einführung

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

Die Produkte dürfen nur bestimmungsgemäß eingesetzt werden. Wenn sie nicht bestimmungsgemäß eingesetzt werden, dann können Situationen entstehen, die Sach- und Personenbeschädigung nach sich ziehen.



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

Bevor Sie die Produkte der Firma Bosch 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

Rexroth IndraMotion MLC

Die IndraMotion MLC und ihre Funktionsmodule sind für Motion-/Logic-Anwendungen geeignet.



Die IndraMotion MLC und ihre Funktionsmodule 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 Hardware- Komponenten und mit der in den jeweiligen Dokumentationen und den Funktionsbeschreibungen angegebenen und spezifizierten Soft- und Firmware erfolgen.

Die IndraMotion MLC und ihre Funktionsmodule sind für den Einsatz in ein- und mehrachsigen Antriebs- und Steuerungsaufgaben entwickelt worden.

Wichtige Gebrauchshinweise

Für den applikationsspezifischen Einsatz des Maschinenbedien- und Visualisierungsterminals stehen Gerätetypen mit unterschiedlicher Ausstattung und unterschiedlichen Schnittstellen zur Verfügung.

Typische Anwendungsbereiche für IndraMotion MLC und ihre Funktionsmodule sind:

- [Handhabungs- und Montagesysteme]
- [Verpackungs- und Lebensmittelmaschinen]
- [Druck- und Papierverarbeitungsmaschinen]

Rexroth IndraMotion MLC und ihre Funktionsmodule dürfen nur unter den in den weiterführenden Dokumentationen 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 Rexroth IndraMotion MLC und ihrer Funktionsmodule außerhalb der vorgenannten Anwendungsgebiete oder unter anderen als den in der Dokumentation beschriebenen Betriebsbedingungen und angegebenen technischen Daten gilt als "nicht-bestimmungsgemäß".

Die Rexroth IndraMotion MLC und ihre Funktionsmodule dürfen nicht eingesetzt werden, wenn sie

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

3 Sicherheitshinweise für elektrische Antriebe und Steuerungen

3.1 Begriffsdefinitionen

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

Sicherheitshinweise für elektrische Antriebe und Steuerungen

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

3.2 Grundsätzliche Hinweise

3.2.1 Benutzung und Weitergabe der Sicherheitshinweise

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

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

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

3.2.2 Voraussetzungen für den sicheren Gebrauch

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

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

Sicherheitshinweise für elektrische Antriebe und Steuerungen

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

Der Maschinenhersteller und Anlagenerrichter muss für seine individuelle Anwendung die Eignung

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

Länderspezifische Vorschriften, die vom Anwender zu berücksichtigen sind

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

Sicherheitshinweise für elektrische Antriebe und Steuerungen

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

3.2.3 Gefahren durch falschen Gebrauch

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

3.3 Gefahrenbezogene Hinweise

3.3.1 Schutz gegen Berühren elektrischer Teile und von Gehäusen



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

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

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

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

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

Sicherheitshinweise für elektrische Antriebe und Steuerungen

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

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

Abb.3-1: Mindestquerschnitt der Schutzleiterverbindung

3.3.2 Schutzkleinspannung als Schutz gegen elektrischen Schlag

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

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

3.3.3 Schutz vor gefährlichen Bewegungen

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

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

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

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

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

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

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

Vermeidung von Unfällen, Körperverletzung und/oder Sachschaden:

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

Sicherheitshinweise für elektrische Antriebe und Steuerungen

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

3.3.4 Schutz vor magnetischen und elektromagnetischen Feldern bei Betrieb und Montage

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

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

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

3.3.5 Schutz gegen Berühren heißer Teile

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

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

3.3.6 Schutz bei Handhabung und Montage

Verletzungsgefahr durch unsachgemäße Handhabung! Körperverschüttung 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).

Sicherheitshinweise für elektrische Antriebe und Steuerungen

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



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

3.3.8 Schutz vor unter Druck stehenden Leitungen

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

Verletzungsgefahr durch unsachgemäße Handhabung von unter Druck stehenden Leitungen!

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



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

3.4 Erläuterung der Signalwörter und der Signalgrafik

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

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

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

Sicherheitshinweise für elektrische Antriebe und Steuerungen

 GEFAHR

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

 WARNUNG

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

 VORSICHT

Bei Nichtbeachtung dieses Sicherheitshinweises können mittelschwere oder leichte Körpverletzung eintreten.

HINWEIS

Bei Nichtbeachtung dieses Sicherheitshinweises können Sachschäden eintreten.

4 Erste Schritte - IndraLogic

4.1 Ziel des Projekts "Erste Schritte - IndraLogic"

Anhand des Entwurfes und der Realisierung einer einfachen Ampelsteuerung wird eine Einführung in die SPS-Programmierung und Inbetriebnahme der Steuerung gegeben. Im Projekt werden die SPS-Programmiersprachen Strukturierter Text (ST) und Ablaufsprache (AS), Funktionsbausteinsprache (FUP) und wenn Sie möchten, Continuous Function Chart (CFC), verwendet.

Eine einfache Inbetriebnahmevisualisierung rundet das Projekt ab.

In diesem Abschnitt werden einige typische Bedienabläufe in IndraWorks beschrieben. Wenn Sie anhand dieser Beschreibungen vorgehen, werden Sie mit der Benutzung von IndraWorks vertraut und erkennen, wie die einzelnen Komponenten zusammenspielen.

Projektwiederherstellung:

Unabhängig von Ihrer Hardware z.B. XLC L25, L45, L65 (mit oder ohne SERCOS III-Bus) oder MLC L25, L45, L65 können Sie das Projekt auf Ihrem Rechner mit Ihrer Steuerung regenerieren.

Nutzen Sie dazu die in Ihrem Installationsverzeichnis unter "Iw:\Programme\Rexroth\IndraWorks\Projects" abgelegte Exportdatei "Application_FirstSteps_IndraLogic_Basics.iwx". Detaillierte Erläuterungen finden Sie unter [Projekt wiederherstellen, Seite 84](#).



IndraWorks wird standardmäßig in "c:\Programme\Rexroth" installiert. Das Installationsverzeichnis von IndraWorks kann sich vom angegebenen Installationspfad unterscheiden.

Teilaufgaben

- Projekt mit Steuerung anlegen,
- SPS-Programm bearbeiten,
- SPS-Visualisierung bearbeiten,
- SPS-Programm simulieren,
- SPS-Programm auf die Steuerung laden.

4.2 Technische Voraussetzungen

- PC mit installiertem IndraWorks Engineering für IndraMotion MLC, 12VRS,
Steuerung, Typ "IndraMotion MLC L25", "IndraMotion MLC L45" oder "IndraMotion MLC L65",
aktuelle Firmware ist geladen.

Die Schaffung dieser Voraussetzungen sind beschrieben in

- Rexroth IndraMotion MLC 12VRS Funktionsbeschreibung, DOK-IM*ML*-SYSTEM**V12-RExx-DE-P und
- IndraWorks Online-Hilfe, **System IndraLogic XLC / IndraMotion MLC ▶ XLC/MLC-Funktionsbeschreibung ▶ Inbetriebnahme und Bedienung**,

Steuerung, z.B. Typ "IndraLogic XLC L65.1", die aktuelle Firmware ist geladen,

- die IP-Adresse ist bekannt,
- die Steuerung ist über Ethernet mit dem Programmiergerät (PC) verbunden.

Erste Schritte - IndraLogic

- Falls ein SERCOS III-Bus vorhanden ist, muss dieser gebrückt werden.

4.3 Neues IndraWorks-Projekt erstellen

4.3.1 Steuerung zurücksetzen (CMLx5)

Die Steuerung kann von vorheriger Nutzung alte Programmbestandteile, Parametrierungen und SPS-Daten enthalten, die in diesem Schritt durch das Speicherlöschen entfernt werden.

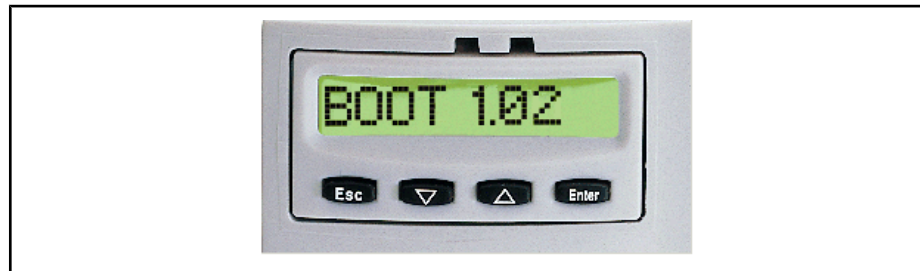


Abb.4-1: Reboot der Steuerung

Nach dem Einschalten der Steuerung sind die beiden äußeren Tasten unter dem Display (<Esc> und <Enter>) bei Erscheinen von **BOOT 1.02** zu drücken und so lange zu halten, bis die Ausgabe **BOOTSTOP** erscheint. Danach sind die beiden inneren Tasten zu drücken, es erscheint zunächst **CLEANUP**, dann **REBOOT**, und die Tasten können wieder losgelassen werden.

Die Steuerung läuft nun bis **BB RDY / RDY**, der Speicher ist geleert, eine eventuell vorhandene Motion-Komponente und die SPS-Komponente sind betriebsbereit.

4.3.2 IndraWorks starten

Starten Sie IndraWorks über **Start ▶ Programme ▶ Rexroth ▶ IndraWorks ▶ Engineering**.

Die Entwicklungsumgebung von IndraWorks wird gestartet. Alternativ kann IndraWorks über ein Desktop-Symbol gestartet werden.



Desktop-Symbol:

4.3.3 Neues Projekt erstellen

Um ein neues Projekt anzulegen, klicken Sie im Arbeitsbereich auf "Leeres Projekt erzeugen".

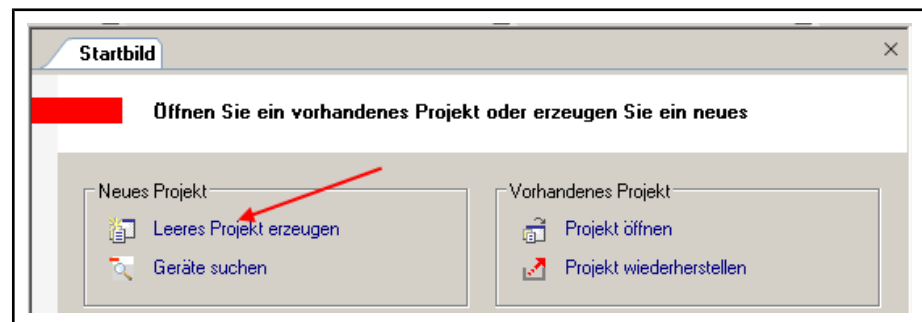


Abb.4-2: Leeres Projekt erzeugen

Alternativ können Sie ein neues Projekt über **Datei ▶ Neu ▶ Projekt...** anlegen.

Der Assistent zur Vergabe des Projektnamens und der Speicherplatzeinstellungen für ein neues IndraWorks-Projekt startet, siehe folgende Abbildung.

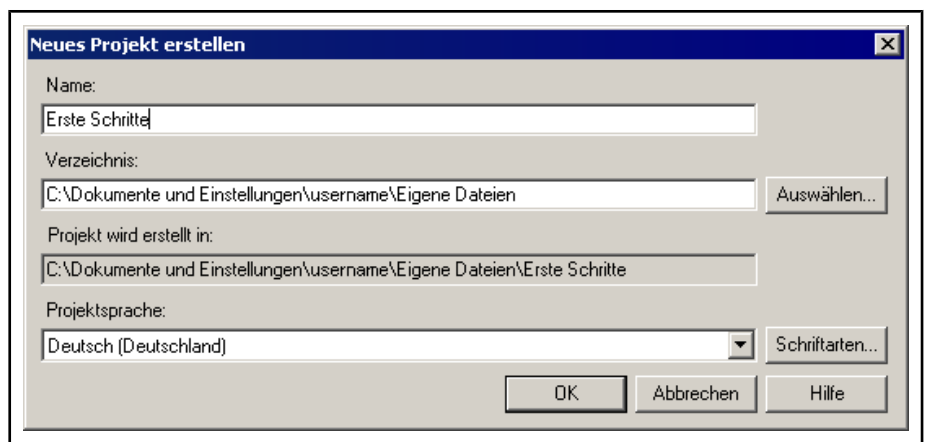


Abb.4-3: Einstellungen zum Abspeichern eines neuen IndraWorks-Projektes

- | | |
|---------------------------------|--|
| Name | Geben Sie dem Projekt einen beliebigen Namen, Vorschlag: "Erste Schritte". |
| Verzeichnis | Das Feld enthält voreingestellt einen Speicherplatz im Verzeichnis C:\Dokumente und Einstellungen\username\Eigene Dateien.
Alternativ können Sie auch ein anderes Verzeichnis angeben. |
| Projekt wird erstellt in | Verzeichnis und Projektname werden automatisch angelegt. |
| Projektsprache | Mit der Auswahl der Projektsprache legen Sie die Sprache ihrer Anwendertexte fest.

Das neu angelegte Projekt erscheint im Projektextplorer mit dem vorher definierten Projektnamen, siehe folgende Abbildung. |

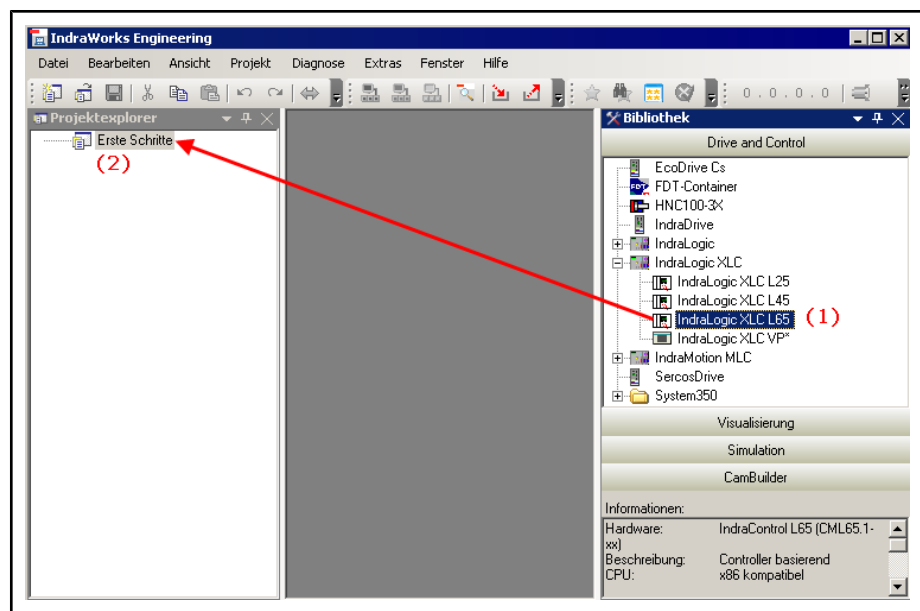


Abb.4-4: Erste Schritte Projekt

4.4 Neues Gerät in ein Projekt einfügen

Ziehen Sie aus der Bibliothek "Drive and Control" das gewünschte Gerät (z.B. Ordner "IndraLogic XLC", "IndraLogic XLC L65") in das neu angelegte Projekt im Projektextplorer.

Erste Schritte - IndraLogic



(1) Gerät: IndraLogic XLC L65

(2) Projekt: Erste Schritte

Abb.4-5: *Gerät einfügen*

Anschließend startet, wie nachstehend beschrieben, automatisch ein Assistent zur Vergabe notwendiger Basiseinstellungen.

4.5 Basiseinstellungen zum Gerät vorgeben

4.5.1 Geräteeinstellungen

Geben Sie die allgemeinen Daten zum eingefügten Gerät an.

Abb.4-6: *Geräteeinstellungen*

- Gerätename** Geben Sie bei "Gerätenamen" einen beliebigen Gerätenamen ein, z.B. "DCC_Control". Das Feld enthält voreingestellt den Namen aus der Bibliothek. Das Gerät erscheint später unter dem angegebenen Namen im Projektexplorer.
- Kommentar** Fügen Sie zur ausführlichen Beschreibung des Gerätes einen beliebigen Kommentar ein. Dieses Feld ist optional.
- Ersteller** Geben Sie den Namen des Projekterstellers an. Dieses Feld ist optional.



Diese Angaben lassen sich auch später noch ändern.

Bestätigen Sie die Eingaben mit der Schaltfläche "Weiter>>".

4.5.2 Kommunikationseinstellungen

Definieren Sie im Dialog "Konfiguration" die Einstellungen zur Firmware und der Kommunikation zwischen IndraWorks und Ihrem Gerät (hier: IndraLogic XLC L65).

Abb.4-7: Konfiguration

- Gerätetyp** Der Gerätetyp ist durch die Steuerungsauswahl vorbestimmt. Gibt es für einen Gerätetyp mehrere unterschiedliche Bestückungen, ist eine manuelle Ergänzung notwendig (Beispiel: IndraLogic XLC Lx5 gibt es in der Ausführung mit SERCOS III-Bus (CMLx5.1-3P) und ohne SERCOS III-Bus (CMLx5.1-NP).
- Firmware-Version** Hier kann die Version der Firmware ausgewählt werden. In der Regel kann diese Einstellung so belassen werden.
- Firmware-Release** Hier wird das Release der Firmware innerhalb einer Firmwareversion eingestellt. Auch diese Einstellung sollte so belassen werden.

IP-Adresse  Die Eintragungen zur Ethernet-Kommunikation werden im [Simulationsmodus](#), [Seite 58](#), nicht ausgewertet. Der Verbindungstest entfällt.

- SPS-Gateway** Geben Sie unter "Gerät IP-Adresse" Ihre Zieladresse ein, d.h. die IP-Adresse der Steuerung, auf die Sie Ihre Applikation laden möchten. Definieren Sie unter "Gateway IP-Adresse" die Adresse des Rechners, auf dem der Gateway-Server läuft. Voreingestellt erscheint hier "localhost" als Rechnername (Adresse). Dies bedeutet, dass der lokal installierte Gateway angesprochen wird. Diese Einstellung sollte nicht verändert werden.
- Verbindungstest** Klicken Sie auf die Schaltfläche "Verbindungstest", um zu überprüfen, ob die Kommunikation zwischen Rechner und der Zielsteuerung erfolgreich ist. Das Ergebnis des Tests wird im Feld "Ergebnis des Verbindungstests" protokolliert.

Erste Schritte - IndraLogic

Bei erfolgreichem SPS-Kommunikationstest wird die IP-Adresse angezeigt. Eventuell auftretende Probleme werden im Fenster "Ergebnis des Verbindungstests" durch blinkende rote oder gelbe Icons gekennzeichnet.

Die folgenden Probleme können auftreten:

- Ausgabe: "Kommunikation zur Steuerung nicht erfolgreich."
Ursachen:
 - Die Steuerung ist nicht eingeschaltet.
 - Das Netzkabel ist nicht angeschlossen.
 - Die IP-Adresse der Steuerung stimmt nicht mit der eingestellten IP-Adresse überein.
Lösung: IP-Adresse im Dialog ändern oder IP-Adresse der Steuerung ändern (s. Rexroth IndraMotion MLC Funktionsbeschreibung, Inbetriebnahme der Steuerung).
 - Die Netzwerkmaske des PCs und/oder der Steuerung ist zu klein eingestellt.
Lösung: Netzwerkeinstellungen überprüfen und korrigieren (s. Rexroth IndraMotion MLC Funktionsbeschreibung, Inbetriebnahme der Steuerung).
- Ausgabe: "Die projektierte Firmware stimmt nicht mit der Firmware in der Steuerung überein."
Ursachen:
 - Die Firmware der Steuerung ist nicht aktuell.
Lösung: Firmwareupdate durchführen (s. Rexroth IndraMotion MLC Funktionsbeschreibung, Inbetriebnahme der Steuerung).
 - Die ausgewählte Steuerung entspricht nicht dem realen Steuerungstyp.

Sicherer Online-Modus Ist diese Option aktiv, erfolgt vor sicherheitsrelevanten Aktivitäten eine Rückfrage mit Bitte um Bestätigung.

SPS-Programmiersprache Hier wird die bevorzugte SPS-Programmiersprache eingestellt. Wählen Sie hier ST (strukturierter Text) aus.
Klicken Sie auf die Schaltfläche "Weiter >>" um zur nächsten Seite zu gelangen.

4.5.3 Funktionsmoduleinstellungen



Der Dialog "Geräte- und Funktionsmoduleinstellungen" kann sich, je nach Gerätetyp, unterscheiden. In diesem Projekt wird exemplarisch der Dialog des Gerätes "IndraLogic XLC L65" beschrieben.

Geben Sie die gewünschten Geräte- und Funktionsmoduleinstellungen ein.

Erste Schritte - IndraLogic

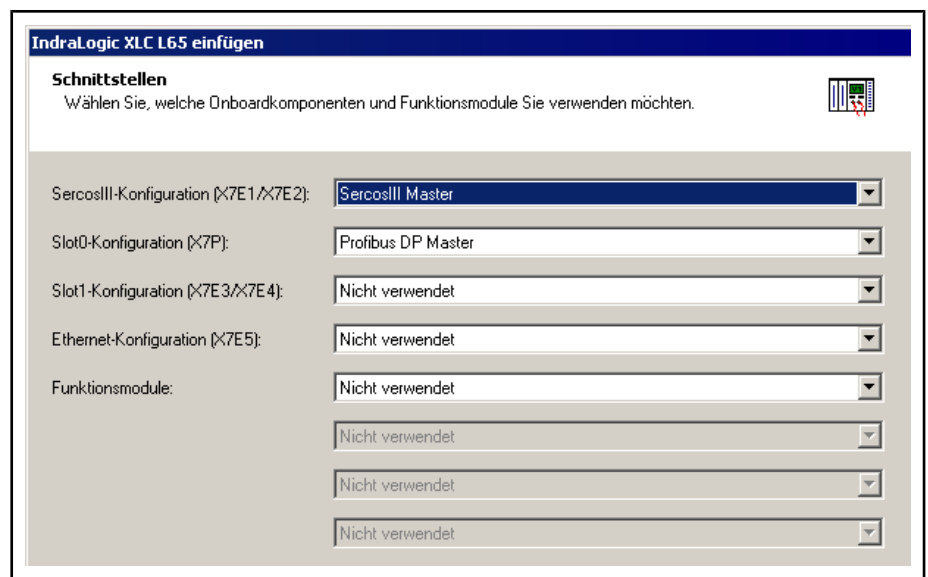


Abb.4-8: Funktionsmoduleinstellungen

- Sercos III-Konfiguration** Aktuelle SERCOS III-Konfiguration, die beim Einfügen des Gerätes gewählt werden kann. Zur Auswahl stehen "Sercos III-Master" oder "Nicht verwendet".
- Slot0-Konfiguration** Aktuelle Profibuskonfiguration, die beim Einfügen des Gerätes gewählt werden kann. Zur Auswahl stehen "PROFIBUS DP-Master", "PROFIBUS DP-Slave" oder "Nicht verwendet".
- Slot1-Konfiguration** Aktuelle PROFINET- oder EtherNet/IP-Konfiguration, die beim Einfügen des Gerätes gewählt werden kann. Zur Auswahl stehen "Nicht verwendet", bzw. alternativ eine PROFINET- oder EtherNet/IP-Schnittstelle.
- EtherNet-Konfiguration** Da Sie diese Schnittstelle als Engineering-Schnittstelle benötigen (Verbindung zu Ihrem PC direkt oder über das Betriebsnetz) darf der Ethernet/IP adapter nicht aktiviert werden.
- Funktionsmodul 1-4** An der "IndraLogic XLC L65" können bis zu vier Funktionsmodule angekoppelt werden.
Bei Bedarf können Sie aus der jeweiligen Liste auswählen.
Bestätigen Sie die Eingaben mit der Schaltfläche "Fertigstellen".
Im Projektextplorer erscheint das neu angelegte Gerät.

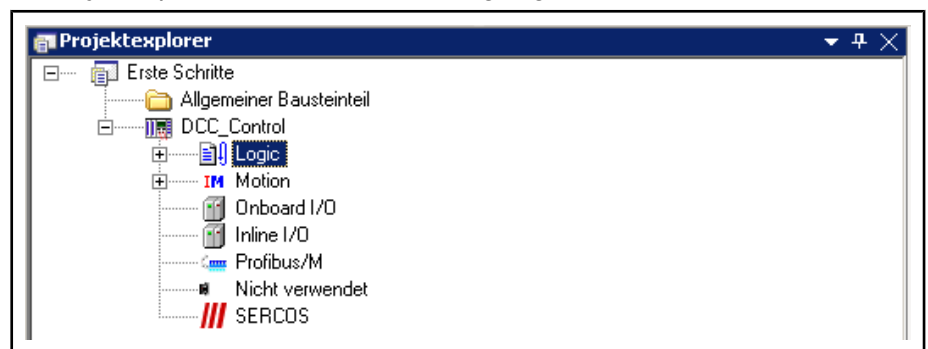


Abb.4-9: Projektextplorer: Projekt "Erste Schritte" mit einer IndraLogic XLC L65-Steuerung mit SERCOS III-Bus



Das Bild im Projektnavigator widerspiegelt die in den Dialogen "Konfiguration" und "Funktionsmodule" gewählten Komponenten.

4.6 SPS-Programm erstellen

4.6.1 Die Steuerung einer Ampelanlage

Hier soll ein Beispielprogramm "Mini-Ampelanlage" beschrieben werden. Das Programm soll zwei Ampeln an einer Kreuzung steuern. Beide Ampeln werden sich in ihren Rot- und Grünphasen abwechseln. Um Unfälle zu vermeiden, sollen zwischen den Rot- und Grünphasen auch noch Gelbumschaltphasen oder Gelb- und Rot-Umschaltphasen vorgesehen werden.

In diesem Beispiel werden Sie sehen:

1. Wie sich zeitabhängige Programme mit den Sprachmitteln der IEC 61131-3 darstellen lassen.
2. Wie Sie mit Hilfe von IndraLogic die verschiedenen Sprachen der Norm editieren können.
3. Wie Sie die erstellten Programmtteile miteinander verbinden können.

4.6.2 Programme und Funktionsbausteine anlegen

Programm "PlcProg" in "CFC" oder in "ST" einfügen

Die POU (Programmorganisationseinheit/ Program Organization Unit) "MlcProg" und "PlcProg" werden voreingestellt mit der Sprache "Strukturierter Text (ST)" angelegt und sind bereits nach dem Anlegen einer L65 vorhanden. "MlcProg" wird in unserem Projekt nicht benötigt, stört aber auch nicht.



Für das Programm "PlcProg" können Sie wählen:

- Erstellung in "Strukturierter Text (ST)", überspringen Sie den Rest des Kapitels und setzen fort mit [Programm SEQUENCE in AS einfügen, Seite 34](#).
- Erstellung in "Continuous Function Chart (CFC)", setzen Sie im Text fort.

Hierzu gehen Sie folgendermaßen vor:

1. Markieren Sie im Projektextplorer die POU "PlcProg".
2. Wählen Sie im Kontextmenü, durch klicken auf die rechte Maustaste, den Menüpunkt **Löschen** aus.



Die markierte POU "PlcProg" kann auch mit der Taste <Entf> gelöscht werden.

Fügen Sie nun eine neue POU in Ihr Projekt ein, indem Sie die POU aus der Bibliothek "SPS Objekte/ Allgemeine Objekte" auf den Knoten "Application" ziehen. Alternativ können Sie im Kontextmenü des Applikationsknoten den Befehl **Hinzufügen ► POU** verwenden:

Erste Schritte - IndraLogic

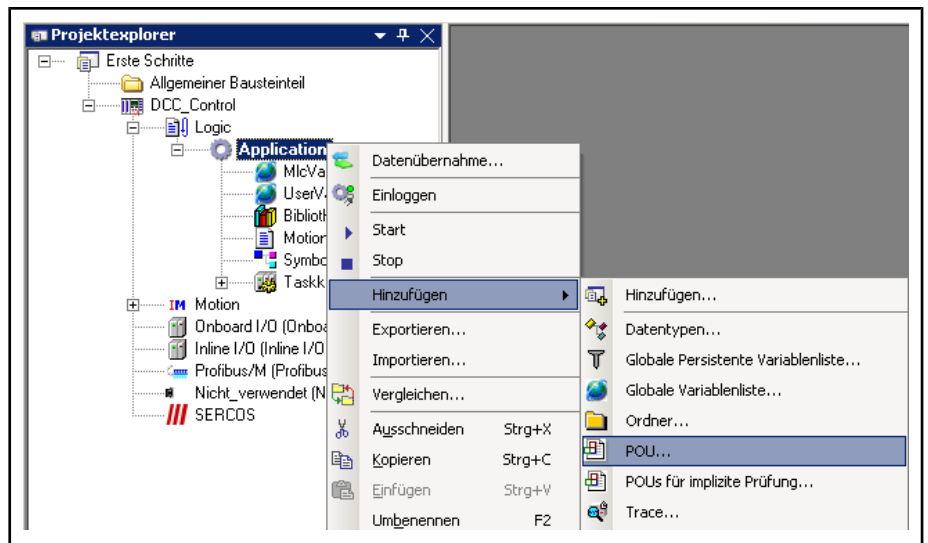


Abb.4-10: POU aus Bibliothek einfügen

Der Dialog "Objekt hinzufügen" wird geöffnet. Im Dialog "Objekt hinzufügen" nehmen Sie folgende Einstellungen vor:

1. Geben Sie der POU im Eingabefeld "Name:" den Namen "PlcProg".
2. Wählen Sie unter "Typ:" im Auswahlbereich "Programm" und als "Implementierungssprache:" (Programmiersprache) "Continuous Function Chart (CFC)" aus.

Bestätigen Sie mit "Fertigstellen".

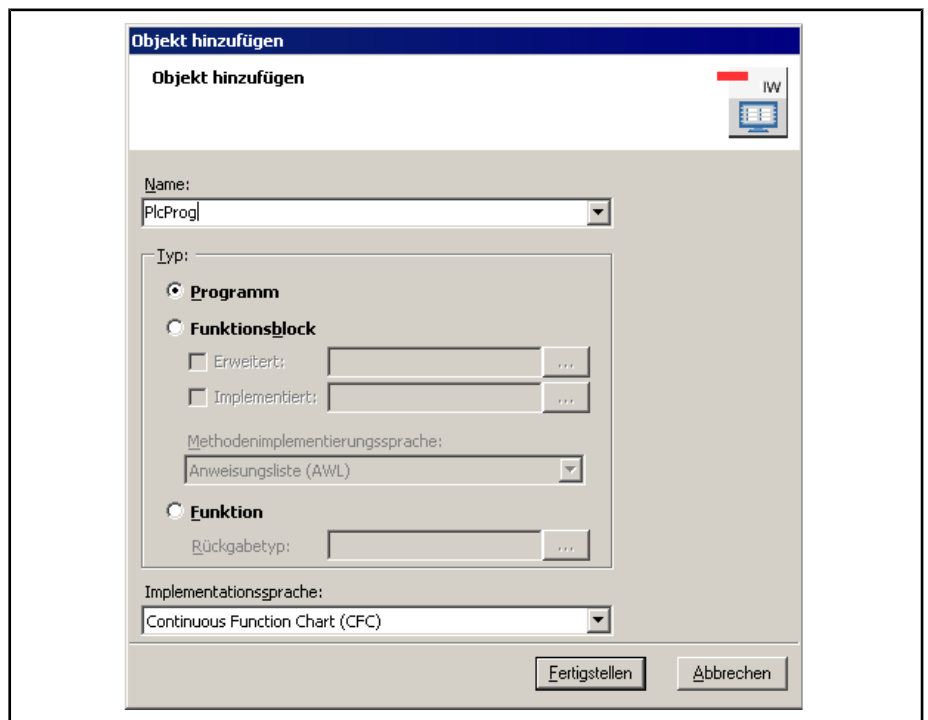


Abb.4-11: POU "PlcProg" konfigurieren

Das Objekt POU "PlcProg", Sprache "CFC" ist nun im Projekt verfügbar.

Erste Schritte - IndraLogic

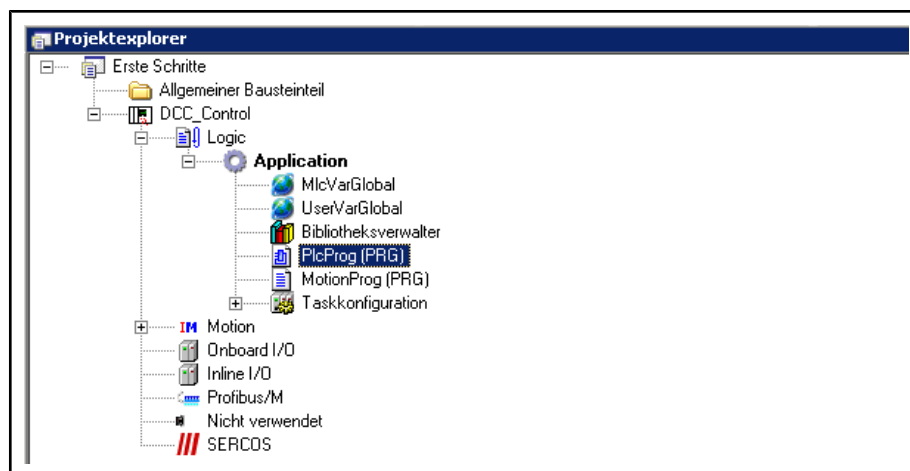


Abb.4-12: POU "PlcProg" in die Applikation eingefügt

Funktion von "PlcProg"

In "PlcProg" wird ein eingehendes Startsignal mit dem Ampelphasenablauf gekoppelt und die Ausgänge für die Ansteuerung der einzelnen Farben der Ampel angesteuert.

Programm "SEQUENCE" in "AS" einfügen

Markieren Sie "Application" und klicken Sie die rechte Maustaste. Wählen Sie aus dem Kontextmenü **Hinzufügen ► POU**.

Der Dialog "Objekt hinzufügen" wird geöffnet. Im Dialog "Objekt hinzufügen" nehmen Sie folgende Einstellungen vor:

1. Geben Sie der POU im Eingabefeld "Name:" den Namen "SEQUENCE".
2. Wählen Sie unter "Typ:" im Auswahlbereich "Programm" und als "Implementationsprache:" (Programmiersprache) "Ablaufsprache (AS)" aus.
3. Bestätigen Sie mit "Fertigstellen".

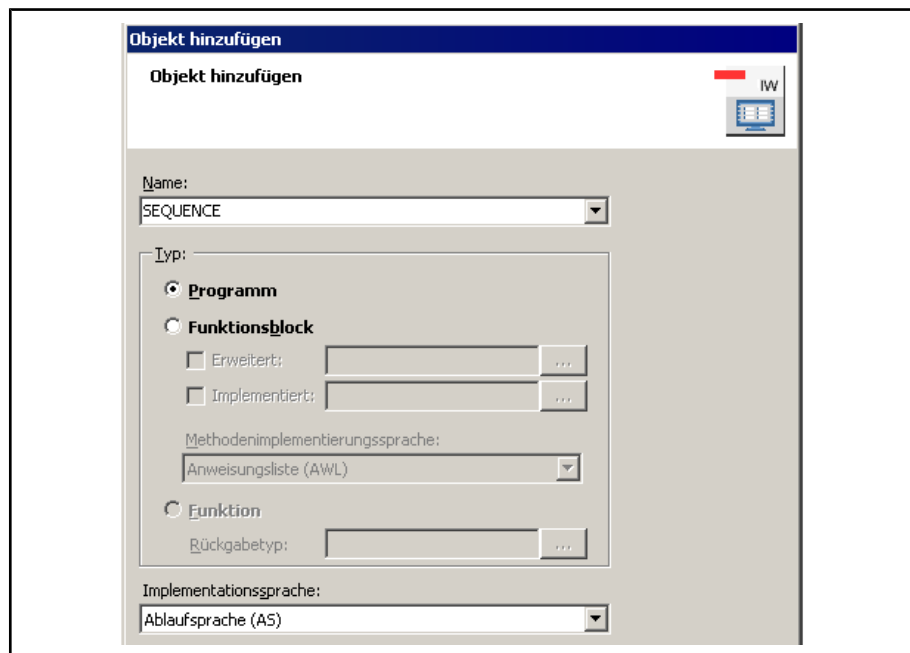


Abb.4-13: Programm "SEQUENCE" hinzufügen

Das Objekt Programm "SEQUENCE" ist nun im Projekt verfügbar.

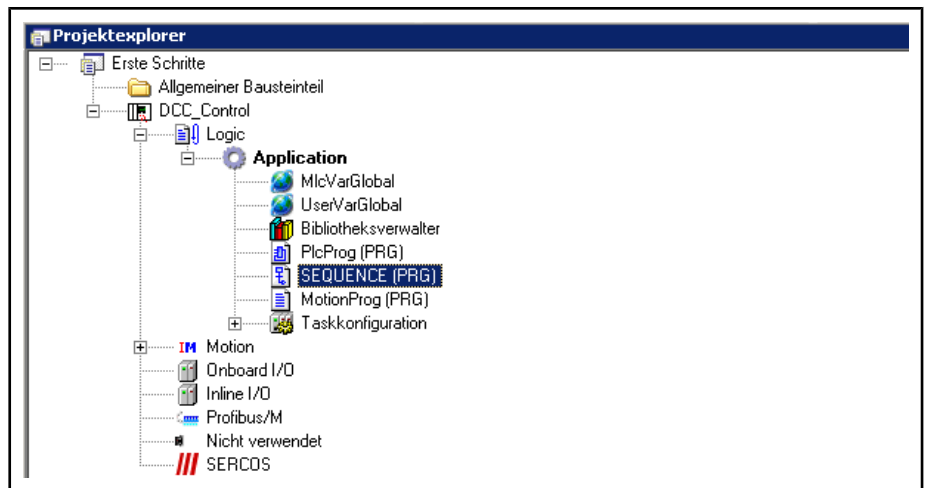


Abb.4-14: Programm "SEQUENCE" als Teil der Applikation

Funktion von "SEQUENCE"

Im Programm "SEQUENCE" wird der Ablauf der Ansteuerungsphasen der Ampel programmiert. Hierbei wird z.B. der Baustein "WAIT" aufgerufen oder Werte für die einzelnen Ampelphasen im Baustein "TRAFFICSIGNAL" gespeichert, so dass das richtige Ampellicht zur richtigen Zeit und mit der gewünschten Dauer leuchtet.

Funktionsbaustein "TRAFFICSIGNAL" in "FUP" einfügen

Markieren Sie "Application" und klicken Sie die rechte Maustaste. Wählen Sie aus dem Kontextmenü **Hinzufügen ► POU**.

Der Dialog "Objekt hinzufügen" wird geöffnet. Im Dialog "Objekt hinzufügen" nehmen Sie folgende Einstellungen vor:

1. Geben Sie der POU im Eingabefeld "Name:" den Namen "TRAFFICSIGNAL".
2. Wählen Sie unter "Typ:" im Auswahlbereich "Funktionsbaustein" und als "Implementationssprache:" (Programmiersprache) "Funktionsplan (FUP)" aus.
3. Bestätigen Sie mit "Fertigstellen".

Erste Schritte - IndraLogic

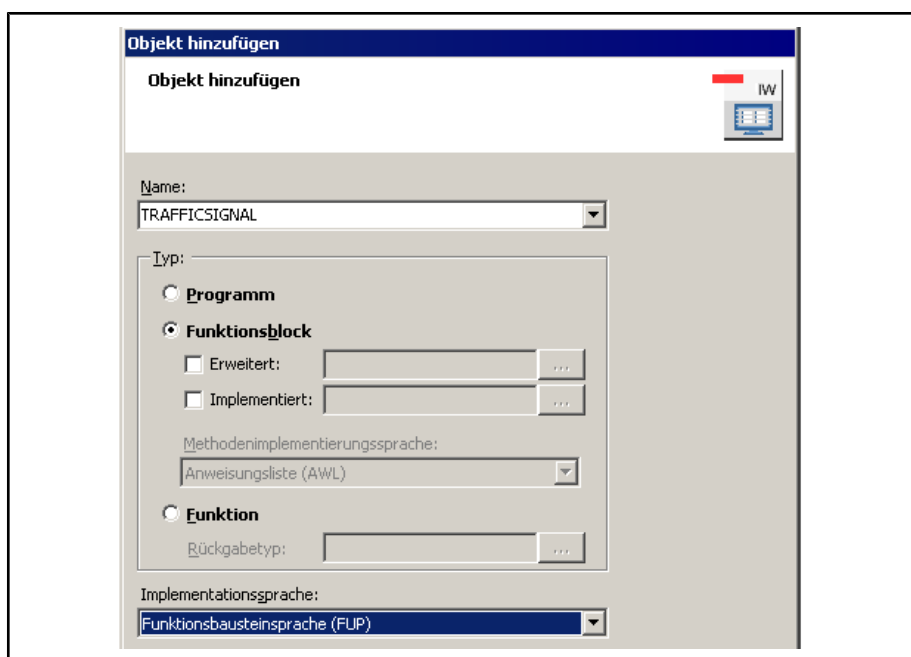


Abb.4-15: Funktionsbaustein "TRAFFICSIGNAL" hinzufügen

Die POU "TRAFFICSIGNAL" vom Typ "Funktionsbaustein" erscheint nun im Projektbaum unter der Applikation.

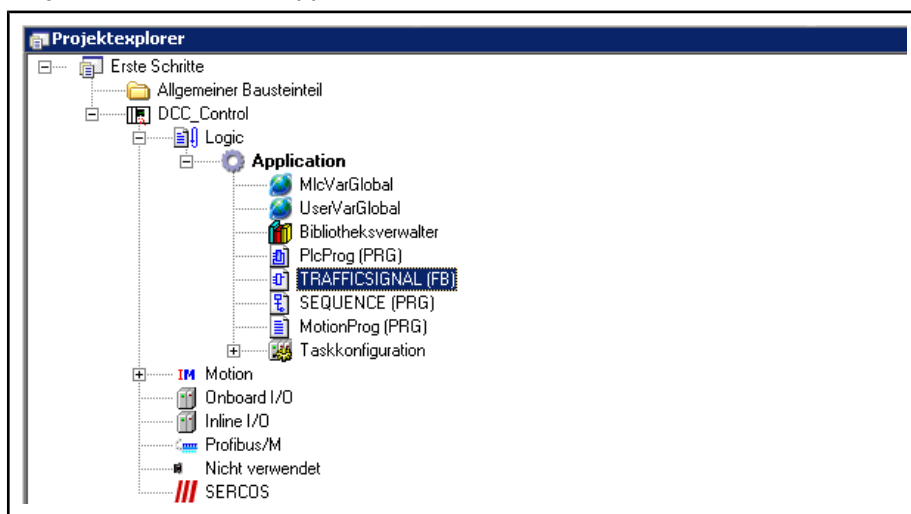


Abb.4-16: Funktionsbaustein "TRAFFICSIGNAL" als Teil der Applikation

Funktion von "TRAFFICSIGNAL"

Im Funktionsbaustein "TRAFFICSIGNAL" werden die einzelnen Ampelphasen den Ampellampen zugeordnet. Die rote Lampe der Ampel leuchtet bei der roten Ampelphase oder bei der gelb-roten Ampelphase. Die gelbe Lampe leuchtet bei der gelben Ampelphase oder bei der gelb-roten Ampelphase. Die grüne Lampe leuchtet bei der grünen Ampelphase. Siehe auch nachfolgende Tabelle.

Ampelphase	leuchtet Rot	leuchtet Gelb	leuchtet Grün
Rot	X		
Gelb-Rot	X	X	
Gelb		X	
Grün			X

Abb.4-17: Zuordnung Ampelphasen

Funktionsbaustein "WAIT" in "ST" einfügen

Als letzten Baustein fügen Sie eine POU mit dem Namen "WAIT" ein.

Markieren Sie "Application" und klicken Sie die rechte Maustaste. Wählen Sie aus dem Kontextmenü **Hinzufügen ► POU**.

Der Dialog "Objekt hinzufügen" wird geöffnet. Im Dialog "Objekt hinzufügen" nehmen Sie folgende Einstellungen vor:

1. Geben Sie der POU im Eingabefeld "Name:" den Namen "WAIT".
2. Wählen Sie unter "Typ:" im Auswahlbereich "Funktionsbaustein" und als "Implementationssprache:" (Programmiersprache) "Strukturierter Text (ST)" aus.
3. Bestätigen Sie mit "Fertigstellen".

Siehe nachfolgende Abbildung.

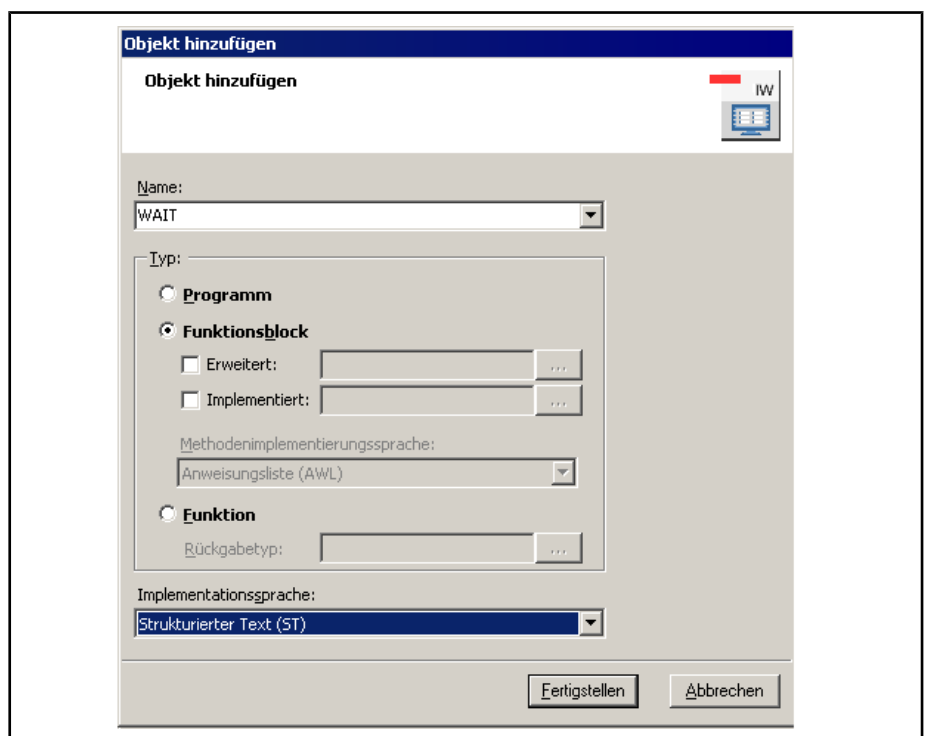


Abb.4-18: Funktionsbaustein "WAIT" hinzufügen

Die POU "WAIT" vom Typ "Funktionsbaustein" ist nun Bestandteil Ihrer Applikation.

Erste Schritte - IndraLogic

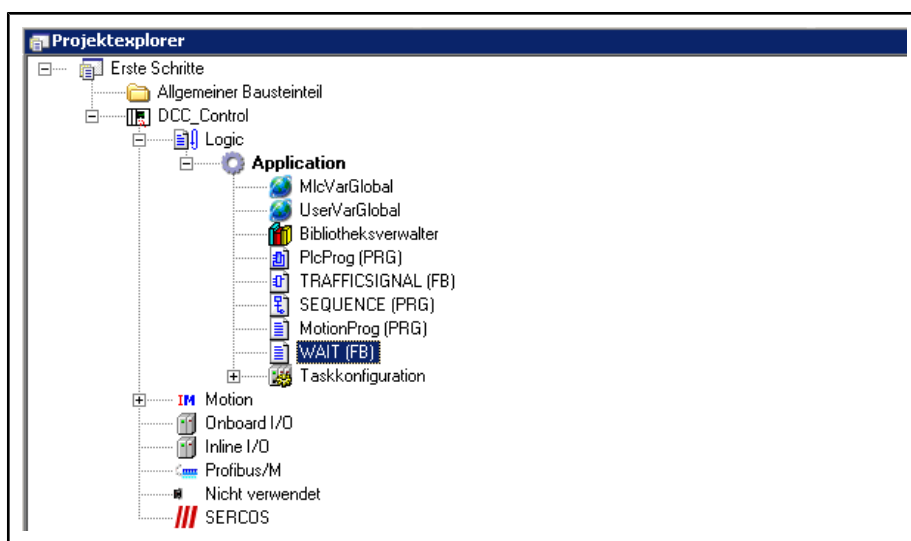


Abb.4-19: Funktionsbaustein "WAIT" als Teil der Applikation

Funktion von "WAIT"

Im Baustein POU "WAIT" wird eine einfache Zeitstufe programmiert, der einen Baustein "TP" aus der Standardbibliothek (Standard.library) von IndraWorks benutzt. Mit diesem Timer kann man die Länge jeder Ampelphase angeben. Als Eingangsvariable bekommt der Baustein die Dauer der Ampelphase in Millisekunden und liefert als Ausgangsvariable den booleschen Wert "TRUE", sobald die Zeit abgelaufen ist.

4.6.3 Programme und Funktionsbausteine editieren – erster Teil

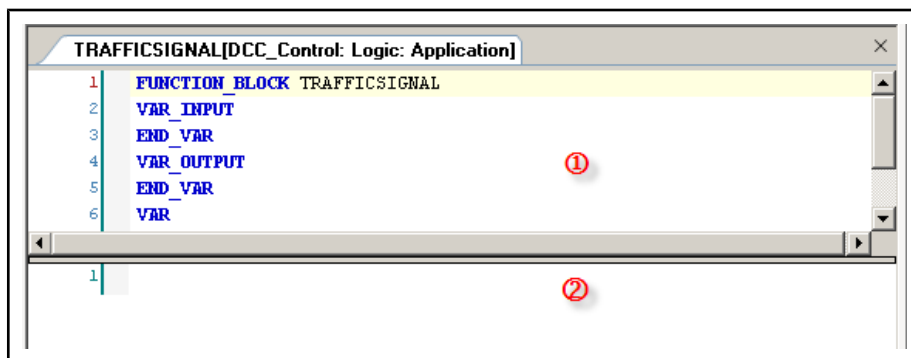
Funktionsbaustein "TRAFFICSIGNAL" in "FUP" editieren

Öffnen der POU

Öffnen Sie den Funktionsbaustein "TRAFFICSIGNAL" indem Sie sich mit dem Mauszeiger auf die POU "TRAFFICSIGNAL" zeigen und mit der rechten Maustaste das Kontextmenü öffnen. Wählen Sie im Kontextmenü **Öffnen** aus. Der Editor des Funktionsbausteins "TRAFFICSIGNAL" wird geöffnet, siehe nachfolgende Abbildung "Editor des Funktionsbausteins 'TRAFFICSIGNAL' geöffnet".



Alternativ können Sie den Editor einer POU auch per Doppelklick öffnen.



- (1) Deklarationsteil
- (2) Implementationsteil oder Rumpf

Abb.4-20: Editor des Funktionsbausteins "TRAFFICSIGNAL" geöffnet

Der POU Editor

Der Editor ist in zwei Bereiche gliedert:

1. in den Deklarationsteil und

2. den Implementationsteil

Im Deklarationsteil werden die Ein- und Ausgangsvariablen des Funktionsbausteins und lokalen Variablen angelegt.

Im Implementationsteil wird das Anwenderprogramm editiert.

Variablen deklarieren

Es gibt zwei Möglichkeiten in einer POU Variablen zu deklarieren:

1. Die Variablen direkt im Deklarationsteil anlegen.
2. Mit der Programmierung ohne Deklaration beginnen und im Eingabefenster die durch das System ermittelten Daten ergänzen (empfohlen).

Das Eingabefenster öffnet sich automatisch, wenn die eingegebene Variable dem Programm nicht bekannt ist.



1. Instanzen von Funktionsbausteinen, von Strukturen (STRUCTs) und von Datenfeldern (ARRAYs) sollten Sie direkt im Deklarationsteil anlegen.
2. Variablen lassen sich einfacher während der Programmierung mit Unterstützung des Eingabefensters anlegen.
Achten Sie bitte auf die richtige Zuordnung zum Gültigkeitsbereich und den richtigen Datentyp.
3. Wird Ihnen eine "bereits deklarierte" Variable noch einmal zur Deklaration angeboten, handelt es sich meist um einen Schreibfehler....

Anwenderprogramm erstellen

Der Funktionsbaustein "TRAFFICSIGNAL" bekommt als Eingangsvariable "STATUS" vom Typ "INTEGER", den das Programm verarbeiten soll. Die Ausgangsvariablen "GREEN", "YELLOW" oder "RED", für die entsprechenden Ampelphasen, werden auf "TRUE" gesetzt. Der Funktionsbaustein "TRAFFICSIGNAL" wird später in "PlcProg" aufgerufen.

Erstes Netzwerk editieren

Editiert wird das Anwenderprogramm im Anweisungsteil. Selektieren Sie das erste Netzwerk im Anweisungsteil, indem Sie rechts in den Editor neben die "1" klicken. Öffnen Sie mit der rechten Maustaste das Kontextmenü und wählen Sie **Bausteinanruf einfügen**. Alternativ wählen Sie im Menü **FUP/KOP/ AWL ► Bausteinanruf einfügen**.

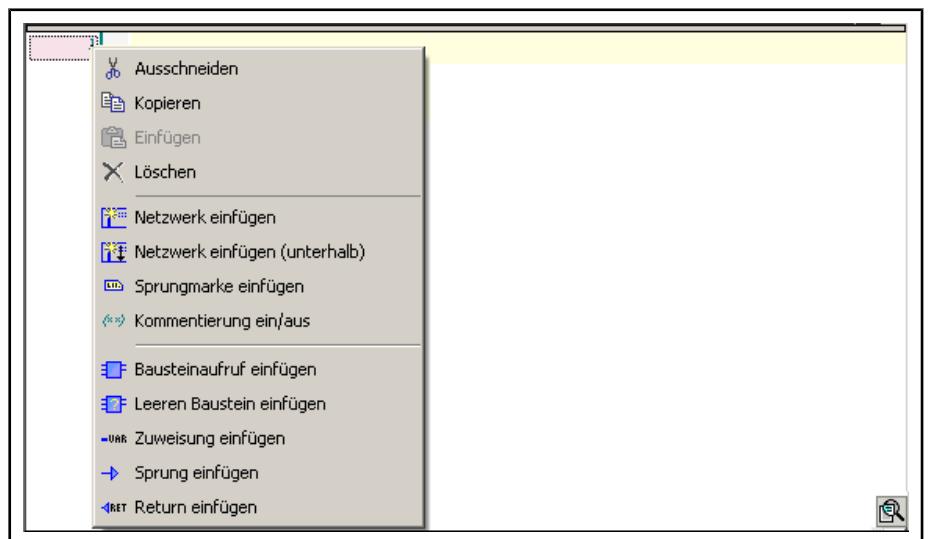


Abb.4-21: Bausteinanruf einfügen

Das Fenster "Eingabehilfe" wird geöffnet:

Erste Schritte - IndraLogic

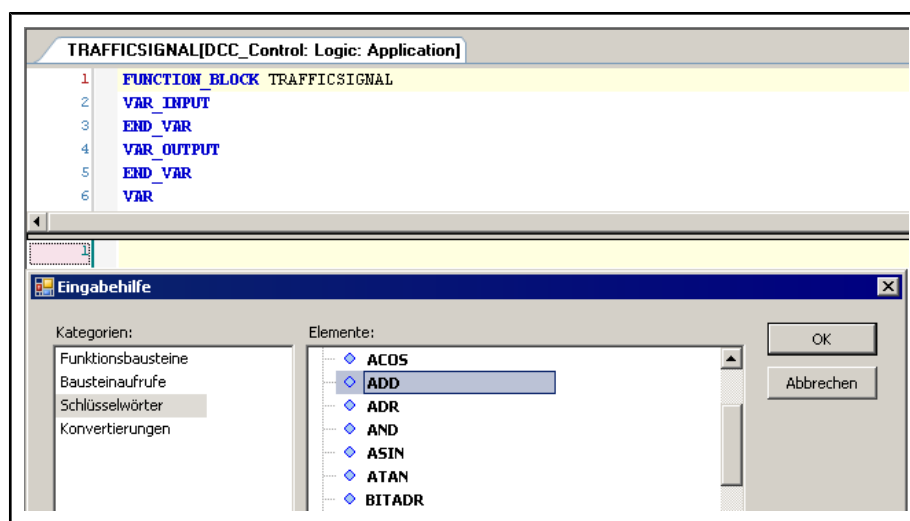


Abb.4-22: Eingabehilfe – Bausteinanruf einfügen "ADD"

Im ersten Netzwerk soll eine Box mit dem Operator "ADD" und zwei Eingängen angelegt werden.

1. Wählen Sie in der "Eingabehilfe" unter "Kategorien:" die "Schlüsselwörter" aus.
2. Markieren Sie unter "Elemente:" den Operator "ADD".
Bestätigen Sie mit "OK".

Der Anweisungsteil sieht folgendermaßen aus:

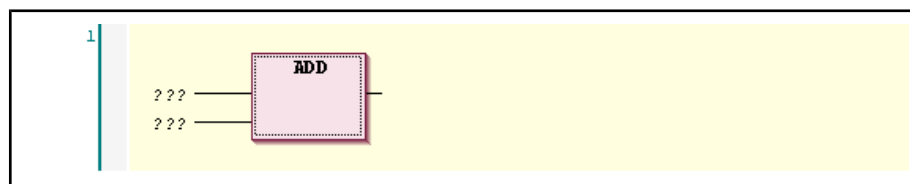


Abb.4-23: Baustein "ADD" in Netzwerk eingefügt

Die Aufgabe besteht nun darin, dass der "ADD"-Baustein einen anderen Namen und Funktion erhalten soll.

1. Klicken Sie hierzu mit dem Mauszeiger auf den Text "ADD", so dass dieser markiert erscheint, und ändern Sie den Text in "EQ" ab.
2. Markieren Sie nun, links am Baustein mit dem Namen "EQ", den oberen Eingang "???" und tragen Sie dort den Namen der Variable "STATUS" ein.

Das Fenster zur Deklaration von Variablen öffnet sich.

Vorgeschlagen wird: VAR, STATUS, INT.

3. Da "STATUS" eine Eingangsvariable des Funktionsbausteins ist, muss der Vorschlag in "VAR_INPUT, STATUS, INT" geändert werden.

Erste Schritte - IndraLogic

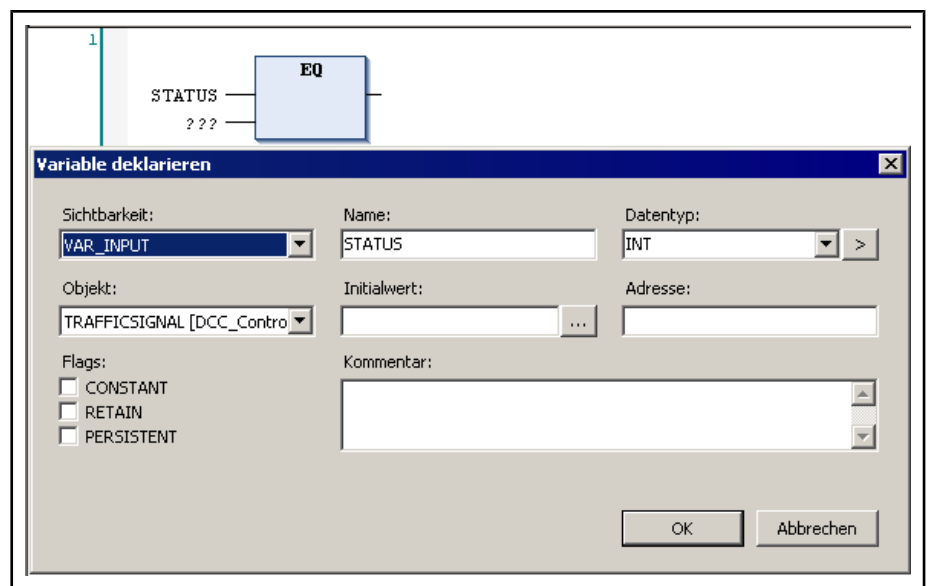


Abb.4-24: Deklaration der Eingangsvariablen "STATUS"

Im Deklarationsteil des Funktionsbausteines erscheint im Bereich "VAR_INPUT...END_VAR" die Deklarationszeile STATUS: INT;.

4. Am unteren Eingang "???" geben Sie "1" ein.

"STATUS" wird vier mögliche Zustände für die verschiedenen Ampelphasen haben. Diese Zustände lauten "Grün", "Gelb", "Gelb-Rot" und "Rot".

Die vier Zustände sollen als Ausgangsvariable decodiert werden. Für die vier Zustände werden drei Variablen "GREEN", "YELLOW" und "RED" benötigt.

1. Klicken Sie nun im Netzwerk "1" in einen Bereich hinter den Baustein "EQ", um dem Baustein eine Zuweisung anzufügen. Hierzu klicken Sie auf die rechte Maustaste und wählen im Kontextmenü **Zuweisung einfügen** aus. Klicken Sie auf den Ausgang des Bausteins "???" und geben ihm den Variablennamen "GREEN".

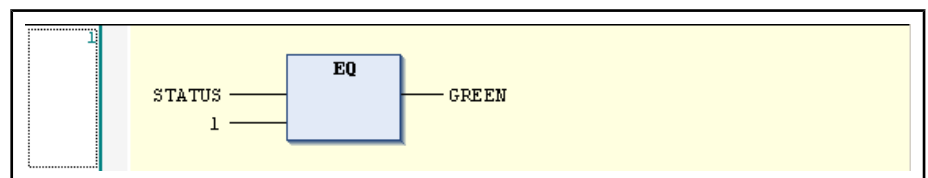


Abb.4-25: Baustein "EQ"

Das Fenster zur Deklaration von Variablen öffnet sich.

2. Da "GREEN" eine Ausgangsvariable des Funktionsbausteins ist, muss der Vorschlag in "VAR_OUTPUT, GREEN, BOOL" geändert werden.

Im Deklarationsteil des Funktionsbausteines erscheint im Bereich "VAR_OUTPUT...END_VAR" die Deklarationszeile GREEN: BOOL;.

Funktion von "EQ" Die Eingangs-Variable "STATUS" wird mit "1" verglichen. Das Ergebnis ist "TRUE", wenn "STATUS" den Wert "1" hat. Das Ergebnis ist "FALSE", wenn "STATUS" den Wert ungleich "1" hat. Das Ergebnis wird der Ausgangsvariable "GREEN" zugewiesen.

Netzwerk einfügen Nun muss noch die Auswertung der Eingangsvariable "STATUS" für die Ampelphasen "gelb" und "rot" realisiert werden. Hierzu müssen zwei weitere Netzwerke eingefügt werden.

Erste Schritte - IndraLogic

Zweites Netzwerk editieren

1. Dazu klicken Sie in das erste Netzwerk und wählen über das Kontextmenü **Netzwerk einfügen (unterhalb)**.

Erstellen Sie einen Baustein "EQ" im zweiten Netzwerk.

1. Klicken Sie mit der rechten Maustaste in das zweite Netzwerk.
2. Wählen Sie im Kontextmenü **Bausteinaufruf einfügen**.
3. Wählen Sie in der "Eingabehilfe" zu "Bausteinaufruf einfügen" unter "Kategorien:" die "Schlüsselworte".
Wählen Sie unter "Elemente:" das Element "EQ" aus.
Bestätigen Sie mit der Schaltfläche "OK".
4. Markieren Sie den Ausgang des Bausteins "EQ" und klicken Sie auf die rechte Maustaste.
5. Wählen Sie im Kontextmenü **Bausteinaufruf einfügen**.
6. Wählen Sie in der "Eingabehilfe" zu "Bausteinaufruf einfügen" unter "Kategorien:" die "Schlüsselworte" aus.
Wählen Sie unter "Elemente:" das Element "OR" aus.
Bestätigen Sie mit der Schaltfläche "OK".

Für die weitere Beschaltung der Bausteins "OR" gehen Sie wie folgt vor:

1. Markieren Sie nun den Ausgang des Bausteins "OR" und klicken Sie auf die rechte Maustaste.
2. Fügen Sie am Ausgang über das Kontextmenü **Zuweisung einfügen**.
3. Geben Sie nun am Ausgang des Bausteins "???" den Namen der Ausgangsvariable "YELLOW" ein.
4. Komplettieren Sie den Eintrag im Fenster zur Deklaration der Variablen "YELLOW".

Im Deklarationsteil des Funktionsbausteines erscheint im Bereich "VAR_OUTPUT...END_VAR" die Deklarationszeile `YELLOW: BOOL;`

1. Selektieren Sie nun den zweiten (freien) Eingang des Bausteins "OR", indem Sie mit der rechten Maustaste rechts neben "???" direkt am Baustein auf den Anschluss klicken.
2. Wählen Sie, mit einem Klick auf die rechte Maustaste, aus dem Kontextmenü **Bausteinaufruf einfügen** aus.
3. Wählen Sie in der "Eingabehilfe" zu "Bausteinaufruf einfügen" unter "Kategorien:" die "Schlüsselworte" aus.
Wählen Sie unter "Elemente:" das Element "EQ" aus.
Bestätigen Sie mit der Schaltfläche "OK".

Die Eingänge der beiden Bausteine "EQ" müssen noch belegt werden:

1. Klicken Sie auf den ersten Eingang des oberen Bausteins "EQ" im zweiten Netzwerk.
2. Geben Sie "STATUS" ein und bestätigen Sie mit <Enter>.



Da "STATUS" bereits deklariert ist, erscheint das Deklarationsfenster dieses Mal nicht.

3. Klicken Sie auf den zweiten Eingang des selben Bausteins "EQ".
Geben Sie "2" ein und bestätigen Sie mit <Enter>.

Erste Schritte - IndraLogic

4. Klicken Sie auf den ersten Eingang des unteren Bausteins "EQ" im zweiten Netzwerk.
Geben Sie "STATUS" ein und bestätigen Sie mit <Enter>.
5. Klicken Sie auf den zweiten Eingang des selben Bausteins "EQ".
Geben Sie "4" ein und bestätigen Sie mit <Enter>.

Netzwerk kopieren

Das dritte Netzwerk erstellen Sie am einfachsten, indem Sie das zweite Netzwerk kopieren.

Um das Netzwerk zu kopieren gehen Sie wie folgt vor:

1. Markieren Sie das zweite Netzwerk und öffnen mit einem Klick auf die rechte Maustaste das Kontextmenü.
2. Wählen Sie im Kontextmenü **Kopieren** aus.
3. Öffnen Sie erneut das Kontextmenü mit einem Klick auf die rechte Maustaste
4. Wählen Sie **Einfügen** aus.

Das kopierte Netzwerk wird nun unter dem Netzwerk "2" mit der Nummer "3" eingefügt.

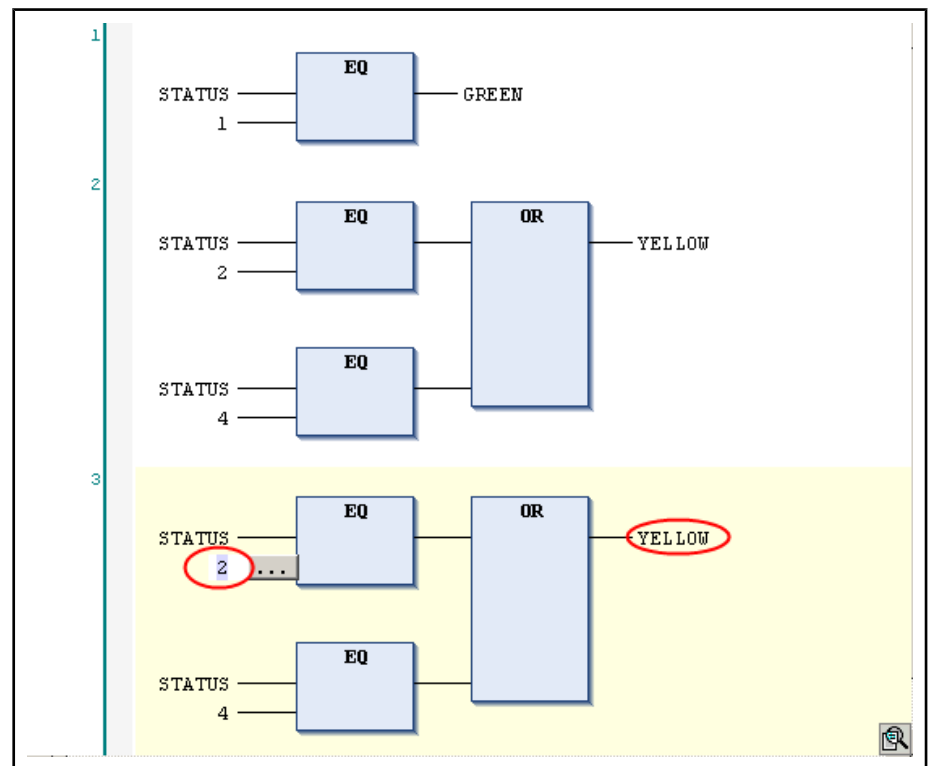


Abb.4-26: Drittes Netzwerk durch Kopie eingefügt

Drittes Netzwerk editieren

Um das dritte Netzwerk zu editieren, gehen Sie folgendermaßen vor:

1. Selektieren Sie in Netzwerk 3 am oberen "EQ-Baustein" den unteren Eingang mit "2".
2. Klicken Sie mit der linken Maustaste auf die "2" und geben Sie "3" ein.
3. Klicken Sie nun mit der linken Maustaste am Ausgang des Bausteins "OR" auf die Variablenzuweisung "YELLOW" und geben Sie "RED" ein.
4. Kompletieren Sie den Eintrag im Fenster zur Deklaration der Variablen "RED".

Erste Schritte - IndraLogic

Im Deklarationsteil des Funktionsbausteines erscheint im Bereich "VAR_OUTPUT...END_VAR" die Deklarationszeile RED: BOOL;.

Der erste Baustein POU "TRAFFICSIGNAL" ist nun fertiggestellt:

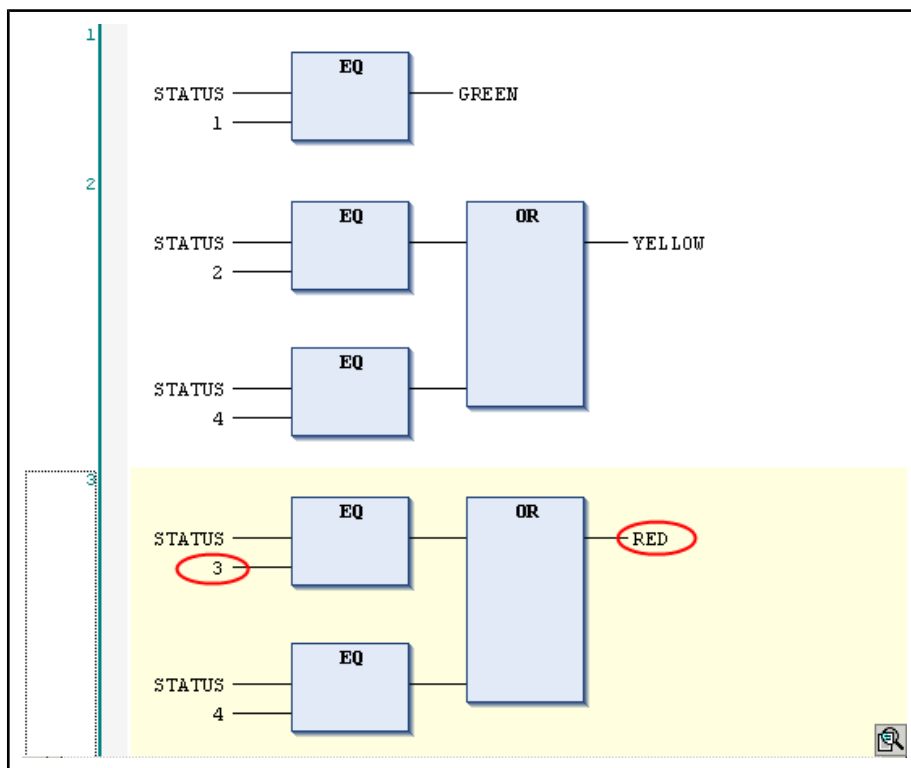


Abb.4-27: POU "TRAFFICSIGNAL" fertiggestellt

Beiläufig zur Implementation ist der Deklarationsteil entstanden:

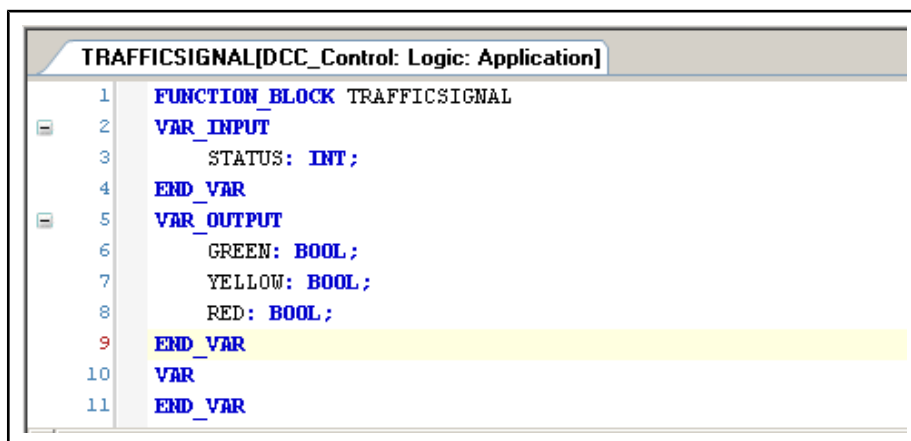


Abb.4-28: Deklarationsteil des FBs "TRAFFICSIGNAL"

Funktionsbaustein "WAIT" in "ST" editieren

Öffnen der POU

Öffnen Sie die POU "WAIT" durch Doppelklick auf das Objekt im Projektnavigator:

Variablen deklarieren und vorbelegen

Der Baustein "WAIT" erhält als Eingangsvariable eine Variable "WAITING-TIME" vom Typ "TIME". Als Ausgabe liefert der Baustein "WAIT" eine Variable "OK" vom Datentyp "BOOL".

Benötigt wird eine Instanz des Bausteins "TP" ZAB: TP;.

Erste Schritte - IndraLogic

Die aktuelle Zeit des Bausteines soll auf die Variable ETVar: TIME; geschrieben werden.

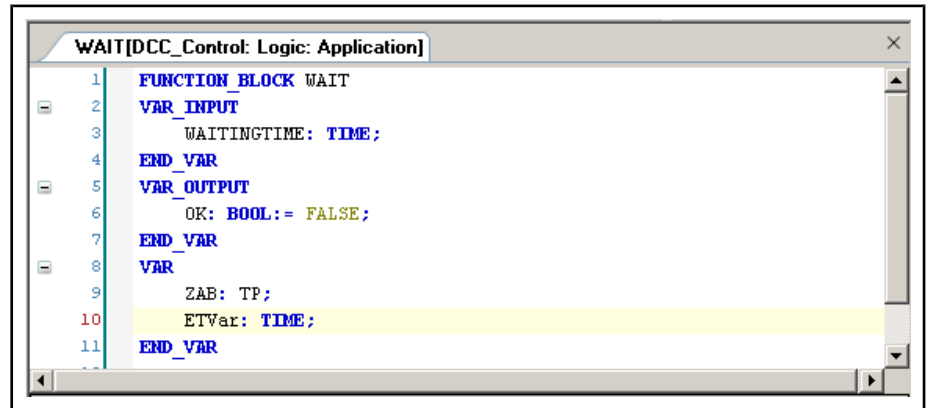


Abb.4-29: Variablendeklaration POU "WAIT"

Baustein "TP" aus Standard-Bibliothek

Der Baustein "TP" mit der Funktion als Pulsgeber befindet sich in der Standardbibliothek von IndraWorks. Um den Inhalt der Bibliothek "Standard" aus dem Bibliotheksverwalter einsehen zu können, gehen Sie wie folgt vor:

1. Öffnen Sie den "Bibliotheksverwalter" mit einem Doppelklick im Projektor.
2. Markieren Sie im "Bibliotheksverwalter" die Bibliothek "Standard".
3. Öffnen Sie die Baumstruktur des Ordners "Timer", indem Sie links neben dem Ordner "Timer" auf das "+" klicken.

Im Baum befindet sich der Baustein "TP". Klicken Sie auf den Baustein "TP", um weitere Informationen zu erhalten.

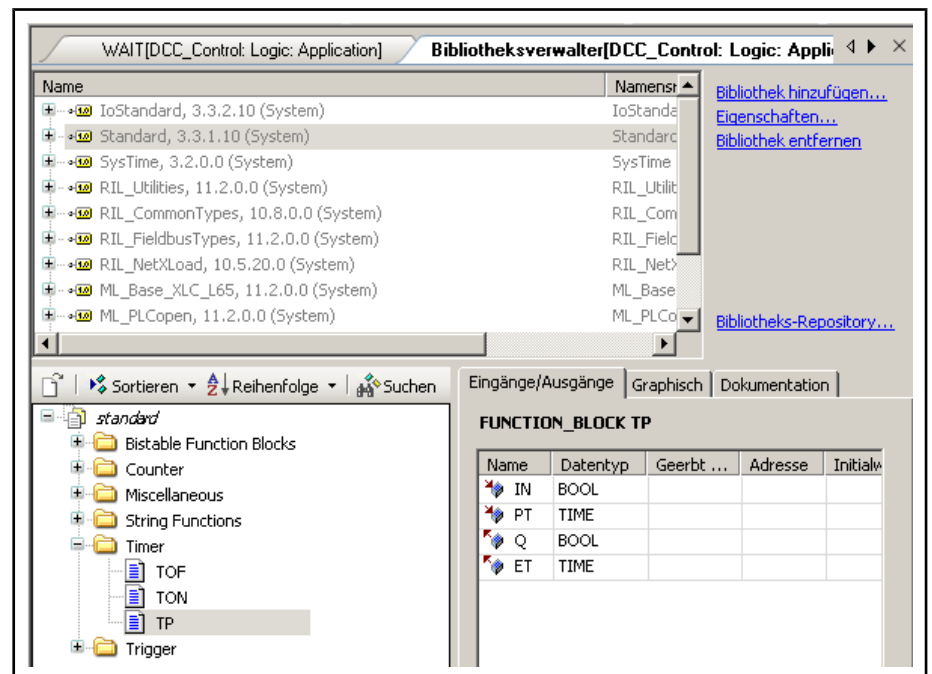


Abb.4-30: Bibliotheksverwalter Standardbaustein "TP"

Der Baustein "TP" hat zwei Eingänge ("IN" und "PT") und zwei Ausgänge ("Q" und "ET").

Erste Schritte - IndraLogic

Funktion von "TP"

Solange "IN" den booleschen Wert "FALSE" hat, ist "ET" = "0" und "Q" ist "FALSE".

Sobald "IN" den booleschen Wert "TRUE" liefert, wird am Ausgang "ET" die Zeit in Millisekunden aufaddiert.

Erreicht "ET" den Wert "PT", wird "ET" nicht mehr weiter gezählt.

"Q" liefert so lange "TRUE", wie "ET" kleiner als "PT" ist.

Sobald der Wert der Variablen "PT" erreicht ist, liefert "Q" den booleschen Wert "FALSE".



Die Ausgänge einer Funktionsbausteininstanz können entweder durch Kopieren auf eine Variable ausgewertet werden

ZAB(IN:=TRUE,PT:=WAITINGTIME,ET=>ETVar);

oder mittels Punktoperator

ZAB.Q.

Implementation erstellen in ST

Füllen Sie den Implementationsteil wie nachstehend gezeigt aus:

```

WAIT[DCC_Control: Logic: Application]
1  FUNCTION_BLOCK WAIT
2  VAR_INPUT
3    WAITINGTIME: TIME;
4  END_VAR
5  VAR_OUTPUT
6    OK: BOOL := FALSE;
7
8  IF ZAB.Q = FALSE THEN // Wenn Zeitstufe abgelaufen, erneut 1-0-1 triggern
9    ZAB(IN:=FALSE,PT:=WAITINGTIME,ET=>ETVar);
10   ZAB(IN:=TRUE,PT:=WAITINGTIME,ET=>ETVar);
11 ELSE
12   ZAB(); // Zeitstufe läuft weiter
13 END_IF
14 OK:= NOT ZAB.Q; // Warte, solange Zeitstufe läuft
  
```

Abb.4-31: POU "WAIT" in ST fertiggestellt

Funktion von "WAIT"

Für den Fall, das die Zeitstufe abgelaufen ist, wird sie mit einem 1-0-1 Signalwechsel an "IN" und der aktuellen Wartezeit "WAITINGTIME" neu gestartet. Andernfalls läuft die Zeitstufe unter Aktualisierung ihrer Ausgänge weiter.

Nach Ablauf der Zeitstufe (ZAB.Q = 0) wird der Ausgang "OK" für einen SPS-Zyklus "TRUE".

Programm "SEQUENCE" in "AS" editieren

Öffnen des Programms

Öffnen Sie die POU "SEQUENCE" durch Doppelklick auf das Objekt im Projektnavigator:

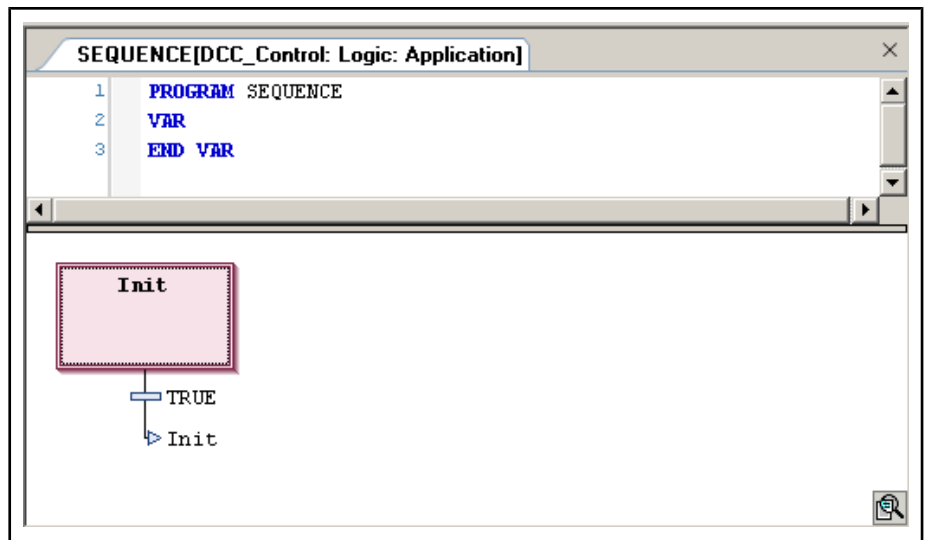


Abb.4-32: Editor Programm "SEQUENCE" geöffnet

Variablen deklarieren

Der Baustein "SEQUENCE" erhält als Eingangsvariable eine Variable "START" vom Typ "BOOL".

Als Ausgabe liefert der Baustein die Ausgangsvariablen "TRAFFICSIGNAL1", "TRAFFICSIGNAL2" vom Typ "INT" und als lokale Variable hat der Baustein die Variable "DELAY" vom Typ "WAIT". Es wird zunächst nur die Variable DELAY: WAIT; angelegt. Die anderen Variablen werden später mit der Eingabehilfe im Implementationsteil deklariert.

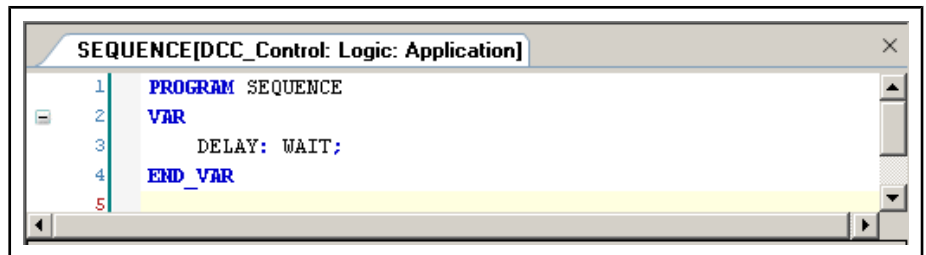


Abb.4-33: Variablendeklaration POU "SEQUENCE" - erster Teil

Anwenderprogramm erstellen

Am Anfang einer Ablaufkette steht im Rumpf des "AS-Editors" eine Aktion "Init", eine nachfolgende Transition "TRUE" und ein Rücksprung zu "Init". Siehe [Editor POU "SEQUENCE" geöffnet, Seite 47](#).

Struktur der Ablaufkette anlegen

Zunächst wird die Struktur der Ablaufkette festgelegt, indem die einzelnen Schritte und Transitionen eingefügt werden. Nach dem Einfügen der benötigten Schritte und Transitionen werden die einzelnen Aktionen zu den Schritten und Transitionen programmiert. Für jede Ampelphase wird ein Schritt benötigt. Um einen Schritt mit nachfolgender Transition einzufügen, gehen Sie wie folgt vor:

1. Markieren Sie Transition "TRUE" und öffnen Sie mit der rechten Maustaste das Kontextmenü.
2. Wählen Sie aus dem Kontextmenü **Schritt-Transition danach einfügen**.
3. Markieren Sie jeweils die letzte Transition und fügen Sie drei weitere Schritte und Transitionen ein.

Die Ablaufkette sieht nun, wie folgt, aus:

Erste Schritte - IndraLogic

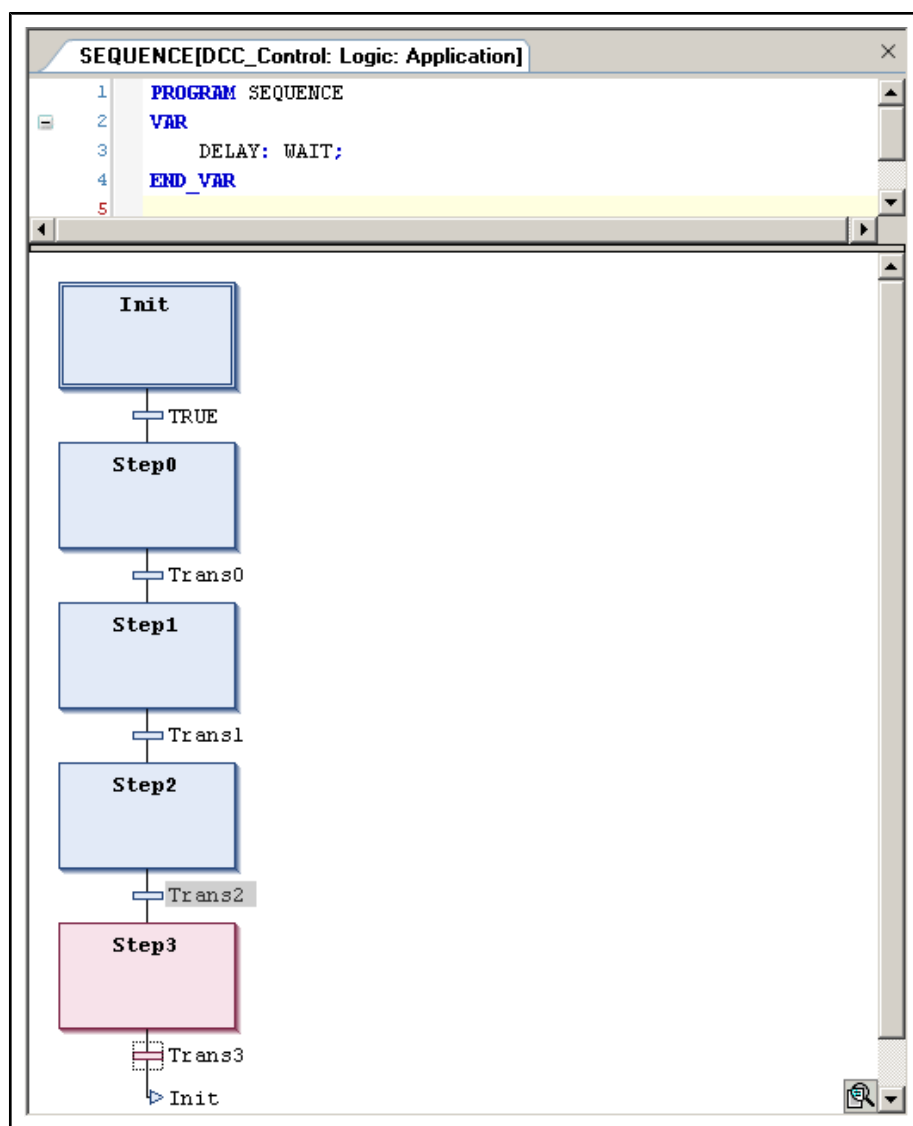


Abb.4-34: Programm "SEQUENCE" Schritte eingefügt



Alternativ können Sie Schritte und Transitionen auch über das Menü **SFC ► Schritt-Transition danach einfügen** einfügen.

Transitionen programmieren

Um Transitionen zu programmieren gibt es zwei Möglichkeiten:

1. Änderung des Transitionsnamens: Dies ist dann sinnvoll, wenn nur eine Weberschaltbedingung benötigt wird, z.B. bei der Abfrage einer Variablen.
2. Programmieren der Transition: Dies ist dann sinnvoll, wenn das Ergebnis verknüpfter Variablen als Weberschaltbedingung dienen soll.

Um eine Transition zu programmieren, muss eine Transition hinzugefügt werden. Hier mit Doppelklick auf die "Transition", vergeben für die Transition einen Namen und legen die Implementationssprache (Programmiersprache) fest, in der die Transition geschrieben werden soll. Die Transition kann dann, wie eine Funktion, in einem Editor programmiert werden.

Als erste Transition deklarieren Sie die Transition "START". Um die Transition "START" zu deklarieren, gehen Sie wie folgt vor:

Erste Schritte - IndraLogic

1. Klicken Sie auf den Transitionsnamen "TRUE" der ersten Transition.
 2. Geben Sie in das Eingabefeld "START" ein.
- Da die Variable "START" im Deklarationsteil noch nicht deklariert wurde, öffnet sich ein Fenster zum Deklarieren der Variablen.

Abb.4-35: Eingabehilfe zur Variablendeklaration

3. Wählen Sie bei "Datentyp:" den Datentyp "BOOL" aus.
4. Bei "Sichtbarkeit:" wählen Sie den Variablentyp "VAR_INPUT" aus.
5. Bestätigen Sie anschließend mit "OK".

Der ersten Transition wird die Variable "START" zugewiesen. Ist "START" = "TRUE", wird die nächste Aktion durchgeführt. Die Weiterschaltbedingung ist erfüllt.

Für die zweite und alle nachfolgenden Transitionen gehen Sie wie folgt vor:

1. Klicken Sie auf den Transitionsnamen.
2. Ändern Sie den Transitionsnamen in DELAY . OK ab.

Die übrigen Transitionen sind dann erfüllt, wenn die Ausgangsvariable "OK" von "DELAY" bzw. von dem Funktionsbaustein "WAIT" den Wert "TRUE" haben.

Die Ablaufkette sieht nun folgendermaßen aus:

Erste Schritte - IndraLogic

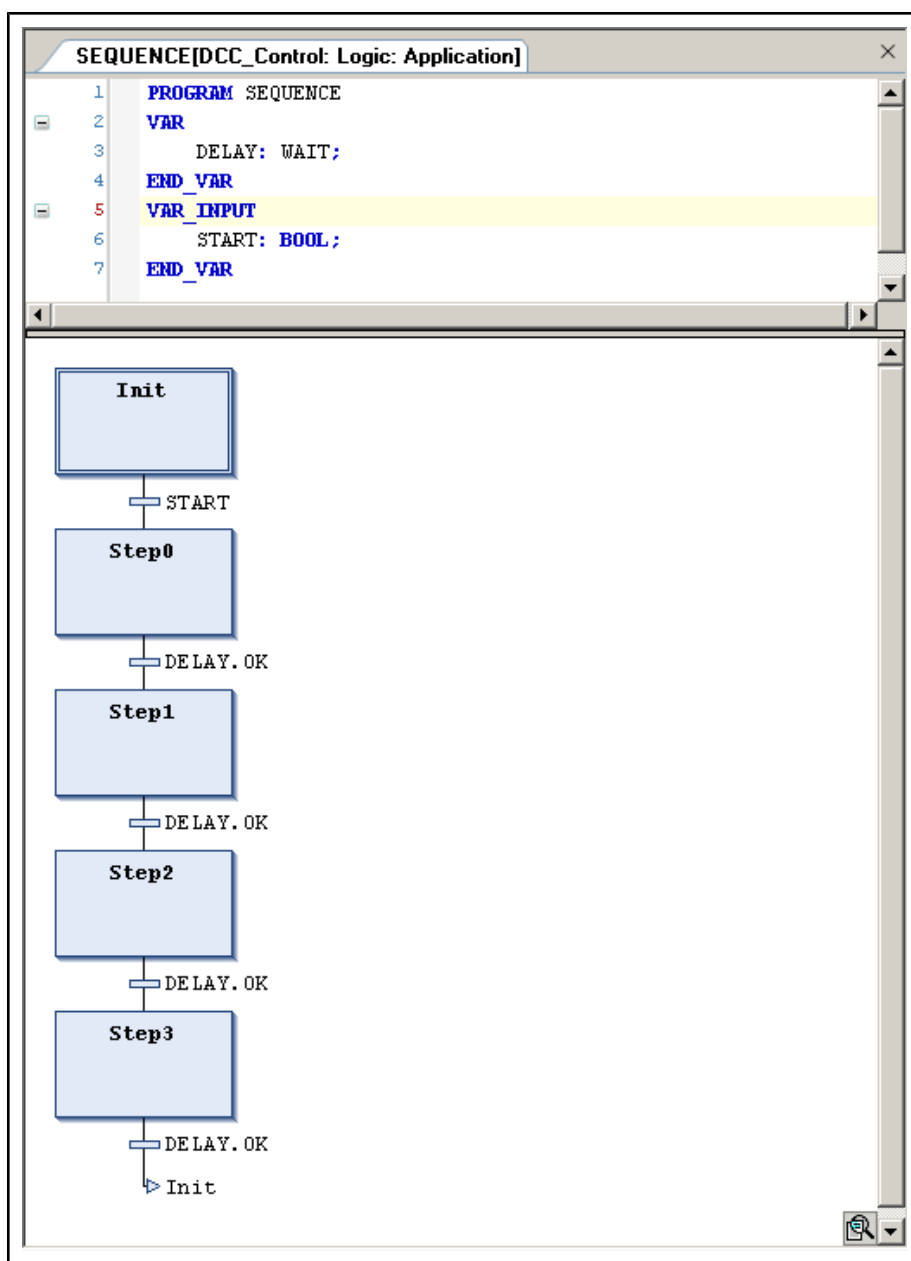


Abb.4-36: Schritte der Ablaufkette angelegt

Schritte benennen

Der Schritt "Init" behält als einziger Schritt seinen Namen bei. Die restlichen Schritte werden umbenannt. Sie sollen nacheinander die Namen "Switch1", "Green2", "Switch2" und "Green1" erhalten.

Zum Umbenennen der einzelnen Schritte gehen Sie wie folgt vor:

1. Klicken Sie auf den Namen des Schrittes.
2. Geben Sie den gewünschten Namen ein.
3. Bestätigen Sie mit der "Eingabetaste".

Funktion der Schritte

In den Schritten "Switch1" und "Switch2" werden die "Gelbphasen" realisiert. Im Schritt "Green1" wird die grüne Phase der Ampel 1 und bei "Green2" wird die grüne Phase der Ampel 2 realisiert.

Erste Schritte - IndraLogic

Rücksprungadresse abändern

Der Rücksprung nach dem Durchlaufen der Schritte soll nun nicht mehr nach "Init" erfolgen, sondern es soll zum Schritt "Switch1" gesprungen werden. Hierzu klicken Sie auf den Namen des Rücksprungs "Init" und geben dort das neue Rücksprungziel "Switch1" ein.

Die Ablaufkette sieht nun wie folgt aus:

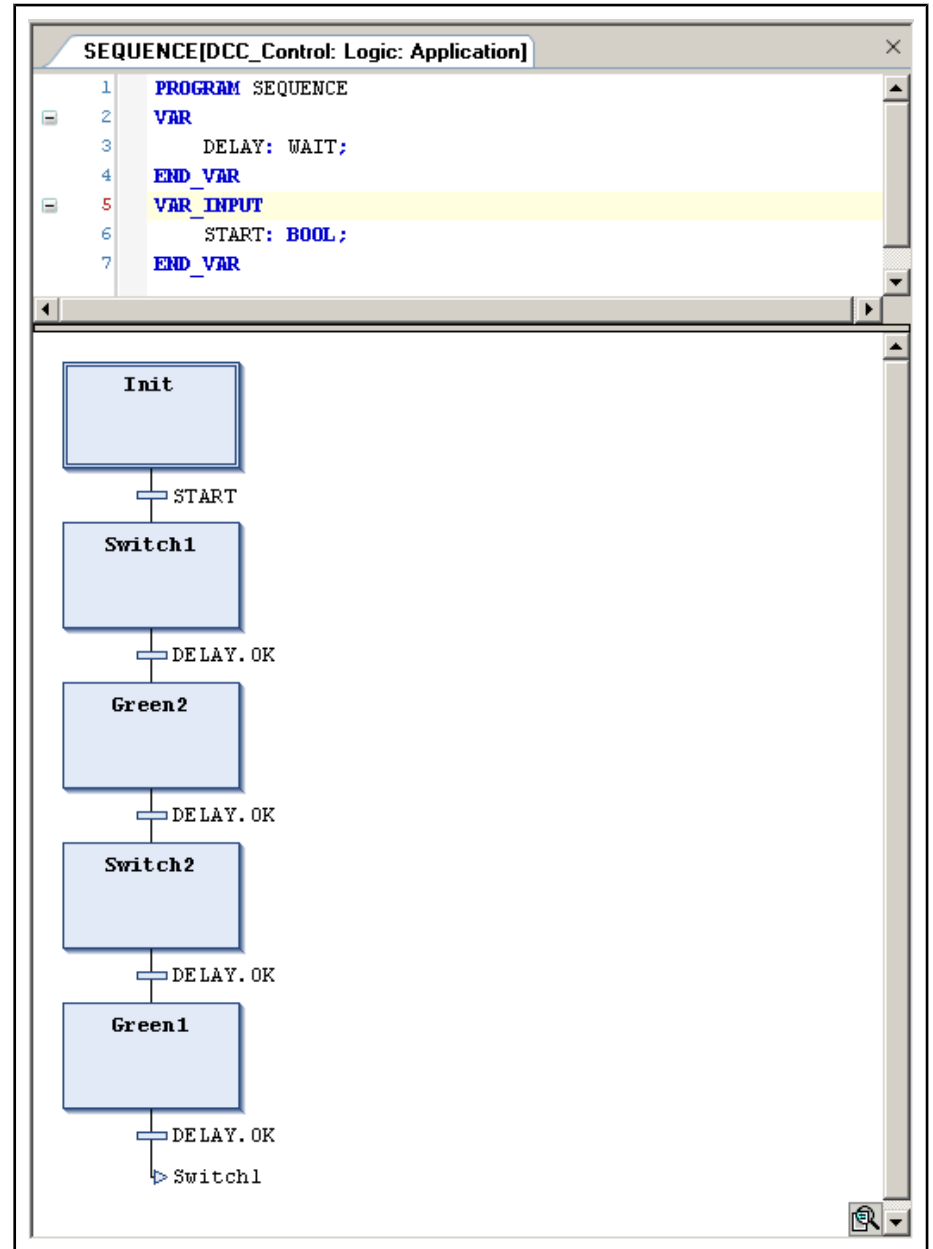


Abb.4-37: Schritte der Ablaufkette benannt

Schritte und Aktionen

Ein in Ablaufsprache geschriebener Baustein besteht aus einer Folge von Schritten, die über gerichtete Verbindungen (Transitionen) miteinander verbunden sind. Den Schritten werden Aktionen zugeordnet, in denen der zu bearbeitende Programmcode abgelegt ist.

"IndraLogic" unterstützt folgende Arten von Aktionen:

Erste Schritte - IndraLogic

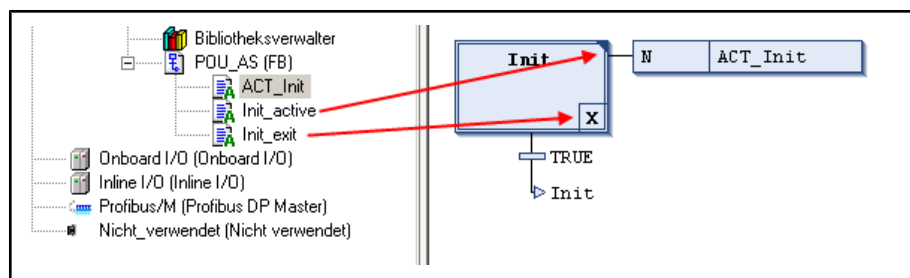


Abb.4-38: Arten der Aktionen eines Schrittes

- **Eingangsaktion:** Eine Eingangsaktion wird beim Aktivwerden des Schrittes genau einmal bearbeitet. Hier: „Init_active“. Ein Schritt mit Eingangsaktion wird durch ein schwarzes Dreieck in der rechten oberen Ecke gekennzeichnet.
- **Assoziierte Aktion:** Die Aktion wird separat als Bestandteil der POU angelegt und mit den gewünschten Schritten assoziativ verbunden. Hier: „ACT_Init“. Ihr Zeitverhalten wird durch das Bestimmungszeichen (hier: N) bestimmt. Die assoziierten Aktionen erscheinen rechts vom Schritt. Wird eine solche Aktion inaktiv, wird sie genau einmal nachbearbeitet.
- **Ausgangsaktion:** Eine Ausgangsaktion wird nur einmal ausgeführt, bevor der Schritt deaktiviert wird. Hier: „Init_exit“. Ein Schritt mit Ausgangsaktion wird durch ein "X" in der rechten unteren Ecke gekennzeichnet.

Aktion "Init" mit "ST" programmieren

In der Aktion zum Schritt "Init" werden die Variablen "TRAFFICSIGNAL1" und "TRAFFICSIGNAL2" initialisiert. Der Status von "TRAFFICSIGNAL1" der Ampel 1 soll "1" (grün) sein. Der Status von "TRAFFICSIGNAL2" der Ampel 2 soll "3" (rot) sein.

Zum Editieren von "Init" gehen Sie wie folgt vor:

1. Doppelklick auf das Kästchen des ersten Schrittes "Init" im Programm "SEQUENCE".
Ein Fenster "Aktion hinzufügen" öffnet sich.
2. Wählen Sie hier als Implementationssprache (Programmiersprache) "ST" aus.
3. Bestätigen Sie die Einstellungen mit der Schaltfläche "Öffnen".
Im Projektextplorer unter der POU "ABLAUF (FB)" wird eine Verzweigung mit der Aktion "Init_active" erzeugt.

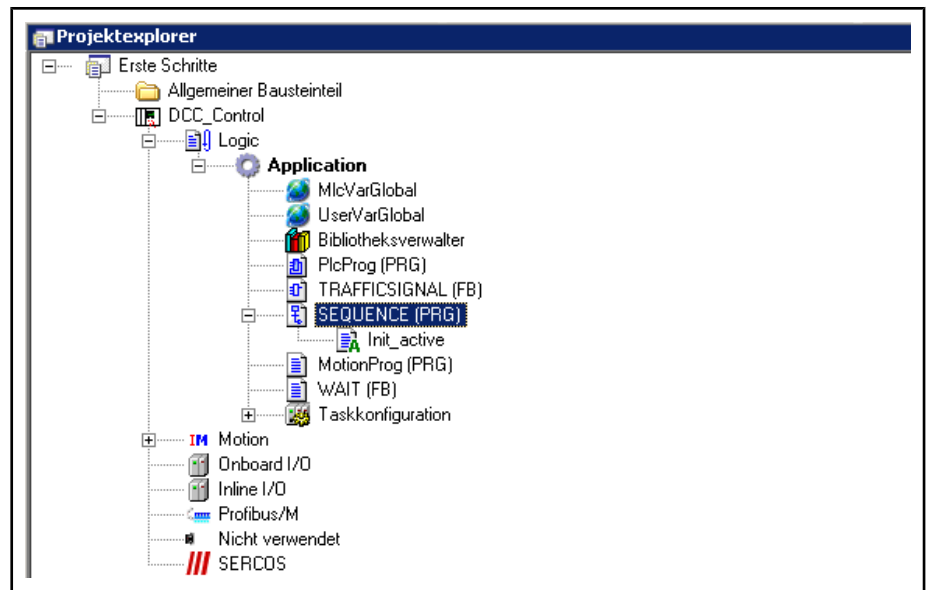


Abb.4-39: POU "SEQUENCE" im Projektexplorer mit Aktion "Init_active"
Der Editor der Aktion "Init_active" wird geöffnet.

4. Geben Sie im Editor folgende Programmzeilen ein:

Aktion "Init_active"

```
TRAFFICSIGNAL1 := 1;  
TRAFFICSIGNAL2 := 3;
```



Beim Programmieren der Aktion "Init_active" sind die Variablen "TRAFFICSIGNAL1" und "TRAFFICSIGNAL2" noch nicht bekannt. Das Fenster "Variable deklarieren" öffnet sich automatisch. Wählen Sie bei beiden Variablen die Sichtbarkeit "VAR_OUTPUT" und den Datentyp "INT" aus.

Aktion "Switch1" mit "ST" programmieren

Der Editor der Aktion "Init" sieht wie folgt aus:

In der Aktion zum Schritt "Switch1" wechselt der STATUS von Ampel1 (TRAFFICSIGNAL1) auf "2" (gelb) und der STATUS von Ampel 2 (TRAFFICSIGNAL2) auf "4" (gelb-rot). Es wird eine Verzögerungszeit von zwei Sekunden festgelegt.

Um "Switch1" zu editieren, gehen Sie wie folgt vor:

1. Doppelklick auf das Kästchen des zweiten Schrittes "Switch1" in der POU "SEQUENCE".

Das Fenster "Aktion hinzufügen" öffnet sich.

2. Wählen Sie hier als Implementationssprache (Programmiersprache) "Strukturierter Text (ST)" aus.

3. Bestätigen Sie Ihre Eingaben mit der Schaltfläche "Öffnen".

Im Projektexplorer unter der POU "SEQUENCE (FB)" wird eine Verzweigung mit der Aktion "Switch1_active" erzeugt.

Der Editor der Aktion "Switch1_active" wird geöffnet.

4. Geben Sie im Editor folgende Programmzeilen ein:

Erste Schritte - IndraLogic

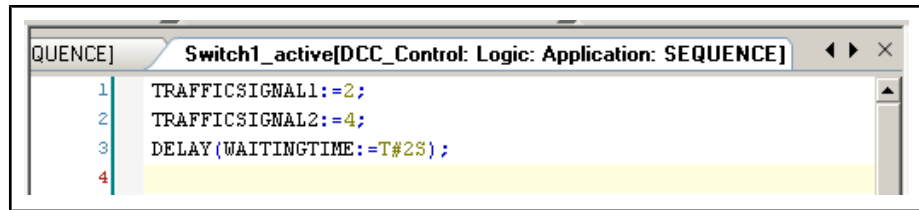


Abb.4-40: Editor der Aktion "Switch1_active"

Aktion "Green2" mit "ST" programmieren

In der Aktion zum Schritt "Green2" wechselt der STATUS von Ampel 1 (TRAFFICSIGNAL1) auf "3" (rot) und der STATUS von Ampel 2 (TRAFFICSIGNAL2) auf "1" (grün). Es wird eine Verzögerungszeit von fünf Sekunden festgelegt.

Um "Green2" zu editieren, gehen Sie wie folgt vor:

1. Doppelklick auf das Kästchen des dritten Schrittes "Green2" in der POU "SEQUENCE".

Das Fenster "Aktion hinzufügen" öffnet sich.

2. Wählen Sie hier als Implementationssprache (Programmiersprache) "Strukturierter Text (ST)" aus.
3. Bestätigen Sie die Eingabe mit der Schaltfläche "Öffnen".

Im Projektextplorer unter der POU "SEQUENCE (FB)" wird eine Verzweigung mit der Aktion "Green2_active" erzeugt. Der Editor der Aktion "Green2_active" wird geöffnet.

4. Geben Sie im Editor folgende Programmzeilen ein:

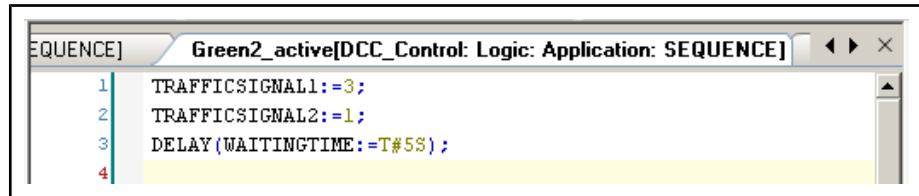


Abb.4-41: Editor der Aktion "Green2_active"

Aktion "Switch2" mit "ST" programmieren

In der Aktion zum Schritt "Switch2" wechselt der STATUS von Ampel 1 (TRAFFICSIGNAL1) auf "4" (gelb-rot) und der STATUS von Ampel 2 (TRAFFICSIGNAL2) auf "2" (gelb). Es wird eine Verzögerungszeit von zwei Sekunden festgelegt.

Um "Switch2" zu editieren, gehen Sie wie folgt vor:

1. Doppelklick auf das Kästchen des vierten Schrittes "Switch2" in der POU "SEQUENCE".

Ein Fenster "Aktion hinzufügen" öffnet sich.

2. Wählen Sie hier als Implementationssprache (Programmiersprache) "Strukturierter Text (ST)" aus.
3. Bestätigen Sie die Eingabe mit der Schaltfläche "Öffnen".

Im Projektextplorer unter der POU "SEQUENCE (FB)" wird eine Verzweigung mit der Aktion "Switch2_active" erzeugt. Der Editor der Aktion "Switch2_active" wird geöffnet.

4. Geben Sie im Editor folgende Programmzeilen ein:

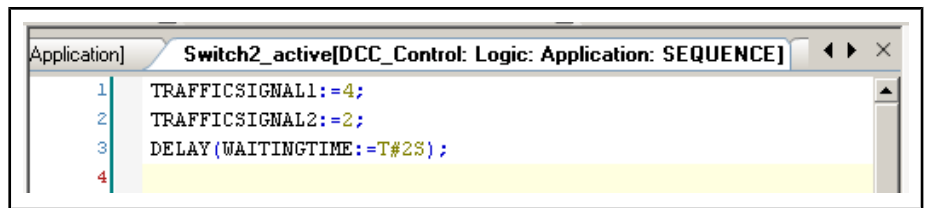


Abb.4-42: Editor der Aktion "Switch2_active"

Aktion "Green1" mit "ST" programmieren

In der Aktion zum Schritt "Green1" wechselt der STATUS von AMPEL 1 auf "1" (grün) und der STATUS von AMPEL 2 auf "3" (rot). Es wird eine Verzögerungszeit von fünf Sekunden festgelegt.

Um "Green1" zu editieren gehen Sie wie folgt vor:

1. Doppelklick auf das Kästchen des fünften Schrittes "Green1".
Ein Fenster "Aktion hinzufügen" öffnet sich.
2. Wählen Sie hier als Implementationssprache (Programmiersprache) "Strukturierter Text (ST)" aus.
3. Bestätigen Sie die Eingabe mit der Schaltfläche "Öffnen".
Im Projektextplorer unter der POU "SEQUENCE (FB)" wird eine Verzweigung mit der Aktion "Green1_active" erzeugt.
Der Editor der Aktion "Green1_active" wird geöffnet.
4. Geben Sie im Editor folgende Programmzeilen ein:

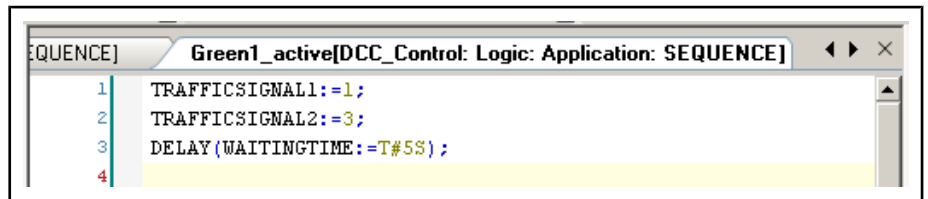


Abb.4-43: Editor der Aktion "Green1_active"

Der fertige Funktionsbaustein "SEQUENCE"

Der Funktionsbaustein "SEQUENCE" und die Struktur im Projektextplorer ist nun fertiggestellt, siehe nachfolgende Abbildung.

Erste Schritte - IndraLogic

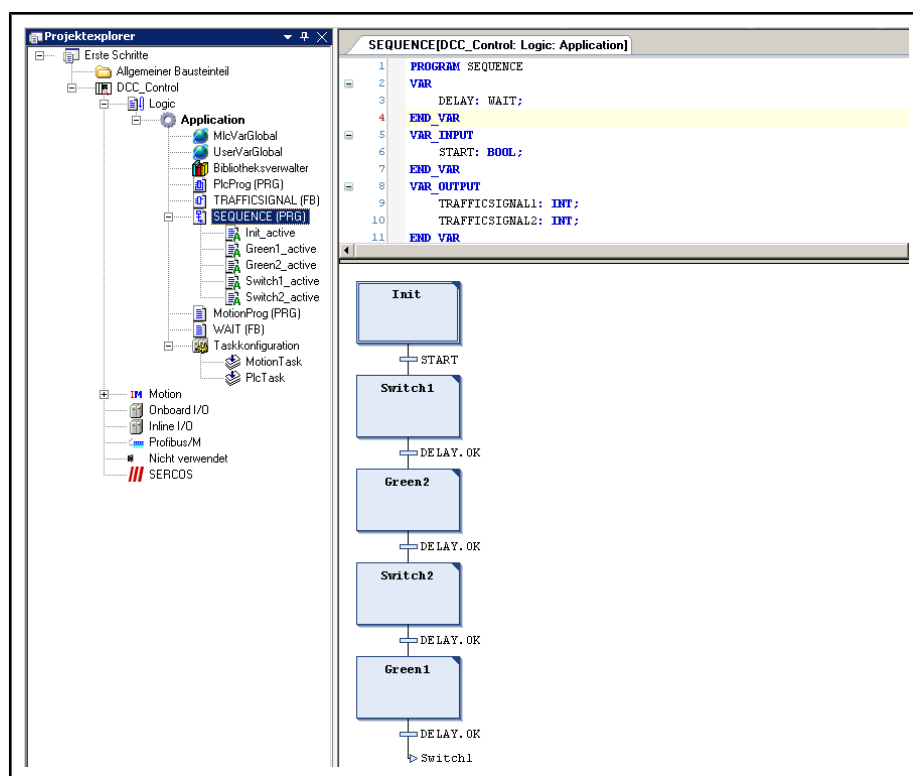


Abb.4-44: POU "SEQUENCE" fertiggestellt

4.6.4 Taskkonfiguration

Die Task Eine "Task" ist eine Aufgabe die ausgeführt werden soll. Die Aufgabe besteht darin, Programme durch Vereinbarungen von "Task-Typen", wie z.B. "Zyklisch", "Freilaufend", und durch Zuweisungen von Laufzeiten aufzurufen. Mit einer "Task" wird der Programmablauf organisiert.

Der Taskkonfigurator Der Taskkonfigurator definiert eine oder mehrere Tasks, die die Ausführung einer Applikation kontrollieren.

Hinzufügen der Task "SEQUENCE" In der Applikation befinden sich zwei POUs vom Typ "Programm": "PlcProg" und "SEQUENCE". "PlcProg" wird automatisch beim Einfügen einer Steuerung angelegt. Eine Task "PlcProg" wird ebenso beim Einfügen einer Steuerung angelegt, damit die POU "PlcProg" ablauffähig ist. Nun wird noch eine Task für "SEQUENCE" hinzugefügt, um die POU "SEQUENCE" aufrufen zu können.

Um die Task "SEQUENCE" hinzuzufügen, gehen Sie wie folgt vor:

1. Öffnen Sie im Projektexplorer die Konfiguration "PlcTask" mit einem Doppelklick unterhalb der "Taskkonfiguration".
2. Wählen Sie im Abschnitt "POUs" der geöffneten Konfiguration "Aufruf hinzufügen".

Das Fenster "Eingabehilfe" öffnet sich automatisch.

3. Öffnen Sie in der "Eingabehilfe" unter "Elemente:" den Baum bei "DCC_Control" bis "SEQUENCE" und "PlcProg" sichtbar sind.
4. Wählen Sie "SEQUENCE" und bestätigen Sie mit der Schaltfläche "OK". Die "PlcTask-Konfiguration" sieht nun folgendermaßen aus:

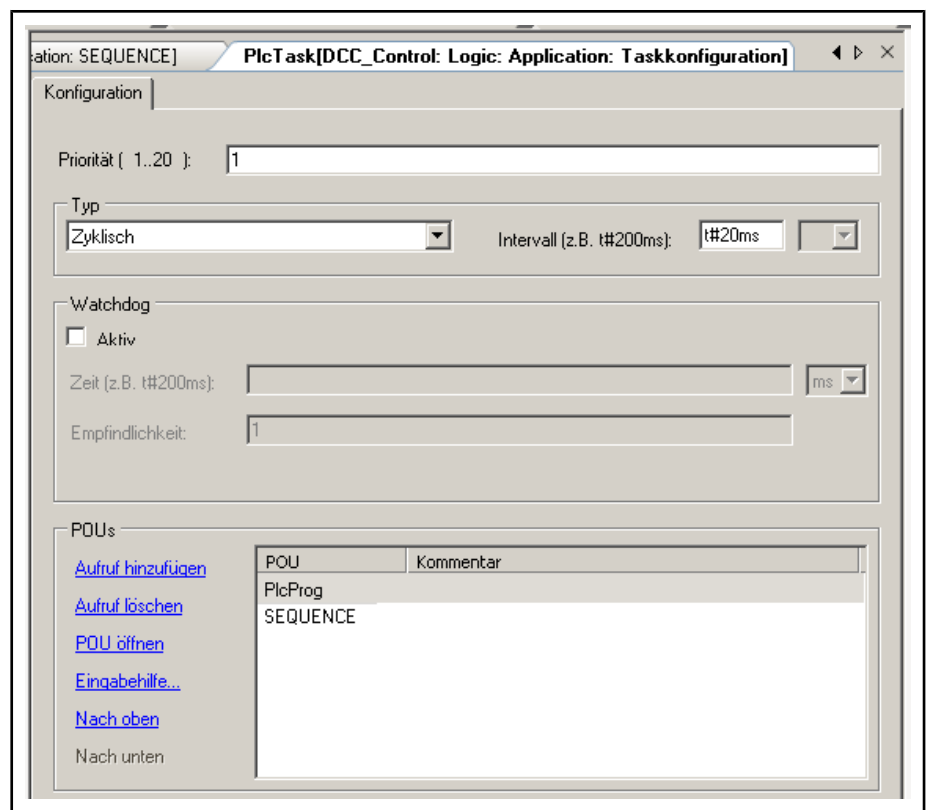


Abb.4-45: Taskkonfiguration

4.6.5 Globale Variablen deklarieren

Funktion von GVL

Die Ampelanlage soll über den Schalter "ON" eingeschaltet werden.

Bei den zwei Ampeln sollen, je nach Programmzustand für die entsprechende Ampelphase, die jeweiligen Ampelleuchten angesteuert werden.

Für den einen Eingang und die sechs Ausgänge der beiden Ampeln müssen Eingangs- und Ausgangsvariablen angelegt werden.

Der Zugriff auf die physikalischen Adressen der Ein- und Ausgänge erfolgt mit der Syntax "%IX" für einen Eingang und "%QX" für einen Ausgang.

Der Größenpräfix "X" beschreibt, auf welchen Bereich der Adressen zugegriffen werden soll. In diesem Beispiel werden nur auf Bitadressen zugegriffen.

Die Variablen sollen nicht nur lokal in dem Baustein "PlcProg" verfügbar sein, sondern auch in anderen Bausteinen. Deshalb werden die Variablen nicht in der POU "PlcProg", sondern in einer Globalen Variablen Liste "GVL" deklariert.

"Globale Variablen Liste (GVL)" anlegen

Hierzu müssen Sie zunächst eine "GVL" in das Projekt einfügen (Kontextmenü von "Application", **Hinzufügen...** ► **Globale Variablenliste...**).



Wird die Globale Variablen Liste (GVL) unterhalb von "Application" angelegt, stehen die Variablen allen POU's der Applikation zur Verfügung.

Variablen mit Adressen deklarieren

Die "Global Variable List" erhält neue Variablen vom Typ "BOOL". Legen Sie folgende Variablen an:

Erste Schritte - IndraLogic

Globale Variablenliste "GVL"

```
VAR_GLOBAL
  ON : BOOL;
  A1_green AT %QX0.0 : BOOL;
  A1_yellow AT %QX0.1 : BOOL;
  A1_red AT %QX0.2 : BOOL;
  A2_green AT %QX0.3 : BOOL;
  A2_yellow AT %QX0.4 : BOOL;
  A2_red AT %QX0.5 : BOOL;
END_VAR
```

Dem Namen der Variable folgt nach der Zuweisung "AT" mit einem Prozentzeichen beginnend der Bereichspräfix, mit "Q" für Ausgänge und "I" für Eingänge. Danach folgt der Größenpräfix, mit "X" für Bitadressierung.

Mit A1_green AT %QX0.0: BOOL; wird die Variable "A1_green" vom Datentyp "BOOL" dem physikalischen Ausgangsbit "Q 0.0" zugewiesen.



Bei Verwendung einer realen Steuerung kontrollieren Sie bitte, dass es diese Ausgänge wirklich gibt.

Die Steuerungen XLC / MLC L45 und L65 besitzen sie als Onboard-Ausgänge (Standard-Mapping).

Die Steuerungen XLC / MLC L25 müssen durch Inline-Module oder einen Feldbus erweitert sein. Wenn nicht, verzichten Sie auf die absolute Adresse oder Sie erhalten bei der Code-Erzeugung eine Warnung.

4.6.6 Testen von "SEQUENCE" im Simulationsmodus

Der Programmbaustein "SEQUENCE" ist soweit fertiggestellt, dass er getestet werden kann. Der Test soll im Simulationsmodus durchgeführt werden.

"Application" übersetzen

Um die Applikation in die Steuerung laden zu können, muss die Applikation zunächst fehlerfrei vom "Compiler" übersetzt werden. Um die Applikation zu übersetzen, wählen Sie im Menü **Erstellen ► Application übersetzen**. In der "Aufgabenliste" unterhalb des Arbeitsfensters steht nach dem Übersetzen: "Übersetzung abgeschlossen -- 0 Fehler, 0 Warnungen: Bereit zum Download!"

Steuerung simulieren

Ist zum Testen keine Steuerung vorhanden, können Sie eine Steuerung simulieren. Um eine Steuerung zu simulieren, klicken Sie im "Projektexplorer" auf die Steuerung "DCC_Control" und öffnen Sie mit einem Klick auf die rechte Maustaste das Kontextmenü. Klicken Sie auf **Simulation** um den Simulationsmodus zu aktivieren. Ist die Simulation aktiv, wird ein Haken bei "Simulation" gesetzt.

Erste Schritte - IndraLogic

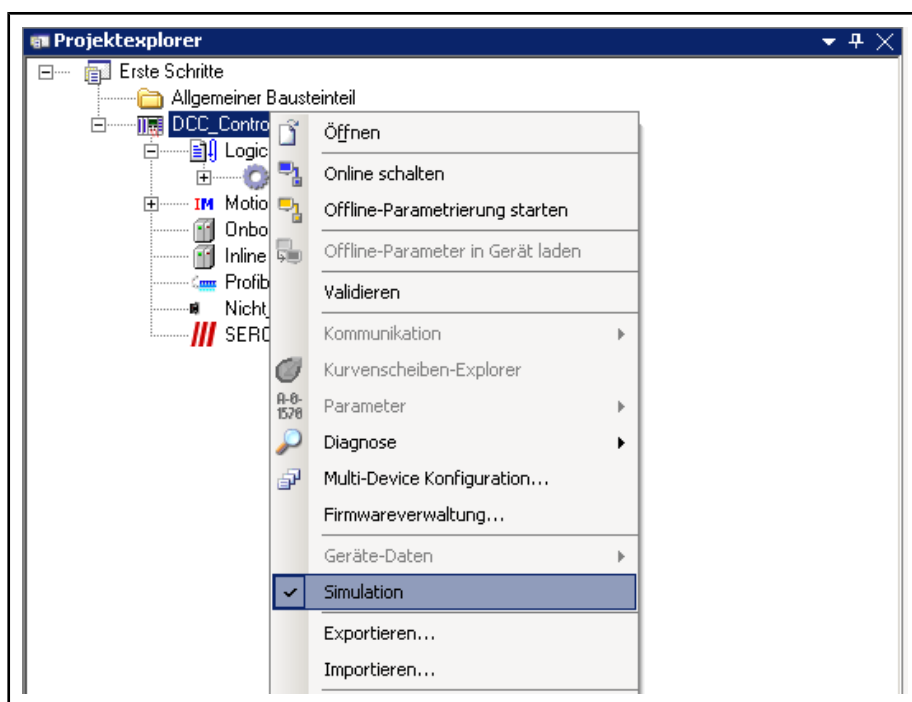


Abb.4-46: Auswahl Simulation

Einloggen in die Steuerung

Nun soll sich auf die Steuerung eingeloggt und die Applikation in die Steuerung geladen werden. Um sich auf die Steuerung einzuloggen, wählen Sie im Menü **Debuggen ► Einloggen**

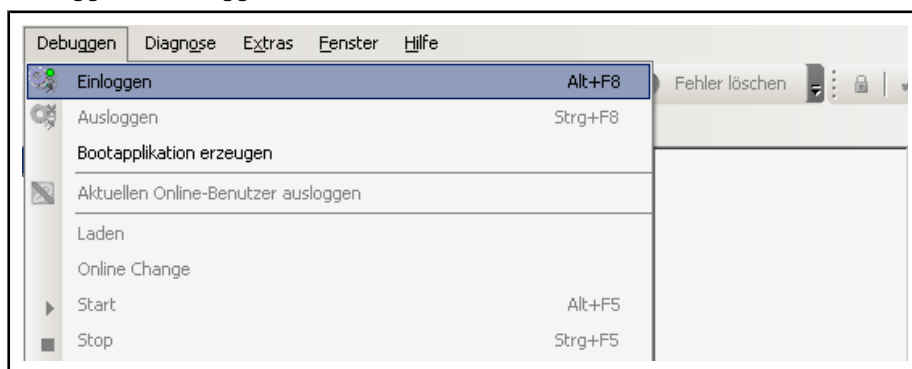


Abb.4-47: Einloggen auf der Steuerung

Der Projektexplorer hat nun folgendes Erscheinungsbild:

Erste Schritte - IndraLogic

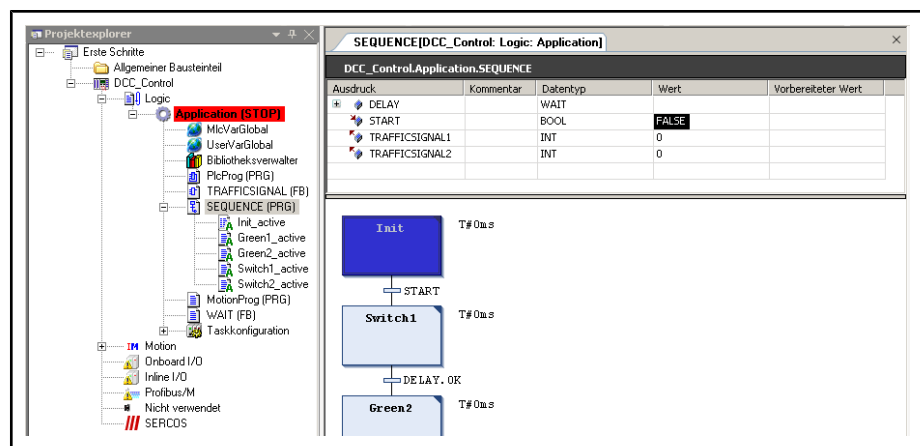


Abb.4-48: In Steuerung (Simulation) eingeloggt – Applikation STOP

"Application" starten

Um die Applikation zu starten, klicken Sie mit der rechten Maustaste auf die Applikation und wählen im Kontextmenü **Start** oder im Hauptmenü **Debuggen ▶ Start**.

Die Applikation läuft nun auf der Steuerung ab.

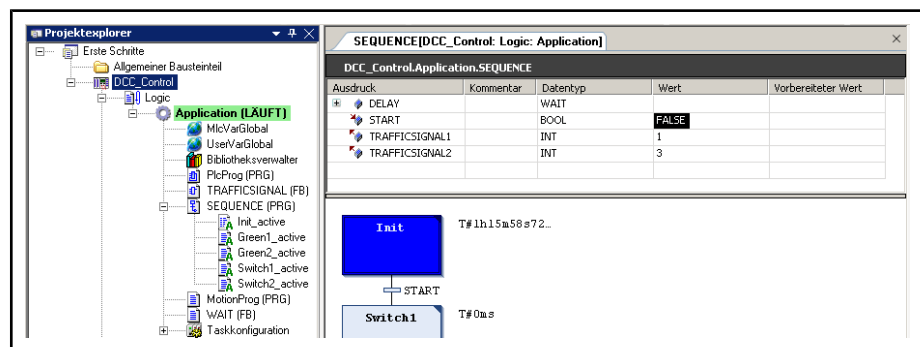


Abb.4-49: In Steuerung eingeloggt – Applikation (LÄUFT)

"SEQUENCE" starten

Öffnen Sie nun die POU "SEQUENCE". Im Online-Modus des Deklarationsteils sind die Variablen und ihre aktuellen Werte zu sehen. Klicken Sie im Deklarationsteil in der Zeile bei der Variablen "Start" in die Spalte "Vorbereiteter Wert" bis dort "TRUE" erscheint.

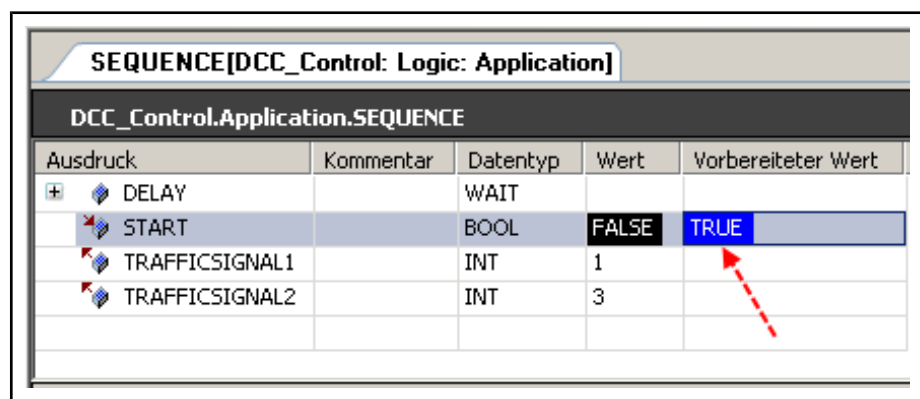


Abb.4-50: Deklarationsteil POU "SEQUENCE" online

Wählen Sie im Menü **Debuggen ▶ Werte schreiben** (<Strg>+<F7>).

Die Variable "START" wird auf "TRUE" gesetzt.

Erste Schritte - IndraLogic

SEQUENCE[DCC_Control: Logic: Application]				
DCC_Control.Application.SEQUENCE				
Ausdruck	Kommentar	Datentyp	Wert	Vorbereiteter Wert
DELAY		WAIT		
WAITING...		TIME	T#5s	
OK		BOOL	FALSE	
ZAB		TP		
ETVar		TIME	T#0ms	
START		BOOL	TRUE	
TRAFFICSIG...		INT	3	
TRAFFICSIG...		INT	1	

Abb.4-51: Deklarationsteil POU "SEQUENCE" online

Die einzelnen Schritte und Transitionen der POU "SEQUENCE" werden abgearbeitet. Am Ende erfolgt der Rücksprung zu Schritt "Switch1".

Die Werte für die Variable "START" werden später in der POU "PlcProg" gesetzt.

Aus Steuerung ausloggen

Bevor Sie sich aus der Steuerung ausloggen, stoppen Sie die Applikation. Um die Applikation zu stoppen, klicken Sie mit der rechten Maustaste auf die Applikation und wählen Sie im Kontextmenü **Stop** oder im Hauptmenü **Debuggen** ► **Stop**.

Die Applikation wird auf der Steuerung gestoppt.

The screenshot shows the IndraWorks interface. On the left, the 'Projektexplorer' (Project Explorer) displays a tree structure with 'Erste Schritte' (First Steps) expanded, showing 'DCC_Control' and 'Logic'. Under 'Logic', 'Application (STOP)' is highlighted. The main window shows the 'SEQUENCE[DCC_Control: Logic: Application]' POU declaration. The table below is identical to the one in Abb.4-51, but the 'START' variable is now set to 'TRUE'. Below the table, a ladder logic diagram is visible, showing an 'Init' block connected to a 'START' block, which is then connected to a 'Switch1' block.

Abb.4-52: In Steuerung eingeloggt - Applikation (STOP)

Loggen Sie sich jetzt aus, indem Sie im Menü **Debuggen** ► **Ausloggen** auswählen.

IndraWorks befindet sich im Offline-Modus.



Bei einem erneuten Einloggen werden die zum Zeitpunkt des letzten Ausloggen vorhandenen Werte verwendet.

Sie können mit einer Neubelegung starten, wenn Sie vor dem Einloggen die alten Daten löschen (Menüpunkt **Erstellen** ► **Bereinigen**).

4.6.7 Programme und Funktionsbausteine editieren – zweiter Teil

Programm "PlcProg" in "ST" editieren



Die Implementation des Programms "PlcProg" wurde alternativ in zwei Programmiersprachen angeboten:

- Programmierung in "Continuous Function Chart (CFC)", gehen Sie zu [Programm PlcProg in CFC editieren, Seite 65](#),
- Programmierung in "Strukturiertem Text (ST)", setzen Sie im Text fort.

Funktion von "PlcProg"

Im Baustein "SEQUENCE" wurde der zeitliche Ablauf der Ampelphasen von den beiden Ampeln umgesetzt. Nun sollen Ein- und Ausgänge einer SPS angesteuert und ausgewertet werden. Für den einen Eingang und die sechs Ausgänge für die beiden Ampeln wurden in der "Global Variabel List (GVL)" Eingangs- und Ausgangsvariablen angelegt. Diese Variablen können nun in das Programm "PlcProg" verwendet werden. Die Applikation wurde bisher nur im Simulationsmodus getestet. Nun kann die Applikation auch in einer Steuerung getestet werden.

Instanzen deklarieren

Das Programm "PlcProg" erhält eine Variable "LIGHT1" vom Typ "TRAFFIC-SIGNAL" und eine Variable "LIGHT2" ebenfalls vom Typ "TRAFFICSIGNAL". Der Funktionsbaustein "TRAFFICSIGNAL" wird somit instanziiert.

Zum Anlegen der Variable orientieren Sie sich an [Variablen deklarieren, Seite 39](#).

Legen Sie die Variable LIGHT1: TRAFFICSIGNAL;

und LIGHT2: TRAFFICSIGNAL an:

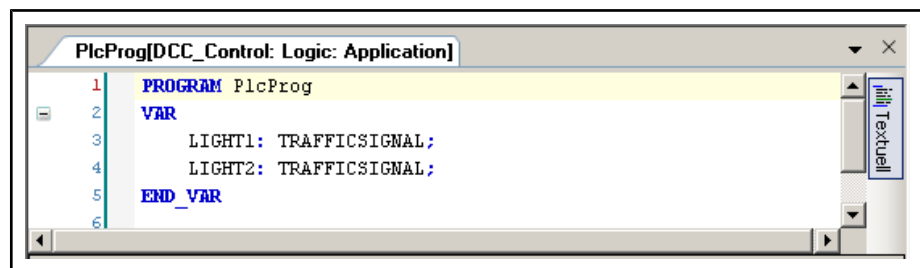


Abb.4-53: Deklarationsteil POU "PlcProg"

Rumpf editieren

Editiert wird die Programm "PlcProg" in der Programmiersprache "ST". Als erstes soll der Baustein "SEQUENCE" aufgerufen werden. Am Eingang "START" des Programms "SEQUENCE" wird die globale Variable "ON" angelegt.

Im Folgenden werden die beiden Funktionsbausteininstanzen "LIGHT1" und "LIGHT2" vom Typ "TRAFFICSIGNAL" aufgerufen.

Am Eingang "STATUS" von "LIGHT1" wird der Ausgang "TRAFFICSIGNAL1" des Programms "SEQUENCE" angelegt. Die drei Ausgänge "GREEN", "YELLOW" und "RED" der Instanz "LIGHT1" werden auf die globalen Variablen "A1_green", "A1_yellow" und "A1_red" geschrieben.

Mit der Funktionsbausteininstanz "LIGHT2" wird analog zu "LIGHT1" verfahren.



"A1_green" heißt korrekt "GVL.A1_green" oder kurz ".A1_green" und so weiter, da es sich um globale Variablen handelt.

Das System nimmt jedoch, wenn kein "A1_green" lokal deklariert ist, ersatzweise die globale Variable gleichen Namens.

Das Programm "PlcProg" sieht nun folgendermaßen aus:

```

1  PROGRAM PlcProg
2  VAR
3      LIGHT1: TRAFFICSIGNAL;
4      LIGHT2: TRAFFICSIGNAL;
5  END_VAR

1  SEQUENCE (START:=ON);
2
3  LIGHT1 (STATUS:=SEQUENCE.TRAFFICSIGNAL1,
4      GREEN=>A1_green,
5      YELLOW=>A1_yellow,
6      RED=>A1_red);
7
8  LIGHT2 (STATUS:=SEQUENCE.TRAFFICSIGNAL2,
9      GREEN=>A2_green,
10     YELLOW=>A2_yellow,
11     RED=>A2_red);
12

```

Abb.4-54: Programm "PlcProg" fertiggestellt

Testen der Applikation im Simulationsmodus, PlcProg in ST

"Application" übersetzen

Um die Applikation in die Steuerung laden zu können, muss die Applikation zunächst fehlerfrei vom "Compiler" übersetzt werden. Um die Applikation zu übersetzen, wählen Sie im Menü **Erstellen ► Übersetzen**. In der "Aufgabenliste" unterhalb des Arbeitsfensters steht nach dem Übersetzen: "Übersetzung abgeschlossen -- 0 Fehler, 0 Warnungen: Bereit zum Download!"



Es wird immer nur die Applikation übersetzt, die gerade aktiv ist.

Sind **mehrere** Applikationen **verfügbar**, muss die Applikation, die übersetzt werden soll, aktiv gesetzt werden (Kontext-Menü "Application" **Aktive Application setzen**).

Steuerung simulieren

Da zum Testen eventuell keine Steuerung vorhanden ist, kann man eine Steuerung simulieren. Um eine Steuerung zu simulieren, klicken Sie im "Projektfexplorer" auf die Steuerung "DCC_Control" und öffnen Sie mit einem Klick auf die rechte Maustaste das Kontextmenü. Klicken Sie auf **Simulation**, um den Simulationsmodus zu aktivieren. Ist die Simulation aktiv, wird ein Haken bei "Simulation" gesetzt.

In Steuerung einloggen

Nun soll sich auf die Steuerung eingeloggt und die Applikation in die Steuerung geladen werden. Um sich auf die Steuerung einzuloggen, wählen Sie im Menü **Debuggen ► Einloggen**.

Die Applikation steht danach im Zustand "STOP".

"Application" starten

Um die Applikation zu starten, klicken Sie mit der rechten Maustaste auf die Applikation und wählen im Kontextmenü **Start**. Die "Application" läuft nun als Simulation ab.

Die Ampel einschalten und Ampelzyklus beobachten

Nach dem Starten der Applikation ist die "LIGHT1" im Zustand "Grün" und die "LIGHT2" im Zustand "Rot". Die Ampel muss eingeschalten werden, damit der

Erste Schritte - IndraLogic

Ampelzyklus durchlaufen wird. Um die Ampel einzuschalten, gehen Sie wie folgt vor:

1. Öffnen Sie durch einen Doppelklick auf die "GVL" im Projektexplorer die "Globale Variablen Liste".

Ausdruck	Datentyp	Wert	Vorbereiteter Wert
ON	BOOL	FALSE	
A1_green	BOOL	TRUE	
A1_yellow	BOOL	FALSE	
A1_red	BOOL	FALSE	
A2_green	BOOL	FALSE	
A2_yellow	BOOL	FALSE	
A2_red	BOOL	TRUE	

Abb.4-55: "GVL" online

2. Klicken Sie bei der Variablen "ON" in die Spalte "Vorbereiteter Wert", so dass in dieser Spalte "TRUE" erscheint.
3. Um den Wert auf die Steuerung zu schreiben, klicken Sie im Menü **Debuggen** ► **Werte schreiben**.

Der Ampelzyklus wird nun durchlaufen. Im Baustein "PlcProg" können Sie den Ampelzyklus beobachten.

Ausdruck	Kommentar	Datentyp	Wert	Vorbereiteter Wert
LIGHT1		TRAFFICSIGNAL		
STATUS		INT	4	
GREEN		BOOL	FALSE	
YELLOW		BOOL	TRUE	
RED		BOOL	TRUE	
LIGHT2		TRAFFICSIGNAL		
STATUS		INT	2	
GREEN		BOOL	FALSE	
YELLOW		BOOL	TRUE	
RED		BOOL	FALSE	

```

1  SEQUENCE (START:= GVL.ON(TRUE) );
2
3  LIGHT1 (STATUS 4 := SEQUENCE.TRAFFICSIGNAL1 4,
4    GREEN FALSE=> GVL.A1_green FALSE,
5    YELLOW TRUE=> GVL.A1_yellow TRUE,
6    RED TRUE=> GVL.A1_red TRUE);
7
8  LIGHT2 (STATUS 2 := SEQUENCE.TRAFFICSIGNAL2 2,
9    GREEN FALSE=> GVL.A2_green FALSE,
10   YELLOW TRUE=> GVL.A2_yellow TRUE,
11   RED FALSE=> GVL.A2_red FALSE);
12  RETURN
  
```

Abb.4-56: Programm "PLCProg" online

Applikation stoppen
Aus Steuerung ausloggen

Stoppen Sie das SPS-Programm über die Menüleiste mit **Debuggen** ► **Stop**.
Um sich aus der Steuerung auszuloggen, wählen Sie aus der Menüleiste **Debuggen** ► **Ausloggen** aus.

Programm "PlcProg" in "CFC" editieren



Die Implementation des Programms "PlcProg" wurde alternativ in zwei Programmiersprachen angeboten:

- Programmierung in "Strukturiertem Text (ST)", gehen Sie zu [Programm PlcProg in ST editieren, Seite 62](#),
- Programmierung in "Continuous Function Chart (CFC)", setzen Sie im Text fort.

Funktion von "PlcProg"

Im Funktionsbaustein "SEQUENCE" wurde der zeitliche Ablauf der Ampelphasen von den beiden Ampeln umgesetzt. Nun sollen Ein- und Ausgänge einer SPS angesteuert und ausgewertet werden. Für den einen Eingang und die sechs Ausgänge für die beiden Ampeln wurden in der "Global Variabel List (GVL)" Eingangs- und Ausgangsvariablen angelegt. Diese Variablen können nun in der POU "PlcProg" verwendet werden. Die Applikation wurde bisher nur im Simulationsmodus getestet. Nun kann die Applikation auch in einer Steuerung getestet werden.

Instanzen deklarieren

Das Programm "PlcProg" erhält eine Instanz "LIGHT1" vom Typ "TRAFFICSIGNAL" und eine Instanz "LIGHT2" ebenfalls vom Typ "TRAFFICSIGNAL".

Der Funktionsbaustein "TRAFFICSIGNAL" wird somit instanziiert.

Legen Sie die Variable LIGHT1: TRAFFICSIGNAL; und LIGHT2: TRAFFICSIGNAL an:

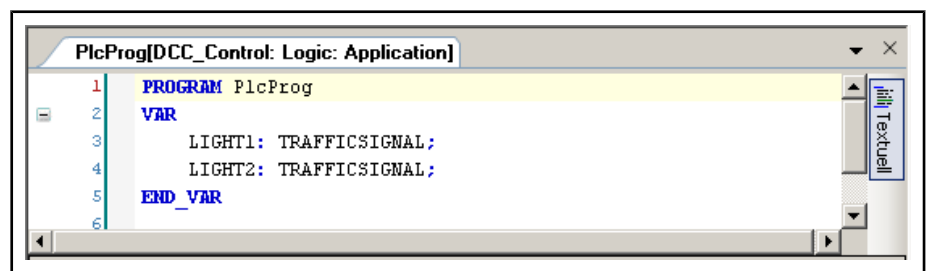


Abb.4-57: Deklarationsteil POU "PlcProg"

Rumpf editieren

Editiert wird die POU "PlcProg" in der Programmiersprache "CFC". Als erstes soll der Baustein "SEQUENCE" aufgerufen werden:

1. Wählen Sie im Fenster "Werkzeuge" das Element "Baustein" mit einem Klick aus:

Erste Schritte - IndraLogic

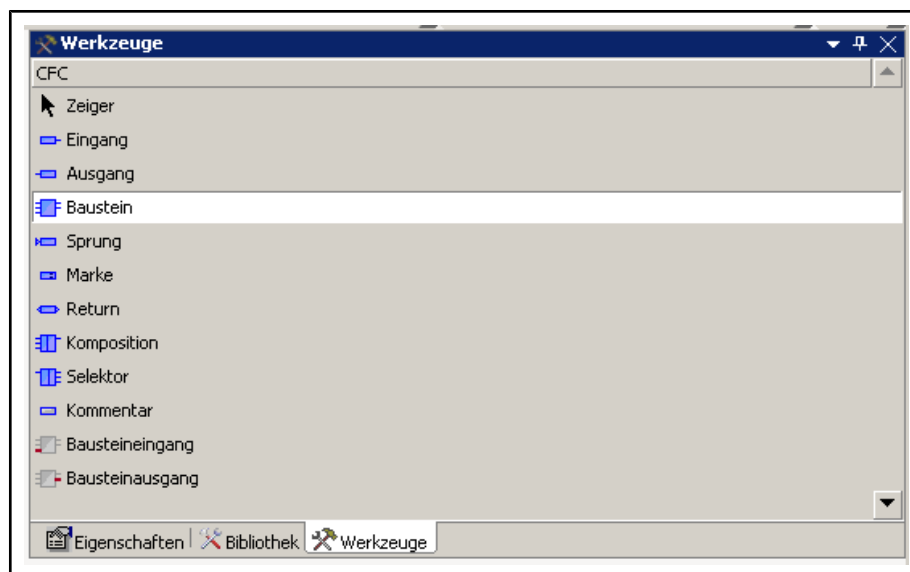


Abb.4-58: Werkzeuge – Element Baustein

2. Begeben Sie sich mit der Maus in den Rumpf des Editors "PlcProg" und klicken auf die linke Maustaste, damit der Baustein dort eingefügt wird.
3. Klicken Sie den Namen "???" des Bausteins an und geben Sie dem Baustein den Namen "SEQUENCE".

Der Baustein sieht nun folgendermaßen aus:

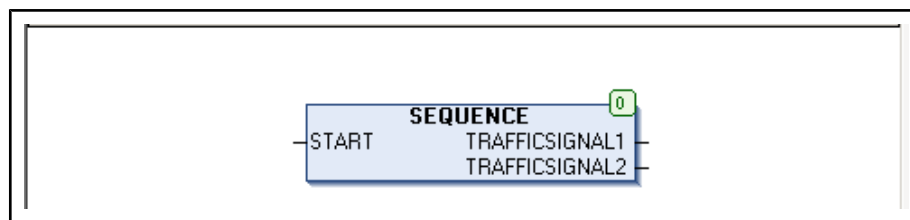


Abb.4-59: Baustein SEQUENCE

Es wird am Ausgang "TRAFFICSIGNAL1" des Bausteins "SEQUENCE" ein Baustein "LIGHT1" mit der Funktion von "TRAFFICSIGNAL" angefügt:

1. Wählen Sie im Fenster "Werkzeuge" das Element "Baustein" mit einem Klick aus.
2. Begeben Sie sich mit der Maus in den Rumpf des Editors "PlcProg" und klicken auf die linke Maustaste, damit der Baustein dort eingefügt wird.
3. Klicken Sie den Namen "???" des Bausteins an und geben Sie dem Baustein den Typnamen "TRAFFICSIGNAL".
4. Geben Sie dem Baustein "TRAFFICSIGNAL" bei "???" den Instanznamen "LIGHT1".

Der Baustein sieht nun folgendermaßen aus:

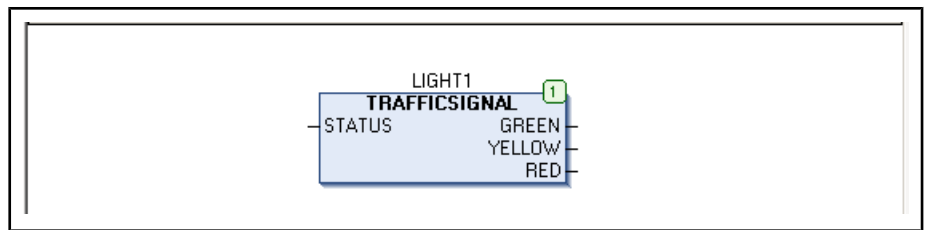


Abb.4-60: Baustein TRAFFICSIGNAL – LIGHT1

Als nächstes wird am Ausgang "TRAFFICSIGNAL2" des Bausteins "SEQUENCE" ein Baustein "LIGHT2" mit der Funktion von "TRAFFICSIGNAL" angefügt:

1. Wählen Sie im Fenster "Werkzeuge" das Element "Baustein" mit einem Klick aus.
2. Begeben Sie sich mit der Maus in den Rumpf des Editors "PlcProg" und klicken auf die linke Maustaste, damit der Baustein dort eingefügt wird.
3. Klicken Sie den Namen "???" des Bausteins an und geben Sie dem Baustein den Namen "TRAFFICSIGNAL".
4. Geben Sie dem Baustein "TRAFFICSIGNAL" bei "???" den Instanznamen "LIGHT2".

Der Baustein sieht nun folgendermaßen aus:

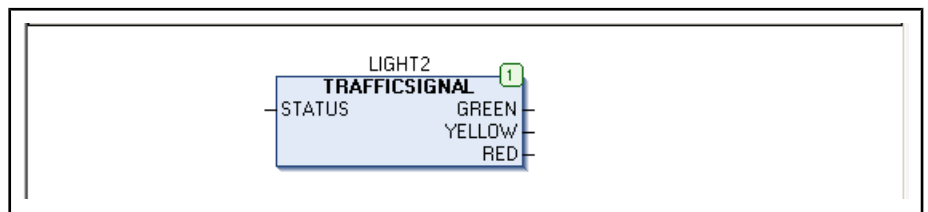


Abb.4-61: Baustein TRAFFICSIGNAL – LIGHT2

Als nächstes werden die Elemente "Eingang" und "Ausgang" eingefügt:

1. Wählen Sie im Fenster "Werkzeuge" das Element "Eingang" mit einem Klick aus.
2. Begeben Sie sich mit der Maus in den Rumpf des Editors "PlcProg" und klicken auf die linke Maustaste, damit der Eingang dort eingefügt wird.
3. Klicken Sie den Namen "???" des Eingangs an und geben Sie dem Eingang den Namen "ON".
4. Wählen Sie im Fenster "Werkzeuge" das Element "Ausgang" mit einem Klick aus.
5. Begeben Sie sich mit der Maus in den Rumpf des Editors "PlcProg" und klicken auf die linke Maustaste, damit der Ausgang dort eingefügt wird.
6. Fügen Sie fünf weitere Ausgänge in "PlcProg" ein.
7. Geben Sie den Ausgängen die Bezeichnung "A1_green", "A1_yellow", "A1_red", "A2_green", "A2_yellow", "A2_red".



"A1_green" heißt korrekt "GVL.A1_green" oder kurz ".A1_green" und so weiter, da es sich um globale Variablen handelt.

Das System nimmt jedoch, wenn kein "A1_green" lokal deklariert ist, ersatzweise die globale Variable gleichen Namens.

Um das Netzwerk fertigzustellen gehen Sie wie folgt vor:

Erste Schritte - IndraLogic

1. Klicken Sie den Ausgang des Elements "ON" mit der linken Maustaste an und halten Sie diese gedrückt. Bewegen Sie den Mauszeiger auf den Eingang "START" des Bausteins "SEQUENCE", damit die beiden Elemente miteinander verbunden werden.
2. Klicken Sie den Ausgang "TRAFFICSIGNAL1" des Elementes "SEQUENCE" mit der linken Maustaste an und halten Sie diese gedrückt. Bewegen Sie den Mauszeiger auf den Eingang "STATUS" des Bausteins "TRAFFICSIGNAL", damit die beiden Elemente miteinander verbunden werden.
3. Klicken Sie den Ausgang "TRAFFICSIGNAL2" des Elementes "SEQUENCE" mit der linken Maustaste an und halten Sie diese gedrückt. Bewegen Sie den Mauszeiger auf den Eingang "STATUS" des Bausteins "LIGHT2", damit die beiden Elemente miteinander verbunden werden.
4. Klicken Sie den Ausgang "GREEN" des Elementes "LIGHT1" mit der linken Maustaste an und halten Sie diese gedrückt. Bewegen Sie den Mauszeiger auf den Eingang "A1_green" des Ausgangs, damit die beiden Elemente miteinander verbunden werden.
5. Wiederholen Sie den letzten Schritt entsprechend für die anderen Ausgänge.

Das Programm "PlcProg" sieht nun folgendermaßen aus:

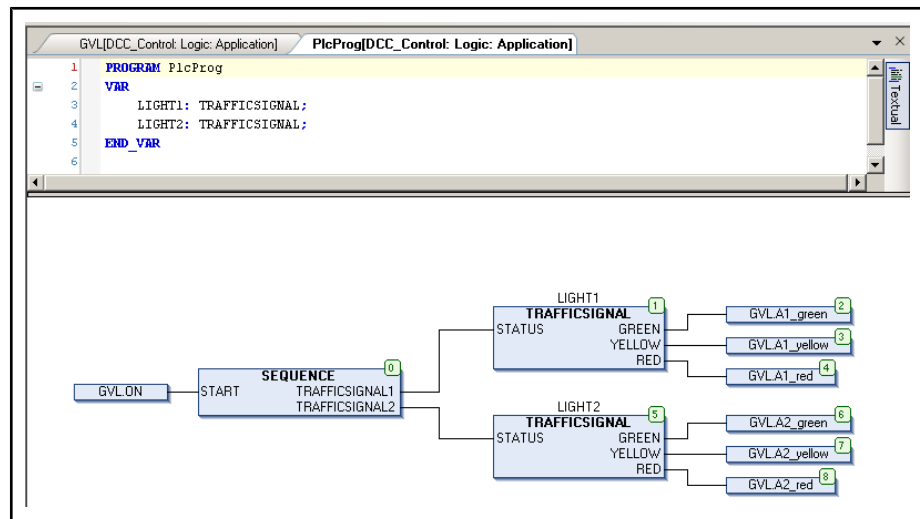


Abb.4-62: POU "PlcProg" fertiggestellt



Die Elemente, wie z. B. "Eingang", "Ausgang" und "Baustein" können auch über das Menü **CFC** eingefügt werden.

Testen der Applikation im Simulationsmodus, PlcProg in CFC

"Application" übersetzen

Um die Applikation in die Steuerung laden zu können, muss die Applikation zunächst fehlerfrei vom "Compiler" übersetzt werden. Um die Applikation zu übersetzen, wählen Sie im Menü **Erstellen ► Application übersetzen**. In der "Aufgabenliste" unterhalb des Arbeitsfensters steht nach dem Übersetzen: "Übersetzung abgeschlossen -- 0 Fehler, 0 Warnungen: Bereit zum Download!"



Es wird immer nur die Applikation übersetzt, die gerade aktiv ist.

Sind **mehrere** Applikationen **verfügbar**, muss die Applikation, die übersetzt werden soll, aktiv gesetzt werden (Kontext-Menü "Application" **Aktive Application setzen**).

Erste Schritte - IndraLogic

- Steuerung simulieren** Da zum Testen eventuell keine Steuerung vorhanden ist, kann man eine Steuerung simulieren. Um eine Steuerung zu simulieren, klicken Sie im "Projektextplorer" auf die Steuerung "DCC_Control" und öffnen Sie mit einem Klick auf die rechte Maustaste das Kontextmenü. Klicken Sie auf **Simulation**, um den Simulationsmodus zu aktivieren. Ist die Simulation aktiv, wird ein Haken bei "Simulation" gesetzt.
- In Steuerung einloggen** Nun soll sich auf die Steuerung eingeloggt und die Applikation in die Steuerung geladen werden. Um sich auf die Steuerung einzuloggen, wählen Sie im Menü **Debuggen ► Einloggen**. Sie können die neue Applikation sowohl per Online-Änderung als auch neu übersetzt in die Steuerung laden.
- "Application" starten** Um die Applikation zu starten, klicken Sie mit der rechten Maustaste auf die Applikation und wählen im Kontextmenü **Start**. Die "Application" läuft nun als Simulation ab.
- Die Ampel einschalten und Ampelzyklus beobachten** Nach dem Starten der Applikation ist die "LIGHT1" im Zustand "Grün" und die "LIGHT2" im Zustand "Rot". Die Ampel muss eingeschalten werden, damit der Ampelzyklus durchlaufen wird. Um die Ampel einzuschalten, gehen Sie wie folgt vor:
1. Öffnen Sie durch einen Doppelklick auf die "GVL" im Projektextplorer die "Globale Variablen Liste".

Ausdruck	Datentyp	Wert	Vorbereiteter Wert
ON	BOOL	FALSE	
A1_green	BOOL	TRUE	
A1_yellow	BOOL	FALSE	
A1_red	BOOL	FALSE	
A2_green	BOOL	FALSE	
A2_yellow	BOOL	FALSE	
A2_red	BOOL	TRUE	

Abb.4-63: "GVL" online

2. Klicken Sie bei der Variablen "ON" in die Spalte "Vorbereiteter Wert", so dass in dieser Spalte "TRUE" erscheint.
3. Um den Wert auf die Steuerung zu schreiben, klicken Sie im Menü **Debuggen ► Werte schreiben**.
Der Ampelzyklus wird nun durchlaufen. Im Baustein "PlcProg" können Sie den Ampelzyklus beobachten.

Erste Schritte - IndraLogic

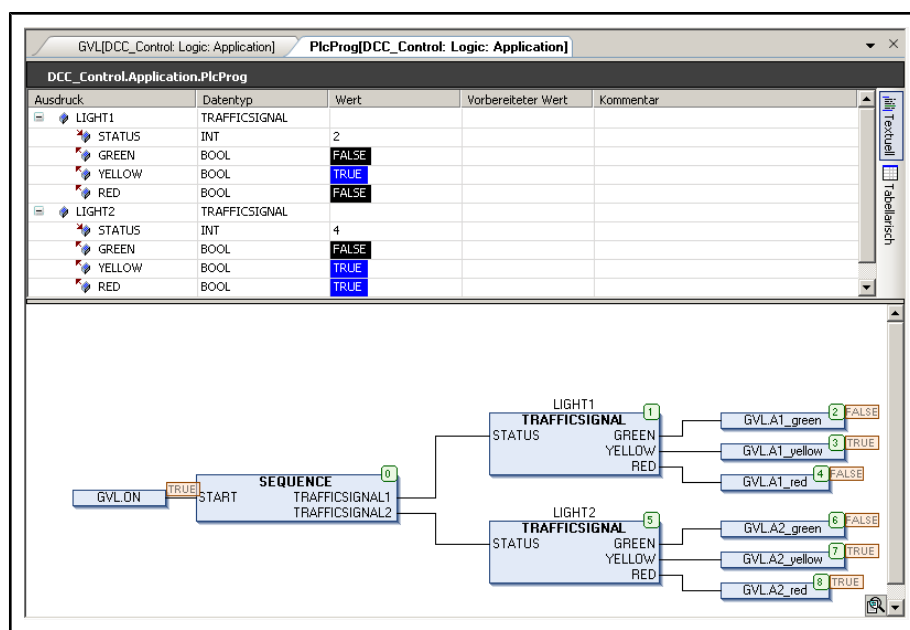


Abb.4-64: Programm "PlcProg" online

- Applikation stoppen** Stoppen Sie das SPS-Programm über die Menüleiste mit **Debuggen ► Stop**.
- Aus Steuerung ausloggen** Um sich aus der Steuerung auszuloggen, wählen Sie aus der Menüleiste **Debuggen ► Ausloggen** aus.

4.6.8 Testen der Applikation auf der Steuerung

- Simulationsmodus deaktivieren** Falls vorhanden, soll die Applikation in einer Steuerung getestet werden. Dazu müssen die [Kommunikationseinstellungen im Projekt, Seite 29](#), und der gewünschten Steuerung übereinstimmen. Beenden Sie den Simulationsmodus, indem Sie im "Projektexplorer" auf die Steuerung "DCC_Control" mit einem Klick auf die rechte Maustaste das Kontextmenü öffnen. Klicken Sie auf **Simulation** um den Simulationsmodus zu deaktivieren. Ist die Simulation deaktiviert, wird kein Haken bei "Simulation" gesetzt.
- "Application" übersetzen** Damit Sie die Applikation in die Steuerung laden können, muss die Applikation zunächst fehlerfrei vom "Compiler" übersetzt werden. Um die Applikation zu übersetzen, wählen Sie im Menü **Erstellen ► Application übersetzen**. In der "Aufgabenliste" unterhalb des Arbeitsfensters steht nach dem Übersetzen: "Übersetzung abgeschlossen -- 0 Fehler, 0 Warnungen: Bereit zum Download!"



Es wird immer nur die Applikation übersetzt, die gerade aktiv ist.

Sind **mehrere** Applikationen **verfügbar**, muss die Applikation, die übersetzt werden soll, aktiv gesetzt werden (Kontext-Menü "Application" **Aktive Application setzen**).

- In Steuerung einloggen** Nun soll sich auf die Steuerung eingeloggt und die Applikation in die Steuerung geladen werden. Um sich auf die Steuerung einzuloggen, wählen Sie im Menü **Debuggen ► Einloggen**.

Die Applikation steht danach im Zustand "STOP".

- "Application" starten** Um die Applikation zu starten, klicken Sie mit der rechten Maustaste auf die Applikation und wählen im Kontextmenü **Start**. Die Applikation läuft nun auf der Steuerung ab.

Erste Schritte - IndraLogic

Bootapplikation erzeugen

Erzeugen Sie über die Menüleiste ein Bootprojekt mit **Debuggen ► Bootapplikation erzeugen**. Wird dieser Befehl online ausgeführt, wird das kompilierte Projekt so auf der Steuerung abgelegt, dass es die Steuerung bei einem Neustart automatisch laden kann.

Die Ampel einschalten und Ampelzyklus beobachten

Nach dem Starten der Applikation ist die "Ampel 1" im Zustand "Grün" und die "Ampel 2" im Zustand "Rot". Die Ampel muss eingeschalten werden, damit der Ampelzyklus durchlaufen wird. Um die Ampel einzuschalten, gehen Sie wie folgt vor:

1. Öffnen Sie durch einen Doppelklick auf die "GVL" im Projektextplorer die Globale Variablen Liste.

Ausdruck	Datentyp	Wert	Vorbereiteter Wert
ON	BOOL	FALSE	
A1_green	BOOL	TRUE	
A1_yellow	BOOL	FALSE	
A1_red	BOOL	FALSE	
A2_green	BOOL	FALSE	
A2_yellow	BOOL	FALSE	
A2_red	BOOL	TRUE	

Abb.4-65: "GVL" Online

2. Klicken Sie bei der Variablen "Ein" in die Spalte "Vorbereiteter Wert", so dass in dieser Spalte "TRUE" erscheint.
3. Um den Wert auf die Steuerung zu schreiben, klicken Sie im Menü **Debuggen ► Werte schreiben**.

Der Ampelzyklus wird nun durchlaufen.

Sie können die Abarbeitung des Ampelzyklus in den einzelnen Programm-POUs (PlcProg, SEQUENCE) und der Globalen Variablenliste (GVL) verfolgen.

Öffnen Sie den Funktionsbaustein TRAFFICSIGNAL. Hier wird Ihnen zunächst ein Auswahlfenster mit der Alternative "PlcProg.LIGHT1" oder "PlcProg.LIGHT2" angeboten.

Erste Schritte - IndraLogic

The screenshot displays two windows from the IndraLogic software. The top window is titled 'DCC_Control.Application.PlcProg.LIGHT1' and contains a table with the following data:

Ausdruck	Kommentar	Datentyp	Wert	Vorbereiteter Wert
STATUS		INT	1	
GREEN		BOOL	TRUE	
YELLOW		BOOL	FALSE	
RED		BOOL	FALSE	

Below the table is a logic diagram showing a variable 'STATUS' with value '1' connected to an 'EQ' (Equality) block. The output of the 'EQ' block is 'GREEN' with value 'TRUE'.

The bottom window is titled 'DCC_Control.Application.PlcProg.LIGHT2' and contains a table with the following data:

Ausdruck	Kommentar	Datentyp	Wert	Vorbereiteter Wert
STATUS		INT	3	
GREEN		BOOL	FALSE	
YELLOW		BOOL	FALSE	
RED		BOOL	TRUE	

Below the table is a logic diagram showing a variable 'STATUS' with value '3' connected to an 'EQ' (Equality) block. The output of the 'EQ' block is 'GREEN' with value 'FALSE'.

Abb.4-66: Statusanzeige in den beiden FB-Instanzen von TRAFFICSIGNAL

Kontrolle der realen Ein- und Ausgänge

Die Steuerungen IndraMotion MLC / IndraLogic XLC L45 und L65 besitzen on-board Ausgänge, die den Adressen %QX0.0 ... %QX0.7 zugeordnet sind (Mapping). Sie können an diesen Ausgängen zusätzlich den Ampelzyklus verfolgen.

Applikation stoppen
Aus Steuerung ausloggen

Stoppen Sie das SPS-Programm über die Menüleiste mit **Debuggen ► Stop**.
Um sich aus der Steuerung auszuloggen, wählen Sie aus der Menüleiste **Debuggen ► Ausloggen** aus.

4.7 Visualisierung zum SPS-Programm erstellen

4.7.1 Visualisierung in IndraLogic

Mit der Visualisierung von IndraLogic können Sie dynamische Vorgänge schnell und einfach visuell darstellen. Hierzu wird der Status der Variablen in bestimmten zeitlichen Abständen abgefragt und die entsprechenden Grafikelemente werden ihren Zuständen entsprechend dargestellt. Das in ["SPS-Programm erstellen," Seite 32](#), beschriebene SPS-Programm soll visualisiert werden. Hierzu sollen nachfolgend zur programmierten Ampelanlage zwei Ampeln und ein EIN-Schalter gezeichnet und ihren Zuständen entsprechend dargestellt werden.

4.7.2 Erstellen einer neuen Visualisierung

Um eine neue Visualisierung zu erstellen, gehen Sie wie folgt vor:

1. Fügen Sie eine Visualisierung in die Applikation ein, indem Sie aus der "Bibliothek (SPS Objekte/ VI Logic Objekte)" das Element "Visualisierung" in den "Projektfexplorer" auf die Applikation ziehen.

Folgendes Fenster öffnet sich:

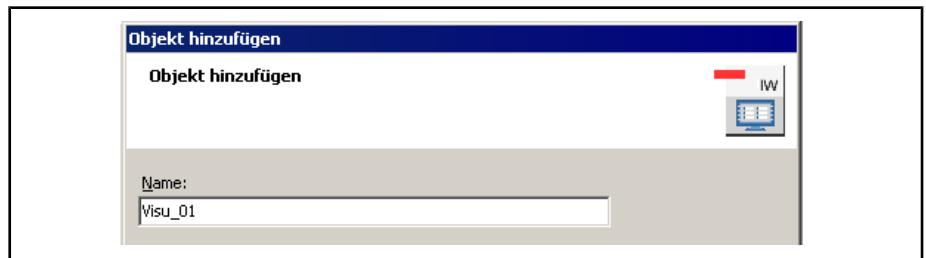


Abb.4-67: Visualisierung der Applikation hinzufügen

2. Geben Sie der Visualisierung einen beliebigen Namen und bestätigen Sie mit "Fertigstellen".

Die Visualisierung wird gemeinsam mit einer Instanz des Visualisierungsmanagers in die Applikation eingefügt:

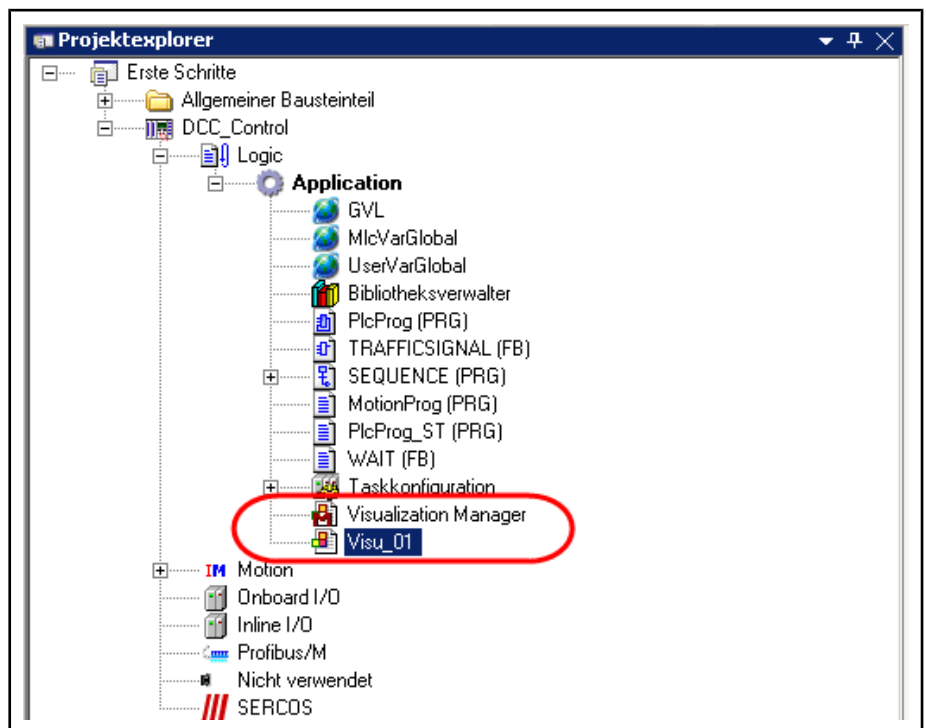


Abb.4-68: Visualisierung in Applikation eingefügt

Weitere hier nicht benötigte Elemente werden zusätzlich im Ordner "Allgemeiner Baustein" generiert.

4.7.3 Visualisierung editieren

Erstes Element "Ampellampe" einfügen und editieren

Visualisierungseditor öffnen

Öffnen Sie den Visualisierungseditor mit einem Doppelklick auf die Visualisierung im Projektexplorer.

Angeboten wird ein "Blatt zum Zeichnen", bei dem sich die Position (X, Y)= (0, 0) in der linken oberen Ecke befindet.

Über **Extras** ► **Optionen** ► **IndraLogic 2G** ► **Visualisierung** kann ein Raster sichtbar gemacht und aktiviert werden.

Angeboten werden drei Ordner in der Werkzeug-Box:

Erste Schritte - IndraLogic

- Basis, allgemeine Visualisierungsobjekte,
- Windows Control, Elemente zur Gestaltung eines Dialoges,
- Complex Control, komplexe Elemente z.B. "Messinstrumente", Trace-Objekte.

Element Ellipse einfügen

Es soll das erste Element "Lampe" der "Ampelleuchte" in die Visualisierung eingefügt werden. Elemente, die in eine Visualisierung eingefügt werden können, befinden sich im Fenster "Werkzeuge", siehe nachfolgende Abbildung.

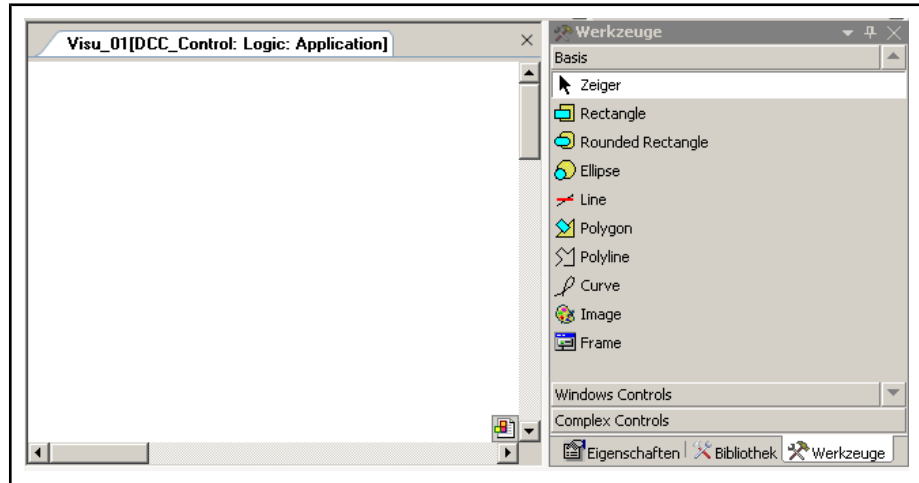


Abb.4-69: Fenster "Werkzeuge"

Um das erste Element "Lampe" einzufügen, gehen Sie wie folgt vor:

1. Wählen Sie im Fenster "Werkzeuge" das Element "Ellipse" mit einem Klick auf die linke Maustaste aus.
2. Bewegen Sie die Maus in das Fenster der Visualisierung und klicken Sie erneut auf die linke Maustaste.

Ein Element "Ellipse" wurde in den Visualisierungseitor eingefügt.

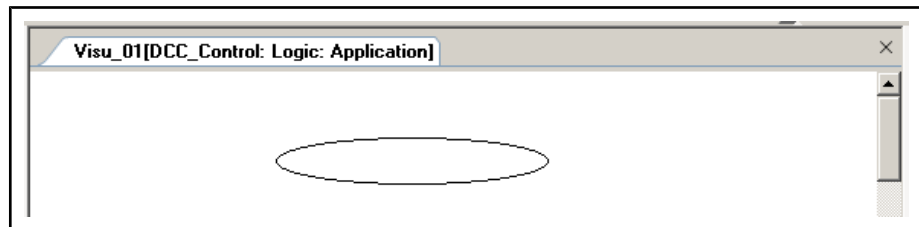


Abb.4-70: Editor Visualisierung – Erste Ellipse

Form der Ellipse verändern

Die eingefügte "Ellipse" soll die entsprechende Form einer "Ampellampe" bekommen.

Um die Form und Position der "Ellipse" zu verändern, gehen Sie wie folgt vor:

1. Markieren Sie die "Ellipse", indem Sie mit der linken Maustaste auf die "Ellipse" klicken.
2. Öffnen Sie das Fenster "Eigenschaften", indem Sie auf das Register "Eigenschaften" klicken.

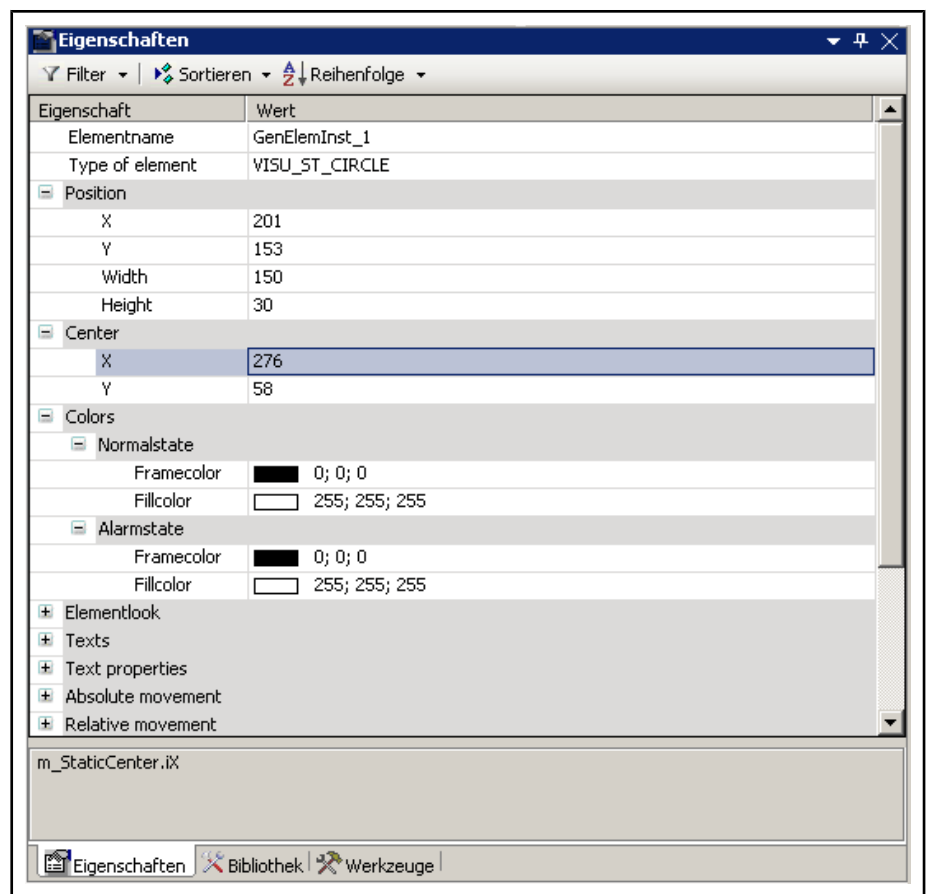


Abb.4-71: Fenster "Eigenschaften" der Ellipse

3. Um die Position (X, Y) und die Größe (Width, Height) zu verändern, wählen Sie unter der Eigenschaft "Position" für
 - "X" = "60"
 - "Y" = "150"
 - "Width" = "30"
 - "Height" = "30".
4. Um den Mittelpunkt der "Ellipse" zu verändern, wählen Sie unter Eigenschaft "Center" (X, Y) für
 - "X" = "75"
 - "Y" = "165".

Eigenschaften der Ampellampe verändern

Die Füllfarbe der Ampellampe soll verändert werden. Die Ampellampe soll grau sein, wenn Sie nicht aktiv ist, und soll rot sein, wenn sie aktiv ist.

Um die Füllfarben zu verändern gehen Sie wie folgt vor:

1. Um die Füllfarbe für den Normalzustand (Normalstate) zu verändern, klicken Sie unter Eigenschaft "Colors" und "Normalstate" unter "Fillcolor" auf die Schaltfläche "...".
Das Fenster "Farbe" öffnet sich.
2. Klicken Sie im Fenster "Farbe" auf die Schaltfläche "FillColors". Wählen Sie "Silver" (grau).

Erste Schritte - IndraLogic

Um die Ampellampe zu kopieren und einzufügen gehen Sie wie folgt vor:

1. Markieren Sie mit einem Mausklick die Ampellampe.
2. Öffnen Sie mit der rechten Maustaste das Kontextmenü und wählen Sie **Kopieren** aus.
3. Klicken Sie in einen Bereich des Visualisierungseditors und wählen Sie über das Kontextmenü **Einfügen**.
4. Klicken Sie nochmals in einen freien Bereich des Visualisierungseditors und wählen Sie im Kontextmenü **Einfügen**, um eine weitere Ampellampe einzufügen.
5. Markieren Sie mit einem linken Mausklick die erste eingefügte Ampellampe.

Elemente "Ampellampe" Position ändern

Die beiden neu eingefügten Ampelleuchten müssen nun positioniert werden.

Um die Ampelleuchten zu positionieren, gehen Sie wie folgt vor:

1. Markieren Sie mit der linken Maustaste die erste eingefügte Ampellampe.
2. Um die Position zu verändern, wählen Sie im Fenster "Eigenschaften" unter der Eigenschaft "Position" für $(X, Y) = (60, 195)$.
3. Um den Mittelpunkt zu verändern, wählen Sie unter Eigenschaft "Center" für $(X, Y) = (75, 210)$.
4. Markieren Sie mit der linken Maustaste die zweite eingefügte Ampellampe.
5. Um die Position zu verändern, wählen Sie im Fenster "Eigenschaften" unter der Eigenschaft "Position" für $(X, Y) = (60, 240)$.
6. Um den Mittelpunkt zu verändern, wählen Sie unter Eigenschaft "Center" für $(X, Y) = (75, 255)$.

Elemente "Ampellampe" Eigenschaften ändern

Die beiden neuen Ampellampen bekommen neue Farben für den "Alarmstate" zugewiesen.

Gehen Sie wie folgt vor:

1. Markieren Sie die mittlere Ampellampe.
2. Um die Füllfarbe für den Alarmzustand (Alarmstate) zu verändern, klicken Sie unter Eigenschaft "Colors" und "Alarmstate" unter "FillColor" auf die Schaltfläche "...".
3. Das Fenster "Color" öffnet sich.
4. Klicken Sie im Fenster "Farbe" auf die Schaltfläche "FillColor". Wählen Sie "Yellow" (gelb).
5. Wiederholen Sie Schritt 1 - 3 für die untere Ampellampe
6. Klicken Sie im Fenster "Farbe" auf die Schaltfläche "FillColor". Wählen Sie "Green" (grün).

Die Variablen zum Ansprechen der verschiedenen Ampelfarben müssen Sie noch anpassen.

Gehen Sie hierzu wie folgt vor:

1. Markieren Sie die mittlere Ampellampe.
2. Um der Visualisierung die Variable zur Änderung der Signalfarbe mitzuteilen, klicken Sie doppelt unter der Eigenschaft "Colorvariables" und "ToggleColor" in die Spalte "Wert".
Klicken Sie auf die Schaltfläche "...".
Der Dialog "Eingabehilfe" öffnet sich.

Erste Schritte - IndraLogic

3. Unter "Elemente" im Fenster der "Eingabehilfe" sehen Sie "Application", "GVL".
4. Wählen Sie unter den "GVL-Variablen" die Variable "A1_yellow" aus.
5. Markieren Sie die untere Ampellampe.
6. Wählen Sie unter den "GVL-Variablen" die Variable "A1_green" aus.

Element "Ampelgehäuse" einfügen und editieren

Elemente des Ampelgehäuses Die drei Lampen zur Darstellung der Signalzustände sind nun fertiggestellt. Nun soll die Ampel auch als solche erkennbar sein. Es werden nun nachfolgend drei Elemente eingefügt, die zur visuellen Darstellung einer Ampelanlage dienen, diese sind:

1. das Gehäuse der Signallampen,
2. der Mast für das Ampelgehäuse sowie
3. der Ampelfuß.

Element "Rectangle" einfügen Um das Gehäuse der Signallampen einzufügen gehen Sie wie folgt vor:

1. Wählen Sie aus dem Fenster "Werkzeuge" das Element "Rectangle" aus.
2. Klicken Sie in einen freien Bereich des Visualisierungseditors, damit das Element "Rectangle" eingefügt wird.

Elemente "Ampelgehäuse" Position ändern Um die Position und die Form des Gehäuses zu verändern, gehen Sie wie folgt vor:

1. Markieren Sie das eingefügte Element "Rectangle" und öffnen Sie das Fenster "Eigenschaften".
2. Um die Position zu verändern, wählen Sie im Fenster "Eigenschaften" unter der Eigenschaft "Position" für (X, Y)= (50, 135).
3. Um die Form zu verändern, wählen Sie im Fenster "Eigenschaften" unter der Eigenschaft "Position" für (Width, Height)= (50, 150).
4. Um den Mittelpunkt zu verändern, wählen Sie unter Eigenschaft "Center" für (X, Y)= (50, 210).
5. Klicken Sie bei dem markierten Objekt auf die rechte Maustaste, um das Kontextmenü zu öffnen. Wählen Sie **Folge ► Nach hinten legen**.

Elemente "Ampelgehäuse" Eigenschaften ändern Um die Füllfarbe für das Gehäuse der Signallampen zu ändern, gehen Sie wie folgt vor:

1. Klicken Sie im Fenster "Eigenschaften" unter "Colors", "Normalstate" und "FillColor" auf die Schaltfläche "...".
Das Fenster "Color" öffnet sich.
2. Klicken Sie im Fenster "Color" auf die Schaltfläche "Farben definieren".
3. Für die Felder "Rot", "Grün" und "Blau" geben Sie folgendes ein:
(Rot, Grün, Blau)= (5, 5, 10).

Element "Ampelmast" einfügen und editieren

Element "Rectangle" einfügen Als nächstes wird der Mast des Ampelgehäuses konstruiert.

Gehen Sie dabei wie folgt vor:

1. Wählen Sie aus dem Fenster "Werkzeuge" das Element "Rectangle" aus.
2. Klicken Sie in einen freien Bereich des Visualisierungseditors damit das Element "Rectangle" eingefügt wird.

Element "Ampelmast" Position ändern 1. Markieren Sie das eingefügte Element "Rectangle" und öffnen Sie das Fenster "Eigenschaften".

2. Um die Position zu verändern, wählen Sie im Fenster "Eigenschaften" unter der Eigenschaft "Position" für (X, Y) = (70, 285).
3. Um die Form zu verändern, wählen Sie im Fenster "Eigenschaften" unter der Eigenschaft "Position" für (Width, Height) = (10, 195).
4. Um den Mittelpunkt zu verändern, wählen Sie unter Eigenschaft "Center" für (X, Y) = (75, 380).

Element "Ampelmast" Eigenschaften ändern

Als Füllfarbe für den Mast der Signallampen wählen Sie bitte Dunkelgrau (Dim-Gray) (Colors, Normalstate, Fillcolor, (Rot, Grün, Blau) = (105, 105, 105)).

Element "Ampelfuß" einfügen und editieren

Element "Rectangle" einfügen

Für die erste Ampel fehlt noch der Ampelfuß.

Um den Ampelfuß zu konstruieren, gehen Sie wie folgt vor:

1. Wählen Sie aus dem Fenster "Werkzeuge" das Element "Rectangle" aus und fügen es im Arbeitsfeld ein.
2. Markieren Sie das eingefügte Element "Rectangle" und öffnen Sie das Fenster "Eigenschaften".
3. Um die Position zu verändern, wählen Sie im Fenster "Eigenschaften" unter der Eigenschaft "Position" für (X, Y) = (15, 480).
4. Um die Form zu verändern, wählen Sie im Fenster "Eigenschaften" unter der Eigenschaft "Position" für (Width, Height) = (120, 30).
5. Um den Mittelpunkt zu verändern, wählen Sie unter Eigenschaft "Center" für (X, Y) = (75, 495).

Element "Ampelfuß" Eigenschaften ändern

Als Füllfarbe für den Ampelfuß der Signallampen wählen Sie bitte Grau (Dim-Grey) (Colors, Normalstate, Fillcolor, (Rot, Grün, Blau) = (105, 105, 105)).

Die erste Ampel hat nun folgendes Erscheinungsbild:

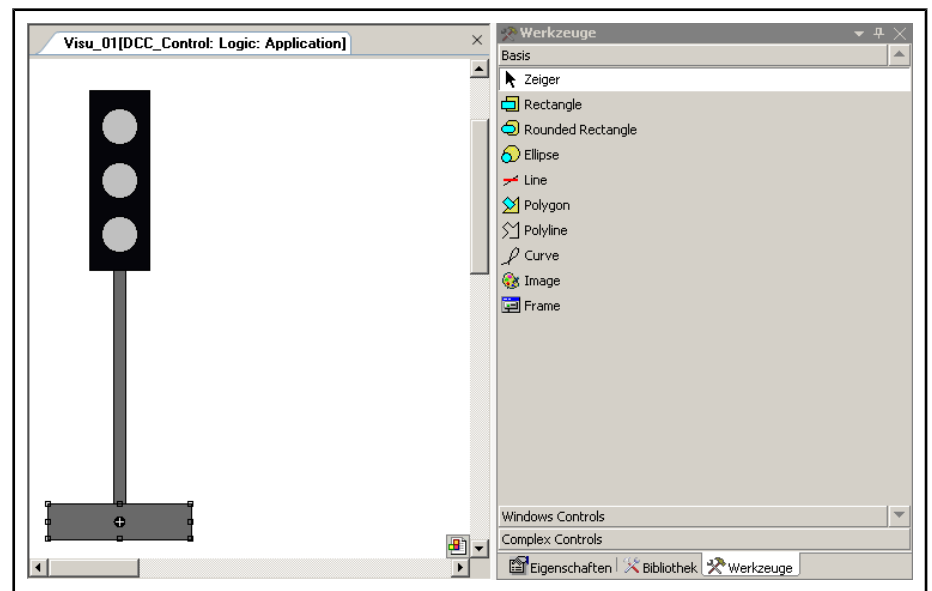


Abb.4-73: Erste Ampel

Zweite Ampel einfügen und editieren

Die zweite Ampel

Um die zweite Ampel zu erstellen, markieren Sie alle Elemente der Ampel. Hierzu klicken Sie mit der linken Maustaste in einen Bereich unten rechts neben die Ampel und halten die Maustaste gedrückt. Fahren Sie mit der Maus links oberhalb der Ampel. Lassen Sie die Maus los. Die Elemente sind nun alle markiert.

Erste Schritte - IndraLogic



Alternativ können Sie die Elemente markieren, indem Sie mit gedrückter Taste <Strg> jedes einzelne Element markieren.

Wählen Sie über das Kontextmenü **Kopieren** und **Einfügen**. Nehmen Sie das markierte Objekt und legen Sie es neben der ersten Ampel ab.

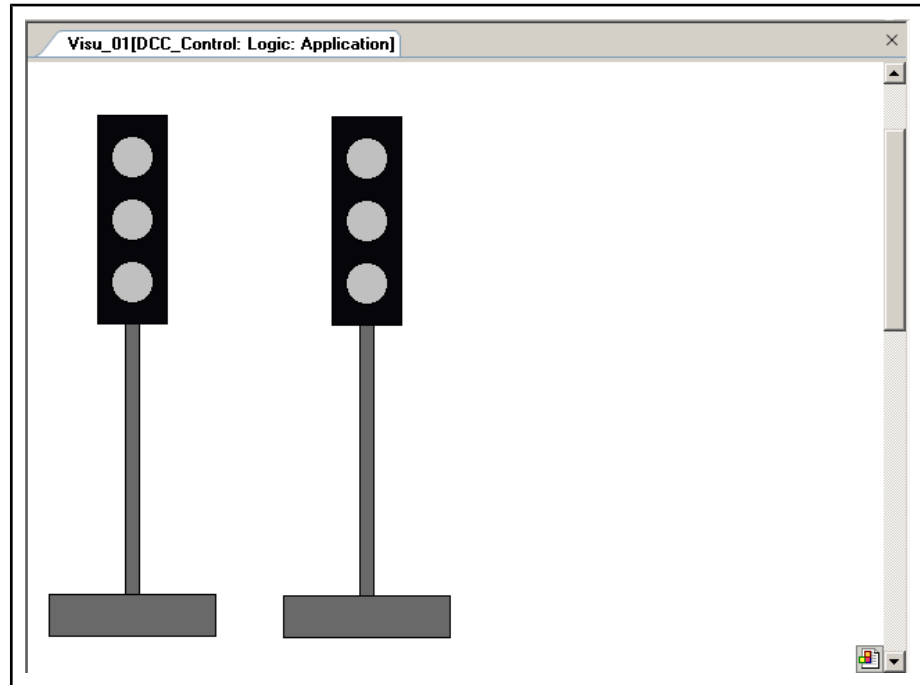


Abb.4-74: Zwei Ampeln

Sie müssen noch die Variablen für die Umschaltung der verschiedenen Ampelphasen anpassen. Ändern Sie die Variablen im Fenster "Eigenschaften" bei "ToggleColor" entsprechend ab.

"GVL.A1_red" wird zu "GVL.A2_red",

"GVL.A1_yellow" wird zu "GVL.A2_yellow" und

"GVL.A1_green" wird zu "GVL.A2_green".

EIN-Schalter einfügen und editieren

Element "Button" einfügen

Die Ampel soll zu Beginn eingeschaltet werden können. Hierzu soll ein "Button" eingefügt werden:

1. Wählen Sie aus dem Fenster **Werkzeuge ► Windows Control** das Element "Button" aus.
2. Klicken Sie in einen freien Bereich des Visualisierungseditors damit das Element "Button" eingefügt wird.

Element "Button" Position ändern

1. Um die Position zu verändern, wählen Sie im Fenster "Eigenschaften" unter der Eigenschaft "Position" für (X, Y)= (130, 55).
2. Um die Form zu verändern, wählen Sie im Fenster "Eigenschaften" unter der Eigenschaft "Position" für (Width, Height)= (70, 40).
3. Um den Mittelpunkt zu verändern, wählen Sie unter Eigenschaft "Center" und für (X, Y)= (165, 75).

Element "Button" Eigenschaften ändern

1. Unter der Eigenschaft "Texts" und "Text" geben Sie "EIN" ein.
2. Unter der Eigenschaft "Eingaben" und "OnMouseClicked" klicken Sie auf "Konfigurieren"

Erste Schritte - IndraLogic

Das Fenster "Eingabekonfiguration" öffnet sich.

3. Markieren Sie im Fenster "Eingabekonfiguration" im linken Fensterbereich "Variable umschalten" und betätigen Sie die Schaltfläche ">".
"Variable umschalten" wird in den rechten Fensterbereich eingefügt.
4. Klicken Sie in das Eingabefeld "Variable" und geben Sie "GVL.ON" ein.
5. Bestätigen Sie die Eingabe mit der Schaltfläche "OK".

Beschriftung einfügen und editieren

Element "Rectangle" einfügen

Die beiden Ampeln sollen nun noch eine Beschriftung "Ampel 1" und "Ampel 2" erhalten.

Um die Ampeln zu beschriften, gehen Sie wie folgt vor:

1. Wählen Sie aus dem Fenster "Werkzeuge" das Element "Rectangle" aus.
2. Klicken Sie in einen freien Bereich des Visualisierungseditors damit das Element "Rectangle" eingefügt wird.

Element "Rectangle" Position ändern

1. Markieren Sie das eingefügte Element "Rectangle" und öffnen Sie das Fenster "Eigenschaften".
2. Um die Position zu verändern, wählen Sie im Fenster "Eigenschaften" unter der Eigenschaft "Position" für (X, Y) = (15, 520).
3. Um die Form zu verändern, wählen Sie im Fenster "Eigenschaften" unter der Eigenschaft "Position" für (Width, Height) = (120, 30).
4. Um den Mittelpunkt zu verändern, wählen Sie unter Eigenschaft "Center" für (X, Y) = (75, 535).

Element "Rectangle" Eigenschaften ändern

1. Im Feld "Text" bei "Texts" geben Sie "Ampel 1" ein.
2. Klicken Sie im Fenster "Eigenschaften" unter "Colors", "Normalstate" und "Framecolor" auf die Schaltfläche "...".

Das Fenster "Farbe" öffnet sich.

3. Klicken Sie im Fenster "Farbe" auf die Schaltfläche "Farben definieren".
4. Für die Felder "Rot", "Grün" und "Blau" geben Sie folgendes ein: (Rot, Grün, Blau) = (255, 255, 255)

Element "Rectangle" kopieren und Eigenschaften anpassen

1. Kopieren Sie die Beschriftung, legen Sie sie unter der Ampel zwei ab.
2. Im Feld "Text" bei "Texts" geben Sie "Ampel 2" ein.

Die Visualisierung ist nun fertiggestellt:

Erste Schritte - IndraLogic

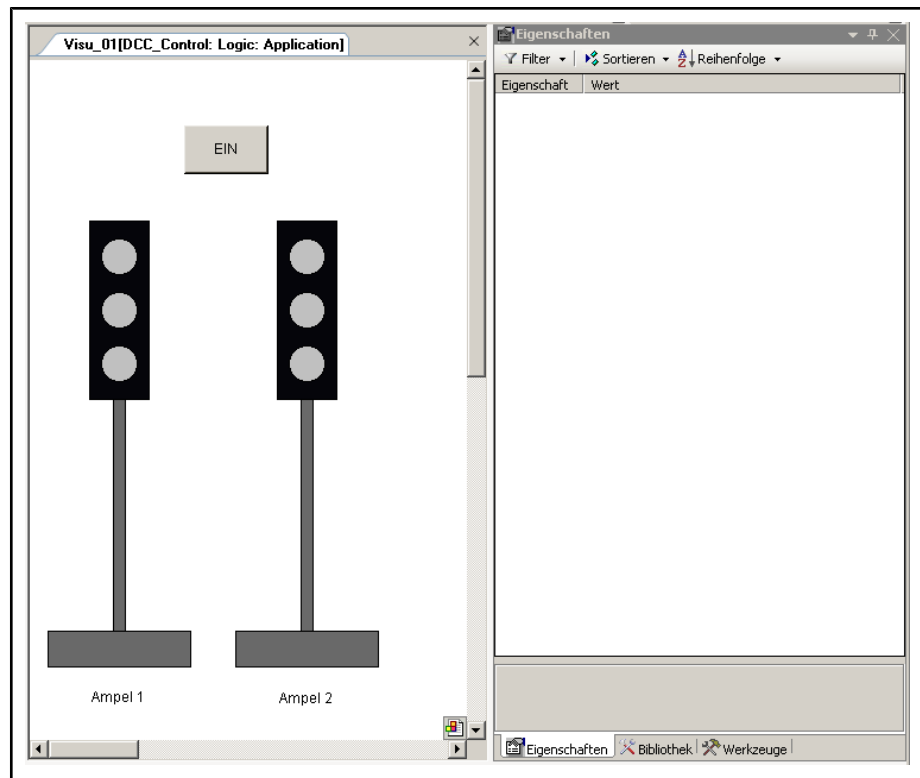


Abb.4-75: Editor Visualisierung fertiggestellt

Anzeige von Variablenwerten

Element "Rectangle" einfügen

Zur Inbetriebnahme einer Visualisierung oder als Dauerlösung kann es sinnvoll sein, den Wert einer Variablen formatiert anzuzeigen.

Die beiden Ampeln sollen unter ihrer Beschriftung "Ampel 1" und "Ampel 2" eine Anzeige ihrer jeweiligen Variablen "STATUS" erhalten.

Um die Ampeln zu beschriften, gehen Sie wie folgt vor:

1. Wählen Sie aus dem Fenster "Werkzeuge" das Element "Rectangle" aus.
2. Klicken Sie in einen freien Bereich des Visualisierungseditors damit das Element "Rectangle" eingefügt wird.

Element "Rectangle" Position ändern

1. Markieren Sie das eingefügte Element "Rectangle" und öffnen Sie das Fenster "Eigenschaften".
2. Um die Position zu verändern, wählen Sie im Fenster "Eigenschaften" unter der Eigenschaft "Position" für (X, Y) = (15, 570).
3. Um die Form zu verändern, wählen Sie im Fenster "Eigenschaften" unter der Eigenschaft "Position" für (Width, Height) = (120, 30).
4. Um den Mittelpunkt zu verändern, wählen Sie unter Eigenschaft "Center" für (X, Y) = (75, 585).

Element "Rectangle" Eigenschaften ändern

1. Im Feld "Text" bei "Texts" geben Sie "Ampel 1: %s" ein.
2. Klicken Sie ins Feld "Text Variable" bei "Text variables" und wählen über die Schaltfläche "..." die Variable "PlcProg.LIGHT1.STATUS" aus.

Der Wert der Variablen "PlcProg.LIGHT1.STATUS" erscheint anstelle der Formatangabe "%s" im Online-Modus.

Element "Rectangle" kopieren und Eigenschaften anpassen

1. Kopieren Sie die Anzeige, legen Sie sie unter der Ampel zwei ab.

Erste Schritte - IndraLogic

2. Ersetzen Sie für Anzeige 2:

- Text "Ampel 1: %s" durch "Ampel 2: %s",
- Text Variable "PlcProg.LIGHT1.STATUS" durch "PlcProg.LIGHT2.STATUS".

Visualisierung mit ergänzter Anzeige:

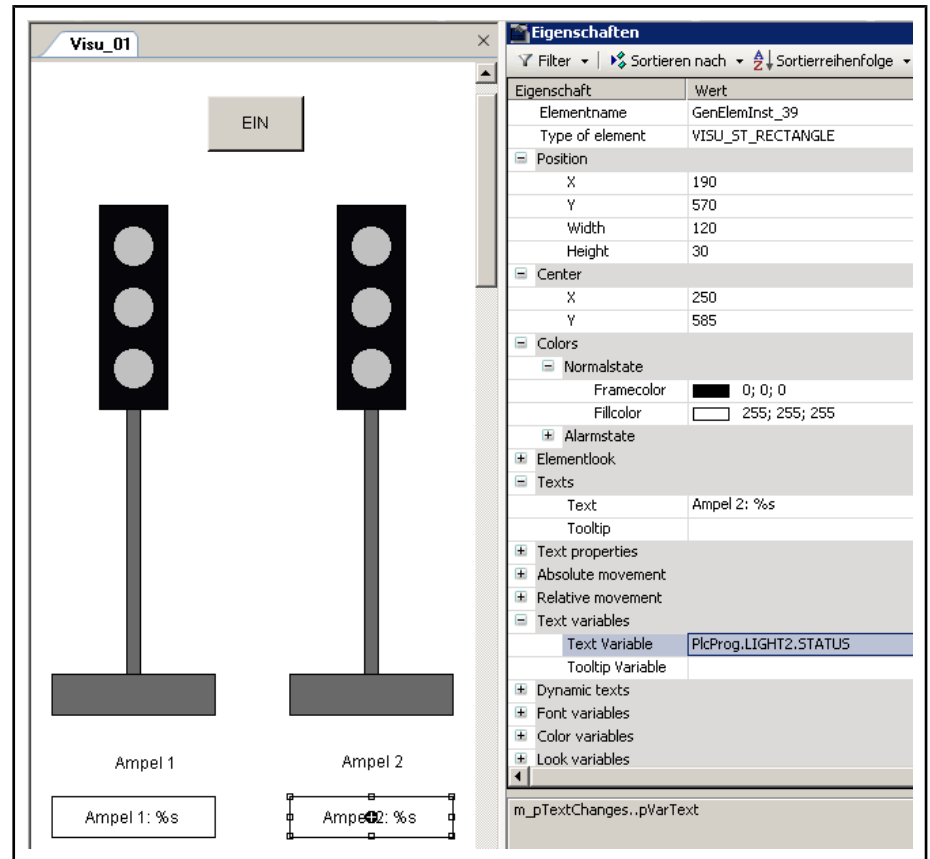


Abb.4-76: Editor Visualisierung durch Anzeige ergänzt

4.7.4 Visualisierung testen

Die Visualisierung kann im Simulationsmodus oder, falls vorhanden, mit einer realen Steuerung getestet werden.

Um die Visualisierung im Simulationsmodus zu testen, orientieren Sie sich an ["Testen der Applikation im Simulationsmodus \(PlcProg in ST\)"](#), Seite 63 bzw. ["Testen der Applikation im Simulationsmodus \(PlcProg in CFC\)"](#), Seite 68.

Um die Visualisierung in mit einer Steuerung zu testen, orientieren Sie sich an ["Testen der Applikation mit einer Steuerung"](#), Seite 70.

Öffnen Sie bei gestarteter Applikation das Fenster "Visualisierung":

Erste Schritte - IndraLogic

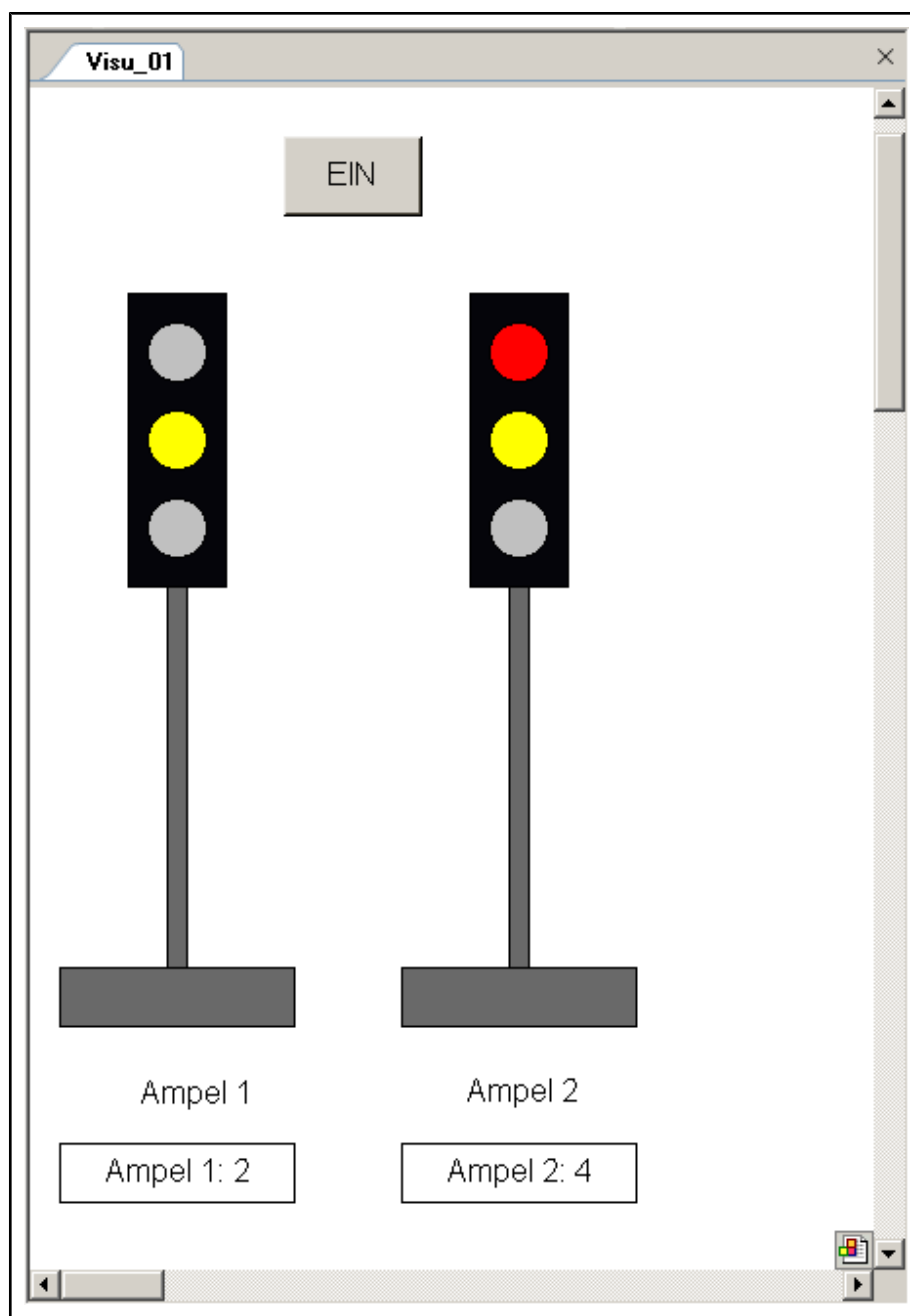


Abb.4-77: Visualisierung online

Klicken Sie auf die Schaltfläche "EIN". Die Ampelzyklen werden durchlaufen.

4.8 Projekt wiederherstellen

Ablage Ausgehend vom erstellten Projekt wurde eine Exportdatei erzeugt, die es unabhängig von Ihrer Steuerungshardware (XLC L25, L45, L65 (mit oder ohne SERCOS III-Bus) oder MLC L25, L45, L65) erlaubt, das Projekt wieder zu regenerieren. Das regenerierte Projekt kann als Simulation oder unter Nutzung Ihrer Steuerung laufen.

Die Exportdatei wurde vom Knoten "Application" beginnen erstellt und enthält folgende Bestandteile:

Erste Schritte - IndraLogic

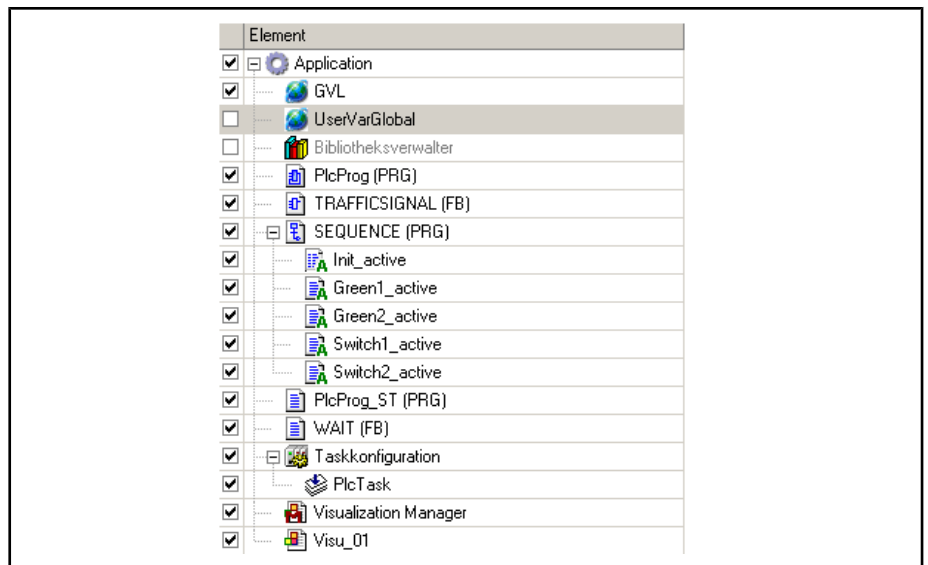


Abb.4-78: Bestandteile der Exportdatei

Die angegebenen Komponenten werden beim Import überschrieben.

Wiederherstellen

Um das Projekt auf einer Steuerung Ihrer Wahl (XLC L25, L45, L65 (mit oder ohne SERCOS III-Bus) oder MLC L25, L45, L65) lauffähig zu machen, ist etwas Eigenbeteiligung erforderlich:

1. Löschen von Altdaten in Ihrer Steuerung
Löschen Sie den alten Speicherinhalt Ihrer Steuerung.
Details sind unter [Steuerung zurücksetzen \(CMLx5\)](#), Seite 26, beschrieben.
Der Speicher der Steuerung ist geleert, eine eventuell vorhandene Motion-Komponente und die SPS-Komponente sind betriebsbereit.
2. Anlage eines neuen Projektes
Legen Sie ein neues leeres Projekt an. **Datei ► Neu ► Projekt**. Der Projektname ist beliebig, empfohlen wird "Erste Schritte".
Sie verfügen über ein neues leeres Projekt.
3. Anlegen der Steuerung
Ziehen Sie eine Steuerung Ihrer Wahl (XLC L25, L45, L65 (mit oder ohne SERCOS III-Bus) oder MLC L25, L45, L65) aus der Gerätebibliothek auf das leere Projekt. Der Steuerungsname ist beliebig, empfohlen wird "DCC_Control".
Komplettieren Sie den Gerätetyp und die IP-Adresse.
Führen Sie einen Verbindungstest durch. (Nicht erforderlich, wenn nur "simuliert" werden soll.
Schließen Sie den Einfügevorgang wie gewohnt ab.
In Ihrem Projekt befindet sich die gewünschte Steuerung.
4. Import der Exportdatei "Application_FirstSteps_IndraLogic_Basics.iwx"
Öffnen Sie im Projektnavigator den Steuerungsknoten bis **Logic ► Application**.
Kontextmenü "Application", **Import**.

Erste Schritte - IndraLogic

Wählen Sie die Exportdatei "Application_FirstSteps_IndraLogic_Basics.iwx" in "Iw:\Programme\Rexroth\IndraWorks\Projects" aus, Operation "Überschreiben", fertigstellen.

Die benötigten Dateien wurden kopiert, das Projekt ist im Simulationsmode und auf der Steuerung lauffähig.

Projekt testen

Das Projekt enthält "PlcProg" in CFC und "PlcProg_ST" in Strukturiertem Text (ST).

Um das Projekt im Simulationsmodus zu testen, orientieren Sie sich an ["Testen der Applikation im Simulationsmodus \(PlcProg in CFC\)"](#), Seite 63 bzw. ["Testen der Applikation im Simulationsmodus \(PlcProg in ST\)"](#), Seite 63.

Um das Projekt mit einer Steuerung zu testen, orientieren Sie sich an ["Testen der Applikation mit einer Steuerung"](#), Seite 70.

5 Erste Schritte - Motion

5.1 Ziel des Projekts "Erste Schritte - Motion"

Vorgegeben wird eine virtuelle Achse, die mit einer konstanten Geschwindigkeit laufen soll. Diese Achse arbeitet als Leitachse für eine reale Achse, die über ein Getriebe gekoppelt ist.

Die reale Achse enthält z.B. einen Antrieb "IndraDrive M", der über ein SERCOS III-Kabel mit der Steuerung verbunden ist.



Verfügen Sie über keinen Antrieb, entfällt der nachstehende Arbeitsschritt "Antriebsscannen".

Legen Sie an Stelle der realen Achse eine zweite virtuelle Achse an, und verwenden Sie diese anstelle der realen Achse.

Teilaufgaben

- IndraWorks Projekt und Steuerung und Antrieben anlegen,
- Laden des Motion-Projekts aus die Steuerung,
- Programmieren des Bewegungsablaufs,
- SPS-Programm auf die Steuerung laden und testen.
- Aufzeichnung der Bewegung der Achsen mittels Trace.



Die Programmierung der Bewegungsabläufe kann alternativ in Funktionsbausteinsprache (FUP) oder Strukturiertem Text (ST) erfolgen.

- [Programmieren des Bewegungsablaufes in Funktionsbausteinsprache \(FUP\), Seite 98,](#)
- [Programmieren des Bewegungsablaufes in Strukturiertem Text \(ST\), Seite 121.](#)

5.2 Technische Voraussetzungen

- PC mit installiertem IndraWorks Engineering für IndraMotion MLC, 12VRS,
Steuerung, Typ "IndraMotion MLC L25", "IndraMotion MLC L45" oder "IndraMotion MLC L65",
aktuelle Firmware ist geladen.

Die Schaffung dieser Voraussetzungen sind beschrieben in

- Rexroth IndraMotion MLC 12VRS Funktionsbeschreibung, DOK-IM*ML*-SYSTEM**V12-RExx-DE-P und
- IndraWorks Online-Hilfe, **System IndraLogic XLC / IndraMotion MLC ▶ XLC/MLC-Funktionsbeschreibung ▶ Inbetriebnahme und Bedienung,**

Steuerung, z.B. Typ "IndraMotion MLC L65.1", die aktuelle Firmware ist geladen,

- die IP-Adresse ist bekannt,
- die Steuerung ist über Ethernet mit dem Programmiergerät (PC) verbunden.

Antriebsregelgerät, z.B. Typ "IndraDrive M", die aktuelle Firmware ist geladen.

- das Antriebsregelgerät ist mit der Steuerung über SERCOS III verbunden.

Erste Schritte - Motion

- die Adresse des Antriebsregelgerätes am SERCOS-Bus wird gescannt.
oder
- ein gebrückter SERCOS-Bus, wenn kein Antrieb vorhanden ist.

5.3 Anlegen des Motion-Teilprojektes

5.3.1 Steuerung zurücksetzen (CMLx5)

Die Steuerung kann von vorheriger Nutzung alte Programmbestandteile, Parametrierungen und SPS-Daten enthalten, die in diesem Schritt durch das Speicherlöschen entfernt werden.

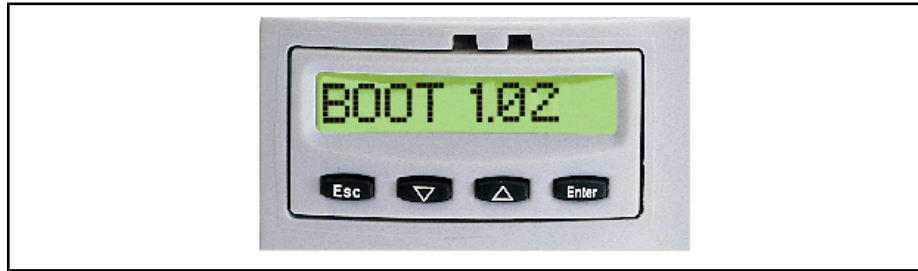


Abb.5-1: Reboot der Steuerung

Nach dem Einschalten der Steuerung sind die beiden äußeren Tasten unter dem Display (<Esc> und <Enter>) bei Erscheinen von **BOOT 1.02** zu drücken und so lange zu halten, bis die Ausgabe **BOOTSTOP** erscheint. Danach sind die beiden inneren Tasten zu drücken, es erscheint zunächst **CLEANUP**, dann **REBOOT**, und die Tasten können wieder losgelassen werden.

Die Steuerung läuft nun bis **BB STOP**, der Speicher ist geleert, die Motion-Komponente betriebsbereit und die SPS-Komponente in der Abarbeitung gestoppt.

5.3.2 IndraWorks starten

Starten Sie IndraWorks über **Start ▶ Programme ▶ Rexroth ▶ IndraWorks ▶ Engineering**.

Die Entwicklungsumgebung von IndraWorks wird gestartet. Alternativ kann IndraWorks über ein Desktop-Symbol gestartet werden.



Desktop-Symbol:

5.3.3 Neues Projekt erstellen

Um ein neues Projekt anzulegen, klicken Sie im Arbeitsbereich auf "Leeres Projekt erzeugen".

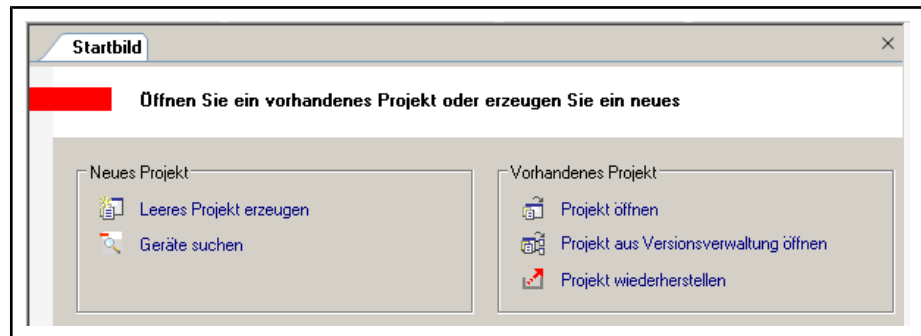


Abb.5-2: Leeres Projekt erzeugen

Erste Schritte - Motion

Alternativ können Sie ein neues Projekt über **Datei ► Neu ► Projekt...** anlegen. Der Assistent zur Vergabe des Projektnamens und der Speicherplatzeinstellungen für ein neues IndraWorks-Projekt startet, siehe folgende Abbildung.

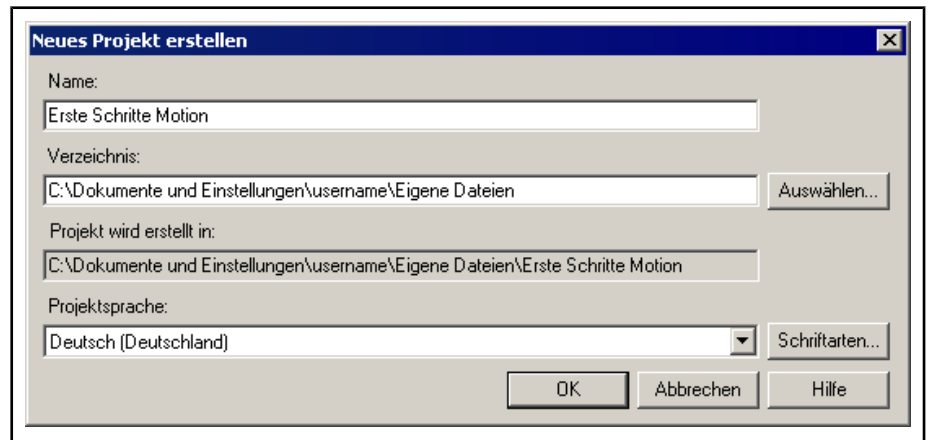


Abb.5-3: Einstellungen zum Abspeichern eines neuen IndraWorks-Projektes

- | | |
|---------------------------------|---|
| Name | Geben Sie dem Projekt einen beliebigen Namen, Vorschlag: "Erste Schritte - Motion". |
| Verzeichnis | Das Feld enthält voreingestellt einen Speicherplatz im Verzeichnis C:\Dokumente und Einstellungen\username\Eigene Dateien.
Alternativ können Sie auch ein anderes Verzeichnis angeben. |
| Projekt wird erstellt in | Verzeichnis und Projektname werden automatisch angelegt. |
| Projektsprache | Mit der Auswahl der Projektsprache legen Sie die Sprache ihrer Anwendertexte fest.

Das neu angelegte Projekt erscheint im Projektexplorer mit dem vorher definierten Projektnamen, siehe folgende Abbildung. |

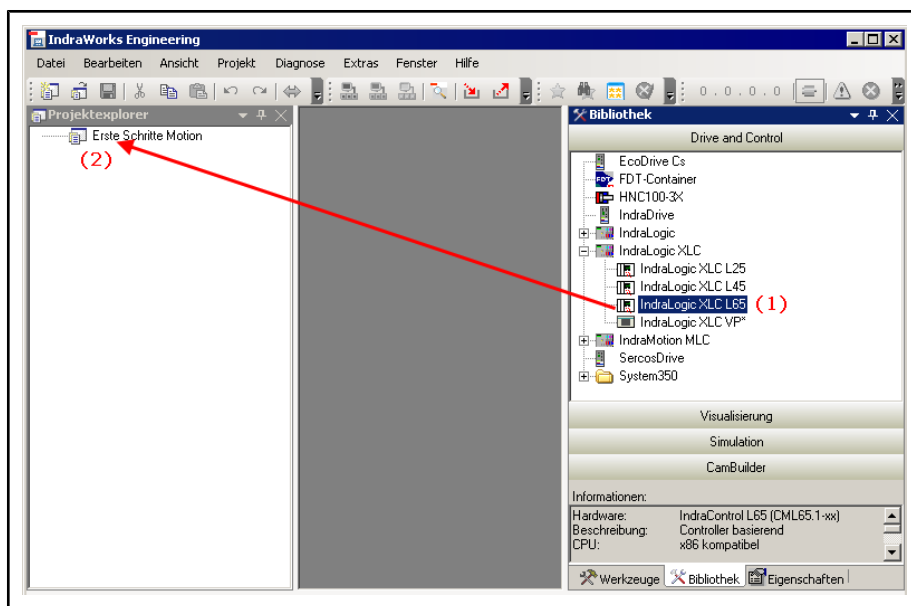


Abb.5-4: Erste Schritte - Motion-Projekt

5.3.4 Neues Gerät in ein Projekt einfügen

Ziehen Sie aus der Bibliothek "Drive and Control" das gewünschte Gerät (z.B. "IndraLogic XLC L65") in das neu angelegte Projekt im Projektexplorer.

Erste Schritte - Motion



- (1) Gerät: IndraLogic XLC L65
(2) Projekt: Erste Schritte - Motion

Abb.5-5: Gerät einfügen

Anschließend startet, wie nachstehend beschrieben, automatisch ein Assistent zur Vergabe notwendiger Basiseinstellungen.

5.3.5 Basiseinstellungen zum Gerät vorgeben

Geräteeinstellungen

Geben Sie die allgemeinen Daten zum eingefügten Gerät an.

Abb.5-6: Geräteeinstellungen

- Gerätename** Geben Sie bei "Gerätenamen" einen beliebigen Gerätenamen ein, z.B. "DCC_Control". Das Feld enthält voreingestellt den Namen aus der Bibliothek. Das Gerät erscheint später unter dem angegebenen Namen im Projektexplorer.
- Kommentar** Fügen Sie zur ausführlichen Beschreibung des Gerätes einen beliebigen Kommentar ein. Dieses Feld ist optional.
- Ersteller** Geben Sie den Namen des Projekterstellers an. Dieses Feld ist optional.



Diese Angaben lassen sich auch später noch ändern.

Bestätigen Sie die Eingaben mit der Schaltfläche "Weiter>>".

Kommunikationseinstellungen

Definieren Sie im Dialog "Konfiguration" die Einstellungen zur Firmware und der Kommunikation zwischen IndraWorks und Ihrem Gerät (hier: IndraLogic XLC L65).

Abb.5-7: Konfiguration

- | | |
|-------------------------|---|
| Gerätetyp | Der Gerätetyp ist durch die Steuerungsauswahl vorbestimmt. Gibt es für einen Gerätetyp mehrere unterschiedliche Bestückungen, ist eine manuelle Ergänzung notwendig (Beispiel: IndraLogic XLC Lx5 gibt es in der Ausführung mit SERCOS III-Bus (CMLx5.1-3P) und ohne SERCOS III-Bus (CMLx5.1-NP)). |
| Firmware-Version | Hier kann die Version der Firmware ausgewählt werden. In der Regel kann diese Einstellung so belassen werden. |
| Firmware-Release | Hier wird das Release der Firmware innerhalb einer Firmwareversion eingestellt. Auch diese Einstellung sollte so belassen werden. |
| IP-Adresse | Geben Sie unter "Gerät IP-Adresse" Ihre Zieladresse ein, d.h. die IP-Adresse der Steuerung, auf die Sie Ihre Applikation laden möchten. |
| SPS-Gateway | Definieren Sie unter "Gateway IP-Adresse" die Adresse des Rechners, auf dem der Gateway-Server läuft. Voreingestellt erscheint hier "localhost" als Rechnername (Adresse). Dies bedeutet, dass der lokal installierte Gateway angesprochen wird. Diese Einstellung sollte nicht verändert werden. |
| Verbindungstest | <p>Klicken Sie auf die Schaltfläche "Verbindungstest", um zu überprüfen, ob die Kommunikation zwischen Rechner und der Zielsteuerung erfolgreich ist. Das Ergebnis des Tests wird im Feld "Ergebnis des Verbindungstests" protokolliert.</p> <p>Bei erfolgreichem SPS-Kommunikationstest wird die IP-Adresse angezeigt.</p> <p>Eventuell auftretende Probleme werden im Fenster "Ergebnis des Verbindungstests" durch blinkende rote oder gelbe Icons gekennzeichnet.</p> <p>Die folgenden Probleme können auftreten:</p> |

Erste Schritte - Motion

- Ausgabe: "Kommunikation zur Steuerung nicht erfolgreich."
Ursachen:
 - Die Steuerung ist nicht eingeschaltet.
 - Das Netzkabel ist nicht angeschlossen.
 - Die IP-Adresse der Steuerung stimmt nicht mit der eingestellten IP-Adresse überein.
Lösung: IP-Adresse im Dialog ändern oder IP-Adresse der Steuerung ändern (s. Rexroth IndraMotion MLC Funktionsbeschreibung, Inbetriebnahme der Steuerung).
 - Die Netzwerkmaske des PCs und/oder der Steuerung ist zu klein eingestellt.
Lösung: Netzwerkeinstellungen überprüfen und korrigieren (s. Rexroth IndraMotion MLC Funktionsbeschreibung, Inbetriebnahme der Steuerung).
- Ausgabe: "Die projektierte Firmware stimmt nicht mit der Firmware in der Steuerung überein."
Ursachen:
 - Die Firmware der Steuerung ist nicht aktuell.
Lösung: Firmwareupdate durchführen (s. Rexroth IndraMotion MLC Funktionsbeschreibung, Inbetriebnahme der Steuerung).
 - Die ausgewählte Steuerung entspricht nicht dem realen Steuerungstyp.

Sicherer Online-Modus

Ist diese Option aktiv, erfolgt vor sicherheitsrelevanten Aktivitäten eine Rückfrage mit Bitte um Bestätigung.

SPS-Programmiersprache

Hier wird die bevorzugte SPS-Programmiersprache eingestellt. Wählen Sie hier ST (strukturierter Text) aus.

Klicken Sie auf die Schaltfläche "Weiter >>" um zur nächsten Seite zu gelangen.

Funktionsmoduleinstellungen

Der Dialog "Geräte- und Funktionsmoduleinstellungen" kann sich, je nach Gerätetyp, unterscheiden. In diesem Projekt wird exemplarisch der Dialog des Gerätes "IndraLogic XLC L65" beschrieben.

Geben Sie die gewünschten Geräte- und Funktionsmoduleinstellungen ein.

Erste Schritte - Motion

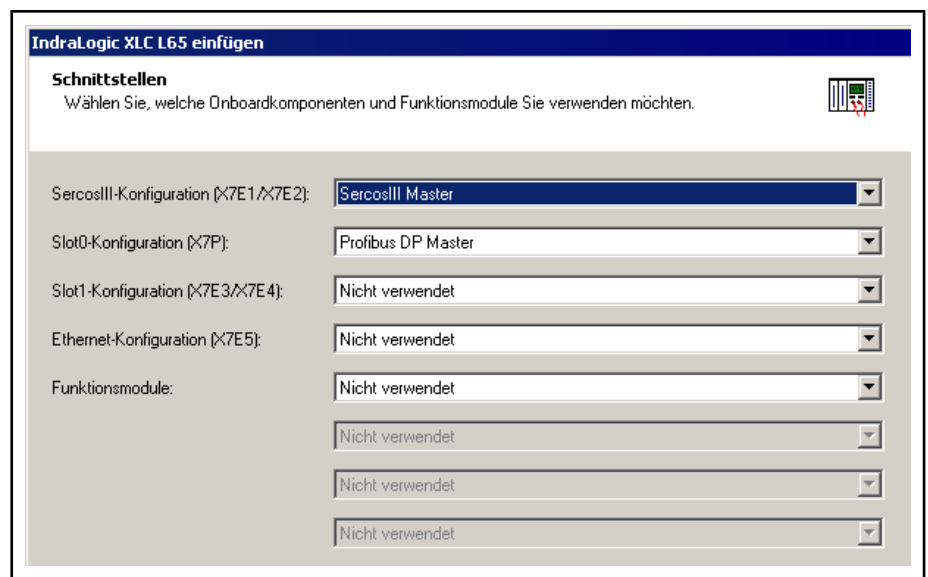


Abb.5-8: Funktionsmoduleinstellungen

- | | |
|---------------------------------|--|
| Sercos III-Konfiguration | Aktuelle SERCOS III-Konfiguration, die beim Einfügen des Gerätes gewählt werden kann. Zur Auswahl stehen "Sercos III-Master" oder "Nicht verwendet". |
| Slot0-Konfiguration | Aktuelle Profibuskonfiguration, die beim Einfügen des Gerätes gewählt werden kann. Zur Auswahl stehen "PROFIBUS DP-Master", "PROFIBUS DP-Slave" oder "Nicht verwendet". |
| Slot1-Konfiguration | Aktuelle PROFINET- oder EtherNet/IP-Konfiguration, die beim Einfügen des Gerätes gewählt werden kann. Zur Auswahl stehen "Nicht verwendet", bzw. alternativ eine PROFINET- oder EtherNet/IP-Schnittstelle. |
| EtherNet-Konfiguration | Da Sie diese Schnittstelle als Engineering-Schnittstelle benötigen (Verbindung zu Ihrem PC direkt oder über das Betriebsnetz) darf der Ethernet/IP adapter nicht aktiviert werden. |
| Funktionsmodul 1-4 | An der "IndraLogic XLC L65" können bis zu vier Funktionsmodule angekoppelt werden.
Bei Bedarf können Sie aus der jeweiligen Liste auswählen.
Bestätigen Sie die Eingaben mit der Schaltfläche "Fertigstellen".
Im Projektextplorer erscheint das neu angelegte Gerät. |

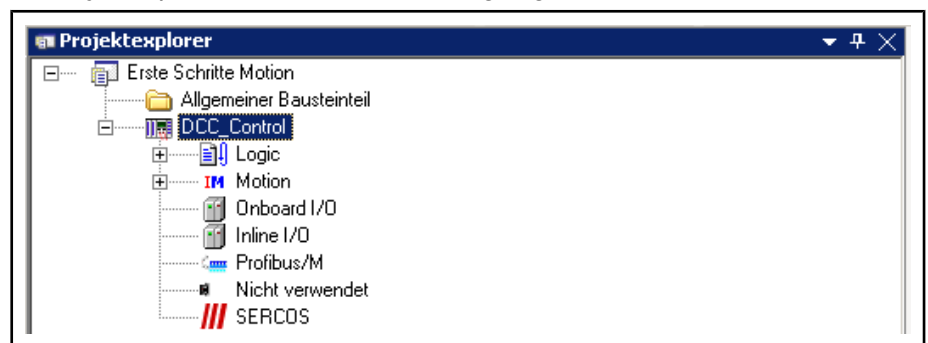


Abb.5-9: Projektextplorer: Projekt "Erste Schritte - Motion" mit einer IndraLogic XLC L65-Steuerung mit SERCOS III-Bus



Das Bild im Projektnavigator widerspiegelt die in den Dialogen "Konfiguration" und "Funktionsmodule" gewählten Komponenten.

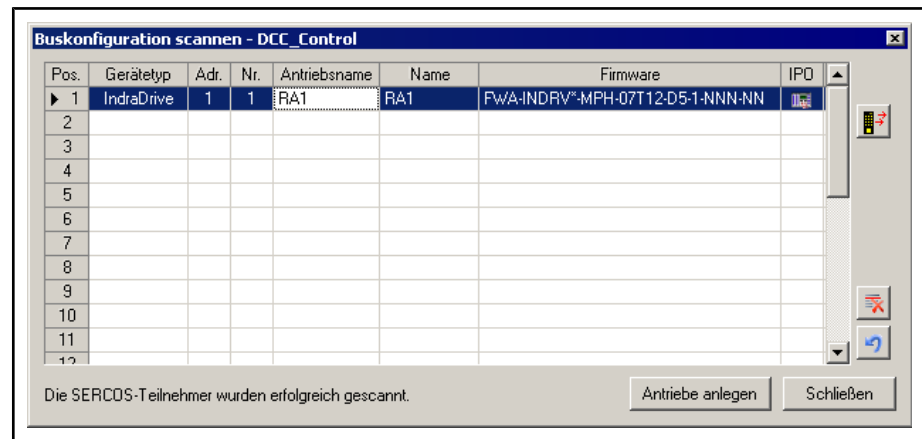
Erste Schritte - Motion

5.3.6 SERCOS III-Antriebe (reale Achsen) zum Projekt hinzufügen

Die einfachste Art, die Daten aller angeschlossenen Antriebe zu ermitteln, ist das Scannen des SERCOS III-Busses (Kontext-Menü "SERCOS"-Knoten **Buskonfiguration scannen**).

Zum Scannen schaltet IndraWorks die Steuerung nach Rückfrage in den Parametriermodus (P2).

Das Ergebnis wird im Folgebild angezeigt:



Pos	Topologieadresse (Reihenfolge im Ring)
Gerätetyp	Antriebsfamilie
Adr	SERCOS-Adresse
Nr	Achsnummer
Antriebsname	default Antrieb _n , umbenannt
Name	Name der Achse / des IO-Gerätes
Firmware	Firmwareidentifikation
IPO	Interpolation in Steuerung oder im Antrieb, hier in der Steuerung
Abb.5-10:	Liste der SERCOS-Antriebe, Antriebsname auf RA1 geändert



Interpolation in Steuerung oder im Antrieb :
Umschaltung durch Doppelklick auf das Icon.

Betätigen Sie "Antriebe anlegen".

Die gescannte Buskonfiguration wird in den Projektexplorer übernommen. Der Antrieb wird als "Reale Achse" im Motion-Knoten gespiegelt

Die Achsnummer "1" kann alternativ zum Achsnamen "RA1" verwendet werden.

Das Projekt enthält das Gerät "DCC_Control" z.B. vom Typ "IndraLogic XLC L65" und die reale Achse "RA1" (Ordner **Motion** ► **Reale Achsen**) bzw. Antrieb "RA1" (Ordner **SERCOS**):

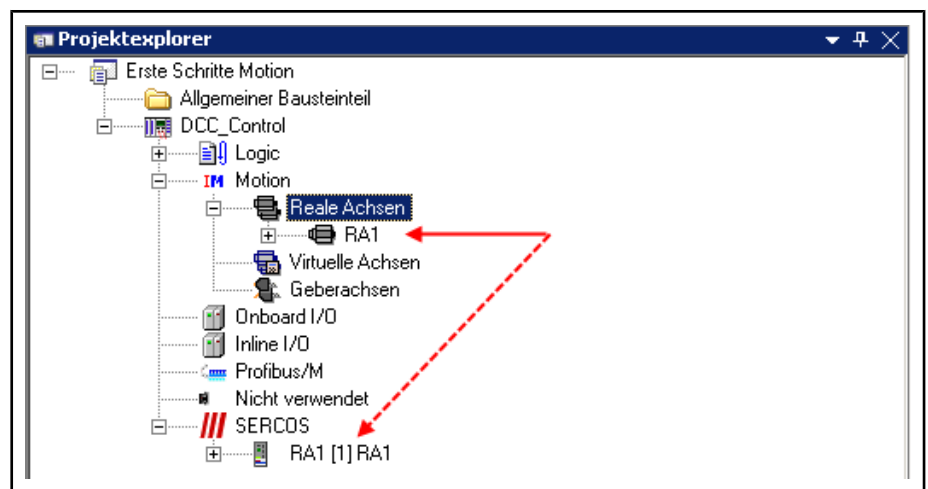


Abb.5-11: Antrieb / Reale Achse "RA1" ist dem Projekt hinzugefügt

5.3.7 Virtuelle Achse zum Projekt hinzufügen

Im Unterschied zu realen Achsen, die an unterschiedlichen Interfaces, z.B. SERCOS oder Profibus laufen können und entsprechend vereinbart oder gesamt werden müssen, ist eine virtuelle Achse nur zu deklarieren. Im Projektnavigator ist auf dem Ordner **Motion** ► **Virtuelle Achse** mit der Rechten-Maus-Taste der Kontextmenü-Punkt **Hinzufügen** zu betätigen.

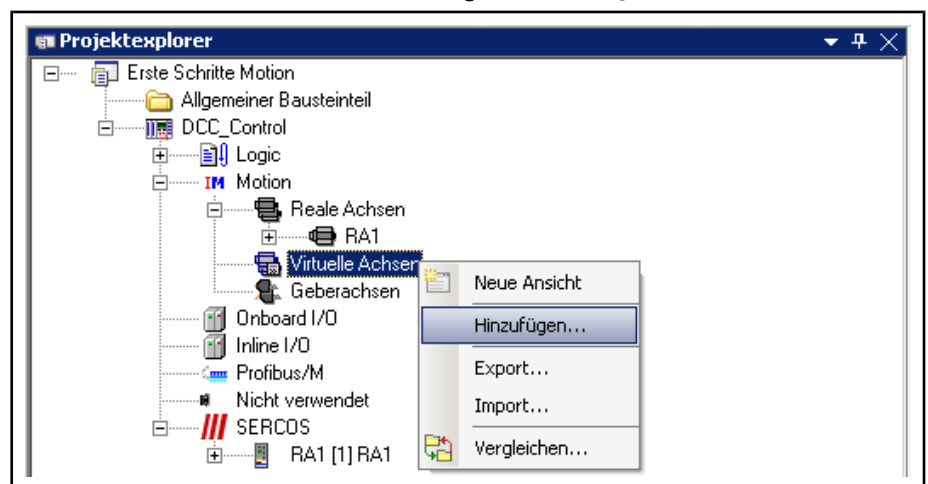


Abb.5-12: Vereinbaren einer virtuellen Achse

Analog zum Eintrag der Steuerung liefert ein Wizard die erforderlichen Dialogfenster zum Ausfüllen.

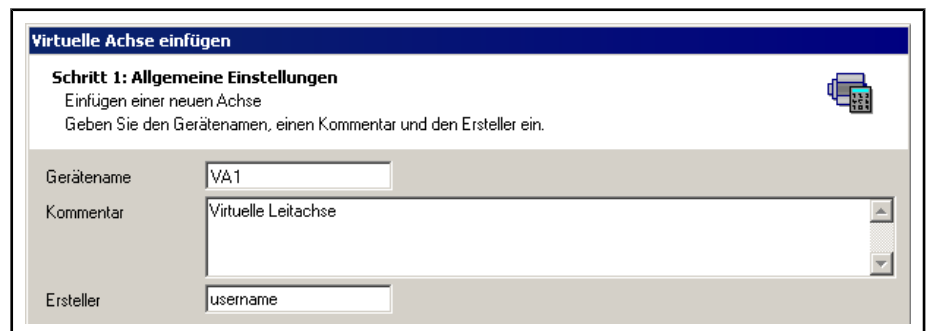


Abb.5-13: Dialog "Allgemeine Einstellungen" für eine virtuelle Achse

Erste Schritte - Motion

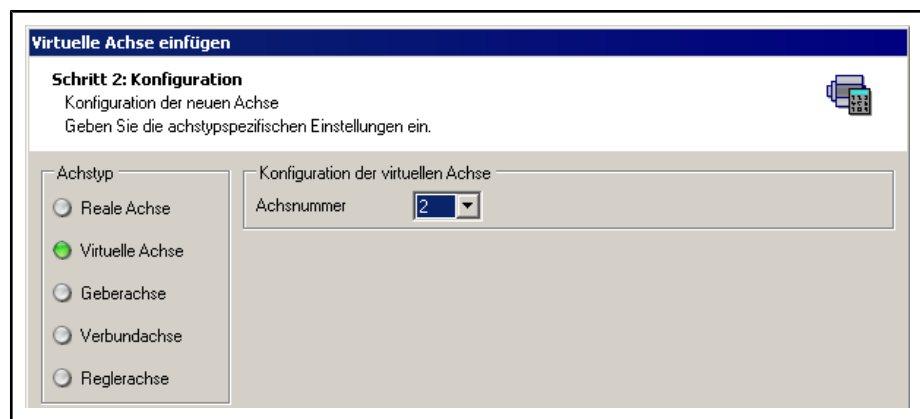


Abb.5-14: Dialog "Konfiguration" für eine virtuelle Achse

Die Achsnummer "2" kann alternativ zum Achsnamen "VA1" verwendet werden.

Klicken Sie auf "Fertigstellen".

5.3.8 Download des Motion-Teilprojektes

Mit dem Anlegen der Steuerung und aller Achsen im Projekt "Erste Schritte - Motion" sowie der Parametrierung der Steuerungs- und Achsbestandteile ist der erste Abschnitt der Bearbeitung abgeschlossen.

Die bisher gesammelten Informationen sind jetzt in die Steuerung zu laden. Dazu sind das gewünschte Gerät (DCC_Control) zu markieren und das Icon der Projekt-Symbolleiste oder der Menüpunkt **Projekt ► Gerät Online schalten** zu betätigen:

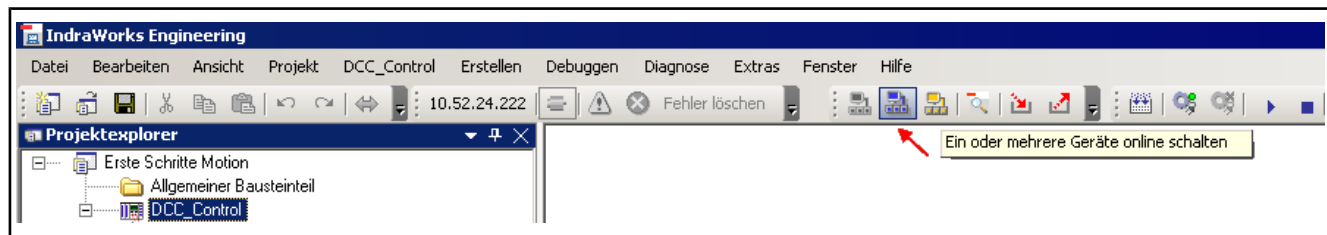


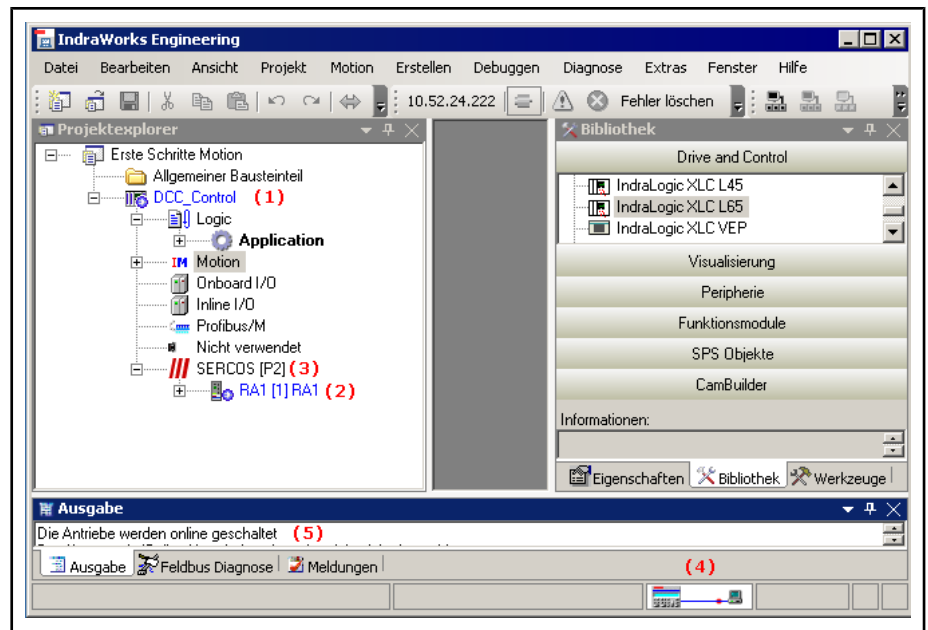
Abb.5-15: Motion-Daten in die Steuerung laden

Der Anwender wird auf Differenzen zwischen dem Projekt auf der Steuerung und seinem Projekt hingewiesen. Nach Zustimmung erfolgt das Überschreiben. Die Anzeige der Steuerung ändert sich von **BB STOP** über mehrere Zwischenschritte nach **P2 STOP**.



Ohne diesen Download-Schritt fehlen der Steuerung die im Hintergrund erarbeiteten Daten.

Erste Schritte - Motion



- (1) aktive Steuerung (online)
- (2) aktiver Antrieb (online)
- (3) Kommunikation auf dem Niveau "P2", Parametriermodus
- (4) Online-Information
- (5) Statuszeile

Abb.5-16: Erfolgreicher Abschluss des Downloads (Statuszeile)



Die "Online"-Information (4) in der Fußzeile wird nur sichtbar, wenn die gewünschte Steuerung oder einer ihrer Unterordner aktiv ist.



Vor dem Wechsel nach BB (betriebsbereit) muss die in P2 notwendige Parametrierung der Antriebe (Achsen) mit plausiblen Werten abgeschlossen sein, ansonsten kommt es zu Phasenumschaltfehlern, die ein Erreichen der Phase BB nicht zulassen und ein erneutes Phasenschalten nach P2 notwendig machen (Plausibilitätsprüfung beim Phasenschalten durch Antriebe und Steuerung).

Dies gilt generell, wenn versucht wird, die Steuerung nach BB bzw. Antriebe nach Ab (Antrieb bereit) zu schalten.



Wurde die Betriebsart des Antriebs umgeschaltet IndraDrive (Berechnung im Antrieb) auf IndraDrive (Berechnung in der Steuerung) oder umgekehrt, kann es beim Wechsel von P2 nach BB zu einer Fehlermeldung kommen. In diesem Fall müssen die Basisparameter des Antriebs nachgeladen werden. Kontextmenü RA1**Parameter** ▶ **Basisparameter laden...**

Abschließend wird die Steuerung aus dem "Parametriermodus (P2)" in den "Betriebsmodus (BB)" geschaltet (Kontextmenü SERCOS-Knoten **SERCOS-Zustand** ▶ **Phase P4**).

Erste Schritte - Motion



In Abhängigkeit von der Art des Motorgebers kann ein Fehler "F2174, Referenzverlust Motorgeber" auftreten.

Der Fehler kann entweder über <ESC> an den Displays des Antriebs und der Steuerung quittiert oder besser über **DCC_Control ► Motion ► Reale Achse ► RA1 ► Maßbezug herstellen ► Maßbezug Motorgeber** "Absolutmaß setzen" beseitigt werden.

Der Antrieb ist nun im Status "Ab", Antrieb bereit.

Die Steuerung, Motion-Komponente in "BB", betriebsbereit, SPS-Komponente in "STOP"

5.4 Programmieren des Bewegungsablaufes in Funktionsbausteinsprache (FUP)

5.4.1 Programmieren des Bewegungsablaufes in FUP, Allgemeines

Der Bewegungsablauf wird in einem SPS-Programm unter Nutzung von PLCopen-Funktionsbausteinen programmiert (Programmiersystem IndraLogic 2G nach EN 61131-3).

Beim Anlegen des Projektes mit der Steuerung DCC_Control, der realen Achse RA1 und der virtuellen Achse VA1 wurden unterhalb des Ordners "Logic" erste Aktivitäten zur Programmierung des Bewegungsablaufes begonnen:

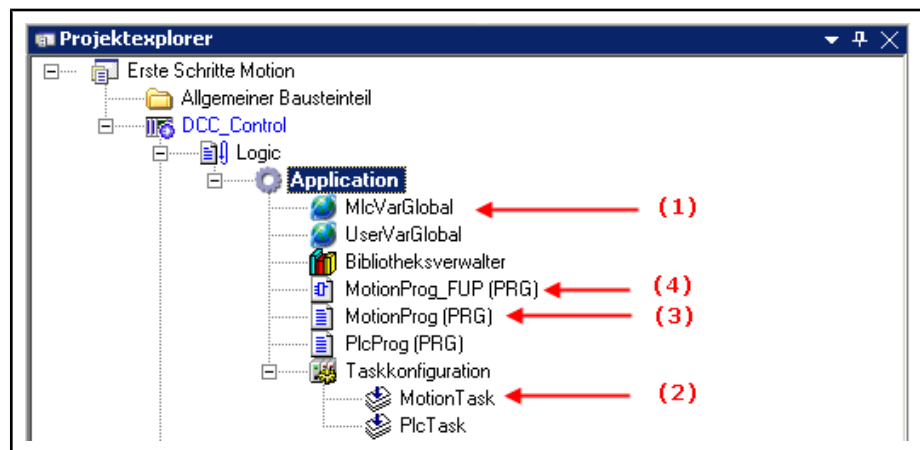


Abb.5-17: Projektexplorer nach Expandieren der Knoten "Logic"

Angelegt wurden die Programme:

- **MotionProg (3)** (Vorwahl ST, Strukturierter Text),
- **PlcProg** (Vorwahl ST, Strukturierter Text), nicht verwendet.

Das dritte Programm **MotionProg_FUP (4)**, wurde manuell ergänzt: **Application**, rechte Maustaste, **Hinzufügen ► POU**, Eingabe des Namens und Auswahl von FUP als Programmiersprache.

Die Programme sind noch leer.

Angelegt wurden unter **Taskkonfiguration** die **Tasks**:

- **MotionTask (2)**, (Priorität 2, extern ereignisgesteuert durch SERCOS-Task, Watchdog aktiv) für "MotionProg".

Das Programm/ die Programme, die unter dieser Task laufen, müssen komplett innerhalb des fest vorgegebenen Teil eines SERCOS-Zyklus

Erste Schritte - Motion

abgearbeitet werden. Überschreiten sie diesen Rahmen, wird die Steuerung gestoppt.

- **PlcTask**, (niederpriore zyklische Task) für PlcProg, nicht verwendet.

Das Programm wird in der verbleibenden Zeit zumindest stückweise, also im Bedarfsfall über mehrere Zyklen, abgearbeitet.

Angelegt wurden Globale Variablen / Globale Konstanten:

- **UserVarGlobal**, reserviert für den Anwender, in "Erste Schritte - Motion" nicht genutzt,
- **MlcVarGlobal (1)**, enthält die Adresskonstanten für die Achsen RA1 und VA1.

Programm:

```
(* Automated generated code by MLC. *)
(* Please don't edit, the code will be overwritten...*)
VAR_GLOBAL CONSTANT
(*virtual axis "VA1" with axis number 2*)
VA1: AXIS_REF :=(CntlNo:=LOCAL_CNTRL,AxisNo:=AXIS_2);
(*real axis "RA1" with axis number 1*)
RA1: AXIS_REF :=(CntlNo:=LOCAL_CNTRL,AxisNo:=AXIS_1);
(*...*)
END_VAR
```



Wir nutzen für die "Ersten Schritte Motion" die MotionTask.

Je nach gewünschter Programmiersprache ist ihr entweder **MotionProg** oder **MotionProg_FUP** zugeordnet.

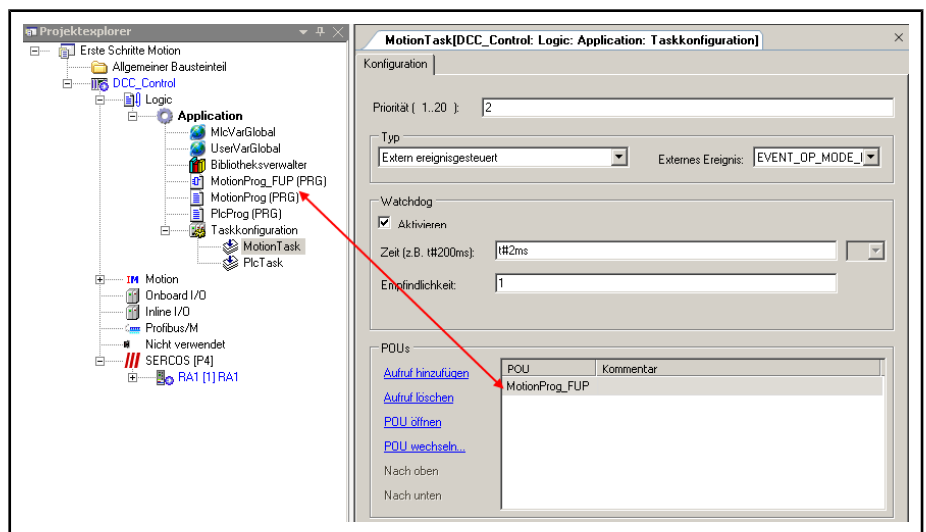


Abb.5-18: Aufruf löschen: MotionProg; Aufruf hinzufügen: MotionProg_FUP

5.4.2 Programmieren der virtuellen Achse VA1, Programmiersprache FUP



Wenn Sie Ihre Programmierung in FUP (Funktionsbausteinsprache) grafisch anschaulich ausführen wollen, setzen Sie in diesem Abschnitt fort.

Wenn Sie sich für ST (Strukturierten Text) entscheiden, gehen Sie zum Abschnitt [Programmieren der virtuellen Achse VA1, Programmiersprache ST, Seite 123](#).

Entsprechend der [Ziel des Projekts Erste Schritte - Motion, Seite 87](#), werden Instanzen der PLCopen-Bausteine MC_MoveVelocity und MC_Stop unter Nutzung von "VA1" aufgerufen.

Erste Schritte - Motion



Eine Instanz des MC_Power wird für virtuelle Achsen nicht benötigt.

Gehen Sie dazu folgendermaßen vor:

1. Ordnen Sie der "MotionTask" das Programm "MotionProg_FUP" zu.
2. Öffnen Sie das Programm "MotionProg_FUP" (Doppelklick Linke-Maus-Taste).

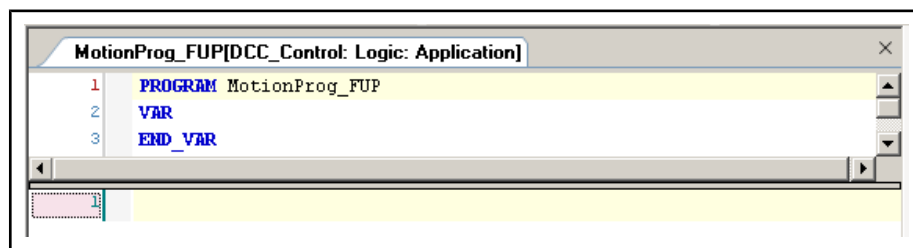


Abb.5-19: Neues Programm in FUP (Funktionsbausteinsprache)

Fügen Sie in der Implementation, Netzwerk 1, per rechter Maustaste den ersten Funktionsbaustein ein:

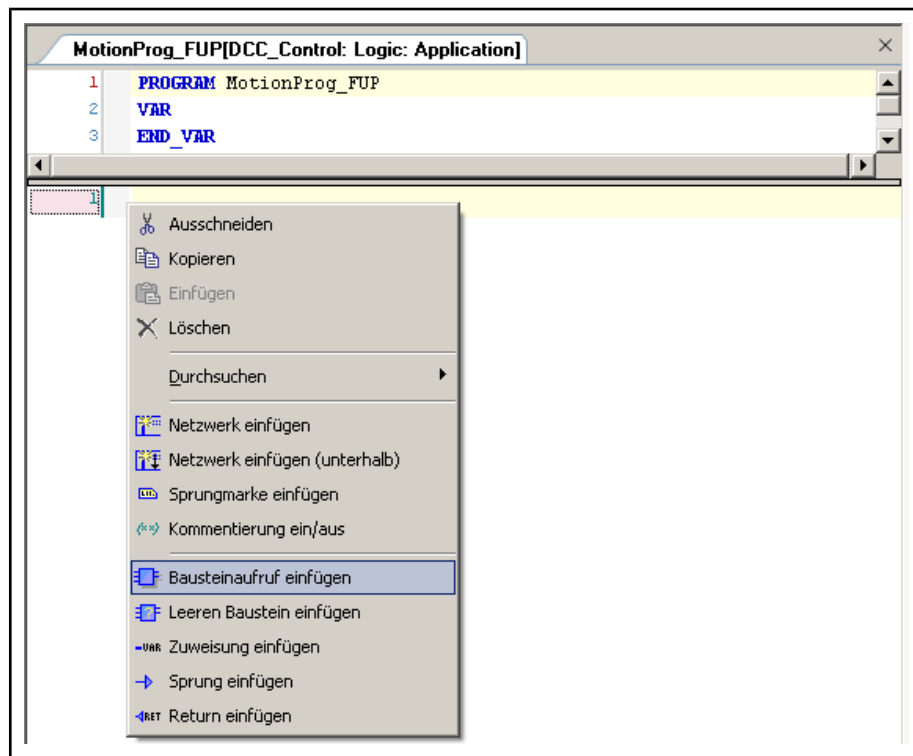


Abb.5-20: Generieren eines neuen Bausteins

Wählen Sie dazu im geöffneten Auswahlfenster

Funktionsbausteine ► ML_PLCopen ► POU ► Motion ► MC_MoveVelocity aus.

Erste Schritte - Motion

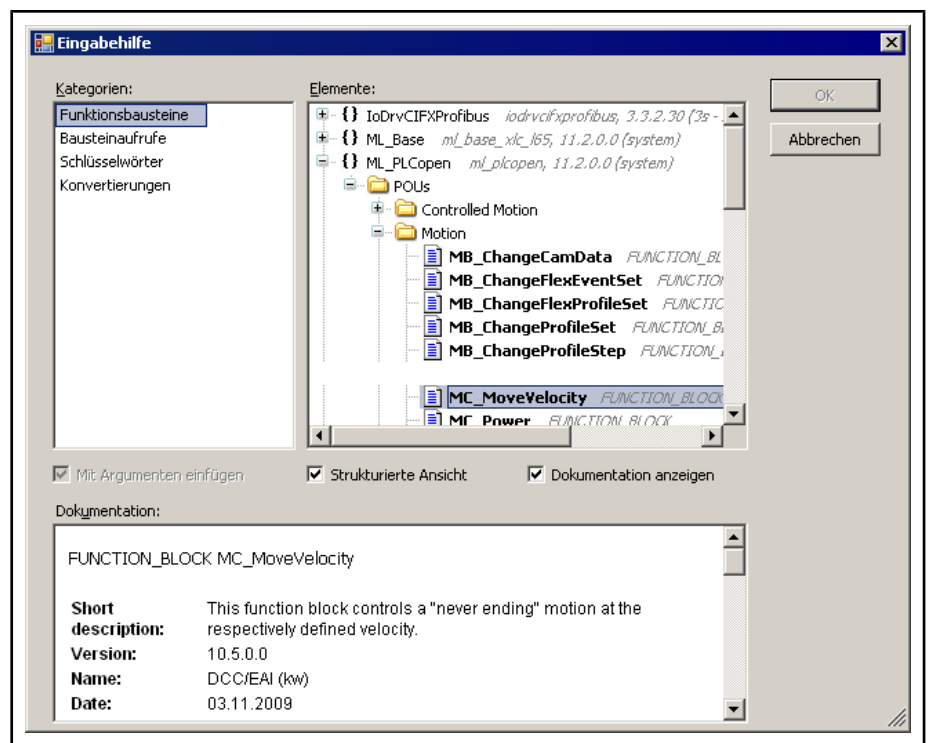


Abb.5-21: Auswahlfenster für Funktionsbausteintypen

Bestätigen Sie jetzt das Einfügen des Bausteines von MC_MoveVelocity mit "OK", und markieren Sie im Programm "MotionProg_FUP" durch Klick mit der linken Maustaste die drei Fragezeichen ??? oberhalb des Bausteines. Geben Sie als Instanznamen für diesen Baustein "MoveVelVA1" ein, und schließen Sie die Eingabe mit <Enter> ab. Daraufhin öffnet sich das Fenster der "Variablendeklaration", welches Sie sofort mit "OK" quittieren können:

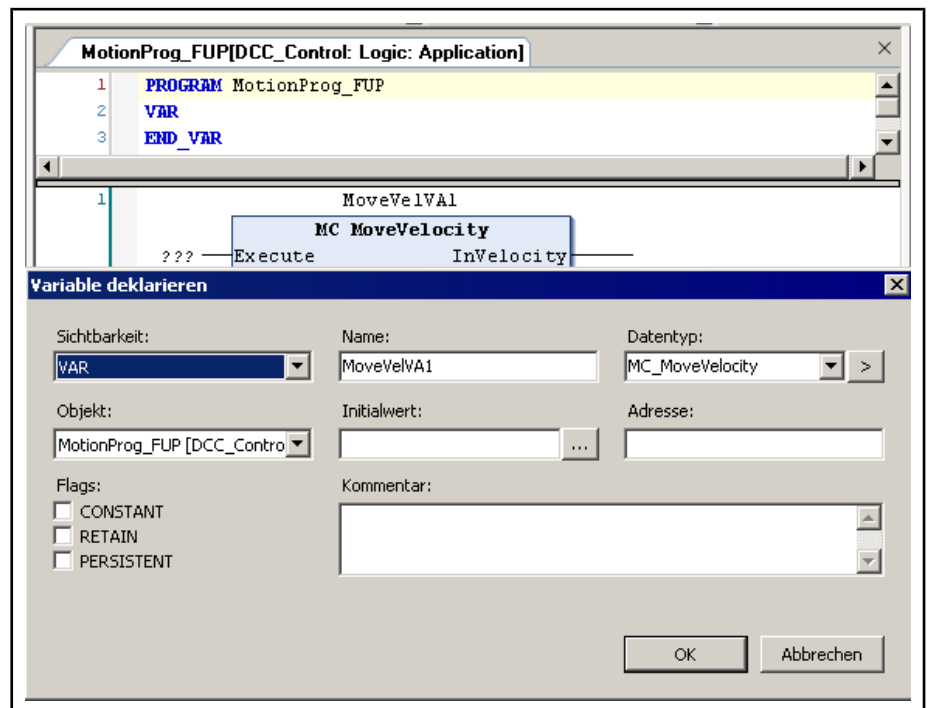


Abb.5-22: Deklaration des Instanznamens eines Bausteines, Typ "MC_MoveVelocity" mit Hilfe des Fensters "Variablendeklaration"

Erste Schritte - Motion

Damit haben Sie die Instanz MoveVelVA1 des PLC-Open-Bausteines MC_MoveVelocity deklariert. Beschalten Sie nun die Eingänge des Bausteines.

Analog zum Instanznamen des Bausteines beschalten Sie den Eingang "Execute" mit einer Variable mit Namen "ex_MoveVelVA1", Typ BOOL:

- Markieren Sie die 3 Fragezeichen vor dem Eingang "Execute"
- Geben Sie den Namen "ex_MoveVelVA1" ein, und schließen Sie mit <Enter> ab
- Das sich öffnende Fenster zur "Variablendeklaration" können Sie sofort mit <Enter> bestätigen

Um den Eingang "Velocity" zu beschalten, gehen Sie wie folgt vor:

- Markieren Sie die 3 Fragezeichen vor dem Eingang "Velocity"
- Geben Sie den Namen "VcomVA1" ein, <Enter>
- Als Typ tragen Sie REAL und als Initialwert 200 (U/min) ein, <Enter>

Für die Eingänge "Acceleration" tragen Sie bitte 20 (rad/s²) und für "Deceleration" 20 (rad/s²) ein.



Die Werte werden direkt an den Bausteineingängen eingetragen.

Alternativ kann eine Variable mit dem entsprechenden Wert verwendet werden.

Um den Eingang "Axis" zu beschalten, gehen Sie wie folgt vor:

- Klicken Sie die **???** an und wählen die nebenstehende Eingabehilfe

Wählen Sie **Variablen** ► **Application** ► **MlcVarGlobal** ► **VA1**.

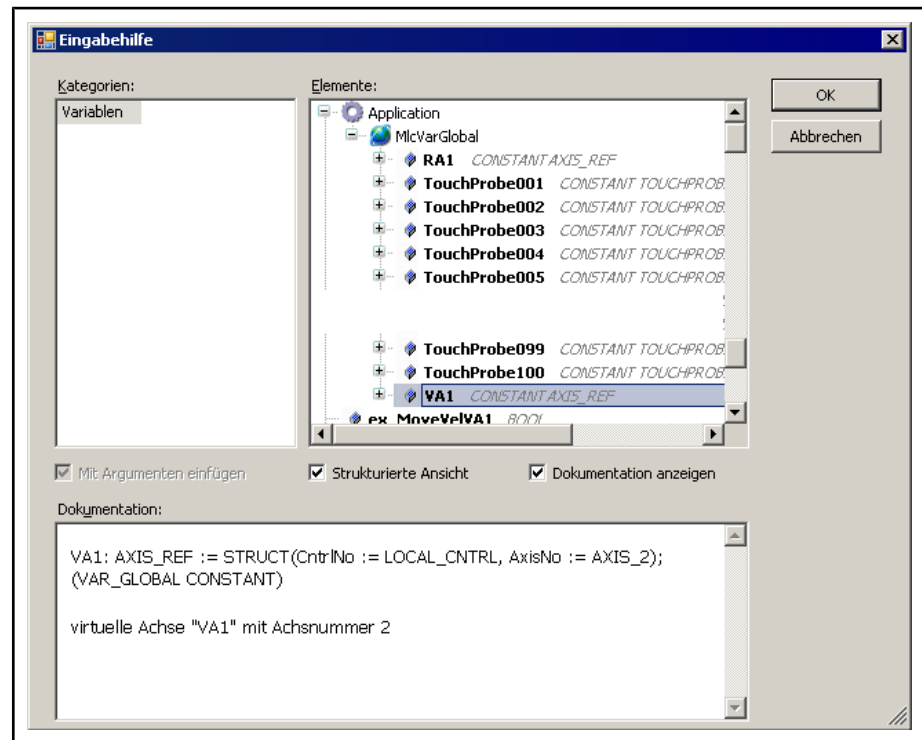


Abb.5-23: Auswahl der Achsinstanz bei Beschaltung eines Funktionsbausteines vom Typ "MC_MoveVelocity"

- Bestätigen Sie mit "OK".

Erste Schritte - Motion

- Schließen Sie die Eingabe des Namens mit <Enter> ab.



Der Name einer globalen Konstanten setzt sich aus dem Namen des Ordners und des Elementes zusammen:

MlcVarGlobal.VA1

Um eine Rückmeldung zu erhalten, ob der Funktionsbaustein korrekt gearbeitet hat, werden seine Ausgänge beschaltet.

Markieren Sie dazu durch Mausklick den Bereich rechts neben dem Ausgang "InVelocity" und wählen Sie im Kontextmenü (Klick rechte Maustaste)

Zuweisung einfügen aus:

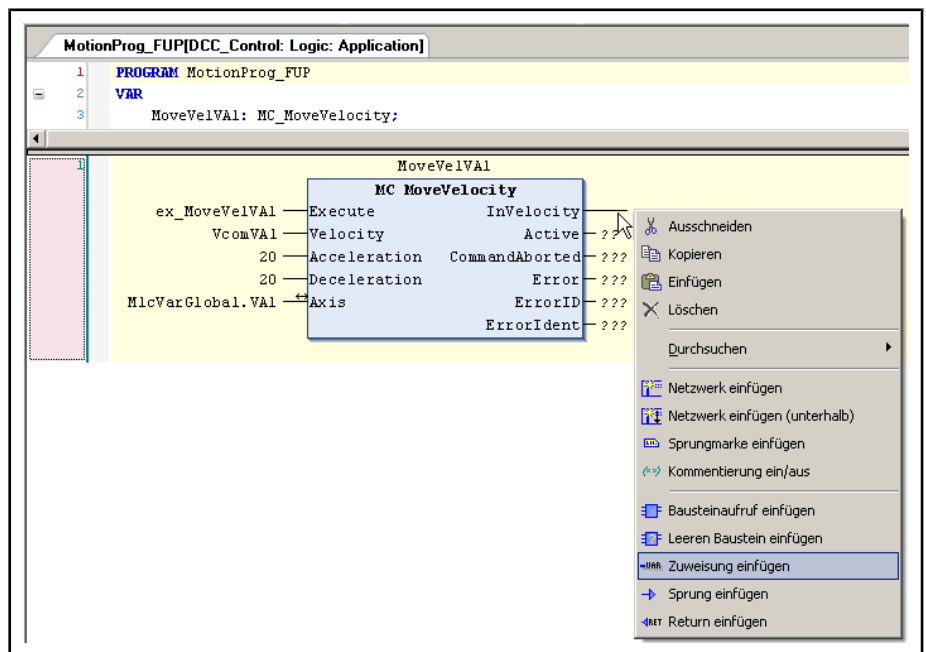


Abb.5-24: Zuweisung am Ausgang eines Funktionsbausteines

In bekannter Art und Weise beschalten Sie den Ausgang "InVelocity" des Bausteines mit der Variablen "in_MoveVelVA1", Typ BOOL:

- Markieren Sie die 3 Fragezeichen ??? am Ausgang "InVelocity".
- Geben Sie den Namen "in_MoveVelVA1" ein, <Enter>.
- Das sich öffnende Fenster zur "Variablendeklaration" können Sie sofort mit <Enter> bestätigen.
- Beschalten Sie den Active-Ausgang mit "ac_MoveVelVA1".
- Beschalten Sie den CommandAborted-Ausgang mit "ca_MoveVelVA1".
- Beschalten Sie den Error-Ausgang mit "er_MoveVelVA1".
- Beschalten Sie den ErrorID-Ausgang mit "ID_MoveVelVA1".
- Beschalten Sie den Error-Ausgang mit "Ident_MoveVelVA1".

Damit ist der PLCopen-Baustein vom Typ MC_MoveVelocity entsprechend der Aufgabenstellung beschaltet. Der Deklarationsteil wurde automatisch mit komplettiert.

Erste Schritte - Motion

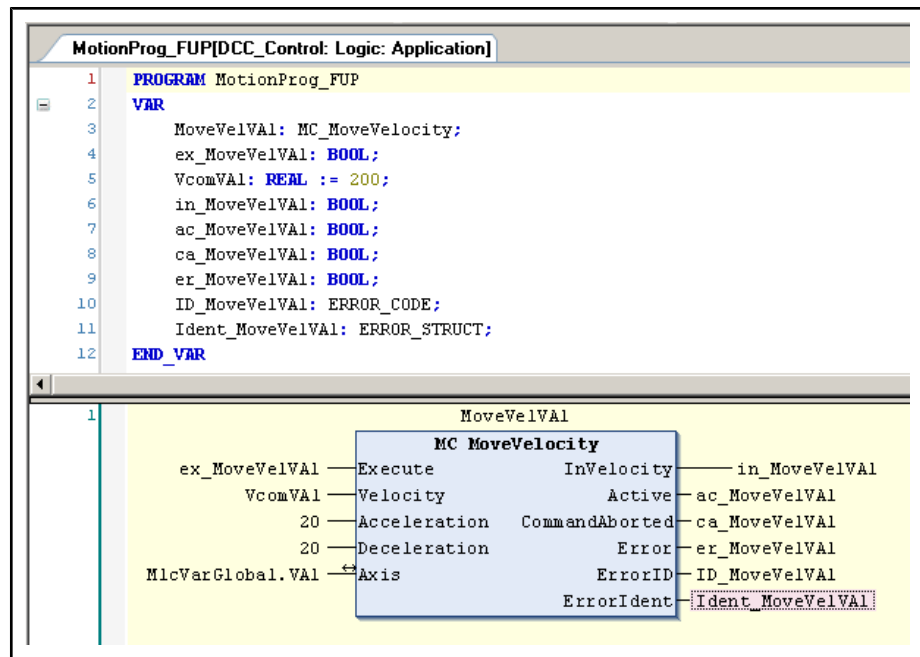


Abb.5-25: PLCopen-FB Typ "MC_MoveVelocity" ist eingefügt und beschaltet



Immer wenn Sie einen Baustein abgeschlossen haben, lassen Sie eine Testübersetzung <Umsch>+<F11> laufen. Beseitigen Sie danach eventuell vorhandene Fehler sofort.

Generieren Sie nun eine Instanz des PLCopen-Bausteins MC_Stop.

MC_Stop MC_Stop einfügen und eine Instanz "StopVA1" deklarieren

Neues Netzwerk anfügen, Klick mit rechter Maustaste in Netzwerk 1,

Netzwerk einfügen (unterhalb)

MC_Stop einfügen und eine Instanz "StopVA1" deklarieren

Eingänge beschalten und wenn nötig Eingangsvariablen deklarieren:

"Execute" → "ex_StopVA1" vom Typ BOOL

"Deceleration" → 50

"Axis" → "MlcVarGlobal.VA1" vom Typ "AXIS_REF")

Ausgänge:

"Done" → "do_StopVA1" vom Typ BOOL

"Active" → "ac_StopVA1" vom Typ BOOL

"CommandAborted" → "ca_StopVA1" vom Typ BOOL

"Error" → "er_StopVA1" vom Typ BOOL

"ErrorID" → "ID_StopVA1" vom Typ ERROR_CODE

"ErrorIdent" → "Ident_StopVA1" vom Typ ERROR_STRUCT

Istgeschwindigkeit "VactVA1"

Als Letztes soll ein Netzwerk, in dem die aktuelle Geschwindigkeit der Achse VA1 "VactVA1" (Datentyp REAL) abgelesen werden kann, implementiert werden.

Dazu muss die aktuelle Geschwindigkeit aus dem Datensatz der Achse (Shared Memory) gelesen werden.

Erste Schritte - Motion

Die Daten aller Achsen, die eine MLC steuern kann, sind in dem Feld

GVL_Base.AxisData: ARRAY [1..xx¹⁾] OF ML_AXISDATA_SM

enthalten, wobei als Index die Achsnummer zu verwenden ist.

Die Variable wird unter dem Namen

GVL_Base.AxisData[2].rActualVelocity_i,

für die virtuelle Achse, Achsnummer 2, bereitgestellt (Achsnummer 1 hat RA1, da das Scannen von RA1 vor der Deklaration von VA1 erfolgte).

- Neues Netzwerk anfügen, Klick mit rechter Maustaste in Netzwerk 2, **Netzwerk einfügen (unterhalb),**
- **Zuweisung einfügen,**
- Auf den rechten **???** ist "VactVA1" einzufügen, <Enter>, Typ "REAL" im Deklarationsfenster auswählen <Enter>.
- Auf die linken **???** gehen und danach **...** anklicken und unter Nutzung der Eingabehilfe zu **Variablen** ► **ML_BASE** ► **Global Variables List** ► **GVL_Base** ► **AxisData[]** navigieren:

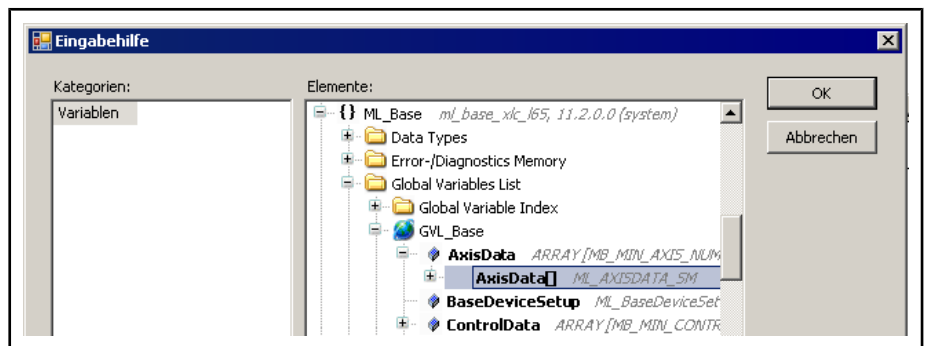


Abb.5-26: Geschwindigkeitswert VA1 aus der Globalen Variablenliste der Bibliothek ML_Base.library

Die weitere Eingabe wird durch die Intellisense-Funktion unterstützt. Dies bedeutet, dass nach der Eingabe des "." bei "GVL_Base.AxisData[2]." alle verfügbaren Elemente aufgerollt werden und zur Auswahl stehen.

"rActualVelocity_i" mit <Enter> bestätigen.

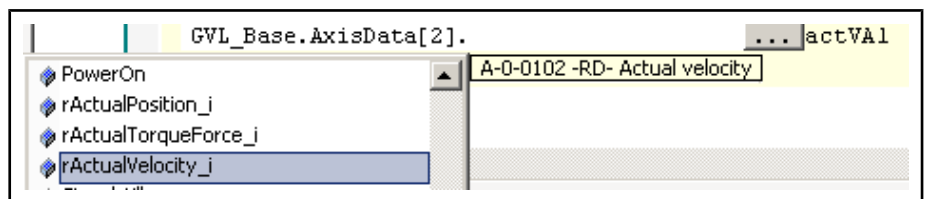


Abb.5-27: Auswahl der Variablen aktuelle Geschwindigkeit

Damit ist die Implementierung des Bewegungsablaufes entsprechend der Aufgabenstellung abgeschlossen:

¹⁾ Die Anzahl der steuerbaren Achsen ist vom Typ der Steuerung und ihrer Firmware abhängig, aktuell für L65: 64 Achsen

Erste Schritte - Motion

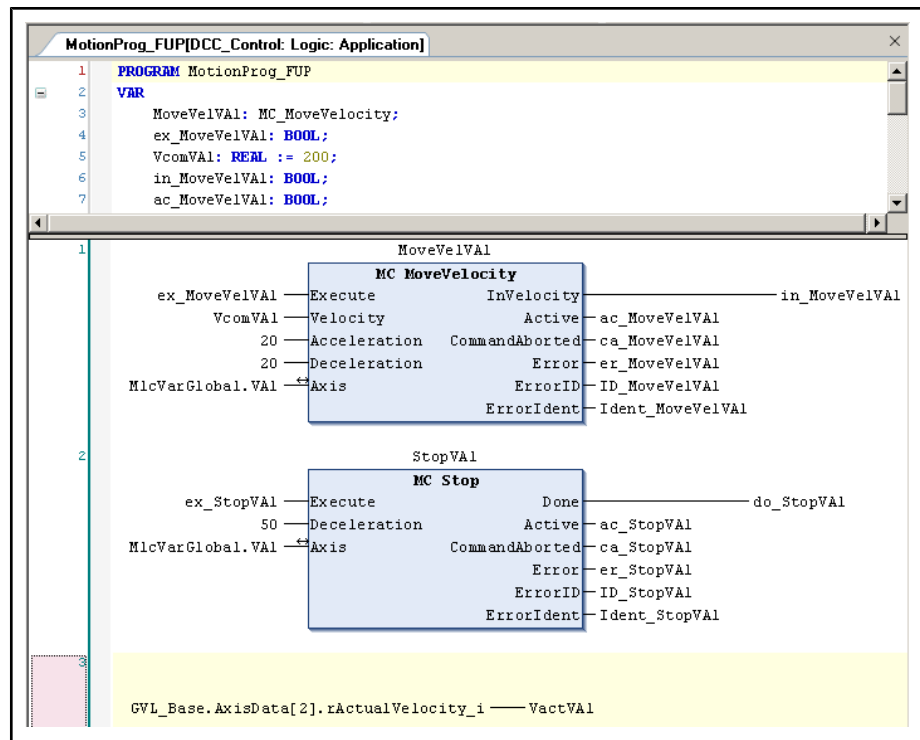


Abb.5-28: Implementation der virtuellen Achse VA1

5.4.3 Download des SPS-Teilprojekts virtuelle Achse VA1 (FUP)

Der zuvor implementierte Bewegungsablauf muss jetzt noch übersetzt, in die Steuerung übertragen und gestartet werden.

Die Übersetzung erreichen Sie im Menüpunkt

- **Erstellen ► Übersetzen**, Übersetzen des hinzu gekommenen Anteils der Applikation.
- **Erstellen ► Neu übersetzen**, Übersetzen der kompletten Applikation.

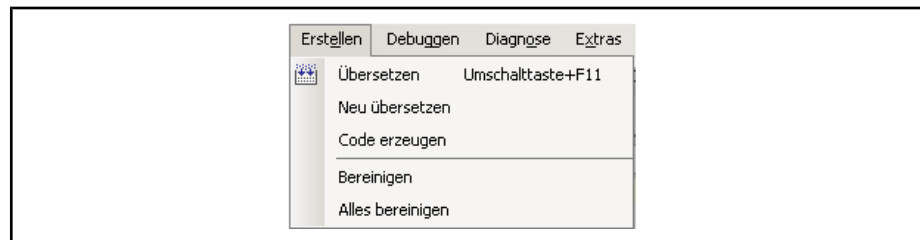


Abb.5-29: Applikation übersetzen

Zum Übertragen der kompilierten Daten in die Zielsteuerung wählen Sie den Menüpunkt **Debuggen ► Einloggen** (nachfolgendes Bild).

Erste Schritte - Motion



Abb.5-30: Übertragen der kompilierten Daten in die Steuerung

In Abhängigkeit ob eine Applikation bereits geladen ist und ob diese Applikation eine Vorgängerin Ihrer aktuellen Applikation ist, werden Sie um Zustimmung zum Download gebeten.

Nach erfolgreicher Übertragung ist die Applikation noch gestoppt (rot im Projektnavigators und in der Statuszeile), und die Variablen in Deklaration und Implementation sind zur Anzeige der Werte vorbereitet. Die Werte sind jedoch noch nicht gültig.

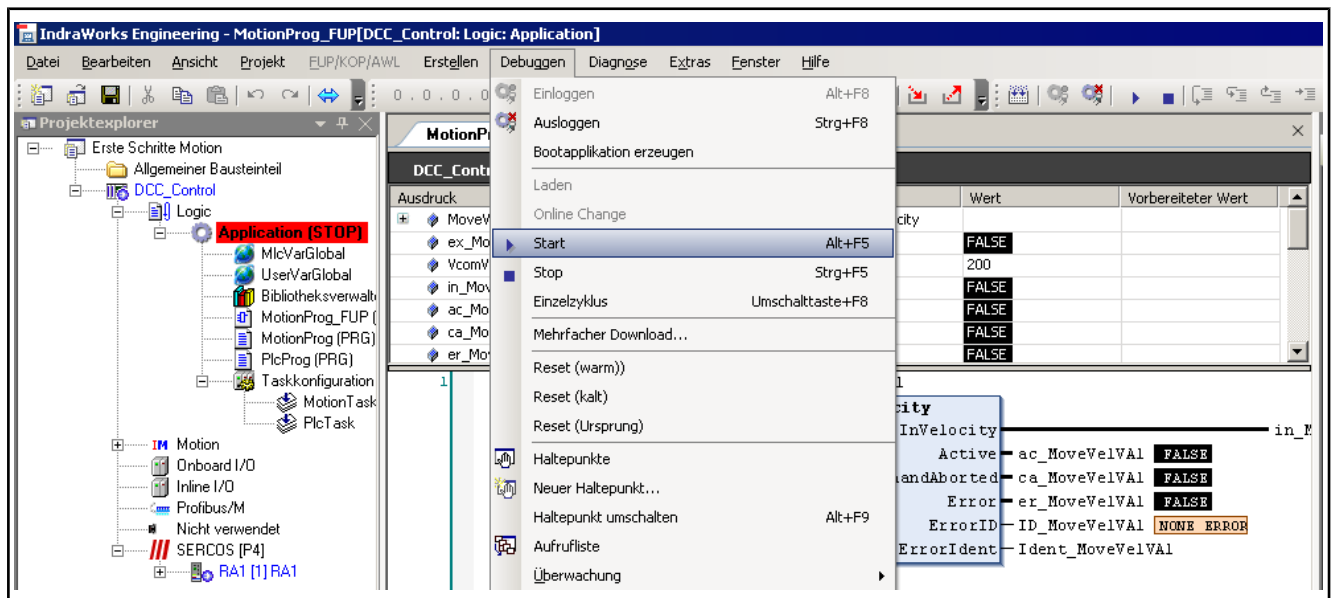


Abb.5-31: Starten der Programmbearbeitung in der Steuerung (1)

Starten Sie die Bearbeitung in der Steuerung über **Debuggen ▶ Start**.

Erste Schritte - Motion

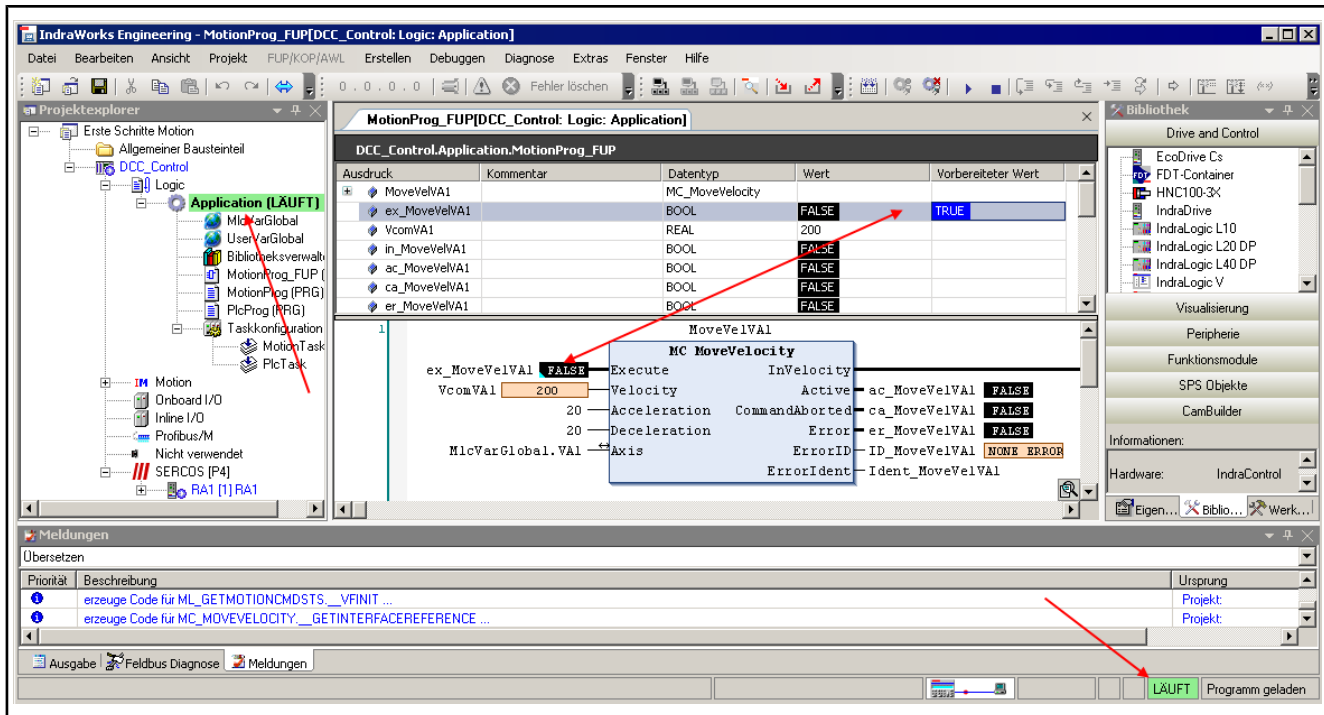


Abb.5-32: Starten der Programmbearbeitung in der Steuerung (2)

Der Bewegungsablauf selbst wird angestoßen, indem nacheinander die Abarbeitung der programmierten Funktionsbausteine aktiviert wird.

"Werte schreiben" der Eingangsvariablen

Um die Bearbeitung des "MoveVelVA1" vom Typ "MC_MoveVelocity" zu aktivieren, muss am "Execute"-Eingang des Bausteins eine positive Flanke auftreten.

Doppelklick mit der linken Maustaste auf das FALSE hinter der Eingangsvariablen "ex_MoveVelVA1". Jetzt ist hinter "ex_MoveVelVA1" ein markiertes "FALSE" und in der Deklarationszeile zunächst ein "TRUE", nach dem nächsten Klick "FALSE" und dann ein leerer Platz (deaktiviert) zu sehen.

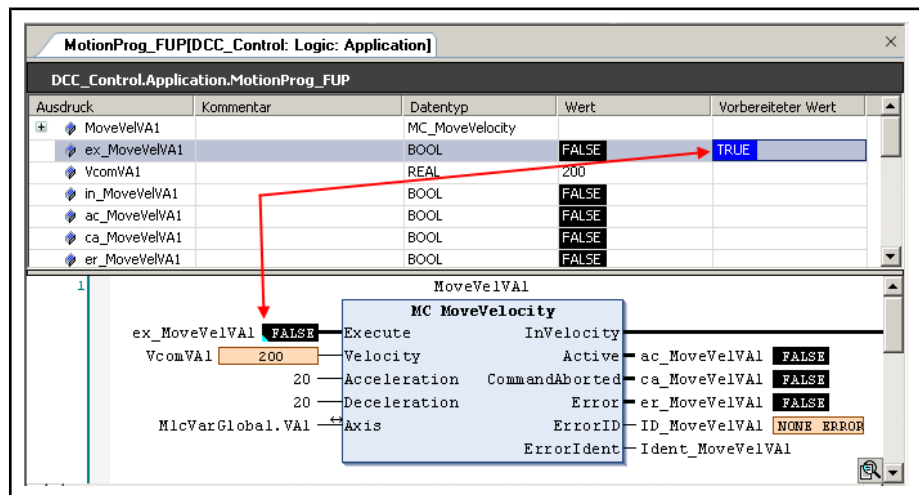


Abb.5-33: Vorbereitung des "Werte schreiben"

Über **Debuggen** ► **Werte schreiben** <Strg>+<F7> wird "ex_MoveVelVA1" in der Steuerung auf "TRUE" gesetzt.

Erste Schritte - Motion



- "Werte schreiben" bedeutet, dass **einmalig** zu Beginn des nächsten Zyklus die Variable auf den gewünschten Wert gesetzt wird.
- "Werte forcen" bedeutet, dass **permanent** zu Beginn der folgenden Zyklen die Variable auf den gewünschten Wert gesetzt wird.

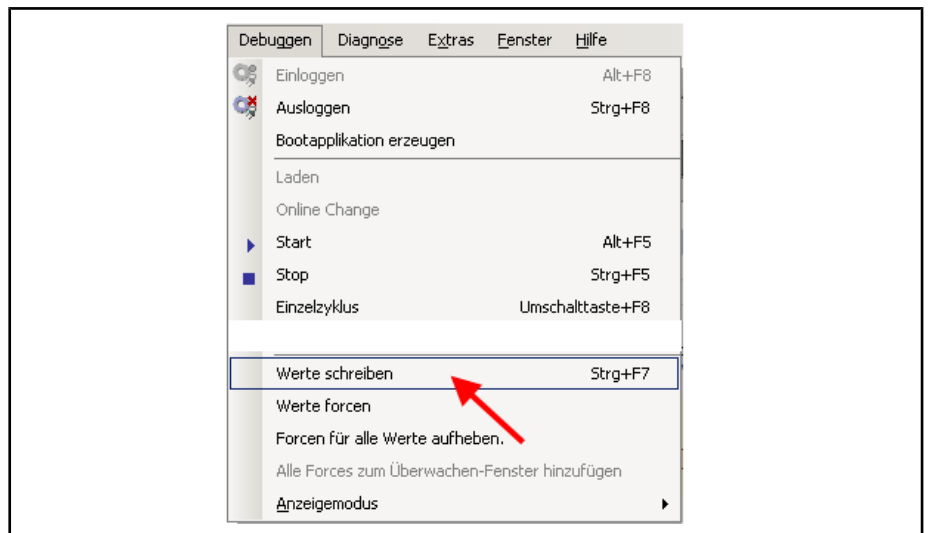


Abb.5-34: Werte schreiben

Die dadurch entstehende positive Flanke setzt die Ausgänge "InVelocity" und "Active" vom Funktionsbaustein "MoveVelVA1" auf "TRUE" ①.

Die Achse bewegt sich mit 200 U/min ②.

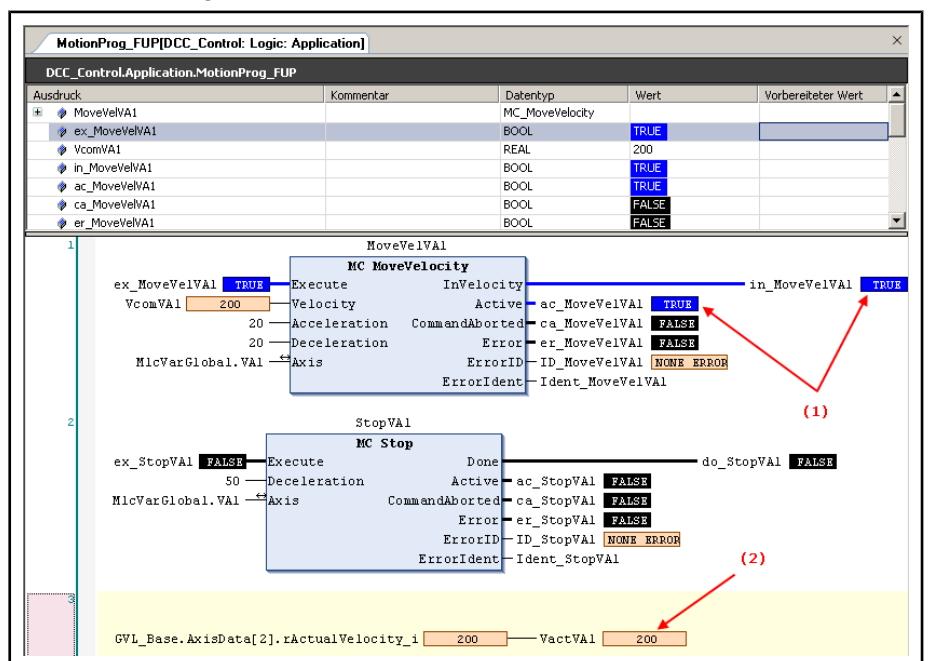


Abb.5-35: Nach "Werte schreiben" am Eingang "Execute"

Schalten Sie "ex_MoveVelVA1" wieder auf "FALSE". Die Achse läuft weiter mit 200 U/min.

Erste Schritte - Motion

Sie können durch Setzen der Variable "ex_StopVA1" am Eingang "Execute" des Funktionsbausteines "StopVA1" vom Typ MC_Stop die Bewegung der Achse stoppen.

Stoppen Sie die Applikation und loggen sich aus.

5.4.4 Aufzeichnung der Bewegung von Achse VA1 (Trace), FUP

IndraLogic liefert zur Aufzeichnung von Bewegungsvorgängen ein Trace-Werkzeug. Mit seiner Hilfe sollen

- die Achsgeschwindigkeit "VactVA1",
- für "MoveVelVA1" (MC_MoveVelocity)
 - "ex_MoveVelVA1" (Execute)
 - "in_MoveVelVA1" (InVelocity)
 - "ca_MoveVelVA1" (CommandAborted)

und

- für "StopVA1" (MC_Stop)
 - "ex_StopVA1" (Execute)

aufgezeichnet werden.

Dazu muss zunächst die Traceaufzeichnung konfiguriert werden.

Fügen Sie über das Kontextmenü von **Application Hinzufügen ► Trace...** das Objekt "Trace_VA1" ein. Öffnen Sie das Objekt mit einem Doppelklick.

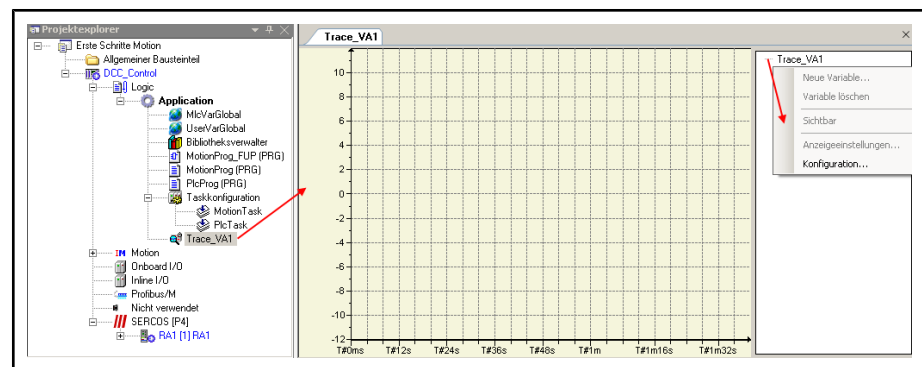


Abb.5-36: Dialog: Neues Trace-Objekt "Trace_VA1" mit Kontextmenü

Über das Kontextmenü auf der rechten Blattseite kann das Trace-Objekt erfolgen:

- Konfiguration...
- Anzeigeeinstellungen... und
- Auswahl der Neuen Variablen...

Konfiguration...

Das neue Trace-Objekt ist zunächst zu konfigurieren:

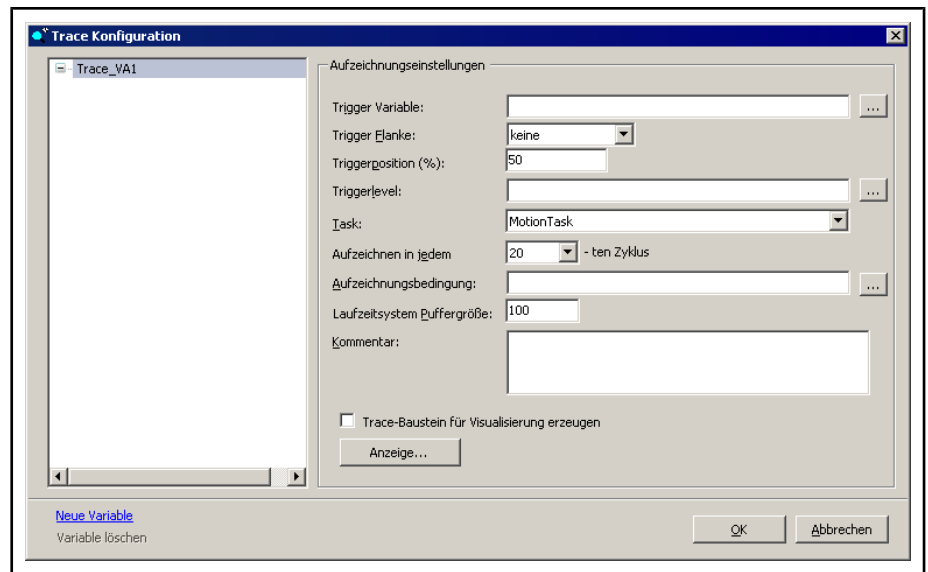


Abb.5-37: Dialog: Neues Trace-Objekt "Trace_VA1", Minimalkonfiguration

Notwendige Einstellungen sind

- Task: **MotionTask**
- Aufzeichnung in jedem **20**-ten Zyklus, also $20 \cdot 2\text{ms} = 40\text{ms}$ -

Task: Von der Liste der verfügbaren Tasks wählen Sie diejenige, nach deren Abarbeitung der Wert der Trace-Variablen gelesen werden soll. Default: Erste Task im Task Konfigurationsbaum.

Aufzeichnen in jedem ...-ten Zyklus: Geben Sie die Anzahl der Zyklen an, die verstreichen sollen, bevor der Wert der Trace-Variablen erneut gelesen werden soll. Mit der Formel (Puffergröße * Aufzeichnen in jedem x-ten Zyklus * Task-Intervall) kann die Mindest-Aufzeichnungszeitspanne abgeschätzt werden.

Aufzeichnungsbedingung: Eine boolesche Variable, ein Wert oder ein boolescher Ausdruck kann hier eingetragen werden. Sobald diese Bedingung eintritt, also Wert TRUE erhält, wird die Wertaufzeichnung gestartet. Wenn hier nichts angegeben ist, startet die Wertaufzeichnung sobald die Trace Konfiguration in die Steuerung geladen wurde und die Applikation läuft.

Laufzeitsystem Puffergröße: Größe des Ringpuffers in Bytes, der im Laufzeitsystem für das Sammeln der Trace-Variablenwerte der gerade konfigurierten Aufzeichnung vorgesehen ist. Der Pufferinhalt wird dann als Wertekurve(n) im Trace-Fenster angezeigt. Default: 100.



Bei einer Änderung der Konfiguration muss das Trace-Objekt erneut eingeloggt werden.

Anzeigeeinstellungen...

Als Nächstes können Einstellungen zur Anzeige vorgenommen werden. Diese Einstellungen werden auch ohne erneutes Einloggen wirksam.

Erste Schritte - Motion

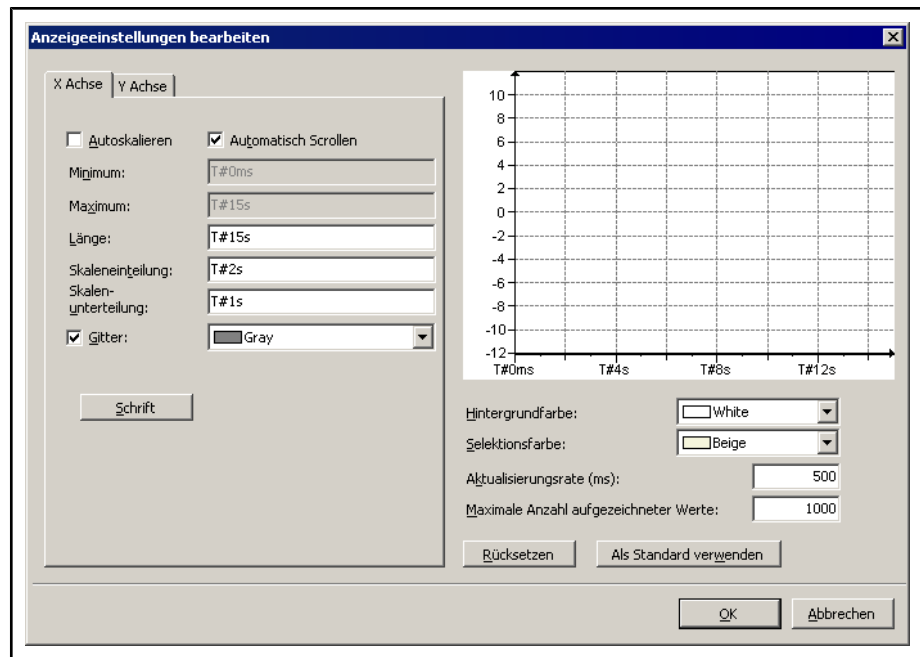


Abb.5-38: Dialog: Anzeigeeinstellungen

Anzuwählen ist **Automatisch Scrollen**, hierbei wird ein Fenster der **Länge 15s** über den Kurvenzug gelegt. Die **Skaleneinteilung** wird dabei im 2s Raster mit einer **Skalenunterteilung** von 1s vorgenommen.



Der zu betrachtende Vorgang sollte also nicht länger als 15s dauern.

Auswahl der Neuen Variablen..

Nun sind die gewünschten Variablen zu benennen und ihr Erscheinungsbild einzustellen;

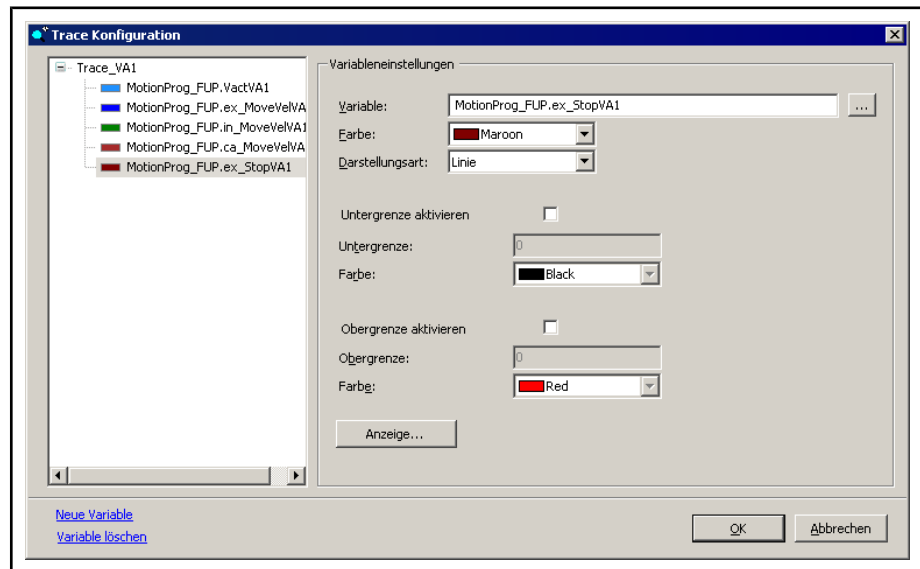


Abb.5-39: Dialog: Anlegen von neuen Variablen und Festlegung des Erscheinungsbildes

Farbe und **Darstellungsart** beziehen sich auf den Kurvenverlauf der **Variablen**.

Erste Schritte - Motion

Zusätzlich können bei jeweils aktivierter Grenze die Teile des Kurvenzuges, die einen Minimalwert unter- bzw. einen Maximalwert überschreiten farblich gegen den übrigen Kurvenzug abgesetzt werden.

Menü zum Steuern der Trace-Aufzeichnung:

Das Kontextmenü erscheint, wenn mit der Rechten-Maus-Taste in den linken Teil des Trace-Fensters geklickt wird.

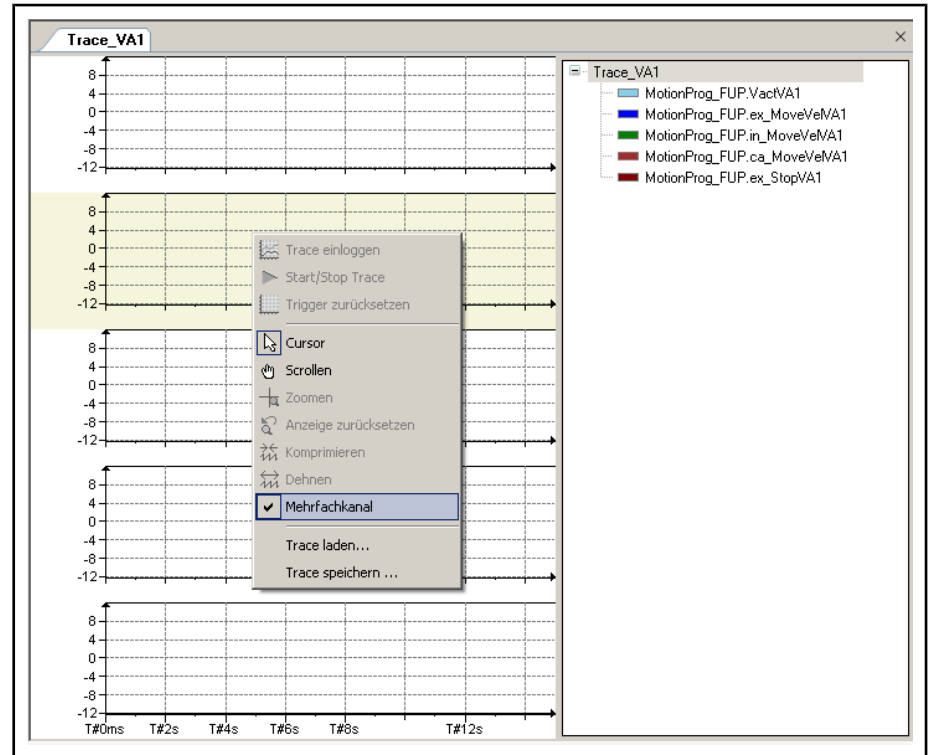


Abb.5-40: Kontextmenü zum Steuern der Trace-Aufzeichnung

Trace einloggen: Bevor eine konfigurierte Trace-Aufzeichnung auf der Steuerung für die dort laufende Applikation aktiv werden kann, muss die "Trace-Applikation" explizit dorthin geladen werden

Start/Stop Trace: Dieser Befehl startet bzw. stoppt die Darstellung der Trace-Aufzeichnung.

Wenn die Aufzeichnung gerade gestoppt ist, steht vor dem Befehl das Symbol 

Wenn die Aufzeichnung gerade läuft, steht vor dem Befehl das Symbol  und ein Ausführen des Befehls bedeutet, dass die Aufzeichnung gestoppt wird, d.h. im Trace-Fenster wird der gerade dargestellte Abschnitt der Variablenwertkurve(n) "eingefroren".

Mehrfachkanal: Dieses Kommando ermöglicht es, die Ansicht der Trace-Kurve zu wechseln. Standardmäßig werden die Kurven aller Variablen im selben Diagramm dargestellt. In Mehrfachkanal-Ansicht wird die Kurve jeder Variablen in ihrem eigenen Diagramm aber mit identischer X-Achse visualisiert. Zoomen und Scrollen beeinflusst die X-Achse aller Diagramme simultan.

Cursor: Solange dieser Befehl aktiviert ist (Default), wird die aktuelle Position des Cursors im Koordinatensystem des Trace-Fensters immer in der Statusleiste des Programmiersystems angezeigt.

Scrollen: In diesem Modus wird der Mauszeiger einen kurzen Moment lang als Doppelpfeil dargestellt und man kann die Zeitachse der gerade dargestellte Trace durch entsprechende Bewegung des Cursors im Trace-Fenster nach links oder rechts verschieben.

Erste Schritte - Motion

Anzeige zurücksetzen: Mit diesem Befehl kann die Anzeige der Aufzeichnung(en) auf die Standardwerte zurückgesetzt werden, nachdem diese beispielsweise durch Zoomen verändert wurde(n).

Durchführung des Trace:

1. Einloggen der Application
Herstellen der Verbindung zur Steuerung.
2. Trace einloggen
Bei laufender Application werden über das Trace-Kontextmenü "Trace einloggen" die erforderlichen Daten in die Steuerung übertragen.
Die Trace-Aufzeichnung läuft danach automatisch an.
3. Werte schreiben
Folgende Werte werden analog zu "[Werte schreiben](#)" der [Eingangsvariablen, Seite 108](#), gesetzt:
 - ex_MoveVelVA1 auf TRUE,
 - ex_StopVA1 auf TRUE.
4. Beenden der Aufzeichnung
Stoppen des Trace durch den Trace-Kontextmenüpunkt "Start/Stop Trace".
5. Auswertung der Trace-Aufzeichnung
Einstellen der Fensterdaten (finden des nachstehenden Bildes).

Tastenkombinationen	Aktion im Trace-Diagramm
<Alt>+<links/rechts>-Pfeiltasten	horizontales Scrollen der X-Achse aller Diagramme
<nach oben/nach unten>-Pfeiltasten	horizontales Scrollen nur des selektierten Diagramms
<+/->-Tasten	simultanes Zoomen des Koordinatensystems entlang der X- und Y-Achse aller Diagramme
<Umsch>+<+/->-Tasten	horizontales Zoomen aller Diagramme entlang der X-Achse
<Strg>+<+/->-Tasten	vertikales Zoomen aller Diagramme entlang der Y-Achse
<links/rechts>-Pfeiltasten	Scrollen des Cursors nach rechts bzw. links

Abb.5-41: Tastenkombinationen des Trace-Editors bei aktiviertem Mehrfachkanal

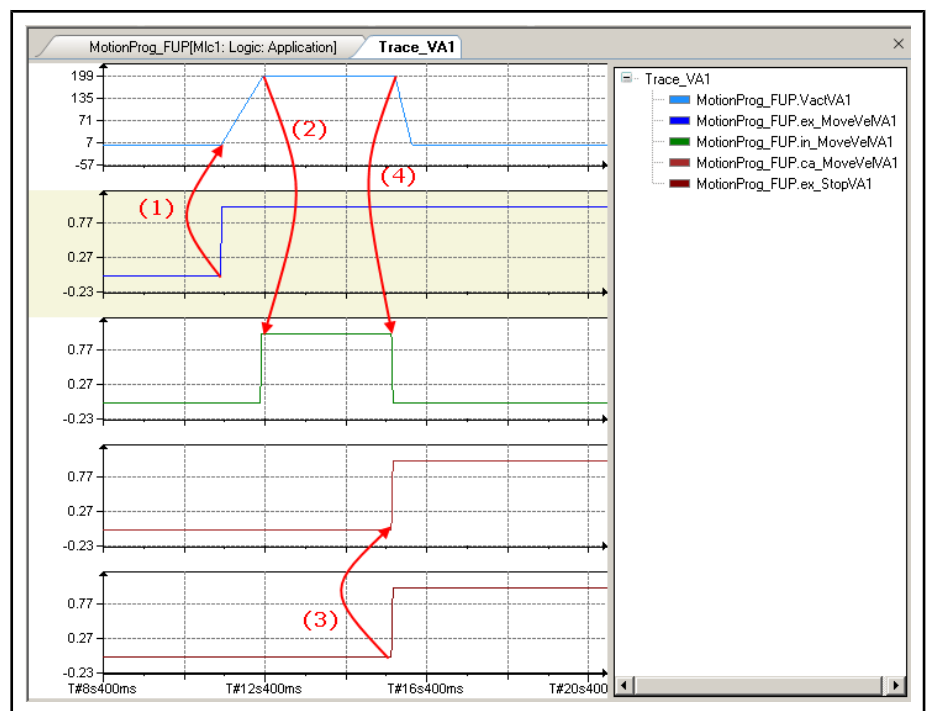


Abb.5-42: Traceaufzeichnung für die virtuelle Achse VA1

Auswertung der Trace-Aufzeichnung

- ① - Start des Hochlaufes der virtuellen Achse
- ② - Solldrehzahl erreicht
- ③ - Stoppen der virtuellen Achse
- ④ - Virtuelle Achse wird abgebremst.

Zu erkennen ist, dass sich die virtuelle Achse analog zu einer realen, gleich-parametrierten Achse verhält. (Vergleiche Geschwindigkeitsverlauf, Beschleunigung mit 20 rad/s^2 und Verzögerung mit 50 rad/s^2).

5.4.5 Programmieren der realen Achse RA1 (FUP)



Ab dieser Stelle benötigen Sie die "reale" Antriebstechnik oder

Sie verwenden ersatzweise eine zweite virtuelle Achse **ohne** MC_Power!!

Die Verwendung einer **realen** Achse setzt zunächst eine Instanz des PLCopen-Bausteins MC_Power voraus.

Als Nächstes werden jeweils eine Instanz von MC_GearIn und MC_GearOut benötigt.

Die reale Achse wird mit einer Instanz von MC_Stop angehalten.

Die Programmierung erfolgt analog zum Abschnitt [Programmieren der virtuellen Achse VA1](#), [Programmiersprache ST](#), Seite 123.

MC_Power

Neues Netzwerk anfügen, Klick mit rechter Maustaste in Netzwerk 3,

Netzwerk einfügen (unterhalb)

MC_Power einfügen und eine Instanz "PowerRA1" deklarieren

Eingänge beschalten und wenn nötig Eingangsvariablen deklarieren:

Erste Schritte - Motion

"Enable" → "en_PowerRA1" vom Typ BOOL

"Axis" → "MlcVarGlobal.RA1" vom Typ "AXIS_REF")

Ausgänge:

"Status" → "st_PowerRA1" vom Typ BOOL

"Error" → "er_PowerRA1" vom Typ BOOL

"ErrorID" → "ID_PowerRA1" vom Typ ERROR_CODE

"ErrorIdent" → "Ident_PowerRA1" vom Typ ERROR_STRUCT

MC_GearIn

Neues Netzwerk anfügen, Klick mit rechter Maustaste in Netzwerk 4,

Netzwerk einfügen (unterhalb)

MC_GearIn einfügen und Instanz "GearInRA1" deklarieren

Eingänge beschalten und wenn nötig Eingangsvariablen deklarieren:

"Execute" → "ex_GearInRA1" vom Typ BOOL

"RatioNumerator" → 5

"RatioDenominator" → 10

"MasterFineadjust" → 20 (%)

"Master" → "MlcVarGlobal.VA1" vom Typ "AXIS_REF")

"Slave" → "MlcVarGlobal.RA1" vom Typ "AXIS_REF")

Ausgänge:

"InGear" → "ig_GearInRA1" vom Typ BOOL

"Active" → "ac_GearInRA1" vom Typ BOOL

"CommandAborted" → "ca_GearInRA1" vom Typ BOOL

"Error" → "er_GearInRA1" vom Typ BOOL

"ErrorID" → "ID_GearInRA1" vom Typ ERROR_CODE

"ErrorIdent" → "Ident_GearInRA1" vom Typ ERROR_STRUCT

MC_GearOut

Neues Netzwerk anfügen, Klick mit rechter Maustaste in Netzwerk 5,

Netzwerk einfügen (unterhalb)

MC_GearOut einfügen und eine Instanz "GearOutRA1" deklarieren

Eingänge beschalten und wenn nötig Eingangsvariablen deklarieren:

"Execute" → "ex_GearOutRA1" vom Typ BOOL

"Slave" → "MlcVarGlobal.RA1" vom Typ "AXIS_REF")

Ausgänge:

"Done" → "do_GearOutRA1" vom Typ BOOL

"Active" → "ac_GearOutRA1" vom Typ BOOL

"Error" → "er_GearOutRA1" vom Typ BOOL

"ErrorID" → "ID_GearOutRA1" vom Typ ERROR_CODE

"ErrorIdent" → "Ident_GearOutRA1" vom Typ ERROR_STRUCT

MC_Stop

Neues Netzwerk anfügen, Klick mit rechter Maustaste in Netzwerk 6,

Netzwerk einfügen (unterhalb)

MC_Stop einfügen und eine Instanz "StopRA1" deklarieren

Eingänge beschalten und wenn nötig Eingangsvariablen deklarieren:

"Execute" → "ex_StopRA1" vom Typ BOOL

"Deceleration" → 50 (rad/s²)

"Axis" → "MlcVarGlobal.RA1" vom Typ "AXIS_REF")

Ausgänge:

"Done" → "do_StopRA1" vom Typ BOOL

"Active" → "ac_StopRA1" vom Typ BOOL

"CommandAborted" → "ca_StopRA1" vom Typ BOOL

"Error" → "er_StopRA1" vom Typ BOOL

"ErrorID" → "ID_StopRA1" vom Typ ERROR_CODE

"ErrorIdent" → "Ident_StopRA1" vom Typ ERROR_STRUCT

Istgeschwindigkeit "VactRA1"

Als Letztes soll ein Netzwerk, in dem die aktuelle Geschwindigkeit der Achse RA1 "VactRA1" (Datentyp REAL) abgelesen werden kann, implementiert werden.

Dazu muss die aktuelle Geschwindigkeit aus dem Datensatz der Achse (Shared Memory) gelesen werden.

Die Daten aller Achsen, die eine MLC steuern kann, sind in dem Feld

GVL_Base.AxisData: ARRAY [1..xx²] OF ML_AXISDATA_SM

enthalten, wobei als Index die Achsnummer zu verwenden ist.

Die Variable wird unter dem Namen

GVL_Base.AxisData[1].rActualVelocity_i,

für die reale Achse, Achsnummer 1, bereitgestellt (Achsnummer 1 hat RA1, da das Scannen von RA1 vor der Deklaration von VA1 erfolgte).

- Neues Netzwerk anfügen, Klick mit rechter Maustaste in Netzwerk 7, **Netzwerk einfügen (unterhalb)**
- Zuweisung einfügen,
- Auf den rechten **???** ist "VactRA1" einzufügen, <Enter>, Typ "REAL" im Deklarationsfenster auswählen <Enter>.
- Auf die linken **???** gehen und danach  anklicken und unter Nutzung der Eingabehilfe zu **Variablen ▶ ML_BASE ▶ Global Variables List ▶ GVL_Base ▶ AxisData[]** navigieren:

2) Die Anzahl der steuerbaren Achsen ist vom Typ der Steuerung und ihrer Firmware abhängig, aktuell für L65: 64 Achsen

Erste Schritte - Motion

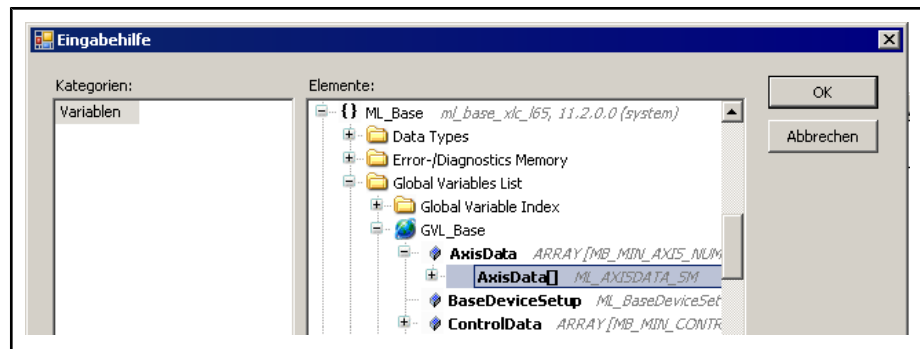


Abb.5-43: Geschwindigkeitswert VA1 aus der Globalen Variablenliste der Bibliothek ML_Base.library

Die weitere Eingabe wird durch die Intellisense-Funktion unterstützt. Dies bedeutet, dass nach der Eingabe des "." bei "GVL_Base.AxisData[1]." alle verfügbaren Elemente aufgerollt werden und zur Auswahl stehen.

"rActualVelocity_i" mit <Enter> bestätigen.

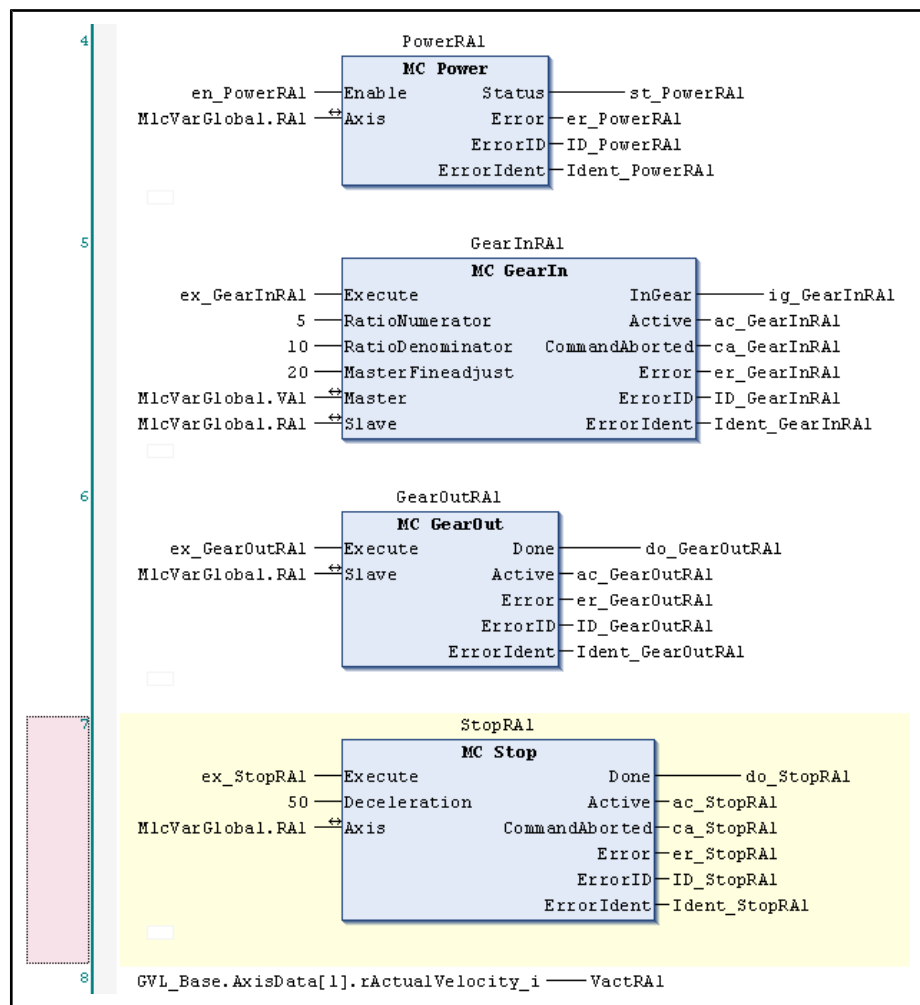


Abb.5-44: Implementation der realen Achse

5.4.6 Download des kompletten SPS-Projektes (FUP)

Der Ladevorgang verläuft analog zum Abschnitt [Download des SPS-Teilprojekts virtuelle Achse VA1 \(FUP\)](#), Seite 106.

Das neue Projekt überschreibt dabei nach Rückfrage das alte Projekt.

5.4.7 Workflow "Virtuelle Achse → Elektronisches Getriebe → Reale Achse"

Der nachstehende Arbeitsablauf zeigt das Starten und Stoppen der Achsen und des Getriebes.

1. Leistungszuschaltung für die reale Achse
Zuschalten des "en_PowerRA1" (MC_Power) durch "Werte schreiben".
st_PowerRA1= TRUE (MC_Power).
2. Start der virtuellen Achse
Zuschalten des "ex_MoveVelVA1" (MC_MoveVelocity) durch "Werte schreiben".
Hochlauf bei VactVA1 = 200 U/min beendet.
3. Nachführen der realen Achse
Zuschalten des "ex_GearInRA1" (MC_GearIn) durch "Werte schreiben".
Das zu erwartenden Übersetzungsverhältnis des Getriebes ergibt sich zu:
$$\dot{U}_v = (\text{RatioNumerator} / \text{RatioDenominator}) * (1 + \text{MasterFineadjust})$$

also bei VactVA1 = 200 U/min ergibt sich VactRA1 = 120 U/min.



Der MasterFineadjust-Wert wird in "%" vorgegeben.

4. Befindet sich der MC_GearIn "InGear", so kann die Geschwindigkeit "VcomVA1" der virtuellen Achse durch "Werte schreiben" (Doppelklick auf die Variable, gewünschten Wert eintragen) verändert werden, die reale Achse folgt ihr.
5. Abschalten des "ex_GearInRA1" (MC_GearIn) durch "Werte schreiben".
in_GearInRA1= FALSE, ac_GearInRA1= TRUE.
Die reale Achse folgt der virtuellen weiterhin.
6. Auflösen der Synchronität
Zuschalten des "ex_GearOutRA1" (MC_GearOut) durch "Werte schreiben".
Die Abhängigkeit der realen Achse von der virtuellen ist aufgehoben. Beide Achsen drehen jedoch mit unveränderter Geschwindigkeit weiter.
7. Änderung der Sollgeschwindigkeit "VcomVA1"
Die veränderte Sollgeschwindigkeit wirkt sich nur noch auf die virtuelle Achse aus, "VactRA1" bleibt unverändert.
8. Abschalten des "ex_GearOutRA1" durch "Werte schreiben".
do_GearOutRA1= FALSE
9. Stoppen der virtuellen Achse
Zuschalten des "ex_StopVA1" (MC_Stop)
Die virtuelle Achse wird abgebremst auf VactVA1= 0 U/min.
10. Abschalten des "ex_StopVA1" (MC_Stop).
11. Stoppen der realen Achse
Zuschalten des "ex_StopRA1" (MC_Stop).

Erste Schritte - Motion

Die reale Achse wird abgebremst auf VactRA1= 0 U/min. Sie wird nach Erreichen des Stillstandes aktiv in dieser Position gehalten.

12. Abschalten des "ex_StopRA1" (MC_Stop).

Das aktive Halten der realen Achse ist beendet.

13. Leistungsabschaltung für die reale Achse

Abschalten von "en_PowerRA1" (MC_Power) durch "Werte schreiben".



Nur möglich im Zustand Standstill nach Ausführung des Stops an RA1.

5.4.8 Aufzeichnung der Bewegung beider Achsen (Trace), FUP

Das Trace-Werkzeug wurde analog zum Abschnitt [Aufzeichnung der Bewegung von Achse VA1 \(Trace\), FUP, Seite 110](#), verwendet, um den Hochlauf der virtuellen Achse und das Aufsynchronisieren sichtbar zu machen.

Der nachstehende Arbeitsablauf zeigt das Starten und Stoppen der Achsen und des Getriebes für die Trace-Aufzeichnung.

1. Leistungszuschaltung für die reale Achse
Zuschalten des "en_PowerRA1" (MC_Power).
2. Start der virtuellen Achse
Zuschalten des "ex_MoveVelVA1" (MC_MoveVelocity).
3. Nachführen der realen Achse
Zuschalten des "ex_GearInRA1" (MC_GearIn).
4. Drehzahlsollwertsprung der virtuellen Achse
Erhöhen des Drehzahlsollwertes
Wirksamschalten durch "ex_MoveVelVA1" TRUE ⇒ FALSE ⇒ TRUE.

Die nachstehende Abbildung zeigt das Trace-Ergebnis:

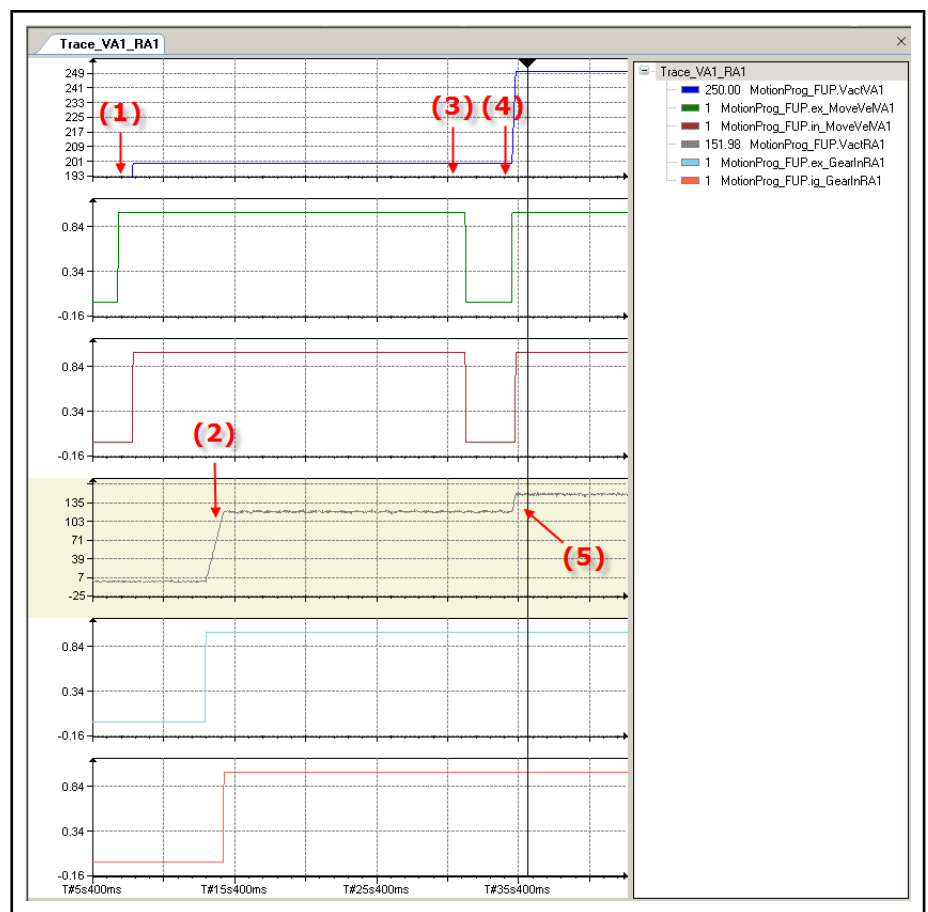


Abb.5-45: Traceaufzeichnung für beide Achsen

Auswertung der Trace-Aufzeichnung

- ① - Start des Hochlaufes der virtuellen Achse,
- ② - Zuschalten der Getriebe-Funktion, Hochlauf der realen Achse,
- ③ - Erhöhen der Solldrehzahl der virtuellen Achse, "ex_MoveVelVA1" TRUE ⇒ FALSE,
- ④ - Wirksamwerden der Solldrehzahl der virtuellen Achse, "ex_MoveVelVA1" FALSE ⇒ TRUE,
- ⑤ - Sprungantwort an der realen Achse.

5.5 Programmieren des Bewegungsablaufes in Strukturiertem Text (ST)

5.5.1 Programmieren des Bewegungsablaufes in ST, Allgemeines

Der Bewegungsablauf wird in einem SPS-Programm unter Nutzung von PLCopen-Funktionsbausteinen programmiert (Programmiersystem IndraLogic 2G nach EN 61131-3).

Beim Anlegen des Projektes mit der Steuerung DCC_Control, der realen Achse RA1 und der virtuellen Achse VA1 wurden unterhalb des Ordners "Logic" erste Aktivitäten zur Programmierung des Bewegungsablaufes begonnen:

Erste Schritte - Motion

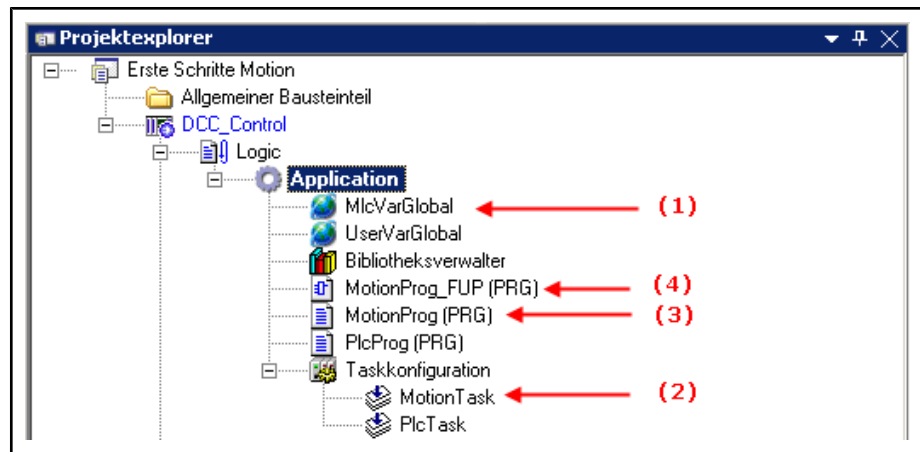


Abb.5-46: Projektexplorer nach Expandieren der Knoten "Logic"

Angelegt wurden die Programme:

- **MotionProg (3)** (Vorwahl ST, Strukturierter Text),
- **PlcProg** (Vorwahl ST, Strukturierter Text), nicht verwendet.

Das dritte Programm **MotionProg_FUP (4)**, wurde manuell ergänzt: **Application**, rechte Maustaste, **Hinzufügen** ► **POU**, Eingabe des Namens und Auswahl von FUP als Programmiersprache.

Die Programme sind noch leer.

Angelegt wurden unter **Taskkonfiguration** die **Tasks**:

- **MotionTask (2)**, (Priorität 2, extern ereignisgesteuert durch SERCOS-Task, Watchdog aktiv) für "MotionProg".

Das Programm/ die Programme, die unter dieser Task laufen, müssen komplett innerhalb des fest vorgegebenen Teil eines SERCOS-Zyklus abgearbeitet werden. Überschreiten sie diesen Rahmen, wird die Steuerung gestoppt.

- **PlcTask**, (niederpriore zyklische Task) für PlcProg, nicht verwendet.

Das Programm wird in der verbleibenden Zeit zumindest stückweise, also im Bedarfsfall über mehrere Zyklen, abgearbeitet.

Angelegt wurden **Globale Variablen / Globale Konstanten**:

- **UserVarGlobal**, reserviert für den Anwender, in "Erste Schritte - Motion" nicht genutzt,
- **MlcVarGlobal (1)**, enthält die Adresskonstanten für die Achsen RA1 und VA1.

Programm:

```
(* Automated generated code by MLC. *)
(* Please don't edit, the code will be overwritten...*)
VAR_GLOBAL CONSTANT
(*virtual axis "VA1" with axis number 2*)
VA1: AXIS_REF :=(CntrlNo:=LOCAL_CNTRL,AxisNo:=AXIS_2);
(*real axis "RA1" with axis number 1*)
RA1: AXIS_REF :=(CntrlNo:=LOCAL_CNTRL,AxisNo:=AXIS_1);
(*...*)
END_VAR
```



Wir nutzen für die "Ersten Schritte Motion" die MotionTask.

Je nach gewünschter Programmiersprache ist ihr entweder **MotionProg** oder **MotionProg_FUP** zugeordnet.

Erste Schritte - Motion

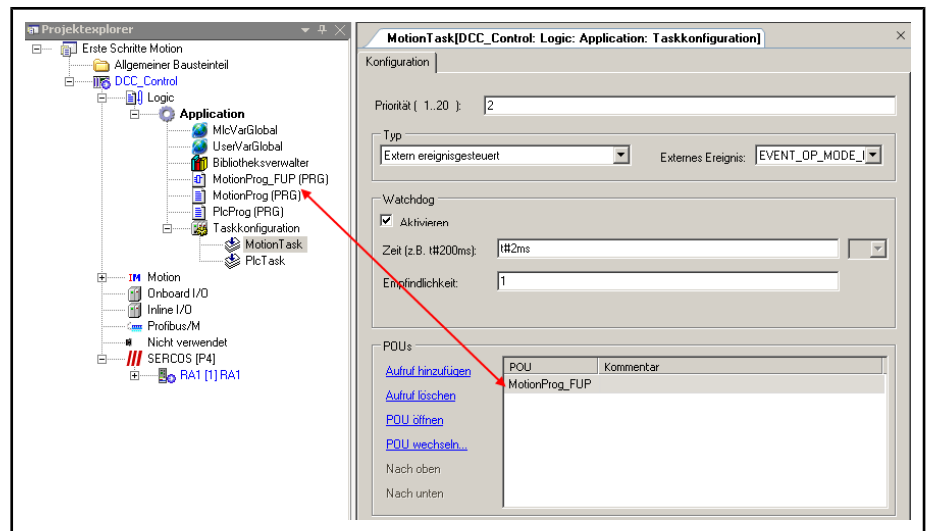


Abb.5-47: Aufruf löschen: MotionProg_FUP; Aufruf hinzufügen: MotionProg

5.5.2 Programmieren der virtuellen Achse VA1, Programmiersprache ST



Der Abschnitt verwendet ST (Strukturierter Text) als Programmiersprache. Funktionell entsteht das gleiche Ergebnis, wie bei den Abschnitten mit der Programmiersprache FUP.

Entsprechend der [Ziel des Projekts Erste Schritte - Motion, Seite 87](#), werden Instanzen der PLCopen-Bausteine MC_MoveVelocity und MC_Stop unter Nutzung von "VA1" aufgerufen.



Eine Instanz des MC_Power wird für virtuelle Achsen nicht benötigt.

Gehen Sie dazu folgendermaßen vor:

1. Ordnen Sie der MotionTask das Programm "MotionProg" zu.
2. Öffnen Sie das Programm "MotionProg" (Doppelklick Linke-Maus-Taste).

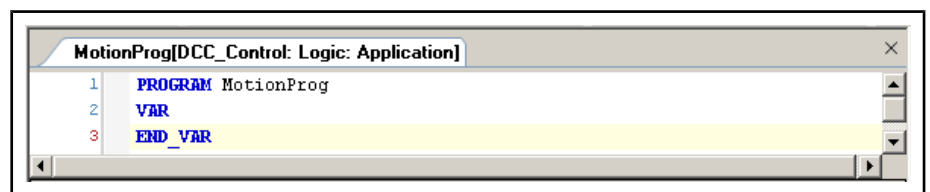


Abb.5-48: Neues Programm in ST (Strukturierter Text), nach Löschen der Dummy-Zeile

Im Unterschied zur Programmierung in der Sprache FUP sind für die Sprache ST (Strukturierter Text) zunächst die Instanzen der Funktionsbausteine im Deklarationsteil, Bereich VAR ... END_VAR, zu deklarieren.

Erst dann können sie in der Implementation verwendet werden.

Tragen Sie zunächst im Bereich VAR ... END_VAR den Namen der Instanz des Funktionsbausteins MC_MoveVelocity "MoveVelVA1" ein. Rufen Sie per rechter Maustaste das Kontextmenü und hier die Eingabehilfe (alternativ <F2>-Fenster).

Erste Schritte - Motion

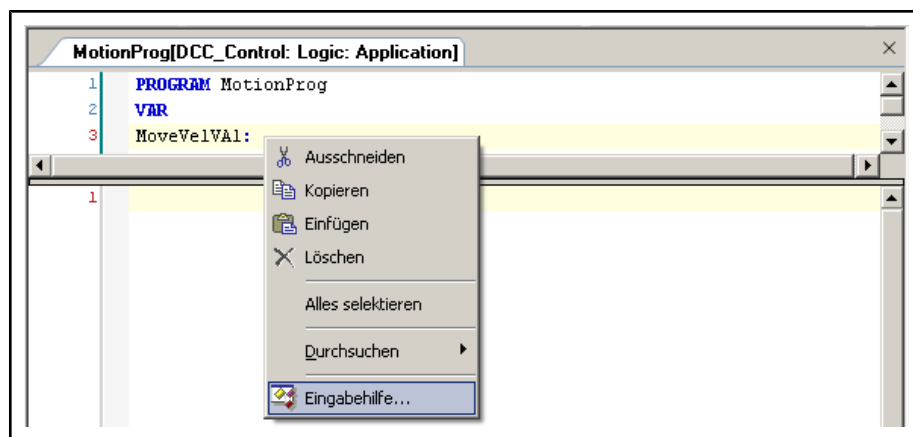


Abb.5-49: Deklaration der Instanz MoveVelVA1 vom Typ MC_MoveVelocity

Wählen Sie dazu im geöffneten Auswahlfenster

Benutzerdefinierte Typen ► **ML_PLCopen** ► **POUs** ► **Motion** ► **MC_MoveVelocity** aus.

Wählen Sie die Option "Mit Argumenten einfügen" ab.

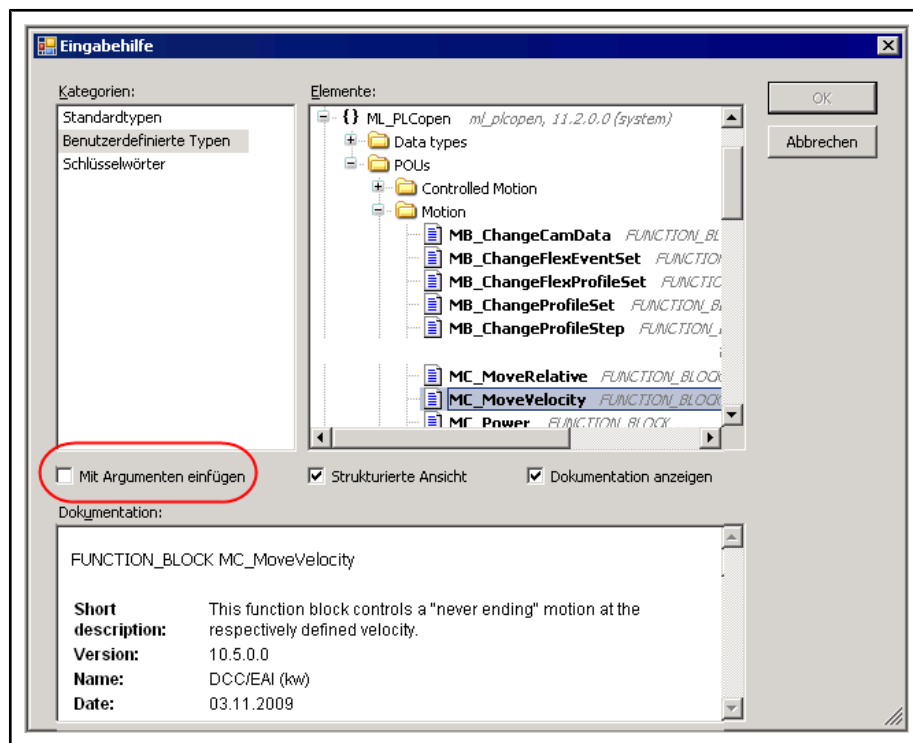


Abb.5-50: Auswahlfenster Funktionsbausteine

Nach Bestätigung mit "OK" komplettieren Sie bitte die Zeile mit einem ";".

Gehen Sie in den Implementationsteil des Editors auf Zeile 1 und betätigen <F2>, Eingabehilfe.

Wählen Sie im geöffneten Auswahlfenster **Instanzaufrufe** ► **MoveVelVA1** aus.

Wählen Sie die Option "Mit Argumenten einfügen".

Erste Schritte - Motion

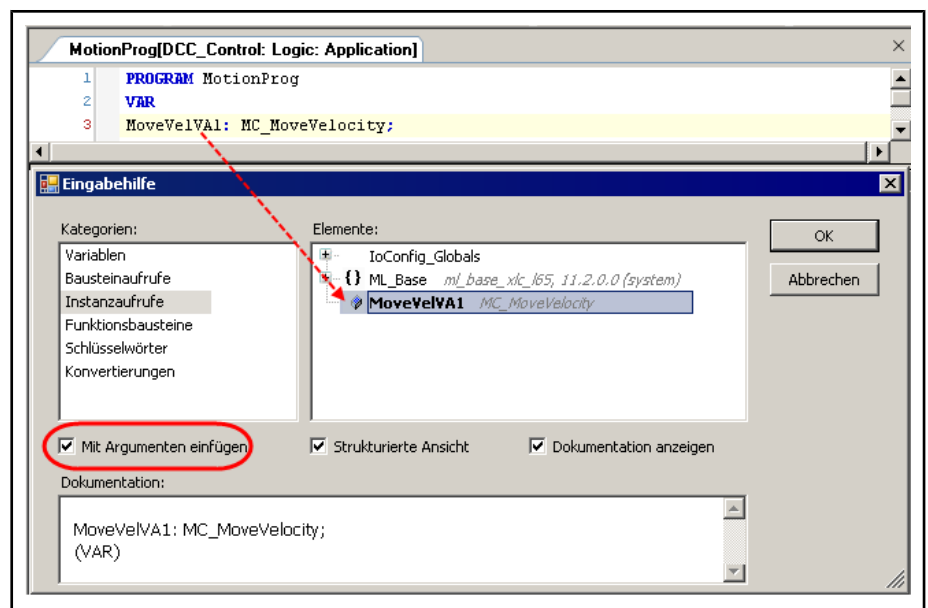


Abb.5-51: Übernahme der Instanz in die Implementation

Bestätigen Sie jetzt das Einfügen der Instanz mit "OK".

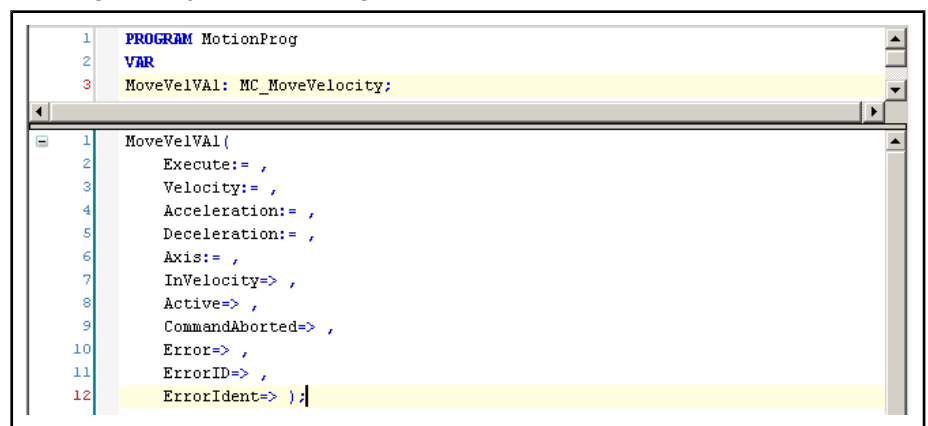


Abb.5-52: Gerüst einer Funktionsbaustein-Instanz

Beschalten Sie nun die Eingänge des Bausteines.

Beginnen Sie mit dem Eingang "Execute":

- Geben Sie den Namen "ex_MoveVelVA1" ein, und schließen Sie mit "Enter" ab.
- Das sich öffnende Fenster zur "Variablendeklaration" können Sie sofort mit "Enter" bestätigen.

Um den Eingang "Velocity" zu beschalten, gehen Sie wie folgt vor:

- Geben Sie den Namen "VcomVA1" ein, Bestätigung mit "Enter".
- Als Typ tragen Sie "REAL" und als Initialwert 200 (U/min) ein, Bestätigung mit "Enter".

Für die Eingänge "Acceleration" tragen Sie bitte 20 (rad/s²) und für "Deceleration" 20 (rad/s²) ein.



Die Zahlen werden direkt an den Bausteineingängen eingetragen. Alternativ kann eine Variable mit dem entsprechenden Wert verwendet werden.

Erste Schritte - Motion

Um den Eingang "Axis" zu beschalten, gehen Sie wie folgt vor:

- Rechte-Maus-Taste, wählen der nebenstehenden Eingabehilfe:
Wählen Sie **Variablen** ► **Application** ► **MlcVarGlobal** ► **VA1**.

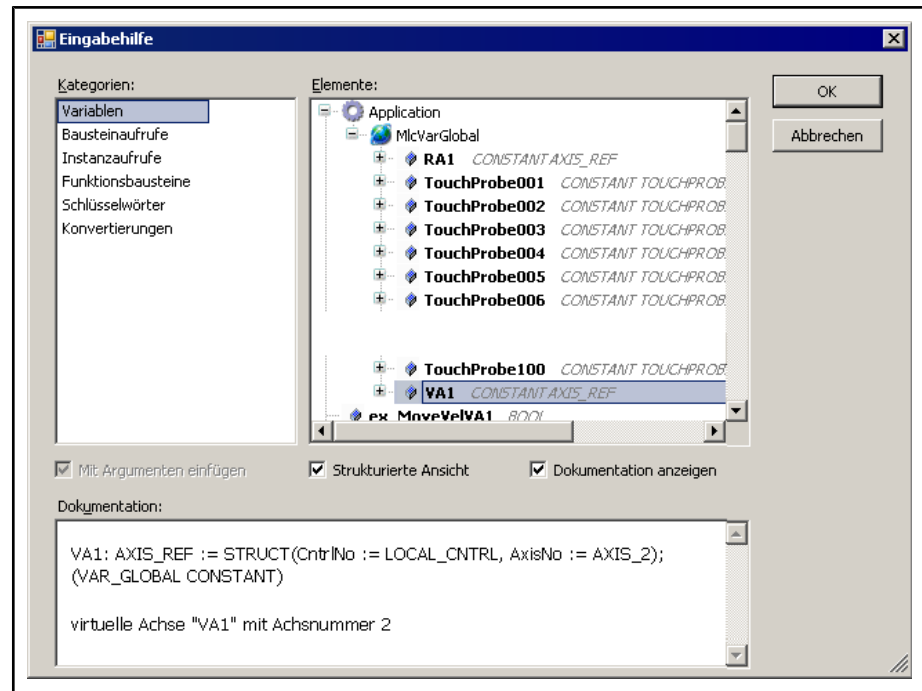


Abb.5-53: Auswahl der Achsinstanz bei Beschaltung eines Funktionsbausteines vom Typ "MC_MoveVelocity"

- Bestätigen Sie mit "OK".



Der Name einer globalen Konstanten setzt sich aus dem Namen des Ordners und des Elementes zusammen:
`MlcVarGlobal.VA1`

Um eine Rückmeldung zu erhalten, ob der Funktionsbaustein korrekt gearbeitet hat, werden seine Ausgänge beschaltet.

- In bekannter Art und Weise beschalten Sie den Ausgang "InVelocity" des Bausteines mit der Variablen "iv_MoveVelVA1", Typ BOOL:
- Geben Sie den Namen "iv_MoveVelVA1" ein, <Enter>.
- Das sich öffnende Fenster zur "Variablendeklaration" können Sie sofort mit <Enter> bestätigen.
- Beschalten Sie den CommandAborted-Ausgang mit "ca_MoveVelVA1".
- Beschalten Sie den Error-Ausgang mit "er_MoveVelVA1".
- Beschalten Sie den ErrorID-Ausgang mit "ID_MoveVelVA1".
- Beschalten Sie den ErrorIdent-Ausgang mit "Ident_MoveVelVA1".

Damit ist der PLCopen-Baustein vom Typ MC_MoveVelocity entsprechend der Aufgabenstellung beschaltet. Der Deklarationsteil wurde automatisch mit komplettiert.

```

2  VAR
3  MoveVelVal: MC_MoveVelocity;
4  ex_MoveVelVal: BOOL;
5  VcomVal: REAL := 200;
6  iv_MoveVelVal: BOOL;
7  ac_MoveVelVal: BOOL;
8  ca_MoveVelVal: BOOL;
9  er_MoveVelVal: BOOL;
10 ID_MoveVelVal: ERROR_CODE;
11 Ident_MoveVelVal: ERROR_STRUCT;
12 END VAR

1 MoveVelVal(
2   Execute:= ex_MoveVelVal,
3   Velocity:= VcomVal,
4   Acceleration:= 20,
5   Deceleration:= 20,
6   Axis:= MlcVarGlobal.VA1,
7   InVelocity=> iv_MoveVelVal,
8   Active=> ac_MoveVelVal,
9   CommandAborted=> ca_MoveVelVal,
10  Error=> er_MoveVelVal,
11  ErrorID=> ID_MoveVelVal,
12  ErrorIdent=> Ident_MoveVelVal);

```

Abb.5-54: PLOpen-FB Typ "MC_MoveVelocity" ist eingefügt und beschaltet



Immer wenn Sie einen Baustein abgeschlossen haben, lassen Sie eine Testübersetzung <Umsch>+<F11> laufen. Beseitigen Sie danach eventuell vorhandene Fehler sofort.

Generieren Sie nun eine Instanz des PLOpen-Bausteins MC_Stop.

MC_Stop

Instanz "StopVA1" von MC_Stop deklarieren und einfügen

Eingänge beschalten und wenn nötig Eingangsvariablen deklarieren:

"Execute" → "ex_StopVA1" vom Typ BOOL

"Deceleration" → 50

"Axis" → "MlcVarGlobal.VA1" vom Typ "AXIS_REF")

Ausgänge:

"Done" → "do_StopVA1" vom Typ BOOL

"Active" → "ac_StopVA1" vom Typ BOOL

"CommandAborted" → "ca_StopVA1" vom Typ BOOL

"Error" → "er_StopVA1" vom Typ BOOL

"ErrorID" → "ID_StopVA1" vom Typ ERROR_CODE

"ErrorIdent" → "Ident_StopVA1" vom Typ ERROR_STRUCT

Istgeschwindigkeit "VactVA1"

Als Letztes soll ein Netzwerk, in dem die aktuelle Geschwindigkeit der Achse VA1 "VactVA1" (Datentyp REAL) abgelesen werden kann, implementiert werden.

Dazu muss die aktuelle Geschwindigkeit aus dem Datensatz der Achse (Shared Memory) gelesen werden.

Die Daten aller Achsen, die eine Steuerung steuern kann, sind in dem Feld

GVL_Base.AxisData: ARRAY [1..xx³] OF ML_AXISDATA_SM

enthalten, wobei als Index die Achsnummer zu verwenden ist.

Die Variable wird unter dem Namen

Erste Schritte - Motion

GVL_Base.AxisData[2].rActualVelocity_i,

für die virtuelle Achse, Achsnummer 2, bereitgestellt (Achsnummer 1 hat RA1, da Scannen von RA1 vor der Deklaration von VA1 erfolgte).

- Neue Zeile anfügen,
- `VactVA1:=` einfügen, Deklaration komplettieren,
- unter Nutzung der Eingabehilfe zu **Variablen ▶ ML_Base ▶ Global Variables List ▶ GVL_Base ▶ AxisData[]** navigieren:

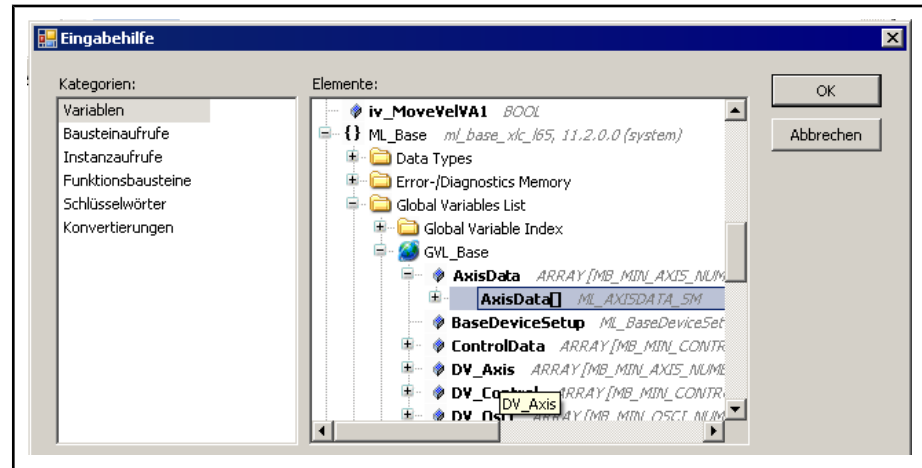


Abb.5-55: Geschwindigkeitswert VA1 aus der Globalen Variablenliste der Bibliothek ML_Base.library

Die weitere Eingabe wird durch die Intellisense-Funktion unterstützt. Dies bedeutet, dass nach der Eingabe des "." bei "GVL_Base.AxisData[2]." alle verfügbaren Elemente aufgerollt werden und zur Auswahl stehen.

"rActualVelocity_i" mit <Enter> bestätigen.

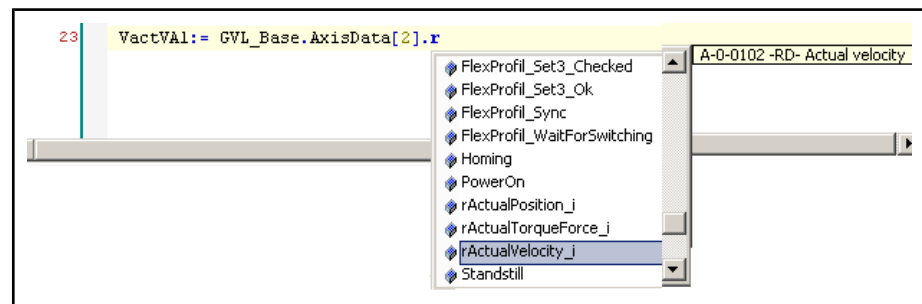


Abb.5-56: Auswahl der Variablen aktuelle Geschwindigkeit

- Abschluss der Zeile mit ";" nicht vergessen.

Damit ist die Implementierung des Bewegungsablaufes entsprechend der Aufgabenstellung abgeschlossen:

3) Die Anzahl der steuerbaren Achsen ist vom Typ der Steuerung und ihrer Firmware abhängig, aktuell für L65: 64 Achsen

```

1  PROGRAM MotionProg
2  VAR
3  MoveVelVAL: MC_MoveVelocity;
4  ex_MoveVelVAL: BOOL;
5  VcomVAL: REAL := 200;
6  iv_MoveVelVAL: BOOL;
7  ac_MoveVelVAL: BOOL;

1  MoveVelVAL(
2  Execute:= ex_MoveVelVAL,
3  Velocity:= VcomVAL,
4  Acceleration:= 20,
5  Deceleration:= 20,
6  Axis:= MlcVarGlobal.VAL,
7  InVelocity=> iv_MoveVelVAL,
8  Active=> ac_MoveVelVAL,
9  CommandAborted=> ca_MoveVelVAL,
10 Error=> er_MoveVelVAL,
11 ErrorID=> ID_MoveVelVAL,
12 ErrorIdent=> Ident_MoveVelVAL);
13 StopVAL(
14 Execute:= ex_StopVAL,
15 Deceleration:= 50,
16 Axis:= MlcVarGlobal.VAL,
17 Done=> do_StopVAL,
18 Active=> ac_StopVAL,
19 CommandAborted=> ca_StopVAL,
20 Error=> er_StopVAL,
21 ErrorID=> ID_StopVAL,
22 ErrorIdent=> Ident_StopVAL);
23 VactVAL:= GVL_Base.AxisData[2].rActualVelocity_i;

```

Abb.5-57: Implementation der virtuellen Achse VA1

5.5.3 Download des SPS-Teilprojektes virtuelle Achse VA1 (ST)

Der zuvor implementierte Bewegungsablauf muss jetzt noch übersetzt, in die Steuerung übertragen und gestartet werden.

Die Übersetzung erreichen Sie im Menüpunkt

- **Erstellen ► Übersetzen,**
Übersetzen des hinzu gekommenen Anteils der Applikation.
- **Erstellen ► Neu übersetzen,**
Übersetzen der kompletten Applikation.

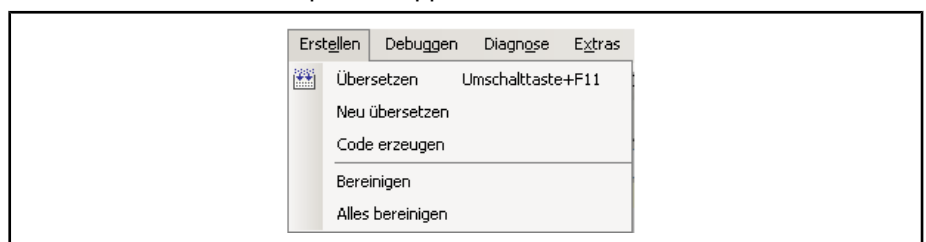


Abb.5-58: Applikation übersetzen

Zum Übertragen der kompilierten Daten in die Zielsteuerung wählen Sie den Menüpunkt **Debuggen ► Einloggen** (nachfolgendes Bild).

Erste Schritte - Motion



Abb.5-59: Übertragen der kompilierten Daten in die Steuerung

In Abhängigkeit ob eine Applikation bereits geladen ist und ob diese Applikation eine Vorgängerin Ihrer aktuellen Applikation ist, werden Sie um Zustimmung zum Download gebeten.

Nach erfolgreicher Übertragung ist die Applikation noch gestoppt (rot im Projektnavigator und in der Statuszeile), und die Variablen in Deklaration und Implementation sind zur Anzeige der Werte vorbereitet. Die Werte sind jedoch noch nicht gültig.

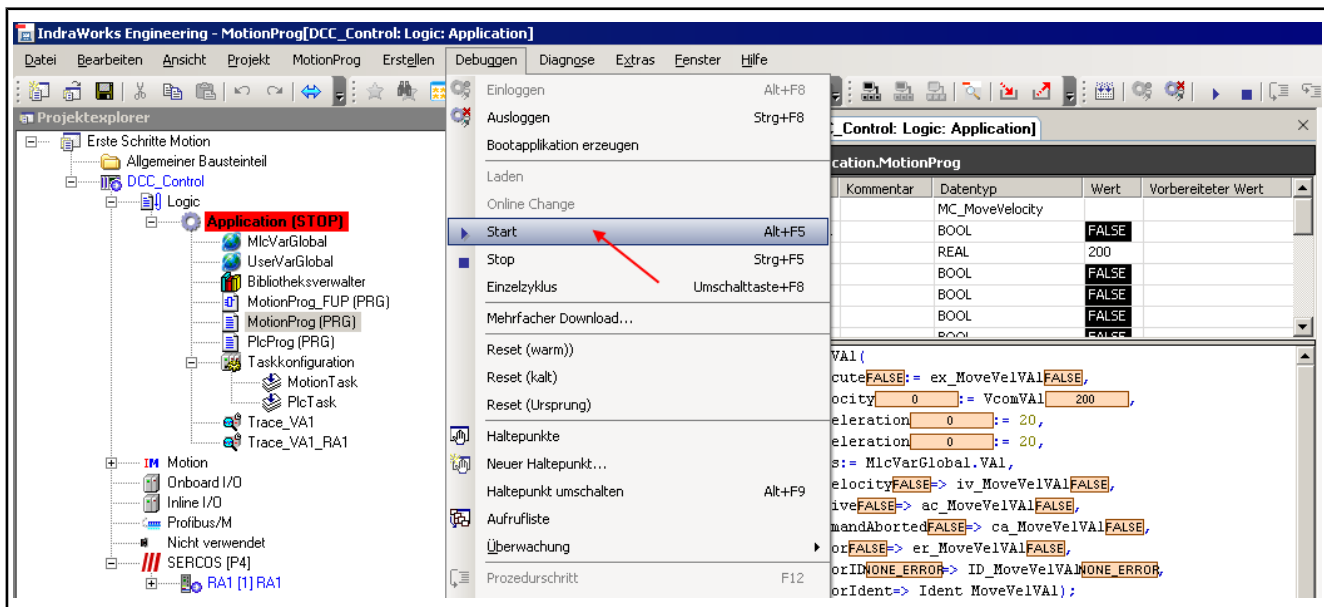


Abb.5-60: Starten der Programmbearbeitung in der Steuerung (1)

Starten Sie die Bearbeitung in der Steuerung über **Debuggen ▶ Start**.

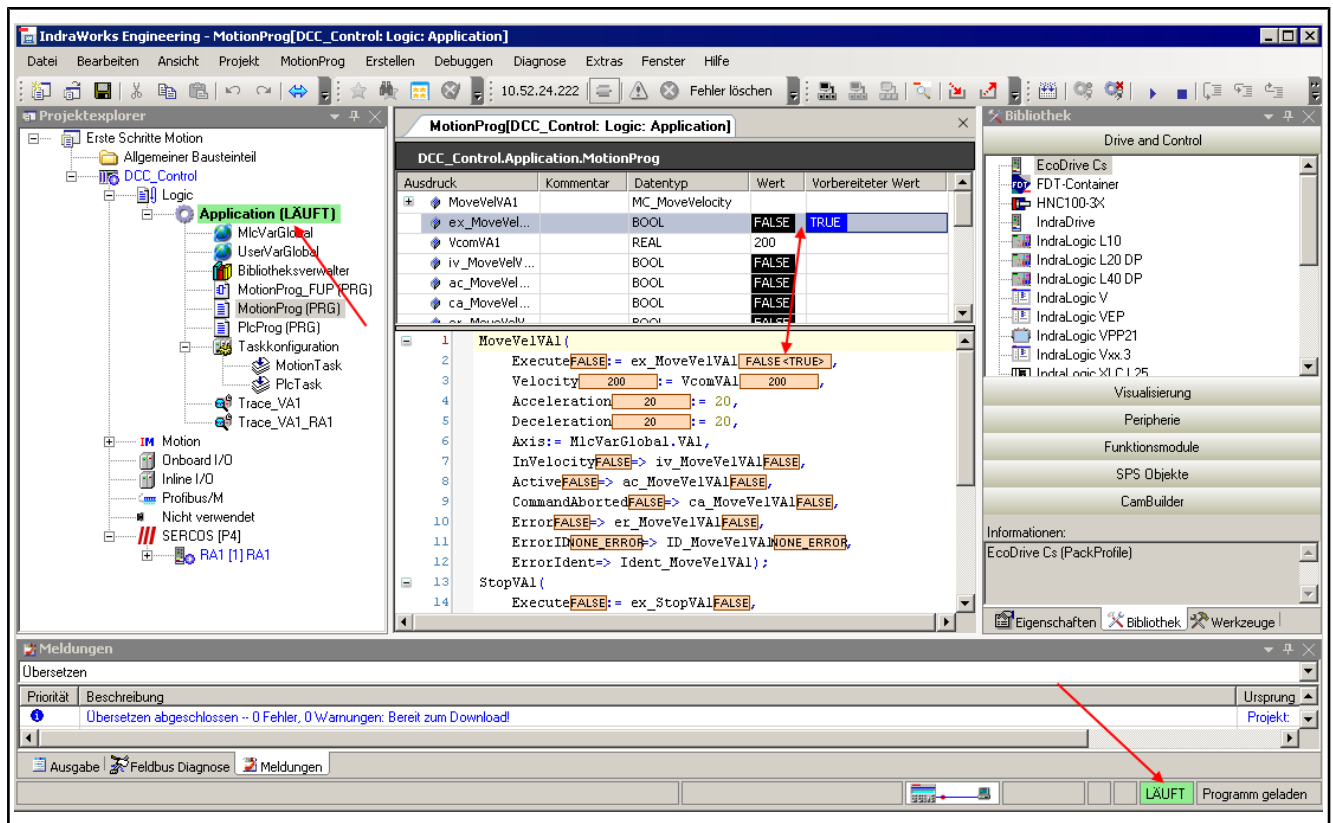


Abb.5-61: Starten der Programmbearbeitung in der Steuerung (2)

Der Bewegungsablauf selbst wird angestoßen, indem nacheinander die Abarbeitung der programmierten Funktionsbausteine aktiviert wird.

"Werte schreiben", Eingangsvariablen

Um die Bearbeitung des "MoveVelVA1" vom Typ "MC_MoveVelocity" zu aktivieren, muss am "Execute"-Eingang des Bausteins eine positive Flanke auftreten.

Klick mit der linken Maustaste auf das FALSE hinter der Eingangsvariablen "ex_MoveVelVA1", das Fenster "Wert vorbereiten" öffnet, "TRUE" eintragen und bestätigen.

Jetzt ist hinter "ex_MoveVelVA1" ein markiertes "FALSE<TRUE>" und in der Deklarationszeile ein "TRUE" zu sehen.

Erste Schritte - Motion

MotionProg[DCC_Control: Logic: Application]

DCC_Control.Application.MotionProg

Ausdruck	Kommentar	Datentyp	Wert	Vorbereiteter Wert
MoveVelVA1		MC_MoveVelocity		
ex_MoveVelVA1		BOOL	FALSE	TRUE
VcomVA1		REAL	200	
iv_MoveVelVA1		BOOL	FALSE	
ac_MoveVelVA1		BOOL	FALSE	
ca_MoveVelVA1		BOOL	FALSE	
ex_MoveVelVA1		BOOL	FALSE	

1 MoveVelVA1 (
2 ExecuteFALSE := ex_MoveVelVA1 FALSE<TRUE>,
3 Velocity 200 := VcomVA1 200,
4 Acceleration 20 := 20,
5 De Wert vorbereiten
6 Ax
7 In
8 Ac
9 Co
10 Er Aktueller Wert: FALSE
11 Er
12 Er
13 Was möchten Sie tun?
14 ☒ Einen neuen Wert für die nächste Schreib- oder Force-Operation vorbereiten:
15 TRUE
16 Ax
17 Do
18 Ac
19 Co
20 Er
21 Er
22 Er
23 VactVA
24 RETURN

Wert vorbereiten

Ausdruck: MotionProg_FUP.ex_MoveVelVA1
Datentyp: BOOL
Aktueller Wert: FALSE

Was möchten Sie tun?

☒ Einen neuen Wert für die nächste Schreib- oder Force-Operation vorbereiten:

☐ Vorbereitung mit einem Wert aufheben
☐ Den Force aufheben, ohne den Wert zu verändern.
☐ Den Force aufheben und den Wert auf den Wert zurücksetzen, den er vor der Force-Operation hatte.

OK Abbrechen

Abb.5-62: Vorbereitung des "Werte schreiben"

Über **Debuggen** ► **Werte schreiben** <Strg>+<F7> wird "ex_MoveVelVA1" in der Steuerung auf "TRUE" gesetzt.



- "Werte schreiben" bedeutet, dass **einmalig** zu Beginn des nächsten Zyklus die Variable auf den gewünschten Wert gesetzt wird.
- "Werte forcen" bedeutet, dass **permanent** zu Beginn der folgenden Zyklen die Variable auf den gewünschten Wert gesetzt wird.

Erste Schritte - Motion

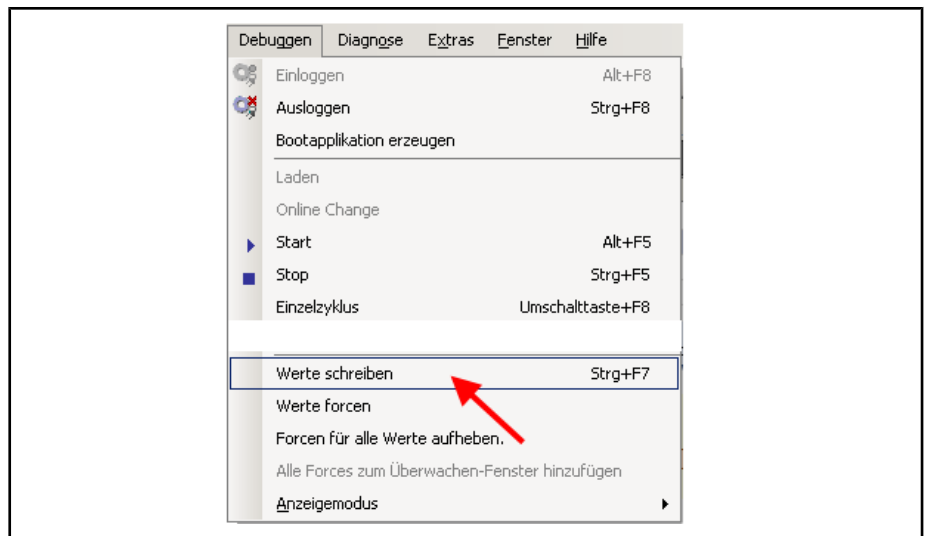


Abb.5-63: Werte schreiben

Die dadurch entstehende positive Flanke setzt die Ausgänge "InVelocity" und "Active" vom Funktionsbaustein "MoveVelVA1" auf "TRUE" (1), (nachstehendes Bild).

Die Achse bewegt sich mit 200 U/min (2).

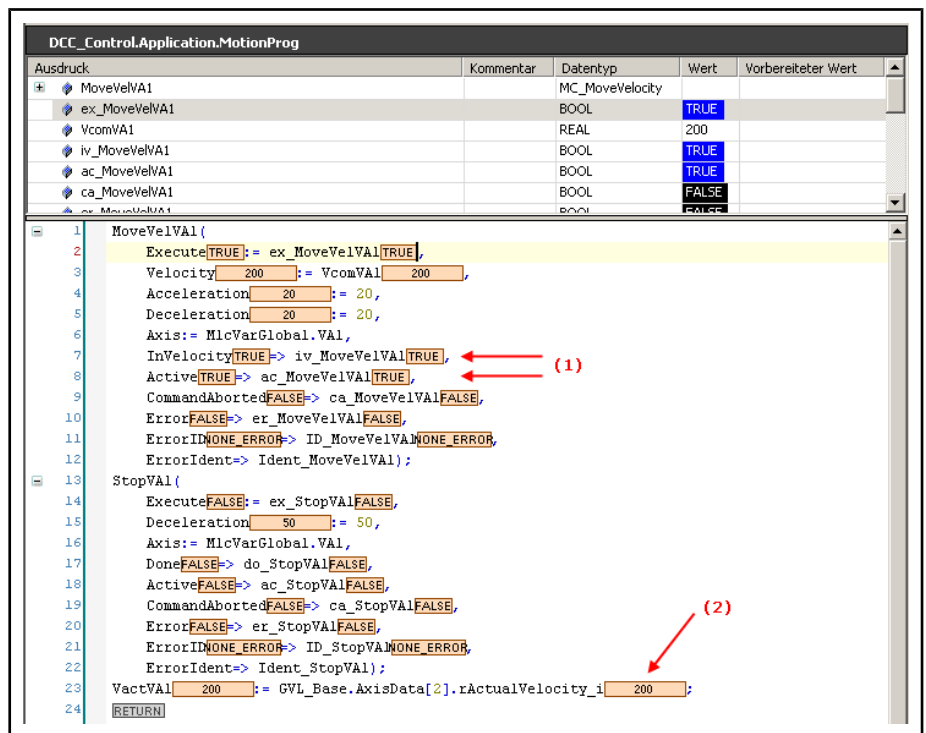


Abb.5-64: Nach "Werte schreiben" am Eingang "Execute"

Schalten Sie "ex_MoveVelVA1" wieder auf "FALSE". Die Achse läuft weiter mit 200 U/min.

Sie können durch Setzen der Variable "ex_StopVA1" am Eingang "Execute" des Funktionsbausteines "StopVA1" vom Typ MC_Stop die Bewegung der Achse stoppen.

Stoppen Sie die SPS und loggen Sie aus.

Erste Schritte - Motion

5.5.4 Aufzeichnung der Bewegung von Achse VA1 (Trace), ST

IndraLogic liefert zur Aufzeichnung von Bewegungsvorgängen ein Trace-Werkzeug. Mit seiner Hilfe sollen

- die Achsgeschwindigkeit "VactVA1",
- für "MoveVelVA1" (MC_MoveVelocity)
 - "ex_MoveVelVA1" (Execute)
 - "in_MoveVelVA1" (InVelocity)
 - "ca_MoveVelVA1" (CommandAborted)

und

- für "StopVA1" (MC_Stop)
 - "ex_StopVA1" (Execute)

aufgezeichnet werden.

Dazu muss zunächst die Traceaufzeichnung konfiguriert werden.

Fügen Sie über das Kontextmenü von **Application Hinzufügen ► Trace...** das Objekt "Trace_VA1" ein. Öffnen Sie das Objekt mit einem Doppelklick.

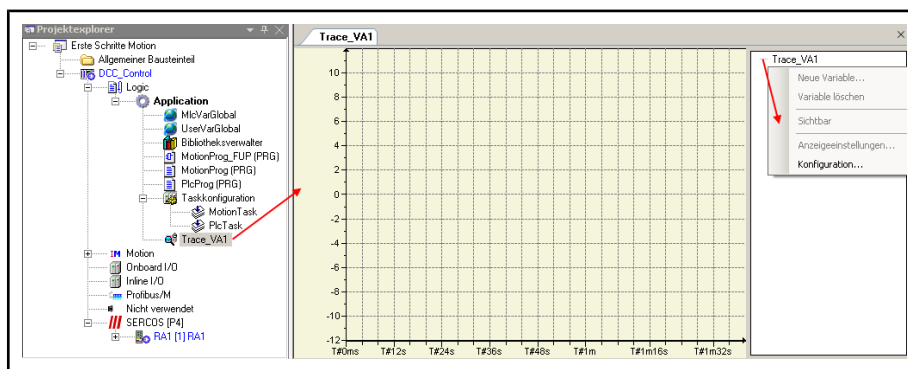


Abb.5-65: Dialog: Neues Trace-Objekt "Trace_VA1" mit Kontextmenü

Über das Kontextmenü auf der rechten Blattseite kann das Trace-Objekt erfolgen:

- Konfiguration...
- Anzeigeeinstellungen... und
- Auswahl der Neuen Variablen...

Konfiguration...

Das neue Trace-Objekt ist zunächst zu konfigurieren:

Erste Schritte - Motion

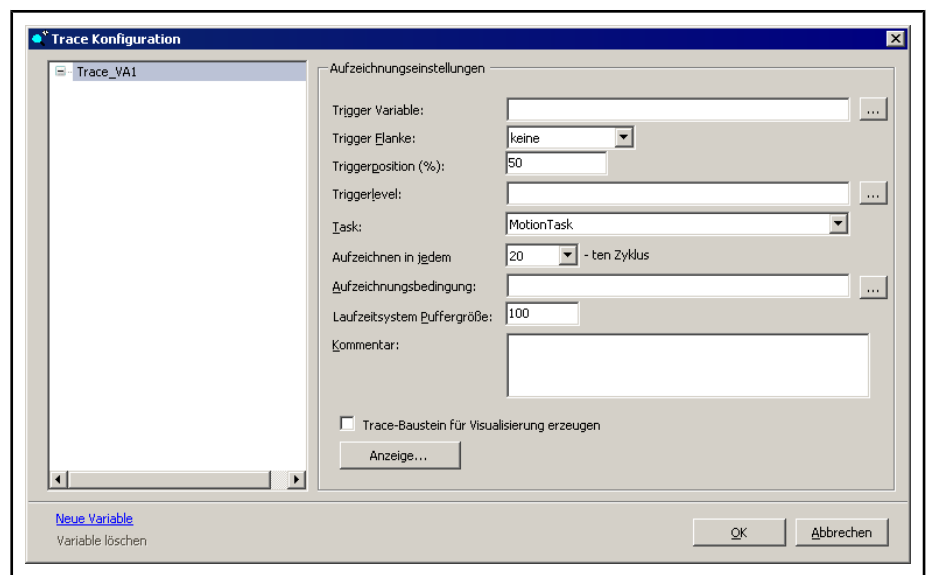


Abb.5-66: Dialog: Neues Trace-Objekt "Trace_VA1", Minimalkonfiguration

Notwendige Einstellungen sind

- Task: **MotionTask**
- Aufzeichnung in jedem **20**-ten Zyklus, also $20 \cdot 2\text{ms} = 40\text{ms}$

Task: Von der Liste der verfügbaren Tasks wählen Sie diejenige, nach deren Abarbeitung der Wert der Trace-Variablen gelesen werden soll. Default: Erste Task im Task Konfigurationsbaum.

Aufzeichnen in jedem ...-ten Zyklus: Geben Sie die Anzahl der Zyklen an, die verstreichen sollen, bevor der Wert der Trace-Variablen erneut gelesen werden soll. Mit der Formel (Puffergröße * Aufzeichnen in jedem x-ten Zyklus * Task-Intervall) kann die Mindest-Aufzeichnungszeitspanne abgeschätzt werden.

Aufzeichnungsbedingung: Eine boolesche Variable, ein Wert oder ein boolescher Ausdruck kann hier eingetragen werden. Sobald diese Bedingung eintritt, also Wert TRUE erhält, wird die Wertaufzeichnung gestartet. Wenn hier nichts angegeben ist, startet die Wertaufzeichnung sobald die Trace Konfiguration in die Steuerung geladen wurde und die Applikation läuft.

Laufzeitsystem Puffergröße: Größe des Ringpuffers in Bytes, der im Laufzeitsystem für das Sammeln der Trace-Variablenwerte der gerade konfigurierten Aufzeichnung vorgesehen ist. Der Pufferinhalt wird dann als Wertekurve(n) im Trace-Fenster angezeigt. Default: 100.



Bei einer Änderung der Konfiguration muss das Trace-Objekt erneut eingeloggt werden.

Anzeigeeinstellungen...

Als Nächstes können Einstellungen zur Anzeige vorgenommen werden. Diese Einstellungen werden auch ohne erneutes Einloggen wirksam.

Erste Schritte - Motion

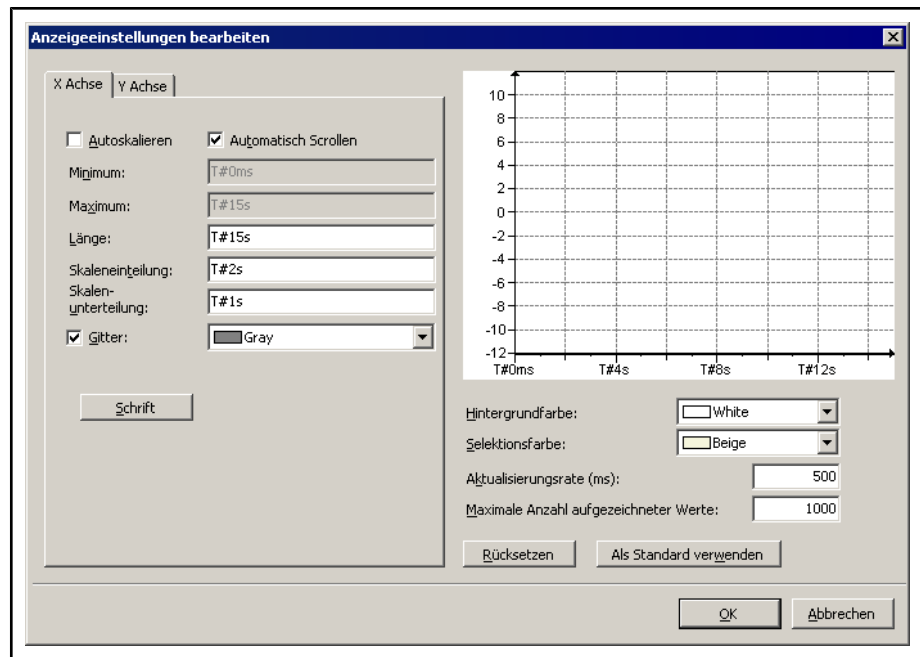


Abb.5-67: Dialog: Anzeigeeinstellungen

Anzuwählen ist **Automatisch Scrollen**, hierbei wird ein Fenster der **Länge 15s** über den Kurvenzug gelegt. Die **Skaleneinteilung** wird dabei im 2s Raster mit einer **Skalenunterteilung** von 1s vorgenommen.



Der zu betrachtende Vorgang sollte also nicht länger als 15s dauern.

Auswahl der Neuen Variablen..

Nun sind die gewünschten Variablen zu benennen und ihr Erscheinungsbild einzustellen;

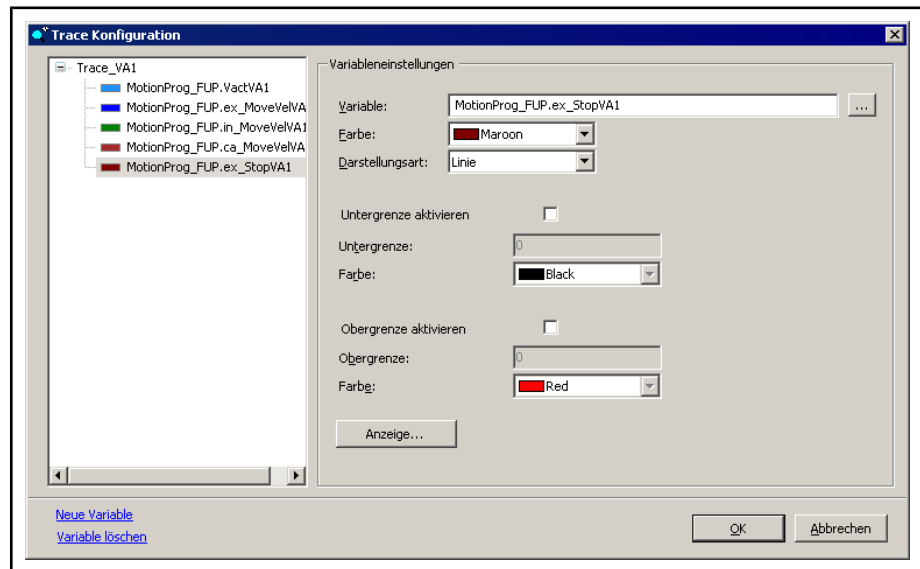


Abb.5-68: Dialog: Anlegen von neuen Variablen und Festlegung des Erscheinungsbildes

Farbe und **Darstellungsart** beziehen sich auf den Kurvenverlauf der **Variablen**.

Erste Schritte - Motion

Menü zum Steuern der Trace-Aufzeichnung:

Zusätzlich können bei jeweils aktivierter Grenze die Teile des Kurvenzuges, die einen Minimalwert unter- bzw. einen Maximalwert überschreiten farblich gegen den übrigen Kurvenzug abgesetzt werden.

Das Kontextmenü erscheint, wenn mit der Rechten-Maus-Taste in den linken Teil des Trace-Fensters geklickt wird.

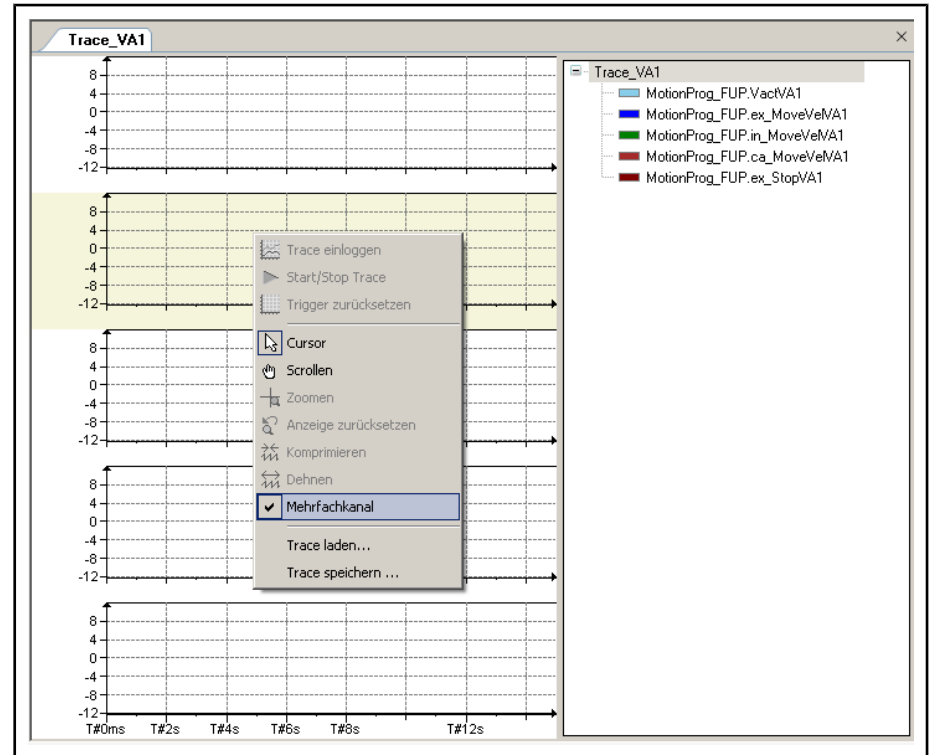


Abb.5-69: Kontextmenü zum Steuern der Trace-Aufzeichnung

Trace einloggen: Bevor eine konfigurierte Trace-Aufzeichnung auf der Steuerung für die dort laufende Applikation aktiv werden kann, muss die "Trace-Applikation" explizit dorthin geladen werden

Start/Stop Trace: Dieser Befehl startet bzw. stoppt die Darstellung der Trace-Aufzeichnung.

Wenn die Aufzeichnung gerade gestoppt ist, steht vor dem Befehl das Symbol 

Wenn die Aufzeichnung gerade läuft, steht vor dem Befehl das Symbol  und ein Ausführen des Befehls bedeutet, dass die Aufzeichnung gestoppt wird, d.h. im Trace-Fenster wird der gerade dargestellte Abschnitt der Variablenwertkurve(n) "eingefroren".

Mehrfachkanal: Dieses Kommando ermöglicht es, die Ansicht der Trace-Kurve zu wechseln. Standardmäßig werden die Kurven aller Variablen im selben Diagramm dargestellt. In Mehrfachkanal-Ansicht wird die Kurve jeder Variablen in ihrem eigenen Diagramm aber mit identischer X-Achse visualisiert. Zoomen und Scrollen beeinflusst die X-Achse aller Diagramme simultan.

Cursor: Solange dieser Befehl aktiviert ist (Default), wird die aktuelle Position des Cursors im Koordinatensystem des Trace-Fensters immer in der Statusleiste des Programmiersystems angezeigt.

Scrollen: In diesem Modus wird der Mauszeiger einen kurzen Moment lang als Doppelpfeil dargestellt und man kann die Zeitachse der gerade dargestellte Trace durch entsprechende Bewegung des Cursors im Trace-Fenster nach links oder rechts verschieben.

Erste Schritte - Motion

Anzeige zurücksetzen: Mit diesem Befehl kann die Anzeige der Aufzeichnung(en) auf die Standardwerte zurückgesetzt werden, nachdem diese beispielsweise durch Zoomen verändert wurde(n).

Durchführung des Trace:

1. Einloggen der Application
Herstellen der Verbindung zur Steuerung.
2. Trace einloggen
Bei laufender Application werden über das Trace-Kontextmenü "Trace einloggen" die erforderlichen Daten in die Steuerung übertragen.
Die Trace-Aufzeichnung läuft danach automatisch an.
3. Werte schreiben
Folgende Werte werden analog zu "[Werte schreiben](#)" der [Eingangsvariablen, Seite 131](#), gesetzt:
 - ex_MoveVelVA1 auf TRUE,
 - ex_StopVA1 auf TRUE.
4. Beenden der Aufzeichnung
Stoppen des Trace durch den Trace-Kontextmenüpunkt "Start/Stop Trace".
5. Auswertung der Trace-Aufzeichnung
Einstellen der Fensterdaten (finden des nachstehenden Bildes).

Tastenkombinationen	Aktion im Trace-Diagramm
<Alt>+<links/rechts>-Pfeiltasten	horizontales Scrollen der X-Achse aller Diagramme
<nach oben/nach unten>-Pfeiltasten	horizontales Scrollen nur des selektierten Diagramms
<+/->-Tasten	simultanes Zoomen des Koordinatensystems entlang der X- und Y-Achse aller Diagramme
<Umsch>+<+/->-Tasten	horizontales Zoomen aller Diagramme entlang der X-Achse
<Strg>+<+/->-Tasten	vertikales Zoomen aller Diagramme entlang der Y-Achse
<links/rechts>-Pfeiltasten	Scrollen des Cursors nach rechts bzw. links

Abb.5-70: Tastenkombinationen des Trace-Editors bei aktiviertem Mehrfachkanal

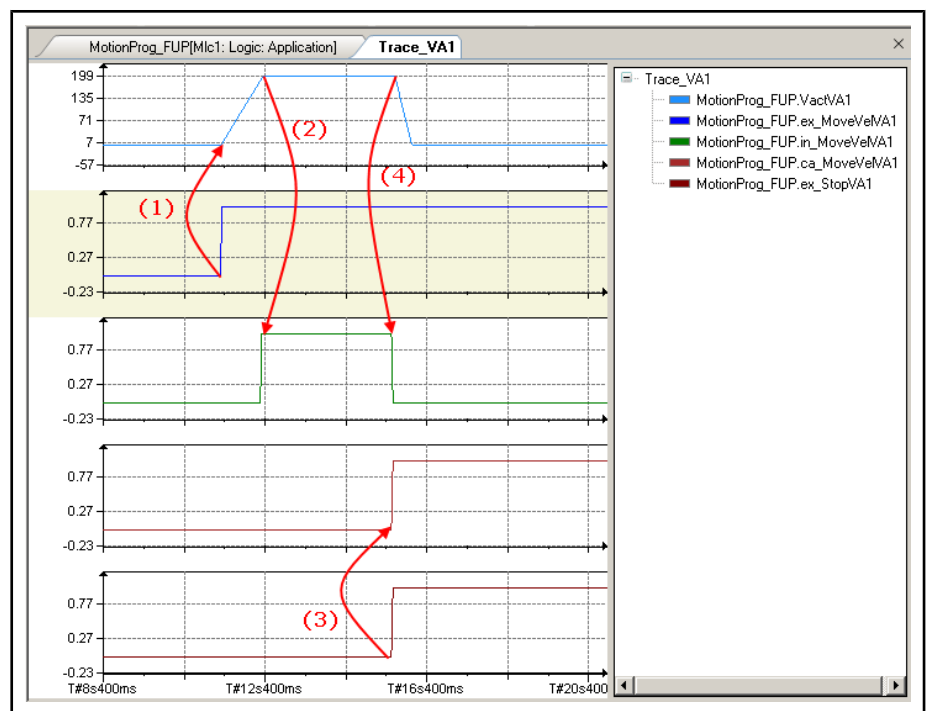


Abb.5-71: Traceaufzeichnung für die virtuelle Achse VA1

Auswertung der Trace-Aufzeichnung

- ① - Start des Hochlaufes der virtuellen Achse
- ② - Sollgeschwindigkeit erreicht
- ③ - Stoppen der virtuellen Achse
- ④ - Virtuelle Achse wird abgebremst.

Zu erkennen ist, dass sich die virtuelle Achse analog zu einer realen, gleichparametrierten Achse verhält. (Vergleiche Geschwindigkeitsverlauf, Beschleunigung mit 20 rad/s^2 und Verzögerung mit 50 rad/s^2).

5.5.5 Programmieren der realen Achse RA1 (ST)



Ab dieser Stelle benötigen Sie die "reale" Antriebstechnik oder

Sie verwenden ersatzweise eine zweite virtuelle Achse **ohne** MC_Power!!

Die Verwendung einer **realen** Achse setzt zunächst eine Instanz des PLCopen-Bausteins MC_Power voraus.

Als Nächstes werden jeweils eine Instanz von MC_GearIn und MC_GearOut benötigt.

Die reale Achse wird mit einer Instanz von MC_Stop angehalten.

Die Programmierung erfolgt analog zu [Programmieren der virtuellen Achse VA1](#), [Programmiersprache FUP](#), Seite 99.

MC_Power

Neue Zeile,

MC_Power einfügen und eine Instanz "PowerRA1" deklarieren

Eingänge beschalten und wenn nötig Eingangsvariablen deklarieren:

Erste Schritte - Motion

"Enable" → "en_PowerRA1" vom Typ BOOL

"Axis" → "MlcVarGlobal.RA1" vom Typ "AXIS_REF")

Ausgänge:

"Status" → "st_PowerRA1" vom Typ BOOL

"Error" → "er_PowerRA1" vom Typ BOOL

"ErrorID" → "ID_PowerRA1" vom Typ ERROR_CODE

"ErrorIdent" → "Ident_PowerRA1" vom Typ ERROR_STRUCT

MC_GearIn

Neue Zeile,

MC_GearIn einfügen und Instanz "GearInRA1" deklarieren

Eingänge beschalten und wenn nötig Eingangsvariablen deklarieren:

"Execute" → "ex_GearInRA1" vom Typ BOOL

"RatioNumerator" → 5

"RatioDenominator" → 10

"MasterFineadjust" → 20 (%)

"Master" → "MlcVarGlobal.VA1" vom Typ "AXIS_REF")

"Slave" → "MlcVarGlobal.RA1" vom Typ "AXIS_REF")

Ausgänge:

"InGear" → "ig_GearInRA1" vom Typ BOOL

"Active" → "ac_GearInRA1" vom Typ BOOL

"CommandAborted" → "ca_GearInRA1" vom Typ BOOL

"Error" → "er_GearInRA1" vom Typ BOOL

"ErrorID" → "ID_GearInRA1" vom Typ ERROR_CODE

"ErrorIdent" → "Ident_GearInRA1" vom Typ ERROR_STRUCT

MC_GearOut

Neue Zeile,

MC_GearOut einfügen und eine Instanz "GearOutRA1" deklarieren

Eingänge beschalten und wenn nötig Eingangsvariablen deklarieren:

"Execute" → "ex_GearOutRA1" vom Typ BOOL

"Slave" → "MlcVarGlobal.RA1" vom Typ "AXIS_REF")

Ausgänge:

"Done" → "do_GearOutRA1" vom Typ BOOL

"Active" → "ac_GearOutRA1" vom Typ BOOL

"Error" → "er_GearOutRA1" vom Typ BOOL

"ErrorID" → "ID_GearOutRA1" vom Typ ERROR_CODE

"ErrorIdent" → "Ident_GearOutRA1" vom Typ ERROR_STRUCT

MC_Stop

Neue Zeile,

MC_Stop einfügen und eine Instanz "StopRA1" deklarieren

Eingänge beschalten und wenn nötig Eingangsvariablen deklarieren:

"Execute" → "ex_StopRA1" vom Typ BOOL

"Deceleration" → 50 (rad/s²)

"Axis" → "MlcVarGlobal.RA1" vom Typ "AXIS_REF")

Ausgänge:

"Done" → "do_StopRA1" vom Typ BOOL

"Active" → "ac_StopRA1" vom Typ BOOL

"CommandAborted" → "ca_StopRA1" vom Typ BOOL

"Error" → "er_StopRA1" vom Typ BOOL

"ErrorID" → "ID_StopRA1" vom Typ ERROR_CODE

"ErrorIdent" → "Ident_StopRA1" vom Typ ERROR_STRUCT

Istgeschwindigkeit "VactRA1"

Als Letztes soll ein Netzwerk, in dem die aktuelle Geschwindigkeit der Achse RA1 "VactRA1" (Datentyp REAL) abgelesen werden kann, implementiert werden.

Dazu muss die aktuelle Geschwindigkeit aus dem Datensatz der Achse (Shared Memory) gelesen werden.

Die Daten aller Achsen, die eine MLC steuern kann, sind in dem Feld

GVL_Base.AxisData: ARRAY [1..xx⁴] OF ML_AXISDATA_SM

enthalten, wobei als Index die Achsnummer zu verwenden ist.

Die Variable wird unter dem Namen

GVL_Base.AxisData[1].rActualVelocity_i,

für die reale Achse, Achsnummer 1, bereitgestellt (Achsnummer 1 hat RA1, da das Scannen von RA1 vor der Deklaration von VA1 erfolgte).

- Neue Zeile anfügen,
- VactRA1 := einfügen, Deklaration komplettieren,
- unter Nutzung der Eingabehilfe zu **Globale Variablen ▶ ML_BASE ▶ Global Variables List ▶ GVL_Base ▶ AxisData** navigieren:

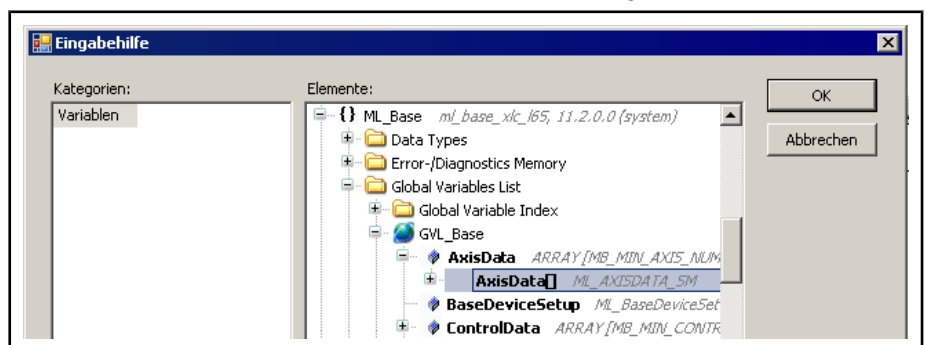


Abb.5-72: Geschwindigkeitswert VA1 aus der Globalen Variablenliste der Bibliothek ML_Base.library

⁴) Die Anzahl der steuerbaren Achsen ist vom Typ der Steuerung und ihrer Firmware abhängig, aktuell für L65: 64 Achsen

Erste Schritte - Motion

Die weitere Eingabe wird durch die Intellisense-Funktion unterstützt. Dies bedeutet, dass nach der Eingabe des "." bei "GVL_Base.AxisData[1]." alle verfügbaren Elemente aufgerollt werden und zur Auswahl stehen.

"rActualVelocity_i" mit <Enter> bestätigen.

- Abschluss der Zeile mit ";" nicht vergessen.

```

25 PowerRA1( Enable:= en_PowerRA1,
26     Axis:= MlcVarGlobal.RA1,
27     Status=> st_PowerRA1,
28     Error=> er_PowerRA1,
29     ErrorID=> ID_PowerRA1,
30     ErrorIdent=> Ident_PowerRA1);
31
32 GearInRA1( Execute:= ex_GearInRA1,
33     RatioNumerator:= 5,
34     RatioDenominator:= 10,
35     MasterFineadjust:= 20,
36     Master:= MlcVarGlobal.VA1,
37     Slave:= MlcVarGlobal.RA1,
38     InGear=> ig_GearInRA1,
39     Active=> ac_GearInRA1,
40     CommandAborted=> ca_GearInRA1,
41     Error=> er_GearInRA1,
42     ErrorID=> ID_GearInRA1,
43     ErrorIdent=> Ident_GearInRA1);
44
45 GearOutRA1( Execute:= ex_GearOut,
46     Slave:= MlcVarGlobal.RA1,
47     Done=> do_GearOut,
48     Active=> ac_GearOut,
49     Error=> er_GearOut,
50     ErrorID=> ID_GearOut,
51     ErrorIdent=> Ident_GearOut);
52
53 StopRA1( Execute:= ex_StopRA1,
54     Deceleration:= 50,
55     Axis:= MlcVarGlobal.RA1,
56     Done=> do_StopRA1,
57     Active=> ac_StopRA1,
58     CommandAborted=> ca_StopRA1,
59     Error=> er_StopRA1,
60     ErrorID=> ID_StopRA1,
61     ErrorIdent=> Ident_StopRA1);
62
63 VactRA1:= GVL_Base.AxisData[1].rActualVelocity_i;
```

Abb.5-73: Implementation der realen Achse

5.5.6 Download des kompletten SPS-Projektes (ST)

Der Ladevorgang verläuft analog zum Abschnitt [Download des SPS-Teilprojektes virtuelle Achse VA1 \(ST\)](#), Seite 129.

Das neue Projekt überschreibt dabei das alte nach Rückfrage.

5.5.7 Workflow "Virtuelle Achse → Elektronisches Getriebe → Reale Achse"

Der nachstehende Arbeitsablauf zeigt das Starten und Stoppen der Achsen und des Getriebes.

1. Leistungszuschaltung für die reale Achse
Zuschalten des "en_PowerRA1" (MC_Power) durch "Werte schreiben".
st_PowerRA1= TRUE (MC_Power).
2. Start der virtuellen Achse
Zuschalten des "ex_MoveVelVA1" (MC_MoveVelocity) durch "Werte schreiben".
Hochlauf bei VactVA1 = 200 U/min beendet.
3. Nachführen der realen Achse
Zuschalten des "ex_GearInRA1" (MC_GearIn) durch "Werte schreiben".
Das zu erwartenden Übersetzungsverhältnis des Getriebes ergibt sich zu:
$$\dot{U}_v = (\text{RatioNumerator} / \text{RatioDenominator}) * (1 + \text{MasterFineadjust})$$

also bei VactVA1 = 200 U/min ergibt sich VactRA1 = 120 U/min.



Der MasterFineadjust-Wert wird in "%" vorgegeben.

4. Befindet sich der MC_GearIn "InGear", so kann die Geschwindigkeit "VcomVA1" der virtuellen Achse durch "Werte schreiben" (Doppelklick auf die Variable, gewünschten Wert eintragen) verändert werden, die reale Achse folgt ihr.
5. Abschalten des "ex_GearInRA1" (MC_GearIn) durch "Werte schreiben".
in_GearInRA1= FALSE, ac_GearInRA1= TRUE.
Die reale Achse folgt der virtuellen weiterhin.
6. Auflösen der Synchronität
Zuschalten des "ex_GearOutRA1" (MC_GearOut) durch "Werte schreiben".
Die Abhängigkeit der realen Achse von der virtuellen ist aufgehoben. Beide Achsen drehen jedoch mit unveränderter Geschwindigkeit weiter.
7. Änderung der Sollgeschwindigkeit "VcomVA1"
Die veränderte Sollgeschwindigkeit wirkt sich nur noch auf die virtuelle Achse aus, "VactRA1" bleibt unverändert.
8. Abschalten des "ex_GearOutRA1" durch "Werte schreiben".
do_GearOutRA1= FALSE
9. Stoppen der virtuellen Achse
Zuschalten des "ex_StopVA1" (MC_Stop)
Die virtuelle Achse wird abgebremst auf VactVA1= 0 U/min.
10. Abschalten des "ex_StopVA1" (MC_Stop).
11. Stoppen der realen Achse
Zuschalten des "ex_StopRA1" (MC_Stop).

Erste Schritte - Motion

Die reale Achse wird abgebremst auf VactRA1= 0 U/min. Sie wird nach Erreichen des Stillstandes aktiv in dieser Position gehalten.

12. Abschalten des "ex_StopRA1" (MC_Stop).

Das aktive Halten der realen Achse ist beendet.

13. Leistungsabschaltung für die reale Achse

Abschalten von "en_PowerRA1" (MC_Power) durch "Werte schreiben".



Nur möglich im Zustand Standstill nach Ausführung des Stops an RA1.

5.5.8 Aufzeichnung der Bewegung beider Achsen (Trace), ST

Das Trace-Werkzeug wurde analog zum Abschnitt [Aufzeichnung der Bewegung von Achse VA1 \(Trace\), ST, Seite 134](#), verwendet, um den Hochlauf der virtuellen Achse und das Aufsynchronisieren sichtbar zu machen.

Der nachstehende Arbeitsablauf zeigt das Starten und Stoppen der Achsen und des Getriebes für die Trace-Aufzeichnung.

1. Leistungszuschaltung für die reale Achse
Zuschalten des "en_PowerRA1" (MC_Power).
2. Start der virtuellen Achse
Zuschalten des "ex_MoveVelVA1" (MC_MoveVelocity).
3. Nachführen der realen Achse
Zuschalten des "ex_GearInRA1" (MC_GearIn).
4. Drehzahlsollwertsprung der virtuellen Achse
Erhöhen des Drehzahlsollwertes
Wirksamschalten durch "ex_MoveVelVA1" TRUE ⇒ FALSE ⇒ TRUE.

Die nachstehende Abbildung zeigt das Trace-Ergebnis:

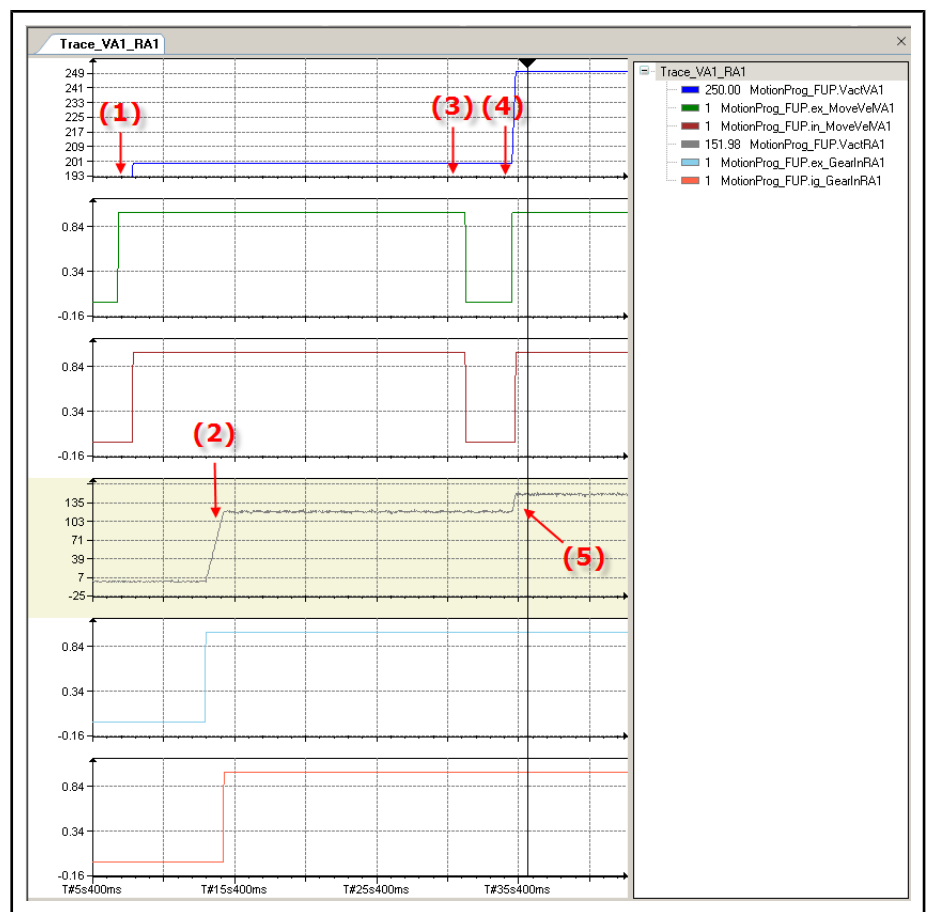


Abb.5-74: Traceaufzeichnung für beide Achsen

Auswertung der Trace-Aufzeichnung

- ① - Start des Hochlaufes der virtuellen Achse,
- ② - Zuschalten der Getriebe-Funktion, Hochlauf der realen Achse,
- ③ - Erhöhen der Solldrehzahl der virtuellen Achse, "ex_MoveVelVA1" TRUE $\Rightarrow$ FALSE,
- ④ - Wirksamwerden der Solldrehzahl der virtuellen Achse, "ex_MoveVelVA1" FALSE $\Rightarrow$ TRUE,
- ⑤ - Sprungantwort an der realen Achse.

5.6 Projekt wiederherstellen

Ablage Ausgehend vom erstellten Projekt wurde eine Exportdatei erzeugt, die es unabhängig von der Steuerungshardware z.B. XLC L25, L45, L65 (mit SERCOS III-Bus) oder MLC L25, L45, L65 und der Antriebstechnik erlaubt, Teile des Projektes wieder zu regenerieren.

Die Exportdatei wurde vom Knoten "Application" beginnen erstellt und enthält folgende Bestandteile:

Erste Schritte - Motion

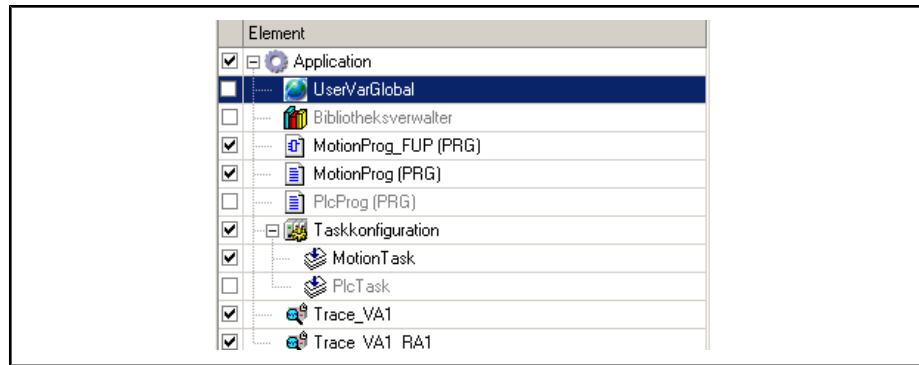


Abb.5-75: Bestandteile der Exportdatei

Die angegebenen Komponenten werden beim Import überschrieben.



Die Exportdatei kann als Grundlage für "Erste Schritte FlexProfile" genutzt werden.

Wiederherstellen

Um das Projekt auf einer Steuerung Ihrer Wahl z.B. XLC L25, L45, L65 (mit SERCOS III-Bus) oder MLC L25, L45, L65 lauffähig zu machen, ist etwas Eigenbeteiligung erforderlich:

1. Löschen von Altdaten in Ihrer Steuerung
Löschen Sie den alten Speicherinhalt Ihrer Steuerung.
Details sind unter [Steuerung zurücksetzen \(CMLx5\)](#), Seite 88, beschrieben.
Der Speicher der Steuerung ist geleert, eine eventuell vorhandene Motion-Komponente und die SPS-Komponente sind betriebsbereit.
2. Anlage eines neuen Projektes
Legen Sie ein neues leeres Projekt an. **Datei ► Neu ► Projekt**. Der Projektname ist beliebig, empfohlen wird "Erste Schritte Motion".
Sie verfügen über ein neues leeres Projekt.
3. Anlegen der Steuerung
Ziehen Sie eine Steuerung Ihrer Wahl z.B. XLC L25, L45, L65 (mit SERCOS III-Bus) oder MLC L25, L45, L65 aus der Gerätebibliothek auf das leere Projekt. Der Steuerungsname ist beliebig, empfohlen wird "DCC_Control".
Komplettieren Sie den Gerätetyp und die IP-Adresse.
Führen Sie einen Verbindungstest durch.
Schließen Sie den Einfügevorgang wie gewohnt ab.
In Ihrem Projekt befindet sich die gewünschte Steuerung.
4. Anlegen des Motion-Teilprojektes
Scannen Sie vom SERCOS III-Knoten aus den SERCOS-Bus.
Dem Antrieb wird die Achsnummer "1" zugewiesen, er ist umzubenennen und erhält den Namen "RA1".
5. Vereinbaren Sie eine virtuelle Achse, sie erhält die Achsnummer "2" und den Namen "VA1".

Erste Schritte - Motion

In der Globalen Variablenliste "MlcVarGlobal" werden die Achsnummern und Achsnamen als Achsreferenzen vom Typ "AXIS_REF" zur Verfügung gestellt.

Das [Kap. 5.3.8 "Download des Motion-Teilprojektes" auf Seite 96](#) kann stattfinden.

6. Import der Exportdatei "Application_FirstSteps_Motion.iwx"

Öffnen Sie im Projektnavigator den Steuerungsknoten bis **Logic ► Application**.

Kontextmenü "Application", **Import**.

Wählen Sie die Exportdatei "Application_FirstSteps_Motion.iwx" in "Iw: \Programme\Rexroth\IndraWorks\Projects" aus, Operation "Überschreiben", fertigstellen.

Die benötigten Dateien wurden kopiert, das Projekt ist auf der Steuerung lauffähig.

6 Erste Schritte - FlexProfile

6.1 Ziel des Projekts "Erste Schritte - FlexProfile"

Vorgegeben wird eine virtuelle Achse, die mit einer konstanten Geschwindigkeit laufen soll. Diese Achse arbeitet als Leitachse für eine reale Achse, die über ein Bewegungsprofil "FlexProfile" gekoppelt ist.

Die reale Achse enthält z.B. einen Antrieb "IndraDrive M", der über ein SERCOS III-Kabel mit der Steuerung verbunden ist.

Das Bewegungsprofil besteht aus einem Bewegungsschritt der eine einfache Rast-in-Rast-Bewegung realisiert. Auf einer Leitachsumdrehung von 360° soll dabei ein Folgeachshub von 180° verfahren werden. Nachfolgendes Bild zeigt das zugehörige Bewegungsdiagramm.

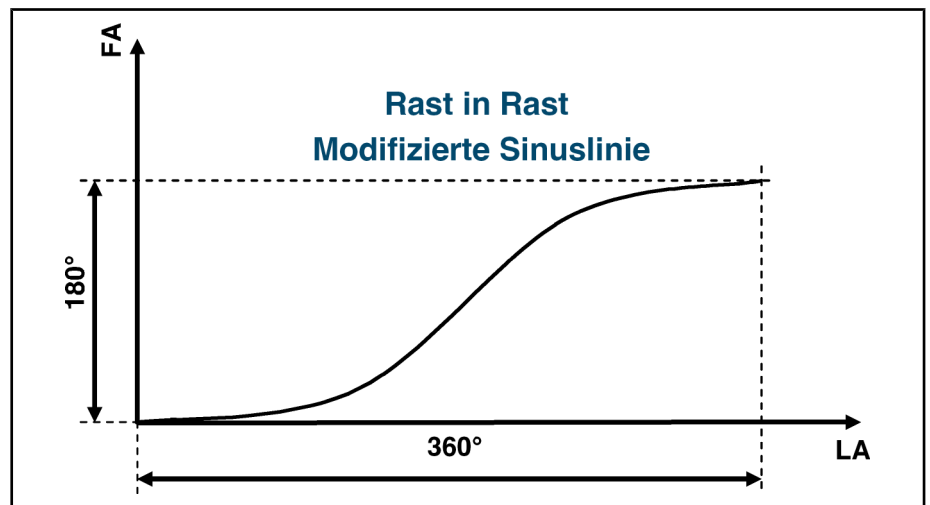


Abb. 6-1: Kopplungsprofil zwischen der Leit- (VA1) und Folgeachse (RA1)

Der Folgeachse können dabei bis zu vier Bewegungsprofile mit jeweils maximal 16 (64) Schritten¹⁾.

Wir nutzen für die "Ersten Schritte - FlexProfile" das FlexProfile 0 in RA1.

Ein FlexProfile kann auf zwei unterschiedlichen Wegen erstellt werden:

- IndraWorks, Dialog "FlexProfile" der Folgeachse, siehe [Erstellen des FlexProfiles über IndraWorks-Dialoge, Seite 151](#),
oder
- IndraLogic, SPS-Programmcode mit einem Funktionsbaustein MB_ChangeFlexProfileSet zur Datenübergabe vom SPS-Programm an die Folgeachse, siehe [Erstellen des FlexProfiles über die SPS, Seite 154](#).

Programmierung in ST siehe [FlexProfile, Datenübergabe in ST, Seite 154](#),

Programmierung in FUP siehe [FlexProfile, Datenübergabe in FUP, Seite 156](#).

Die eigentliche Abarbeitung wird über den Baustein ML_FlexProfile gestartet. (ML_FlexProfile übernimmt damit die Aufgabe, die MC_GearIn in den "Ersten Schritten - Motion" erfüllt hat, die Anbindung der Folgeachse an die Leitachse. Der Baustein ML_FlexProfile wird durch den Baustein MC_GearOut deaktiviert.

- Programmierung in ST siehe [ML_FlexProfile in ST, Seite 158](#),

¹⁾ Die Anzahl der Schritte innerhalb eines FlexProfiles hängt von der Firmwareversion und vom Steuerungstyp ab.

Erste Schritte - FlexProfile

- Programmierung in FUP siehe [ML_FlexProfile in FUP, Seite 160](#).

6.2 Technische Voraussetzungen

- PC mit installiertem IndraWorks Engineering für IndraMotion MLC, 12VRS,
Steuerung, Typ "IndraMotion MLC L25", "IndraMotion MLC L45" oder "IndraMotion MLC L65",
aktuelle Firmware ist geladen.
- IP-Adresse ist bekannt
- Steuerung ist über Ethernet mit dem Programmiergerät verbunden.
Antriebsregelgerät, z.B. Typ "IndraDrive M", aktuelle Firmware ist geladen
- Das Antriebsregelgerät ist mit der Steuerung über einen SERCOS-Bus verbunden.
- Die Adresse des Antriebsregelgerätes am SERCOS-Bus ist bekannt oder ein gebrückter SERCOS-Bus, wenn kein Antrieb vorhanden ist.



Weitere Hinweise zu FlexProfile siehe Dokumentation

"Rexroth IndraMotion MLC 12VRS Funktionsbeschreibung" Kapitel 8.7 "Flexprofile"

(DOK-IM*MLC-FUNC****V12-RExx-DE-P).

FlexProfile-Funktionalität wird von folgenden Achstypen (Folgeachse) unterstützt:

- Virtuelle Achse,
- SercosDrive,
- IndraDrive mit Interpolation in der Steuerung.

Verwenden Sie bitte als Leitachse die virtuelle Achse VA1 und als Folgeachse die reale Achse RA1, IndraDrive **mit Interpolation in der Steuerung**.

6.3 Erstellen des Motion-Teilprojektes

Um das Projekt auf einer Steuerung Ihrer Wahl z.B. IndraLogic XLC L25, L45, L65 (mit SERCOS III-Bus) oder IndraMotion MLC L25, L45, L65 lauffähig zu machen, ist etwas Eigenbeteiligung erforderlich:

1. Löschen von Altdaten in Ihrer Steuerung

Löschen Sie den alten Speicherinhalt Ihrer Steuerung.

Details sind unter [Steuerung zurücksetzen \(CMLx5\), Seite 88](#), beschrieben.

Der Speicher der Steuerung ist geleert, eine eventuell vorhandene Motion-Komponente und die SPS-Komponente sind betriebsbereit.

2. Anlage eines neuen Projektes

Legen Sie ein neues leeres Projekt an. **Datei ► Neu ► Projekt**. Der Projektname ist beliebig, empfohlen wird "Erste Schritte FlexProfile".

Sie verfügen über ein neues leeres Projekt.

3. Anlegen der Steuerung

Ziehen Sie eine Steuerung Ihrer Wahl z.B. IndraLogic XLC L25, L45, L65 (mit SERCOS III-Bus) oder IndraMotion MLC L25, L45, L65 aus der Ge-

Erste Schritte - FlexProfile

rätebibliothek auf das leere Projekt. Der Steuerungsname ist beliebig, empfohlen wird "DCC_Control".

Komplettieren Sie den Gerätetyp und die IP-Adresse.

Führen Sie einen Verbindungstest durch.

Schließen Sie den Einfügevorgang wie gewohnt ab.

In Ihrem Projekt befindet sich die gewünschte Steuerung.

4. Anlegen des Motion-Teilprojektes

Scannen Sie vom SERCOS III-Knoten aus den SERCOS-Bus.



Wählen Sie "Abarbeitung" in der Steuerung!

Dem Antrieb wird die Achsnummer "1" zugewiesen, er ist umzubenennen und erhält den Namen "RA1".

5. Vereinbaren Sie eine virtuelle Achse, sie erhält die Achsnummer "2" und den Namen "VA1".

In der Globalen Variablenliste "MlcVarGlobal" werden die Achsnummern und Achsnamen als Achsreferenzen vom Typ "AXIS_REF" zur Verfügung gestellt.

Das [Download des Motionprojektes, Seite 96](#), kann stattfinden. (Die Motion-Komponente befindet sich dann im Zustand "Betriebsbereit (BB)").

6. Import der Exportdatei "Application_FirstSteps_Motion.iwx"

Öffnen Sie im Projektnavigator den Steuerungsknoten bis **Logic ► Application**.

Kontextmenü "Application", **Import**.

Wählen Sie die Exportdatei "Application_FirstSteps_Motion.iwx" in "Iw: \Programme\Rexroth\IndraWorks\Projects" aus, Operation "Überschreiben", fertigstellen.

Die benötigten Dateien wurden kopiert, das Projekt ist auf der Steuerung lauffähig.

6.4 Erstellen eines FlexProfiles

6.4.1 Erstellen eines FlexProfiles, Allgemeines

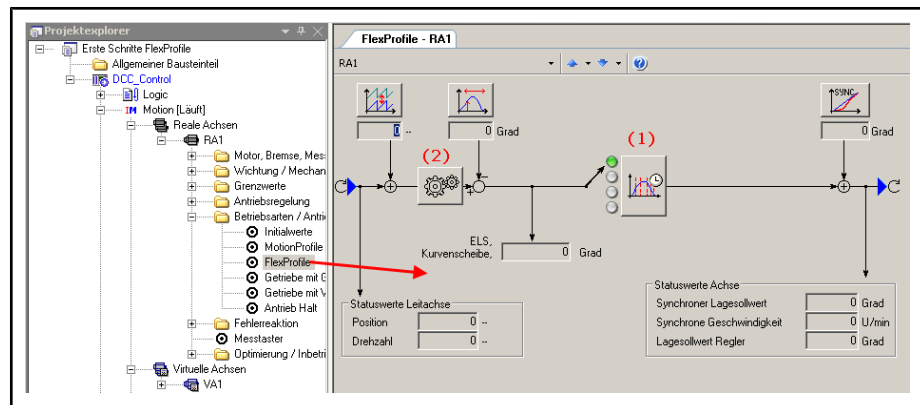


Das Erstellen oder Modifizieren eines FlexProfiles setzt voraus, dass Steuerung und PC **online** gekoppelt sind.

6.4.2 Erstellen des FlexProfiles über IndraWorks-Dialoge

Durch Anklicken des Eintrags "FlexProfile" im Projektexplorer Folgeachse (RA1) öffnet sich ein Dialog, in dem die allgemeine Profilkonfiguration vorgenommen werden kann.

Erste Schritte - FlexProfile



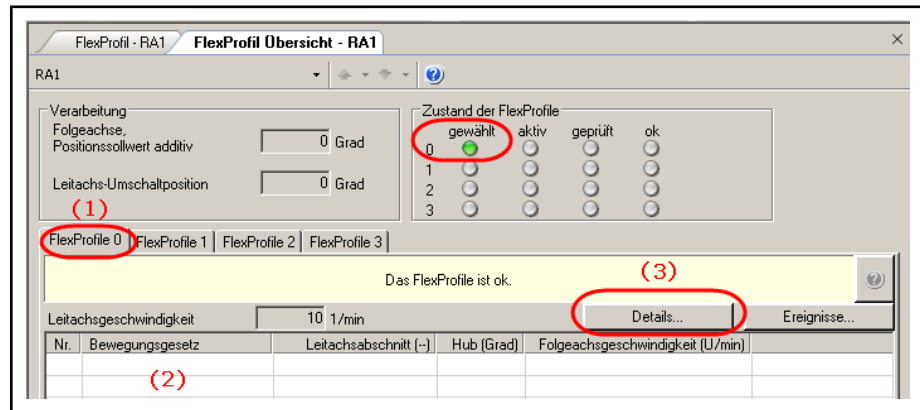
(1) Übergang zu den Dialogfenstern "FlexProfile 0"... "FlexProfile 3"

(2) Übergang zum Dialogfenster "Getriebeeinstellung"

Abb.6-2: Dialog FlexProfile mit Wirkungskette zwischen Leit- und Folgeachse

Öffnen Sie die FlexProfile-Übersicht per Mausklick auf (1).

Das nachfolgende Dialogfenster zeigt den Verarbeitungsstatus für die vier der Achse zuordenbaren FlexProfile, Register "FlexProfile 0" bis "FlexProfile 3".



(1) Register "FlexProfile 0" bis "FlexProfile 3", gewählt "FlexProfile 0"

(2) Übersicht über die Bewegungsschritte 0...15 (63) des gewählten Flex-Profils (schreibgeschützt)

(3) Übergang zu den Details des gewählten FlexProfils

Abb.6-3: Dialog: FlexProfile Übersichtsfenster, gewählt wird FlexProfile 0

Die Erstellung eines neuen Bewegungsprofils kann über "Details" (1) vorgenommen werden.

Dazu wird automatisch der FlexProfile-Editor geöffnet.

Es wird eine leere Konfigurationstabelle gezeigt, in der jede Zeile einen Profilschritt (0..15 (63)) definiert.

Die dazu benötigten Leitachsquellen, Bewegungsgesetze, Bewegungsschritt-Typen können aus einem Dropdown-Menü ausgewählt werden.

Tragen Sie diese, wie auf unten gezeigtem Bild ein, und konfigurieren Sie so den gewünschten Bewegungsverlauf.

Erste Schritte - FlexProfile

FlexProfile - RA1 FlexProfile Übersicht - RA1 Details FlexProfile 0 - RA1

RA1

Das FlexProfile ist ok.

Leitachsgeschwindigkeit 10 1/min

übernehmen und prüfen

Nr	Bezug	Hub	Leitachsabschnitt	Einheit	Bewegungsgesetz	Schrittart	Folgeachs anfangsgeschwindigkeiten	Folgeachs anfangsbeschleunigungen	Folgeachs anfangsdrücke	Folgeachs endgeschwindigkeiten	Folgeachs endbeschleunigungen	Folgeachs enddrücke	Folgeachs verfahrensgeschwindigkeiten	Folgeachs verfahrensbeschleunigungen	Folgeachs verfahrensdrücke	Wendepunktverschiebungen
							U/min	rad/s ²	rad/s ³	U/min	rad/s ²	rad/s ³	U/min	rad/s ²	rad/s ³	
0	Achse	180	360	--	Rast in Rast, modifizierte Sinuslinie	Standard Schritt	0	0	0	0	0	0	0	0	0	0,5
1					(1)											

- (1) Konfiguration der einzelnen Bewegungsschritte (0...15 (63)) des gewählten FlexProfiles
- (2) Übergang, übernehmen der Daten des Bewegungsprofils und Prüfung, ob das Profil korrekt ist

Abb.6-4: Dialog: FlexProfile Konfigurationsfenster

Für das FlexProfile 0 wurden in obigem Dialogfenster nachstehende Daten eingetragen:

Bezug	Achse
Hub	180 Grad
Leitachsabschnitt	360
Einheit	-
Bewegungsgesetz	Rast in Rast, modifizierte Sinuslinie
Schritt-Art	Standard-Schritt
... weitere ...	0
Wendepunktverschiebung	0,5

Nach beendeter FlexProfile-Konfiguration müssen die Änderungen in die Steuerung übertragen werden. Diese wird durch die Betätigung von "Übernehmen und Prüfen" (2) ausgelöst.



Ein Bewegungsprofil kann nur gestartet werden, wenn dieses geprüft und als gültig markiert wurde.

Eine Veränderung der Konfiguration setzt den "geprüften" Status zurück.

Ein Bewegungsprofil, das gegenwärtig aktiv ist, kann nicht verändert werden.

Im FlexProfile-Übersichtsfenster wird nach erfolgreicher Übernahme der Zustand des Bewegungsprofils nun als "geprüft" und "ok" angezeigt.

Erste Schritte - FlexProfile

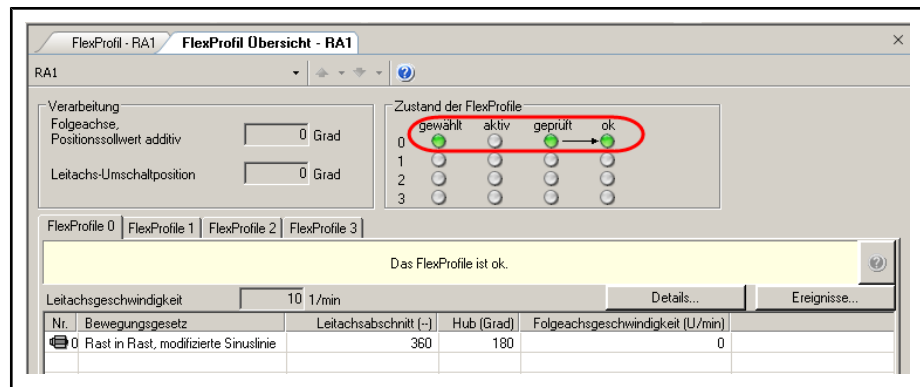


Abb.6-5: Dialog: FlexProfile Übersichtsfenster, FlexProfile 0, geprüft → ok

Das so konfigurierte Bewegungsprofil "FlexProfile 0" kann nun mit dem SPS-Baustein "ML_FlexProfile" gestartet werden.

6.4.3 Erstellen des FlexProfiles über die SPS

Das Konfigurieren eines FlexProfiles soll nun alternativ über die SPS erfolgen.



Bitte beachten Sie, dass die Ausführungszeit des Bausteins "MB_ChangeFlexProfileSet" je nach Größe und Komplexität eines zu schreibenden Profils etwas länger dauern kann.

Aus diesem Grund sollte der Baustein nur in frei laufenden SPS-Prozessen verwendet werden.

Die Programmierung erfolgt aus diesem Grund für die "Ersten Schritte - FlexProfile" in den Programmen [PlcProg, Seite 154](#), bzw. [PlcProg_FUP, Seite 156](#). Beide Programme werden alternativ über die PlcTask gesteuert.

Der Einsatz innerhalb einer SERCOS-synchronen Task (z.B. "MotionTask") wird bei realen Programmen nicht empfohlen.

Zu lösende Aufgaben

1. Bereitstellen der Daten für die Bewegungsschritte 0...15 (63) der vier FlexProfiles "FlexProfile 0"... "FlexProfile 3".

```
profileAxis: ARRAY[0..15] OF MB_PROFILE_STEP;
```

Das ARRAY "profileAxis" enthält damit die Daten eines FlexProfiles.

Für die "Ersten Schritten - FlexProfile" ist somit der Datensatz "0" vom Typ "MB_PROFILE_STEP" in unserem Programm zu belegen.

Dies entspricht [Abb. 6-4 "Dialog: FlexProfile Konfigurationsfenster"](#) auf [Seite 153](#).

2. Übergabe der Daten des FlexProfiles aus dem SPS-Programm an die Steuerung (Achse RA1).

Diese Aufgabe übernimmt die Instanz "ChangeFlexProfileSetRA1" vom Typ "MB_ChangeFlexProfileSet".

Dies entspricht [Abb. 6-5 "Dialog: FlexProfile Übersichtsfenster, FlexProfile 0, geprüft → ok"](#) auf [Seite 154](#).

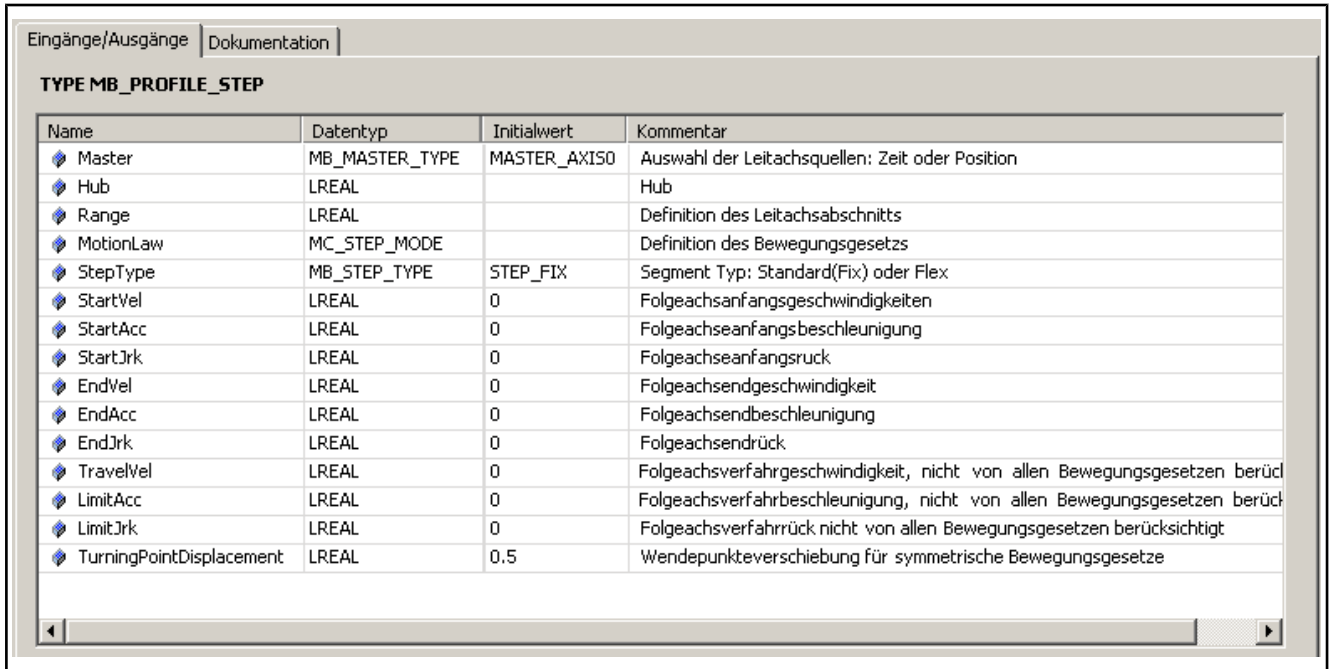
FlexProfile, Datenübergabe in ST

Bei der Bereitstellung des Datensatzes wird zunächst die Variable "profileAxis" deklariert.

Danach werden ihre Strukturelemente mit den gewünschten Werten belegt.

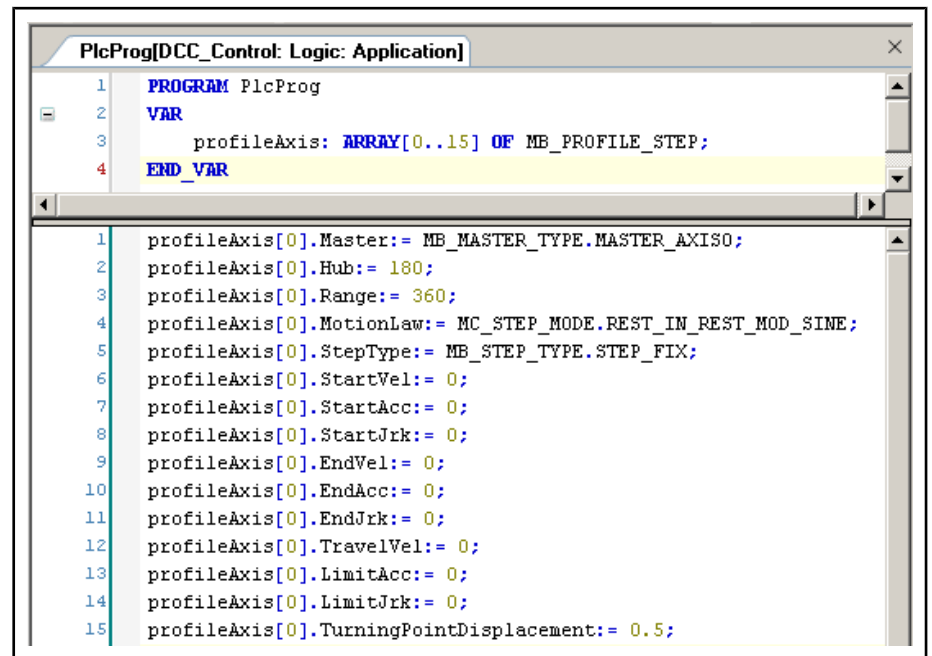
Erste Schritte - FlexProfile

Die Auflistung der Strukturelemente kann dem Bibliotheksverwalter entnommen werden **ML_Base_xxx_yyy ▶ Data Types ▶ Flex Structs ▶ MB_PROFILE_STEP²⁾**.



Name	Datentyp	Initialwert	Kommentar
Master	MB_MASTER_TYPE	MASTER_AXIS0	Auswahl der Leitachsquellen: Zeit oder Position
Hub	LREAL		Hub
Range	LREAL		Definition des Leitachsabschnitts
MotionLaw	MC_STEP_MODE		Definition des Bewegungsgesetzes
StepType	MB_STEP_TYPE	STEP_FIX	Segment Typ: Standard(Fix) oder Flex
StartVel	LREAL	0	Folgeachsanzfangsgeschwindigkeiten
StartAcc	LREAL	0	Folgeachseanfngsbeschleunigung
StartJrk	LREAL	0	Folgeachseanfngsruck
EndVel	LREAL	0	Folgeachsendgeschwindigkeit
EndAcc	LREAL	0	Folgeachsendbeschleunigung
EndJrk	LREAL	0	Folgeachsendruck
TravelVel	LREAL	0	Folgeachsverfahrensgeschwindigkeit, nicht von allen Bewegungsgesetzen berück
LimitAcc	LREAL	0	Folgeachsverfahrensbeschleunigung, nicht von allen Bewegungsgesetzen berück
LimitJrk	LREAL	0	Folgeachsverfahrensruck nicht von allen Bewegungsgesetzen berücksichtigt
TurningPointDisplacement	LREAL	0.5	Wendepunkteverschiebung für symmetrische Bewegungsgesetze

Abb.6-6: Beschreibung der Struktur MB_PROFILE_STEP in Bibliotheksverwalter
Nutzen Sie zur Eingabe den "."-Operator mit nachfolgender Intellisense-Funktion.



```

PROGRAM PlcProg
VAR
    profileAxis: ARRAY[0..15] OF MB_PROFILE_STEP;
END_VAR

profileAxis[0].Master:= MB_MASTER_TYPE.MASTER_AXIS0;
profileAxis[0].Hub:= 180;
profileAxis[0].Range:= 360;
profileAxis[0].MotionLaw:= MC_STEP_MODE.REST_IN_REST_MOD_SINE;
profileAxis[0].StepType:= MB_STEP_TYPE.STEP_FIX;
profileAxis[0].StartVel:= 0;
profileAxis[0].StartAcc:= 0;
profileAxis[0].StartJrk:= 0;
profileAxis[0].EndVel:= 0;
profileAxis[0].EndAcc:= 0;
profileAxis[0].EndJrk:= 0;
profileAxis[0].TravelVel:= 0;
profileAxis[0].LimitAcc:= 0;
profileAxis[0].LimitJrk:= 0;
profileAxis[0].TurningPointDisplacement:= 0.5;
    
```

Abb.6-7: Datensatz "profileAxis" mit Wertezuordnung in ST

Ab Zeile 17 wird in gewohnter Weise die Instanz "ChangeFlexProfileSetRA1" des Funktionsbausteins "MB_ChangeFlexProfileSet" zur Übergabe des Datenfeldes an die Steuerung eingegeben.

²⁾ ML_Base_xxx_yyy: xxx steht für XLC oder MLC; yyy steht für L25, L45 oder L65

Erste Schritte - FlexProfile

```

15  profileAxis[0].TurningPointDisplacement:= 0.5;
16
17  ChangeFlexProfileSetRA1(
18      Execute:= ex_ChangeFlexProfileSetRA1,
19      SetNumber:= SetNum,
20      NumberOfSteps:= 1,
21      MasterAxisVelocity:= 10,
22      ProfileStepAdr:= ADR(profileAxis),
23      Axis:= MlcVarGlobal.RA1,
24      Done=> do_ChangeFlexProfileSetRA1,
25      Active=> ac_ChangeFlexProfileSetRA1,
26      Error=> er_ChangeFlexProfileSetRA1,
27      ErrorID=> ID_ChangeFlexProfileSetRA1,
28      ErrorIdent=> Ident_ChangeFlexProfileSetRA1);

```

Abb.6-8: Funktionsbaustein zur Übergabe des Datenfeldes an die Steuerung (SetNum = 0 für FlexProfile 0)

Die Übergabe des Datensatzes erfolgt durch "Werte schreiben" am Eingang "ex_ChangeFlexProfileSetRA1", TRUE/FALSE.

Der Zustand des FlexProfiles ist danach geprüft/ok.



Verwenden Sie SetNum= 2. Sie schreiben damit auf "FlexProfile 2" und können das Ergebnis mit "FlexProfile 0", Eingabe über den IndraWorks-Dialog, vergleichen.

FlexProfile, Datenübergabe in FUP

Bei der Bereitstellung des Datensatzes wird zunächst die Variable "profileAxis" deklariert.

Danach werden ihre Strukturelemente mit den gewünschten Werten belegt.

Die Auflistung der Strukturelemente kann dem Bibliotheksverwalter entnommen werden **ML_Base_xxx_yyy ▶ Data Types ▶ Flex Structs ▶ MB_PROFILE_STEP³⁾**.

Eingänge/Ausgänge Dokumentation			
TYPE MB_PROFILE_STEP			
Name	Datentyp	Initialwert	Kommentar
Master	MB_MASTER_TYPE	MASTER_AXIS0	Auswahl der Leitachsquellen: Zeit oder Position
Hub	LREAL		Hub
Range	LREAL		Definition des Leitachsabschnitts
MotionLaw	MC_STEP_MODE		Definition des Bewegungsgesetzes
StepType	MB_STEP_TYPE	STEP_FIX	Segment Typ: Standard(Fix) oder Flex
StartVel	LREAL	0	Folgeachsanzugsgeschwindigkeiten
StartAcc	LREAL	0	Folgeachsanzugsbeschleunigung
StartJrk	LREAL	0	Folgeachsanzugsruck
EndVel	LREAL	0	Folgeachsendgeschwindigkeit
EndAcc	LREAL	0	Folgeachsendbeschleunigung
EndJrk	LREAL	0	Folgeachsendruck
TravelVel	LREAL	0	Folgeachsverfahrgeschwindigkeit, nicht von allen Bewegungsgesetzen berüch
LimitAcc	LREAL	0	Folgeachsverfahrbeschleunigung, nicht von allen Bewegungsgesetzen berüch
LimitJrk	LREAL	0	Folgeachsverfahrdruck nicht von allen Bewegungsgesetzen berücksichtigt
TurningPointDisplacement	LREAL	0.5	Wendepunkteverschiebung für symmetrische Bewegungsgesetze

Abb.6-9: Beschreibung der Struktur MB_PROFILE_STEP in Bibliotheksverwalter

³⁾ ML_Base_xxx_yyy: xxx steht für XLC oder MLC; yyy steht für L25, L45 oder L65

Erste Schritte - FlexProfile

Nutzen Sie zur Eingabe der Zuweisungen den "."-Operator mit nachfolgender Intellisense-Funktion.

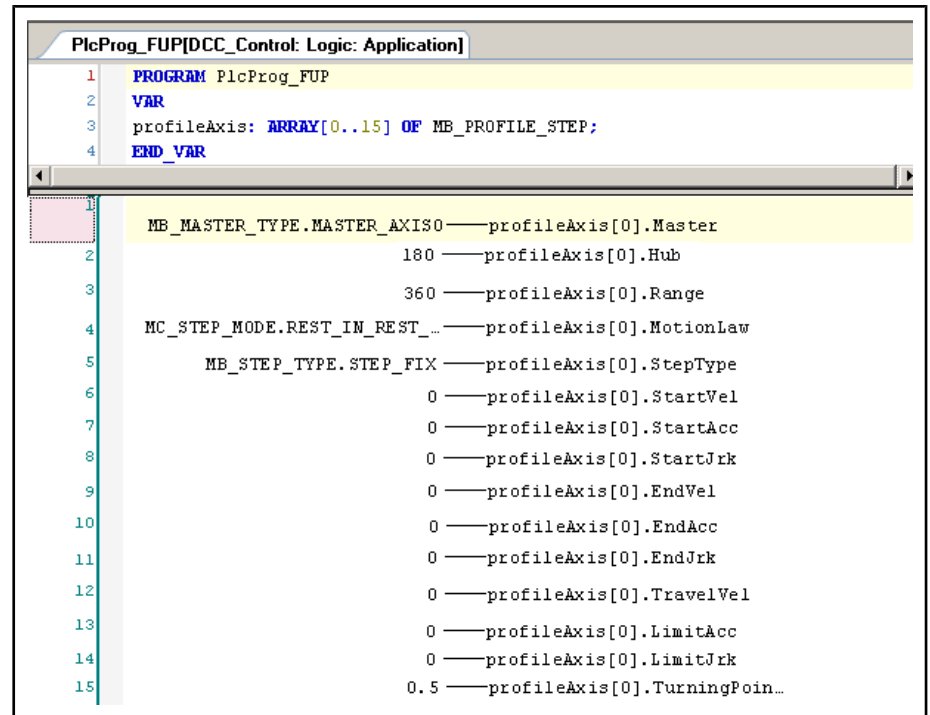


Abb.6-10: Datensatz "profileAxis" mit Wertezuordnung in FUP

Als Netzwerk Nummer 16 wird in gewohnter Weise die Instanz "ChangeFlexProfileSetRA1" des Funktionsbausteins "MB_ChangeFlexProfileSet" zur Übergabe des Datenfeldes an die Steuerung eingegeben.

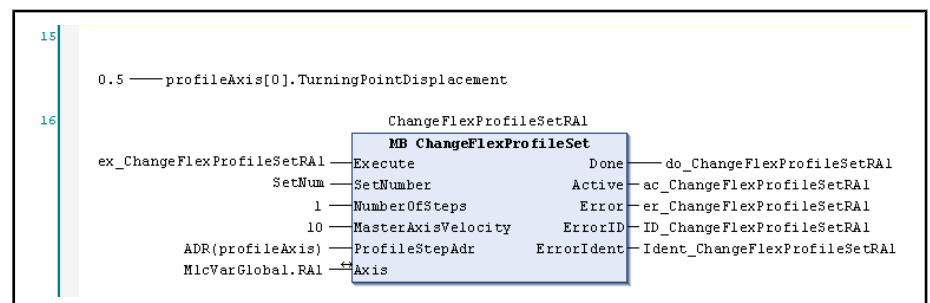


Abb.6-11: Funktionsbaustein zur Übergabe des Datenfeldes an die Steuerung (SetNum = 0 für FlexProfile 0)

Die Übergabe des Datensatzes erfolgt durch "Werte schreiben" am Eingang "ex_ChangeFlexProfileSetRA1", TRUE/FALSE.

Der Zustand des FlexProfiles ist danach geprüft/ok.



Verwenden Sie SetNum= 1. Sie schreiben damit auf "FlexProfile 1" und können das Ergebnis mit "FlexProfile 0", Eingabe über den IndraWorks-Dialog, vergleichen.

6.5 Kompletierung des SPS-Teilprojektes, Ausführen eines FlexProfile-Bewegungsschrittes

Sobald das Bewegungsprofil auf die Achse geschrieben ist, kann es durch Ausführen des Bausteins "ML_FlexProfile" gestartet werden. Dieser stellt die

Erste Schritte - FlexProfile

Kopplung zwischen Leit- und Folgeachse anhand des zuvor festgelegten und selektierten Bewegungsprofils her und synchronisiert die Folgeachse.

Eine Bewegung der Folgeachse ergibt sich ab diesem Zeitpunkt nur noch in Abhängigkeit der Leitachsbewegung. Im einfachsten Fall dreht die Leitachse mit konstanter Geschwindigkeit, beispielweise durch Aufruf des PLCopen Bausteins "MC_MoveVelocity". Auf Seiten der Folgeachse wird dann das gewünschte Bewegungsprofil abgefahren.

Importieren Sie dazu die Exportdatei "Application_FirstSteps_IndraLogic_Motion.iwx" aus dem Verzeichnis ""Iw:\Programme\Rexroth\IndraWorks\Projects"", **DCC_Control ► Logic ► Application**, Kontextmenü "Import".

In dieser Datei sind in Strukturiertem Text (MotionProg) und in Funktionsbausteinsprache (MotionProg_FUP) die bereits in "Erste Schritte Motion" erprobten Programme enthalten.

Folgende Änderungen sind notwendig:

1. Ändern Sie **VcomVA1** von 200 U/min auf 10 U/min.
2. **MoveVelVA1:**
 - "Acceleration" → 100 [rad/s²]
 - "Deceleration" → 100 [rad/s²]
3. Löschen Sie die Instanz "GearInRA1" von MC_GearIn. Sie wird durch die Instanz "FlexProfileRA1" von ML_FlexProfile ersetzt.

ML_FlexProfile in ST

Fügen Sie analog zu obiger Beschreibung einen Baustein des Typs "ML_FlexProfile" ein und beschalten ihn, wie nachfolgend beschrieben.

ML_FlexProfile einfügen und eine Instanz "FlexProfileRA1" deklarieren:

Deklarationsteil von "MotionProg":

1. Leere Zeile vor END_VAR einfügen.
2. FlexProfileRA1: eingeben, <F2>-Eingabehilfe, **benutzerdefinierte Typen ► ML_PLCOpen ► POUs ► Motion ► ML_FlexProfile**, Option "Mit Argumenten einfügen" abwählen.
3. Zeile komplettieren:
FlexProfileRA1: ML_FlexProfile;

Leere Zeile anstelle von "GearInRA1" anfügen

<F2>-Eingabehilfe, **Instanzaufrufe ► FlexProfileRA1**, Option "Mit Argumenten einfügen"

Eingänge beschalten und wenn nötig Eingangsvariablen deklarieren:

"Execute" → "ex_FlexProfileRA1" vom Typ BOOL

"RatioNumerator" → 1

"RatioDenominator" → 1

"MasterFineadjust" → 0

"SetSelection" → 0

"UseSwitchingPosition" → FALSE

"SwitchingPosition" → 0

"ProfileEntry" → "vom Typ MB_PROFILE_START_POINT"

SLAVE_CURRENT_MASTER_CURRENT,

ML_Base.library, Data Types, Flex Structs, MB_PROFILE_START_POINT

Erste Schritte - FlexProfile

"MasterOffset" → 0

"SlaveOffset" → 0

"SyncType" → "vom Typ MB_SYNC_TYPE"

Initialwert: SYNC_RAMPIN_CATCH_UP, ML_Base.library

"SyncVelocity" → 100

"SyncAcceleration" → 100

"ExecutionMode" → "vom Typ MB_EXECUTION_MODE"

Initialwert: EXECUTE_CYCLIC, ML_Base.library

"Master" → "VA1" vom Typ "AXIS_REF"

(bereits deklarierte Variable aus dem Bereich der "Variablen", Application, MlcVarGlobal)

"Slave" → "RA1" vom Typ "AXIS_REF"

(bereits deklarierte Variable aus dem Bereich der "Variablen", Application, MlcVarGlobal)

Ausgänge:

"InSync" → "is_FlexProfileRA1" vom Typ BOOL

"Active" → "ac_FlexProfileRA1" vom Typ BOOL

"CommandAborted" → "ca_FlexProfileRA1" vom Typ BOOL

"Error" → "er_FlexProfileRA1" vom Typ BOOL

"ErrorID" → "ID_FlexProfileRA1" vom Typ ERROR_CODE

"ErrorIdent" → "Ident_FlexProfileRA1" vom Typ ERROR_STRUCT

```

30      ErrorIdent=> Ident_PowerRA1);
31  FlexProfileRA1(
32      Execute:= ex_FlexProfileRA1,
33      RatioNumerator:= 1,
34      RatioDenominator:= 1,
35      MasterFineadjust:= 0,
36      SetSelection:= 0,
37      UseSwitchingPosition:= FALSE,
38      SwitchingPosition:= 0,
39      ProfileEntry:= MB_PROFILE_START_POINT.SLAVE_CURRENT_MASTER_CURRENT,
40      MasterOffset:= 0,
41      SlaveOffset:= 0,
42      SyncType:= MB_SYNC_TYPE.SYNC_RAMPIN_CATCH_UP,
43      SyncVelocity:= 100,
44      SyncAcceleration:= 100,
45      ExecutionMode:= MB_EXECUTION_MODE.EXECUTE_CYCLIC,
46      Master:= MlcVarGlobal.VA1,
47      Slave:= MlcVarGlobal.RA1,
48      InSync=> is_FlexProfileRA1,
49      Active=> ac_FlexProfileRA1,
50      CommandAborted=> ca_FlexProfileRA1,
51      Error=> er_FlexProfileRA1,
52      ErrorID=> ID_FlexProfileRA1,
53      ErrorIdent=> Ident_FlexProfileRA1);
54  GearOutRA1(
55      Execute:= ex_GearOutRA1,

```

Abb.6-12: "ML_FlexProfile" ist eingefügt und beschaltet (ST)

Erste Schritte - FlexProfile

ML_FlexProfile in FUP

Fügen Sie analog zu obiger Beschreibung einen Baustein des Typs "ML_FlexProfile" ein und beschalten ihn, wie nachfolgend beschrieben.

ML_FlexProfile einfügen und eine Instanz "FlexProfileRA1" deklarieren:

Eingänge beschalten und wenn nötig Eingangsvariablen deklarieren:

"Execute" → "ex_FlexProfileRA1" vom Typ BOOL

"RatioNumerator" → 1

"RatioDenominator" → 1

"MasterFineadjust" → 0

"SetSelection" → 0

"UseSwitchingPosition" → FALSE

"SwitchingPosition" → 0

"ProfileEntry" → "vom Typ MB_PROFILE_START_POINT"

SLAVE_CURRENT_MASTER_CURRENT,

ML_Base.library, Data Types, Flex Structs, MB_PROFILE_START_POINT

"MasterOffset" → 0

"SlaveOffset" → 0

"SyncType" → "vom Typ MB_SYNC_TYPE"

Initialwert: SYNC_RAMPIN_CATCH_UP, ML_Base.library

"SyncVelocity" → 100

"SyncAcceleration" → 100

"ExecutionMode" → "vom Typ MB_EXECUTION_MODE"

Initialwert: EXECUTE_CYCLIC, ML_Base.library

"Master" → "VA1" vom Typ "AXIS_REF"

(bereits deklarierte Variable aus dem Bereich der "Variablen", Application, MlcVarGlobal)

"Slave" → "RA1" vom Typ "AXIS_REF"

(bereits deklarierte Variable aus dem Bereich der "Variablen", Application, MlcVarGlobal)

Ausgänge:

"InSync" → "is_FlexProfileRA1" vom Typ BOOL

"Active" → "ac_FlexProfileRA1" vom Typ BOOL

"CommandAborted" → "ca_FlexProfileRA1" vom Typ BOOL

"Error" → "er_FlexProfileRA1" vom Typ BOOL

"ErrorID" → "ID_FlexProfileRA1" vom Typ ERROR_CODE

"ErrorIdent" → "Ident_FlexProfileRA1" vom Typ ERROR_STRUCT

Erste Schritte - FlexProfile

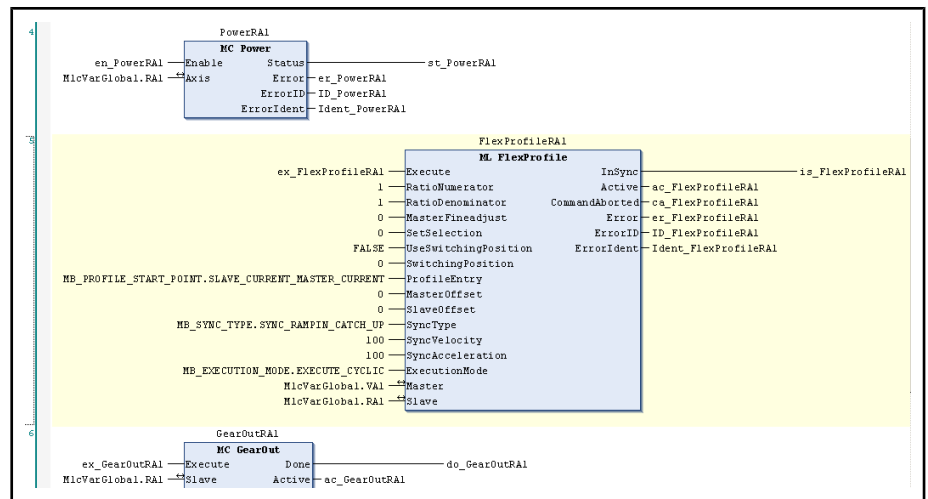


Abb.6-13: "ML_FlexProfile" ist eingefügt und beschaltet (FUP)

6.6 Download und Test des SPS-Teilprojektes

Der zuvor implementierte Bewegungsablauf muss jetzt noch übersetzt, in die Steuerung übertragen und gestartet werden.

Die Übersetzung erreichen Sie im Menüpunkt

- **Erstellen ► Übersetzen**, Übersetzen des hinzu gekommenen Anteils der Applikation.
- **Erstellen ► Neu übersetzen**, Übersetzen der kompletten Applikation.

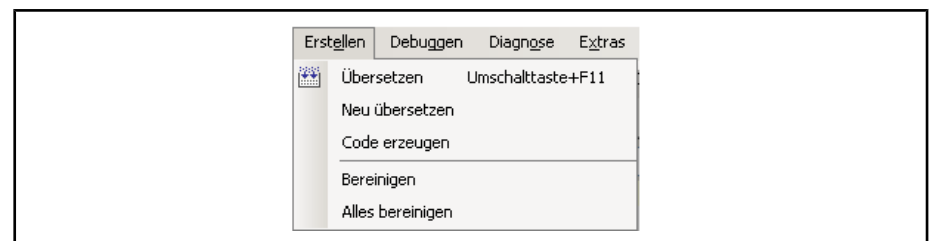


Abb.6-14: Applikation übersetzen

Zum Übertragen der kompilierten Daten in die Zielsteuerung wählen Sie den Menüpunkt **Debuggen ► Einloggen** (nachfolgendes Bild).

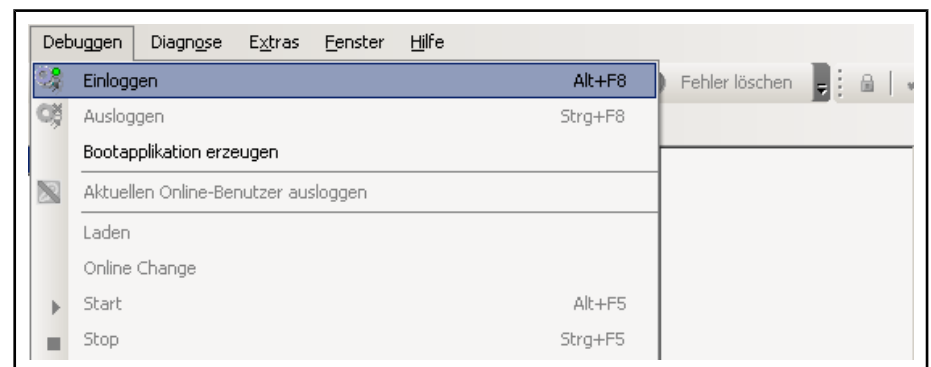


Abb.6-15: Übertragen der kompilierten Daten in die Steuerung

In Abhängigkeit ob eine Applikation bereits geladen ist und ob diese Applikation eine Vorgängerin Ihrer aktuellen Applikation ist, werden Sie um Zustimmung zum Download gebeten.

Erste Schritte - FlexProfile

Nach erfolgreicher Übertragung ist die Applikation noch gestoppt (rot im Projektnavigator und in der Statuszeile), und die Variablen in Deklaration und Implementation sind zur Anzeige der Werte vorbereitet. Die Werte sind jedoch noch nicht gültig.

Starten Sie die Bearbeitung in der Steuerung über **Debuggen ▶ Start**.

6.7 Workflow "Virtuelle Achse → FlexProfile 0 → Reale Achse"

Der nachstehende Arbeitsablauf zeigt das Starten und Stoppen der Achsen und des FlexProfiles:

1. Leistungszuschaltung für die reale Achse

Zuschalten des "en_PowerRA1" (MC_Power) durch "Werte schreiben".
st_PowerRA1= TRUE (MC_Power).

2. Start der virtuellen Achse

Zuschalten des "ex_MoveVelVA1" (MC_MoveVelocity) durch "Werte schreiben".

Hochlauf bei VactVA1 = 10 U/min beendet.

3. Nachführen der realen Achse

Zuschalten des "ex_FlexProfileRA1" (ML_FlexProfile) durch "Werte schreiben".

Das zu erwartenden Kopplungsverhalten ergibt sich zu:

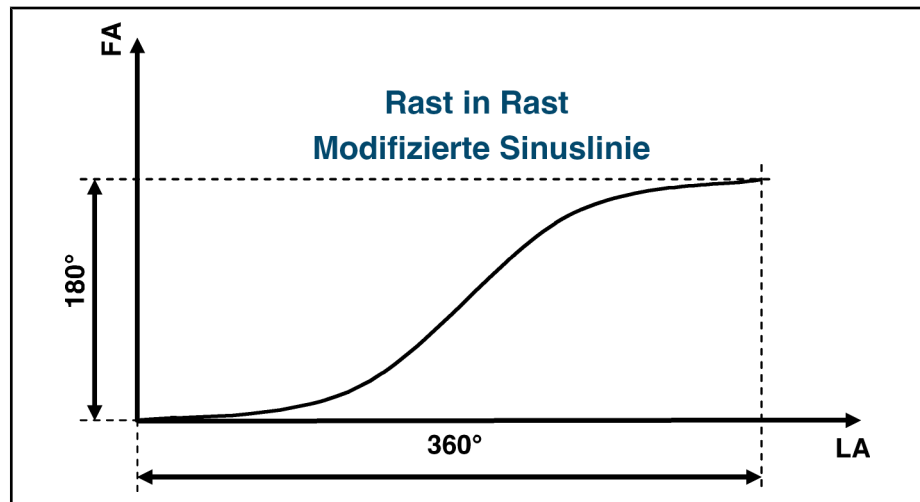


Abb.6-16: Kopplungsprofil zwischen der Leit- (VA1) und Folgeachse (RA1)

4. Abschalten des "ex_FlexProfileRA1" (ML_FlexProfile) durch "Werte schreiben".

is_FlexProfileRA1= FALSE, ac_FlexProfileRA1= TRUE.
Die reale Achse folgt der virtuellen weiterhin.

5. Auflösen der Synchronität

Zuschalten des "ex_GearOutRA1" (MC_GearOut) durch "Werte schreiben".

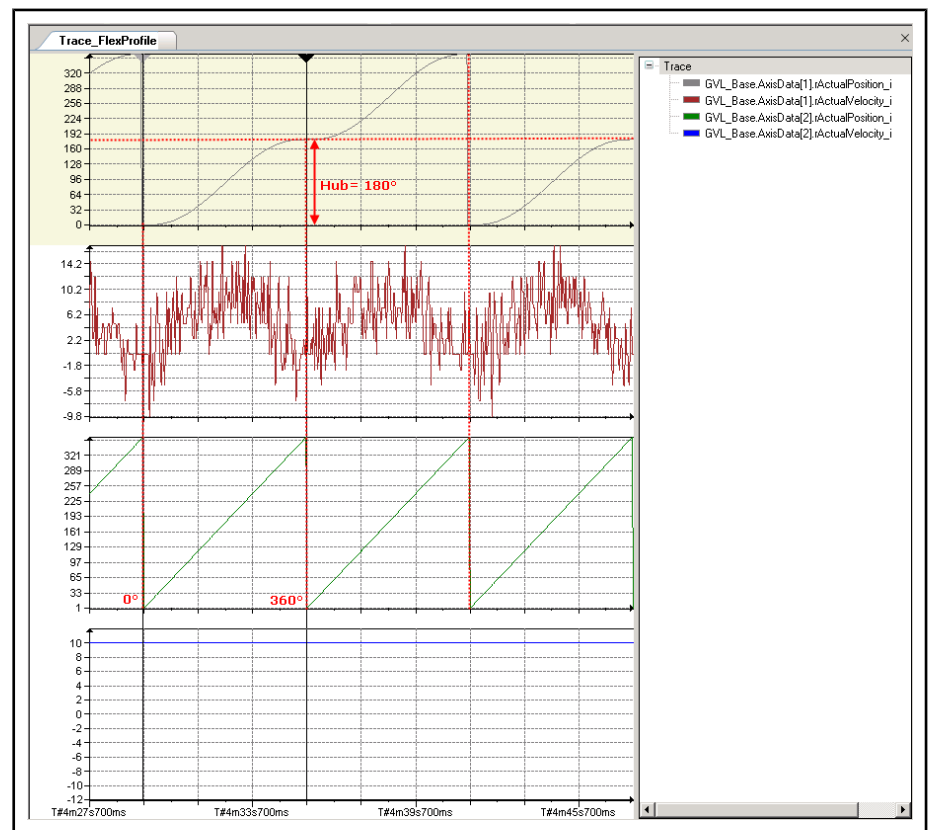
Die Abhängigkeit der realen Achse von der virtuellen ist aufgehoben. Beide Achsen drehen jedoch mit unveränderter Geschwindigkeit weiter.

Erste Schritte - FlexProfile

6. Änderung der Sollgeschwindigkeit "VcomVA1"
Die veränderte Sollgeschwindigkeit wirkt sich nur noch auf die virtuelle Achse aus, "VactRA1" bleibt unverändert.
7. Abschalten des "ex_GearOutRA1" durch "Werte schreiben" True/False.
do_GearOutRA1= FALSE
Die reale Achse bleibt stehen. Sie wird nach Erreichen des Stillstandes aktiv in dieser Position gehalten.
8. Stoppen der virtuellen Achse
Zuschalten des "ex_StopVA1" (MC_Stop)
Die virtuelle Achse wird abgebremst auf VactVA1= 0 U/min.
9. Leistungsabschaltung für die reale Achse
Abschalten von "en_PowerRA1" (MC_Power) durch "Werte schreiben".

6.8 Aufzeichnung der Bewegung beider Achsen (Trace)

In der Traceaufzeichnung, können die Bewegungen der Leit- und Folgeachse dargestellt werden. Es ergibt sich nachfolgender Verlauf.



Grau Folgeachse ("RA1"), Position
Rot Folgeachse ("RA1"), Geschwindigkeit
Grün Leitachse ("VA1"), Position
Blau Leitachse ("VA1"), Geschwindigkeit

Abb.6-17: Aufzeichnung der Bewegung beider Achsen

Die Traceaufzeichnung zeigt unten den Geschwindigkeitsverlauf der Leitachse.

Diese dreht sich mit einer konstanten Geschwindigkeit von 10 [U/min].

Erste Schritte - FlexProfile

Darüber befindet sich die Position der Leitachse.

Das oberste Signal zeigt die Position der Folgeachse.

Es ist zu erkennen, dass das zuvor parametrisierte FlexProfile korrekt abgefahren wird.

6.9 Projekt wiederherstellen

Ausgehend vom erstellten Projekt wurde eine Exportdatei erzeugt, die es unabhängig von der Steuerungshardware (IndraLogic XLC L25, L45, L65 (mit SERCOS III-Bus) oder IndraMotion MLC L25, L45, L65) und der Antriebstechnik erlaubt, Teile des Projektes wieder zu regenerieren.

Die Exportdatei wurde vom Knoten "Application" beginnen erstellt und enthält folgende Bestandteile:

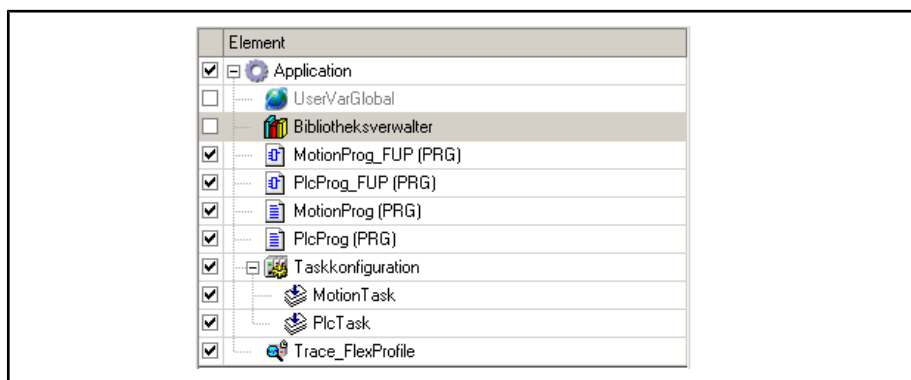


Abb.6-18: Bestandteile der Exportdatei

Die angegebenen Komponenten werden beim Import überschrieben.

Die Exportdatei "Application_FirstSteps_FlexProfile.iwx" befindet sich im Verzeichnis "lw:\Programme\Rexroth\IndraWorks\Projects".

7 Erste Schritte - Robot-Control

7.1 Ziel des Projekts "Erste Schritte - Robot-Control"

Anhand eines einfachen Beispiels soll die grundsätzliche Vorgehensweise beim Erstellen eines **Robot-Control**-Projekts (nachfolgend RoCo-Projekt genannt) mit IndraMotion MLC erläutert werden. Es wird, vom Anlegen eines Projektes bis zum Ausführen des RoCo-Quellcodes, stichpunktartig auf die hierfür minimal notwendige Konfiguration in IndraWorks und die Programmierung in der SPS eingegangen.

Im Beispiel-Projekt "RoCoTestV12" wird davon ausgegangen, dass nur virtuelle Achsen zur Verfügung stehen.

Alternativ soll kurz beschrieben werden, was zu tun ist, wenn reale Achsen zur Verfügung stehen.

Hierzu werden die virtuellen Achsen im Projekt durch reale Achsen ersetzt.

Dabei ist darauf zu achten, dass

1. die realen Achsen wieder die gleiche Achsbezeichnung Axis1, Axis2 und Axis3 tragen und
2. die Achsnummern 1, 2 und 3 in der gleichen Reihenfolge zugeordnet sind.

Dadurch wird gewährleistet, dass das SPS-Programm ohne Anpassung lauffähig ist.



Der Anwender kann entweder das Beispielprojekt von der IndraWorks Installations-DVD aus dem Verzeichnis "\AddOns\MLC\Example-Project\RobotControl" laden; es enthält

- die Parameter-Export-Datei (*.par),
- das RCL-Programm (*.qll),¹⁾
- das SPS-Programm und
- die Visualisierungsdaten

- oder -

der Anwender erstellt das Projekt Schritt für Schritt anhand der nachfolgenden Schritte. Dazu sind die im Anhang aufgeführten Detaildaten zur Unterstützung zu nutzen.

7.2 Erforderliche Hard- und Software

Erforderliche Hardware *Minimal:*

- Steuerung, Typ "IndraMotion MLC Lx5"
- aktuelle Firmware ist geladen
- IP-Adresse ist bekannt
- Programmiergerät (PC/Laptop)
- Steuerung ist über Ethernet mit dem Programmiergerät verbunden

Optional:

- 3 reale Achsen (z.B. IndraDrive, EcoDriveCS,...)
- aktuelle Firmware (ab MPx07) ist geladen
- SRAM Modul (CFL01.1 Y1)

¹⁾ RCL bedeutet **Robot Control Language**

Erste Schritte - Robot-Control

Erforderliche Software IndraWorks-Installation mit MLC als Zielsystem.

7.3 Vorgehensweise beim Erstellen eines RoCo-Projektes

1. IndraWorks starten und ein neues Projekt anlegen

Nach dem Start von IndraWorks wird über **Datei ▶ Neu ▶ Projekt** das gewünschte Projekt angelegt.

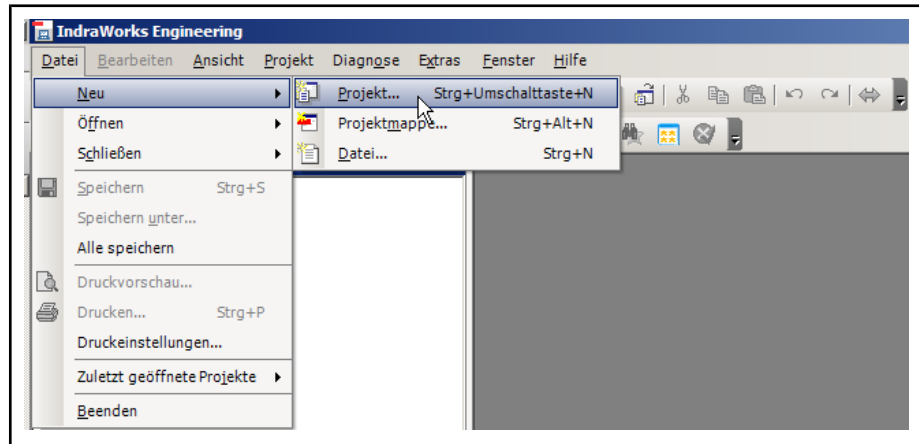


Abb.7-1: Neues IndraWorks Projekt anlegen

2. MLC Steuerung anlegen und konfigurieren

Hierzu wird eine IndraMotion MLC Lx5 aus der Bibliothek in das Projekt eingefügt und deren Konfigurationsdaten wie Geräte-Name, IP-Adresse, Hardwarekonfiguration, etc. spezifiziert.

Im Dialog "Konfiguration" kann das Funktionsmodul SRAM Modul (CFL01.1-Y1) - Speichererweiterung für die MLC Lx5, angelegt werden. Alternativ ist dies durch "Drag" aus der **Gerätebibliothek ▶ FM** und "Drop" im Projekt-Explorer auf der Steuerung möglich.

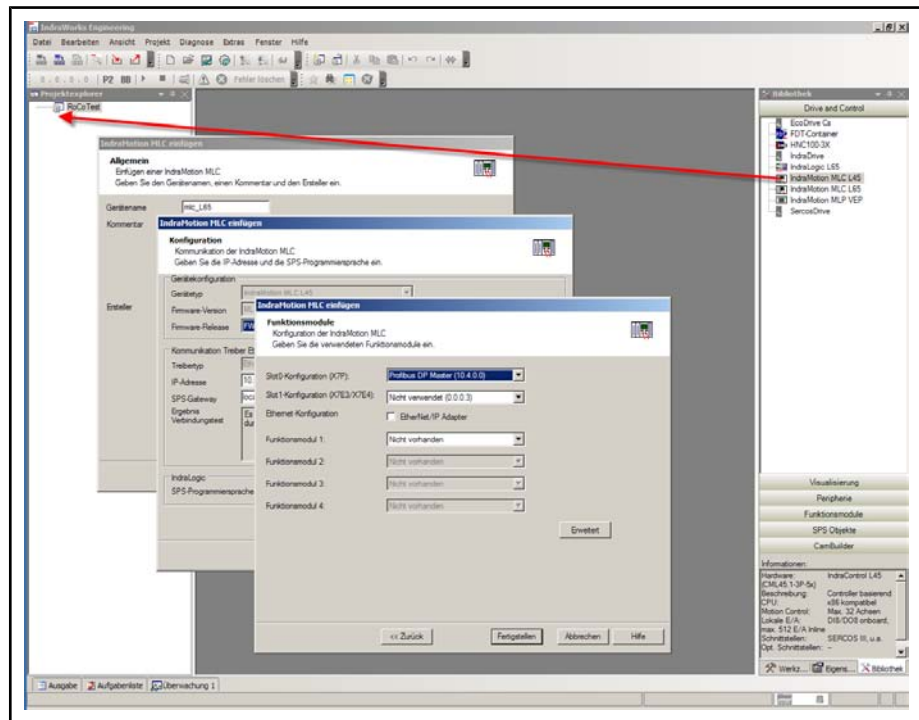


Abb.7-2: MLC anlegen und konfigurieren

3. Drei Achsen anlegen und konfigurieren

Es werden beispielhaft 2 Szenarien beschrieben:

Szenario 1 mit virtuellen Achsen:

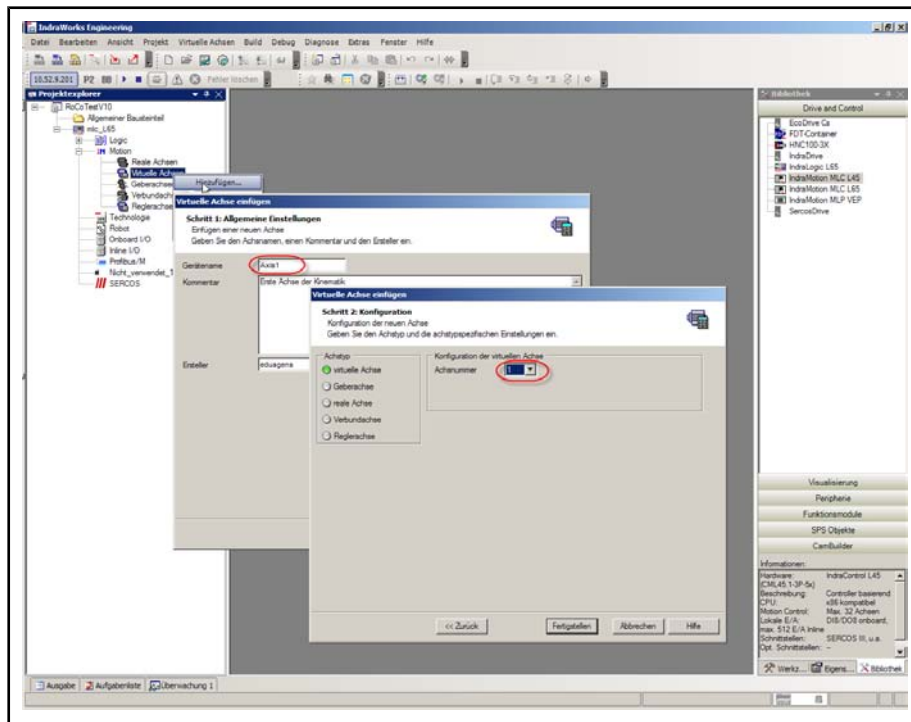


Abb. 7-3: Projekt mit virtuellen Achsen

Szenario 2 mit realen Achsen:

Das Anlegen der realen Achsen erfolgt durch "Drag" aus der **Gerätebibliothek** ► **Drive and Control** und "Drop" im Projekt-Explorer auf den SERCOS-Knoten.

Erste Schritte - Robot-Control

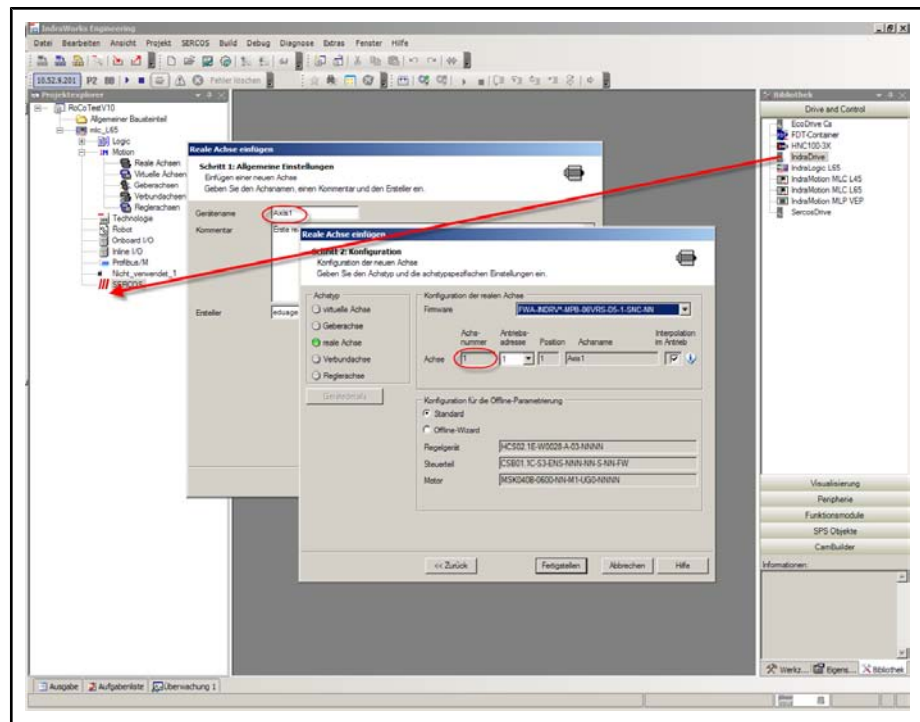


Abb. 7-4: Projekt mit realen Achsen



Für das Beispielprojekt liegen für beide Szenarien Parametersätze bereit, die importiert werden können. Die Parametersätze sind im Projekt-Verzeichnis "RoCoTestV12\RoCoTestV12virtuell.par" (für virtuelle Achsen) oder "RoCoTestV10\RoCoTestV10real.par" (für reale Achsen) abgelegt.

4. Kinematik hinzufügen und konfigurieren

Am Kinematikknoten des MLC-Projektes wird nun eine neue Kinematik hinzugefügt.

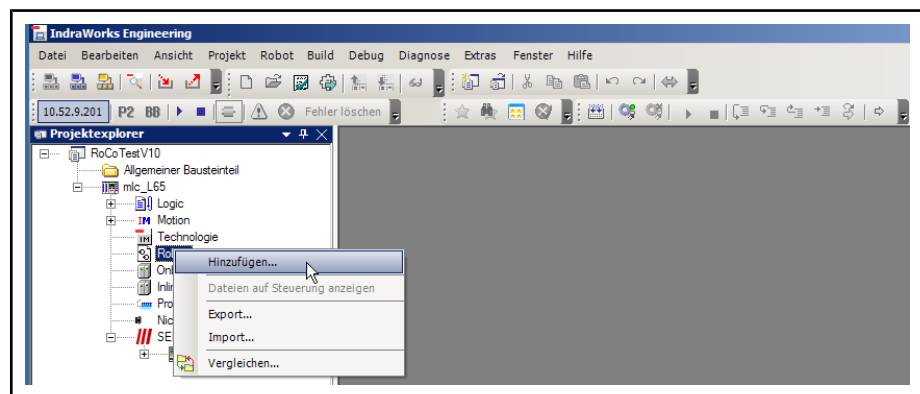


Abb. 7-5: Neue Kinematik hinzufügen

Konfiguration entsprechend den nachfolgenden Dialogen:

Erste Schritte - Robot-Control

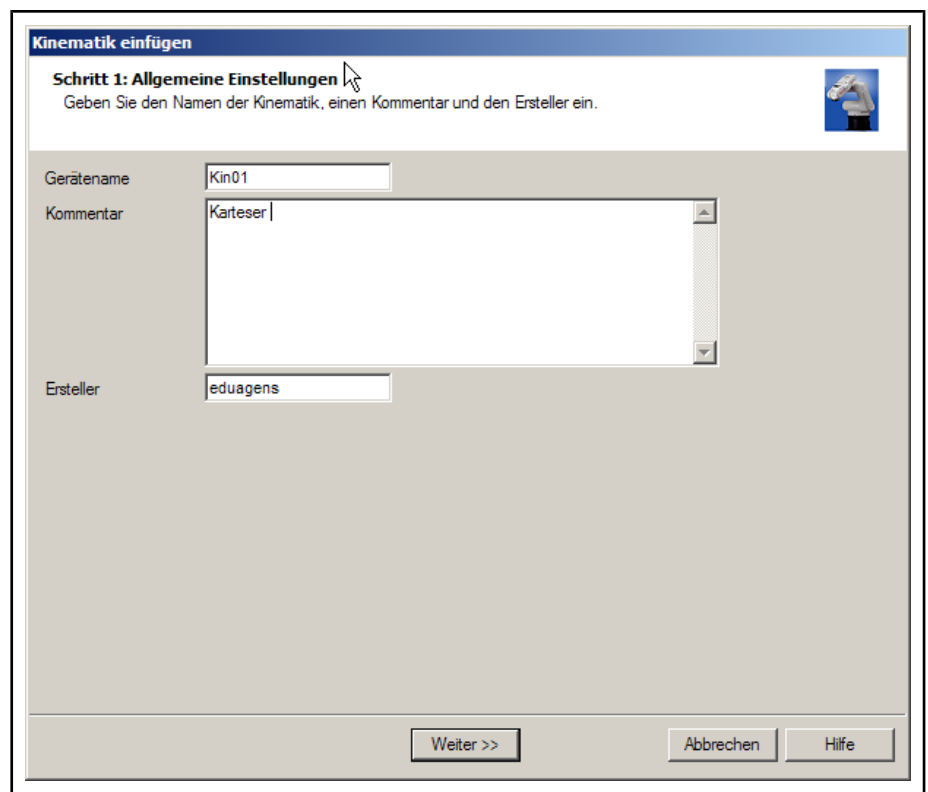


Abb. 7-6: Kinematik-Namen festlegen.



In der SPS wird später eine globale MLC-Variable gleichen Namens vom Typ AXIS_GROUP_REF angelegt.

Alle Kinematik-Funktionsbausteine benötigen diese Variable als Eingangsvariable.

Erste Schritte - Robot-Control

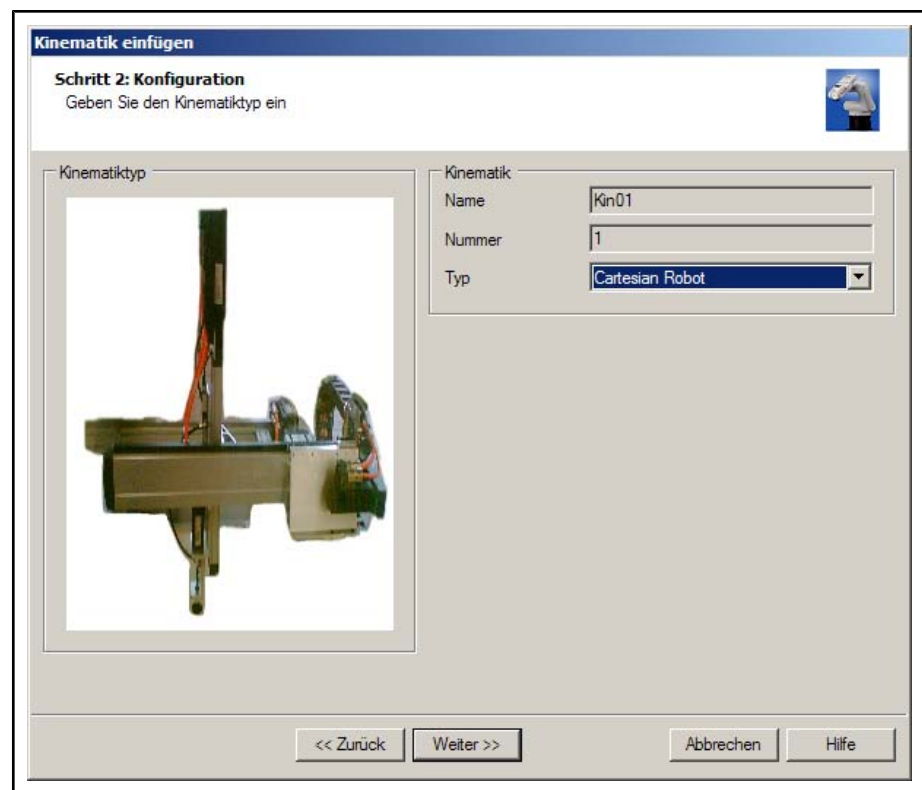


Abb.7-7: Kinematik-Typ festlegen

Im obigem Dialog sind alle Kinematiken zur Auswahl aufgelistet, die von der Steuerung interpretiert werden können.

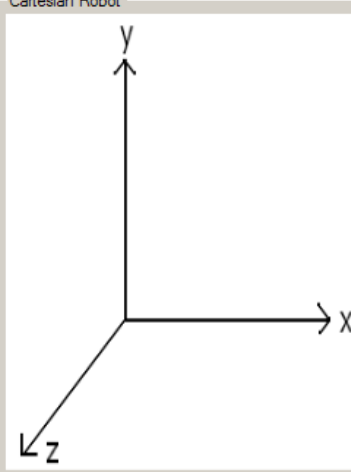
In unserem Beispiel wird der Kinematik-Typ "Cartesian Robot" benötigt.

Erste Schritte - Robot-Control

Kinematik einfügen

Schritt 3: Achszuordnung
Ordnen Sie die für die Kinematik benötigten Achsen zu.

Cartesian Robot



Achszuordnung

Name	Achse Typ	Steuerun	Achse
A1	Positionierungsachse	0	Axis1
A2	Positionierungsachse	0	Axis2
A3	Positionierungsachse	0	Axis3
A4	---	0	
A5	---	0	
A6	---	0	
A7	---	0	
A8	---	0	
A9	---	0	
A10	---	0	
A11	---	0	
A12	---	0	
A13	---	0	

<< Zurück Weiter >> Abbrechen Hilfe

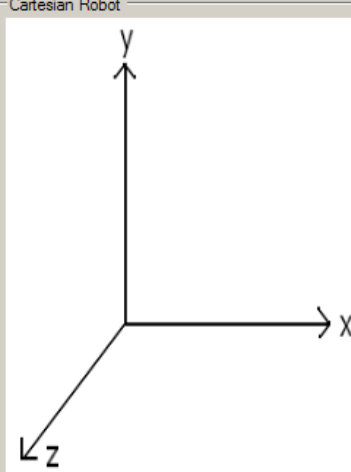
Abb. 7-8: Kinematik-Achsen zuordnen.

Alle zur Kinematik gehörenden Achsen (real oder virtuell) müssen an dieser Stelle zugeordnet und definiert werden.

Kinematik einfügen

Schritt 4: Achslänge und Kopplung
Geben Sie Achslänge und Achskopplung ein.

Cartesian Robot



Achslänge/Spernbereich

L	mm
L1	0
L2	0
L3	0
L4	0
L5	0
L6	0
L7	0
L8	0
L9	0
L10	0
L11	0
L12	0
L13	0
L14	0

Achskopplung

K	
K1	0
K2	0
K3	0
K4	0
K5	0
K6	0
K7	0
K8	0
K9	0
K10	0
K11	0
K12	0
K13	0
K14	0

<< Zurück Weiter >> Abbrechen Hilfe

Abb. 7-9: Achslänge und Achskopplung

Erste Schritte - Robot-Control

Hier werden die für Roboter typischen Achskopplungen und Achslängen eingegeben.

Für das hier besprochene Beispiel ist dies nicht nötig.

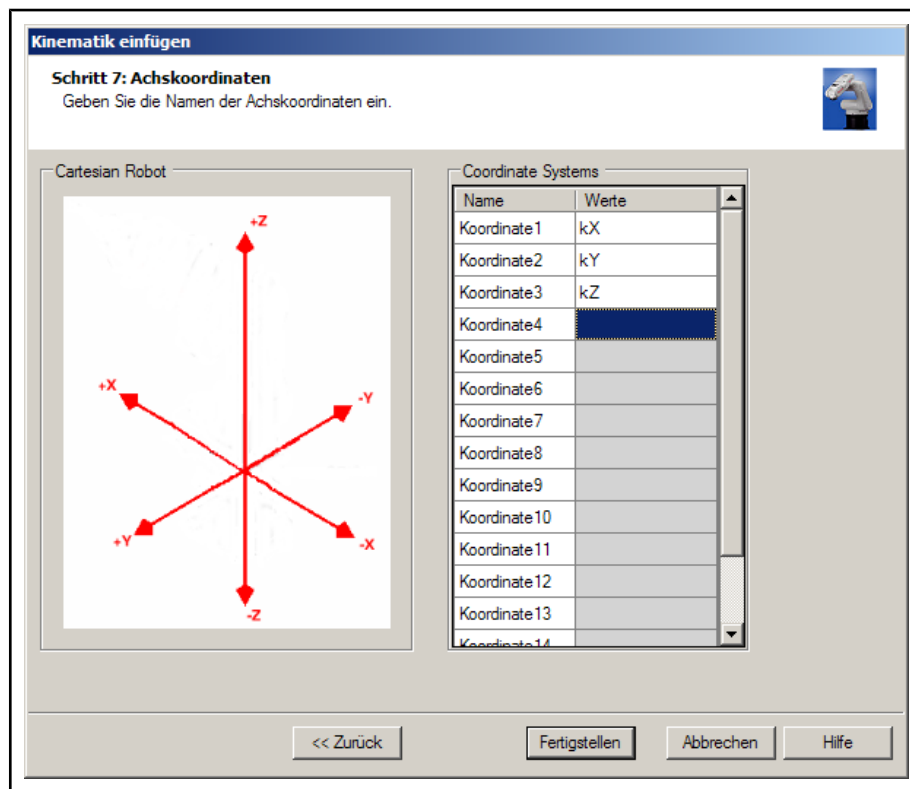


Abb. 7-10: Koordinaten-Namen festlegen

In diesem Schritt werden die Koordinaten Namen definiert.



Die hier festgelegten Koordinatenbezeichnungen finden in der später erfolgenden RoCo-Programmierung ihre Anwendung.

5. Online schalten

Mit dem online Schalten werden die Steuerung in P2 geschaltet und die bis hierher erfolgten Einstellungen in die Steuerung übertragen.

6. Parameter importieren

Für das Beispielprojekt können an dieser Stelle die beim Projekt abliegenden C-, A- und K-Parameter importiert werden.

Auswahl:

Projekt-Verzeichnis "RoCoTestV12\RoCoTestV12virtuell.par" für virtuelle Achsen oder "RoCoTestV12\RoCoTestV12real.par" für reale Achsen.

Nach erfolgreichem Parameterimport sollte sich die Steuerung in BB (P4) schalten lassen.

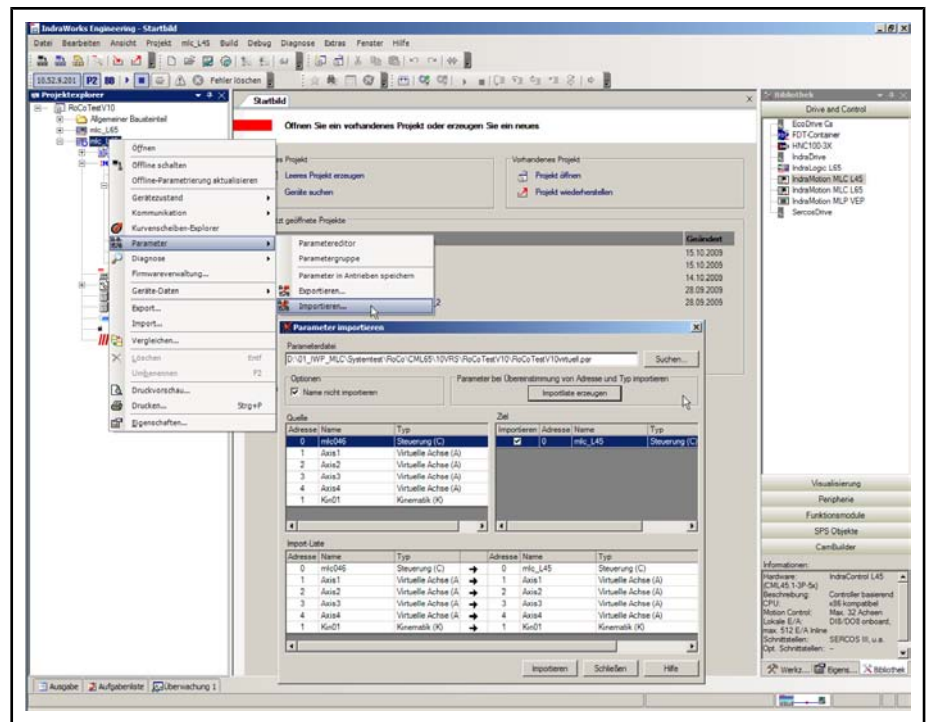


Abb. 7-11: Parameter importieren.

7. Bewegungsprogramm erstellen

Nun wird ein neues RoCo-Bewegungsprogramm für die Kinematik angelegt.

Alternativ kann es aus dem Projekt **RoCo-TestV12** ► **mlc_Lx** ► **_Emb_Prog1.qll** ► **Prog1.qll** mit

- Prog1.QLL, (⇒ Quellcode im Reintextformat),
 - Prog1.PKT, (⇒ Punktetabelle),
 - Prog1.IRD, (⇒ Quellcode im Maschinenformat, Kompilat),
 - Prog1.SYM, (⇒ Symboldatei, zum Debuggen notwendig),
- importiert werden.

Erste Schritte - Robot-Control

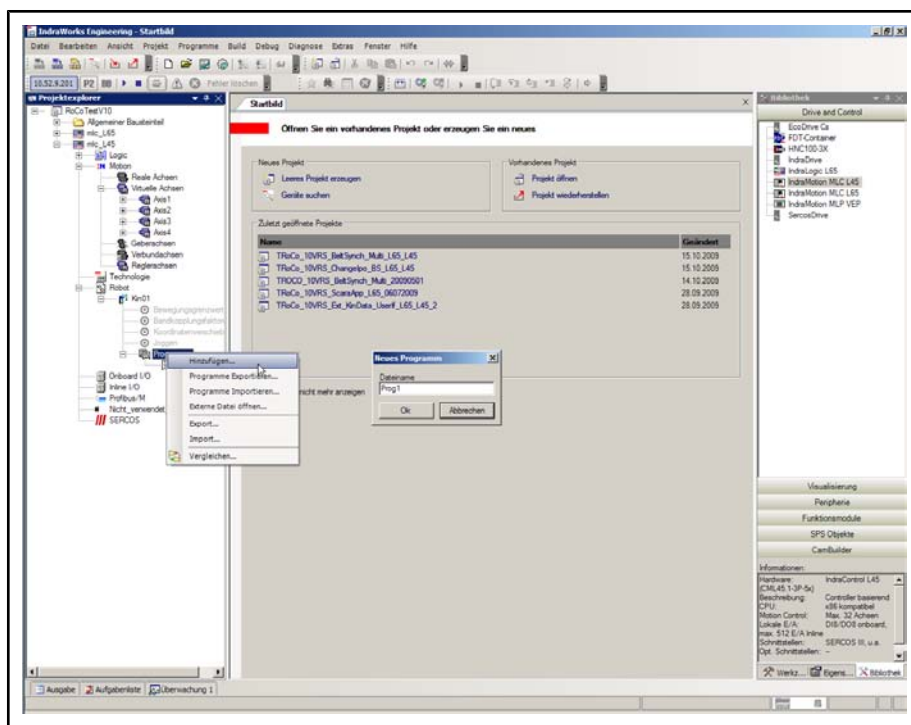


Abb. 7-12: Neues RoCo-Programm anlegen.

Nach dem Editieren des RoCo-Bewegungsprogramms (*.QLL) kann es übersetzt werden.

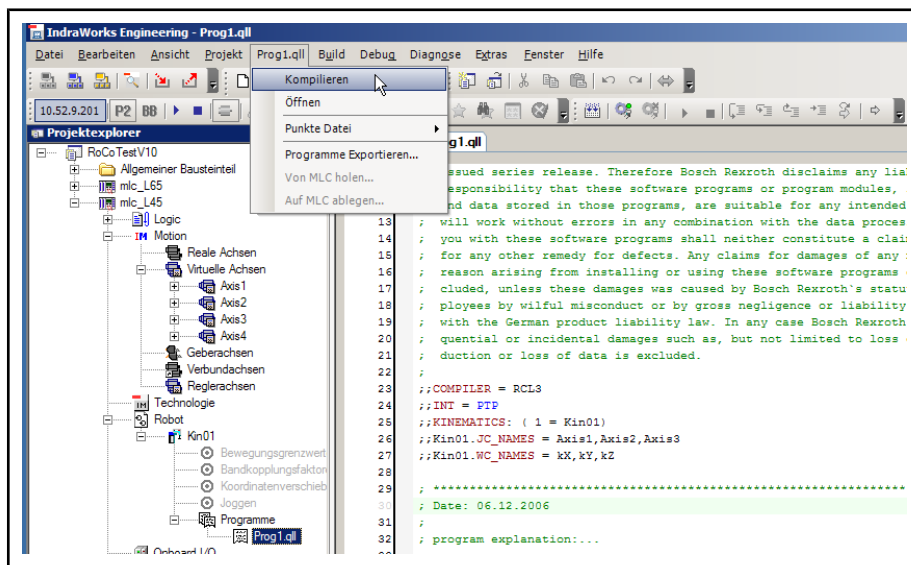


Abb. 7-13: Editieren des RoCo-Bewegungsprogramms

Nach dem Übersetzen wird automatisch eine Punktetabelle (*.PKT) angelegt, die jetzt editiert werden kann.

Erste Schritte - Robot-Control

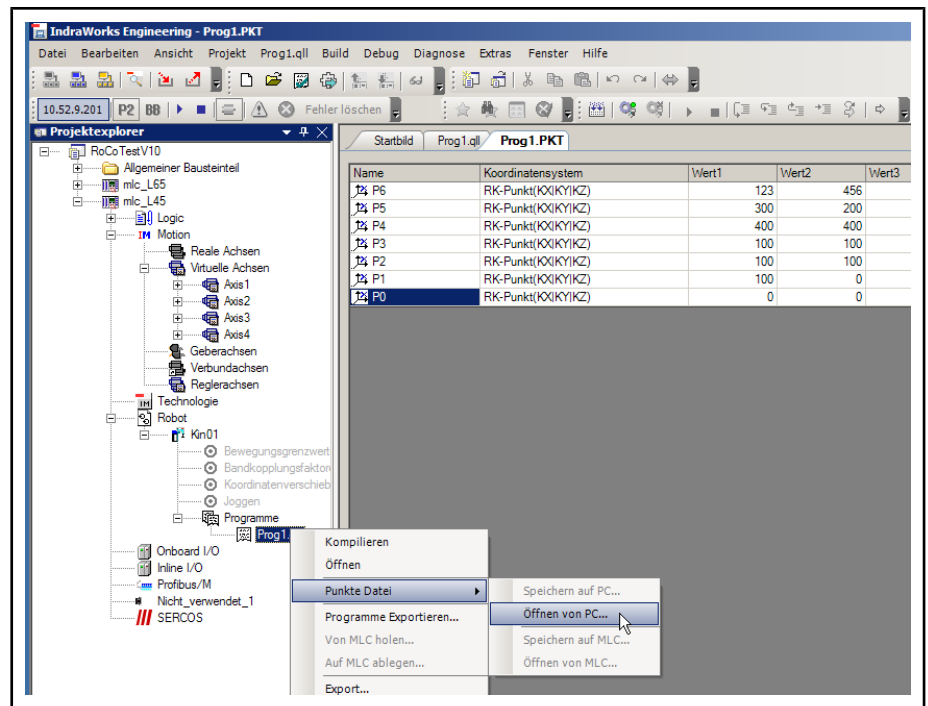


Abb.7-14: Punktetabelle bearbeiten.

Im Online-Betrieb das Bewegungsprogramm, Punktedatei etc. in die Steuerung laden.



Sollte kein SRAM-Funktionsmodul vorhanden sein, muss dieser Schritt nach jedem Aus- und Einschalten der Steuerung wiederholt werden, da sich die Daten im flüchtigen Speicher der Steuerung befinden.

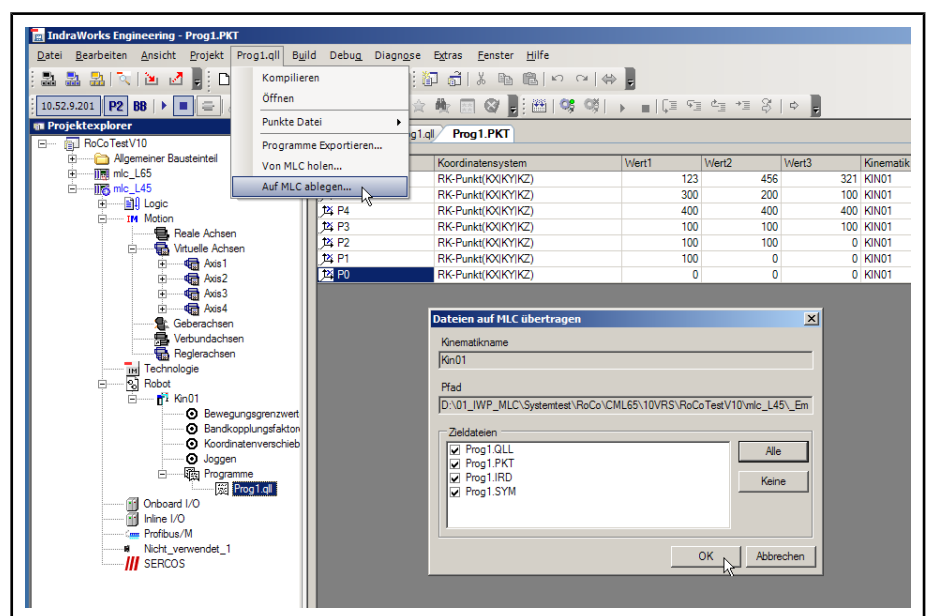


Abb.7-15: RoCo-Programm in die Steuerung laden.

Erste Schritte - Robot-Control

8. RoCo-Status im SPS-Programm

Zur Steuerung des SPS-Teils befindet sich eine Visualisierung im Beispielprojekt. Damit lassen sich alle nötigen Bedienaktionen, vom Zuschalten der Leistung für reale Achsen, bis zum Starten des RoCo-Programms ausführen, sowie der Zustand der Kinematik ablesen.

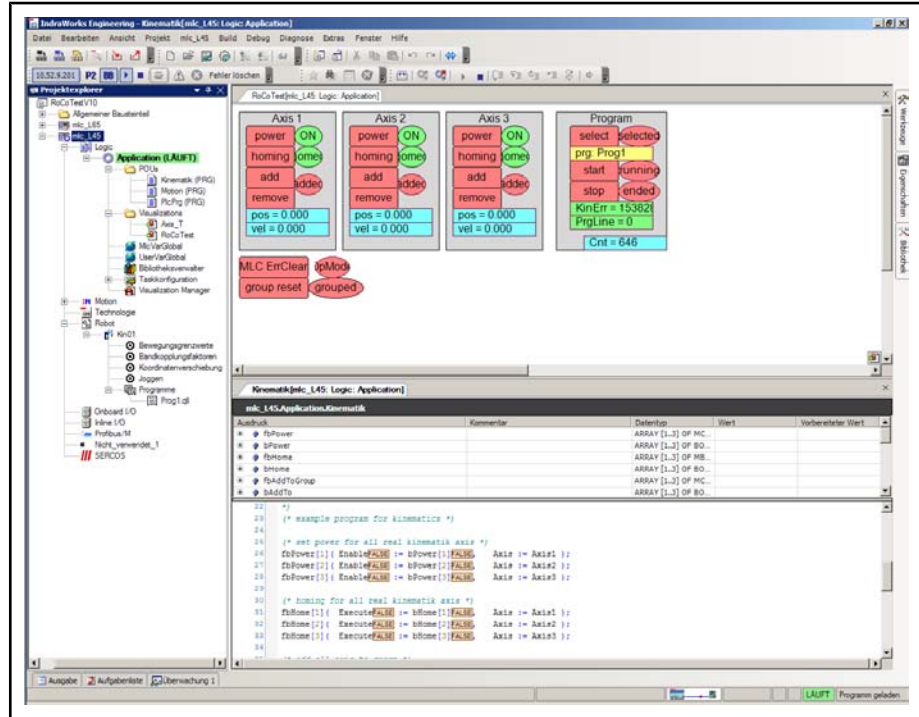


Abb.7-16: RoCo im SPS-Programm

Kurzbeschreibung der SPS-Funktionalität in der programmierten Reihenfolge.

1. Leistung zuschalten.

Dieses Kommando bitte nur für reale Achsen ausführen.

Bei virtuellen Achsen ist dieses Bit immer gesetzt.

Ob einer Achse Leistung zugeschaltet werden konnte, ist am Bit "AxisData[n].Axis_Power" abzulesen.

2. Referenzfahren (Homing) ausführen.

Das Referenzfahren ist nur für reale Achsen ohne Absolutgeberauswertung erforderlich.

Der Status, ob eine Achse referenziert ist, ist an dem Bit "AxisData[n].Axis_Homed" erkennbar.

3. Achsen der Kinematik zuordnen.

In diesem Schritt werden alle Achsen, die zur definierten Kinematik gehören, gruppiert. Dabei wechseln die Achsen von der Achs-Zustandsmaschine zur Kinematik-Zustandsmaschine.

Die Achsen, die gruppiert sind, können nicht mehr mit den PlcOpen Bewegungsbausteinen von der SPS aus gesteuert werden, sondern erhalten ihre Verfahrbefehle direkt vom RoCo-Bewegungsprogramm.

Dies ist erst dann möglich, wenn alle Achsen, die zur Kinematik gehören, gruppiert sind.

Ob sich alle benötigten Achsen in Coordinated Motion befinden, ist am Status "AxisData[x].CoordinatedMotion" zu erkennen.

4. Achsen von der Kinematik lösen

Dies ist erforderlich, um einzelne Achsen außerhalb des RoCo-Bewegungsprogramms verfahren zu können.

5. RoCo-Bewegungsprogramm selektieren

Um ein RoCo-Bewegungsprogramm ausführen zu können, muss es ausgewählt (selektiert) sein. Dies ist nach jedem Programm-Ende und nach jedem Programm-Stopp erneut auszuführen.

6. RoCo-Bewegungsprogramm starten

Dieses Kommando startet das selektierte Programm.

7. RoCo-Bewegungsprogramm stoppen

Hiermit werden alle Achsbewegungen gestoppt und das selektierte Programm zurückgesetzt.

Für den Fehlerfall

1. Kinematik-Gruppe zurücksetzen (MC_GroupReset)

Damit werden alle Kinematik-Fehler gelöscht.

2. MLC Steuerungs- und Antriebsfehler löschen (C-0-1030)

Löscht von der SPS aus alle Antriebsfehler.

7.4 Detail zum Projekt "RoCoTestV12"

Parametervoreinstellungen (minimale Änderungen)

C-Parameter:

Die Belegung der C-Parameter ergeben sich automatisch mit der Übernahme der MLC in den Projektnavigator und der Anlage der virtuellen Achsen.

Für die Achsen sind nachstehende Änderungen in den IndraWorks-Dialogen "Wichtungen/Maßeinheiten", "Einstellungen Statusmeldungen" und "Bewegungsgrenzwerte" sinnvoll:

- A-0-0053 , Wichtungsart für Beschleunigungsdaten: translatorisch
- A-0-0056 , Wichtungsart für Geschwindigkeitsdaten: translatorisch
- A-0-0056 , Wichtungsart für Positionsdaten: translatorisch, absolut
- Grenzwerte und Fensterwerte entsprechend.

C-0-2484, Kinematikkonfigurationsliste und alle K-Parameter ergeben sich aus der Kinematik-Konfiguration.

Listing des RoCo-Programms

```
;;COMPILER = RCL3
;;INT = PTP
;;KINEMATICS: ( 1 = Kin01)
;;Kin01.JC_NAMES = Axis1,Axis2,Axis3
;;Kin01.WC_NAMES = kX,kY,kZ

; *****
; Dieses Programm wurde erstellt von:
; Bearbeiter: rudokohl
; Abteilung: esc3
; Datum: 06.12.2006
;
; Programmbeschreibung:...
; *****

PROGRAM Prog1
; declarations
```

Erste Schritte - Robot-Control

```

Def Point: P0, P1, P2, P3, P4, P5, P6

BEGIN
; program
Kin01.V_PTP = 0.4      ; velocity for point to point movement
Kin01.A = 1000         ; acceleration
Kin01.V = 1200         ; velocity

;LOOP:
;JUMP LOOP

MOVE TO P0             ; move to P0
Kin01.V = 400          ; velocity
WAIT 3                ; wait for 3 seconds
REPEAT 4 TIMES
    MOVE TO P1          ; move to P1
    MOVE TO P2          ; move to P2
    MOVE TO P3          ; move to P3
    MOVE VIA P4, P5 TO P6 ; move via to P6
    WAIT 3              ; wait for 3 seconds
REPEAT_END

PROGRAM_END

```

Listing des SPS-Programms (Kinematik)

```

PROGRAM Kinematik
VAR
    fbPower      : ARRAY[1..3] OF MC_Power;
    bPower       : ARRAY[1..3] OF BOOL;
    fbHome       : ARRAY[1..3] OF MB_Home;
    bHome        : ARRAY[1..3] OF BOOL;
    fbAddToGroup : ARRAY[1..3] OF MC_AddAxisToGroup;
    bAddTo       : ARRAY[1..3] OF BOOL;
    fbRemFromGroup : ARRAY[1..3] OF MC_RemAxisFromGroup;
    bRemove      : ARRAY[1..3] OF BOOL;
    fbGroupReset : MC_GroupReset;
    bGroupReset  : BOOL;
    fbPrgSelect  : MB_IRDProgSelect;
    bPrgSelect   : BOOL;
    fbPrgStart   : MB_IRDProgStart;
    bPrgStart    : BOOL;
    fbPrgStop    : MB_IRDProgStop;
    bPrgStop     : BOOL;
    strProg      : STRING(8) := 'Prog1';
END_VAR

(* example program for kinematics *)

(* set power for all real kinematics axes *)
fbPower[1]( Enable := bPower[1], Axis := Axis1 );
fbPower[2]( Enable := bPower[2], Axis := Axis2 );
fbPower[3]( Enable := bPower[3], Axis := Axis3 );

(* homing for all real kinematics axes *)
fbHome[1]( Execute := bHome[1], Axis := Axis1 );
fbHome[2]( Execute := bHome[2], Axis := Axis2 );
fbHome[3]( Execute := bHome[3], Axis := Axis3 );

(* add all axes to group *)
fbAddToGroup[1]( Execute := bAddTo[1], Axis := Axis1, AxisGroup := Kin01 );
fbAddToGroup[2]( Execute := bAddTo[2], Axis := Axis2, AxisGroup := Kin01 );
fbAddToGroup[3]( Execute := bAddTo[3], Axis := Axis3, AxisGroup := Kin01 );

(* remove all axes from group *)
fbRemFromGroup[1]( Execute := bRemove[1], Axis := Axis1, AxisGroup := Kin01 );
fbRemFromGroup[2]( Execute := bRemove[2], Axis := Axis2, AxisGroup := Kin01 );
fbRemFromGroup[3]( Execute := bRemove[3], Axis := Axis3, AxisGroup := Kin01 );

(* select program *)
fbPrgSelect( Execute := bPrgSelect, IRDProgName := strProg, AxisGroup := Kin01 );

(* start program *)
fbPrgStart( Execute := bPrgStart, AxisGroup := Kin01 );

(* stop program *)
fbPrgStop( Execute := bPrgStop, AxisGroup := Kin01 );

(* reset group *)
fbGroupReset( Execute := bGroupReset, AxisGroup := Kin01 );

END_PROGRAM

```

Listing des SPS-Programms (PlcPrg) zum Fehlerlöschen

```
PROGRAM PlcPrg
VAR
    fbReset      : MB_WriteParameter;
    bErrClear    : BOOL;
END_VAR

(* error clear MLC *)
fbReset( Execute := bErrClear,
        ParameterNumber := FP_C_0_1030,
        Value := 3,
        Axis := Axis1,
        Done=> , Active=> , Error=> , ErrorID=> , ErrorIdent=> );

END_PROGRAM
```

7.5 Robot-Control - Applikationsbeispiele

7.5.1 RoCo-Grundlagen für Pick & Place-Anwendungen

Allgemein

Der Bewegungsablauf des Tool Center Point (nachfolgend TCP genannt) bei einfachen Pick & Place Anwendungen ist in der Regel wie folgt.

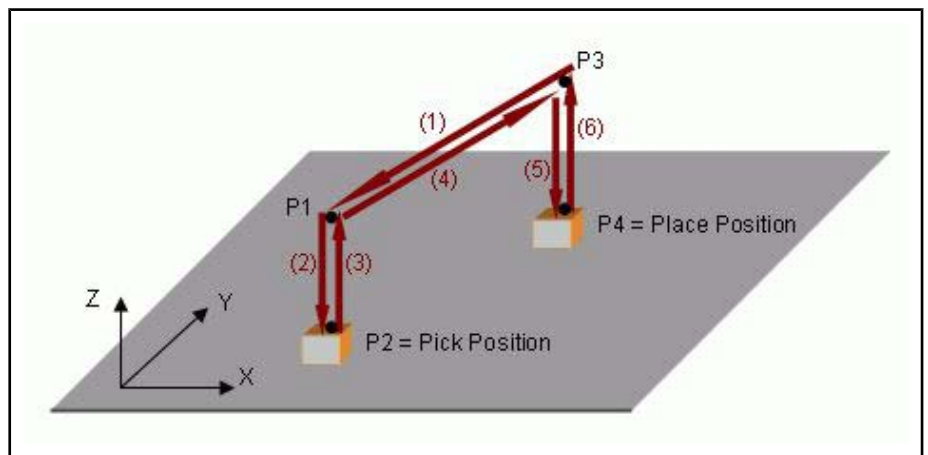


Abb.7-17: Einfache Pick & Place Anwendung

- Beschreibung**
- horizontale Bewegung (1) über den Gegenstand
 - vertikale Bewegung (2) nach unten auf den Gegenstand, um diesen zu greifen
 - vertikale Bewegung (3) mit dem Gegenstand nach oben
 - horizontale Bewegung (4) über die Ablageposition
 - vertikale Bewegung (5) nach unten auf die Ablageposition, um den Gegenstand abzulegen
 - vertikale Bewegung (6) ohne den Gegenstand nach oben
 - warten auf neuen Gegenstand und dann weiter mit (1)

Überschleifen

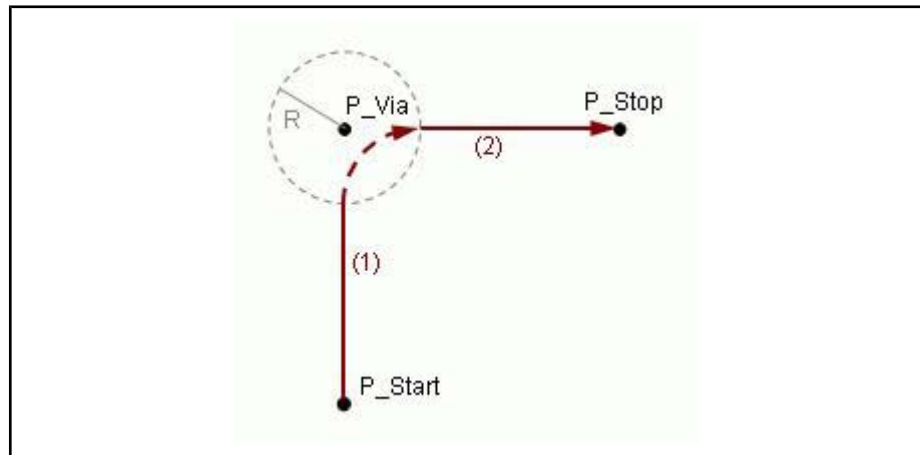
Um den zuvor aufgezeigten Bewegungsablauf harmonischer und damit auch schneller durchführen zu können, besteht die Möglichkeit die einzelnen Bewegungssätze zwischen Gegenstandsaufnahme und Ablage und umgekehrt zu überschleifen.

Erste Schritte - Robot-Control



Überschleifen mit BLOCK_SLOPE bedeutet, dass der nachfolgende Bewegungssatz bei Eintritt in den programmierten Überschleifradius R verfrüht gestartet wird. Innerhalb des Überschleifradius sind beide Bewegungssätze aktiv, d.h. die resultierende Bahnkurve innerhalb der Überschleifkugel ergibt sich aus der Überlagerung der einzelnen Bewegungen.

Liegen die Punkte der überschleifenden Bewegungssätze nicht auf einer Geraden, so werden diese nicht genau durchfahren.



R Überschleifradius

Abb. 7-18: Überschleifen allgemein

Die horizontale Bewegung (2) wird beim Eintritt in den Überschleifradius gestartet.

Innerhalb des Überschleifradius sind beide Verfahrssätze (1) und (2) aktiv.

Die vertikale Bewegung (1) endet erst beim Austritt aus dem Überschleifradius.

Durch das Überschleifen ergibt sich für das oben gezeigte Beispiel folgender Bewegungsablauf. Hierbei werden nur die Punkte P2 und P4 mit "Genau-Halt" angefahren.

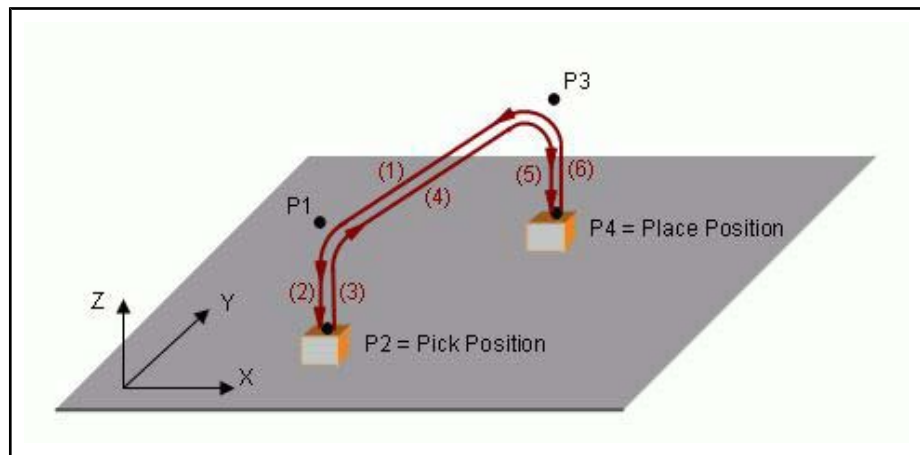


Abb. 7-19: Pick & Place Anwendung mit überschleifenden Bewegungen

Beschreibung

- Horizontale Bewegung (1) über den Gegenstand, die unmittelbar bei Eintritt in den Überschleifradius in die vertikale Bewegung (2) nach unten auf den Gegenstand übergeht.
- Beenden der vertikalen Bewegung (2) nach unten auf den Gegenstand um diesen zu greifen.

Erste Schritte - Robot-Control

- Vertikale Bewegung (3) mit dem Gegenstand nach oben, die unmittelbar bei Eintritt in den Überschleifradius in die horizontale Bewegung (4) übergeht.
- Horizontale Bewegung (4) über die Ablageposition, die unmittelbar bei Eintritt in den Überschleifradius in die vertikale Bewegung (5) nach unten auf die Ablageposition übergeht.
- Beenden der vertikalen Bewegung (5) nach unten auf die Ablageposition um den Gegenstand abzulegen. Überprüfung ob neuer Gegenstand vorhanden ist.
- Wenn neuer Gegenstand vorhanden ist: Vertikale Bewegung (6) ohne den Gegenstand nach oben, die unmittelbar nach Eintritt in den Überschleifradius in die horizontale Bewegung (1) über den Gegenstand übergeht.
- Wenn kein neuer Gegenstand vorhanden ist: Vertikale Bewegung (6) nach P3 und warten auf neuen Gegenstand.



Um mehrere aufeinander folgende Bewegungssätze überschleifend verfahren zu können, müssen einige Randbedingungen erfüllt werden.

Das Überschleifen wird abgebrochen, wenn während der überschleifenden Bewegung folgende Aktionen durchgeführt werden:

- Zugriff auf synchrone Eingänge (POINT; BELT; REAL; INTEGER; BINARY)
- Abfrage der aktuellen Position (POS; @POS)
- Zugriff auf die Bandzähler ("B1")
- Setzen bzw. Rücksetzen der Bandzähler (SYNC)
- Wartezeiten (WAIT)

Bandsynchronisation

Um Gegenstände von einem bewegten Band entnehmen, oder auf einem bewegten Band ablegen zu können, muss mit Bandsynchronisation gearbeitet werden.

Für die Verwendung der Bandsynchronisation müssen folgende Bedingungen erfüllt sein:

- Bandachsen müssen in der Kinematikkonfiguration definiert sein.
- Bandkopplungsfaktoren müssen in der Kinematikkonfiguration definiert sein.
- Der Bandzähler (Kanal 501 - 504) muss im RCL-Programm deklariert werden.
- Bandart 4 muss im RCL-Programm definiert werden (SPC_FCT 21).
- Der Bandsynchrone Arbeitsbereich muss im RCL-Programm definiert werden (SPC_FCT 53).



Zum Aufsynchronisieren auf ein bewegtes Band, muss eine ausreichende Distanz zur Verfügung stehen um den Roboter auf die Geschwindigkeit des Bandes zu beschleunigen. Gleiches gilt auch für das Absynchronisieren.

SYNCHRON

Die Anweisung zum Aktivieren der Bandsynchronisation lautet "SYNCHRON" gefolgt vom Bezeichner des entsprechenden Bandzählers.

Erste Schritte - Robot-Control

Mit dem ersten Bewegungssatz nach der Synchron-Anweisung wird der Roboter auf das Band aufsynchronisiert. Am Ende des Bewegungssatzes ist die Synchronisation abgeschlossen.

In diesem Bewegungssatz muss die Bandkoordinate den Wert des Bandzählers beinhalten, auf den sich die Zielposition bezieht.



Liegt die Position des ersten Bewegungssatzes noch vor dem bandsynchronen Arbeitsbereich, so wird mit der Ausführung gewartet, bis die Position in den bandsynchronen Arbeitsbereich eintritt.



Liegt die Position des ersten Bewegungssatzes bereits hinter der spätest möglichen Start-Position des bandsynchronen Arbeitsbereichs, so wird die Bewegung nicht ausgeführt und das Programm mit Fehler abgebrochen.

SYNCHRON_END

Mit dem letzten Bewegungssatz vor der Anweisung "SYNCHRON_END" wird der Roboter wieder vom Band absynchronisiert. In diesem Bewegungssatz muss der Wert der Bandkoordinate der Zielposition "0" sein.

Für das zuvor gezeigte Beispiel ergibt sich folgender Bewegungsablauf:

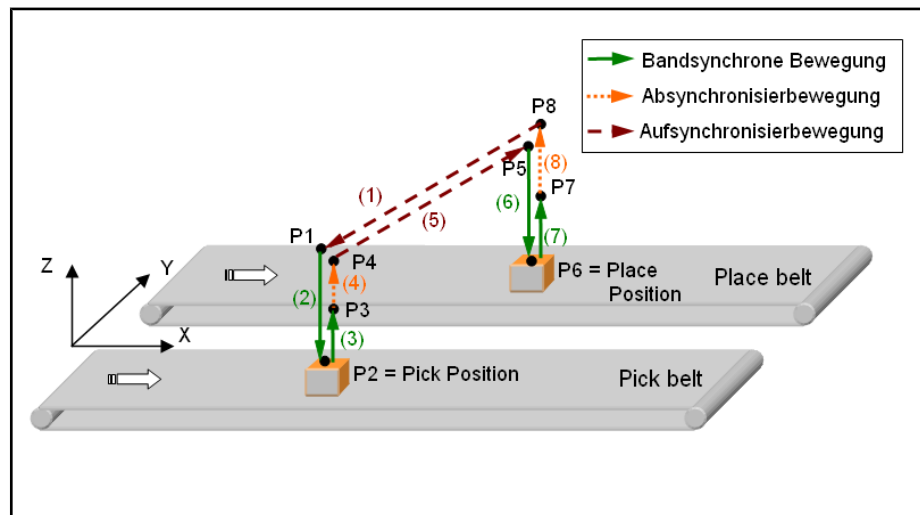


Abb. 7-20: Pick & Place Anwendung mit Bandsynchronisation

Legende

- | | |
|-----------|---|
| P1 | Aufsynchronisierungspunkt für Pick Belt |
| P2 | Bandsynchrone Position des Gegenstands auf Pick Belt |
| P3 | Bandsynchrone Position oberhalb P2 |
| P4 | Absynchronisierungspunkt für Pick Belt = aktuelle Position vor Beginn der Bewegung nach P3, allerdings Z-Koordinate nach oben korrigiert |
| P5 | Aufsynchronisierungspunkt für Place Belt |
| P6 | Bandsynchrone Position des Gegenstands auf Place Belt |
| P7 | Bandsynchrone Position oberhalb P6 |
| P8 | Absynchronisierungspunkt für Place Belt = aktuelle Position vor Beginn der Bewegung nach P7, allerdings Z-Koordinate nach oben korrigiert |

- Beschreibung**
- Aufsynchronisierungsbewegung (1) auf das Band Pick Belt über den Aufsynchronisierungspunkt P1

- Bandsynchrone Bewegung (2) nach P2
 - Greifen des Gegenstands
 - Ermitteln der Absynchronisierposition P4 aus der aktuellen Position



Noch nicht absynchronisieren, da sonst die Bewegung des Greifers relativ zum Band nicht senkrecht ausgeführt wird und somit nachfolgende Gegenstände auf dem Band durch den Gegenstand am Greifer verschoben werden können.

- Bandsynchrone Bewegung (3) nach P3
- Absynchronisierbewegung (4) vom Band Pick Belt über den Absynchronisierpunkt P4
- Aufsynchronisierbewegung (5) auf das Band Place Belt über den Aufsynchronisierpunkt P5
- Bandsynchrone Bewegung (6) nach Punkt P6
 - Ablegen des Gegenstands
 - Ermitteln der Absynchronisierposition P8 aus der aktuellen Position



Noch nicht absynchronisieren, da sonst die Bewegung des Greifers relativ zum Band nicht senkrecht ausgeführt wird und somit ein evtl. vorhandener Karton verschoben werden könnte.

- Bandsynchrone Bewegung (7) nach Punkt P7
- Absynchronisierbewegung (8) vom Band Place Belt über den Absynchronisierpunkt P8
- warten auf neuen Gegenstand und dann weiter mit (1)

Bandsynchronisation mit überschleifenden Bewegungen

Sollen die Bewegungssätze überschleift werden, ergibt sich folgender Bewegungsablauf:

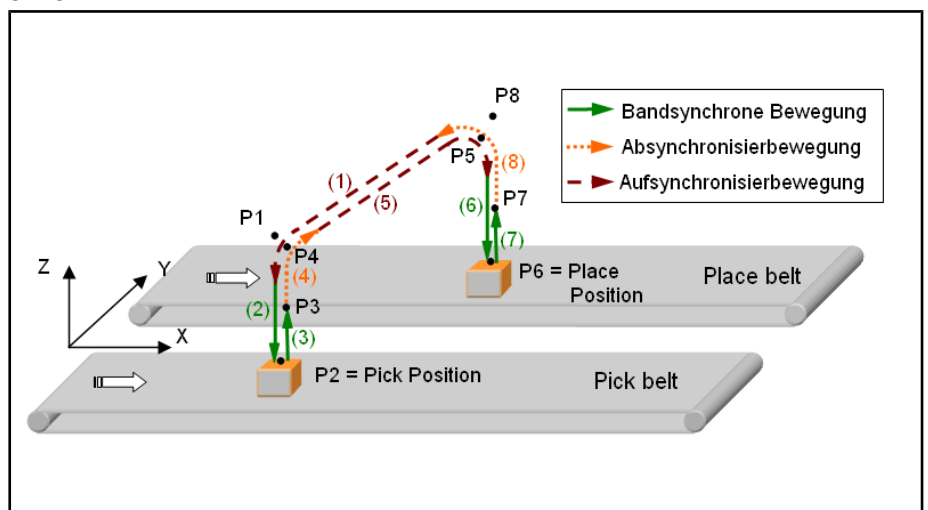


Abb.7-21: Pick&Place-Anwendung mit Bandsynchronisation und überschleifenden Bewegungen

Legende

- P1 Aufsynchronisierpunkt für Pick Belt
- P2 Bandsynchrone Position des Gegenstands auf Pick Belt
- P3 Bandsynchrone Position oberhalb P2

Erste Schritte - Robot-Control

	P4	Absynchronisierpunkt für Pick Belt = aktuelle Position vor Beginn der Bewegung nach P3, allerdings Z-Koordinate nach oben korrigiert
	P5	Aufsynchronisierpunkt für Place Belt
	P6	Bandsynchrone Position des Gegenstands auf Place Belt
	P7	Bandsynchrone Position oberhalb P6
	P8	Absynchronisierpunkt für Place Belt = aktuelle Position vor Beginn der Bewegung nach P7, allerdings Z-Koordinate nach oben korrigiert
	Die Punkte P1; P4; P5 und P8 werden wegen des Überschleifens nicht genau durchfahren.	
Beschreibung	• Aufsynchronisierbewegung (1) auf das Band Pick Belt über den Aufsynchronisierpunkt P1 – Bei Eintritt in den Überschleifradius wird die vertikale Bewegung (2) nach P2 gestartet. Bis zum Austritt aus dem Überschleifradius sind beide Bewegungssätze (1) und (2) aktiv. • Bandsynchrone, nicht überschleifende Bewegung (2) nach P2 – Greifen des Gegenstands – Ermitteln der kompletten Bahn von P2 (Pick) nach P6 (Place)	
		Noch nicht absynchronisieren, da sonst die Bewegung des Greifers relativ zum Band nicht senkrecht ausgeführt wird und somit nachfolgende Gegenstände auf dem Band durch den Gegenstand am Greifer verschoben werden können.
	• Bandsynchrone, überschleifende Bewegung (3) über P3 – Bei Eintritt in den Überschleifradius wird die vertikale Absynchronisierbewegung (4) vom Band Pick Belt über den Absynchronisierpunkt P4 gestartet. Bis zum Austritt aus dem Überschleifradius sind beide Bewegungssätze (3) und (4) aktiv. • Absynchronisierbewegung (4) vom Band Pick Belt über den Absynchronisierpunkt P4 – Bei Eintritt in den Überschleifradius wird die horizontale Aufsynchronisierbewegung (5) auf das Band Place Belt über den Absynchronisierpunkt P5 gestartet. Bis zum Austritt aus dem Überschleifradius sind beide Bewegungssätze (4) und (5) aktiv. • Aufsynchronisierbewegung (5) auf das Band Place Belt über den Aufsynchronisierpunkt P5 – Bei Eintritt in den Überschleifradius wird die vertikale bandsynchrone Bewegung (6) nach P6 gestartet. Bis zum Austritt aus dem Überschleifradius sind beide Bewegungssätze (5) und (6) aktiv. • Bandsynchrone, nicht überschleifende Bewegung (6) nach Punkt P6 – Ablegen des Gegenstands – Überprüfen, ob neuer Gegenstand vorhanden ist Wenn neuer Gegenstand vorhanden ist: Ermitteln der kompletten Bahn von P6 (Place) nach P2 (Pick). Wenn kein neuer Gegenstand vorhanden ist: Ermitteln der Bahn von P6 (Place) bis zur Absynchronisierposition P8 aus der aktuellen Position.	



Noch nicht absynchronisieren, da sonst die Bewegung des Greifers relativ zum Band nicht senkrecht ausgeführt wird und somit ein evtl. vorhandener Karton verschoben werden könnte.

- Bandsynchrone, überschleifende Bewegung (7) über Punkt P7
 - Bei Eintritt in den Überschleifradius wird die vertikale Absynchronisierungsbewegung (8) vom Band Place Belt über den Absynchronisierungspunkt P8 gestartet. Bis zum Austritt aus dem Überschleifradius sind beide Bewegungssätze (7) und (8) aktiv.
- Absynchronisierungsbewegung (8) vom Band Place Belt über den Absynchronisierungspunkt P8

Wenn kein neuer Gegenstand vorhanden ist: warten auf neuen Gegenstand und dann Ermitteln der Bahn von P8 bis P2 (Pick).
- und dann weiter mit (1)

Hinweise zum Überschleifen bei Bandsynchronisation

Um aufeinander folgende Bewegungen überschleifen zu können, müssen folgende Randbedingungen beachtet werden:

- Kommandos, die ein Überschleifen verhindern:
 - Zugriffe auf synchrone Eingänge (Point; Real; Integer; Binary)
 - Zugriffe auf Bandzähler (Belt 501-504)
 - Abfrage der aktuellen Position mittels (POS; @POS)
 - Setzen bzw. "Modulieren" der Bandzähler (SYNC)
 - Wartezeiten (Wait)
 - Pause
- Die Bewegungen zwischen Pick (P2) und Place (P6) (und umgekehrt) können nur dann überschleifen werden, wenn vor Beginn der überschleifenden Bewegung die komplette Bahn ermittelt wurde.

Diese Ermittlung muss also immer dann stattfinden, wenn man sich gerade an der Pick (P2) bzw. Place (P6) Position befindet.
- An der Pick (P2) bzw. Place (P6) Position muss auch die Überprüfung des Bandzählers auf Überschreitung des Modulowerts durchgeführt werden. Die Modulokorrektur des Bandzählers (SYNC-Befehl), darf aber erst nach dem Absynchronisieren P4 bzw. P8 (nicht während einer synchronen Phase) durchgeführt werden.

Grund:

Bandzähler sind synchrone Eingänge (Belt) und verhindern somit das Überschleifen

Realisierung:

Bei Überprüfung lokale Variable setzen wenn Modulowert überschritten wurde und diese Variable anschließend zur Modulokorrektur verwenden.

Einschränkung:

Zur Modulokorrektur muss das "SYNC" Kommando verwendet werden, was ebenfalls ein überschleifen verhindert. Deshalb sollte der Modulowert möglichst Groß eingestellt werden (z.B. 900000 aber nicht mehr als 6 Stellen vor dem Komma, da sonst die kleinen Werte der Bandänderungen innerhalb 2 ms keine Änderung beim aufaddieren bewirken).

Um die oben genannte Einschränkung zu umgehen sollte die Modulokorrektur auch immer dann durchgeführt werden, wenn auf Gegenstände bzw. "Ablageobjekte" gewartet wird.

Erste Schritte - Robot-Control

Beispiel-Programm

Das folgenden RCL-Programm zeigt beispielhaft eine Pick&Place - Anwendung mit überschleifenden Bewegungen bei Bandsynchronisation.

Pick&Place - Anwendung

```

;;COMPILER = RCL3
;;INT = LINEAR
;;KINEMATICS: ( 1 = Kin)
;;kin.JC_NAMES = Ax1,Ax2,Ax3,Ax4,eAx1,eAx2
;;kin.WC_NAMES = Kx,Ky,Kz,Kc,Kpk,Kpl

; *****
; Program description:
; - sample program for belt synchronous pick from infeed belt and belt synchronous
;   place on outfeed belt
; - pick and place positions are transmitted by the PLC
; - all movements between pick and place position (and vice versa) are blending
;   because: - path calculation of all blending moves is done during belt -
;             synchronous non blending movement (at point 'Pick' / 'Place')
;             - check for modulo range of belt and modulo correction of point is
;               done during belt synchronous non blending movement
;               (at point 'Pick' / 'Place')
;             - modulo correction of the belt is done during the non belt -
;               synchronous movement
; *****
PROGRAM Pi_Pl_01
; *****
; *   DECLARATION   *
; *****
VAR:
INPUT POINT :
  701 = xPickPt_i,      ; pick position at pick belt from PLC
  702 = xPlacePt_i     ; place position at place belt from PLC
INPUT REAL :
  201 = rVel_i,         ; commanded velocity from PLC
  202 = rAccel_i        ; commanded acceleration from PLC
INPUT BINARY :
  1001 = bStop_i        ; asynchronous stop signal from PLC
OUTPUT BINARY :
  1 = bFlag_q
CONST :                ; definitions
  FALSE = 0,
  TRUE = 1,
  C_BELT_NO_1 = 1,
  C_BELT_NO_2 = 2,
  C_BK_PTP = 4,         ; belt kind PTP
  MOD_VAL_B1 = 99000.0, ; modulo value for pick belt (mm)
  MOD_VAL_B2 = 99000.0 ; modulo value for place belt (mm)
BELT:
  500 + C_BELT_NO_1 = B1, ; define channel 501 as B1 (pick belt)
  500 + C_BELT_NO_2 = B2 ; define channel 502 as B2 (place belt)
POINT :
  Home,                ; home position
  PickApp,              ; point above point Pick to synchronize to B1
  Pick,                ; pick position
  PickDep,              ; point above point Pick to move synchronous to
  PickDepAsyn,          ; point above POS to unsynchronize from B1
  PlaceApp,             ; point above point Place to synchronize to B2
  Place,               ; place position
  PlaceDep,             ; point above point Place to move synchronous to
  PlaceDepAsyn          ; point above POS to unsynchronize from B2
REAL :
  rTotalLength,        ; variable to setup the belt area
  rBeginLength         ; variable to setup the belt area
BINARY :
  bSyncReq_B1,
  bSyncReq_B2
SPC_FCT :              ; special function to setup the belt kind
  21 = belt_kind(VALUE INTEGER : belt_no ; number of belt to be configured
                 VALUE INTEGER : kind_belt) ; number of belt kind (only value 4)
SPC_FCT :              ; special function to setup the belt area
  53 = belt_area(VALUE INTEGER : BeltNo ; number of belt to be configured
                 JC_POINT : @Base ; start pos. of working area on the belt
                 REAL : rTotalLength ; max. reachable length in belt direction
                 REAL : rBeginLength) ; latest pos. within the working area on
                                     ; the belt, where the synchronization
                                     ; move has to be started
; -----
; *****

```

Erste Schritte - Robot-Control

```
; * IMPLEMENTATION *
; *****
;
BEGIN
  Init          ; call subroutine 'Init'
  WAIT 5        ; wait time in seconds before proceeding with program
  IF rVel_i > 0 THEN BEGIN
    V = rVel_i   ; get commanded velocity from PLC
  END
  IF (rAccel_i > 0) THEN BEGIN
    A = rAccel_i ; get commanded acceleration from PLC
  END
  Pick = xPickPt_i ; get first pick position from PLC
  PickApp = Pick   ; calculate intermediate point 'PickApp' (with Kpk to synchronize)
  PickApp.Kz = Pick.Kz + 100 ; set z coordinate of intermediate point
  LoopMain:      ; begin of main loop
    PickB1       ; call subroutine 'PickB1'
    PlaceB2      ; call subroutine 'PlaceB2'
    If NOT bStop_i THEN BEGIN ; check for asynchronous stop signal from PLC
      JUMP LoopMain ; proceed with main loop
    END
    MOVE TO Home ; move to home position (point 'Home')
  PROGRAM_END

;-----
SUBROUTINE PickB1()
BEGIN
  SYNCHRON B1 ; start synchronisation to pick belt with next movement
  MOVE VIA PickApp ; blending move via point 'PickApp' to synchronize to pick belt
  MOVE TO Pick ; stopping move to point 'Pick' synchronous to pick belt
  ;----- * no blending from here * -----
  WAIT 0.05
  IF rVel_i > 0 THEN BEGIN
    V = rVel_i ; get commanded velocity from PLC
  END
  IF (rAccel_i > 0) THEN BEGIN
    A = rAccel_i ; get commanded acceleration from PLC
  END
  PickDep = Pick ; calculate intermediate point 'PickDep'
  PickDep.Kz = Pick.Kz + 50
  PickDepAsyn = POS ; calculate intermediate point 'PickDepAsyn' (from actual Pos.)
  PickDepAsyn.Kz = Pick.Kz + 100
  PickDepAsyn.kpk = 0 ; set belt value to zero to define synchronization position
  Place = xPlacePt_i
  IF B1 > MOD_VAL_B1 THEN BEGIN
    bSyncReq_B1 = TRUE ; belt modulo has to be done
  END
  ELSE BEGIN
    bSyncReq_B1 = FALSE
  END
  IF Place.kpl > B2 THEN BEGIN
    Place.kpl = Place.kpl - MOD_VAL_B2 ; modulo correction for the place position
  END
  PlaceApp = Place
  PlaceApp.Kz = Place.Kz + 100
  PlaceDep = Place
  PlaceDep.Kz = Place.Kz + 50
  ;----- * no blending till here * -----
  MOVE VIA PickDep ; blending move via point 'PickDep' synchronous to pick belt
  MOVE VIA PickDepAsyn ; blending move via point 'PickDepAsyn' to unsynchronize
  SYNCHRON_END B1
  IF bSyncReq_B1 = TRUE THEN BEGIN
    SYNC B1 >= MOD_VAL_B1 ; belt modulo (subtraction)
  END
  RETURN
SUB_END

;-----
SUBROUTINE PlaceB2()
BEGIN
  SYNCHRON B2
  MOVE VIA PlaceApp
  MOVE TO Place
  ;----- * no blending from here * -----
  WAIT 0.05
  IF rVel_i > 0 THEN BEGIN
    V = rVel_i ; get commanded velocity from PLC
  END
  IF (rAccel_i > 0) THEN BEGIN
    A = rAccel_i ; get commanded acceleration from PLC
  END
  PlaceDepAsyn = POS ; calculate desynchronization point 'PlaceDepAsyn' depending
  ; on actual position
  PlaceDepAsyn.Kz = Place.Kz + 100
```

Erste Schritte - Robot-Control

```

PlaceDepAsyn.kpl = 0
Pick = xPickPt_i ; get next pick position from PLC
IF B2 > MOD_VAL_B2 THEN BEGIN ; check if belt counter has to be modulated
    bSyncReq_B2 = TRUE ; belt modulo has to be done
END
ELSE BEGIN
    bSyncReq_B2 = FALSE
END
IF Pick.kpk > B1 THEN BEGIN
    Pick.kpk = Pick.kpk - MOD_VAL_B1 ; modulo correction for the pick position
END
PickApp = Pick ; calculate intermediate point 'PickApp' (with Kpk to synchronize)
PickApp.Kz = Pick.Kz + 100
PickDep = Pick
PickDep.Kz = Pick.Kz + 50
;----- * no blending till here * -----
MOVE VIA PlaceDep
MOVE VIA PlaceDepAsyn
SYNCHRON_END B2
IF bSyncReq_B2 = TRUE THEN BEGIN ; use 'SYNC' only if B1 > MOD_VAL_B1 otherwise it will
    SYNC B2 >= MOD_VAL_B2 ; avoid blending
    ; belt modulo (subtraction)
END
RETURN
SUB_END

;-----
SUBROUTINE Init()
VAR:
    POINT:
        Base
    JC_POINT: ; joint coordinate point definition
    @Base ; point to setup the startpoint of the belt area
BEGIN
    Home = (0,0,-700,0,0,0) ; define point 'Home'
    V = 1000 ; define path velocity
    A = 1000 ; define path acceleration
    R = 50 ; define maximum blending radius (mm)
    AFACTOR = 1 ; define factor for acceleration (factor for A)
    DFACTOR = 1 ; define factor for deceleration (factor for A)
    BLOCK_SLOPE ; define kind of slope for blending
    belt_kind(C_BELT_NO_1, C_BK_PTP) ; define belt kind for pick belt
    belt_kind(C_BELT_NO_2, C_BK_PTP) ; define belt kind for place belt
    rTotalLength = 600 ; define length of working area over the belt
    rBeginLength = 550 ; define latest position within the working area,
    ; where the synchronization movement has to be started
    Base = (-300,0,-650,0,0,0) ; set point Base in world coordinates to define base
    ; of the pick belt
    @Base = JC(Base) ; convert point Base to joint coordinates and copy to
    ; @Base
    belt_area(C_BELT_NO_1, ; execute special function to setup the working area
        @Base, ; of the pick belt
        rTotalLength,
        rBeginLength)

    Base = (-300, 100, -650, 0, 0, 0) ; set point Base in world coordinates to define base
    ; of the place belt
    @Base = JC(Base) ; convert point Base to joint coordinates and copy to
    ; @Base
    belt_area(C_BELT_NO_2, ; execute special function to setup the working area
        @Base, ; of the place belt
        rTotalLength,
        rBeginLength)

    SYNC B1, bStop_i = FALSE ; set belt counter to 0
    SYNC B2, bStop_i = FALSE ; set belt counter to 0
    MOVE TO Home ; move to home position (point 'Home')
RETURN
SUB_END

```

8 Service und Support

Unser Kundendienst-Helpdesk im Hauptwerk Lohr am Main und der weltweite Service stehen Ihnen mit Rat und Tat zur Seite. Sie erreichen uns täglich **rund um die Uhr - auch am Wochenende und an Feiertagen**.

	Helpdesk	Service-Hotline Welt
Telefon	+49 (0) 9352 40 50 60	Außerhalb Deutschlands nehmen Sie bitte zuerst Kontakt mit unserem für Sie nächstgelegenen Ansprechpartner auf. Die Hotline-Rufnummern entnehmen Sie bitte den Vertriebsadressen im Internet.
Fax	+49 (0) 9352 40 49 41	
E-Mail	service.svc@boschrexroth.de	
Internet	http://www.boschrexroth.com Hier finden Sie auch ergänzende Hinweise zu Service, Reparatur (z.B. Anlieferadressen) und Training.	

Vorbereitung der Informationen

Wir können Ihnen schnell und effizient helfen, wenn Sie folgende Informationen bereithalten:

- detaillierte Beschreibung der Störung und der Umstände
- Angaben auf dem Typenschild der betreffenden Produkte, insbesondere Typenschlüssel und Seriennummern
- Telefon-, Faxnummern und E-Mail-Adresse, unter denen Sie für Rückfragen zu erreichen sind

Index

A

Antriebssystem.....	13
---------------------	----

B

Bestimmungsgemäßer Gebrauch	
Anwendungsbereiche	11
Einleitung	11
Einsatzfälle	11

E

Elektrisches Antriebssystem.....	13
Erste Schritte - FlexProfile.....	149
Bewegungsschritt ausführen,	
ML_FlexProfile	157
Download SPS-Teilprojekt Flexprofile	161
FlexProfile erstellen	151
FlexProfile erstellen über die SPS	154
FlexProfile erstellen über IndraWorks-Di-	
alog	151
Motion-Teilprojekt erstellen	150
Projekt wiederherstellen	164
Technische Voraussetzungen	150
Trace_FlexProfile	163
Workflow Virtuelle Achse → FlexProfile 0 →	
Reale Achse	162
Erste Schritte - IndraLogic.....	25
AS (SFC) editieren	46
Basiseinstellungen zum Gerät	28
Die Steuerung einer Ampelanlage	32
Element einfügen und editieren	79
Erstellen einer neuen Visualisierung	72
Funktionsbaustein einfügen	35
Funktionsmoduleinstellungen	30
FUP editieren	38
Geräteeinstellungen	28
Globale Variablen deklarieren	57
IndraWorks starten	26
Kommunikationseinstellungen	29
Neues Gerät in ein Projekt einfügen	27
Neues IndraWorks-Projekt	26
Neues Projekt erstellen	26
PlcProg in CFC editieren	65
PlcProg in ST editieren	62
POU einfügen	32, 37
Programm einfügen	34
Programme und Funktionsbausteine	32
Programme und Funktionsbausteine edi-	
tieren – zweiter Teil	62
Projekt wiederherstellen	84
SPS-Programm erstellen	32
ST editieren	44
Taskkonfiguration	56
Technische Voraussetzungen	25
Testen im Simulationsmodus	58

E

...Erste Schritte - IndraLogic	
Visualisierung editieren	73
Visualisierung in IndraLogic	72
Visualisierung testen	83
Visualisierung zum SPS-Programm er-	
stellen	72
Visu-Element Anzeige von Variablenwerten	82
Visu-Element Beschriftung	81
Visu-Element Button	80
Visu-Element Ellipse	73
Visu-Element kopieren	76
Visu-Element Rectangle	78
wiederherstellen	84
Erste Schritte - Motion.....	87
Basiseinstellungen zum Gerät	90
Bewegungsablauf programmieren (FUP)	98
Bewegungsablauf programmieren (ST)	121
Download des Motion-Teilprojektes	96
Download des SPS-Projektes RA1 (FUP) .	118
Download des SPS-Projektes RA1 (ST)	142
Download des SPS-Projektes VA1 (FUP) ..	106
Download des SPS-Projektes VA1 (ST)	129
Funktionsmoduleinstellungen	92
Geräteeinstellungen	90
IndraWorks starten	88
Kommunikationseinstellungen	91
Neues Gerät in ein Projekt einfügen	89
Neues Projekt erstellen	88
Programmieren der realen Achse RA1	
(FUP)	115
Programmieren der realen Achse RA1	
(ST)	139
Programmieren der virtuellen Achse VA1	
(FUP)	99
Programmieren der virtuellen Achse VA1	
(ST)	123
Projekt wiederherstellen	145
SERCOS III-Antriebe scannen	94
Technische Voraussetzungen	87
Trace_VA1_RA1 (FUP)	120
Trace_VA1_RA1 (ST)	144
Trace_VA1 (FUP)	110
Trace_VA1 (ST)	134
Virtuelle Achse hinzufügen	95
Werte schreiben, Eingangsvariablen	
(FUP)	108
Werte schreiben, Eingangsvariablen (ST) .	131
Workflow Virtuelle Achse → Elektron-	
isches Getriebe → Reale Achse	119, 143
Erste Schritte - Robot-Control.....	165
Applikationsbeispiele	179
Erforderliche Hard- und Software	165
Pick & Place, Bandsynchronisation	181

Index

E

...Erste Schritte - Robot-Control

Pick & Place, Bandsynchronisation mit überschleifenden Bewegungen	183
Pick & Place, Überschleifen	179
Pick & Place-Anwendungen	179
RoCo-Projekt erstellen	166
RoCoTestV**	177

Exportdatei

Erste Schritte - FlexProfile	164
Erste Schritte - IndraLogic	84
Erste Schritte - Motion	145

N

Nicht-bestimmungsgemäßer Gebrauch.....	12
Folgen, Haftungsausschluss	11

P

PELV	18
------------	----

S

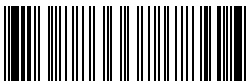
Schutzkleinspannung.....	18
Sicherheitshinweise für elektrische Antriebe und Steuerungen.....	13
Support	
siehe Service-Hotline	189

Z

Zu dieser Dokumentation.....	5
Gültigkeit der Dokumentation	5

Notizen

Bosch Rexroth AG
Electric Drives and Controls
Postfach 13 57
97803 Lohr, Deutschland
Bgm.-Dr.-Nebel-Str. 2
97816 Lohr, Deutschland
Tel. +49 (0)93 52-40-0
Fax +49 (0)93 52-40-48 85
www.boschrexroth.com/electrics



R911333845