

IndraMotion MLC 14VRS

Die ersten Schritte

Inbetriebnahmebeschreibung
R911341516

Ausgabe 02



Titel IndraMotion
MLC 14VRS
Die ersten Schritte

Art der Dokumentation Inbetriebnahmebeschreibung

Dokumentations-Type DOK-MLC***-F*STEP**V14-CO02-DE-P

Interner Ablagevermerk RS-4f210dc5b559587e0a6846a500394072-2-de-DE-14

Änderungsverlauf

Ausgabe	Stand	Bemerkung
Ausgabe 01	2014-05	Erstausgabe
Ausgabe 02	2015-09	Erweiterungen und Verbesserungen

Schutzvermerk © Bosch Rexroth AG 2015

Alle Rechte bei Bosch Rexroth AG, auch für den Fall von Schutzrechtsanmeldungen. Jede Verfügungsbefugnis, wie Kopier- und Weitergaberecht, bei uns.

Verbindlichkeit Die angegebenen Daten dienen allein der Produktbeschreibung und sind nicht als zugesicherte Eigenschaften im Rechtssinne zu verstehen. Änderungen im Inhalt der Dokumentation und Liefermöglichkeiten der Produkte sind vorbehalten.

Redaktion Systemtest, Solution Integration PLC, Safety and Control Hardware, GeKue (KaWa)

Inhaltsverzeichnis

	Seite
1 Zu dieser Dokumentation.....	5
1.1 Gültigkeit der Dokumentation.....	5
1.2 Gliederung der Dokumentation.....	6
1.3 Gebrauch der Sicherheitshinweise.....	6
1.3.1 Aufbau der Sicherheitshinweise.....	6
1.3.2 Erläuterung der Signalwörter und der Signalgrafik.....	6
1.3.3 Verwendete Symbole.....	8
1.3.4 Erläuterung der Signalgrafik auf dem Gerät.....	8
1.4 Erforderliche und ergänzende Dokumentationen XLC/MLC.....	8
1.5 Bezeichnungen und Abkürzungen.....	14
1.6 Kundenfeedback.....	14
2 Erste Schritte – IndraLogic.....	15
2.1 Ziel des Projekts "Erste Schritte – IndraLogic".....	15
2.2 Technische Voraussetzungen.....	15
2.3 Neues IndraWorks-Projekt erstellen.....	16
2.3.1 Steuerung zurücksetzen (CMLx5).....	16
2.3.2 IndraWorks starten.....	16
2.3.3 Neues Projekt erstellen.....	16
2.4 Neues Gerät in ein Projekt einfügen.....	18
2.5 Basiseinstellungen zum Gerät vorgeben.....	18
2.5.1 Geräteeinstellungen.....	18
2.5.2 Kommunikationseinstellungen.....	19
2.5.3 SPS - Eigenschaften.....	21
2.5.4 Funktionspaketauswahl.....	22
2.5.5 Funktionsmoduleinstellungen.....	24
2.6 SPS-Programm erstellen.....	25
2.6.1 Die Steuerung einer Ampelanlage.....	25
2.6.2 Programme und Funktionsbausteine anlegen.....	25
2.6.3 Programme und Funktionsbausteine editieren – erster Teil.....	30
2.6.4 Taskkonfiguration.....	48
2.6.5 Globale Variablen deklarieren.....	49
2.6.6 Testen von "SEQUENCE" im Simulationsmodus.....	50
2.6.7 Programme und Funktionsbausteine editieren – zweiter Teil.....	53
2.6.8 Testen der Applikation auf der Steuerung.....	60
2.7 Visualisierung zum SPS-Programm erstellen.....	62
2.7.1 Visualisierung in IndraLogic.....	62
2.7.2 Erstellen einer neuen Visualisierung.....	63
2.7.3 Visualisierung editieren.....	63
2.7.4 Visualisierung testen.....	74
2.8 Projekt wiederherstellen.....	75

Inhaltsverzeichnis

	Seite
3 Erste Schritte – Motion	79
3.1 Ziel des Projekts "Erste Schritte – Motion"	79
3.2 Technische Voraussetzungen	79
3.3 Anlegen des Motion-Teilprojektes	80
3.3.1 Steuerung zurücksetzen (CMLx5)	80
3.3.2 IndraWorks starten	80
3.3.3 Neues Projekt erstellen	80
3.3.4 Neues Gerät in ein Projekt einfügen	82
3.3.5 Basiseinstellungen zum Gerät vorgeben	82
3.3.6 Sercos III Antriebe (reale Achsen) zum Projekt hinzufügen	89
3.3.7 Virtuelle Achse zum Projekt hinzufügen	90
3.3.8 Download des Motion-Teilprojektes	91
3.4 Programmieren des Bewegungsablaufes in Funktionsbausteinsprache (FUP)	94
3.4.1 Allgemeines	94
3.4.2 Programmieren der virtuellen Achse "vAxis1", Programmiersprache FUP	96
3.4.3 Download des SPS-Teilprojektes virtuelle Achse "vAxis1" (FUP)	103
3.4.4 Aufzeichnung der Bewegung von Achse "vAxis1" (Trace), FUP	106
3.4.5 Programmieren der realen Achse Drive1 (FUP)	112
3.4.6 Download des kompletten SPS-Projektes (FUP)	115
3.4.7 Workflow "Virtuelle Achse → Elektronisches Getriebe → Reale Achse"	115
3.4.8 Aufzeichnung der Bewegung beider Achsen (Trace), FUP	117
3.5 Programmieren des Bewegungsablaufes in Strukturiertem Text (ST)	118
3.5.1 Allgemeines	118
3.5.2 Programmieren der virtuellen Achse "vAxis1", Programmiersprache ST	119
3.5.3 Download des SPS-Teilprojektes virtuelle Achse "vAxis1" (ST)	127
3.5.4 Aufzeichnung der Bewegung von Achse "vAxis1" (Trace), ST	131
3.5.5 Programmieren der realen Achse "Drive1" (ST)	136
3.5.6 Download des kompletten SPS-Projektes (ST)	141
3.5.7 Workflow "Virtuelle Achse → Elektronisches Getriebe → Reale Achse"	141
3.5.8 Aufzeichnung der Bewegung beider Achsen (Trace), ST	142
3.6 Projekt wiederherstellen	143
4 Erste Schritte – FlexProfile	147
4.1 Ziel des Projekts "Erste Schritte – FlexProfile"	147
4.2 Technische Voraussetzungen	148
4.3 Anlegen des Motion-Teilprojektes	148
4.4 Erstellen eines FlexProfiles	150
4.4.1 Erstellen eines FlexProfiles, Allgemeines	150
4.4.2 Erstellen des FlexProfiles über IndraWorks-Dialoge	150
4.4.3 Erstellen des FlexProfiles über die SPS	152
4.5 Komplettierung des SPS-Teilprojektes, Ausführen eines FlexProfile-Bewegungsschrittes	156
4.6 Download und Test des SPS-Teilprojektes	159
4.7 Workflow "Virtuelle Achse → FlexProfile 0 → Reale Achse"	160
4.8 Aufzeichnung der Bewegung beider Achsen (Trace)	161
4.9 Projekt wiederherstellen	162

Inhaltsverzeichnis

	Seite
5 Erste Schritte – Robot-Control V2.....	165
5.1 Ziel des Projekts "Erste Schritte – Robot-Control V2".....	165
5.2 Erforderliche Hard- und Software.....	165
5.3 Vorgehensweise beim Erstellen eines RoCo-Projektes.....	165
5.4 Detail zum Projekt "RoCoV2TestV14".....	177
6 Service und Support.....	183
Index.....	185

1 Zu dieser Dokumentation

1.1 Gültigkeit der Dokumentation

Übersicht über Zielgruppen und Produktphasen

In der folgenden Grafik beziehen sich die umrandeten Aktivitäten, Produktphasen und Zielgruppen auf die vorliegende Dokumentation.

Beispiel: In der Produktphase "Montage (Aufbau)" kann die Zielgruppe "Installateur" mit Hilfe dieser Dokumentation die Aktivität "auspacken, montieren und installieren" ausführen.

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

Erforderliche Qualifikationen: Person, die aufgrund ihrer fachlichen Ausbildung, Kenntnisse und Erfahrungen sowie Kenntnis der einschlägigen Normen und Bestimmungen die ihr übertragenen Arbeiten beurteilen und mögliche Gefahren erkennen kann.

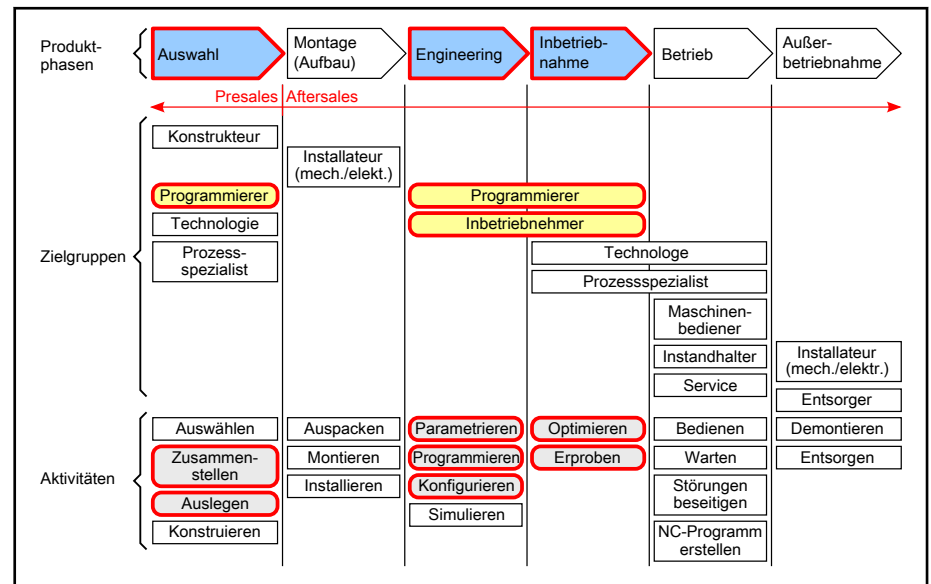


Abb. 1-1: Zuordnung der vorliegenden Dokumentation zu den Zielgruppen, Produktphasen und den Aktivitäten der Zielgruppe

Zweck

Diese Dokumentation enthält folgende Beschreibungen zur IndraMotion MLC 14VRS:

- Erste Schritte – IndraLogic
Grundlagen der Programmierung mit IndraLogic und der Visualisierung
- Erste Schritte – Motion
Grundlagen der Steuerung eines Antriebs aus einem SPS-Programm (PLCopen), Aufzeichnung der Bewegung der Achsen mittels Trace
- Erste Schritte – FlexProfile
Erzeugung und Anwendung von Bewegungsprofilen
- Erste Schritte – Robot-Control V2
Anlegen eines Robot-Control-Projektes bis zum Ausführen des Projektes

Die Dokumentation ist die Grundlage für die Online-Hilfe im System.

Zu dieser Dokumentation



Eine Beschreibung der "Ersten Schritte mit einer SafeLogic-Steuerung" finden Sie in der Dokumentation:

Rexroth IndraWorks

SafeLogic 14VRS

Die Ersten Schritte

Material-Nummer [R911341519](#)

1.2 Gliederung der Dokumentation

Das [Kap. 2 "Erste Schritte – IndraLogic" auf Seite 15](#) beschreibt die Grundlagen der Programmierung mit IndraLogic und der Visualisierung. Das Kapitel ist auch ohne Steuerung im Simulationsmodus (Soft-SPS) nachvollziehbar.

Das [Kap. 3 "Erste Schritte – Motion" auf Seite 79](#) beschreibt die 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.

Das [Kap. 4 "Erste Schritte – FlexProfile" auf Seite 147](#) enthält Informationen zur Erzeugung und Anwendung von Bewegungsprofilen. Das Kapitel baut auf [Kap. 3 "Erste Schritte – Motion" auf Seite 79](#) auf.

Das [Kap. 5 "Erste Schritte – Robot-Control V2" auf Seite 165](#) beschreibt das Anlegen eines Robot-Control V2-Projektes bis zu seiner Ausführung.

Das [Kap. 6 "Service und Support" auf Seite 183](#) enthält Informationen zu unserem Kundendienst-Helpdesk.

1.3 Gebrauch der Sicherheitshinweise

1.3.1 Aufbau der Sicherheitshinweise

Die Sicherheitshinweise sind wie folgt aufgebaut:

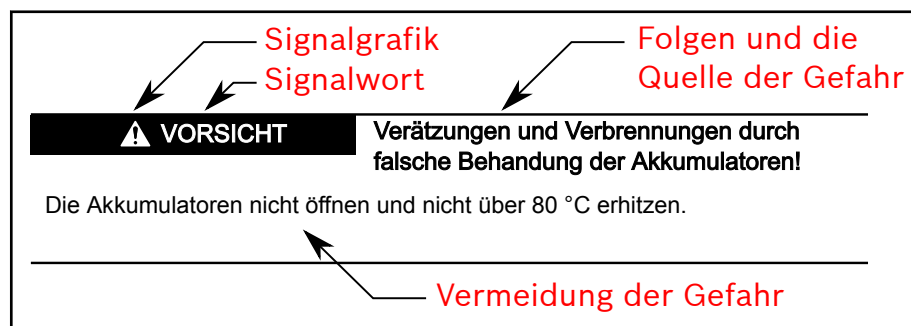


Abb. 1-2: Aufbau der Sicherheitshinweise

1.3.2 Erläuterung der Signalwörter und der Signalgrafik

Die Sicherheitshinweise in der vorliegenden Dokumentation beinhalten bestimmte Signalwörter (Gefahr, Warnung, Vorsicht, Hinweis) und gegebenenfalls 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.

Zu dieser Dokumentation

 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.

Zu dieser Dokumentation

1.3.3 **Verwendete Symbole**

Fingerzeige werden wie folgt dargestellt:



Dies ist ein Hinweis.

Tipps werden wie folgt dargestellt:



Dies ist ein Tipp.

1.3.4 **Erläuterung der Signalgrafik auf dem Gerät**



Beachten Sie vor der Installation und Inbetriebnahme die Dokumentation zu dem Gerät.

1.4 **Erforderliche und ergänzende Dokumentationen XLC/MLC**

Dokumentationstitel mit Typenschlüssel und Materialnummer

Systemübersicht
Rexroth IndraLogic XLC 14VRS Systemübersicht DOK-XLC***-SYSTEM**V14-PRRS-DE-P, R911341482 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.
Rexroth IndraMotion MLC 14VRS Systemübersicht DOK-MLC***-SYSTEM**V14-PRRS-DE-P, R911341503 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.

Tab. 1-1: XLC/MLC-Dokumentationsübersicht - Systemübersicht

Erste Schritte
Rexroth IndraMotion MLC 14VRS Erste Schritte DOK-MLC***-F*STEP**V14-CORS-DE-P, R911341516 Diese Dokumentation beschreibt die "ersten Schritte" mit der IndraMotion MLC und RobotControl. Beschrieben werden die Hard- und Softwarevoraussetzungen und das Erstellen eines Projekts.
Rexroth IndraLogic XLC 14VRS Erste Schritte DOK-XLC***-F*STEP**V14-CORS-DE-P, R911341488 Diese Dokumentation beschreibt die "ersten Schritte" mit der IndraLogic XLC. Beschrieben werden die Hard- und Softwarevoraussetzungen und das Erstellen eines Projekts.
Rexroth IndraMotion MLC 14VRS Projektkonvertierung DOK-XLCMLC-PROCONV*V14-APRS-DE-P, R911341493 Diese Dokumentation beschreibt die Projektkonvertierung der IndraMotion MLC.

Tab. 1-2: XLC/MLC-Dokumentationsübersicht - Erste Schritte

Zu dieser Dokumentation

Engineering
Rexroth IndraWorks 14VRS Software-Installation DOK-IWORKS-SOFTINS*V14-CORS-DE-P, R911344285 Die Dokumentation beschreibt den Installationsprozess von IndraWorks.
Rexroth IndraWorks 14VRS Engineering DOK-IWORKS-ENGINEE*V14-APRS-DE-P, R911343564 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.
Rexroth IndraLogic XLC IndraMotion MLC 14VRS Funktionsbeschreibung DOK-XLCMLC-FUNC***V14-APRS-DE-P, R911341699 Die Dokumentation dient der Beschreibung der Wizards, Kontextmenüs, Dialoge, Steuerungsinbetriebnahme, Gerätekonfiguration, Funktionalitäten der IndraMotion MLC.

Tab. 1-3: XLC/MLC-Dokumentationsübersicht - Engineering

Diagnose und Service
Rexroth IndraLogic XLC IndraMotion MLC 14VRS Diagnosen DOK-XLCMLC-DIAG***V14-RERS-DE-P, R911341480 Diese Dokumentation enthält alle in den Steuerungssystemen IndraLogic XLC und IndraMotion MLC implementierten Steuerungsdiagnosen.
Rexroth IndraLogic XLC IndraMotion MLC 14VRS Parameter DOK-XLCMLC-PARAM***V14-RERS-DE-P, R911341478 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.
Rexroth IndraLogic XLC IndraMotion MLC 14VRS Inbetriebnahme DOK-XLCMLC-STARTUP*V14-CORS-DE-P, R911341500 Das vorliegende Handbuch beschreibt die Inbetriebnahme- und Servicearbeitsschritte für die Systeme IndraLogic XLC und IndraMotion MLC. Es beinhaltet Checklisten für häufig durchzuführende Arbeiten und eine ausführliche Beschreibung der Arbeitsschritte.
Rexroth IndraWorks IndraMotion Service Tool DOK-IWORKS-IMST*****-APRS-DE-P, R911341382 Diese Dokumentation beschreibt das IndraMotion Service Tool (IMST). Das IMST ist ein webbasiertes Diagnosewerkzeug, das zum Zugriff auf ein Steuerungssystem über eine Ethernet-Hochgeschwindigkeitsverbindung dient. Das IMST ermöglicht OEMs, Endanwendern und Kundenbetreuern von überall auf ein System zuzugreifen und Ferndiagnosen zu stellen. Voraussetzung ist ein PC mit Internet Explorer Version 8 oder Firefox 3.5 oder eine höhere Version. Die folgenden Steuerungs-Varianten werden unterstützt: • IndraMotion MLC L25/L45/L65/L75/XM2/VPx • IndraLogic XLC L25/L45/L65/L75

Tab. 1-4: XLC/MLC-Dokumentationsübersicht - Diagnose und Service

Zu dieser Dokumentation

Antriebsregelgeräte
Rexroth IndraDrive Firmware für Antriebsregelgeräte MPH-, MPB-, MPD-, MPC-08 DOK-INDRV*-MP*-08VRS**-APxx-DE-P-DE-P, R911332642 Diese Dokumentation beschreibt alle funktionellen Eigenschaften der IndraDrive-Firmware in den Varianten MPH-08, MPB-08, MPD-08 und MPC-08.
Rexroth IndraDrive Antriebsregelgeräte MPx-02 bis MPx-08 DOK-INDRV*-GEN-**VRS**-PAxx-DE-P, R911297316 Diese Dokumentation enthält die Beschreibung aller, in der Firmware für Antriebsregelgeräte der IndraDrive-Familie implementierten Parameter. Sie unterstützt die Parametrierung der Antriebsregelgeräte.
Rexroth IndraDrive MPx-02 bis MPx-08 und HMV DOK-INDRV*-GEN-**VRS**-WAxx-DE-P, R911297318 Diese Dokumentation enthält die Beschreibung aller in den folgenden Firmwares implementierten Diagnosen: • Antriebsregelgerät-Firmwares MPx-02 bis MPx-08 und • Firmwares der Versorgungsgeräte des Typs "HMV". Sie unterstützt den Maschinenbediener und den Anlagenprogrammierer bei der Fehlerbehebung.
Rexroth IndraDrive Steuerteile CSB02, CSE02, CSH02, CDB02 DOK-INDRV*-CXX02*****-PRRS-DE-P, R911338961
Rexroth IndraDrive MPx-18 Funktionen DOK-INDRV*-MP*-18VRS**-APRS-DE-P, R911338674 Diese Dokumentation beschreibt alle funktionellen Eigenschaften der IndraDrive-Firmware in den Varianten MPB-18, MPM-18, MPC-18 und MPE-18.

xx entsprechender Ausgabestand

Tab. 1-5: XLC/MLC-Dokumentationsübersicht - Antriebsregelgeräte

SPS
Rexroth IndraWorks 14VRS IndraLogic 2G SPS-Programmiersystem DOK-IWORKS-IL2GPRO*V14-APRS-DE-P, R911343570 Diese Dokumentation beschreibt das SPS-Programmiersystem IndraLogic 2G und seine Verwendung. Beschrieben werden die Basisverwendung, Erste Schritte, die Visualisierung, Menüpunkte und die Editoren.
Rexroth IndraWorks 14VRS Basis-Bibliotheken IndraLogic 2G DOK-IL*2G*-BASLIB**V14-LIRS-DE-P, R911343918 Das vorliegende Handbuch beschreibt die systemübergreifenden SPS-Bibliotheken.
Rexroth IndraLogic XLC IndraMotion MLC 14VRS PLCopen-Bibliotheken DOK-XLCMLC-FUNLIB**V14-LIRS-DE-P, R911341490 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.

Tab. 1-6: XLC/MLC-Dokumentationsübersicht - SPS

Zu dieser Dokumentation

Feldbusse
Rexroth IndraWorks 14VRS Feldbusse DOK-IWORKS-FB*****V14-APRS-DE-P, R911341484 Die Dokumentation dient der Beschreibung der von den Systemen IndraLogic XLC, IndraMotion MLC und IndraMotion MTX unterstützten Feldbus- und lokalen Peripherieanschlüssen. Der Schwerpunkt dieser Dokumentation liegt in der Konfiguration, Parametrierung, Inbetriebnahme und Diagnose der verschiedenen Peripherieanschlüssen. Sie ist die Grundlage für die Online-Hilfe.
Rexroth IndraWorks 14VRS Feldbus-Bibliotheken DOK-IWORKS-FB*LIB**V14-LIRS-DE-P, R911343574 Das vorliegende Handbuch beschreibt die Bibliotheken der Feldbusse: RIL_ProfibusDP_02, RIL_ProfibusDP Slave, RIL_ProfinetIO, RIL_ProfinetIODevice, RIL_EtherNetIPAdapter, RIL_MappingList, RIL_SERCOSIII, RIL_Inline, einschließlich deren Diagnosen, sowie die Fehlerreaktion der Bausteine.
Rexroth IndraWorks 12VRS FDT-Container DOK-IWORKS-FDT*CON*V12-APRS-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.
Sercos Systemhandbuch für I/O-Geräte DOK-CTRL-ILS3*****-APxx-DE-P, R911333511 Diese Dokumentation beschreibt die Projektierung, Parametrierung, Inbetriebnahme und Diagnose von I/O-Geräten mit Sercos Schnittstelle.

xx entsprechender Ausgabestand
Tab. 1-7: XLC/MLC-Dokumentationsübersicht - Feldbusse

HMI
Rexroth IndraWorks 14VRS HMI DOK-IWORKS-HMI*****V14-APRS-DE-P, R911343567 Diese Dokumentation beschreibt die Funktionen, die Konfiguration und die Bedienung der Bedienoberflächen IndraWorks HMI Engineering und IndraWorks HMI Operation.
Rexroth IndraWorks 14VRS WinStudio DOK-IWORKS-WINSTUD*V14-APRS-DE-P, R911341584 Dieses "Benutzerhandbuch und Technische Referenz" soll Sie dabei unterstützen, mit Ihrer Software "Rexroth WinStudio" optimale Ergebnisse zu erzielen. Das Dokument bietet technische Informationen und Schritt-für-Schritt-Anleitungen zum Anlegen von webfähigen HMI/SCADA-Programmen.
Rexroth IndraLogic XLC IndraMotion MLC 14VRS HMI-Anbindung DOK-XLCMLC-HMI*****V14-APRS-DE-P, R911341496 Dieses Dokument beschreibt die von der IndraLogic XLC und IndraMotion MLC unterstützten Visualisierungssysteme sowie deren Anbindung.

Tab. 1-8: XLC/MLC-Dokumentationsübersicht - HMI

Zu dieser Dokumentation

Technologie
Rexroth IndraLogic XLC IndraMotion MLC 14VRS Generic Application Template DOK-XLCMLC-TF*GAT**V14-APRS-DE-P, R911341486 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.
Rexroth IndraMotion MLC 14VRS Technologie-Bibliotheken DOK-MLC***-TF*LIB**V14-LIRS-DE-P, R911341510 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.
Rexroth IndraWorks Energy Efficiency Management 14VRS DOK-IWORKS-4EE*****V14-APRS-DE-P, R911339228 Diese Dokumentation beschreibt die HMI-Anbindung des Energiemanagement-Systems und die Funktionsbausteine, Funktionen und Datentypen der Bibliothek "RIL_4EE ". Die Dokumentation enthält ferner die Fehlerreaktion der Bausteine.
Rexroth IndraMotion MLC 14VRS RegisterControl (Bibliothek) DOK-MLC***-REGI*CO*V14-LIRS-DE-P, R911341508 Diese Dokumentation beschreibt die Ein- und Ausgänge der einzelnen Funktionsbausteine und gibt Hinweise für deren Verwendung.
Rexroth IndraMotion MLC 14VRS RegisterControl (Anwendungsbeschreibung) DOK-MLC***-REGI*CO*V14-APRS-DE-P, R911341506 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.
Rexroth IndraMotion MLC 14VRS Anwendung Wicklerfunktionen DOK-MLC***-TF*WIND*V14-APRS-DE-P, R911341512 Diese anwendungsbezogene Systemdokumentation beschreibt die Anwendung der Wickler-Technologiefunktionen.
Rexroth IndraWorks 13VRS CamBuilder DOK-IWORKS-CAMBUIL*V13-APRS-DE-P, R911336290 Diese Dokumentation dient zur Beschreibung der Grundlagen und der Bedienung des CamBuilder, des Werkzeugs zur Kurvenscheibenbearbeitung.
Rexroth IndraMotion MLC 14VRS Robot-Control V2 DOK-MLC***-ROCO*****V14-RERS-DE-P, R911341586 Die Dokumentation informiert über die Robot-Control V2. Die Programmierung über PLCopen steht hierbei im Vordergrund. Beschrieben werden die Programmstruktur, Variablen, Funktionen, Bewegungsanweisungen sowie die notwendigen Parameter des Systems. Die Dokumentation enthält ferner die Beschreibung der Bibliothek "ML_Robot". Robot-Control V2 wird unterstützt durch ein Kinematik-Interface. Die Komponenten dieses Interfaces werden in der Bibliothek "ML_KinTech" beschrieben.

Tab. 1-9: XLC/MLC-Dokumentationsübersicht - Technologie

Zu dieser Dokumentation

Open Core Engineering - OCE
Rexroth IndraLogic XLC IndraMotion MLC 14VRS Erste Schritte MLPI DOK-XLCMLC-MLPI****V14-CORS-DE-P, R911341032 Dieses Dokument beschreibt die Installation und Inbetriebnahme der MLPI-Schnittstelle.
Rexroth IndraLogic XLC IndraMotion MLC 14VRS Automation Interface DOK-XLCMLC-AUT*INT*V14-APRS-DE-P, R911341498 Diese Dokumentation beschreibt den script-basierten Zugriff auf IndraWorks-Projektdaten über die Schnittstelle des Automation Interfaces.

Tab. 1-10: XLC/MLC-Dokumentationsübersicht - OCE

SafeLogic
Rexroth IndraWorks SafeLogic 14VRS Die ersten Schritte DOK-IWORKS-SL*STEP*V14-CORS-DE-P, R911341519 Dieses Dokument beschreibt die Erstinbetriebnahme des Safety-Funktionsmoduls und dessen externer Safety-Peripherie. Am Beispiel eines Projektes werden alle notwendigen Schritte zur Inbetriebnahme einer einfachen Safety-Anwendung durchgeführt.
Rexroth IndraMotion MLC IndraLogic XLC 14VRS SafeLogic Systemübersicht DOK-XLCMLC-SL**SYS*V14-PRRS-DE-P, R911341695 Dieses Dokument beschreibt die Projektierung, die Erstellung und die Inbetriebnahme von sicherheitstechnischen Vorrichtungen.
Rexroth IndraWorks 14VRS SafeLogic Projektkonfiguration DOK-IWORKS-SL**PRJ*V14-APRS-DE-P, R911341693 Dieses Dokument beschreibt die Projektierung von Safety-Anwendungen.
Rexroth IndraControl SafeLogic Funktionsmodul DOK-CTRL-SL**FM****-ITRS-DE-P, R911336575 Dieses Dokument liegt der Hardware bei und beschreibt die Inbetriebnahme, die Installation, die Außerbetriebnahme und Entsorgung der Hardware.

Tab. 1-11: XLC/MLC-Dokumentationsübersicht - SafeLogic

Hardware
Rexroth IndraControl L45/L65/L85 Steuerung DOK-CTRL-ICL45L65L85-PRxx-DE-P, R911332115 Diese Dokumentation dient der Beschreibung der Steuerungen IndraControl L45/L65/L85.
Rexroth IndraControl L25 DOK-CTRL-IC*L25****-PRxx-DE-P, R911328473 Diese Dokumentation dient der Beschreibung der Steuerungen IndraControl L25.
Rexroth IndraControl Lxx 14VRS Funktionsmodule DOK-CTRL-FM*LXX**V14-APRS-DE-P, R911341582 Diese Dokumentation dient der Beschreibung aller Funktionsmodule der Lxx Steuerungen inkl. Engineering und Diagnose.

xx entsprechender Ausgabestand

Tab. 1-12: XLC/MLC-Dokumentationsübersicht - Hardware

Zu dieser Dokumentation

Hydraulik
Rexroth IndraMotion MLC 14VRS Hydraulische Funktionen DOK-MLC***-TF*HMOT*V14-LIRS-DE-P, R911341905 Diese Dokumentation beschreibt die Funktionsbausteine, Funktionen und Datentypen der Bibliotheken MH_TechHydrBase und MH_TechHydrMotion. Die Dokumentation enthält ferner die Fehlerreaktion der Bausteine.
Rexroth IndraMotion MLC 14VRS Sequentielle Programmierung DOK-XLCMLC-SEQPROG*V14-LIRS-DE-P, R911341908 Diese Dokumentation beschreibt die Möglichkeiten der Sequentiellen Programmierung, auch in Verbindung mit GAT, und die Bibliothek MH_HydrControlProg.

Tab. 1-13: XLC/MLC-Dokumentationsübersicht - Hydraulik

1.5 Bezeichnungen und Abkürzungen

Begriff	Erklärung
IndraMotion MLC	kompaktes Motion-Logic-System mit Motion-, Robot- und Logic-Control-Funktionalität
IndraWorks Engineering Framework	Projektierungs- und Inbetriebnahmetool von Bosch Rexroth
IndraDrive	Antriebsregelgerät
Wizard	Assistent, der den Anwender durch mehrere Dialoge für eine ergonomische Dateneingabe führt
Funktionsbaustein	stellt applikationsspezifische Funktionalitäten zur Verfügung
IEC	International Electrotechnical Commission (Normungsgremium für Elektrotechnik)
Sercos	Sercos (Serial realtime communication system) interface ist eine weltweit genormte digitale Schnittstelle zur Kommunikation zwischen Steuerungen und Antrieben
XML	Die Extensible Markup Language (engl. für "erweiterbare Auszeichnungssprache"), abgekürzt XML, ist eine Auszeichnungssprache zur Darstellung hierarchisch strukturierter Daten in Form von Textdaten

Tab. 1-14: Verwendete Bezeichnungen und Abkürzungen

1.6 Kundenfeedback

Anregungen, Wünsche oder Verbesserungen von unseren Kunden haben bei uns einen hohen Stellenwert. Senden Sie uns Ihre Anmerkungen zu den Dokumentationen per E-Mail an Feedback.Documentation@boschrexroth.de. Sie können direkt im elektronischen PDF-Dokument Kommentare einfügen und uns die PDF-Datei zusenden.

2 Erste Schritte – IndraLogic

2.1 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 der Bedienabläufe vorgehen, werden Sie mit der Benutzung von IndraWorks vertraut und erkennen, wie die einzelnen Komponenten zusammenspielen.

Projektwiederherstellung:

Unabhängig von Ihrer Hardware, z. B. IndraMotion MLC L25, L45, L65, L75, XM2 oder VPx, können Sie das Projekt auf Ihrem Rechner mit Ihrer Steuerung wiederherstellen.

Nutzen Sie dazu die in Ihrem Installationsverzeichnis unter "<Laufwerk>:\Programme\Rexroth\IndraWorks\Projects" abgelegte Exportdatei "Application_FirstSteps_IndraLogic_Basics.iwx". Detaillierte Erläuterungen finden Sie unter [Kap. 2.8 "Projekt wiederherstellen" auf Seite 75](#).



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

2.2 Technische Voraussetzungen

- Software:
 - PC mit installiertem IndraWorks Engineering für IndraMotion MLC 14VRS
- Steuerung:
 - IndraMotion MLC L25
 - IndraMotion MLC L45
 - IndraMotion MLC L65
 - IndraMotion MLC L75
 - IndraMotion MLC XM2
 - IndraMotion MLC VPx

Die Schaffung dieser Voraussetzungen sind beschrieben in:

- IndraMotion MLC 14VRS Funktionsbeschreibung, Kapitel "Inbetriebnahme und Bedienung"

Erste Schritte – IndraLogic

Eine Steuerung, z. B. Typ "IndraMotion MLC L65" mit aktueller Firmware 14VRS, muss vorhanden sein. Zusätzlich müssen noch folgende Voraussetzungen erfüllt sein:

- Die IP-Adresse ist bekannt
- Die Steuerung ist über Ethernet mit dem Programmiergerät (PC) verbunden
- Auf der Steuerung ist eine gültige Systemfirmware geladen
- Falls ein Sercos III Bus vorhanden ist, muss der Bus gebrückt werden

2.3 Neues IndraWorks-Projekt erstellen

2.3.1 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.

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 Anzeige **BOOTSTOP** erscheint.

Danach sind die beiden inneren Tasten zu drücken. Es erscheint die Anzeige **CLEANUP?**.

Bestätigen Sie mit <Enter>. Es erscheint die Anzeige **CLEANUP**. Wenn der CleanUp erfolgreich war, erscheint die Anzeige **SUCCESS** und dann **REBOOT**.

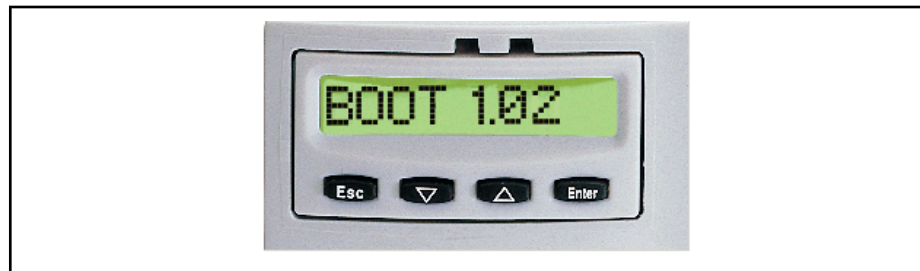


Abb. 2-1: Reboot der Steuerung

Die Steuerung ohne Sercos Interface oder bei gebrücktem Sercos Interface läuft nun bis **BB RDY**. Der Speicher ist gelöscht, eine eventuell vorhandene Motion-Komponente und die SPS-Komponente sind betriebsbereit.

2.3.2 IndraWorks starten

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

Die Entwicklungsumgebung von IndraWorks wird gestartet. Alternativ kann

IndraWorks über das Desktop-Symbol  gestartet werden.

2.3.3 Neues Projekt erstellen

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

Erste Schritte – IndraLogic

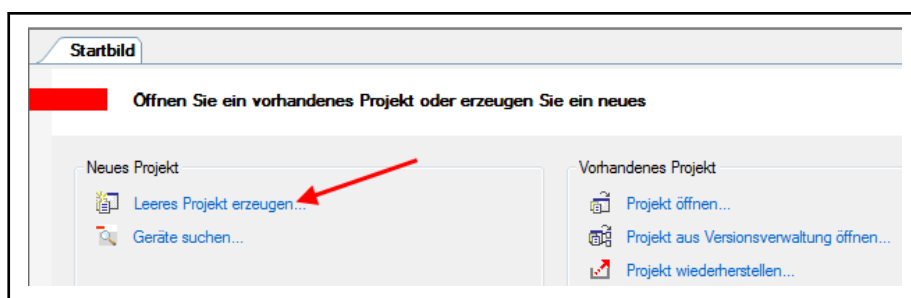


Abb. 2-2: Leeres Projekt erzeugen

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 nachfolgende Abbildung).

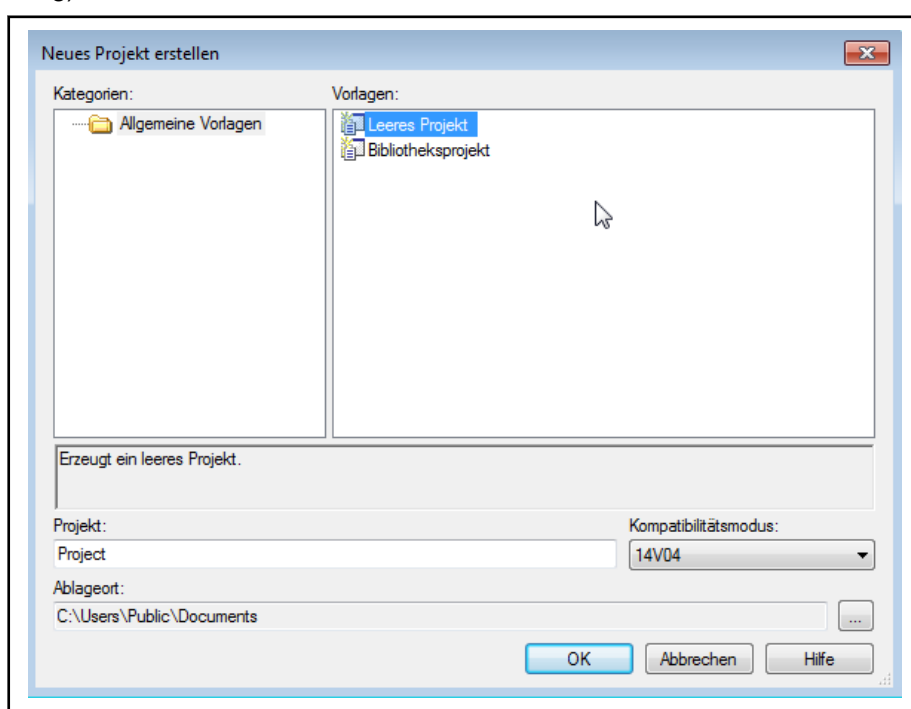


Abb. 2-3: Einstellungen zum Abspeichern eines neuen IndraWorks-Projektes

- | | |
|----------------------------|--|
| Projektname | Geben Sie dem Projekt einen beliebigen Namen, Vorschlag: "Erste Schritte". |
| Ablageort | Das Feld enthält voreingestellt einen Speicherplatz im Verzeichnis "C:\Dokumente und Einstellungen\<username>\Eigene Dateien". Alternativ können Sie auch ein anderes Verzeichnis angeben. |
| Kompatibilitätsmode | Der Leistungsumfang der IndraWorks-Oberfläche entspricht der Version 14V04. |
| Projektsprache | Mit der Auswahl der Projektsprache legen Sie die Sprache ihrer Anwender-texte fest. |

Das neu angelegte Projekt erscheint im Projektexplorer mit dem vorher definierten Projektnamen:



Abb. 2-4: Projekt "Erste Schritte"

Erste Schritte – IndraLogic

2.4 Neues Gerät in ein Projekt einfügen

Ziehen Sie aus der Bibliothek "Drive and Control" das gewünschte Gerät (z. B. Ordner "IndraMotion MLC", "IndraControl L65") in das neu angelegte Projekt im Projektexplorer.



(1) Gerät: IndraMotion MLC L65

(2) Projekt: Erste Schritte

Abb. 2-5: Gerät einfügen

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

2.5 Basiseinstellungen zum Gerät vorgeben

2.5.1 Geräteeinstellungen

Geben Sie die allgemeinen Daten zum eingefügten Gerät an.

Abb. 2-6: Geräteeinstellungen

- Gerätename** Geben Sie bei "Gerätename" einen beliebigen Gerätenamen ein, z. B. "Mlc1". 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 >>**.

2.5.2 Kommunikationseinstellungen

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

Abb. 2-7: Anlegen einer Steuerung (im Beispiel "IndraMotion MLC L65") – Hardware / Kommunikation

Gerätetyp	Der Gerätetyp ist durch die Steuerungsauswahl vorbestimmt. Gibt es für einen Gerätetyp mehrere unterschiedliche Varianten, muss aus der Drop-Down-Liste die Geräte-Variante ausgewählt werden.
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 Kap. 2.6.6 "Testen von SEQUENCE im Simulationsmodus" auf Seite 50 nicht ausgewertet. Wird die Steuerung im Simulationsmodus betrieben, kann der Verbindungstest entfallen.
SPS-Gateway	Geben Sie unter "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). "localhost" bedeutet, dass das lokal installierte Gateway angesprochen wird. Diese Einstellung sollte nicht verändert werden.
SPS-Kommunikation	In der Version 14VRS wird nur noch TCP an der Engineering-Schnittstelle unterstützt.

Erste Schritte – IndraLogic

- UDP wird noch aus Kompatibilitätsgründen für Projekte aus der Version 13VRS angeboten.
- Verbindungstest** Klicken Sie auf die Schaltfläche **Ausführen**, um zu überprüfen, ob die Kommunikation zwischen Rechner und der Zielsteuerung erfolgreich ist. Das Ergebnis des Tests wird im Feld rechts neben "Verbindungstest" protokolliert. Bei erfolgreichem SPS-Kommunikationstest wird die IP-Adresse angezeigt. Eventuell auftretende Probleme werden im Fenster durch blinkende rote oder gelbe Icons gekennzeichnet.

Die folgenden Probleme können auftreten:

- Ausgabe: "Kommunikation zur Steuerung <Steuerungsname> mit der IP-Adresse <IP-Adresse> 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 (IndraMotion MLC 14VRS Funktionsbeschreibung, Kapitel "Inbetriebnahme und Bedienung", Unterkapitel "Einstellen der IP-Adresse")
 - Die Netzwerkmaske des PC und/oder der Steuerung ist falsch eingestellt
Lösung: IP-Adresse im Dialog ändern oder IP-Adresse der Steuerung ändern (IndraMotion MLC 14VRS Funktionsbeschreibung, Kapitel "Inbetriebnahme und Bedienung", Unterkapitel "Einstellen der IP-Adresse")
- Ausgabe: "Die projektierte Firmware stimmt nicht mit der Firmware in der Steuerung überein."
Ursachen:
 - Die Firmware der Steuerung ist nicht aktuell
Lösung: Firmwareupdate durchführen (IndraMotion MLC 14VRS Funktionsbeschreibung, Kapitel "Steuerung", Unterkapitel "Firmwareverwaltung")
 - Die ausgewählte Steuerung entspricht nicht dem realen Steuerungstyp

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

2.5.3 SPS - Eigenschaften

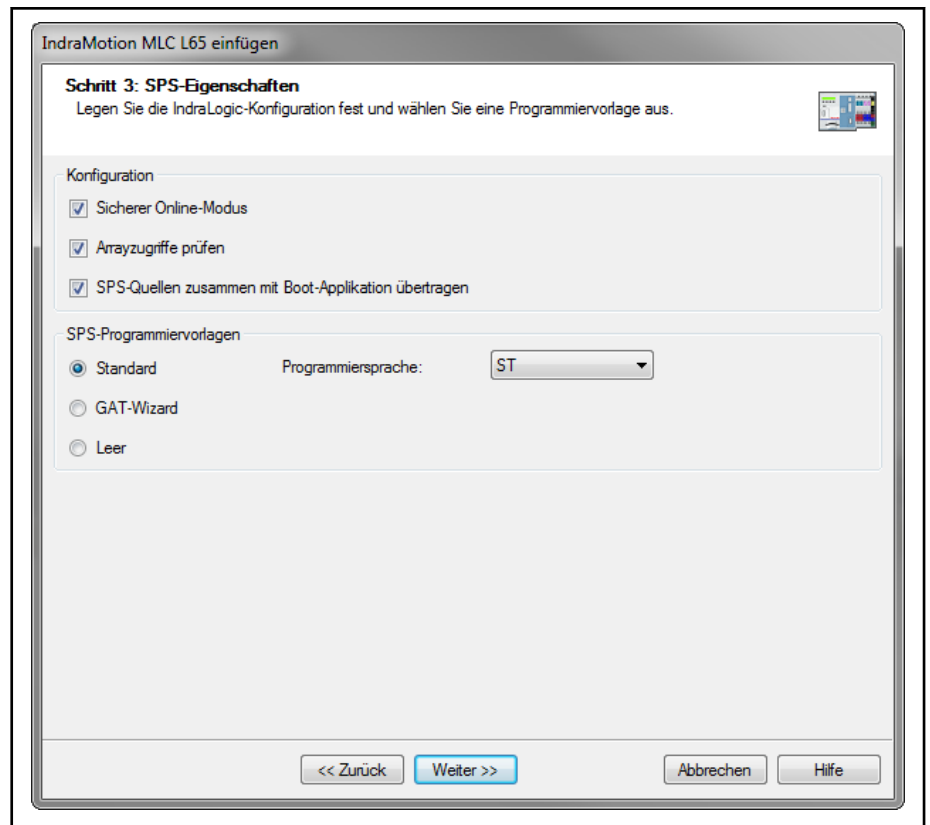


Abb. 2-8: Anlegen einer Steuerung – SPS-Eigenschaften

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

Arrayzugriffe prüfen Ist dieser Schalter aktiviert, dann wird automatisch die Checkfunktion "CheckBounds" aus der Vorlagendatenbank unterhalb des SPS-Applikationsknotens eingefügt.

Wird dieser Schalter bei einer bereits bestehenden Steuerung wieder deaktiviert, so wird die Checkfunktion "CheckBounds" nur dann automatisch wieder entfernt, sofern der folgende Kommentarblock im Implementierungsteil der Checkfunktion noch enthalten ist:

```
(*****  
This function was generated by the 'check array access'  
option of the controllers 'properties' dialog box. If you  
delete this comment, this function will not be removed by  
disabling the 'check array access' option.  
*****)
```

Durch Entfernen des Kommentarblocks kann verhindert werden, dass kundenspezifische Anpassungen im Implementierungsteil der Checkfunktion durch Deaktivieren des Schalters gelöscht werden.

Erste Schritte – IndraLogic



Da mit dem Löschen des Kommentarblockes die automatisch eingefügte Checkfunktion nicht mehr entfernt wird, muss die Funktion "CheckBounds" manuell aus der Applikation gelöscht werden, um die Überwachung der Arraygrenzen zu deaktivieren.

Ein nichtaktivierter Schalter bedeutet somit nicht, dass keine CheckBounds-Funktionalität aktiv ist.

Wird die Checkfunktion eingefügt oder gelöscht, ist kein Einloggen per Onlinechange möglich. Es wird ein Download notwendig.

SPS-Quellen zusammen mit Boot-Applikation übertragen

Die Checkbox ist nur dann aktiv, wenn der SPS-Applikationstyp "Standard" oder "GAT Wizard" aktiviert und keine benutzerdefinierte CheckBounds-Funktion enthalten ist.

Beim Übertragen der Boot-Applikation werden zusätzlich zum Kompilat auch die SPS-Programmquellen (POUs, Tasks usw.) auf die Steuerung übertragen. Verwenden Sie diese Funktion, wenn Sie im Online-Vergleich zusätzlich einen Detailvergleich für SPS-Elemente durchführen wollen.

Programmiervorlagen

Für die zu erzeugende SPS-Applikation kann zwischen folgenden Varianten gewählt werden:

Standard: Es werden automatisch leere Standardprogramme "MotionProg" und "PlcProg" in der gewählten Programmiersprache angelegt. Für weitere SPS-Objekte wird die Programmiersprache beim Anlegen des Objektes ausgewählt.

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

GAT Wizard: In einem zusätzlichen Dialog können Beispielprojekte oder Programmieremplates gewählt werden. Das GAT ist ein lauffähiger Programmierrahmen mit vordefinierten Stellen für Applikationscode. Durch die dialogbasierte Codegenerierung des GAT-Wizards entstehen automatisch professionell gut strukturierte Applikationsprogramme mit vorbereiteten Elementen.

Leer: legt eine leere, nicht lauffähige SPS-Applikation ohne Programme und Tasks an.

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

2.5.4 Funktionspaketauswahl



Die Funktionspakete sind bei den Steuerungen auf Basis IndraControl L25/L45/L65/L75 durch die erworbene Firmware bereits vordefiniert und daher nicht selektierbar. Bei den Steuerungen auf Basis IndraControl XM und VPx sind die Funktionspakete anhand der erworbenen Lizenzen auszuwählen!

1. Wird ein Funktionspaket abgewählt, werden automatisch alle davon abhängigen Funktionspakete ebenfalls abgewählt.
2. Selektiert man ein lizenzpflichtiges Funktionspaket, wird automatisch ein Lizenzicon generiert. Über dieses Icon wird per Tooltip oder per Mausklick die Lizenzierungsinformationen angezeigt.
3. In Abhängigkeit von den aktivierten Funktionspaketen werden anschließend im IndraWorks Projektbaum die Knoten unterhalb der angelegten Steuerung aufgebaut.
4. Die Funktionspakete können auch nach dem Anlegen einer Steuerung über den zugehörigen Dialog "Eigenschaften" hinzugefügt bzw. entfernt werden.

- Bei einer Konvertierung des Gerätes in einen Steuerungstyp mit anderem Funktionsumfang erscheint wie bisher eine Konvertierungsvorschau die dem Anwender über die bevorstehenden Änderungen informiert.



Wird z. B. das Funktionspaket "Motion" entfernt, werden auch alle konfigurierten Achsen gelöscht.

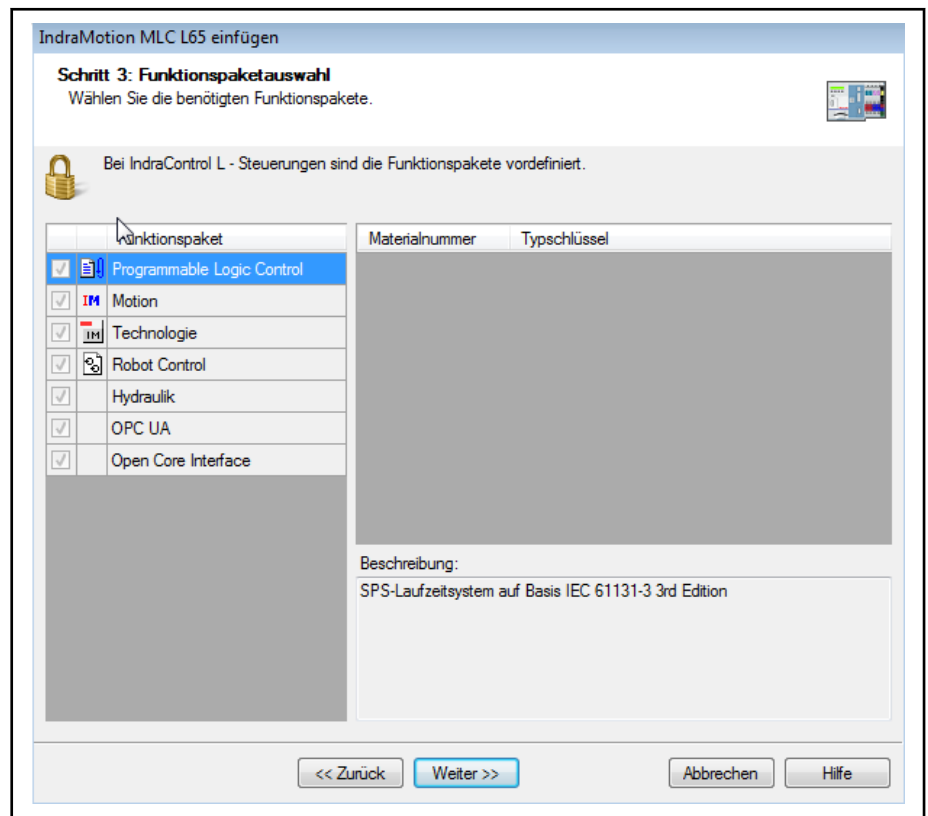


Abb. 2-9: Übersicht der erworbenen Funktionspakete

Programmable Logic Control

Das Funktionspaket "Programmable Logic Control" beinhaltet das SPS-Laufzeitsystem auf Basis der IEC 61131-3 3rd Edition. Das Funktionspaket ist standardmäßig immer aktiviert und kann nicht abgewählt werden.

Motion

Firmware-basierte Ansteuerung von Einzel- und synchronisierten Mehrachs-bewegungen.

Technologie

SPS-Bibliotheken für Technologiefunktionen.

Robot Control

Allgemeine Kinematiken und Transformationen für die Ansteuerung von Robotern.

Hydraulik

Zentrale Regelung von Hydraulikachsen.

OPC UA

OPC UA Server, OPC Unified Architecture, standardisierter Variablen-austausch zwischen Steuerung und Visualisierung.

Open Core Interface

Schnittstelle für Rexroth Open Core Engineering.

Diese als Motion-Logic Programming Interface (MLPI/OCI) bezeichnete Programmierschnittstelle ermöglicht dem Anwender den Zugriff auf Funktionalitäten, die in der Firmware des Systems IndraMotion MLC gekapselt sind. So können schnell und einfach Motion-Control Applikationen in den Programmiersprachen C/C++/C#/VBA/JAVA mit der Entwicklungsumgebung "Visual Studio" entwickelt werden (siehe [Engineering Network](#)).

2.5.5 Funktionsmoduleinstellungen



Der Dialog "Schnittstellen" kann sich, je nach Gerätetyp, unterscheiden. In diesem Projekt wird exemplarisch der Dialog des Gerätes "IndraMotion MLC L65" beschrieben.

Geben Sie die gewünschten Geräte- und Funktionsmoduleinstellungen ein.

Abb. 2-10: Funktionsmoduleinstellungen

Profibus DP (X7P)

Aktuelle Profibuskonfiguration, die beim Einfügen des Gerätes gewählt werden kann. Zur Auswahl stehen "Profibus-DP Master", "Profibus-DP Slave" oder "Nicht verwendet".

Realtime Ethernet (X7E3/X7E4)

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 (X7E5)

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.

Funktionsmodule

An eine "IndraMotion MLC L65/L75 / IndraLogic XLC L65/L75" können bis zu vier Funktionsmodule angeschlossen werden.

Bei Bedarf können Sie die Funktionsmodule aus der Drop-Down-Liste auswählen.

Bestätigen Sie die Eingaben mit der Schaltfläche **Fertigstellen**.

Im Projektexplorer erscheint das neu angelegte Gerät.

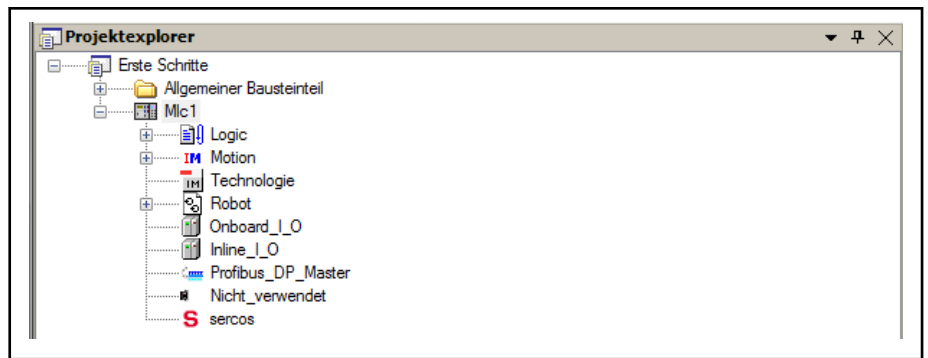


Abb. 2-11: Projektexplorer: Projekt "Erste Schritte" mit einer IndraMotion MLC L65-Steuerung (mit Sercos III Bus)



Der Projektexplorer widerspiegelt die in den Dialogen "Konfiguration" und "Schnittstellen" gewählten Komponenten.

2.6 SPS-Programm erstellen

2.6.1 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 Norm IEC 61131-3 darstellen lassen.
2. Wie Sie mit Hilfe von IndraLogic die verschiedenen Sprachen der Norm editieren können.
3. Wie Sie die erstellten Programmteile miteinander verbinden können.

2.6.2 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 Steuerung vorhanden.

Die POU "MlcProg" wird in unserem Projekt nicht benötigt.



Für das Programm "PlcProg" können Sie wählen:

- Erstellung in "Strukturierter Text (ST)", überspringen Sie den Rest des Kapitels und setzen Sie fort mit [Kap. "Programm SEQUENCE in AS einfügen" auf Seite 28](#)
- Erstellung in "Continuous Function Chart (CFC)", setzen Sie im Text fort

Hierzu gehen Sie folgendermaßen vor:

1. Markieren Sie im Projektexplorer die POU "PlcProg".
2. Öffnen Sie das Kontextmenü mit der rechten Maustaste und wählen Sie den Menüpunkt **Löschen** aus.

Erste Schritte – IndraLogic



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 Applikationsknotens den Befehl **Hinzufügen ► POU** verwenden:

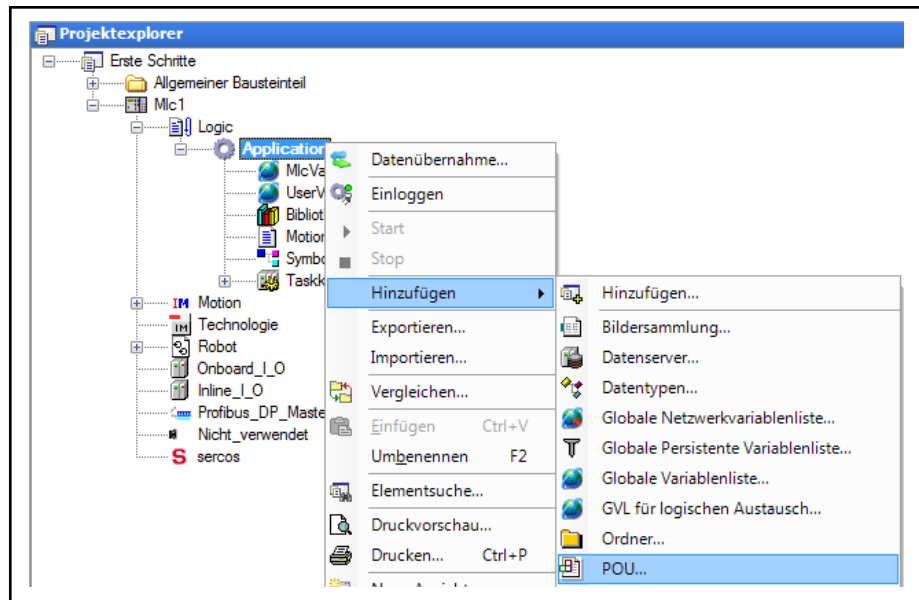


Abb. 2-12: POU aus Bibliothek einfügen

Der Dialog "Objekt hinzufügen" wird geöffnet. Nehmen Sie in diesem Dialog folgende Einstellungen vor:

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

Bestätigen Sie die Eingaben mit **Fertigstellen**.

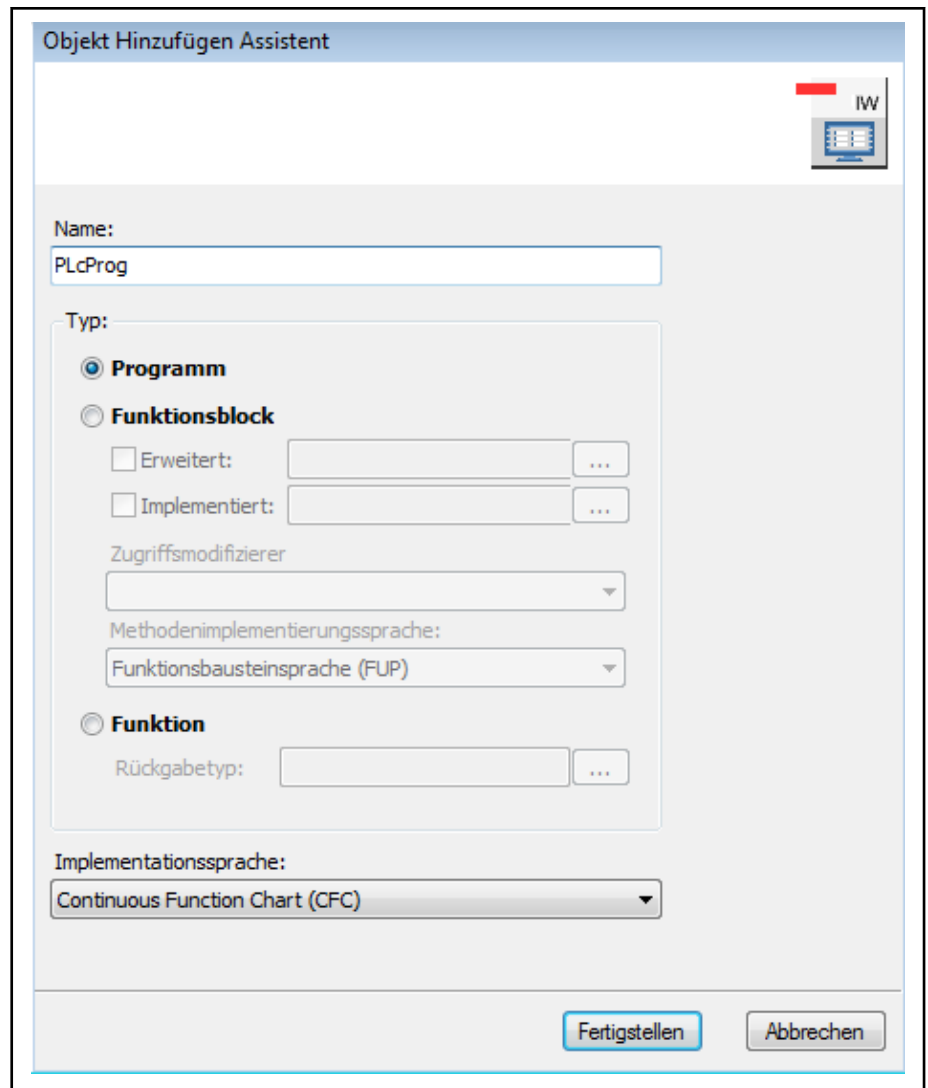


Abb. 2-13: POU "PlcProg" konfigurieren

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

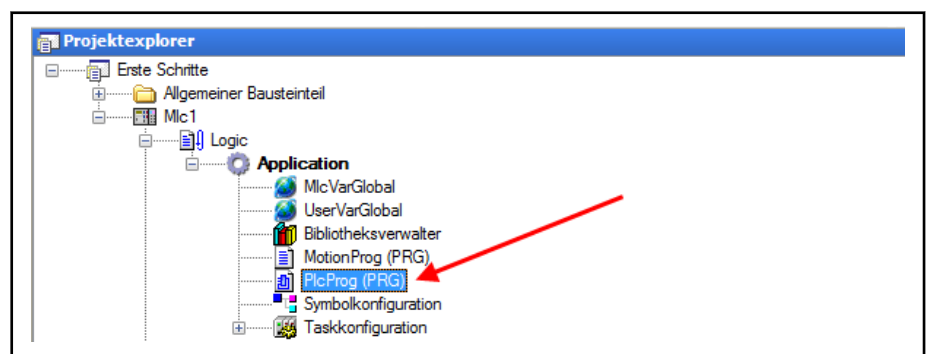


Abb. 2-14: POU "PlcProg" in die Applikation eingefügt

Aufgabe von "PlcProg"

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

Erste Schritte – IndraLogic

Programm "SEQUENCE" in "AS" einfügen

Markieren Sie "Application" und öffnen Sie mit der rechten Maustaste das Kontextmenü des Knotens. Wählen Sie aus dem Kontextmenü **Hinzufügen ► POU**.

Der Dialog "Objekt hinzufügen" wird geöffnet. Nehmen Sie in diesem Dialog 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 die Eingaben mit **Fertigstellen**.

The screenshot shows the 'Objekt Hinzufügen Assistent' dialog box. The 'Name' field is filled with 'SEQUENCE'. Under the 'Typ' section, the 'Programm' radio button is selected. Below it, there are checkboxes for 'Erweitert' and 'Implementiert', both unchecked, followed by empty text boxes and ellipsis buttons. A 'Zugriffsmodifizierer' dropdown is also empty. The 'Methodenimplementierungssprache' dropdown is set to 'Anweisungsliste (AWL)'. The 'Funktion' radio button is unselected. Below it, the 'Rückgabebetyp' field is empty with an ellipsis button. At the bottom, the 'Implementationsprache' dropdown is set to 'Ablaufsprache (AS)'. The 'Fertigstellen' button is highlighted in blue, and the 'Abbrechen' button is greyed out.

Abb. 2-15: Programm "SEQUENCE" hinzufügen

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

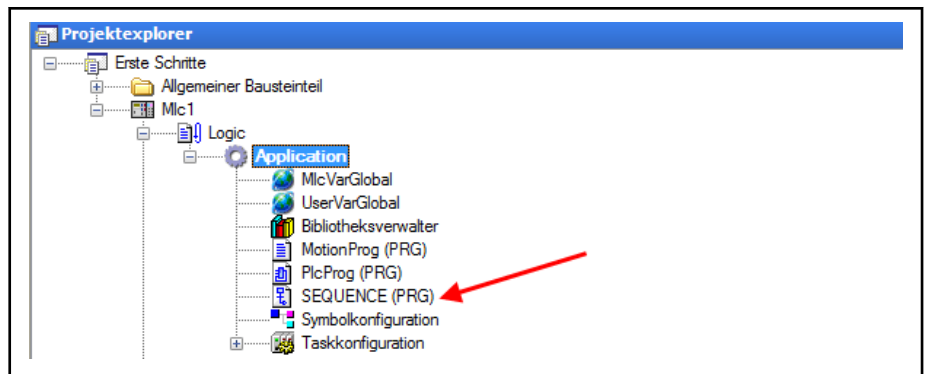


Abb. 2-16: Programm "SEQUENCE" als Teil der Applikation

Aufgabe 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 öffnen Sie mit der rechten Maustaste das Kontextmenü des Knotens. Wählen Sie aus dem Kontextmenü **Hinzufügen ► POU**.

Der Dialog "Objekt hinzufügen" wird geöffnet. Nehmen Sie in diesem Dialog folgende Einstellungen vor:

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

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

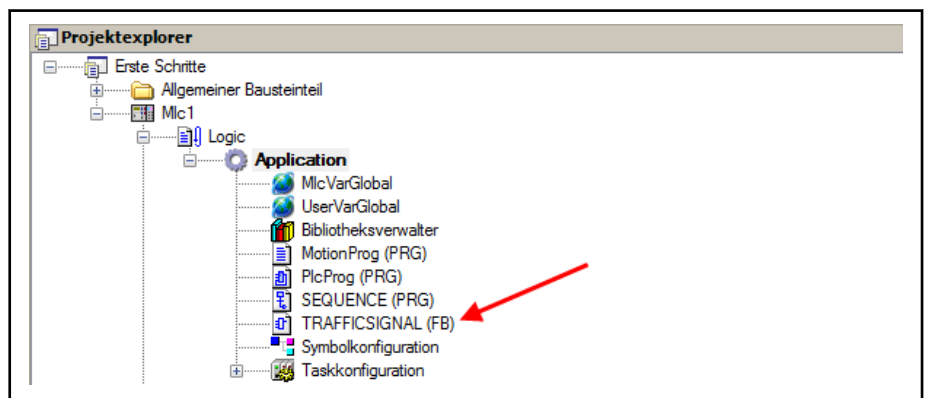


Abb. 2-17: Funktionsbaustein "TRAFFICSIGNAL" als Teil der Applikation

Aufgabe 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.

Erste Schritte – IndraLogic

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

Tab. 2-1: Zuordnung Ampelphasen

Funktionsbaustein "WAIT" in "ST" einfügen

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

Markieren Sie "Application" und öffnen Sie mit der rechten Maustaste das Kontextmenü des Knotens. 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 "Funktionsblock" und als "Implementationssprache:" (Programmiersprache) "Strukturierter Text (ST)" aus.
3. Bestätigen Sie die Eingaben mit **Fertigstellen**.

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

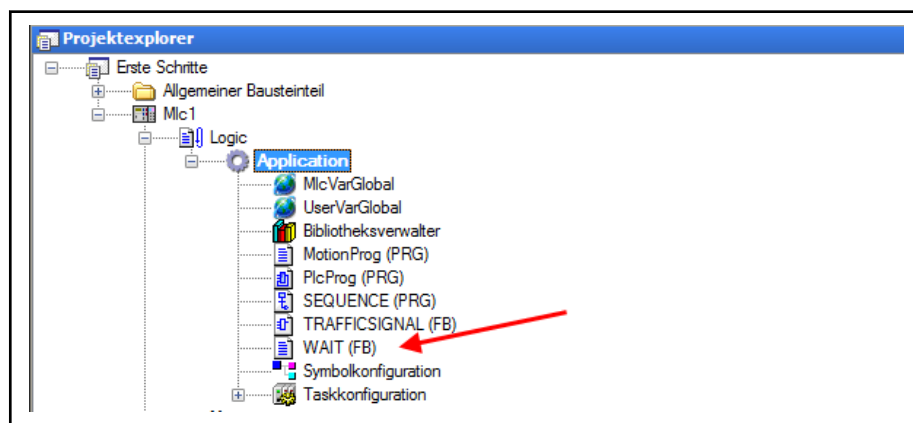


Abb. 2-18: Funktionsbaustein "WAIT" als Teil der Applikation

Aufgabe von "WAIT"

Im Funktionsbaustein "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.

2.6.3 Programme und Funktionsbausteine editieren – erster Teil

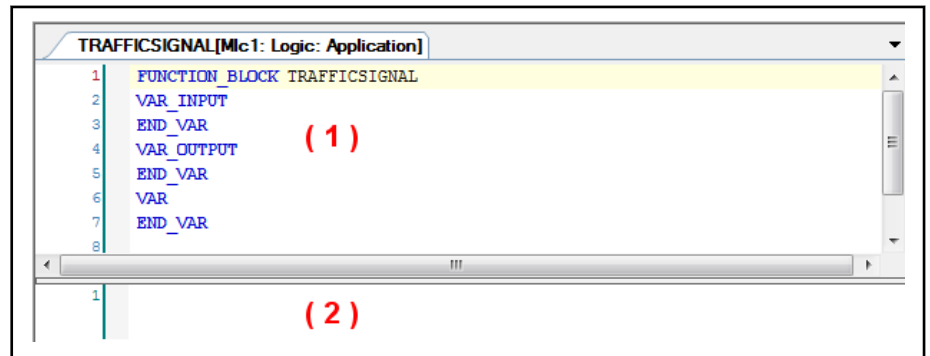
Funktionsbaustein "TRAFFICSIGNAL" in "FUP" editieren

Öffnen der POU

Markieren Sie den Funktionsbaustein "TRAFFICSIGNAL" mit der Maus. Öffnen Sie mit der rechten Maustaste das Kontextmenü. Wählen Sie im Kontextmenü des Funktionsbausteins "TRAFFICSIGNAL" **Öffnen** aus. Der Editor des Funktionsbausteins wird geöffnet (siehe nachfolgende Abbildung).



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



(1) Deklarationsteil

(2) Implementationsteil oder Rumpf

Abb. 2-19: Editor des Funktionsbausteins "TRAFFICSIGNAL" geöffnet

Der POU Editor

Der Editor ist in zwei Bereiche gegliedert:

1. Deklarationsteil.
2. Implementationsteil.

Im Deklarationsteil werden die Ein- und Ausgangsvariablen des Funktionsbausteins und lokale 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 (STRUCT) und von Datenfeldern (ARRAY) 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 Implementationsteil. Selektieren Sie das erste Netzwerk im Implementationsteil, indem Sie rechts in den Editor neben die "1" klicken. Öffnen Sie mit der rechten Maustaste das Kontextmenü und wählen Sie **Bausteinaufruf einfügen**. Alternativ wählen Sie im Menü **FUP/KOP/AWL ► Bausteinaufruf einfügen**.

Erste Schritte – IndraLogic

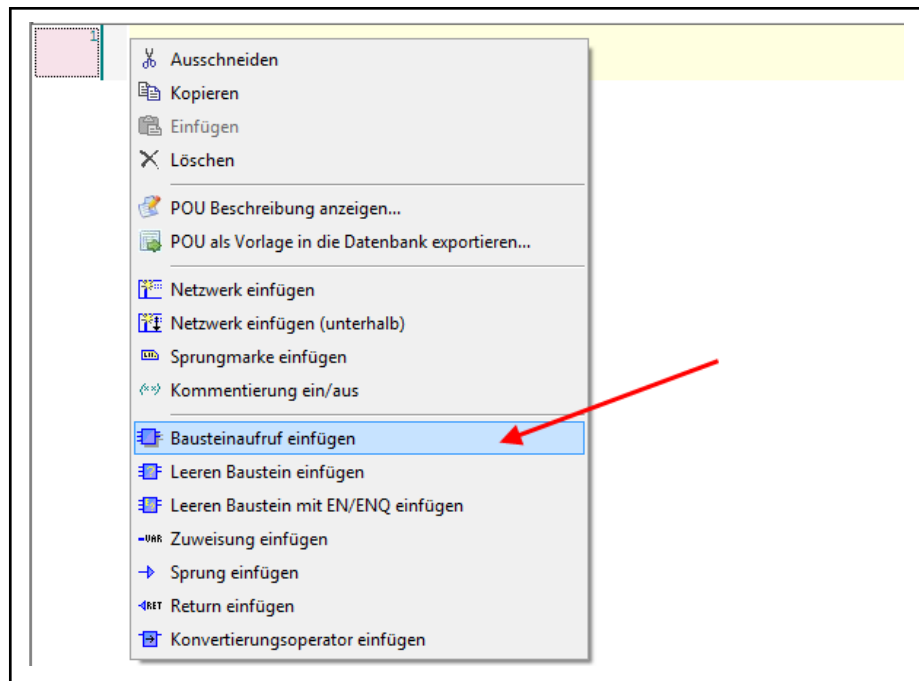


Abb. 2-20: Baustein aufruf einfügen

Das Fenster "Eingabehilfe" wird geöffnet:

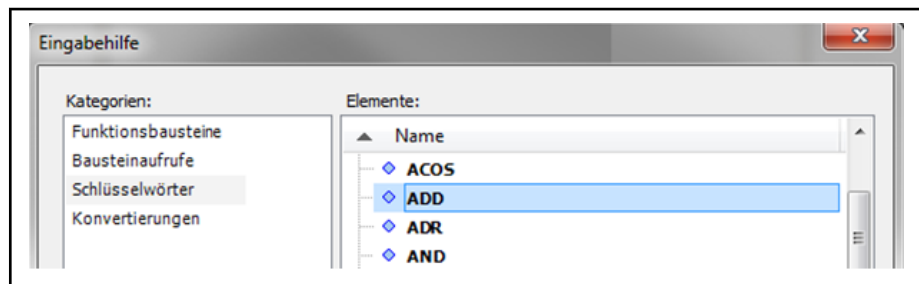


Abb. 2-21: Eingabehilfe – Baustein aufruf "ADD" einfügen

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 ihre Auswahl mit **OK**.

Der Anweisungsteil sieht folgendermaßen aus:

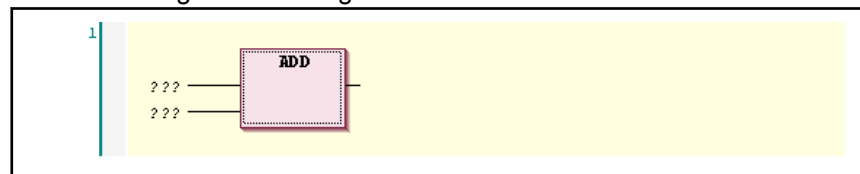


Abb. 2-22: Baustein "ADD" in Netzwerk eingefügt

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

1. Klicken Sie hierzu mit der Maus auf den Text "ADD", so dass dieser Text markiert erscheint, und ändern Sie den Text in "EQ" ab.

Erste Schritte – IndraLogic

2. Markieren Sie nun, links am Baustein mit dem Namen "EQ", den oberen Eingang "???" und tragen Sie 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.

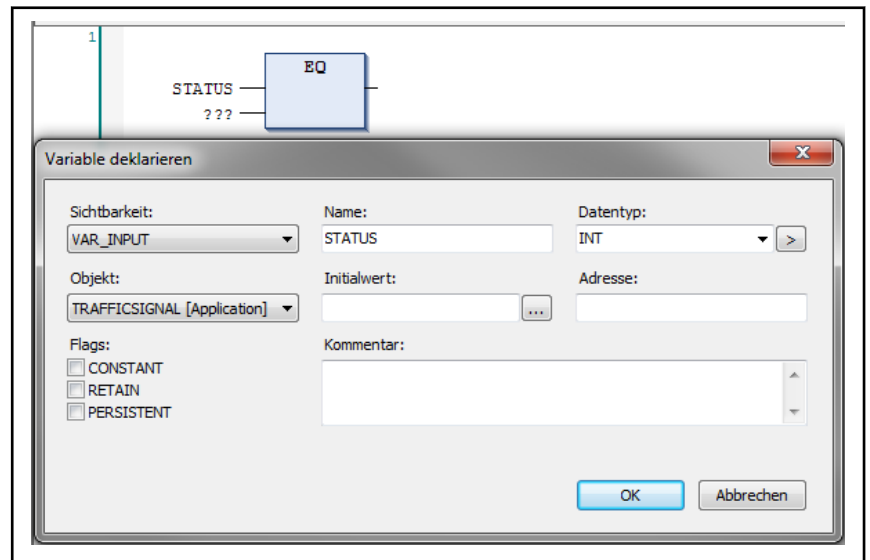


Abb. 2-23: Deklaration der Eingangsvariablen "STATUS"

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 sind:

- Grün
- Gelb
- Gelb-Rot
- Rot

Die vier Zustände sollen als Ausgangsvariable decodiert werden. Für die vier Zustände werden die folgenden drei Variablen benötigt:

- GREEN
- YELLOW
- RED

1. Klicken Sie nun im Netzwerk "1" in einen Bereich hinter den Baustein "EQ", um dem Baustein eine Zuweisung anzufügen. Öffnen Sie hierzu mit der rechten Maustaste das Kontextmenü und wählen Sie **Zuweisung einfügen** aus. Klicken Sie auf den Ausgang des Bausteins "???" und geben ihm den Variablennamen "GREEN".

Erste Schritte – IndraLogic

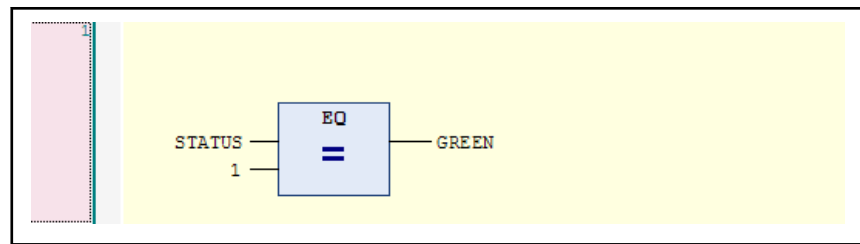


Abb. 2-24: Baustein "EQ"

Das Fenster zur Deklaration von Variablen öffnet sich.

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

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

Aufgabe 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.

- Klicken Sie dazu in das erste Netzwerk und wählen über das Kontextmenü **Netzwerk einfügen (unterhalb)** aus.

Zweites Netzwerk editieren Erstellen Sie einen Baustein "EQ" im zweiten Netzwerk.

- Klicken Sie mit der rechten Maustaste in das zweite Netzwerk.
- Wählen Sie im Kontextmenü **Baustein aufruf einfügen**.
- Wählen Sie in der "Eingabehilfe" zu "Baustein aufruf einfügen" unter "Kategorien:" die "Schlüsselworte".
Wählen Sie unter "Elemente:" das Element "EQ" aus.
Bestätigen Sie ihre Eingaben mit der Schaltfläche **OK**.
- Markieren Sie den Ausgang des Bausteins "EQ" und klicken Sie auf die rechte Maustaste.
- Wählen Sie im Kontextmenü **Baustein aufruf einfügen** aus.
- Wählen Sie in der "Eingabehilfe" zu "Baustein aufruf einfügen" unter "Kategorien:" die "Schlüsselworte" aus.

Wählen Sie unter "Elemente:" das Element "OR" aus.

Bestätigen Sie ihre Eingaben mit der Schaltfläche **OK**.

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

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

Erste Schritte – IndraLogic

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 **Baustein aufruf einfügen** aus.
3. Wählen Sie in der "Eingabehilfe" zu "Baustein aufruf einfügen" unter "Kategorien:" die "Schlüsselworte" aus.

Wählen Sie unter "Elemente:" das Element "EQ" aus.

Bestätigen Sie ihre Eingaben 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 die Eingabe 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 die Eingabe mit <Enter>.
4. Klicken Sie auf den ersten Eingang des unteren Bausteins "EQ" im zweiten Netzwerk.
Geben Sie "STATUS" ein und bestätigen Sie die Eingabe mit <Enter>.
5. Klicken Sie auf den zweiten Eingang des selben Bausteins "EQ".
Geben Sie "4" ein und bestätigen Sie die Eingabe 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 Sie mit der rechten Maustaste das Kontextmenü.
2. Wählen Sie im Kontextmenü **Kopieren** aus.
3. Öffnen Sie erneut mit der rechten Maustaste das Kontextmenü.
4. Wählen Sie **Einfügen** aus.

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

Erste Schritte – IndraLogic

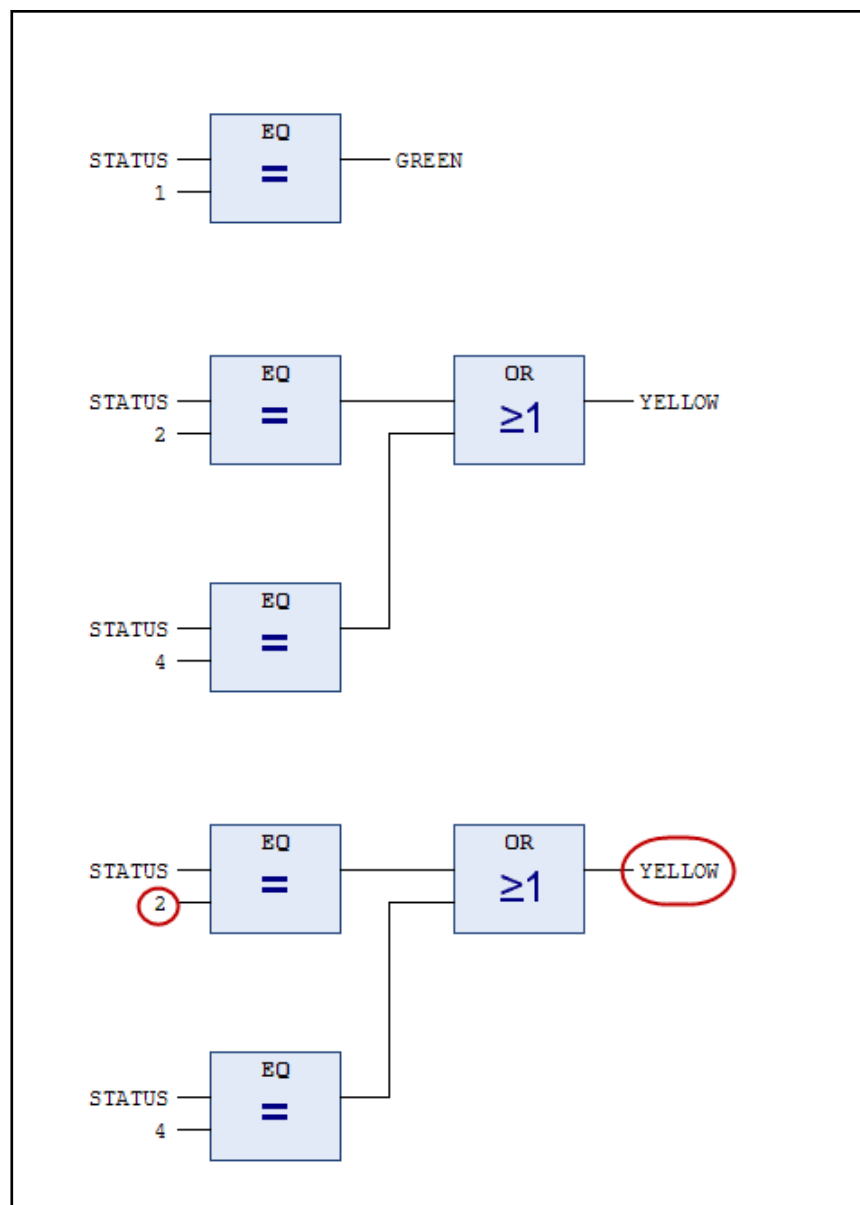


Abb. 2-25: Drittes Netzwerk durch Kopie eingefügt

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".

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

Der erste Baustein POU "TRAFFICSIGNAL" ist nun fertiggestellt:

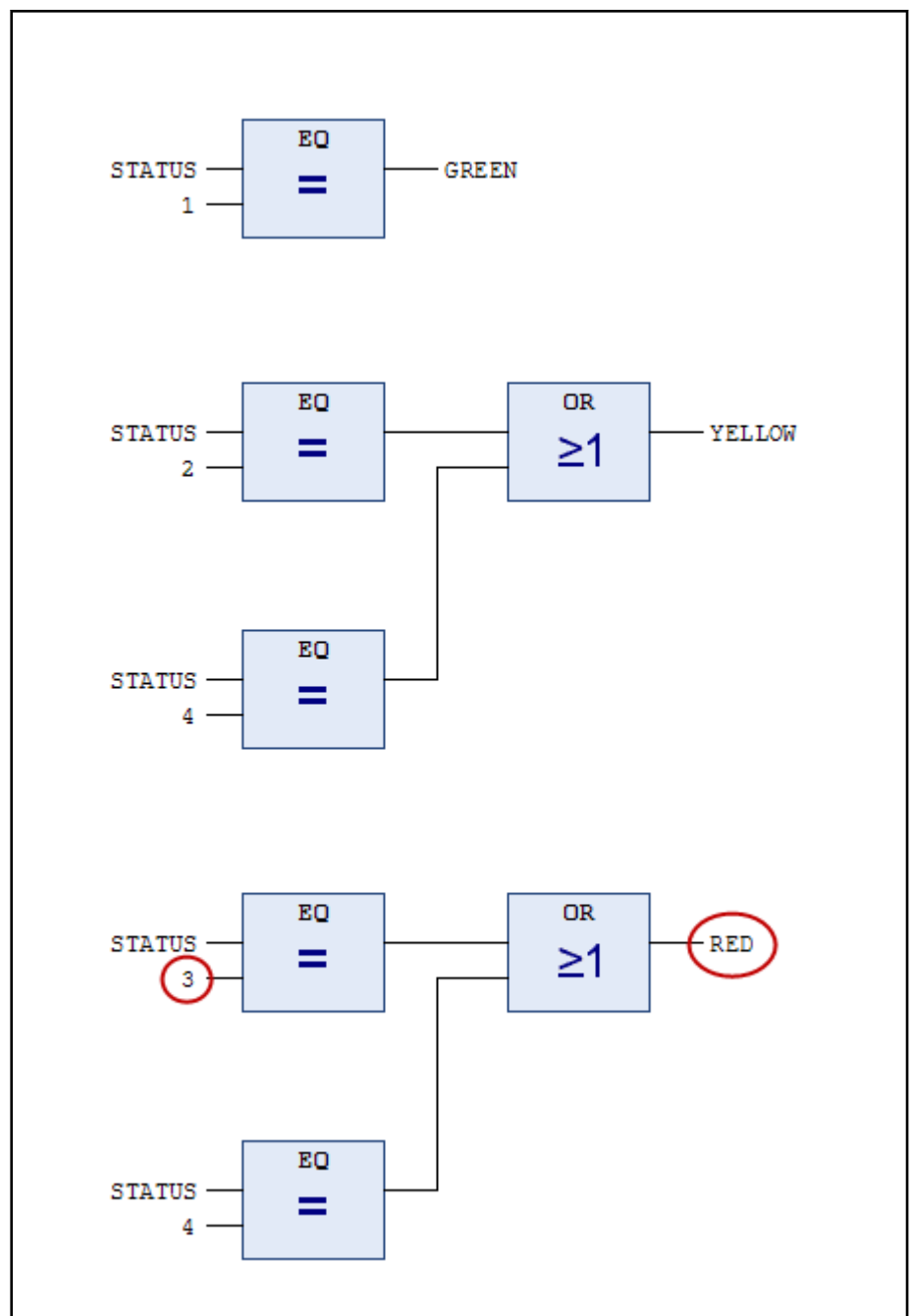


Abb. 2-26: POU "TRAFFICSIGNAL" fertiggestellt

Beiläufig zur Implementation ist der Deklarationsteil entstanden:

Erste Schritte – IndraLogic

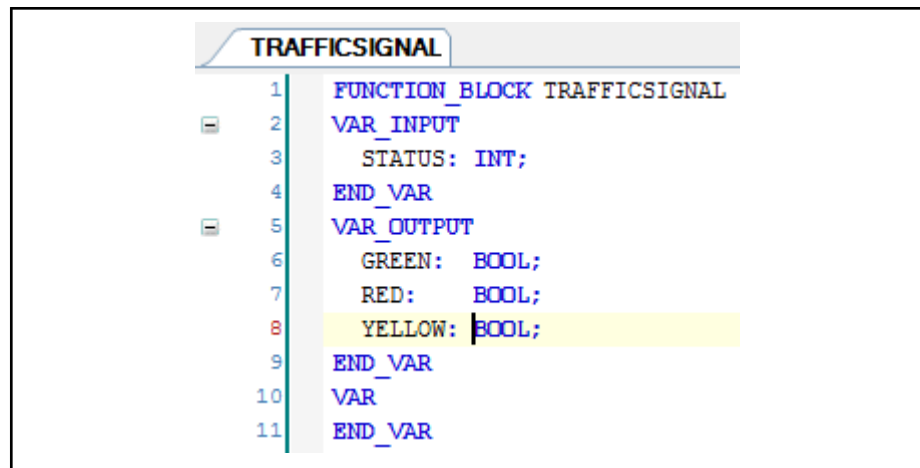


Abb. 2-27: Deklarationsteil des FBs "TRAFFICSIGNAL"

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;`.

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

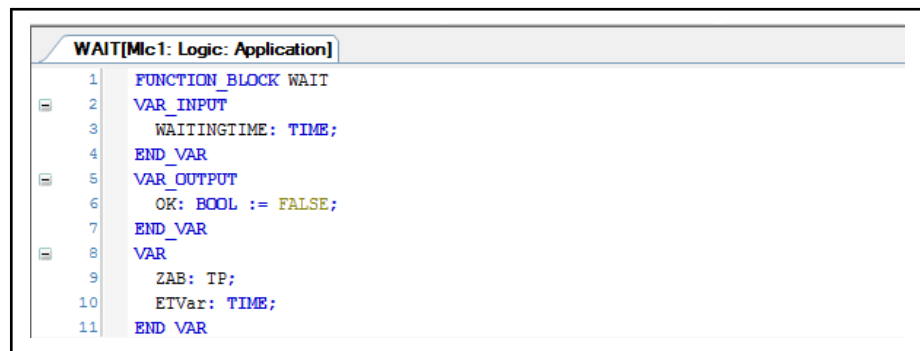


Abb. 2-28: Variablendeklaration POU "WAIT"

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 Projektnavigator.
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.

Erste Schritte – IndraLogic

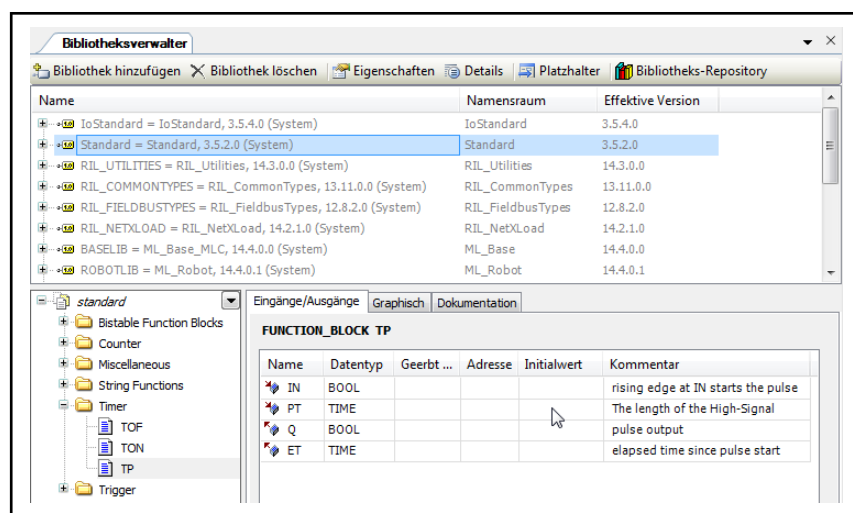


Abb. 2-29: Bibliotheksverwalter Standardbaustein "TP"

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

Aufgabe von "TP"

Solange "IN" den 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 die Zeit am Ausgang "ET" nicht weiter aufaddiert.

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

Sobald der Wert der Variablen "PT" erreicht ist, liefert "Q" den 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:

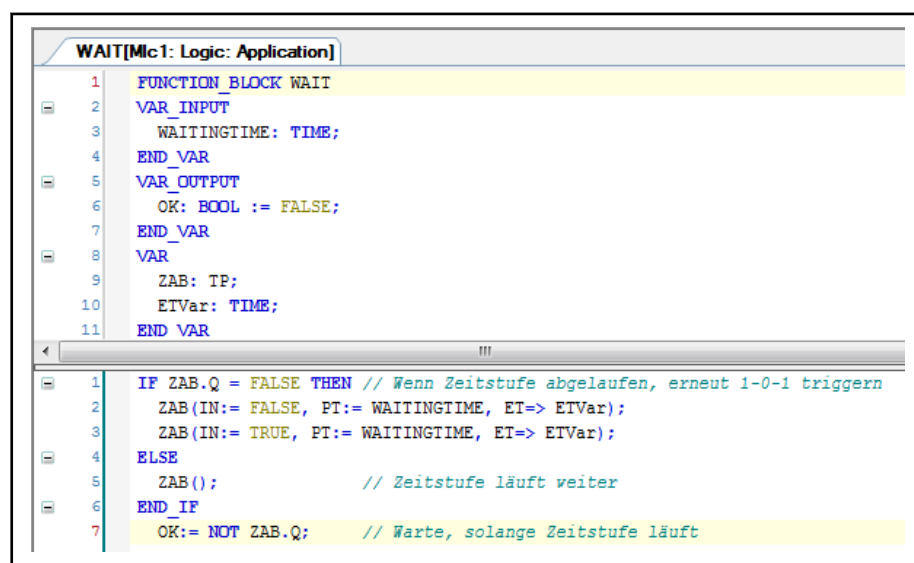


Abb. 2-30: POU "WAIT" in ST fertiggestellt

Erste Schritte – IndraLogic

Aufgabe von "WAIT"

Für den Fall, dass die Zeitstufe abgelaufen ist, wird die Zeitstufe 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:

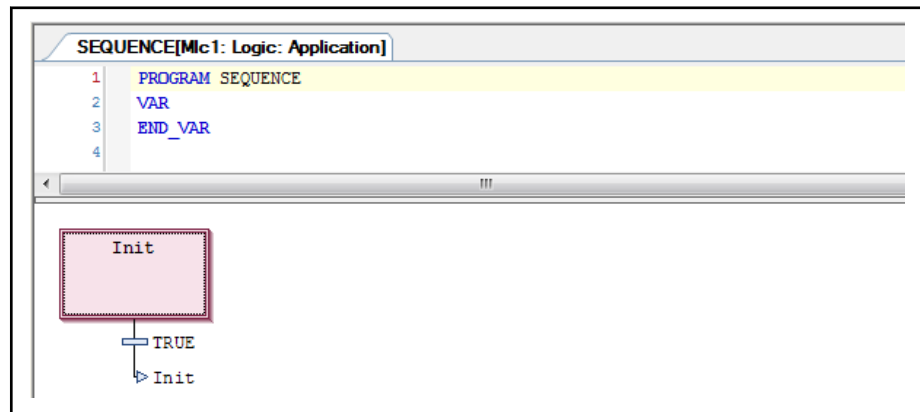


Abb. 2-31: Editor: Programm "SEQUENCE" geöffnet

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.

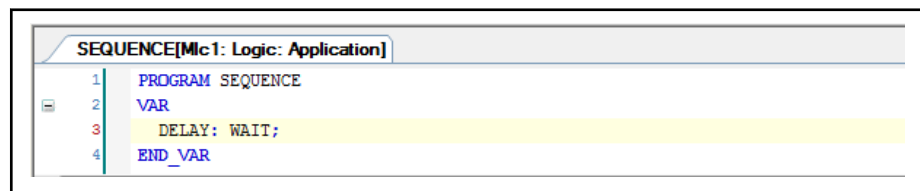


Abb. 2-32: Variablendeklaration POU "SEQUENCE" – erster Teil

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 Abb. 2-31 "Editor: Programm SEQUENCE geöffnet" auf Seite 40).

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** aus.

3. Markieren Sie jeweils die letzte Transition und fügen Sie drei weitere Schritte und Transitionen ein.

Die Ablaufkette sieht nun wie folgt aus:

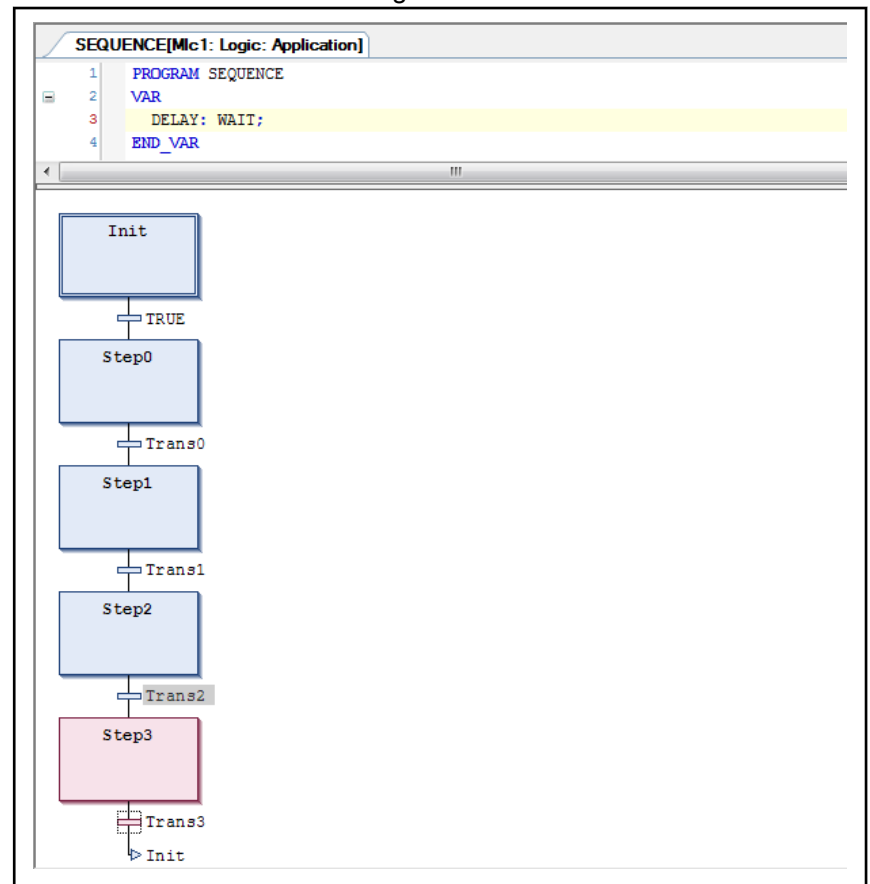


Abb. 2-33: In Programm "SEQUENCE" Schritte eingefügt



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: Die Änderung ist dann sinnvoll, wenn nur eine Weiterschaltbedingung benötigt wird, z. B. bei der Abfrage einer Variablen.
2. Programmieren der Transition: Das Programmieren ist dann sinnvoll, wenn das Ergebnis verknüpfter Variablen als Weiterschaltbedingung dienen soll.

Um eine Transition zu programmieren, muss eine Transition hinzugefügt werden. Mit einem Doppelklick auf die "Transition" können Sie für die Transition einen Namen vergeben und die Implementationssprache (Programmiersprache) festlegen, in der die Transition programmiert 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:

1. Klicken Sie auf den Transitionsnamen "TRUE" der ersten Transition.
2. Geben Sie in das Eingabefeld "START" ein.

Erste Schritte – IndraLogic

Da die Variable "START" im Deklarationsteil noch nicht deklariert wurde, öffnet sich ein Fenster zum Deklarieren der Variablen.

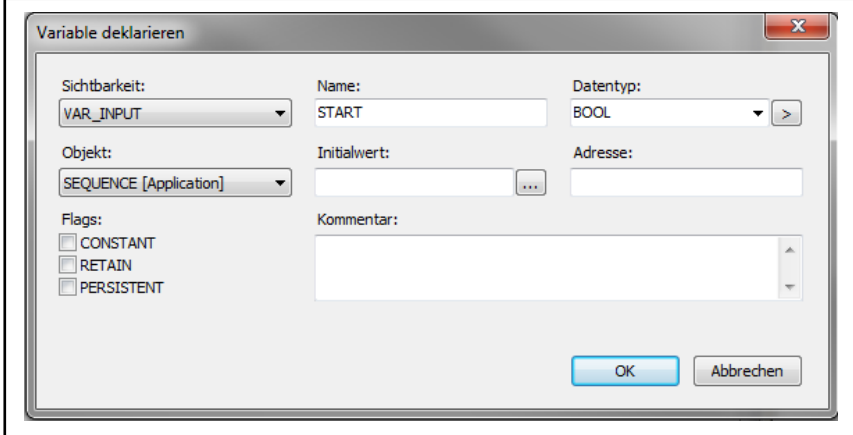


Abb. 2-34: Eingabehilfe zur Variablendeklaration

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

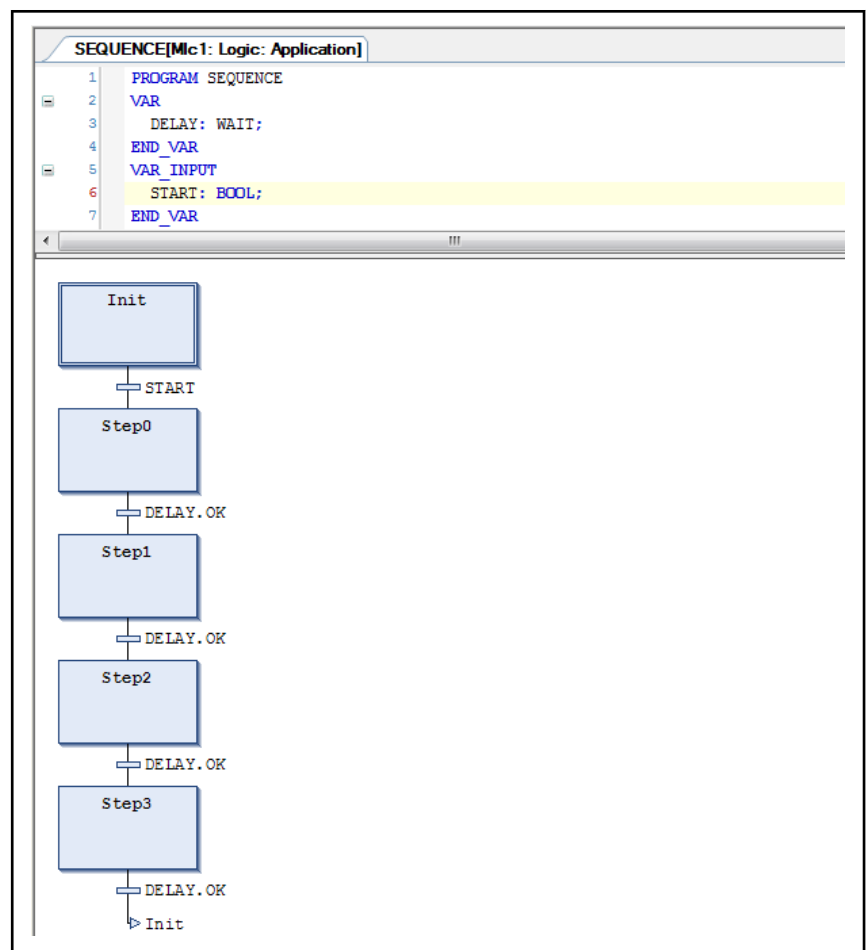


Abb. 2-35: Schritte der Ablaufkette angelegt

Schritte benennen Nur der Schritt "Init" behält seinen Namen bei. Die restlichen Schritte werden umbenannt. Die Schritte 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 ihre Eingaben mit <Enter>.

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.

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 das neue Rücksprungziel "Switch1" ein.

Der Ablauf wiederholt sich auf diese Weise ständig.

Die Ablaufkette sieht nun wie folgt aus:

Erste Schritte – IndraLogic

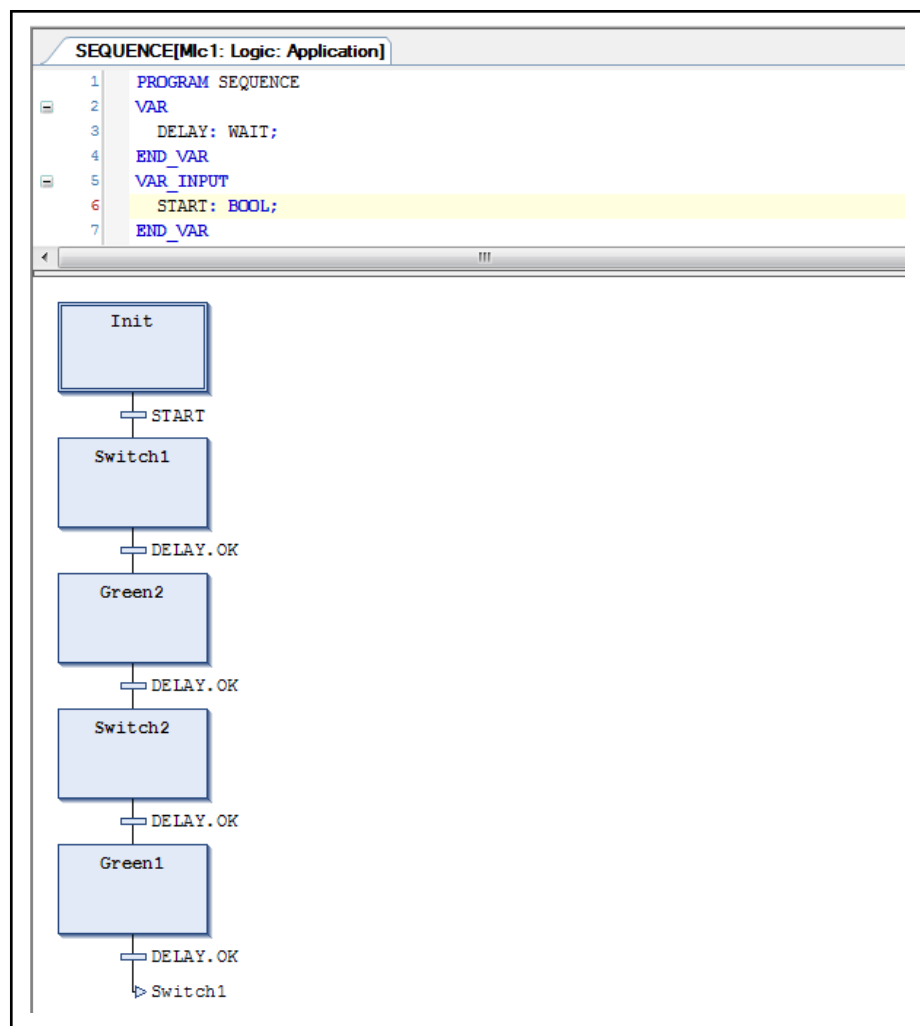


Abb. 2-36: Schritte der Ablaufkette benannt

Schritte und Aktionen

Ein in Ablaufsprache programmierter 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:

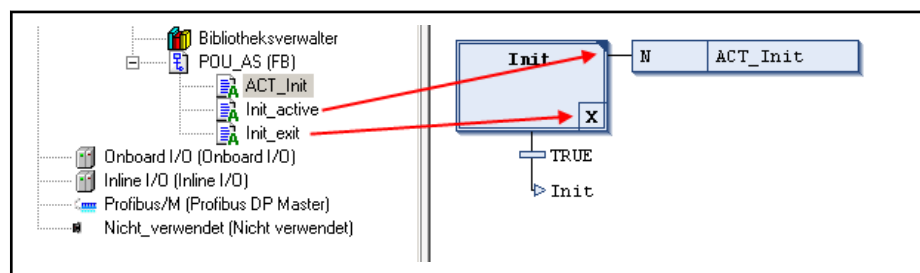


Abb. 2-37: Arten der Aktionen eines Schrittes

- 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".

Das Zeitverhalten wird durch das Bestimmungszeichen (hier: N) bestimmt. Die assoziierten Aktionen erscheinen rechts vom Schritt. Wird eine solche Aktion inaktiv, wird die Aktion 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 "SEQUENCE (FB)" wird eine Verzweigung mit der Aktion "Init_active" erzeugt.

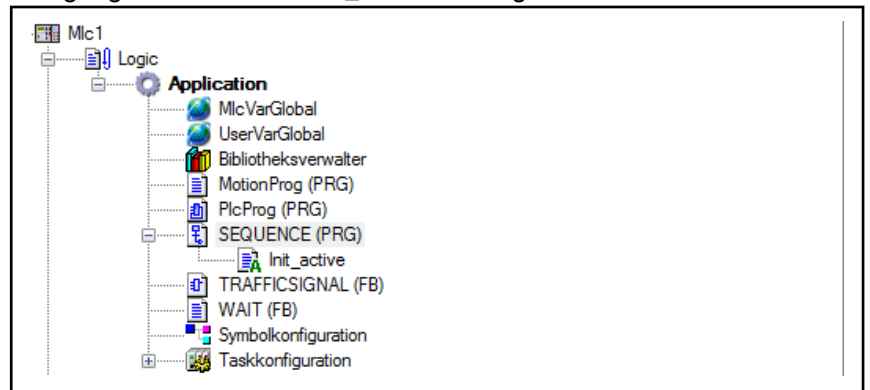


Abb. 2-38: POU "SEQUENCE" im Projektextplorer mit Aktion "Init_active"
Der Editor der Aktion "Init_active" wird geöffnet.

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_OUT-PUT" und den Datentyp "INT" aus.

Aktion "Switch1" mit "ST" programmieren

In der Aktion zum Schritt "Switch1" wechselt der STATUS von Ampel1 (TRAFFICSIGNAL1) auf "2" (gelb) und der STATUS von Ampel 2 (TRAFFIC-

Erste Schritte – IndraLogic

SIGNAL2) 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.

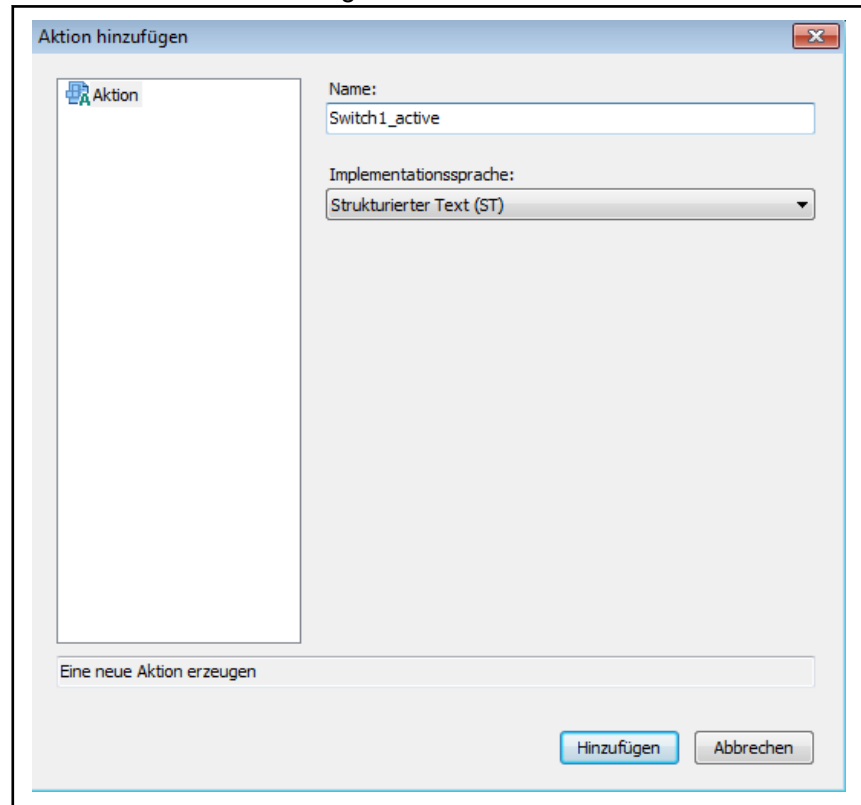


Abb. 2-39: Aktion hinzufügen

2. Wählen Sie hier als Implementationssprache (Programmiersprache) "Strukturierter Text (ST)" aus.
3. Bestätigen Sie Ihre Eingaben mit der Schaltfläche **Hinzufügen**.

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:

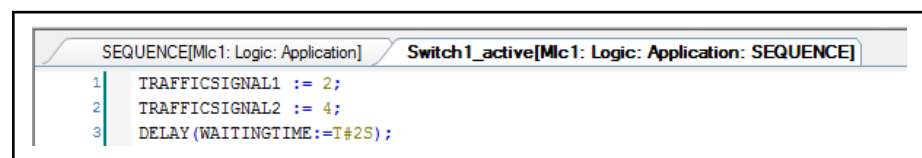


Abb. 2-40: Editor der Aktion "Switch1_active"

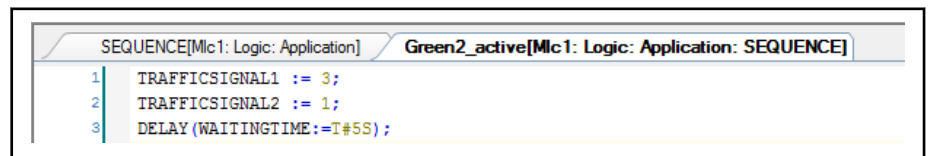
Aktion "Green2" mit "ST" programmieren

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:

Erste Schritte – IndraLogic

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:



```
1 TRAFFICSIGNAL1 := 3;  
2 TRAFFICSIGNAL2 := 1;  
3 DELAY(WAITINGTIME:=T#5S);
```

Abb. 2-41: 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:



```
1 TRAFFICSIGNAL1 := 4;  
2 TRAFFICSIGNAL2 := 2;  
3 DELAY(WAITINGTIME:=T#2S);
```

Abb. 2-42: 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**.

Erste Schritte – IndraLogic

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. 2-43: Editor der Aktion "Green1_active"

Der fertige Funktionsbaustein
"SEQUENCE"

Der Funktionsbaustein "SEQUENCE" und die Struktur im Projektextplorer ist nun fertiggestellt:

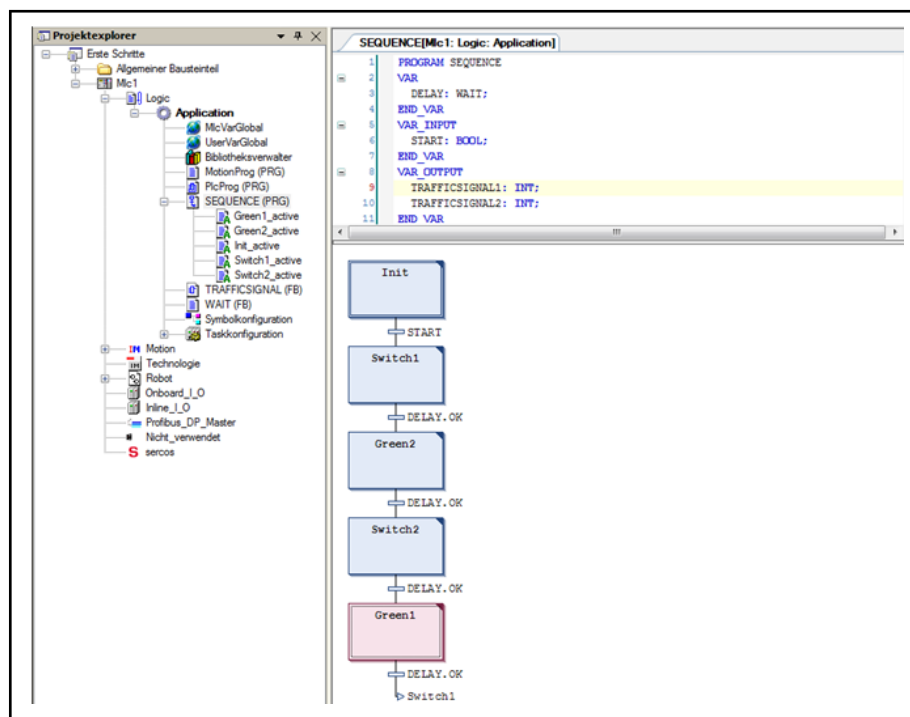


Abb. 2-44: POU "SEQUENCE" fertiggestellt

2.6.4 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 POU's 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 Projektextplorer 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 "Mlc1" 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. 2-45: Taskkonfiguration

2.6.5 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 wird 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 aller POUs der Applikation zur Verfügung.

Variablen mit Adressen deklarieren

Legen Sie in der "Global Variable List" die folgenden Variablen vom Typ "BOOL" 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_VÄR

```

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 IndraMotion MLC L45/L65/L75 besitzen diese Ausgänge als Onboard-Ausgänge (Standard-Mapping).

Die Steuerungen IndraMotion MLC L25, IndraMotion MLC XM2 und IndraMotion MLC VPx besitzen keine Onboard-Ausgänge. Diese Steuerung muss deshalb durch Inline-Ausgangs-Klemmen (nur IndraMotion MLC L25) oder einen Feldbus mit I/O-Modulen erweitert sein. Wenn nicht, verzichten Sie auf die absolute Adresse oder Sie erhalten bei der Code-Erzeugung eine Warnung.

2.6.6 Testen von "SEQUENCE" im Simulationsmodus

Der Programmbaustein "SEQUENCE" ist soweit fertiggestellt, dass der Programmbaustein 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 ► Übersetzen**. In der "Aufgabenliste" unterhalb des Arbeitsfensters steht nach dem Übersetzen: "Kompilierung abgeschlossen -- 0 Fehler, 0 Warnungen".

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 "Mlc1" und öffnen Sie mit der rechten Maustaste das Kontextmenü. Klicken Sie auf **Simulation** um den Simulationsmodus zu aktivieren. Ist die Simulation aktiv, wird ein Haken bei "Simulation" gesetzt.

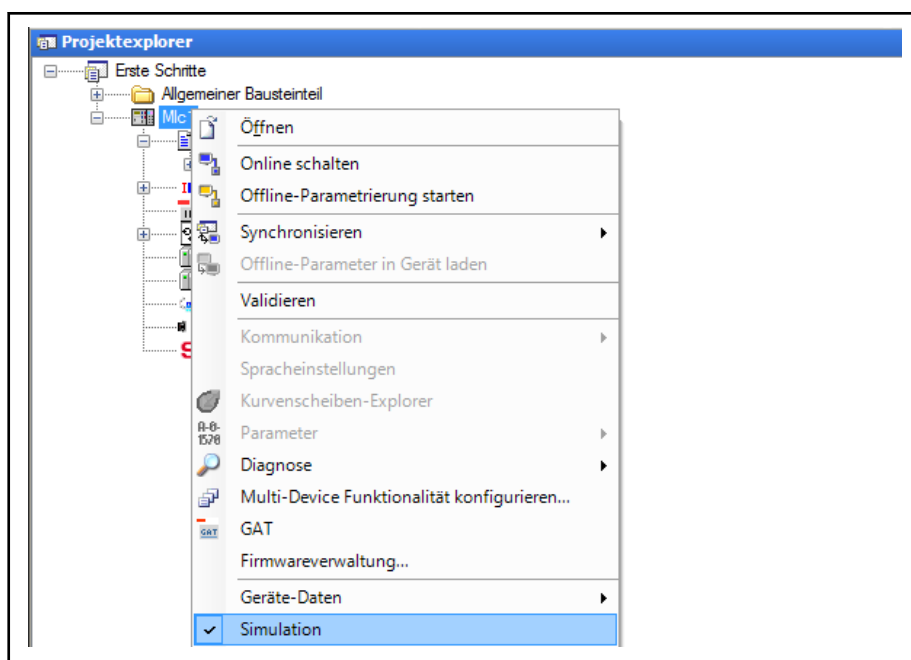


Abb. 2-46: Auswahl "Simulation"

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 in der IndraWorks-Menüleiste **Debuggen** ► **Einloggen**.

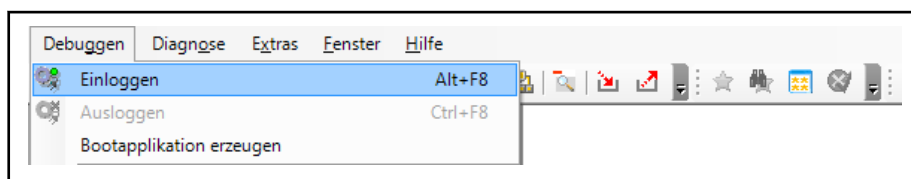


Abb. 2-47: Einloggen auf der Steuerung

Der Projektexplorer hat nun folgendes Erscheinungsbild:

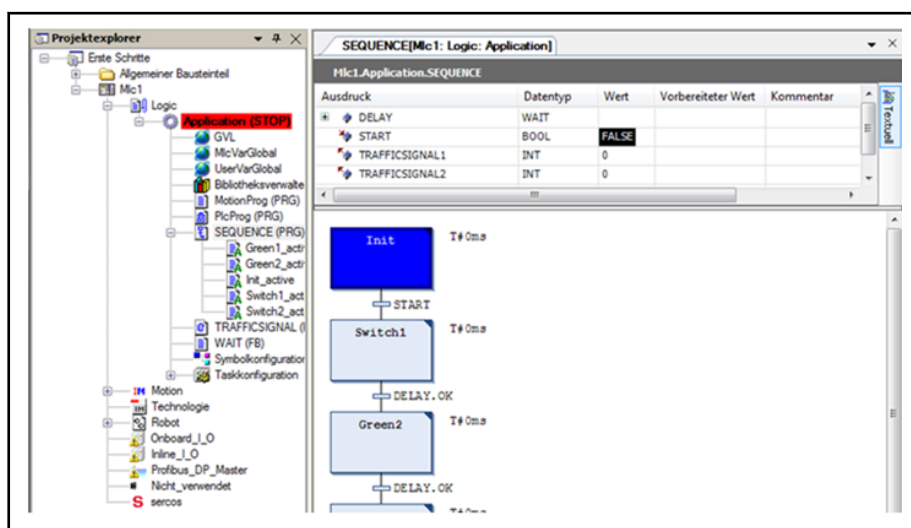


Abb. 2-48: In Steuerung (Simulation) eingeloggt – Applikation in "STOPP"

Applikation starten

Um die Applikation zu starten, klicken Sie mit der rechten Maustaste auf die Applikation und wählen im Kontextmenü **Start** oder in der IndraWorks-Menüleiste **Debuggen** ► **Start**.

Die Applikation läuft nun auf der Steuerung ab.

Erste Schritte – IndraLogic

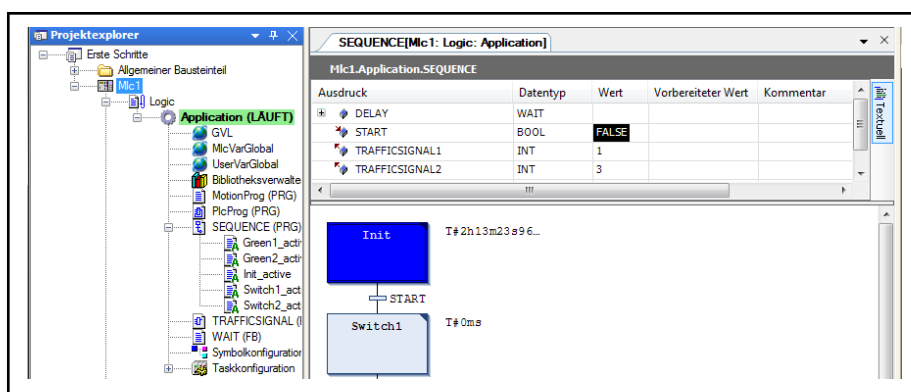


Abb. 2-49: In Steuerung eingeloggt – Applikation läuft

"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 "TRUE" erscheint.

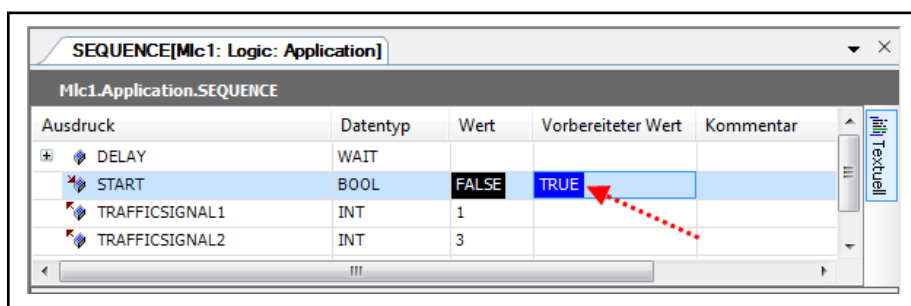


Abb. 2-50: Deklarationsteil POU "SEQUENCE" online

Wählen Sie in der IndraWorks-Menüleiste **Debuggen** ► **Werte schreiben** (<Strg>+<F7>).

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

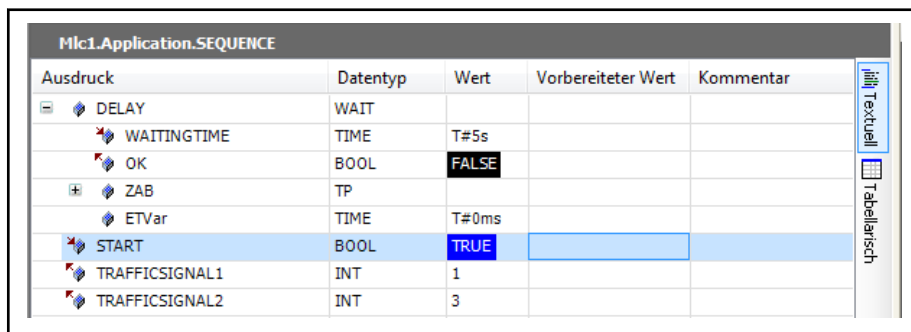


Abb. 2-51: Deklarationsteil POU "SEQUENCE" online

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 in der IndraWorks-Menüleiste **Debuggen** ► **Stop**.

Die Applikation wird auf der Steuerung gestoppt.

Loggen Sie sich jetzt aus, indem Sie in der IndraWorks-Menüleiste **Debuggen** ► **Ausloggen** auswählen.

IndraWorks befindet sich im Offline-Modus.



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

Sie können mit den von Ihnen vorgegebenen Werten starten, wenn Sie vor dem Einloggen die alten Daten löschen (IndraWorks-Menüleiste **Erstellen** ► **Bereinigen**).

2.6.7 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 [Kap. "Programm PlcProg in CFC editieren" auf Seite 56](#)
- Programmierung in "Strukturiertem Text (ST)", setzen Sie im Text fort

Aufgabe 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 im Programm "PlcProg" verwendet werden. Die Applikation wurde bisher nur im Simulationsmodus getestet. Nun kann die Applikation auch in einer realen Steuerung getestet werden.

Instanzen deklarieren

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

Zum Anlegen der Variable orientieren Sie sich an ["Variablen deklarieren" auf Seite 31](#).

Legen Sie die Variable `LIGHT1: TRAFFICSIGNAL;`

und `LIGHT2: TRAFFICSIGNAL; an:`

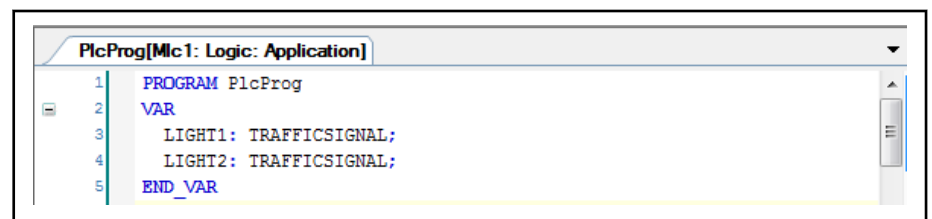


Abb. 2-52: Deklarationsteil POU "PlcProg"

Rumpf editieren

Editiert wird das 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", "YEL-

Erste Schritte – IndraLogic

LOW" 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:

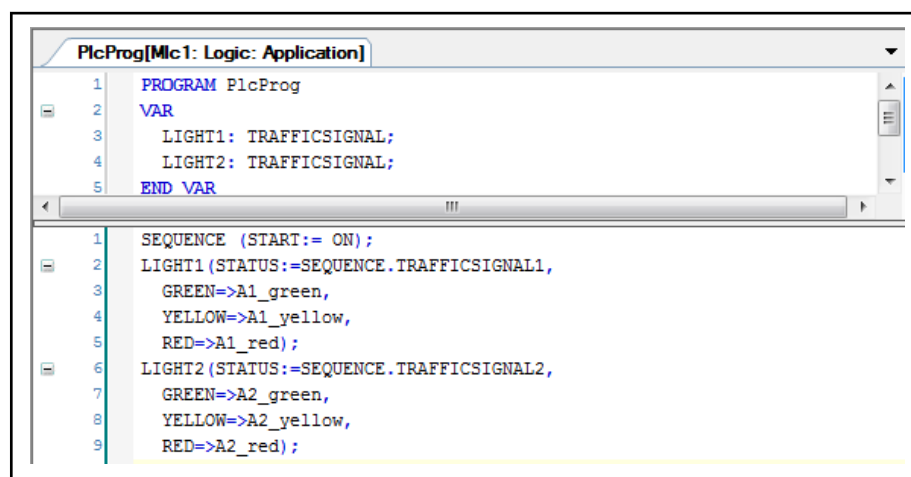


Abb. 2-53: Programm "PlcProg" fertiggestellt

Testen der Applikation im Simulationsmodus, PlcProg in ST

Applikation 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".



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 "Applikation" **Aktive Applikation setzen**).

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 "Mlc1" und öffnen Sie mit der rechten 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 in der IndraWorks-Menüleiste **Debuggen ► Einloggen**.

Die Applikation steht danach im Zustand "STOP".

Applikation starten

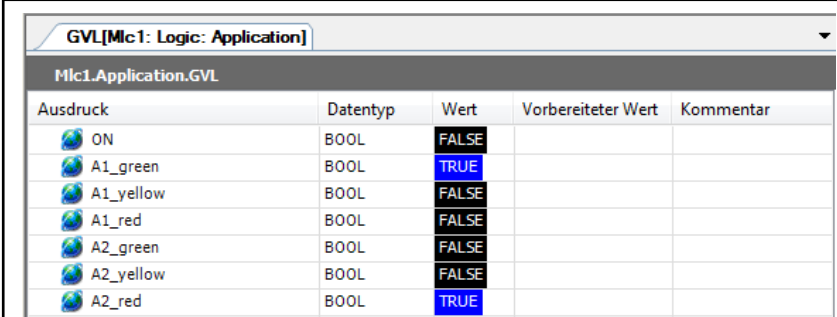
Um die Applikation zu starten, klicken Sie mit der rechten Maustaste auf die Applikation und wählen im Kontextmenü **Start**. Die Applikation läuft nun als Simulation ab.

Erste Schritte – IndraLogic

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 eingeschaltet 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 Projektexplorer die "Globale Variablen Liste".

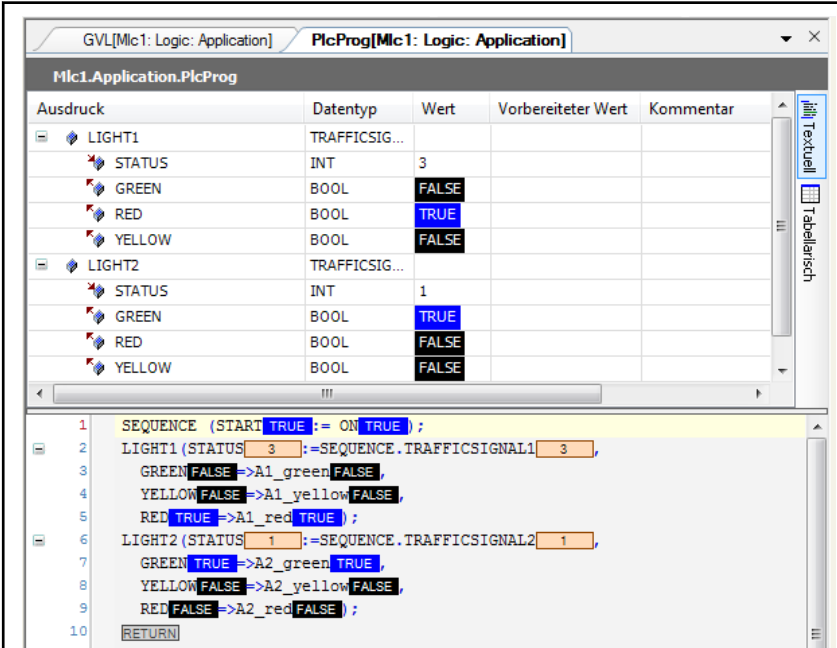


Ausdruck	Datentyp	Wert	Vorbereiteter Wert	Kommentar
ON	BOOL	FALSE		
A1_green	BOOL	TRUE		
A1_yellow	BOOL	FALSE		
A1_red	BOOL	FALSE		
A2_green	BOOL	FALSE		
A2_yellow	BOOL	FALSE		
A2_red	BOOL	TRUE		

Abb. 2-54: "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, wählen Sie in der IndraWorks-Menüleiste **Debuggen** ► **Werte schreiben**.

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



Ausdruck	Datentyp	Wert	Vorbereiteter Wert	Kommentar
LIGHT1	TRAFFICSIG...			
STATUS	INT	3		
GREEN	BOOL	FALSE		
RED	BOOL	TRUE		
YELLOW	BOOL	FALSE		
LIGHT2	TRAFFICSIG...			
STATUS	INT	1		
GREEN	BOOL	TRUE		
RED	BOOL	FALSE		
YELLOW	BOOL	FALSE		

```

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

Abb. 2-55: Programm "PLCProg" online

Applikation stoppen

Um das SPS-Programm zu stoppen, wählen Sie in der IndraWorks-Menüleiste **Debuggen** ► **Stop**.

Aus Steuerung ausloggen

Um sich aus der Steuerung auszuloggen, wählen Sie in der IndraWorks-Menüleiste **Debuggen** ► **Ausloggen**.

Erste Schritte – IndraLogic

Programm "PlcProg" in "CFC" editieren



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

- Programmierung in "Strukturiertem Text (ST)", gehen Sie zu [Kap. "Programm PlcProg in ST editieren" auf Seite 53](#)
- 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 "TRAFFIC-SIGNAL" und eine Instanz "LIGHT2" ebenfalls vom Typ "TRAFFICSIGNAL".

Der Funktionsbaustein "TRAFFICSIGNAL" wird somit instanziiert.

Legen Sie die Variable `LIGHT1: TRAFFICSIGNAL;` und `LIGHT2: TRAF-FICSIGNAL` an:

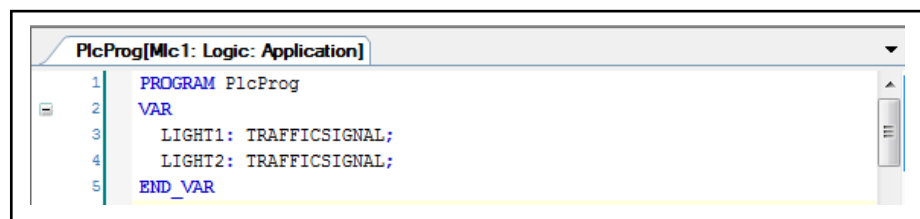


Abb. 2-56: Deklarationsteil POU *PlcProg*

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:

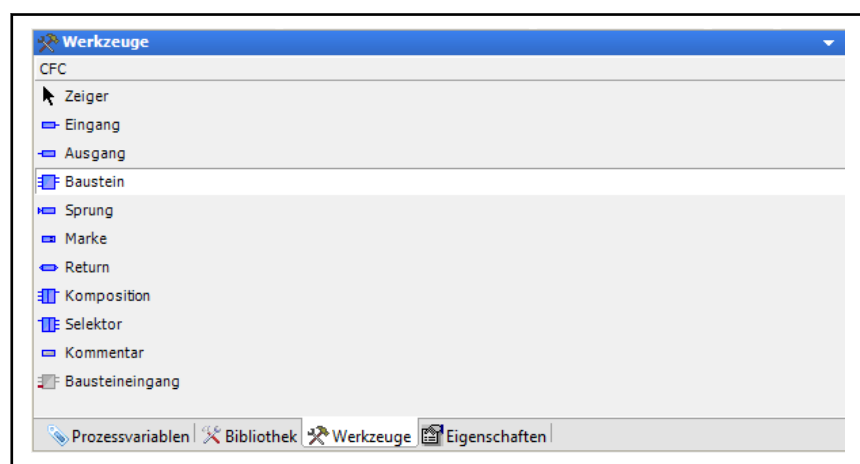


Abb. 2-57: Werkzeuge – Element "Baustein"

2. Bewegen Sie den Mauszeiger in den Rumpf des Editors "PlcProg" und klicken Sie auf die linke Maustaste, damit der Baustein eingefügt wird.

3. Klicken Sie den Namen "???" des Bausteins an und geben Sie dem Baustein den Namen "SEQUENCE".

Der Baustein sieht nun folgendermaßen aus:

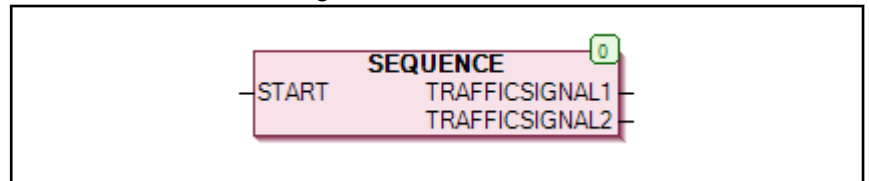


Abb. 2-58: Baustein SEQUENCE

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. Bewegen Sie den Mauszeiger in den Rumpf des Editors "PlcProg" und klicken Sie auf die linke Maustaste, damit der Baustein 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:

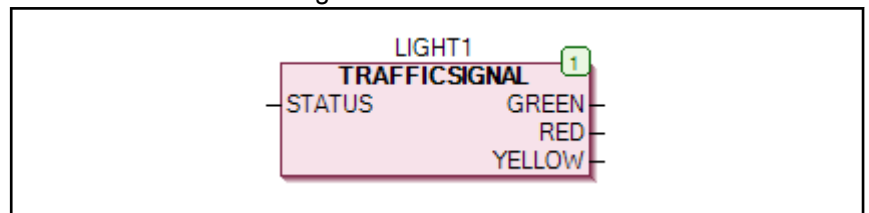


Abb. 2-59: Baustein TRAFFICSIGNAL – LIGHT1

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. Bewegen Sie den Mauszeiger in den Rumpf des Editors "PlcProg" und klicken Sie auf die linke Maustaste, damit der Baustein 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:

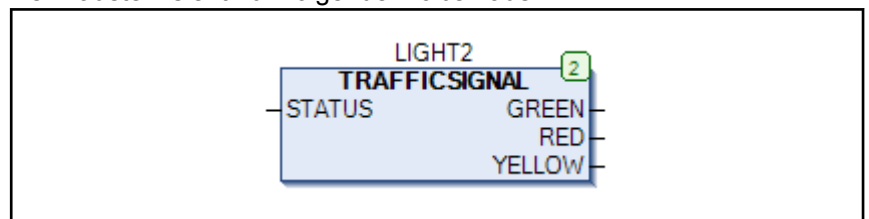


Abb. 2-60: Baustein TRAFFICSIGNAL – LIGHT2

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

Erste Schritte – IndraLogic

1. Wählen Sie im Fenster "Werkzeuge" das Element "Eingang" mit einem Klick aus.
2. Bewegen Sie den Mauszeiger in den Rumpf des Editors "PlcProg" und klicken Sie auf die linke Maustaste, damit der Eingang 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. Bewegen Sie den Mauszeiger in den Rumpf des Editors "PlcProg" und klicken Sie auf die linke Maustaste, damit der Ausgang 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:

1. Klicken Sie den Ausgang des Elements "ON" mit der linken Maustaste an und halten Sie die Maustaste 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 die Maustaste 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 die Maustaste 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 die Maustaste gedrückt. Bewegen Sie den Mauszeiger auf den Eingang "A1_green" des Ausganges, 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:

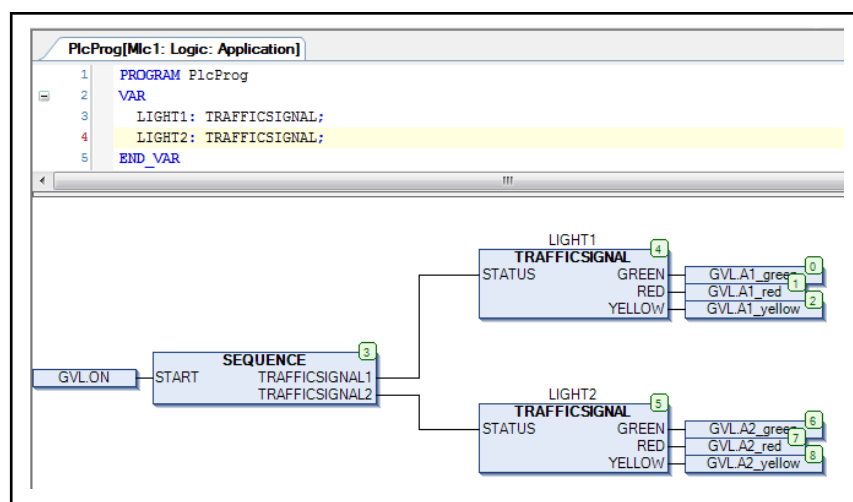


Abb. 2-61: POU "PlcProg" fertiggestellt



Die grünen Nummern an der rechten oberen Ecke der Elemente zeigen die Ausführungsreihenfolge der Elemente an.

Diese Reihenfolge - und damit gegebenenfalls die Funktion der CFC-Schaltung - kann durch den Programmierer bei Bedarf verändert werden.



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

Applikation 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 ► Applikation 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 "Applikation" **Aktive Applikation setzen**).

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 "Mlc1" 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 in der IndraWorks-Menüleiste **Debuggen ► Einloggen**. Sie können die neue Applikation sowohl per Online-Änderung als auch neu übersetzt in die Steuerung laden.

Applikation starten

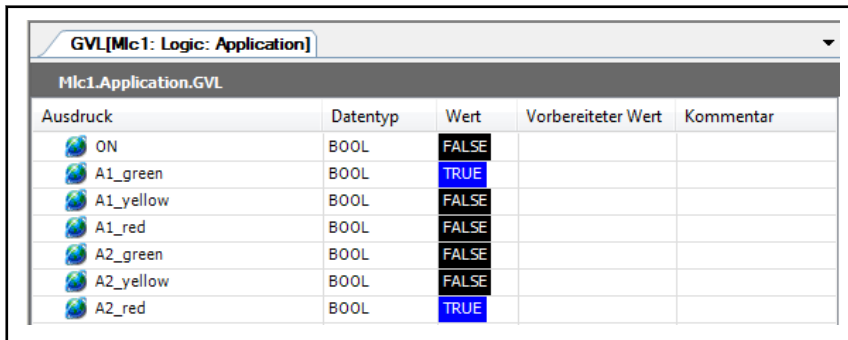
Um die Applikation zu starten, klicken Sie mit der rechten Maustaste auf den Knoten "Applikation" und wählen im Kontextmenü **Start**. Die "Applikation" läuft nun als Simulation ab.

Erste Schritte – IndraLogic

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 eingeschaltet 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 Projektexplorer die "Globale Variablen Liste".



Ausdruck	Datentyp	Wert	Vorbereiteter Wert	Kommentar
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. 2-62: "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 in der IndraWorks-Menüleiste auf **Debuggen ► Werte schreiben**.

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

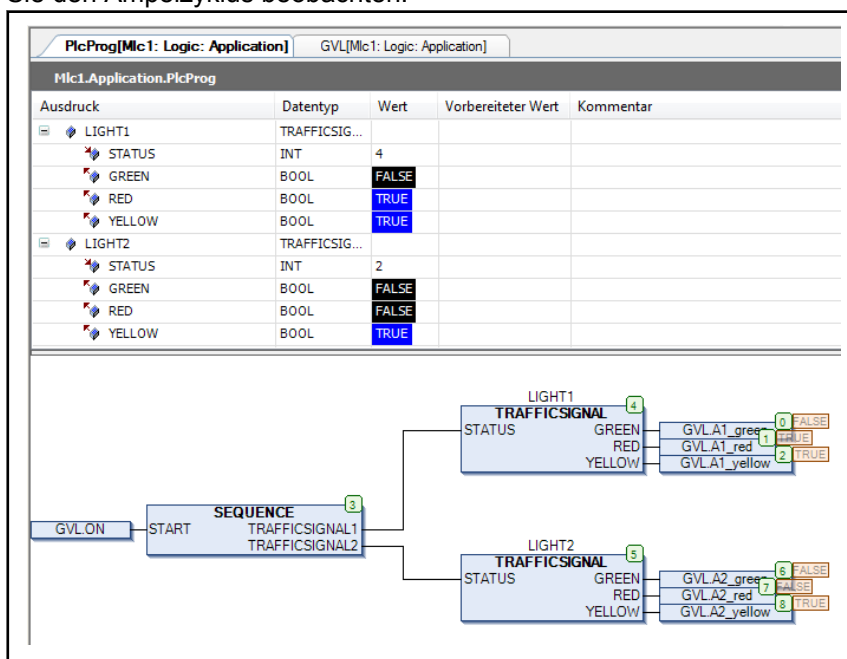


Abb. 2-63: Programm "PlcProg" online

Applikation stoppen

Um das SPS-Programm zu stoppen, wählen Sie in der IndraWorks-Menüleiste **Debuggen ► Stop**.

Aus Steuerung ausloggen

Um sich aus der Steuerung auszuloggen, wählen Sie in der IndraWorks-Menüleiste **Debuggen ► Ausloggen**.

2.6.8 Testen der Applikation auf der Steuerung

Simulationsmodus deaktivieren

Falls vorhanden, soll die Applikation in einer Steuerung getestet werden.

Dazu müssen die [Kap. 2.5.2 "Kommunikationseinstellungen"](#) auf Seite 19 und der gewünschten Steuerung übereinstimmen.

Beenden Sie den Simulationsmodus, indem Sie im "Projektexplorer" auf die Steuerung "Mlc1" mit einem Klick auf die rechte Maustaste das Kontextmenü öffnen. Klicken Sie auf **Simulation** um den Simulationsmodus zu deaktivieren. Ist die Simulation deaktiviert, ist kein Haken bei "Simulation" gesetzt.

Applikation 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 in der IndraWorks- Menüleiste **Erstellen ► Übersetzen** aus. In der "Aufgabenliste" unterhalb des Arbeitsfensters steht nach dem Übersetzen: "Übersetzung abgeschlossen -- 0 Fehler, 0 Warnungen"



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 "Applikation" **Aktive Applikation 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 in der IndraWorks-Menüleiste **Debuggen ► Einloggen**.

Die Applikation steht danach im Zustand "STOP".

Applikation 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.

Bootapplikation erzeugen

Erzeugen Sie über die IndraWorks-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 Projektexplorer die Globale Variablen Liste.

Ausdruck	Datentyp	Wert	Vorbereiteter Wert	Kommentar
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. 2-64: "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 in der IndraWorks-Menüleiste auf **Debuggen ► Werte schreiben**.

Der Ampelzyklus wird nun durchlaufen.

Erste Schritte – IndraLogic

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.

TRAFFICSIGNAL[Mlc1: Logic: Application]

Mlc1.Application.PlcProg.LIGHT1

Ausdruck	Datentyp	Wert	Vorbereiteter Wert	Kommentar
STATUS	INT	4		
GREEN	BOOL	FALSE		
RED	BOOL	TRUE		
YELLOW	BOOL	TRUE		

1

STATUS 4 EQ GREEN FALSE

TRAFFICSIGNAL[Mlc1: Logic: Application]

Mlc1.Application.PlcProg.LIGHT2

Ausdruck	Datentyp	Wert	Vorbereiteter Wert	Kommentar
STATUS	INT	2		
GREEN	BOOL	FALSE		
RED	BOOL	FALSE		
YELLOW	BOOL	TRUE		

1

STATUS 2 EQ GREEN FALSE

Abb. 2-65: Statusanzeige in den beiden FB-Instanzen von TRAFFICSIGNAL

Kontrolle der realen Ein- und Ausgänge

Applikation stoppen

Aus Steuerung ausloggen

Die Steuerungen IndraMotion MLC L45, L65, L75 besitzen Onboard-Ausgänge, die den Adressen %QX0.0 ... %QX0.7 zugeordnet sind (Mapping). Sie können an diesen Ausgängen zusätzlich den Ampelzyklus verfolgen.

Um das SPS-Programm zu stoppen, wählen Sie in der IndraWorks-Menüleiste **Debuggen ► Stop**.

Um sich aus der Steuerung auszuloggen, wählen Sie in der IndraWorks-Menüleiste **Debuggen ► Ausloggen**.

2.7 Visualisierung zum SPS-Programm erstellen

2.7.1 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 [Kap. 2.6 "SPS-Programm erstellen" auf Seite 25](#) beschriebene SPS-Programm soll visualisiert werden. Hierzu sollen nachfolgend zur programmierten Ampelanlage zwei Ampeln und ein Einschalter zwecks Visualisierung gezeichnet und ihren Zuständen entsprechend dargestellt werden.

2.7.2 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 Projektextplorer auf die Applikation ziehen.

Folgendes Fenster öffnet sich:

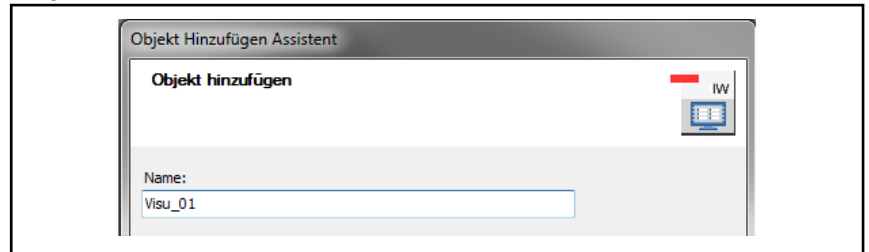


Abb. 2-66: Visualisierung der Applikation hinzufügen

2. Geben Sie der Visualisierung einen beliebigen Namen und bestätigen Sie die Eingabe mit **Fertigstellen**.

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

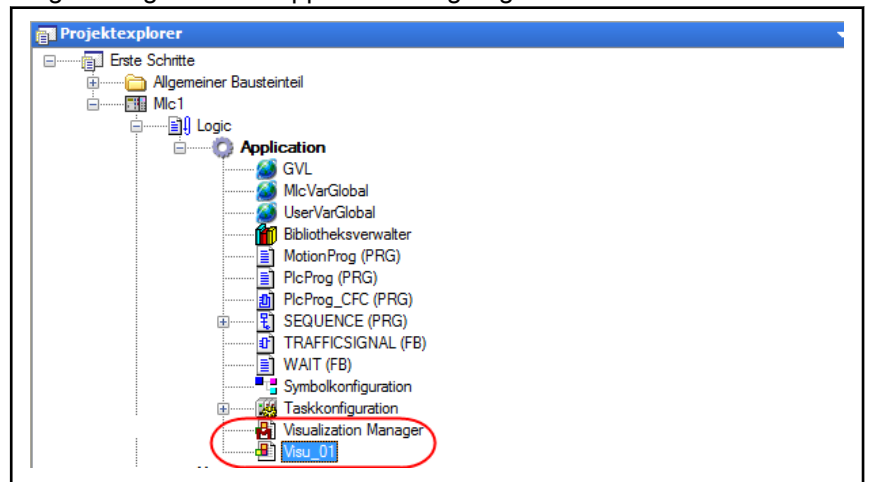


Abb. 2-67: Visualisierung in Applikation eingefügt

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

2.7.3 Visualisierung editieren

Erstes Element "Ampellampe" einfügen und editieren

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

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 fünf Ordner in der Werkzeug-Box:

- **Basis:** allgemeine Visualisierungsobjekte
- **Allgemeine Steuerelemente:** Schaltflächen, Checkboxes, Ladebalken....

Erste Schritte – IndraLogic

- **Messgeräte:** Balkenanzeigen, Zeigerinstrumente, Potentiometer, Histogramm...
- **Lampen, Schalter. Bilder:** Lampen, Dreh- und Kippschalter, Bildwechsler...
- **Spezielle Steuerelemente:** Trace-Objekte, Wartesymbole...

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).

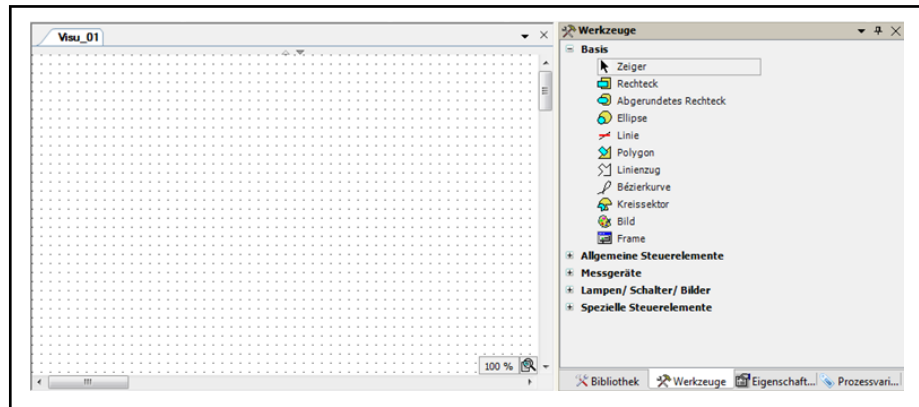


Abb. 2-68: Fenster "Werkzeuge"

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 den Mauszeiger in das Fenster der Visualisierung und klicken Sie erneut auf die linke Maustaste.

Ein Element "Ellipse" wurde in den Visualisierungseitor eingefügt.

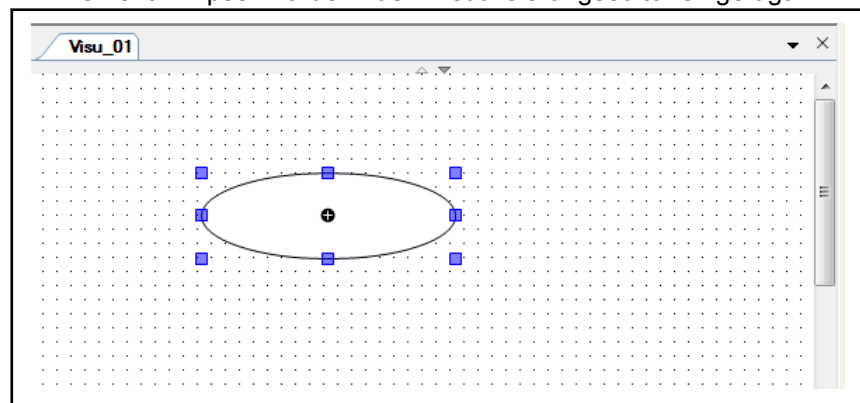


Abb. 2-69: Editor Visualisierung – Ellipse

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.

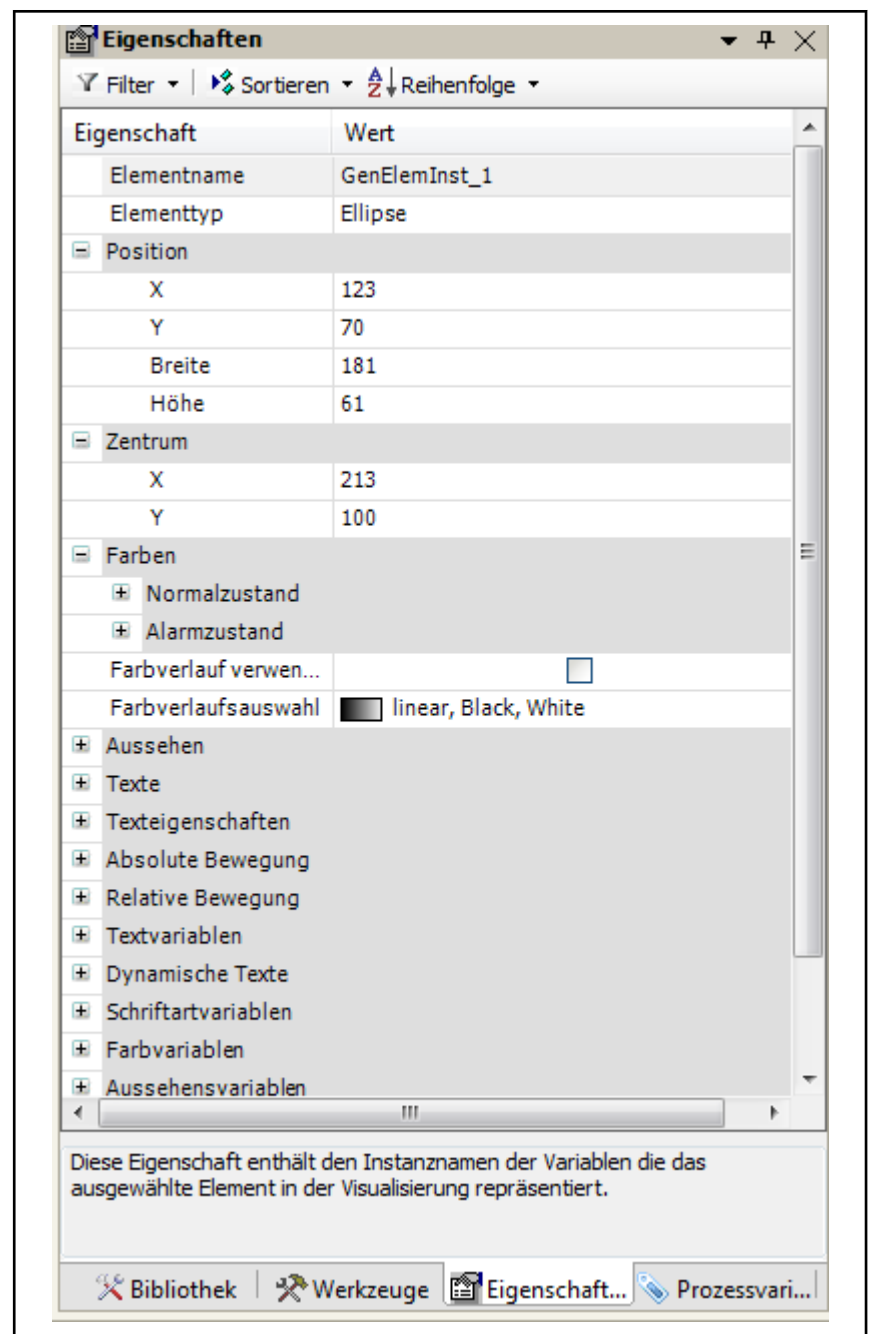


Abb. 2-70: Fenster "Eigenschaften" der Ellipse

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

Erste Schritte – IndraLogic

**Eigenschaften der Ampellampe
verändern**

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

Um die Füllfarben zu verändern gehen Sie wie folgt vor:

1. Um die Füllfarbe für den Normalzustand zu verändern, klicken Sie unter Eigenschaft "Farben" und "Normalzustand" unter "Füllfarbe" auf die Schaltfläche

Das Fenster "Farbe" öffnet sich.

2. Klicken Sie im Fenster "Farbe" auf die Schaltfläche **Füllfarbe**. Wählen Sie "grau".

3. Um die Füllfarbe für den Alarmzustand zu verändern, klicken Sie unter Eigenschaft "Farben" und "Alarmzustand" unter "Füllfarbe" auf die Schaltfläche ...

Das Fenster "Farbe" öffnet sich.

4. Klicken Sie im Fenster "Farbe" auf die Schaltfläche **Füllfarbe**. Wählen Sie "rot".

Im normalen Zustand (Normalzustand) hat die Ampelleuchte die Farbe grau. Die Ampelleuchte soll rot leuchten, wenn der Status der Ampel "Rot" ist. Die Ampel leuchtet rot, wenn die Variable "A1_rot" der "GVL" in der POU "PlcProg" auf "TRUE" gesetzt wird. Um den Zustand der Ampellampe zu dynamisieren, müssen Sie in den Eigenschaften der Ampellampe die Variable für das Ändern der Signalfarbe angeben.

Gehen Sie hierzu wie folgt vor:

1. Um der Visualisierung die Variable zur Änderung der Signalfarbe mitzuteilen, klicken Sie doppelt unter der Eigenschaft "Farbvariablen" und "Farbumschlag" in die Spalte "Wert".

2. Klicken Sie auf die Schaltfläche

Das Fenster "Eingabehilfe" öffnet sich.

3. Unter "Elemente" im Fenster der "Eingabehilfe" sehen Sie "Application", "GVL".

4. Wählen Sie unter den "GVL-Variablen" die Variable "A1_red" aus.

Die "GVL-Variable" wurde in "Farbumschlag" eingefügt:

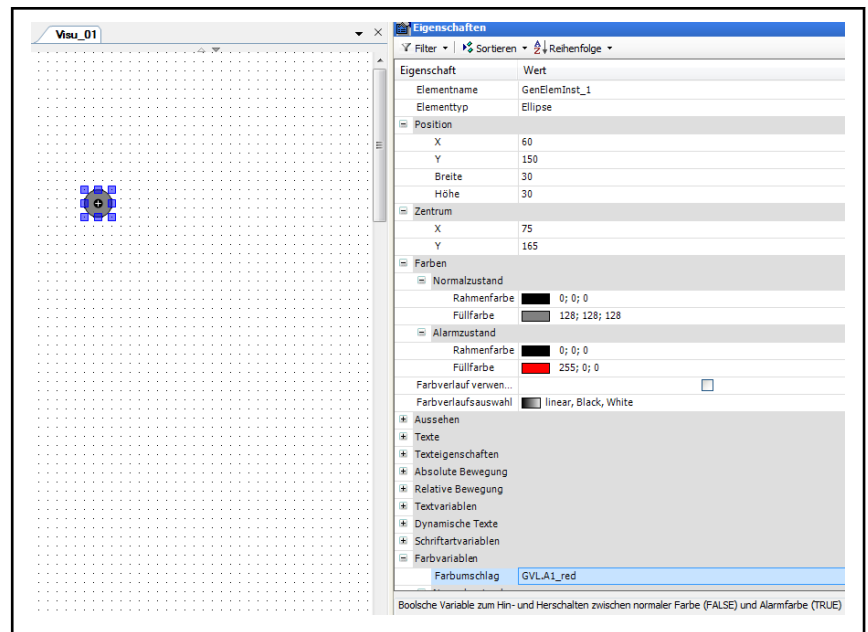


Abb. 2-71: Eigenschaften – Farbumschlag

Element "Ampellampe" kopieren, einfügen und editieren

Element "Ampellampe" kopieren und einfügen

Die erste Ampellampe mit der Signalfarbe "rot" ist nun fertiggestellt. Für die erste Ampelleuchte fehlen nun noch die Ampellampen für die Signalfarbe "gelb" und "grün". Da die zwei anderen Ampellampen die gleiche Form wie die erste Ampellampe besitzen, ist es am Einfachsten, die erste Ampellampe zu kopieren.

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 Ampellampen müssen nun positioniert werden.

Um die Ampellampen 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 "Zentrum" 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).

Erste Schritte – IndraLogic

**Elemente "Ampellampe"
Eigenschaften ändern**

- Um den Mittelpunkt zu verändern, wählen Sie unter Eigenschaft "Zentrum" für (X, Y) = (75, 255).

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

Gehen Sie hierzu wie folgt vor:

- Markieren Sie die mittlere Ampellampe.
- Um die Füllfarbe für den Alarmzustand zu verändern, klicken Sie unter Eigenschaft "Farben" und "Alarmzustand" unter "Füllfarbe" auf die Schaltfläche
- Das Fenster "Farbe" öffnet sich.
- Klicken Sie im Fenster "Farbe" auf die Schaltfläche **Füllfarbe**. Wählen Sie "gelb".
- Wiederholen Sie Schritt 1 - 3 für die untere Ampellampe.
- Klicken Sie im Fenster "Farbe" auf die Schaltfläche **Füllfarbe**. Wählen Sie "grün".

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

Gehen Sie hierzu wie folgt vor:

- Markieren Sie die mittlere Ampellampe.
- Um der Visualisierung die Variable zur Änderung der Signalfarbe mitzuteilen, klicken Sie doppelt unter der Eigenschaft "Farbvariablen" und "Farbumschlag" in die Spalte "Wert".
Klicken Sie auf die Schaltfläche
Der Dialog "Eingabehilfe" öffnet sich.
- Unter "Elemente" im Fenster der "Eingabehilfe" sehen Sie "Application", "GVL".
- Wählen Sie unter den "GVL-Variablen" die Variable "A1_yellow" aus.
- Markieren Sie die untere Ampellampe.
- 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:

- Das Gehäuse der Signallampen.
- Der Mast für das Ampelgehäuse.
- Der Ampelfuß.

Element "Rechteck" einfügen

Um das Gehäuse der Signallampen einzufügen gehen Sie wie folgt vor:

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

**Elemente "Ampelgehäuse"
Position ändern**

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

- Markieren Sie das eingefügte Element "Rechteck" 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 (Breite, Höhe) = (50, 150).
 4. Um den Mittelpunkt zu verändern, wählen Sie unter Eigenschaft "Zentrum" 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 "Farben", "Normalzustand" und "Füllfarbe" auf die Schaltfläche
Das Fenster "Farbe" öffnet sich.
 2. Klicken Sie im Fenster "Farbe" 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 "Rechteck" einfügen** Als nächstes wird der Mast des Ampelgehäuses konstruiert.
Gehen Sie dabei wie folgt vor:
1. Wählen Sie aus dem Fenster "Werkzeuge" das Element "Rechteck" aus.
 2. Klicken Sie in einen freien Bereich des Visualisierungseditors damit das Element "Rechteck" eingefügt wird.
- Element "Ampelmast" Position ändern**
1. Markieren Sie das eingefügte Element "Rechteck" 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 (Breite, Höhe) = (10, 195).
 4. Um den Mittelpunkt zu verändern, wählen Sie unter Eigenschaft "Zentrum" für (X, Y) = (75, 380).
- Element "Ampelmast" Eigenschaften ändern**
- Als Füllfarbe für den Mast der Signallampen wählen Sie bitte Dunkelgrau (DimGray) (Farben, Normalzustand, Füllfarbe, (Rot, Grün, Blau) = (105, 105, 105)).

Element "Ampelfuß" einfügen und editieren

- Element "Rechteck" 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 "Rechteck" aus und fügen Sie das Element im Arbeitsfeld ein.
 2. Markieren Sie das eingefügte Element "Rechteck" 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 (Breite, Höhe) = (120, 30).

Erste Schritte – IndraLogic

Element "Ampelfuß"
Eigenschaften ändern

5. Um den Mittelpunkt zu verändern, wählen Sie unter Eigenschaft "Zentrum" für (X, Y) = (75, 495).

Als Füllfarbe für den Ampelfuß der Signallampen wählen Sie bitte Grau (Dim-Gray) (Farben, Normalzustand, Füllfarbe, (Rot, Grün, Blau) = (105, 105, 105)).

Die erste Ampel hat nun folgendes Erscheinungsbild:

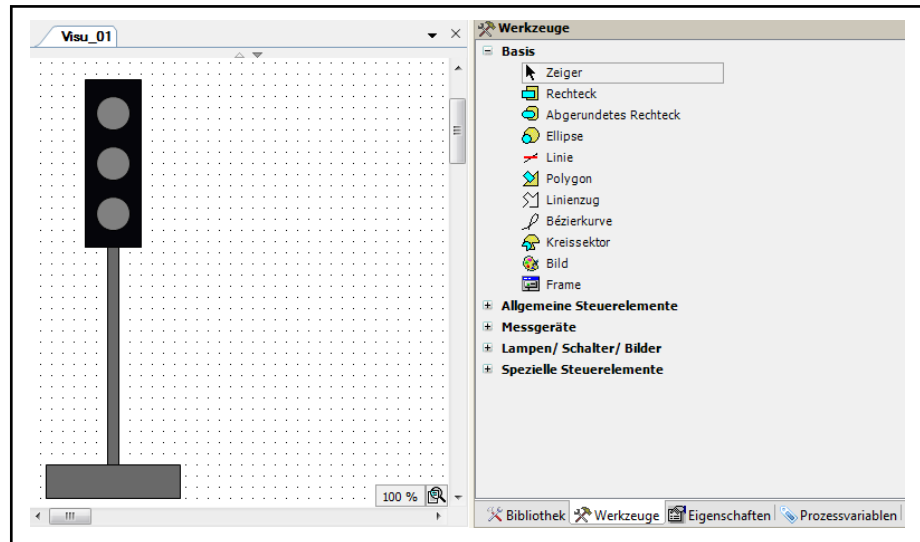


Abb. 2-72: Erste Ampel

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. Bewegen Sie den Mauszeiger auf eine Position links oberhalb der Ampel. Lassen Sie die Maus los. Die Elemente sind nun alle markiert.



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 das Objekt neben der ersten Ampel ab.

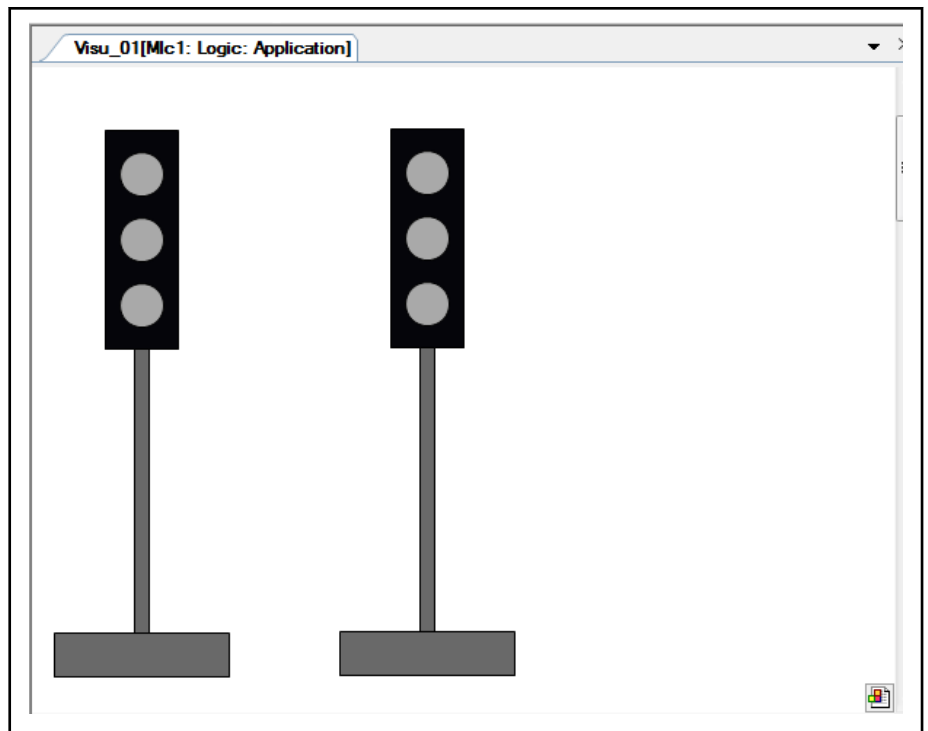


Abb. 2-73: Zwei Ampeln

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

Einschalter einfügen und editieren

Element "Schaltfläche" einfügen

Die Ampel soll zu Beginn eingeschaltet werden können. Hierzu soll eine "Schaltfläche" eingefügt werden:

1. Wählen Sie aus dem Fenster **Werkzeuge ► Allgemeine Steuerelemente** das Element "Schaltfläche" aus.
2. Klicken Sie in einen freien Bereich des Visualisierungseditors, damit das Element "Schaltfläche" eingefügt wird.

Element "Schaltfläche" 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 (Breite, Höhe)= (70, 40).
3. Um den Mittelpunkt zu verändern, wählen Sie unter Eigenschaft "Zentrum" und für (X, Y)= (165, 75).

Element "Schaltfläche" Eigenschaften ändern

1. Unter der Eigenschaft "Texte" und "Text" geben Sie "EIN" ein.
2. Unter der Eigenschaft "Eingabekonfiguration", "OnClick" klicken Sie auf "Konfigurieren".

Das Fenster "Eingabekonfiguration" öffnet sich.

3. Markieren Sie im Fenster "Eingabekonfiguration" im linken Fensterbereich "Variable umschalten" und betätigen Sie die Schaltfläche ►.

Erste Schritte – IndraLogic

"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 "Rechteck" 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 "Rechteck" aus.
2. Klicken Sie in einen freien Bereich des Visualisierungseditors damit das Element "Rechteck" eingefügt wird.

Element "Rechteck" Position ändern

1. Markieren Sie das eingefügte Element "Rechteck" 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 (Breite, Höhe) = (120, 30).
4. Um den Mittelpunkt zu verändern, wählen Sie unter Eigenschaft "Zentrum" für (X, Y) = (75, 535).

Element "Rechteck" Eigenschaften ändern

1. Im Feld "Text" bei "Texte" geben Sie "Ampel 1" ein.
2. Klicken Sie im Fenster "Eigenschaften" unter "Farben", "Normalzustand" und "Rahmenfarbe" 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 "Rechteck" kopieren und Eigenschaften anpassen

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

Die Visualisierung ist nun fertiggestellt:

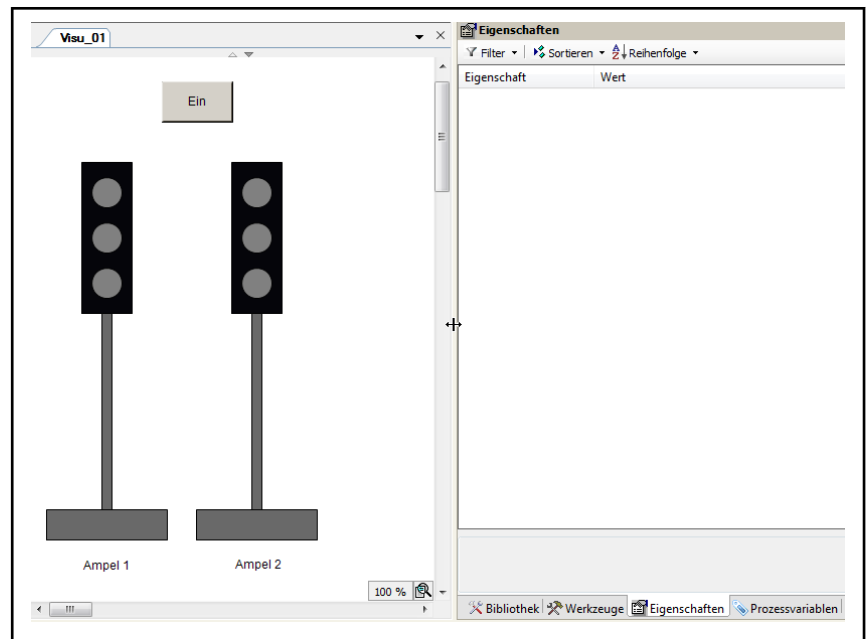


Abb. 2-74: Visualisierung fertiggestellt

Anzeige von Variablenwerten

Element "Rechteck" 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 "Rechteck" aus.
2. Klicken Sie in einen freien Bereich des Visualisierungseditors damit das Element "Rechteck" eingefügt wird.

Element "Rechteck" Position ändern

1. Markieren Sie das eingefügte Element "Rechteck" 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 (Breite, Höhe) = (120, 30).
4. Um den Mittelpunkt zu verändern, wählen Sie unter Eigenschaft "Zentrum" für (X, Y) = (75, 585).

Element "Rechteck" Eigenschaften ändern

1. Im Feld "Text" bei "Texte" geben Sie "Ampel 1: %s" ein.
2. Klicken Sie ins Feld "Text Variable" bei "Textvariablen" 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 "Rechteck" kopieren und Eigenschaften anpassen

1. Kopieren Sie die Anzeige, und legen Sie die Anzeige unter der Ampel zwei ab.
2. Ersetzen Sie für Anzeige 2:
 - Text "Ampel 1: %s" durch "Ampel 2: %s"

Erste Schritte – IndraLogic

- Text Variable "PlcProg.LIGHT1.STATUS" durch "PlcProg.LIGHT2.STATUS"

Visualisierung mit ergänzter Anzeige:

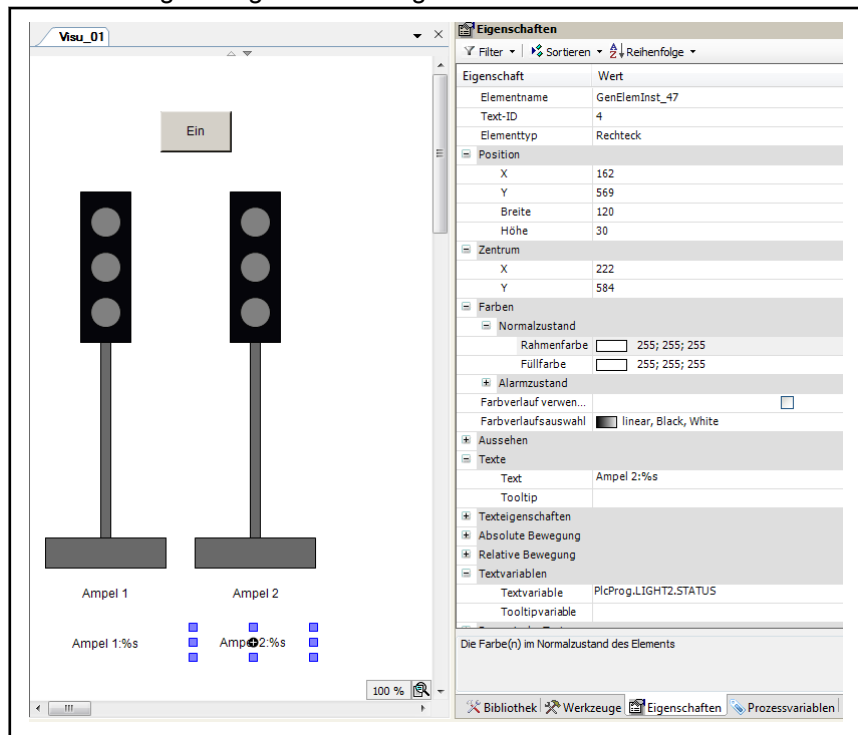


Abb. 2-75: Visualisierung durch Anzeige ergänzt

2.7.4 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 [Kap. "Testen der Applikation im Simulationsmodus, PlcProg in ST"](#) auf Seite 54 bzw. [Kap. "Testen der Applikation im Simulationsmodus, PlcProg in CFC"](#) auf Seite 59.

Um die Visualisierung mit einer Steuerung zu testen, orientieren Sie sich an [Kap. 2.6.8 "Testen der Applikation auf der Steuerung"](#) auf Seite 60.

Öffnen Sie bei gestarteter Applikation das Fenster "Visualisierung":

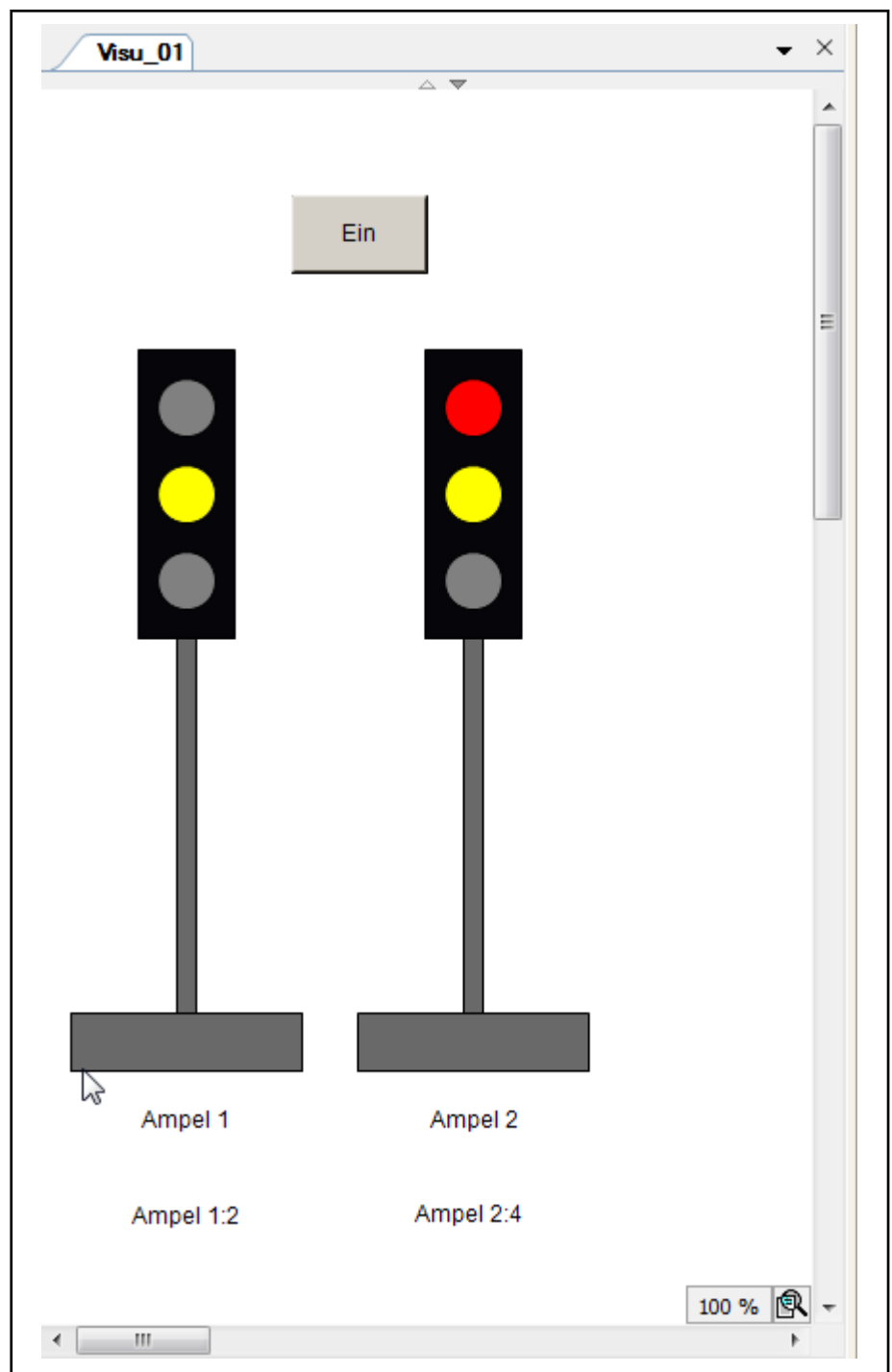


Abb. 2-76: Visualisierung online

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

2.8 Projekt wiederherstellen

Ablage Ausgehend vom erstellten Projekt wurde eine Exportdatei erzeugt, die unabhängig von Ihrer Steuerungshardware (IndraMotion MLC L25, L45, L65, L75, XM2, VPx) erlaubt, das Projekt wiederherzustellen. Das wiederhergestellte Projekt kann als Simulation oder unter Nutzung Ihrer Steuerung laufen.

Erste Schritte – IndraLogic

Die Exportdatei wurde für den Knoten "Application" erstellt und enthält folgende Bestandteile:

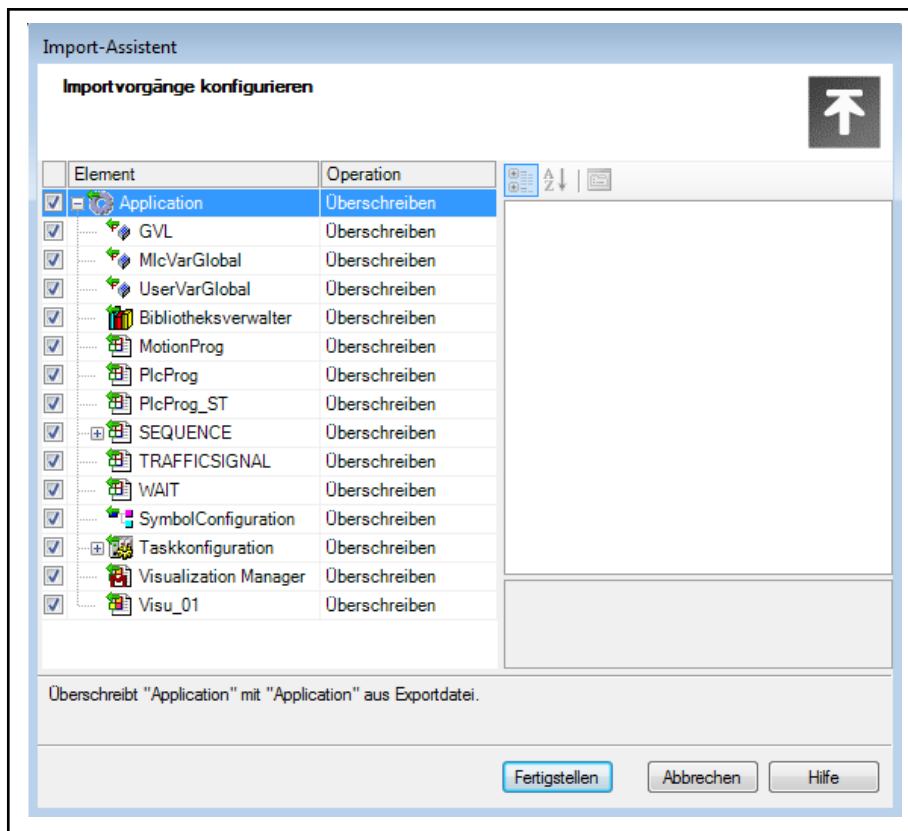


Abb. 2-77: Bestandteile der Exportdatei

Die angegebenen Komponenten werden beim Import überschrieben.

Wiederherstellen

Um das Projekt auf einer Steuerung (IndraMotion MLC L25, L45, L65, L75, XM2, VPx) lauffähig zu machen, gehen Sie wie folgt vor:

1. Löschen von Altdaten in Ihrer Steuerung.
Löschen Sie den alten Speicherinhalt Ihrer Steuerung.
Details sind unter [Kap. 2.3.1 "Steuerung zurücksetzen \(CMLx5\)" auf Seite 16](#) 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 nun über ein neues leeres Projekt.
3. Anlegen der Steuerung.
Ziehen Sie eine Steuerung Ihrer Wahl (IndraMotion MLC L25, L45, L65, L75, XM2, VPx) aus der Gerätebibliothek auf das leere Projekt. Der Steuerungsname ist beliebig, empfohlen wird "Mlc1".
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 nun die gewünschte Steuerung.

4. Import der Exportdatei "Application_FirstSteps_IndraLogic_Basics.iwx".
Öffnen Sie im Projektnavigator den Steuerungsknoten bis **Logic ► Application**.

Wählen Sie im Kontextmenü des "Application"-Knotens **Importieren...** aus.

Wählen Sie die Exportdatei "Application_FirstSteps_IndraLogic_Basics.iwx" in "<Laufwerk>:\Programme\Rexroth\IndraWorks\Projects" aus. Es öffnet sich der Dialog "Importvorgänge konfigurieren". Wählen Sie in diesem Import-Dialog für jedes Element die Operation "Überschreiben" aus. Starten Sie den Importvorgang mit **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 Strukturisiertem Text (ST).

Um das Projekt im Simulationsmodus zu testen, orientieren Sie sich an [Kap. "Testen der Applikation im Simulationsmodus, PlcProg in ST" auf Seite 54](#) bzw. [Kap. "Testen der Applikation im Simulationsmodus, PlcProg in ST" auf Seite 54](#).

Um das Projekt mit einer Steuerung zu testen, orientieren Sie sich an [Kap. 2.6.8 "Testen der Applikation auf der Steuerung" auf Seite 60](#).

3 Erste Schritte – Motion

3.1 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 Arbeitsschritt "Sercos III Antriebe (reale Achsen) zum Projekt hinzufügen".

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

- [Kap. 3.4 "Programmieren des Bewegungsablaufes in Funktionsbausteinsprache \(FUP\)" auf Seite 94](#)
- [Kap. 3.5 "Programmieren des Bewegungsablaufes in Strukturiertem Text \(ST\)" auf Seite 118](#)

3.2 Technische Voraussetzungen

- Software:
 - PC mit installiertem IndraWorks Engineering für IndraMotion MLC 14VRS
- Steuerung:
 - IndraMotion MLC L25
 - IndraMotion MLC L45
 - IndraMotion MLC L65
 - IndraMotion MLC L75
 - IndraMotion MLC XM2
 - IndraMotion MLC VPx

Die Schaffung dieser Voraussetzungen sind beschrieben in:

- IndraMotion MLC 14VRS Funktionsbeschreibung, Kapitel "Inbetriebnahme und Bedienung"

Eine Steuerung, z. B. Typ "IndraMotion MLC L65" mit aktueller Firmware 14VRS, muss vorhanden sein. Zusätzlich müssen noch folgende Voraussetzungen erfüllt sein:

- Die IP-Adresse ist bekannt

Erste Schritte – Motion

- Die Steuerung ist über Ethernet mit dem Programmiergerät (PC) verbunden

Antriebsregelgerät, z. B. Typ "**IndraDrive M**" mit aktueller Firmware:

- Bei realer Achse:
Das Antriebsregelgerät ist mit der Steuerung über Sercos III verbunden
- Die Adresse des Antriebsregelgerätes am Sercos Bus wird gescannt oder
- ein gebrückter Sercos Bus, wenn kein Antrieb vorhanden ist

3.3 Anlegen des Motion-Teilprojektes

3.3.1 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.

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 Anzeige **BOOTSTOP** erscheint.

Danach sind die beiden inneren Tasten zu drücken. Es erscheint die Anzeige **CLEANUP?**.

Bestätigen Sie mit <Enter>. Es erscheint die Anzeige **CLEANUP**. Wenn der CleanUp erfolgreich war, erscheint die Anzeige **SUCCESS** und dann **REBOOT**.

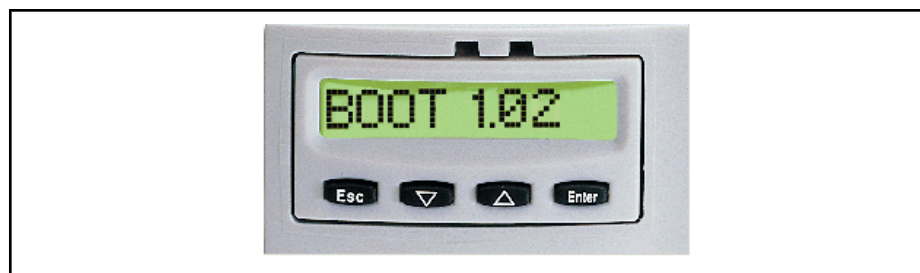


Abb. 3-1: Reboot der Steuerung



Ein Fehler "F0140001, Gerätesollkonfiguration fehlerhaft" kann auftreten und ist mit der <ESC>-Taste zu quittieren.

Die Steuerung läuft nun bis **P2 RDY**. Der Speicher ist gelöscht, eine vorhandene Motion-Komponente ist in der Phase P2 und die SPS-Komponente ist betriebsbereit.

3.3.2 IndraWorks starten

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

Die Entwicklungsumgebung von IndraWorks wird gestartet. Alternativ kann

IndraWorks über das Desktop-Symbol  gestartet werden.

3.3.3 Neues Projekt erstellen

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

Erste Schritte – Motion

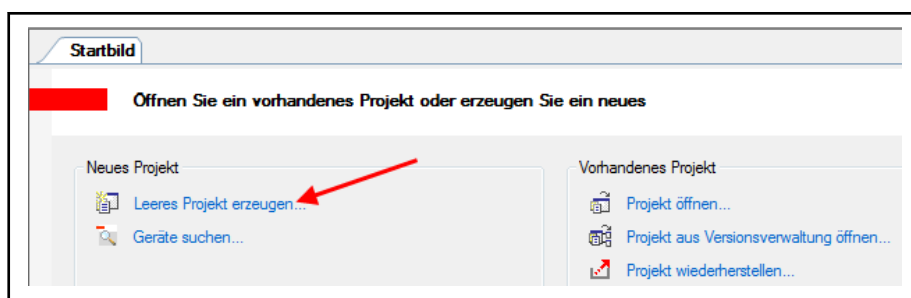


Abb. 3-2: Leeres Projekt erzeugen

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.

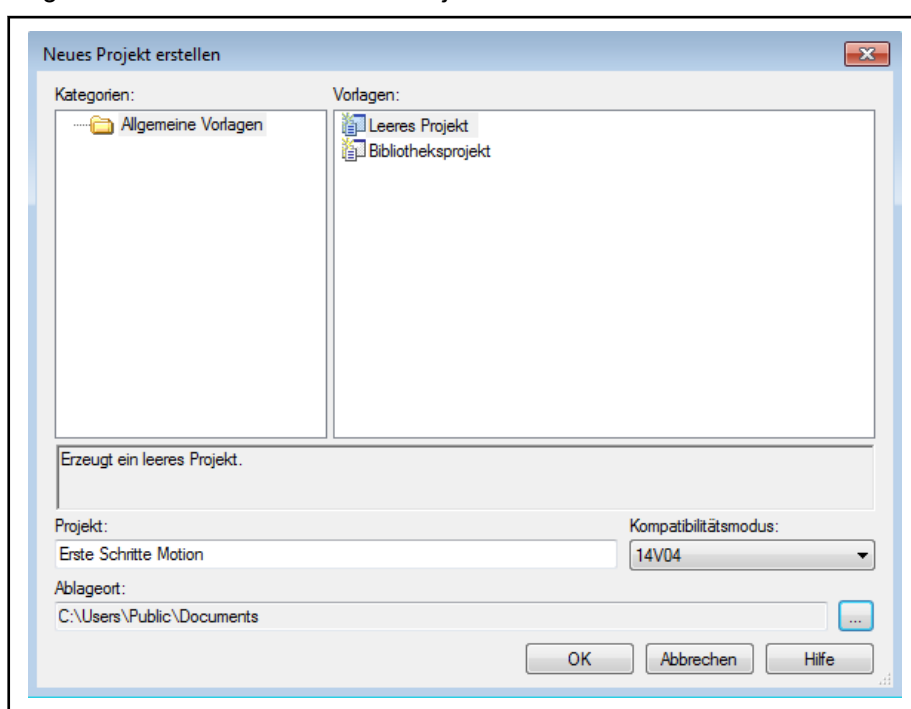


Abb. 3-3: Einstellungen zum Abspeichern eines neuen IndraWorks-Projektes

Projektname Geben Sie dem Projekt einen beliebigen Namen, Vorschlag: "Erste Schritte Motion".

Ablageort Das Feld enthält voreingestellt einen Speicherplatz im Verzeichnis "C:\Dokumente und Einstellungen\<username>\Eigene Dateien".

Alternativ können Sie auch ein anderes Verzeichnis angeben.

Kompatibilitätsmode Der Leistungsumfang der IndraWorks-Oberfläche entspricht der Version 14V04.

Projektsprache Mit der Auswahl der Projektsprache legen Sie die Sprache ihrer Anwender-texte fest.

Das neu angelegte Projekt erscheint im Projektexplorer mit dem vorher definierten Projektnamen:



Abb. 3-4: Projekt "Erste Schritte Motion"

Erste Schritte – Motion

3.3.4 Neues Gerät in ein Projekt einfügen

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



- (1) Gerät: IndraMotion MLC L65
(2) Projekt: Erste Schritte Motion

Abb. 3-5: Gerät einfügen

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

3.3.5 Basiseinstellungen zum Gerät vorgeben

Geräteeeinstellungen

Geben Sie die allgemeinen Daten zum eingefügten Gerät an.

Abb. 3-6: Geräteeeinstellungen

- Gerätemame** Geben Sie bei "Gerätemame" einen beliebigen Gerätenamen ein, z. B. "Mlc1". Das Feld enthält voreingestellt den Namen aus der Bibliothek. Das Gerät erscheint später unter dem angegebenen Namen im Projektextplorer.
- 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.

Geben Sie den Namen des Projekterstellers an. Dieses Feld ist optional.

Kommunikationseinstellungen

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

IndraMotion MLC L65 einfügen

Schritt 2: Hardware / Kommunikation
Wählen Sie die Gerätekonfiguration aus und legen Sie die Kommunikationseinstellungen fest.

Gerätekonfiguration

Gerätetyp:

Firmware-Version:

Firmware-Release:

Ethernet-Kommunikation

IP-Adresse:

SPS-Gateway:

SPS-Kommunikation: ☒ TCP ☐ UDP

Verbindungstest:

Es wurde noch kein Verbindungstest durchgeführt.

Abb. 3-7: Anlegen einer Steuerung (im Beispiel "IndraMotion MLC L65") – Hardware / Kommunikation

- | | |
|--------------------------|---|
| Gerätetyp | Der Gerätetyp ist durch die Steuerungsauswahl vorbestimmt. Gibt es für einen Gerätetyp mehrere unterschiedliche Varianten, muss aus der Drop-Down-Liste die Geräte-Variante ausgewählt werden. |
| 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 Rechnernamen (Adresse). "localhost" bedeutet, dass das lokal installierte Gateway angesprochen wird. Diese Einstellung sollte nicht verändert werden. |
| SPS-Kommunikation | In der Version 14VRS wird nur noch TCP an der Engineering-Schnittstelle unterstützt.

UDP wird noch aus Kompatibilitätsgründen für Projekte aus der Version 13VRS angeboten. |
| Verbindungstest | Klicken Sie auf die Schaltfläche Ausführen , um zu überprüfen, ob die Kommunikation zwischen Rechner und der Zielsteuerung erfolgreich ist. Das Ergebnis des Tests wird im Feld rechts neben "Verbindungstest" protokolliert.
Bei erfolgreichem SPS-Kommunikationstest wird die IP-Adresse angezeigt.
Eventuell auftretende Probleme werden im Fenster durch blinkende rote oder gelbe Icons gekennzeichnet.
Die folgenden Probleme können auftreten: |

Erste Schritte – Motion

- Ausgabe: "Kommunikation zur Steuerung <Steuerungsname> mit der IP-Adresse <IP-Adresse> 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 (siehe IndraMotion MLC 14VRS Funktionsbeschreibung, Kapitel "Inbetriebnahme und Bedienung", Unterkapitel "Einstellen der IP-Adresse")

- Die Netzwerkmaske des PC und/oder der Steuerung ist falsch eingestellt

Lösung: IP-Adresse im Dialog ändern oder IP-Adresse der Steuerung ändern (siehe IndraMotion MLC 14VRS Funktionsbeschreibung, Kapitel "Inbetriebnahme und Bedienung", Unterkapitel "Einstellen der IP-Adresse")

- Ausgabe: "Die projektierte Firmware stimmt nicht mit der Firmware in der Steuerung überein."

Ursachen:

- Die Firmware der Steuerung ist nicht aktuell

Lösung: Firmwareupdate durchführen (siehe IndraMotion MLC 14VRS Funktionsbeschreibung, Kapitel "Steuerung", Unterkapitel "Firmwareverwaltung")

- Die ausgewählte Steuerung entspricht nicht dem realen Steuerungstyp

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

SPS - Eigenschaften

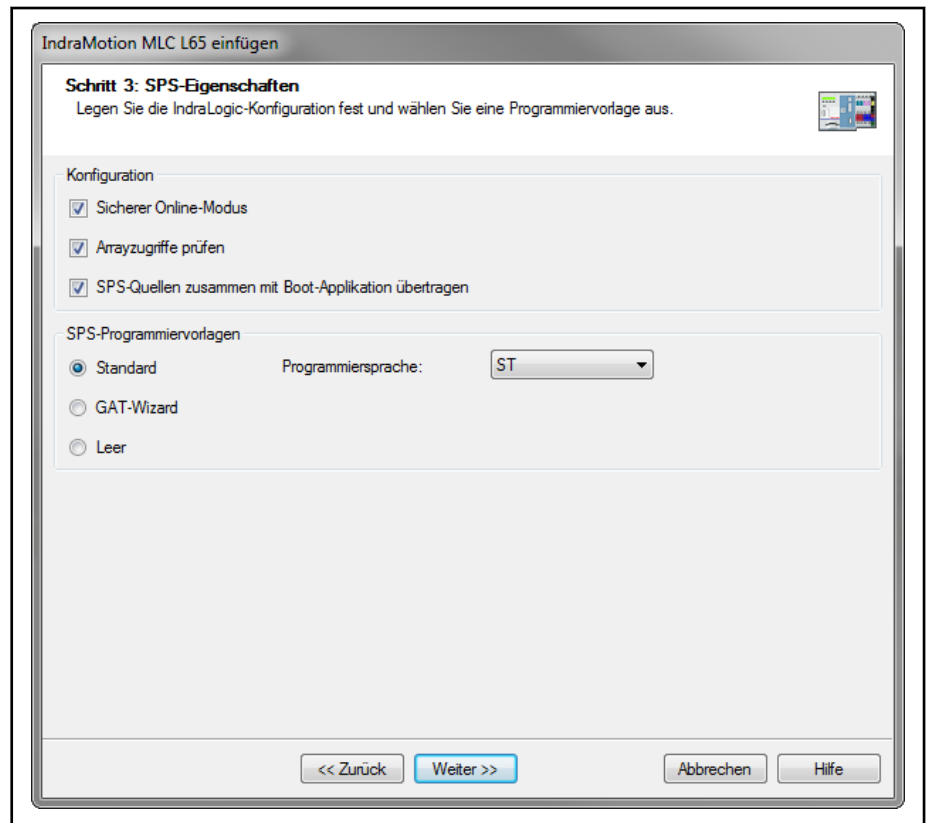


Abb. 3-8: Anlegen einer Steuerung – SPS-Eigenschaften

Sicherer Online-Modus

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

Arrayzugriffe prüfen

Ist dieser Schalter aktiviert, dann wird automatisch die Checkfunktion "CheckBounds" aus der Vorlagendatenbank unterhalb des SPS-Applikationsknotens eingefügt.

Wird dieser Schalter bei einer bereits bestehenden Steuerung wieder deaktiviert, so wird die Checkfunktion "CheckBounds" nur dann automatisch wieder entfernt, sofern der folgende Kommentarblock im Implementierungsteil der Checkfunktion noch enthalten ist:

```
(*****  
This function was generated by the 'check array access'  
option of the controllers 'properties' dialog box. If you  
delete this comment, this function will not be removed by  
disabling the 'check array access' option.  
*****)
```

Durch Entfernen des Kommentarblocks kann verhindert werden, dass kundenspezifische Anpassungen im Implementierungsteil der Checkfunktion durch Deaktivieren des Schalters gelöscht werden.

Erste Schritte – Motion



Da mit dem Löschen des Kommentarblockes die automatisch eingefügte Checkfunktion nicht mehr entfernt wird, muss die Funktion "CheckBounds" manuell aus der Applikation gelöscht werden, um die Überwachung der Arraygrenzen zu deaktivieren.

Ein nichtaktivierter Schalter bedeutet somit nicht, dass keine CheckBounds-Funktionalität aktiv ist.

Wird die Checkfunktion eingefügt oder gelöscht, ist kein Einloggen per Onlinechange möglich. Es wird ein Download notwendig.

SPS-Quellen zusammen mit Boot-Applikation übertragen

Die Checkbox ist nur dann aktiv, wenn der SPS-Applikationstyp "Standard" oder "GAT Wizard" aktiviert und keine benutzerdefinierte CheckBounds-Funktion enthalten ist.

Beim Übertragen der Boot-Applikation werden zusätzlich zum Kompilat auch die SPS-Programmquellen (POUs, Tasks usw.) auf die Steuerung übertragen. Verwenden Sie diese Funktion, wenn Sie im Online-Vergleich zusätzlich einen Detailvergleich für SPS-Elemente durchführen wollen.

Programmiervorlagen

Für die zu erzeugende SPS-Applikation kann zwischen folgenden Varianten gewählt werden:

Standard: Es werden automatisch leere Standardprogramme "MotionProg" und "PlcProg" in der gewählten Programmiersprache angelegt. Für weitere SPS-Objekte wird die Programmiersprache beim Anlegen des Objektes ausgewählt.

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

GAT Wizard: In einem zusätzlichen Dialog können Beispielprojekte oder Programmieremplates gewählt werden. Das GAT ist ein lauffähiger Programmierrahmen mit vordefinierten Stellen für Applikationscode. Durch die dialogbasierte Codegenerierung des GAT-Wizards entstehen automatisch professionell gut strukturierte Applikationsprogramme mit vorbereiteten Elementen.

Leer: legt eine leere, nicht lauffähige SPS-Applikation ohne Programme und Tasks an.

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

Funktionspaketauswahl



Die Funktionspakete sind bei den Steuerungen auf Basis IndraControl L25/L45/L65/L75 durch die erworbene Firmware bereits vordefiniert und daher nicht selektierbar. Bei den Steuerungen auf Basis IndraControl XM und VPx sind die Funktionspakete anhand der erworbenen Lizenzen auszuwählen!

1. Wird ein Funktionspaket abgewählt, werden automatisch alle davon abhängigen Funktionspakete ebenfalls abgewählt.
2. Selektiert man ein lizenzpflichtiges Funktionspaket, wird automatisch ein Lizenzicon generiert. Über dieses Icon wird per Tooltip oder per Mausklick die Lizenzierungsinformationen angezeigt.
3. In Abhängigkeit von den aktivierten Funktionspaketen werden anschließend im IndraWorks Projektbaum die Knoten unterhalb der angelegten Steuerung aufgebaut.
4. Die Funktionspakete können auch nach dem Anlegen einer Steuerung über den zugehörigen Dialog "Eigenschaften" hinzugefügt bzw. entfernt werden.

5. Bei einer Konvertierung des Gerätes in einen Steuerungstyp mit anderem Funktionsumfang erscheint wie bisher eine Konvertierungsvorschau die dem Anwender über die bevorstehenden Änderungen informiert.



Wird z. B. das Funktionspaket "Motion" entfernt, werden auch alle konfigurierten Achsen gelöscht.

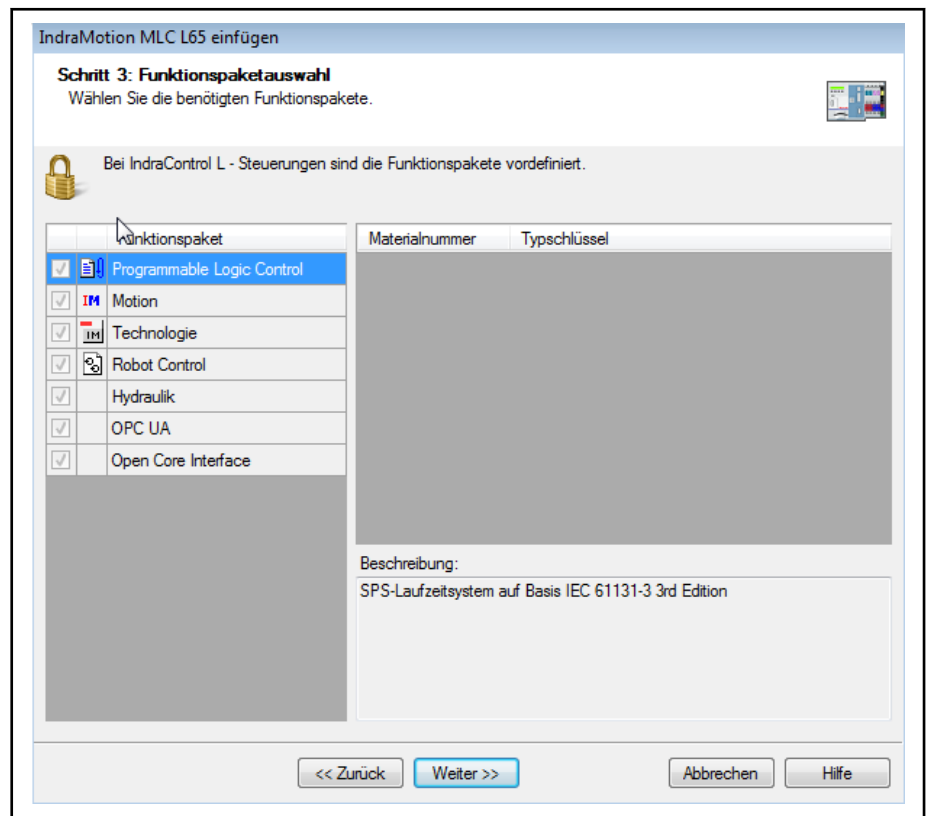


Abb. 3-9: Übersicht der erworbenen Funktionspakete

Programmable Logic Control

Das Funktionspaket "Programmable Logic Control" beinhaltet das SPS-Laufzeitsystem auf Basis der IEC 61131-3 3rd Edition. Das Funktionspaket ist standardmäßig immer aktiviert und kann nicht abgewählt werden.

Motion

Firmware-basierte Ansteuerung von Einzel- und synchronisierten Mehrachs-bewegungen.

Technologie

SPS-Bibliotheken für Technologiefunktionen.

Robot Control

Allgemeine Kinematiken und Transformationen für die Ansteuerung von Robotern.

Hydraulik

Zentrale Regelung von Hydraulikachsen.

OPC UA

OPC UA Server, OPC Unified Architecture, standardisierter Variablenaus-tausch zwischen Steuerung und Visualisierung.

Open Core Interface

Schnittstelle für Rexroth Open Core Engineering.

Diese als Motion-Logic Programming Interface (MLPI/OCI) bezeichnete Programmierschnittstelle ermöglicht dem Anwender den Zugriff auf Funktionalitäten, die in der Firmware des Systems IndraMotion MLC gekapselt sind. So können schnell und einfach Motion-Control Applikationen in den Programmiersprachen C/C++/C#/VBA/JAVA mit der Entwicklungsumgebung "Visual Studio" entwickelt werden (siehe [Engineering Network](#)).

Erste Schritte – Motion

Funktionsmoduleinstellungen



Der Dialog "Schnittstellen" kann sich, je nach Gerätetyp, unterscheiden. In diesem Projekt wird exemplarisch der Dialog des Gerätes "IndraMotion MLC L65" beschrieben.

Geben Sie die gewünschten Geräte- und Funktionsmoduleinstellungen ein.

Abb. 3-10: Funktionsmoduleinstellungen

Profibus DP (X7P)

Aktuelle Profibuskonfiguration, die beim Einfügen des Gerätes gewählt werden kann. Zur Auswahl stehen "Profibus-DP Master", "Profibus-DP Slave" oder "Nicht verwendet".

Realtime Ethernet (X7E3/X7E4)

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 (X7E5)

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.

Funktionsmodule

An eine "IndraMotion MLC L65/L75 / IndraLogic XLC L65/L75" können bis zu vier Funktionsmodule angeschlossen werden.

Bei Bedarf können Sie die Funktionsmodule aus der Drop-Down-Liste auswählen.

Bestätigen Sie die Eingaben mit der Schaltfläche **Fertigstellen**.

Im Projektexplorer erscheint das neu angelegte Gerät.

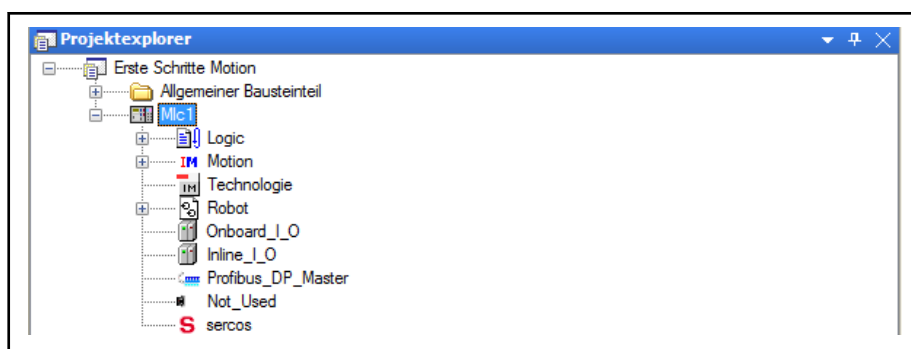


Abb. 3-11: Projektexplorer: Projekt "Erste Schritte Motion" mit einer IndraMotion MLC L65-Steuerung mit Sercos III Bus



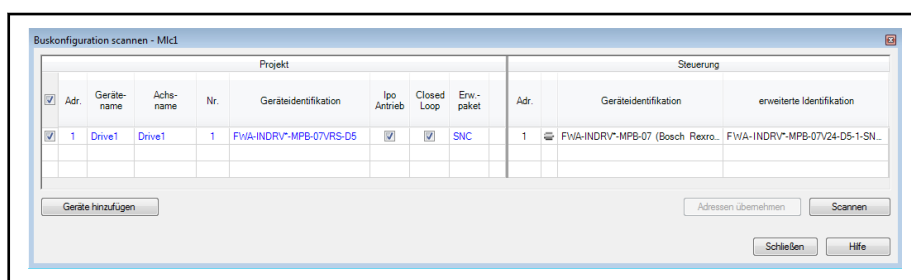
Der Projektnavigator widerspiegelt die in den Dialogen "Konfiguration" und "Funktionsmodule" gewählten Komponenten.

3.3.6 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 (Kontextmenü "Sercos" Knoten **Buskonfiguration scannen**).

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

Das Ergebnis wird im Folgebild angezeigt:



Adr	Sercos Adresse
Gerätename	Default-Antrieb _n , umbenannt
Achsname	Name der Achse / des IO-Gerätes
Nr	Achsnummer
Geräteidentifikation	Antriebsfamilie / Firmwareidentifikation
Closed loop	Lageregelung möglich
Ipo Antrieb	Interpolation in Steuerung oder im Antrieb, hier in der Steuerung

Abb. 3-12: Liste der Sercos Antriebe, Gerätename "Drive1" ist editierbar

Betätigen Sie **Geräte hinzufügen**.

Aus der gescannten Buskonfiguration werden die in Spalte 1 markierten Geräte in den Projektexplorer übernommen.

Ein Antrieb wird als "Reale Achse" im Motion-Knoten gespiegelt.

Die Achsnummer "1" und Achsnamen "Drive1" sind je nach Adressierungsart zu verwenden.

Das Projekt enthält die Steuerung "Mlc1" z. B. vom Typ "IndraMotion MLC L65" und die reale Achse "Drive1" (Ordner **Motion** ► **Reale Achsen**) bzw. Antrieb "Drive1" (Ordner **Sercos**):

Erste Schritte – Motion

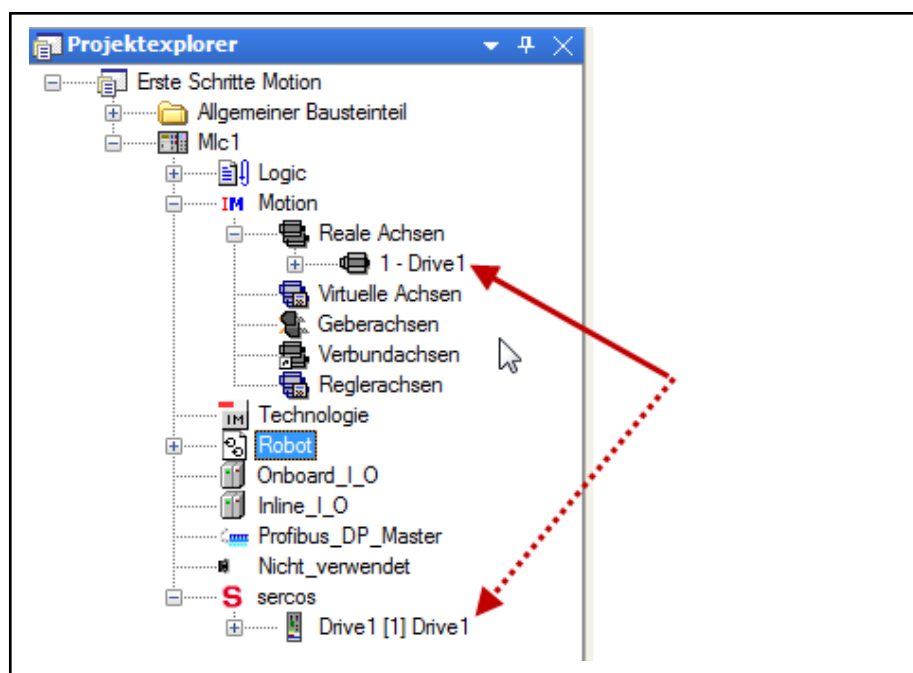


Abb. 3-13: Antrieb / Reale Achse "Drive1" ist dem Projekt hinzugefügt

3.3.7 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 gescannt werden müssen, ist eine virtuelle Achse nur zu deklarieren. Im Projektnavigator ist auf dem Ordner **Motion** ► **Virtuelle Achse** mit der rechten Maustaste der Kontextmenü-Punkt **Hinzufügen** auszuwählen.

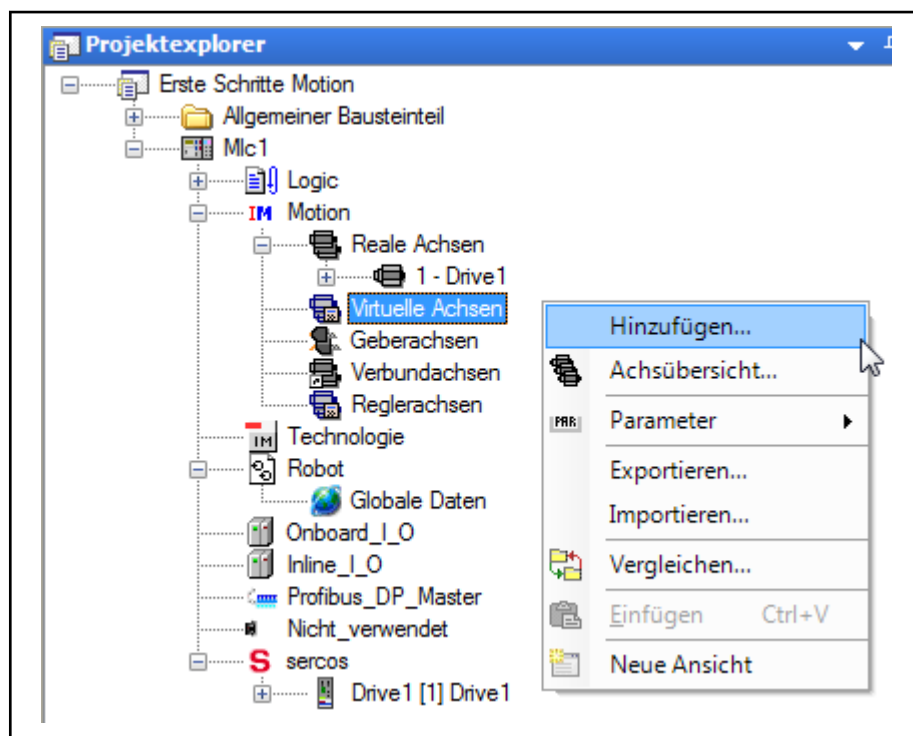


Abb. 3-14: Vereinbaren einer virtuellen Achse

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

Abb. 3-15: Dialog "Allgemeine Einstellungen" für eine virtuelle Achse

Abb. 3-16: Dialog "Konfiguration" für eine virtuelle Achse

Die Achsnummer "2" und Achsname "vAxis1" sind je nach Adressierungsart zu verwenden. Klicken Sie auf **Fertigstellen**.

3.3.8 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 ist das Gerät (Mlc1) zu markieren und das Icon der Symbolleiste "Projekt" oder der Menüpunkt **Projekt ► Gerät Online schalten** zu betätigen:

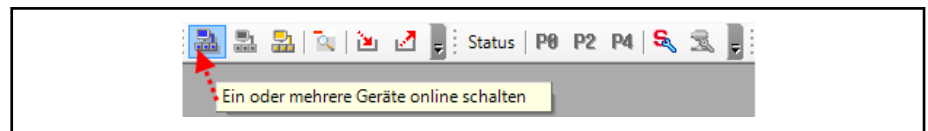


Abb. 3-17: Gerät online schalten

Zwischen der IndraWorks-Bedienoberfläche und der aktiven Steuerung läuft jetzt der Datenaustausch.

(1): Die IndraMotion MLC ist **teilsynchronisiert**, die IndraMotion MLC erwartet den Download des Motion-Teilprojektes.

(2): Der Sercos Feldbus läuft in Phase P2 (Parametriermodus), sichtbar am Sercos Knoten und am Icon der Symbolleiste "Sercos"

Erste Schritte – Motion

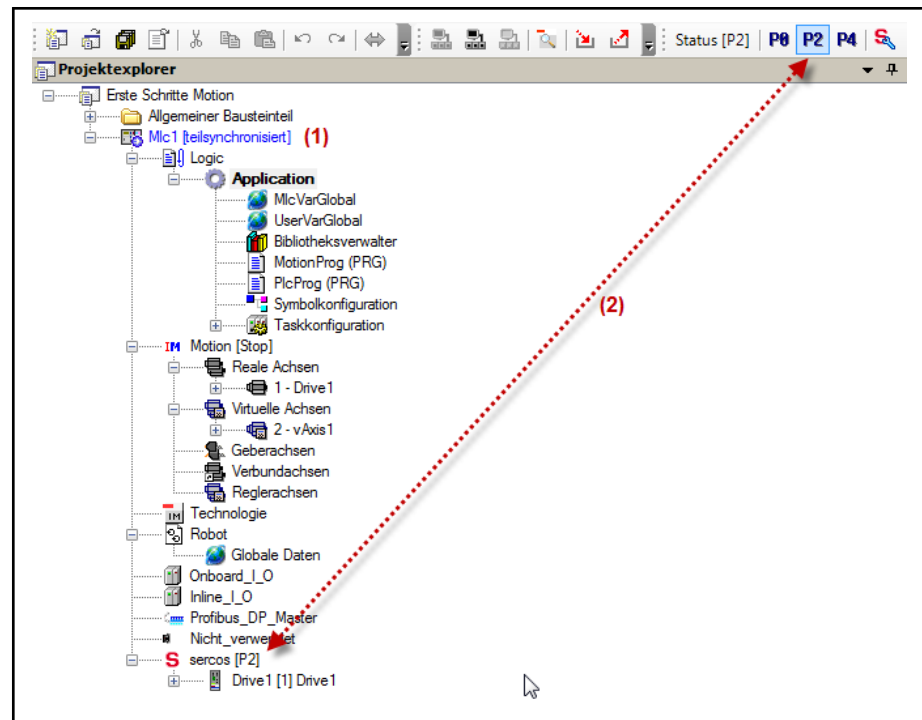


Abb. 3-18: Teilsynchronisierte IndraMotion MLC mit Sercos III Feldbus in P2

Um Synchronität der Steuerung mit dem Projekt zu erreichen, sind die Achskonfigurationsdaten abzugleichen (siehe nachfolgende Abbildung).

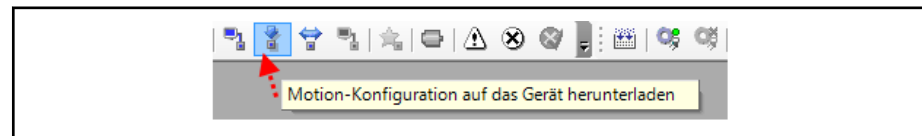


Abb. 3-19: Download der Motion-Konfiguration

Nachfolgender Dialog erlaubt es die Daten des Projektes in die Steuerung zu übertragen ("Verwende IndraWorks Achskonfiguration") oder die Daten der Steuerung ins Projekt zu übernehmen ("Verwende Achskonfiguration Steuerung"). Die markierten Achsen werden durch Drücken von **OK** übertragen.

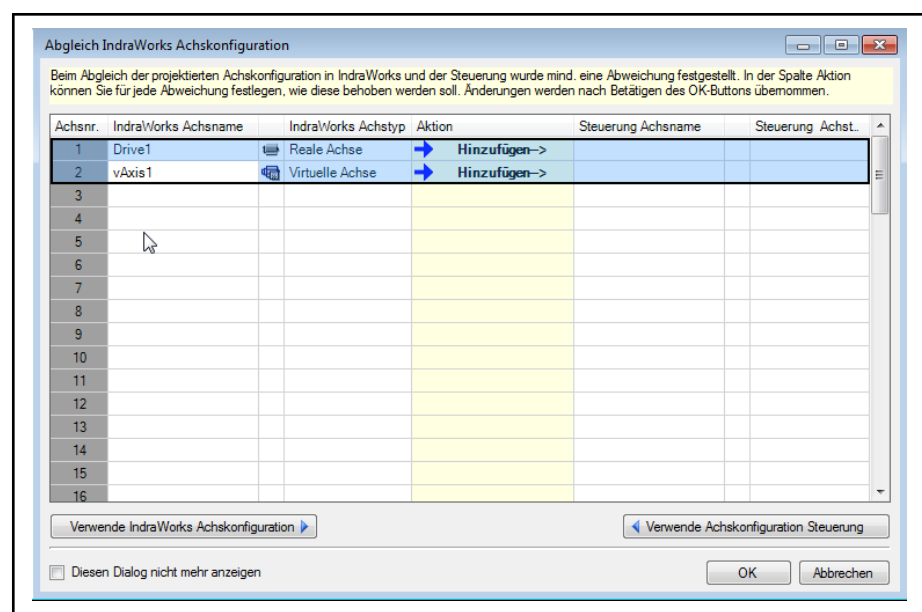


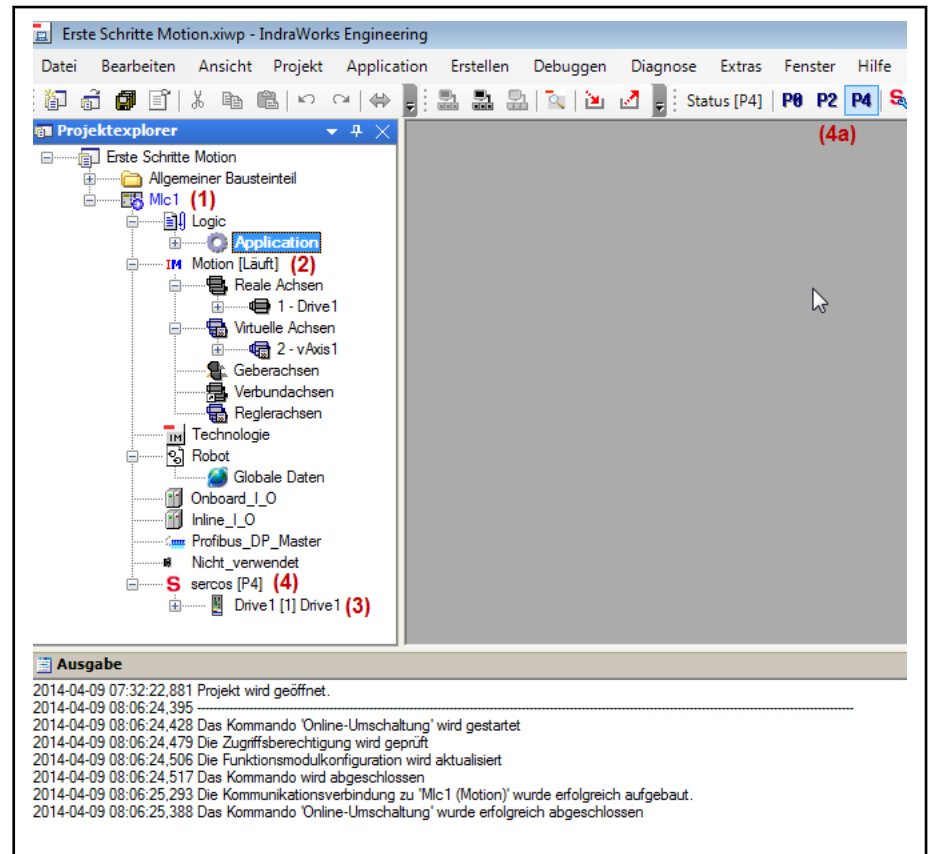
Abb. 3-20: Dialog "Abgleich der IndraWorks-Achskonfiguration"

Erste Schritte – Motion

Die Steuerung läuft nun synchronisiert.



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



- (1) aktive Steuerung (online)
- (2) Motionkomponente läuft fehlerfrei
- (3) aktiver Antrieb (online)
- (4, 4a) Kommunikation im Modus "P4", Betriebsmodus

Abb. 3-21: Erfolgreicher Abschluss des Downloads (Ausgabe)



Die "Online"-Information 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).

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

Erste Schritte – Motion



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. Die Basisparameter können über das Kontextmenü **Parameter ▶ Basisparameter laden...** der Achse (Drive1) nachgeladen werden.

Befindet sich die Steuerung im "Parametriermodus (P2)" muss die Steuerung in den "Betriebsmodus (BB)" geschaltet werden (Kontextmenü Sercos Knoten **Sercos Zustand ▶ Phase P4** oder Symbolleiste Sercos, Icon **P4**).



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

Der Fehler kann über <ESC> am Display des Antriebs oder der Steuerung quittiert oder besser über **Mlc1 ▶ Motion ▶ Reale Achse ▶ Drive1 ▶ Motor, Bremse, Messsysteme ▶ Motorgeber ▶ Maßbezug Motorgeber** "Absolutmaß setzen" beseitigt werden.

Danach Schaltfläche "Fehler löschen"  drücken.

Der Antrieb ist nun im Status "Ab" (Antrieb bereit).

Die Steuerung und Motion-Komponente ist in "BB" (betriebsbereit), die SPS-Komponente in "STOP" (SPS gestoppt) oder "RDY" (bereit).

3.4 Programmieren des Bewegungsablaufes in Funktionsbausteinsprache (FUP)

3.4.1 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 "Mlc1", der realen Achse "Drive1" und der virtuellen Achse "vAxis1" wurden unterhalb des Ordners "Logic" erste Aktivitäten zur Programmierung des Bewegungsablaufes begonnen:

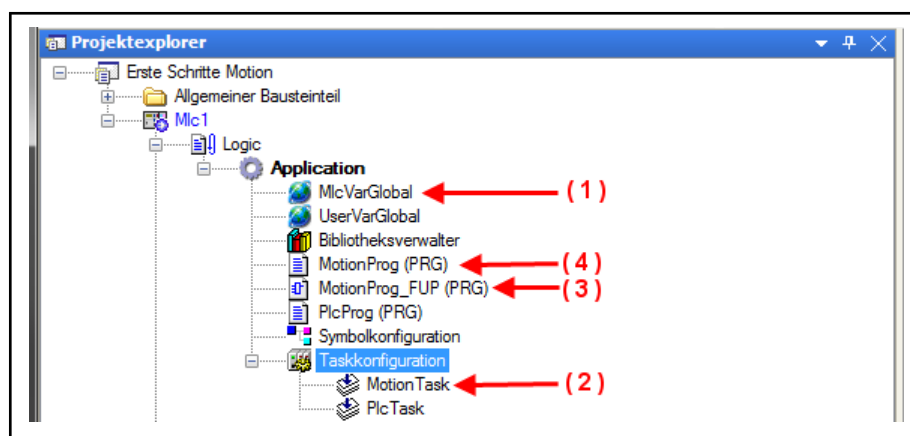


Abb. 3-22: Projektexplorer nach Expandieren des Knotens "Logic"

Angelegt wurden die Programme:

- **MotionProg (4)** (Vorwahl ST, Strukturierter Text)
- **PlcProg** (Vorwahl ST, Strukturierter Text), wird nicht verwendet

Das dritte Programm **MotionProg_FUP (3)**, 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, bzw. die Programme, die unter dieser Task laufen, müssen komplett innerhalb des fest vorgegebenen Teils 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 "Drive1" und "vAxis1"

Programm:

```
(* Automated generated code by MLC. *)
(* Please don't edit, the code will be overwritten...*)
VAR_GLOBAL CONSTANT
(*virtual axis "vAxis1" with axis number 2*)
vAxis1: AXIS_REF := (CntlNo:=LOCAL_CNTRL,AxisNo:=AXIS_2);
(*real axis "Drive1" with axis number 1*)
Drive1: AXIS_REF := (CntlNo:=LOCAL_CNTRL,AxisNo:=AXIS_1);
(*...*)
END_VAR
```



Für das Projekt "Ersten Schritte Motion" wird nur die MotionTask benötigt.

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

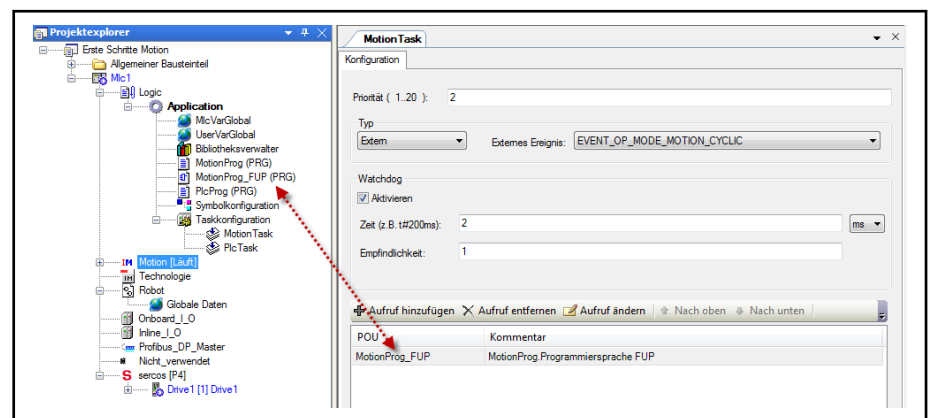


Abb. 3-23: Aufruf löschen: MotionProg; Aufruf hinzufügen: MotionProg_FUP

3.4.2 Programmieren der virtuellen Achse "vAxis1", 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 [Kap. 3.5.2 "Programmieren der virtuellen Achse vAxis1, Programmiersprache ST"](#) auf Seite 119.

Entsprechend der [Kap. 3.1 "Ziel des Projekts Erste Schritte – Motion"](#) auf [Seite 79](#), werden Instanzen der PLCopen-Bausteine "MC_MoveVelocity" und "MC_Stop" unter Nutzung von "vAxis1" aufgerufen.



Eine Instanz des Funktionsbausteins "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).

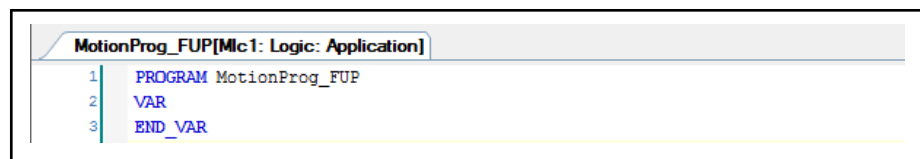


Abb. 3-24: Neues Programm in FUP (Funktionsbausteinsprache)

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

Erste Schritte – Motion

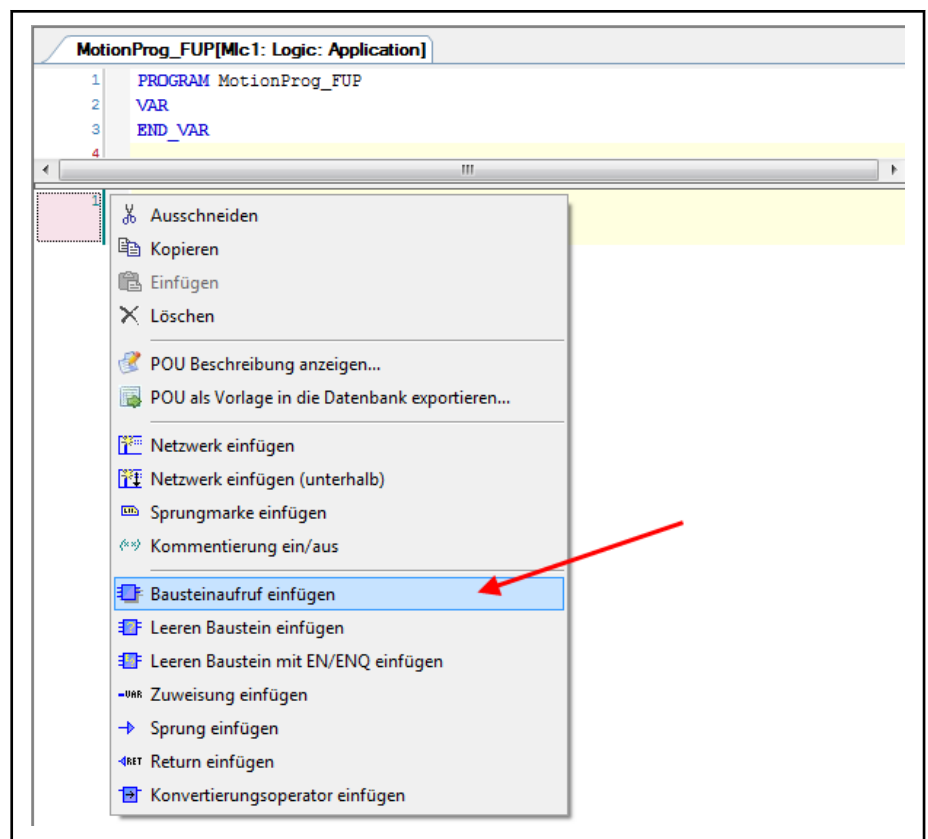


Abb. 3-25: Generieren eines neuen Bausteins

Wählen Sie dazu im geöffneten Auswahlfenster

Funktionsbausteine ▶ ML_PLCopen ▶ POU ▶ Motion ▶ MC_MoveVelocity
 aus.

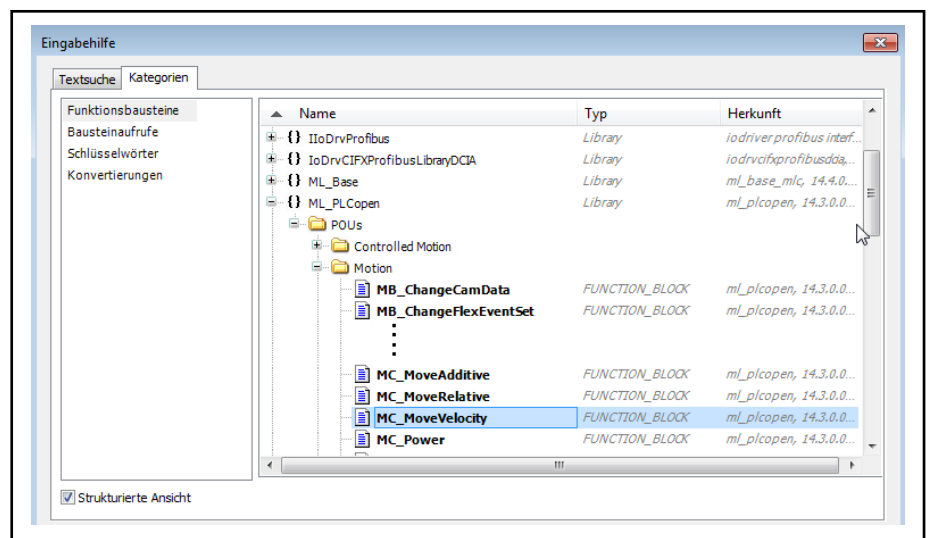


Abb. 3-26: Auswahlfenster für Funktionsbausteintypen

Bestätigen Sie das Einfügen des Bausteins "MC_MoveVelocity" mit **OK** und ergänzen Sie im Programm "MotionProg_FUP" den vorgeschlagenen Instanznamen "MC_MoveVelocity_0" oberhalb des Bausteins durch "MC_MoveVelocity_vAxis1" und schließen Sie die Eingabe mit <Enter> ab. Es öffnet sich das Fenster "Variablendeklaration", welches Sie sofort mit **OK** quittieren können:

Erste Schritte – Motion

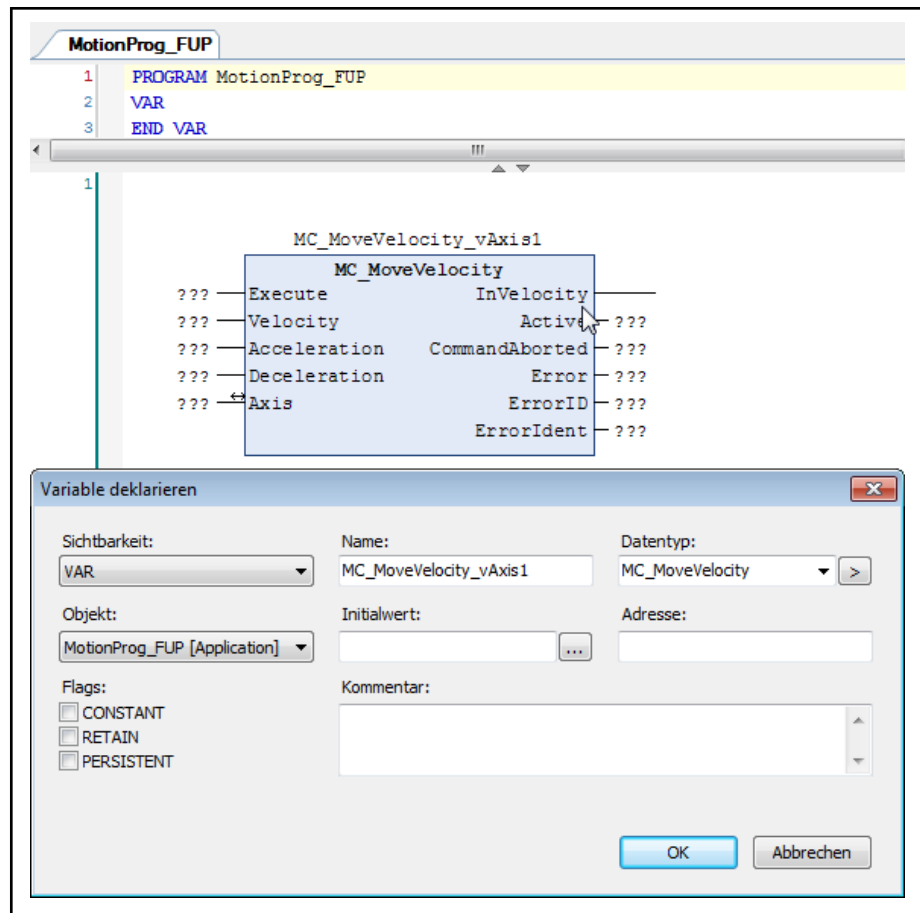


Abb. 3-27: Deklaration des Instanznamens eines Bausteines, Typ "MC_MoveVelocity" mit Hilfe des Fensters "Variablen Deklaration"

Damit haben Sie die Instanz "MC_MoveVelocity_vAxis1" 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_MoveVel_vAxis1", Typ "BOOL":

- Markieren Sie die 3 Fragezeichen vor dem Eingang "Execute"
- Geben Sie den Namen "ex_MoveVel_vAxis1" ein, und schließen Sie mit <Enter> ab
- Das sich öffnende Fenster zur "Variablen Deklaration" 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 "Vcom_vAxis1" ein und bestätigen Sie mit <Enter>
- Geben Sie als Typ "REAL" und als Initialwert 200 (U/min) ein und bestätigen Sie mit <Enter>
- Für die Eingänge "Acceleration" und "Deceleration" tragen Sie bitte 20 (rad/s²) ein



Die Werte 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:

- Klicken Sie **???** an und wählen die nebenstehende Eingabehilfe :
 Wählen Sie **Variablen** ► **Application** ► **MlcVarGlobal** ► **vAxis1**, mit **Namensraum-Präfix** einfügen.

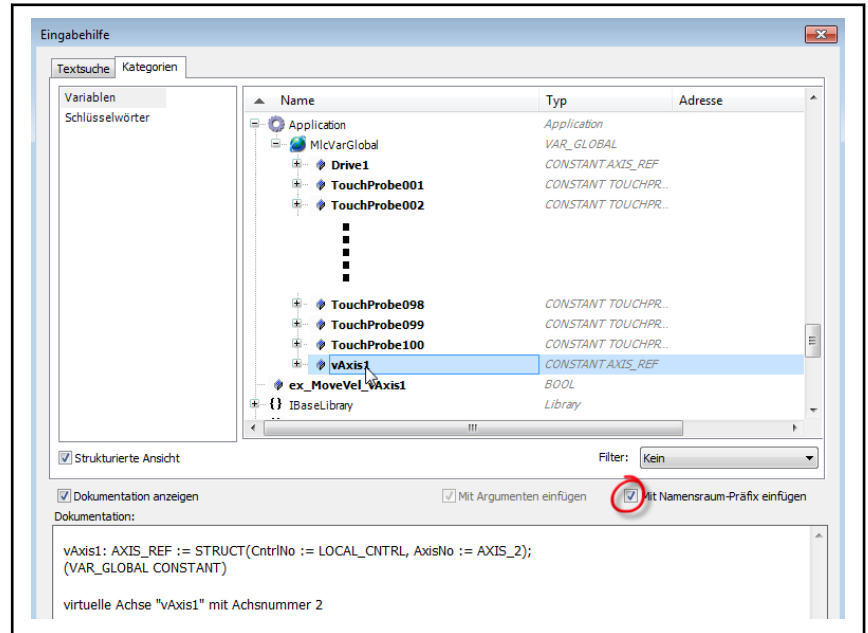


Abb. 3-28: Auswahl der Achsinstanz bei Beschaltung eines Funktionsbausteines vom Typ "MC_MoveVelocity"

- Bestätigen Sie mit **OK**.
- 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.vAxis1`

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:

Erste Schritte – Motion

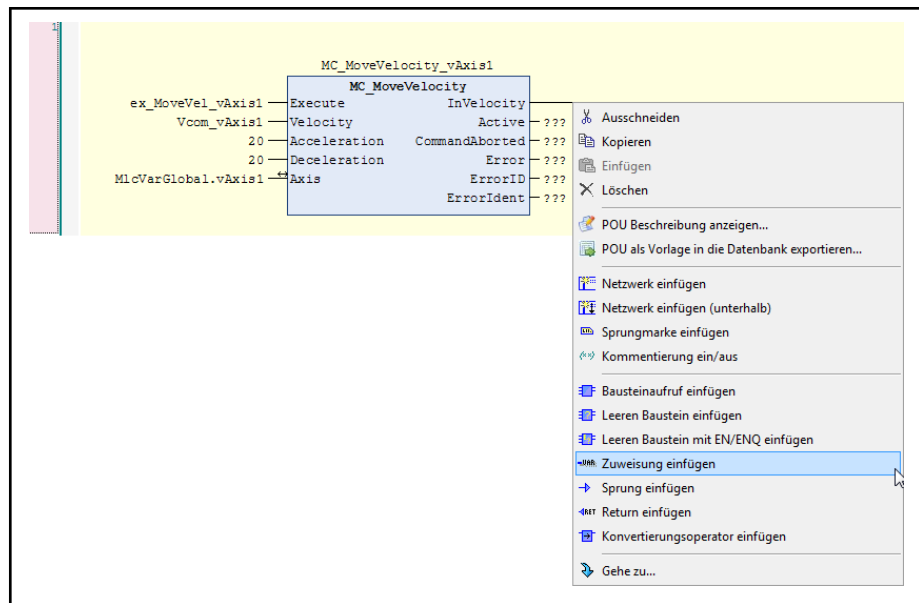


Abb. 3-29: Zuweisung am Ausgang eines Funktionsbausteines

In bekannter Art und Weise beschalten Sie den Ausgang "InVelocity" des Bausteines mit der Variablen "in_MoveVel_vAxis1", Typ "BOOL":

- Markieren Sie die 3 Fragezeichen **???** am Ausgang "InVelocity"
- Geben Sie den Namen "in_MoveVel_vAxis1" ein und bestätigen Sie mit <Enter>
- Das sich öffnende Fenster zur "Variablendeklaration" können Sie sofort mit <Enter> bestätigen
- Beschalten Sie den "Active"-Ausgang mit "ac_MoveVel_vAxis1"
- Beschalten Sie den "CommandAborted"-Ausgang mit "ca_MoveVel_vAxis1"
- Beschalten Sie den "Error"-Ausgang mit "er_MoveVel_vAxis1"
- Beschalten Sie den "ErrorID"-Ausgang mit "ID_MoveVel_vAxis1"
- Beschalten Sie den "ErrorIdent"-Ausgang mit "Ident_MoveVel_vAxis1"

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

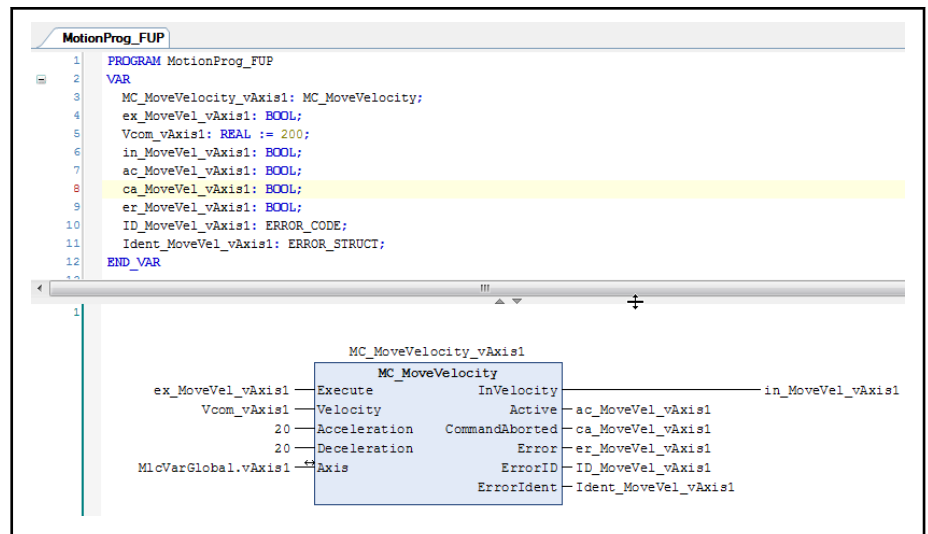


Abb. 3-30: PLCopen-FB Typ "MC_MoveVelocity" ist eingefügt und beschaltet



Immer wenn Sie einen Baustein abgeschlossen haben, sollten Sie eine Testübersetzung (<Umschalt>+<F11>) durchführen. 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 "MC_Stop_vAxis1" deklarieren

Neues Netzwerk anfügen durch Klick mit rechter Maustaste in Netzwerk 1

Netzwerk einfügen (unterhalb)

"MC_Stop" einfügen und eine Instanz "MC_Stop_vAxis1" deklarieren

Eingänge beschalten und wenn nötig Eingangsvariablen deklarieren:

"Execute" → "ex_Stop_vAxis1" vom Typ "BOOL"

"Deceleration" → 50

"Axis" → "MlcVarGlobal.vAxis1" vom Typ "AXIS_REF")

Ausgänge:

"Done" → "do_Stop_vAxis1" vom Typ "BOOL"

"Active" → "ac_Stop_vAxis1" vom Typ "BOOL"

"CommandAborted" → "ca_Stop_vAxis1" vom Typ "BOOL"

"Error" → "er_Stop_vAxis1" vom Typ "BOOL"

"ErrorID" → "ID_Stop_vAxis1" vom Typ "ERROR_CODE"

"ErrorIdent" → "Ident_Stop_vAxis1" vom Typ "ERROR_STRUCT"

Istgeschwindigkeit "Vact_vAxis1"

Als Letztes soll ein Netzwerk, in dem die aktuelle Geschwindigkeit der Achse "vAxis1" "Vact_vAxis1" (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 IndraMotion MLC steuern kann, sind in dem Feld

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

Erste Schritte – Motion

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 ("Drive1" hat Achsnummer 1, da das Scannen von "Drive1" vor der Deklaration von "vAxis1" erfolgte).

- Neues Netzwerk anfügen durch Klick mit rechter Maustaste in Netzwerk 2
- **Netzwerk einfügen (unterhalb)**
- **Zuweisung einfügen**
- Auf den rechten ??? ist "Vact_vAxis1" einzufügen und mit <Enter> zu bestätigen, Typ "REAL" im Deklarationsfenster auswählen und mit <Enter> bestätigen
- Auf die linken ??? gehen und danach  anklicken und unter Nutzung der Eingabehilfe zu **Variablen** ► **ML_BASE** ► **Global Variables List** ► **GVL_Base** ► **AxisData[]** navigieren:

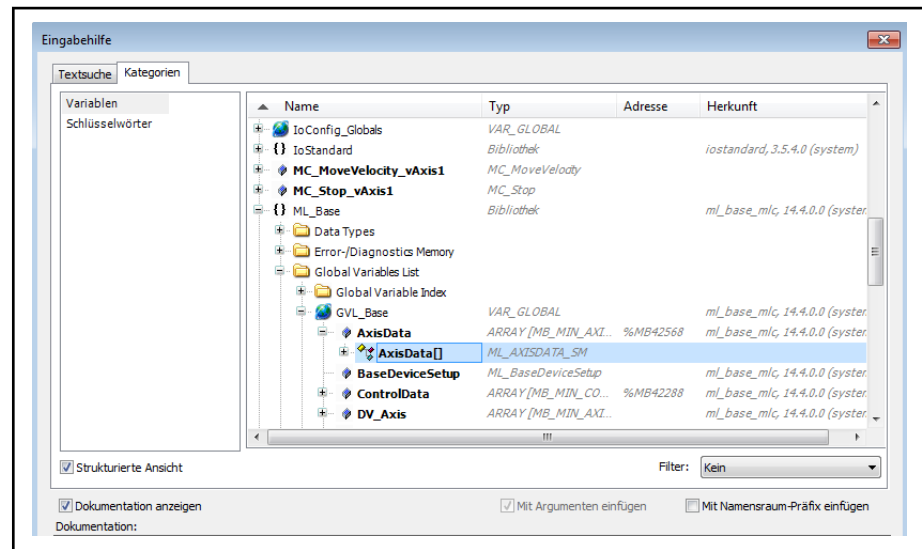


Abb. 3-31: Geschwindigkeitswert "vAxis1" aus der Globalen Variablenliste der Bibliothek "ML_Base.library"

Die weitere Eingabe wird durch die Intellisense-Funktion (automatische Vervollständigung) unterstützt; d. h., dass nach der Eingabe des "." bei "AxisData[2]." alle verfügbaren Elemente aufgerollt werden und zur Auswahl stehen.

"rActualVelocity_i" mit <Enter> bestätigen.

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

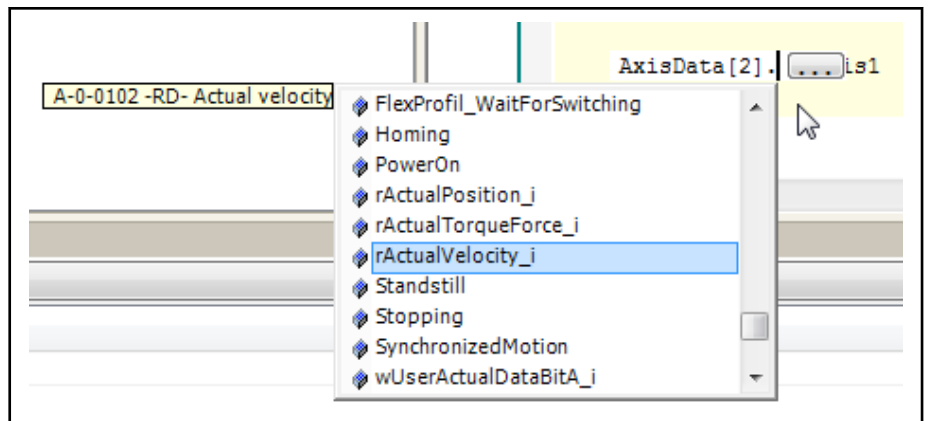


Abb. 3-32: Auswahl der Variablen aktuelle Geschwindigkeit

Damit ist die Implementierung des Bewegungsablaufes entsprechend der Aufgabenstellung abgeschlossen:

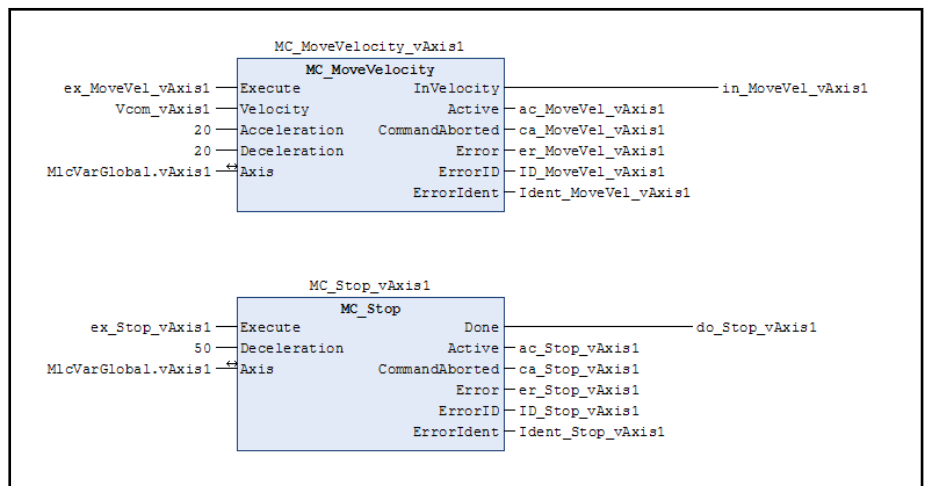


Abb. 3-33: Implementation der virtuellen Achse "vAxis1"



AxisData[].rActualVelocity_i; und
GVL_Base.AxisData[].rActualVelocity_i;
werden vom Compiler als gleichwertig verwendet.

3.4.3 Download des SPS-Teilprojekts virtuelle Achse "vAxis1" (FUP)

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

Die Übersetzung erreichen Sie im IndraWorks-Menüpunkt

- **Erstellen ► Übersetzen**, Übersetzen des hinzu gekommenen Anteils der Applikation
- **Erstellen ► Code erzeugen**, der Befehl ermöglicht eine Code-Generierung zur Überprüfung von Übersetzungsfehlern

Erste Schritte – Motion

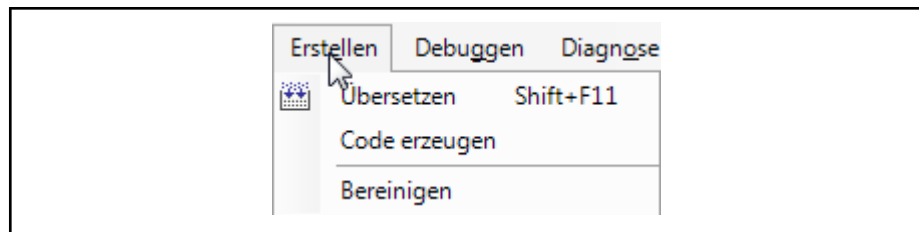


Abb. 3-34: Applikation übersetzen



Die Menüpunkte **Erstellen ▶ Neu übersetzen** und **Erstellen ▶ Alles bereinigen** aus der 13VRS sind standardmäßig nicht mehr vorhanden. Diese Befehle können aber über **Extras ▶ Anpassen ▶ Befehle ▶ Erstellen** wieder hinzugefügt werden.

Zum Übertragen der kompilierten Daten in die Zielsteuerung wählen Sie in der IndraWorks-Menüleiste **Debuggen ▶ Einloggen** (siehe nachfolgende Abbildung).

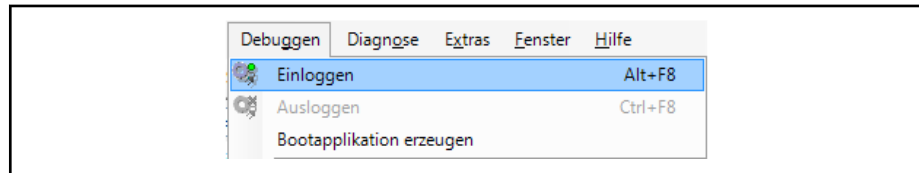


Abb. 3-35: Übertragen der kompilierten Daten in die Steuerung

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.

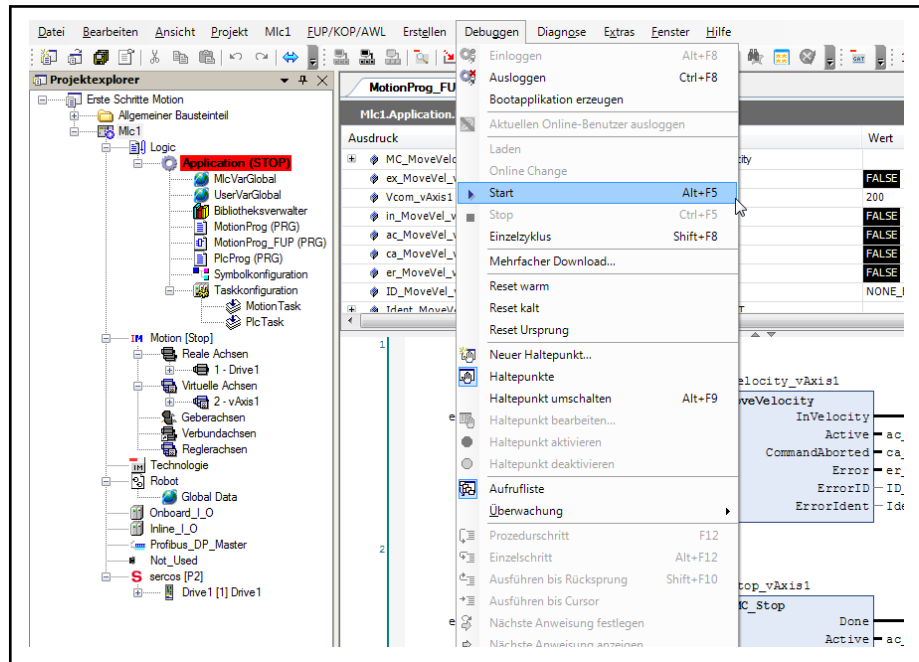


Abb. 3-36: Starten der Programmbearbeitung in der Steuerung (1)

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

Erste Schritte – Motion

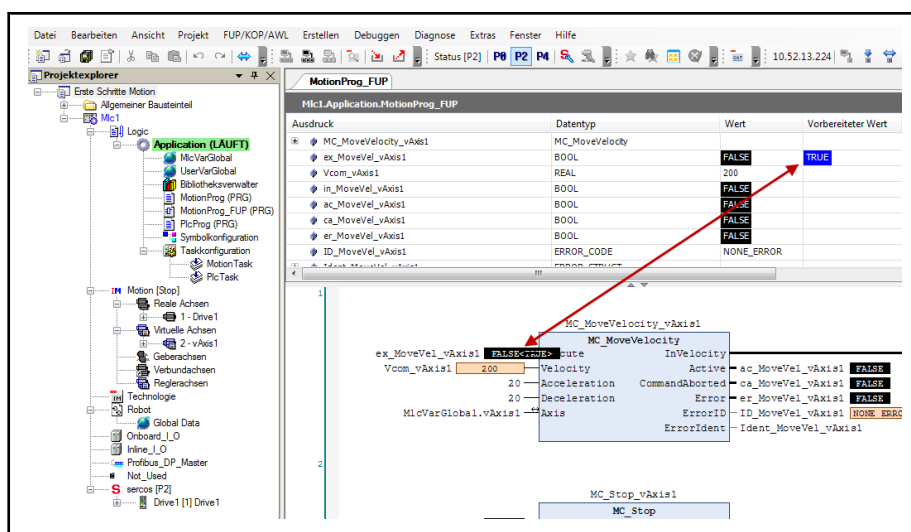


Abb. 3-37: Starten der Programmbearbeitung in der Steuerung (2)

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

"Werte schreiben" der Eingangsvariablen

Um die Bearbeitung des "MC_MoveVelocity_vAxis1" 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_MC_MoveVel_vAxis1". Jetzt ist hinter "ex_MoveVel_vAxis1" ein markiertes "FALSE" und in der Deklarationszeile zunächst ein "TRUE", nach dem nächsten Klick "FALSE" und dann ein leerer Platz (deaktiviert) zu sehen.

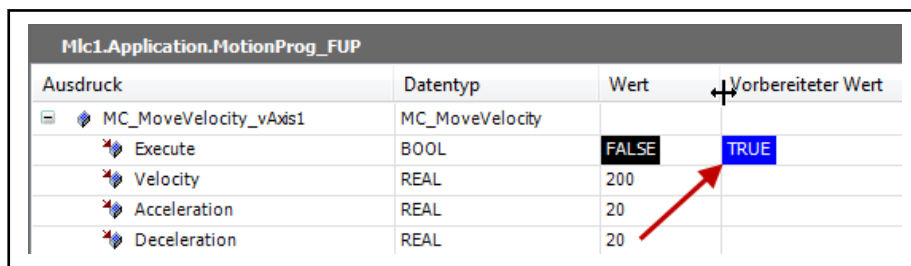


Abb. 3-38: "Werte schreiben" im Deklarationsteil des Programms

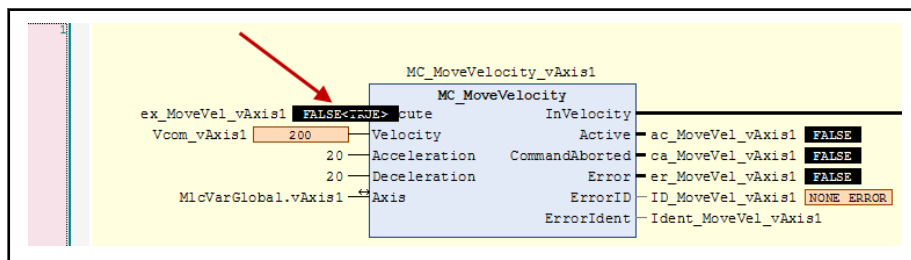


Abb. 3-39: "Werte schreiben" an der Schnittstelle des Funktionsbausteines

Über **Debuggen** ► **Werte schreiben** oder <Strg>+<F7> wird "ex_MC_MoveVel_vAxis1" 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

Die dadurch entstehende positive Flanke setzt die Ausgänge "InVelocity" und "Active" vom Funktionsbaustein "MC_MoveVelocity_vAxis1" auf "TRUE" (1).

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

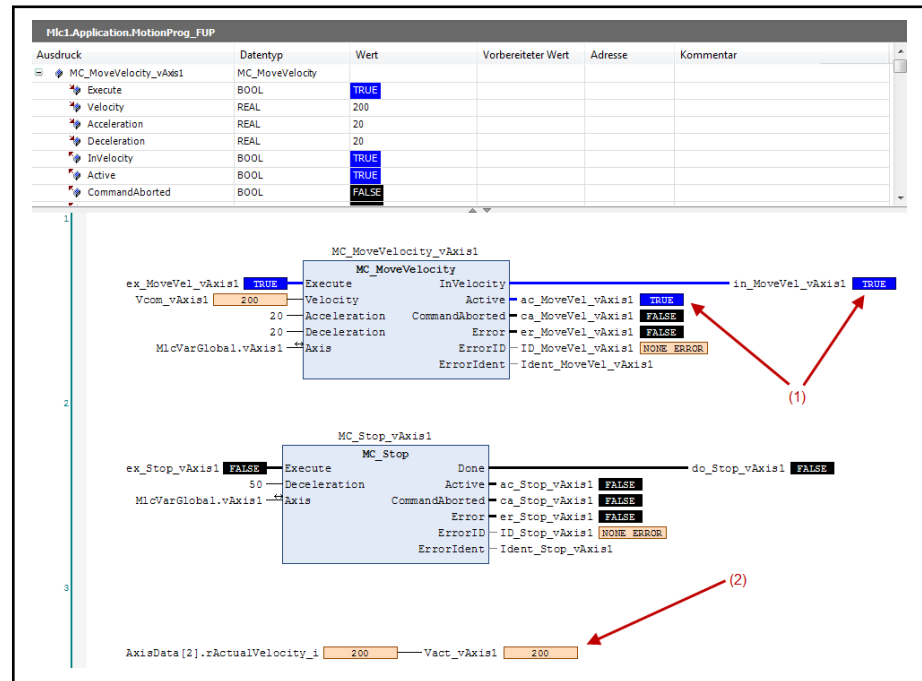


Abb. 3-40: Nach "Werte schreiben" am Eingang "Execute"

Schalten Sie "ex_MoveVel_vAxis1" wieder auf "FALSE". Die Achse läuft weiter mit 200 U/min.

Sie können durch Setzen der Variable "ex_Stop_vAxis1" am Eingang "Execute" des Funktionsbausteines "MC_Stop_vAxis1" vom Typ "MC_Stop" die Bewegung der Achse stoppen.

Stoppen Sie die Applikation und loggen sich aus.

3.4.4 Aufzeichnung der Bewegung von Achse "vAxis1" (Trace), FUP

IndraLogic liefert zur Aufzeichnung von Bewegungsvorgängen ein Trace-Werkzeug. Mit Hilfe dieses Werkzeugs soll Folgendes aufgezeichnet werden.

- die Achsgeschwindigkeit "Vact_vAxis1"
- für "MC_MoveVelocity_vAxis1" (MC_MoveVelocity)
 - "ex_MoveVel_vAxis1" (Execute)
 - "in_MoveVel_vAxis1" (InVelocity)
 - "ca_MoveVel_vAxis1" (CommandAborted)

und

- für "MC_Stop_vAxis1" (MC_Stop)
 - "ex_Stop_vAxis1" (Execute)

Erste Schritte – Motion

Dazu muss zunächst die Traceaufzeichnung konfiguriert werden.

Fügen Sie über das Kontextmenü von "Application" **Hinzufügen ▶ Trace...** das Objekt "Trace_vAxis1" ein. Öffnen Sie das Objekt mit einem Doppelklick.

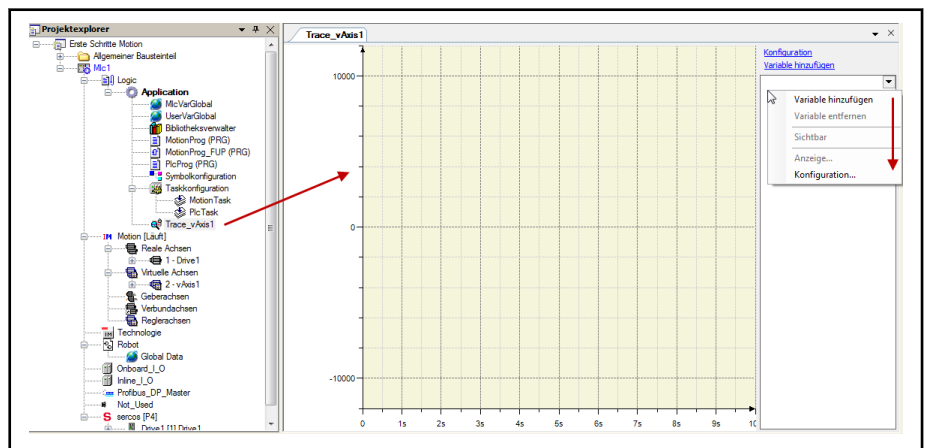


Abb. 3-41: Dialog: Neues Trace-Objekt "Trace_vAxis1" mit Kontextmenü

Über das Kontextmenü auf der rechten Blattseite können für das Trace-Objekt folgende Einstellungen vorgenommen werden:

- Konfiguration
- Anzeigeeinstellungen
- Neue Variable hinzufügen

Konfiguration... Das neue Trace-Objekt ist zunächst zu konfigurieren:

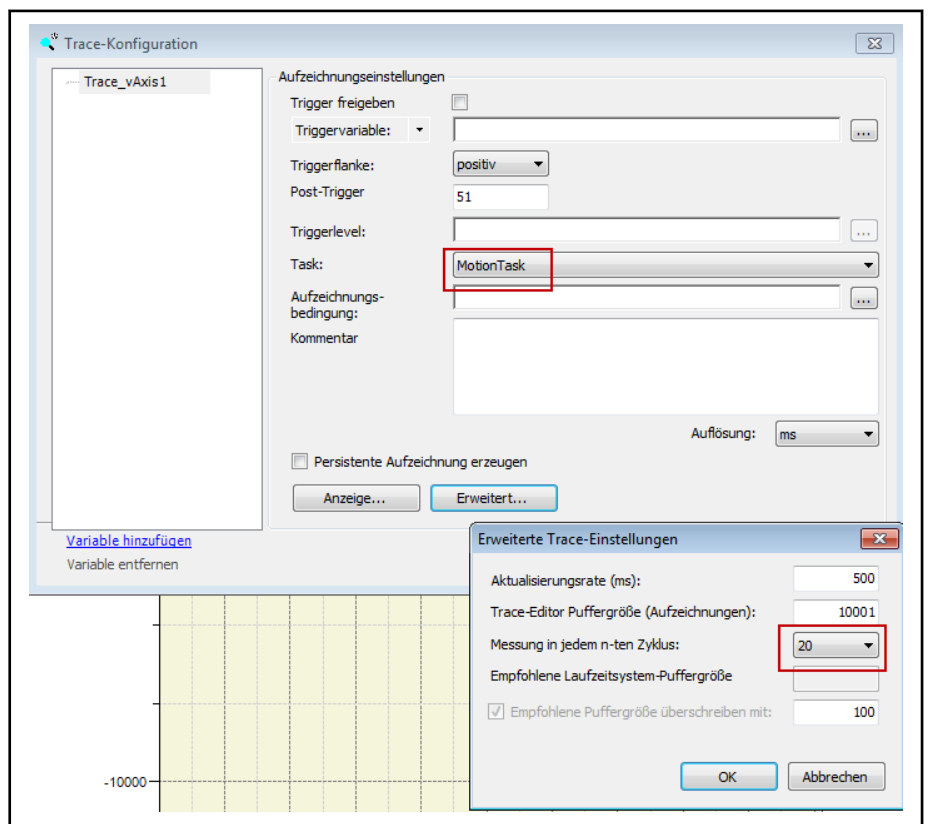


Abb. 3-42: Dialog: Trace-Konfiguration mit erweiterter Trace-Einstellung

Erste Schritte – Motion

Notwendige Einstellungen sind

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

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

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.

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

Empfohlene Laufzeitsystem-Puffergröße: Größe des Ringpuffers in Byte, 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.

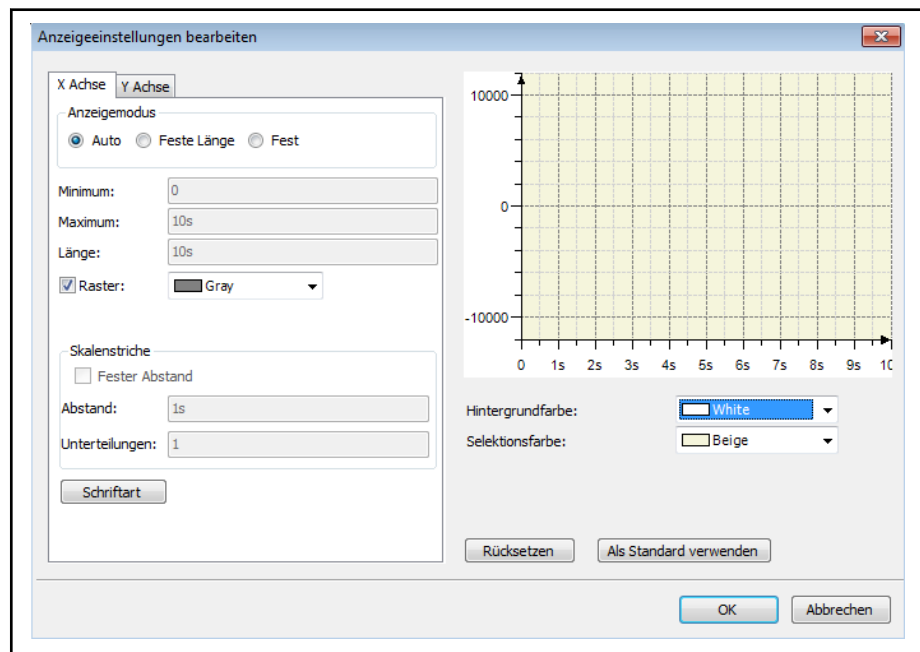


Abb. 3-43: Dialog: Anzeigeeinstellungen

Anzuwählen ist **Auto**, hierbei wird ein Fenster der **Länge** 10 s über den Kurvenzug gelegt. Die **Skaleneinteilung** wird dabei im 1s Raster mit einer **Skalenunterteilung** von 1s vorgenommen.

Auswahl der Neuen Variablen

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

Erste Schritte – Motion

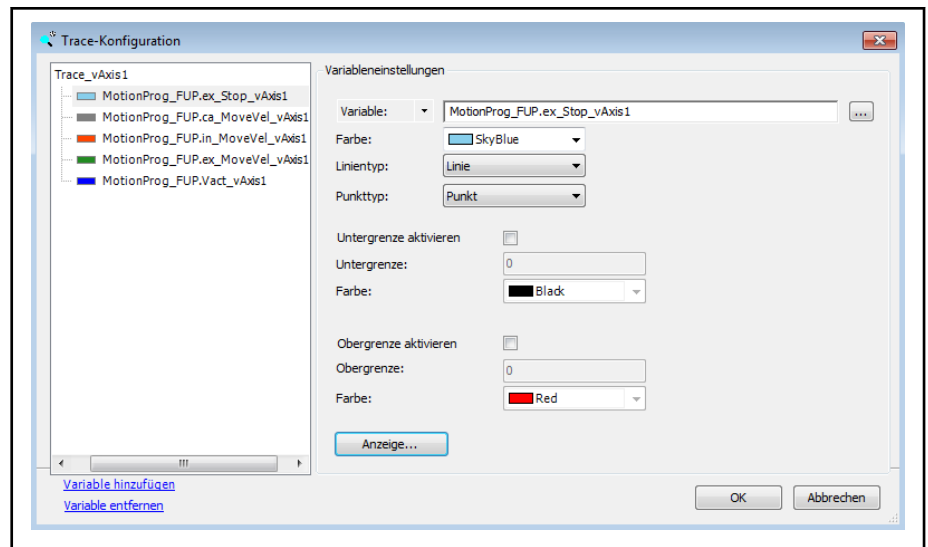


Abb. 3-44: Dialog: Anlegen von neuen Variablen und Festlegung des Erscheinungsbildes

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

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.

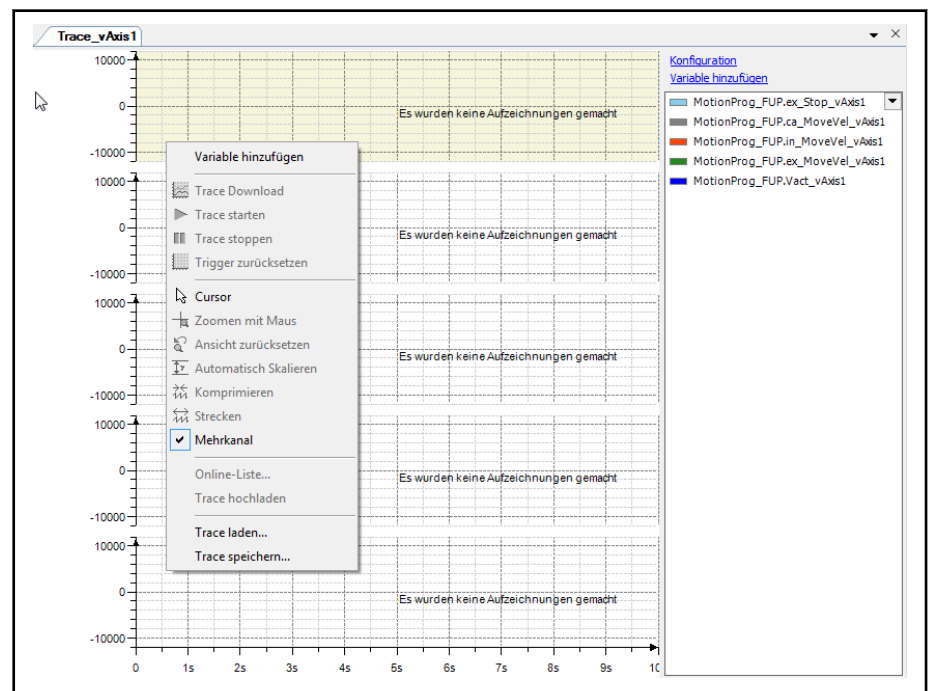


Abb. 3-45: Kontextmenü zum Steuern der Trace-Aufzeichnung

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

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

Erste Schritte – Motion

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 Variablenwertekurve(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.

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

Durchführung des Trace:

1. Einloggen auf der Steuerung.
Herstellen der Verbindung zur Steuerung.
2. Starten der Steuerung.
Steuerung in BB/RUN.
3. Trace Download.
Bei laufender Application werden über das Trace-Kontextmenü "Trace Download" die erforderlichen Daten in die Steuerung übertragen.
Die Trace-Aufzeichnung läuft danach automatisch an.
4. Werte schreiben.
Folgende Werte werden analog zu ["Werte schreiben der Eingangsvariablen" auf Seite 105](#) gesetzt:
 - "ex_MoveVel_vAxis1" auf "TRUE".
 - "ex_Stop_vAxis1" auf "TRUE".
5. Beenden der Aufzeichnung.
Stoppen des Trace durch den Trace-Kontextmenüpunkt "Start/Stop Trace".
6. Auswertung der Trace-Aufzeichnung.
Einstellen der Fensterdaten.

Tastenkombinationen	Aktion im Trace-Diagramm	Entsprechende Steuerung mit Maus
<Alt>+<←/→>-Pfeiltasten	horizontales Scrollen der X-Achse (Zeitachse) aller Diagramme	Aktivieren von Scrollen mit Maus, Klicken in das Diagramm und scrollen der X-Achse mit der Maus während die linke Maustaste gedrückt wird
<↑/↓>-Pfeiltasten, mit oder ohne gedrückte <Alt>-Taste Selektieren eines Diagramms mit <Tab>	horizontales Scrollen des selektierten Diagramms	Aktivieren von Scrollen mit Maus, Klicken in das Diagramm und scrollen der X-Achse mit der Maus während die linke Maustaste und <Strg> gedrückt wird
<↑/↓>-Pfeiltasten, mit oder ohne gedrückte <Alt>-Taste Selektieren eines Diagramms mit <Tab> Zusätzlich <Strg> drücken	Der Graph wird in größeren Schritten verschoben	
<+/->-Tasten	horizontales Zoomen aller Diagramme entlang der X-Achse (die Zeitachse wird für alle Diagramme gestreckt/komprimiert)	- Auswahl "Strecken" oder "Komprimieren":  oder  - bzw. Verwenden des Mausekkrads
<Strg>+<+/->-Tasten Selektieren eines Diagramms mit <Tab>	vertikales Zoomen aller Diagramme entlang der Y-Achse (die der Y-Achse des selektierten Graphen wird gestreckt/komprimiert)	

Tab. 3-1: Tastenkombinationen Trace-Editor mit aktiviertem Mehrfachkanal

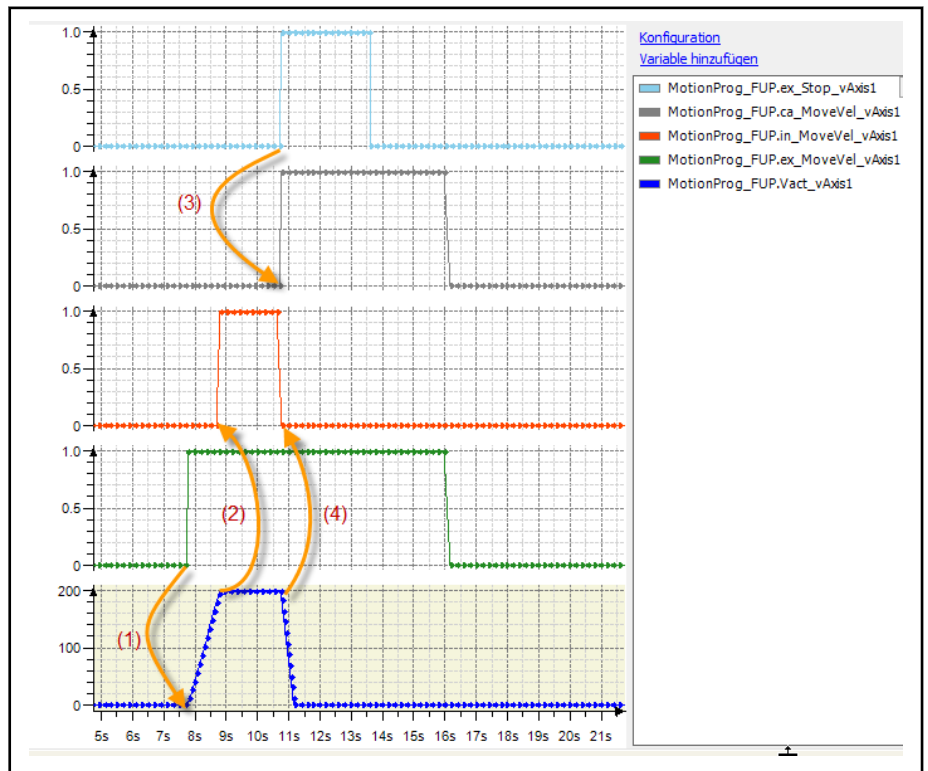


Abb. 3-46: Traceaufzeichnung für die virtuelle Achse "vAxis1"

Erste Schritte – Motion

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).

3.4.5 Programmieren der realen Achse Drive1 (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 [Kap. 3.5.2 "Programmieren der virtuellen Achse vAxis1, Programmiersprache ST"](#) auf Seite 119.

MC_Power

Neues Netzwerk anfügen durch Klick mit rechter Maustaste in Netzwerk 3

Netzwerk einfügen (unterhalb)

"MC_Power" einfügen und eine Instanz "MC_Power_Drive1" deklarieren

Eingänge beschalten und wenn nötig Eingangsvariablen deklarieren:

"Enable" → "en_MC_Power_Drive1" vom Typ "BOOL"

"Axis" → "MlcVarGlobal.Drive1" vom Typ "AXIS_REF"

Ausgänge:

"Status" → "st_Power_Drive1" vom Typ "BOOL"

"Error" → "er_Power_Drive1" vom Typ "BOOL"

"ErrorID" → "ID_Power_Drive1" vom Typ "ERROR_CODE"

"ErrorIdent" → "Ident_Power_Drive1" vom Typ "ERROR_STRUCT"

MC_GearIn

Neues Netzwerk anfügen durch Klick mit rechter Maustaste in Netzwerk 4

Netzwerk einfügen (unterhalb)

"MC_GearIn" einfügen und Instanz "MC_GearIn_Drive1" deklarieren

Eingänge beschalten und wenn nötig Eingangsvariablen deklarieren:

"Execute" → "ex_GearIn_Drive1" vom Typ "BOOL"

"RatioNumerator" → 5

"RatioDenominator" → 10

"MasterFineadjust" → 20 (%)

"Master" → "MlcVarGlobal.vAxis1" vom Typ "AXIS_REF"

"Slave" → "MlcVarGlobal.Drive1" vom Typ "AXIS_REF"

Ausgänge:

"InGear" → "ig_GearIn_Drive1" vom Typ "BOOL"

"Active" → "ac_GearIn_Drive1" vom Typ "BOOL"

"CommandAborted" → "ca_GearIn_Drive1" vom Typ "BOOL"

"Error" → "er_GearIn_Drive1" vom Typ "BOOL"

"ErrorID" → "ID_GearIn_Drive1" vom Typ "ERROR_CODE"

"ErrorIdent" → "Ident_GearIn_Drive1" vom Typ "ERROR_STRUCT"

MC_GearOut

Neues Netzwerk anfügen durch Klick mit rechter Maustaste in Netzwerk 5

Netzwerk einfügen (unterhalb)

"MC_GearOut" einfügen und eine Instanz "MC_GearOut_Drive1" deklarieren

Eingänge beschalten und wenn nötig Eingangsvariablen deklarieren:

"Execute" → "ex_GearOut_Drive1" vom Typ "BOOL"

"Slave" → "MlcVarGlobal.Drive1" vom Typ "AXIS_REF"

Ausgänge:

"Done" → "do_GearOut_Drive1" vom Typ "BOOL"

"Active" → "ac_GearOut_Drive1" vom Typ "BOOL"

"Error" → "er_GearOut_Drive1" vom Typ "BOOL"

"ErrorID" → "ID_GearOut_Drive1" vom Typ "ERROR_CODE"

"ErrorIdent" → "Ident_GearOut_Drive1" vom Typ "ERROR_STRUCT"

MC_Stop

Neues Netzwerk anfügen durch Klick mit rechter Maustaste in Netzwerk 6

Netzwerk einfügen (unterhalb)

"MC_Stop" einfügen und eine Instanz "MC_Stop_Drive1" deklarieren

Eingänge beschalten und wenn nötig Eingangsvariablen deklarieren:

"Execute" → "ex_Stop_Drive1" vom Typ "BOOL"

"Deceleration" → 50 (rad/s²)

"Axis" → "MlcVarGlobal.Drive1" vom Typ "AXIS_REF"

Ausgänge:

"Done" → "do_Stop_Drive1" vom Typ "BOOL"

"Active" → "ac_Stop_Drive1" vom Typ "BOOL"

"CommandAborted" → "ca_Stop_Drive1" vom Typ "BOOL"

"Error" → "er_Stop_Drive1" vom Typ "BOOL"

"ErrorID" → "ID_Stop_Drive1" vom Typ "ERROR_CODE"

"ErrorIdent" → "Ident_Stop_Drive1" vom Typ "ERROR_STRUCT"

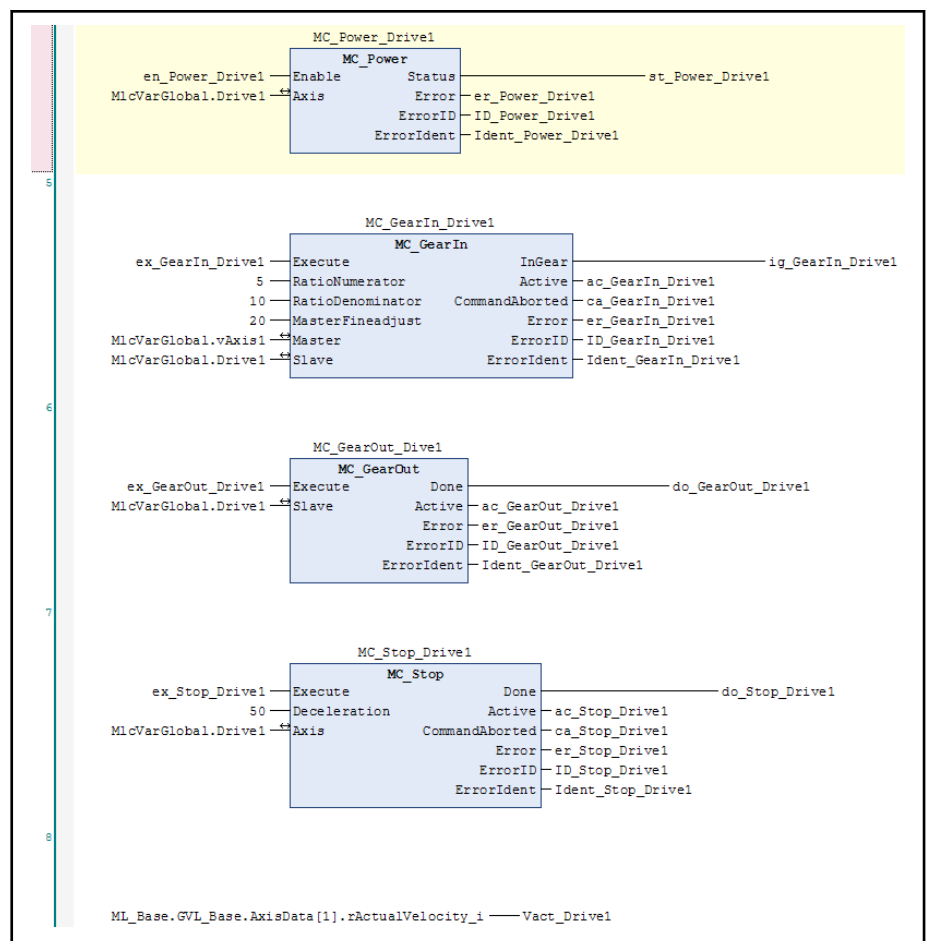


Abb. 3-48: Implementation der realen Achse



AxisData[].rActualVelocity_i; und
 GVL_Base.AxisData[].rActualVelocity_i;
 werden vom Compiler als gleichwertig verwendet.

3.4.6 Download des kompletten SPS-Projektes (FUP)

Der Ladevorgang verläuft analog zum Abschnitt [Kap. 3.4.3 "Download des SPS-Teilprojekts virtuelle Achse vAxis1 \(FUP\)"](#) auf Seite 103.

Das neue Projekt überschreibt dabei nach Rückfrage das alte Projekt.

3.4.7 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_Power_Drive1" (MC_Power) durch "Werte schreiben".

st_Power_Drive1= TRUE (MC_Power).

2. Start der virtuellen Achse.

Zuschalten des "ex_MoveVel_vAxis" (MC_MoveVelocity) durch "Werte schreiben".

Erste Schritte – Motion

Hochlauf bei Vact_vAxis1 = 200 U/min beendet.

3. Nachführen der realen Achse.

Zuschalten des "ex_GearIn_Drive1" (MC_GearIn) durch "Werte schreiben".

Das zu erwartenden Übersetzungsverhältnis des Getriebes ergibt sich zu:

$$\dot{U}_v = (\text{RatioNumerator} / \text{RatioDenominator}) \times (1 + \text{MasterFineadjust})$$

also bei Vact_vAxis1 = 200 U/min ergibt sich Vact_Drive1 = 120 U/min.



Der MasterFineadjust-Wert wird in "%" vorgegeben.

4. Befindet sich der MC_GearIn "InGear", so kann die Geschwindigkeit "Vcom_vAxis1" der virtuellen Achse durch "Werte schreiben" (Doppelklick auf die Variable, gewünschten Wert eintragen) verändert werden. Die reale Achse folgt der virtuellen.

5. Abschalten des "ex_GearIn_Drive1" (MC_GearIn) durch "Werte schreiben".

in_GearIn_Drive1= FALSE, ac_GearIn_Drive1= TRUE.

Die reale Achse folgt der virtuellen weiterhin.

6. Auflösen der Synchronität.

Zuschalten des "ex_GearOut_Drive1" (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 "Vcom_vAxis1".

Die veränderte Sollgeschwindigkeit wirkt sich nur noch auf die virtuelle Achse aus, "Vact_Drive1" bleibt unverändert.

8. Abschalten des "ex_GearOut_Drive1" durch "Werte schreiben".

do_GearOut_Drive1= FALSE.

9. Stoppen der virtuellen Achse.

Zuschalten des "ex_Stop_Drive1" (MC_Stop).

Die virtuelle Achse wird abgebremst auf Vact_vAxis1= 0 U/min.

10. Abschalten des "ex_Stop_Drive1" (MC_Stop).

11. Stoppen der realen Achse.

Zuschalten des "ex_Stop_Drive1" (MC_Stop).

Die reale Achse wird abgebremst auf Vact_Drive1= 0 U/min. Die Achse wird nach Erreichen des Stillstandes aktiv in dieser Position gehalten.

12. Abschalten des "ex_Stop_Drive1" (MC_Stop).

Das aktive Halten der realen Achse ist beendet.

13. Leistungsabschaltung für die reale Achse.

Abschalten von "en_Power_Drive1" (MC_Power) durch "Werte schreiben".



Nur möglich im Zustand "Standstill" nach Ausführung des Stopps an "Drive1".

3.4.8 Aufzeichnung der Bewegung beider Achsen (Trace), FUP

Das Trace-Werkzeug wurde analog zum Abschnitt [Kap. 3.4.4 "Aufzeichnung der Bewegung von Achse vAxis1 \(Trace\), FUP" auf Seite 106](#), 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_Power_Drive1" (MC_Power).
2. Start der virtuellen Achse.
Zuschalten des "ex_MoveVel_vAxis1" (MC_MoveVelocity).
3. Nachführen der realen Achse.
Zuschalten des "ex_GearIn_Drive1" (MC_GearIn).
4. Drehzollsollwertsprung der virtuellen Achse.
Erhöhen des Drehzollsollwertes auf "250".
Wirksamschalten durch "ex_MoveVel_vAxis1" TRUE $\Rightarrow$ FALSE $\Rightarrow$ TRUE.

Die nachstehende Abbildung zeigt das Trace-Ergebnis:

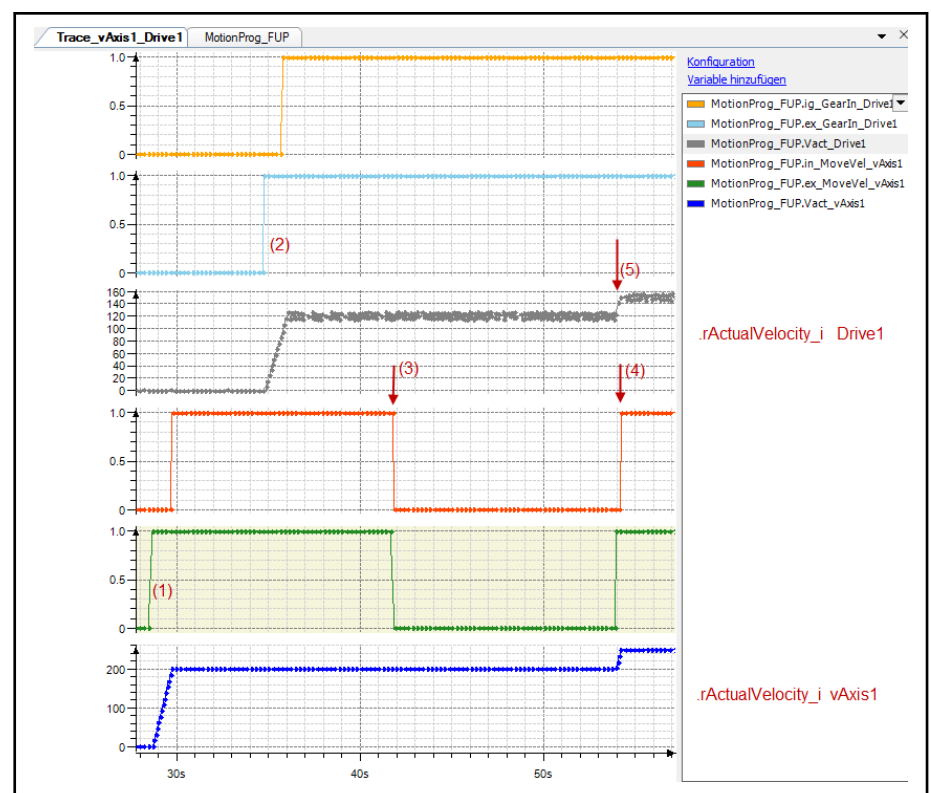


Abb. 3-49: Traceaufzeichnung für beide Achsen

Auswertung der Trace-Aufzeichnung

- (1) - Start des Hochlaufes der virtuellen Achse
- (2) - Zuschalten der Getriebe-Funktion, Hochlauf der realen Achse

Erste Schritte – Motion

- (3) - Erhöhen der Solldrehzahl der virtuellen Achse, "ex_MoveVel_vAxis1" TRUE ⇒ FALSE
- (4) - Wirksamwerden der Solldrehzahl der virtuellen Achse, "ex_MoveVel_vAxis1" FALSE ⇒ TRUE
- (5) - Sprungantwort an der realen Achse

3.5 Programmieren des Bewegungsablaufes in Strukturiertem Text (ST)

3.5.1 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 "Mlc1", der realen Achse "Drive1" und der virtuellen Achse "vAxis1" wurden unterhalb des Ordners "Logic" erste Aktivitäten zur Programmierung des Bewegungsablaufes begonnen:

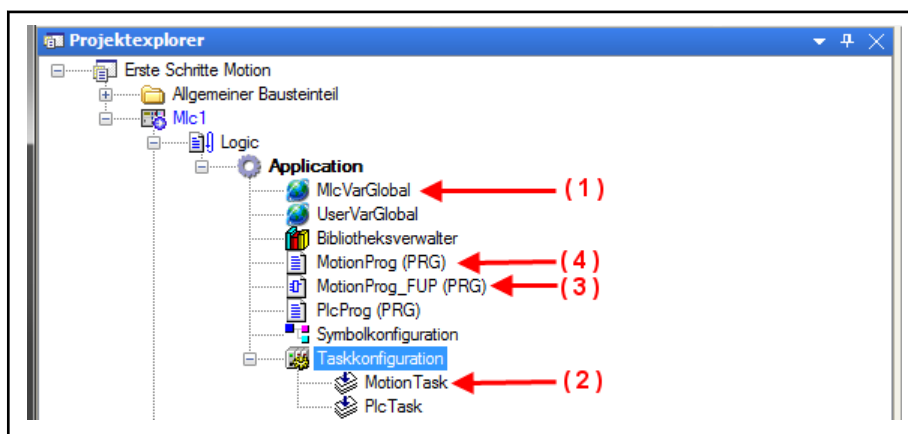


Abb. 3-50: Projektexplorer nach Expandieren der Knoten "Logic"

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 bzw. 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**, (niederpriorie 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 Drive1 und vAxis1

Programm:

```
(* Automated generated code by MLC. *)
(* Please don't edit, the code will be overwritten...*)
VAR_GLOBAL CONSTANT
(*virtual axis "vAxis1" with axis number 2*)
vAxis1: AXIS_REF := (CntlNo:=LOCAL_CNTRL,AxisNo:=AXIS_2);
(*real axis "Drive1" with axis number 1*)
Drive1: AXIS_REF := (CntlNo:=LOCAL_CNTRL,AxisNo:=AXIS_1);
(*...*)
END_VAR
```



Für "Ersten Schritte Motion" wird nur die MotionTask genutzt.

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

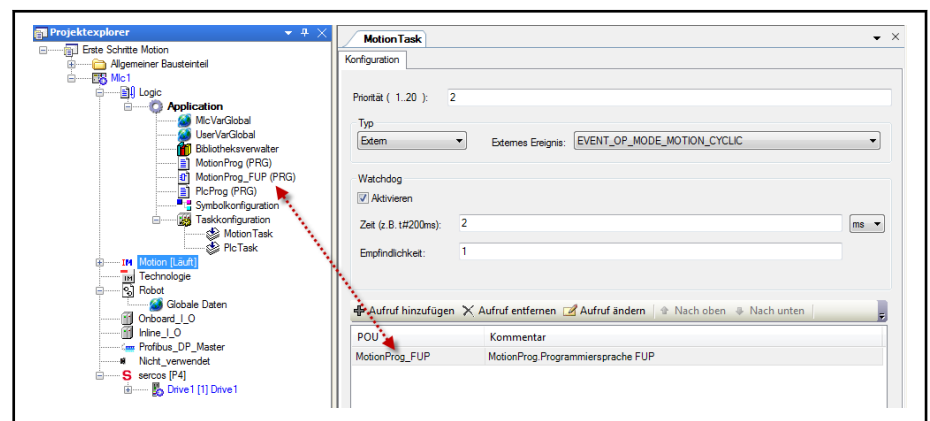


Abb. 3-51: Aufruf löschen: MotionProg_FUP; Aufruf hinzufügen: MotionProg

3.5.2 Programmieren der virtuellen Achse "vAxis1", 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 [Kap. 3.1 "Ziel des Projekts Erste Schritte – Motion" auf Seite 79](#), werden Instanzen der PLCopen-Bausteine "MC_MoveVelocity" und "MC_Stop" unter Nutzung von "vAxis1" 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).

Erste Schritte – Motion



Abb. 3-52: Neues Programm in ST (Strukturierter Text)

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 "MC_MoveVel_vAxis1:" ein. Rufen Sie per rechter Maustaste das Kontextmenü und hier die Eingabehilfe (alternativ <F2>-Fenster) auf.

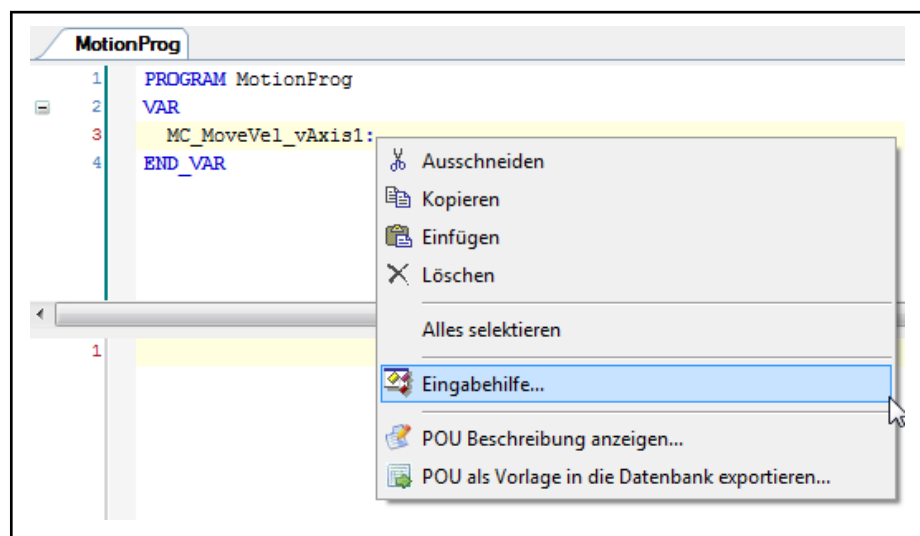


Abb. 3-53: Implementation der Instanz "MoveVel_vAxis1" vom Typ "MC_MoveVelocity"

Wählen Sie dazu im geöffneten Auswahlfenster

Benutzerdefinierte Typen

► ML_PLCopen ► POUs ► Motion ► MC_MoveVelocity aus.

Wählen Sie die Option "Mit Argumenten einfügen" ab.

Erste Schritte – Motion

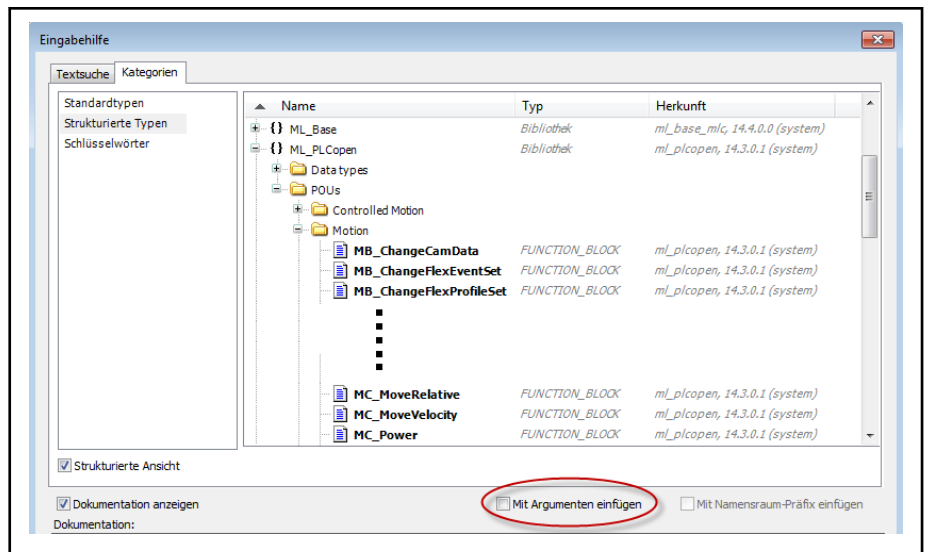


Abb. 3-54: Auswahlfenster Funktionsbausteine

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

Gehen Sie in den Implementationsteil des Editors auf Zeile 1 und öffnen Sie mit <F2> die Eingabehilfe.

Wählen Sie im geöffneten Auswahlfenster **Instanzaufrufe** ► **MoveVel_vAxis1** aus.

Wählen Sie die Option "Mit Argumenten einfügen".

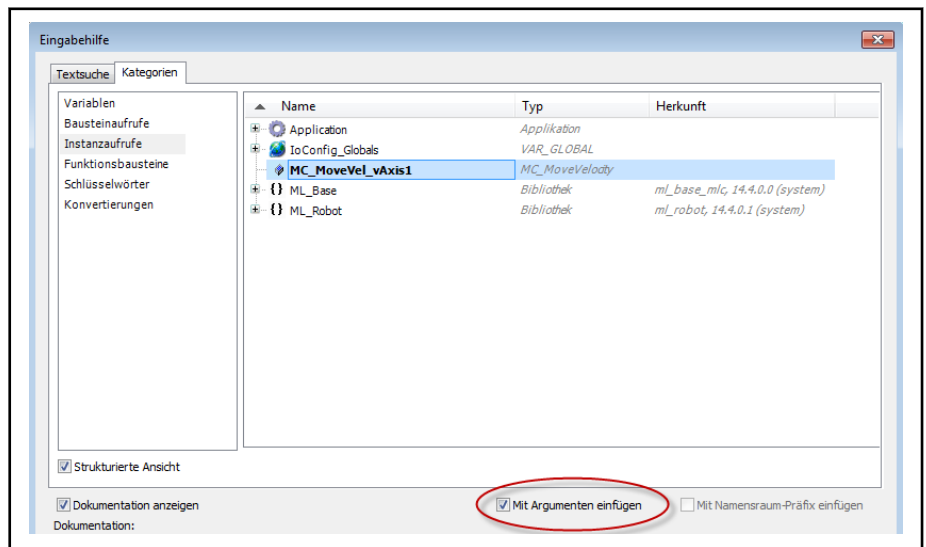


Abb. 3-55: Übernahme der Instanz in die Implementation

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

Erste Schritte – Motion

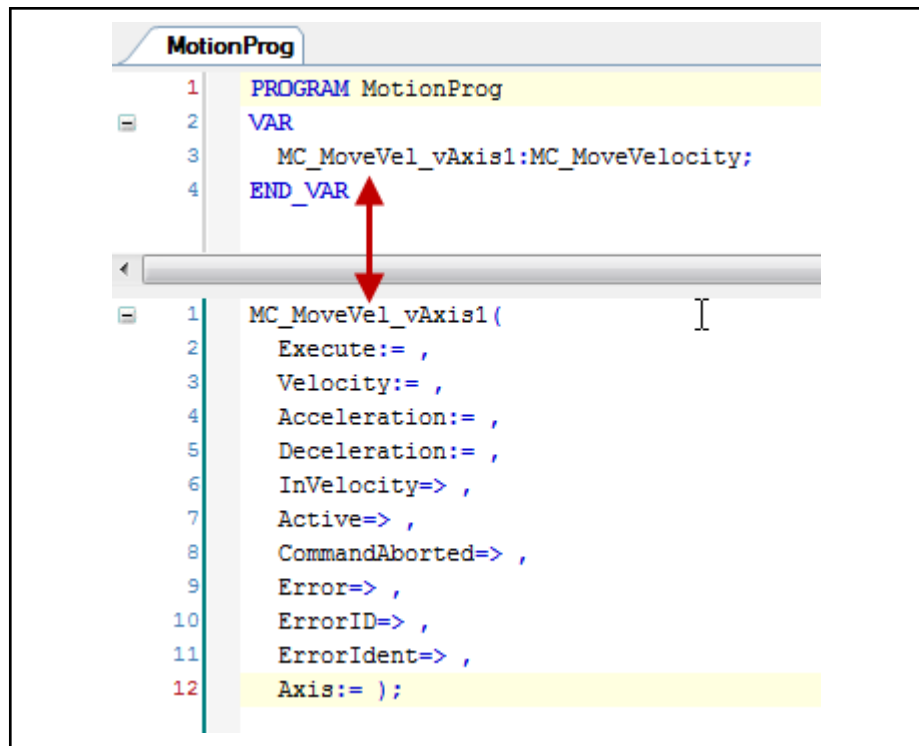


Abb. 3-56: Gerüst einer Funktionsbaustein-Instanz

Beschalten Sie nun die Eingänge des Bausteines.

Beginnen Sie mit dem Eingang "Execute":

- Geben Sie den Namen "ex_MoveVel_vAxis1" ein und klicken die nächste Zeile an.
- 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 "Vcom_vAxis1" ein und klicken die nächste Zeile an.
- Geben Sie als Typ REAL und als Initialwert 200 (U/min) ein und bestätigen Sie mit <Enter>
- Für die Eingänge "Acceleration" und "Deceleration" tragen Sie bitte 20 (rad/s²) ein.



Die Werte werden direkt an den Bausteineingängen eingetragen.

Alternativ kann eine Variable mit dem entsprechenden Wert verwendet werden.

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_MoveVel_vAxis1", Typ "BOOL"
- Geben Sie den Namen "iv_MoveVel_vAxis1" ein und klicken die nächste Zeile an.
- Das sich öffnende Fenster zur "Variablendeklaration" können Sie sofort mit <Enter> bestätigen
- Beschalten Sie den "Active"-Ausgang mit "ac_MoveVel_vAxis1"

Erste Schritte – Motion

- Beschalten Sie den "CommandAborted"-Ausgang mit "ca_MoveVel_vAxis1"
- Beschalten Sie den "Error"-Ausgang mit "er_MoveVel_vAxis1"
- Beschalten Sie den "ErrorID"-Ausgang mit "ID_MoveVel_vAxis1"
- Beschalten Sie den "ErrorIdent"-Ausgang mit "Ident_MoveVel_vAxis1"

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

- Öffnen Sie mit der rechten Maustaste die Eingabehilfe:
Wählen Sie **Variablen** ► **Application** ► **MlcVarGlobal** ► **vAxis1**.

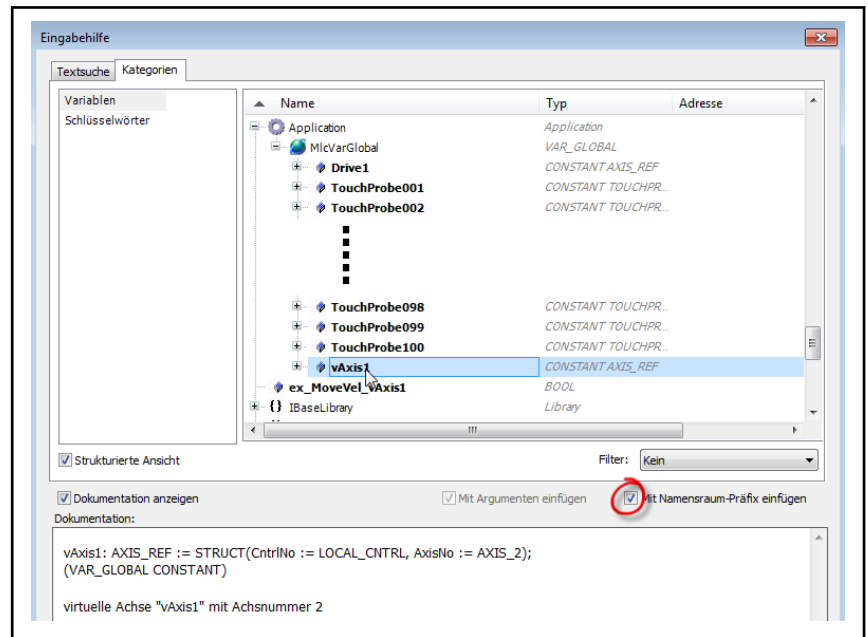


Abb. 3-57: Auswahl der Achsinstanz bei Beschaltung eines Funktionsbausteines vom Typ "MC_MoveVelocity"

- Bestätigen Sie mit **OK**.



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

`MlcVarGlobal.vAxis1`

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

Erste Schritte – Motion

```

MotionProg
2  VAR
3      MC_MoveVel_vAxis1:MC_MoveVelocity;
4      ex_MoveVel_vAxis1: BOOL;
5      Vcom_vAxis1: REAL := 200;
6      iv_MoveVel_vAxis1: BOOL;
7      ac_MoveVel_vAxis1: BOOL;
8      ca_MoveVel_vAxis1: BOOL;
9      er_MoveVel_vAxis1: BOOL;
10     ID_MoveVel_vAxis1: ERROR_CODE;
11     Ident_MoveVel_vAxis1: ERROR_STRUCT;
12 END_VAR

1  MC_MoveVel_vAxis1(
2      Execute:= ex_MoveVel_vAxis1,
3      Velocity:= Vcom_vAxis1,
4      Acceleration:= 20,
5      Deceleration:= 20,
6      InVelocity=> iv_MoveVel_vAxis1,
7      Active=> ac_MoveVel_vAxis1,
8      CommandAborted=> ca_MoveVel_vAxis1,
9      Error=> er_MoveVel_vAxis1,
10     ErrorID=> ID_MoveVel_vAxis1,
11     ErrorIdent=> Ident_MoveVel_vAxis1 ,
12     Axis:= MlcVarGlobal.vAxis1 );
  
```

Abb. 3-58: PLCopen-FB Typ "MC_MoveVelocity" ist eingefügt und beschaltet



Immer wenn Sie einen Baustein abgeschlossen haben, sollten Sie eine Testübersetzung (<Umschalt>+<F11>) durchführen. Beseitigen Sie danach eventuell vorhandene Fehler sofort.

Generieren Sie nun eine Instanz des PLCopen-Bausteins "MC_Stop".

MC_Stop

Instanz "MC_Stop_vAxis1" von "MC_Stop" deklarieren und einfügen

Eingänge beschalten und wenn nötig Eingangsvariablen deklarieren:

"Execute" → "ex_Stop_vAxis1" vom Typ "BOOL"

"Deceleration" → 50

"Axis" → "MlcVarGlobal.vAxis1" vom Typ "AXIS_REF"

Ausgänge:

"Done" → "do_Stop_vAxis1" vom Typ "BOOL"

"Active" → "ac_Stop_vAxis1" vom Typ "BOOL"

"CommandAborted" → "ca_Stop_vAxis1" vom Typ "BOOL"

"Error" → "er_Stop_vAxis1" vom Typ "BOOL"

"ErrorID" → "ID_Stop_vAxis1" vom Typ "ERROR_CODE"

"ErrorIdent" → "Ident_Stop_vAxis1" vom Typ "ERROR_STRUCT"

Istgeschwindigkeit "Vact_vAxis1"

Als Letztes soll ein Netzwerk, in dem die aktuelle Geschwindigkeit der Achse "vAxis1" "Vact_vAxis1" (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

GVL_Base.AxisData[2].rActualVelocity_i

für die virtuelle Achse, Achsnummer 2, bereitgestellt ("Drive1" hat Achsnummer 1, da das Scannen von "Drive1" vor der Deklaration von "vAxis1" erfolgte).

- Neue Zeile anfügen
- Vact_vAxis1:= einfügen, Deklaration komplettieren
- unter Nutzung der Eingabehilfe zu **Variablen ▶ ML_BASE ▶ Global Variables List ▶ GVL_Base ▶ AxisData[]** navigieren:

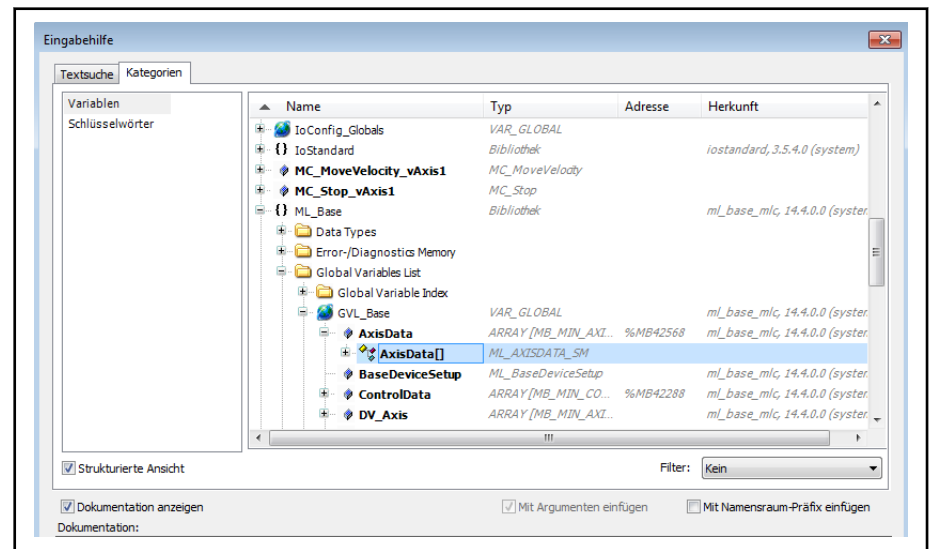


Abb. 3-59: Geschwindigkeitswert "vAxis1" aus der Globalen Variablenliste der Bibliothek ML_Base.library

Die weitere Eingabe wird durch die Intellisense-Funktion (automatische Vervollständigung) unterstützt; d. h., dass nach der Eingabe des "." bei "AxisData[2]." alle verfügbaren Elemente aufgerollt werden und zur Auswahl stehen.

"rActualVelocity_i" mit <Enter> bestätigen.

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

Erste Schritte – Motion

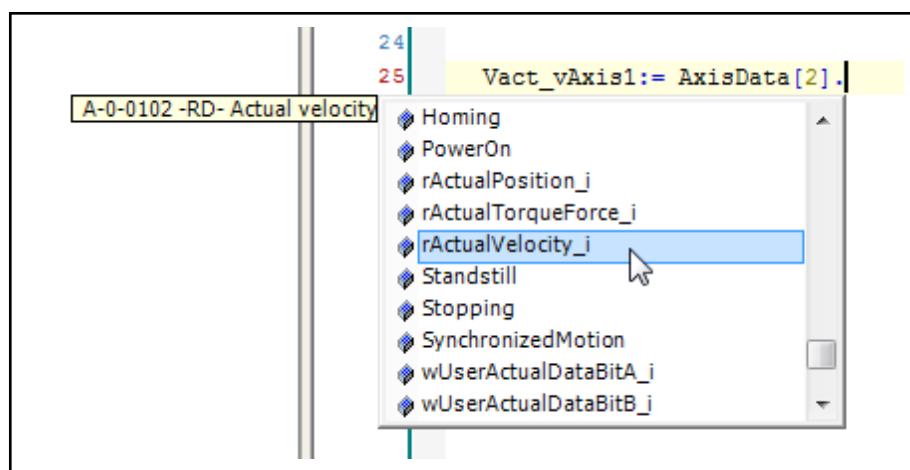


Abb. 3-60: Auswahl der Variablen aktuelle Geschwindigkeit

- Abschluss der Zeile mit ";" nicht vergessen.

Damit ist die Implementierung des Bewegungsablaufes entsprechend der Aufgabenstellung abgeschlossen:

```

MotionProg
1  PROGRAM MotionProg
2  VAR
3      MC_MoveVel_vAxis1:MC_MoveVelocity;
4      ex_MoveVel_vAxis1: BOOL;
5      Vcom_vAxis1: REAL := 200;
6      iv_MoveVel_vAxis1: BOOL;
7      ac_MoveVel_vAxis1: BOOL;
8      ca_MoveVel_vAxis1: BOOL;
9      er_MoveVel_vAxis1: BOOL;
10     ID_MoveVel_vAxis1: ERROR_CODE;
11     Ident_MoveVel_vAxis1: ERROR_STRUCT;

1  MC_MoveVel_vAxis1(
2      Execute:= ex_MoveVel_vAxis1,
3      Velocity:= Vcom_vAxis1,
4      Acceleration:= 20,
5      Deceleration:= 20,
6      InVelocity=> iv_MoveVel_vAxis1,
7      Active=> ac_MoveVel_vAxis1,
8      CommandAborted=> ca_MoveVel_vAxis1,
9      Error=> er_MoveVel_vAxis1,
10     ErrorID=> ID_MoveVel_vAxis1,
11     ErrorIdent=> Ident_MoveVel_vAxis1 ,
12     Axis:= MlcVarGlobal.vAxis1 );

14 MC_Stop_vAxis1(
15     Execute:= ex_Stop_vAxis1,
16     Deceleration:= 50,
17     Done=> do_Stop_vAxis1,
18     Active=> ac_Stop_vAxis1,
19     CommandAborted=> ca_Stop_vAxis1,
20     Error=> er_Stop_vAxis1,
21     ErrorID=> ID_Stop_vAxis1,
22     ErrorIdent=> Ident_Stop_vAxis1,
23     Axis:= MlcVarGlobal.vAxis1);

25 Vact_vAxis1:= AxisData[2].rActualVelocity_i;
    
```

Abb. 3-61: Implementation der virtuellen Achse "vAxis1"



AxisData[].rActualVelocity_i; und
GVL_Base.AxisData[].rActualVelocity_i;
werden vom Compiler als gleichwertig verwendet.

3.5.3 Download des SPS-Teilprojektes virtuelle Achse "vAxis1" (ST)

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

Die Übersetzung erreichen Sie über die IndraWorks-Menüleiste

Erste Schritte – Motion

- **Erstellen ► Übersetzen**
Übersetzen des hinzu gekommenen Anteils der Applikation
- **Erstellen ► Code erzeugen**
Der Befehl ermöglicht eine Code-Generierung zur Überprüfung von Übersetzungsfehlern

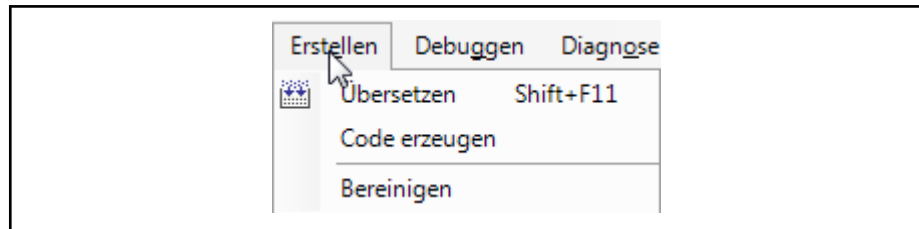


Abb. 3-62: Applikation übersetzen

Zum Übertragen der kompilierten Daten in die Zielsteuerung wählen Sie in der IndraWorks-Menüleiste **Debuggen ► Einloggen** (siehe nachfolgende Abbildung).

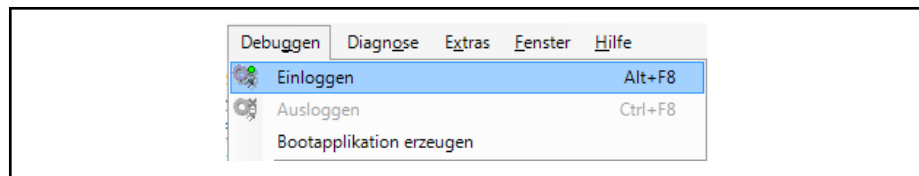


Abb. 3-63: Übertragen der kompilierten Daten in die Steuerung

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.

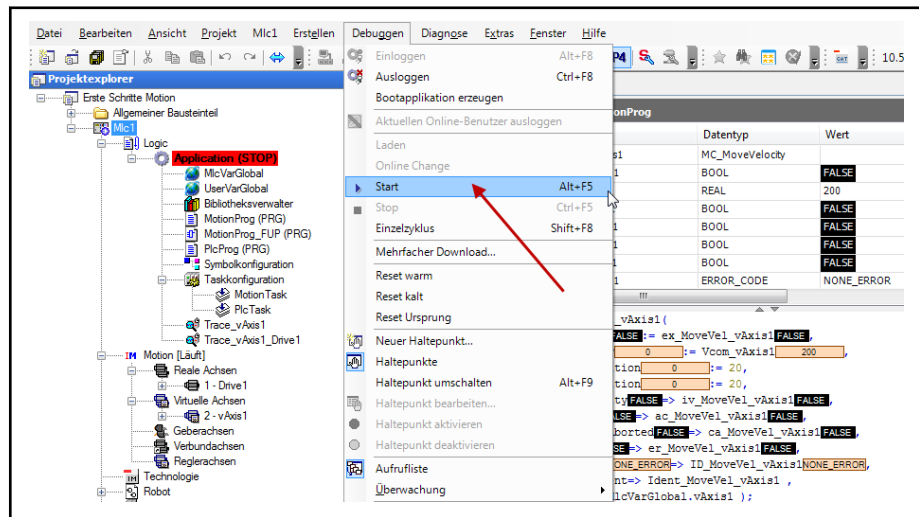


Abb. 3-64: Starten der Programmbearbeitung in der Steuerung (1)

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

Erste Schritte – Motion

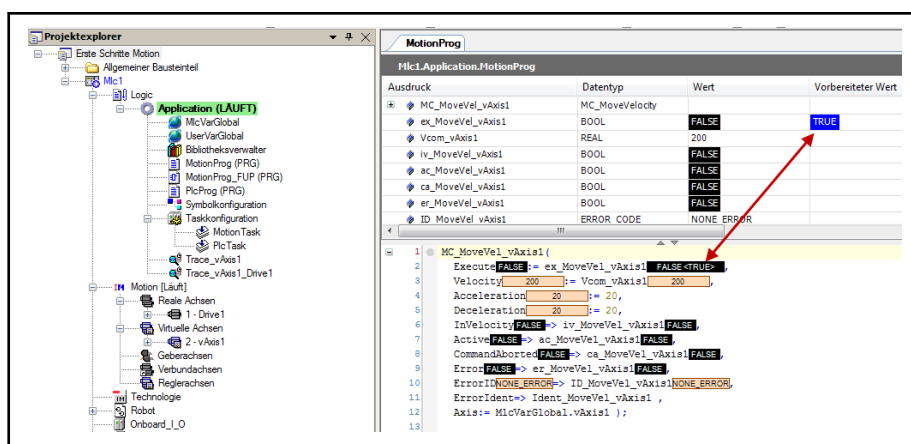


Abb. 3-65: Starten der Programmbearbeitung in der Steuerung (2)

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

"Werte schreiben",
Eingangsvariablen

Um die Bearbeitung des "MoveVel_vAxis1" vom Typ "MC_MoveVelocity" zu aktivieren, muss am "Execute"-Eingang des Bausteins eine positive Flanke auftreten.

Klicken Sie mit der linken Maustaste auf das "FALSE" hinter der Eingangsvariablen "ex_MoveVel_vAxis1", das Fenster "Wert vorbereiten" öffnet, "TRUE" eintragen und bestätigen.

Jetzt ist hinter "ex_MoveVel_vAxis1" ein markiertes "FALSE <TRUE>" und in der Deklarationszeile ein "TRUE" zu sehen.

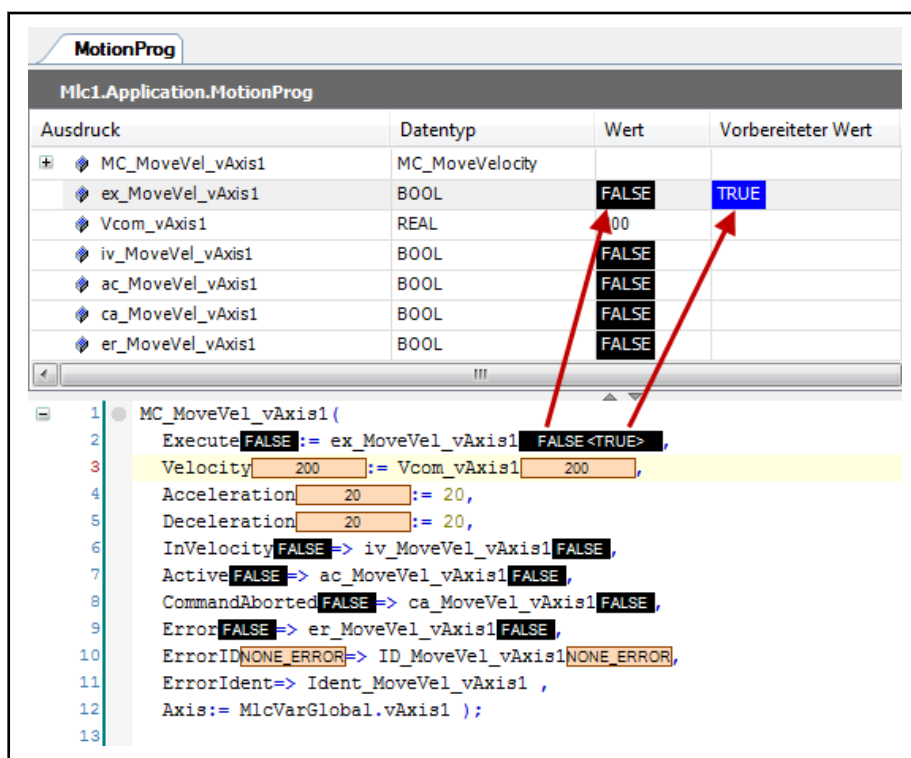


Abb. 3-66: Vorbereitung des "Werte schreiben"

Über **Debuggen** ► **Werte schreiben** <Strg>+<F7> wird "ex_MoveVel_vAxis1" 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

Die dadurch entstehende positive Flanke setzt die Ausgänge "InVelocity" und "Active" vom Funktionsbaustein "MoveVel_vAxis1" auf "TRUE" (1) (siehe nachfolgende Abbildung).

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

The screenshot displays the MotionProg interface for the function block `Mlc1.Application.MotionProg`. It includes a variable declaration table and a ladder logic program.

Ausdruck	Datentyp	Wert	Vorbereiteter Wert	Adr
MC_MoveVel_vAxis1	MC_MoveVelocity			
ex_MoveVel_vAxis1	BOOL	TRUE		
Vcom_vAxis1	REAL	200		
iv_MoveVel_vAxis1	BOOL	TRUE		
ac_MoveVel_vAxis1	BOOL	TRUE		
ca_MoveVel_vAxis1	BOOL	FALSE		
er_MoveVel_vAxis1	BOOL	FALSE		
ID_MoveVel_vAxis1	ERROR_CODE	NONE_ERROR		

The ladder logic program below the table shows the configuration of the `MC_MoveVel_vAxis1` function block. Red arrows and labels (1) and (2) highlight specific parts of the code:

```

1 MC_MoveVel_vAxis1(
2   Execute TRUE := ex_MoveVel_vAxis1 TRUE,
3   Velocity 200 := Vcom_vAxis1 200,
4   Acceleration 20 := 20,
5   Deceleration 20 := 20,
6   InVelocity TRUE => iv_MoveVel_vAxis1 TRUE, (1)
7   Active TRUE => ac_MoveVel_vAxis1 TRUE, (1)
8   CommandAborted FALSE => ca_MoveVel_vAxis1 FALSE,
9   Error FALSE => er_MoveVel_vAxis1 FALSE,
10  ErrorID NONE_ERROR => ID_MoveVel_vAxis1 NONE_ERROR,
11  ErrorIdent => Ident_MoveVel_vAxis1,
12  Axis := MlcVarGlobal.vAxis1 );
13
14 MC_Stop_vAxis1(
15   Execute FALSE := ex_Stop_vAxis1 FALSE,
16   Deceleration 50 := 50,
17   Done FALSE => do_Stop_vAxis1 FALSE,
18   Active FALSE => ac_Stop_vAxis1 FALSE,
19   CommandAborted FALSE => ca_Stop_vAxis1 FALSE,
20   Error FALSE => er_Stop_vAxis1 FALSE,
21   ErrorID NONE_ERROR => ID_Stop_vAxis1 NONE_ERROR,
22   ErrorIdent => Ident_Stop_vAxis1,
23   Axis := MlcVarGlobal.vAxis1 );
24
25 Vact_vAxis1 200 := AxisData[2].rActualVelocity_i 200; RETURN
  
```

Red arrow (1) points to the `InVelocity` and `Active` assignments. Red arrow (2) points to the `Vact_vAxis1` assignment.

Abb. 3-67: Nach "Werte schreiben" am Eingang "Execute"

Schalten Sie "ex_MoveVel_vAxis1" wieder auf "FALSE". Die Achse läuft weiter mit 200 U/min.

Sie können durch Setzen der Variable "ex_Stop_vAxis1" am Eingang "Execute" des Funktionsbausteines "Stop_vAxis1" vom Typ "MC_Stop" die Bewegung der Achse stoppen.

Stoppen Sie die SPS und loggen Sie aus.

3.5.4 Aufzeichnung der Bewegung von Achse "vAxis1" (Trace), ST

IndraLogic liefert zur Aufzeichnung von Bewegungsvorgängen ein Trace-Werkzeug. Mit Hilfe dieses Werkzeugs soll Folgendes aufgezeichnet werden:

- die Achsgeschwindigkeit "Vact_vAxis1"
- für "MC_MoveVel_vAxis1" (MC_MoveVelocity)
 - "ex_MoveVel_vAxis1" (Execute)
 - "in_MoveVel_vAxis1" (InVelocity)
 - "ca_MoveVel_vAxis1" (CommandAborted)

und

- für "MC_Stop_vAxis1" (MC_Stop)
 - "ex_Stop_vAxis1" (Execute)

Dazu muss zunächst die Traceaufzeichnung konfiguriert werden.

Fügen Sie über das Kontextmenü von "Application" **Hinzufügen ► Trace...** das Objekt "Trace_vAxis1" ein. Öffnen Sie das Objekt mit einem Doppelklick.

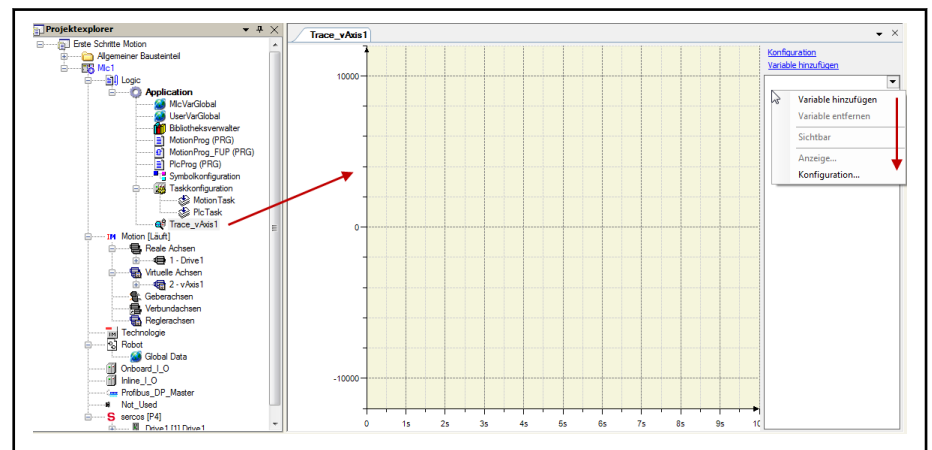


Abb. 3-68: Dialog: Neues Trace-Objekt "Trace_vAxis1" mit Kontextmenü

Über das Kontextmenü auf der rechten Blattseite können für das Trace-Objekt folgende Einstellungen vorgenommen werden:

- Konfiguration
- Anzeigeeinstellungen
- Neue Variable hinzufügen

Konfiguration... Das neue Trace-Objekt ist zunächst zu konfigurieren:

Erste Schritte – Motion

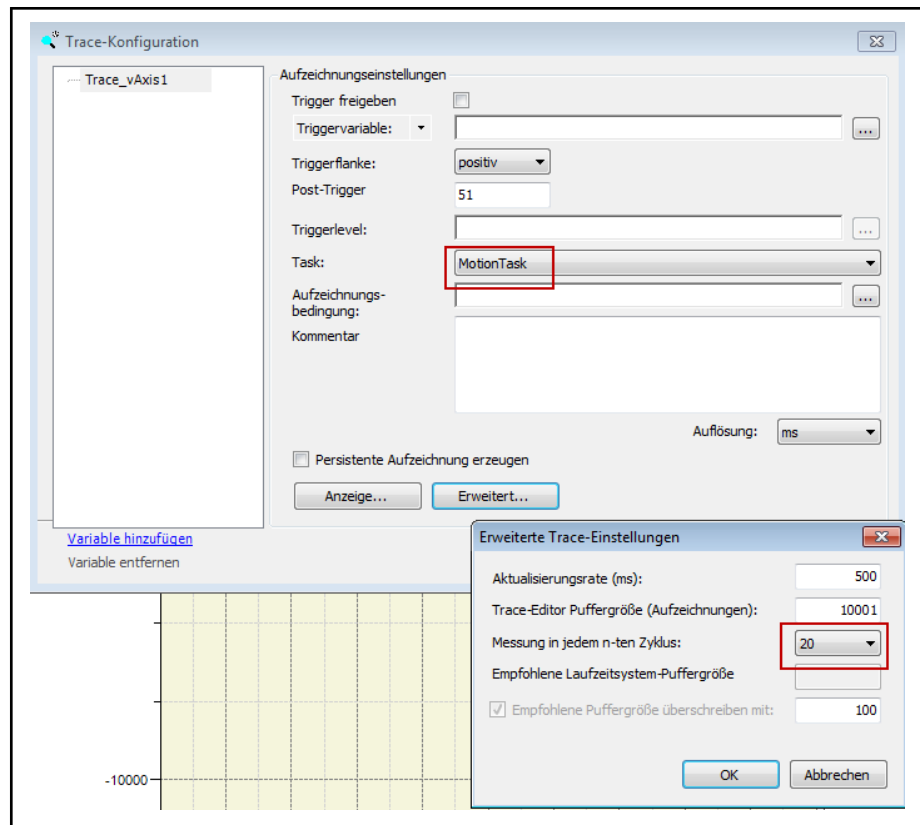


Abb. 3-69: Dialog: Trace-Konfiguration mit erweiterter Trace-Einstellung

Notwendige Einstellungen sind

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

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

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.

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

Empfohlene Laufzeitsystem-Puffergröße: Größe des Ringpuffers in Byte, 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.

Erste Schritte – Motion

Anzeigeeinstellungen

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

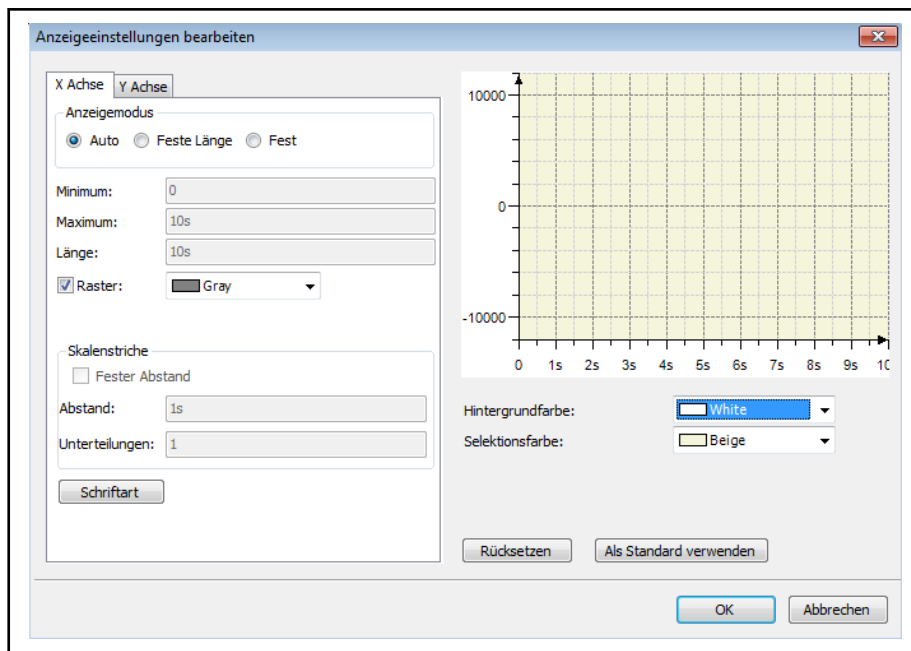


Abb. 3-70: Dialog: Anzeigeeinstellungen

Anzuwählen ist **Auto**, hierbei wird ein Fenster der **Länge** 10 s über den Kurvenzug gelegt. Die **Skaleneinteilung** wird dabei im 1 s Raster mit einer **Skalenunterteilung** von 1 s vorgenommen.

Auswahl der Neuen Variablen

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

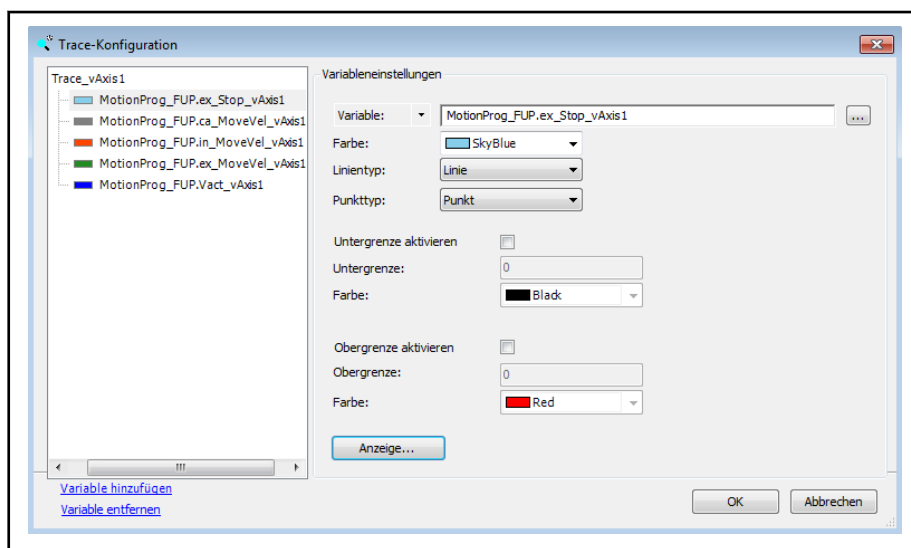


Abb. 3-71: Dialog: Anlegen von neuen Variablen und Festlegung des Erscheinungsbildes

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

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.

Erste Schritte – Motion

Menü zum Steuern der Trace-Aufzeichnung:

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

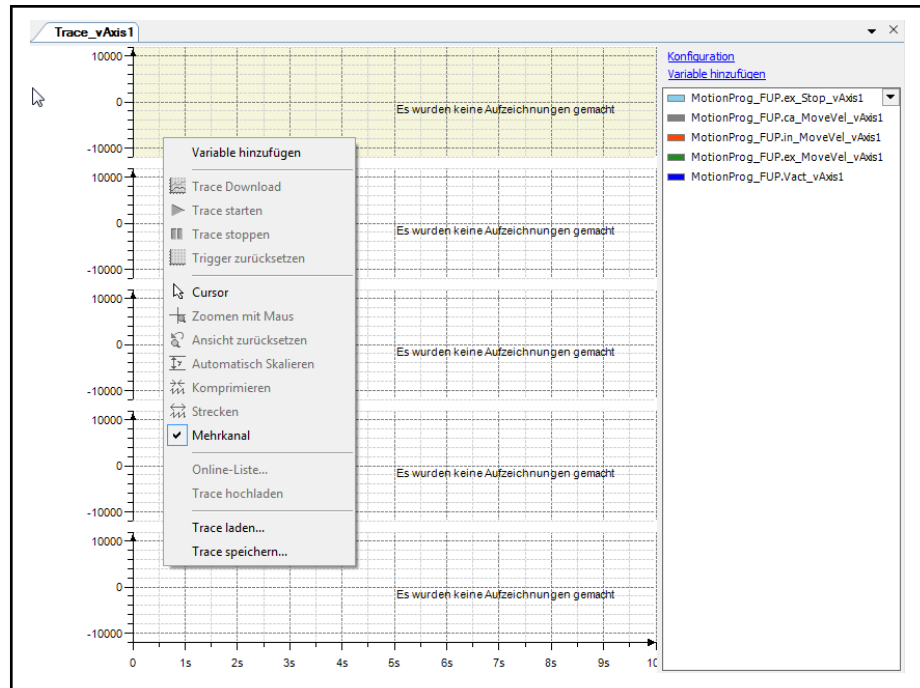


Abb. 3-72: Kontextmenü zum Steuern der Trace-Aufzeichnung

Trace Download: Bevor eine konfigurierte Trace-Aufzeichnung auf der Steuerung für die auf der Steuerung laufende Applikation aktiv werden kann, muss die "Trace-Applikation" explizit auf die Steuerung 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.

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

- Durchführung des Trace:**
1. Einloggen auf der Steuerung.
Herstellen der Verbindung zur Steuerung.
 2. Trace Download.
Bei laufender Application werden über das Trace-Kontextmenü "Trace Download" die erforderlichen Daten in die Steuerung übertragen.
Die Trace-Aufzeichnung läuft danach automatisch an.
 3. Werte schreiben.
Folgende Werte werden analog zu ["Werte schreiben, Eingangsvariablen" auf Seite 129](#) gesetzt:
 - "ex_MoveVel_vAxis1" auf "TRUE"
 - "ex_Stop_vAxis1" auf "TRUE"
 4. Beenden der Aufzeichnung.
Stoppen des Trace durch den Trace-Kontextmenüpunkt "Start/Stop Trace".
 5. Auswertung der Trace-Aufzeichnung.
Einstellen der Fensterdaten.

Tastenkombinationen	Aktion im Trace-Diagramm	Entsprechende Steuerung mit Maus
<Alt>+<←/→>-Pfeiltasten	horizontales Scrollen der X-Achse (Zeitachse) aller Diagramme	Aktivieren von Scrollen mit Maus, Klicken in das Diagramm und scrollen der X-Achse mit der Maus während die linke Maustaste gedrückt wird
<↑/↓>-Pfeiltasten, mit oder ohne gedrückte <Alt>-Taste Selektieren eines Diagramms mit <Tab>	horizontales Scrollen des selektierten Diagramms	Aktivieren von Scrollen mit Maus, Klicken in das Diagramm und scrollen der X-Achse mit der Maus während die linke Maustaste und <Strg> gedrückt wird
<↑/↓>-Pfeiltasten, mit oder ohne gedrückte <Alt>-Taste Selektieren eines Diagramms mit <Tab> Zusätzlich <Strg> drücken	Der Graph wird in größeren Schritten verschoben	
<+/->-Tasten	horizontales Zoomen aller Diagramme entlang der X-Achse (die Zeitachse wird für alle Diagramme gestreckt/komprimiert)	• Auswahl "Strecken" oder "Komprimieren":  oder  • bzw. Verwenden des Mausekzes
<Strg>+<+/->-Tasten Selektieren eines Diagramms mit <Tab>	vertikales Zoomen aller Diagramme entlang der Y-Achse (die der Y-Achse des selektierten Graphen wird gestreckt/komprimiert)	

Tab. 3-2: Tastenkombinationen Trace-Editor mit aktiviertem Mehrfachkanal

Erste Schritte – Motion

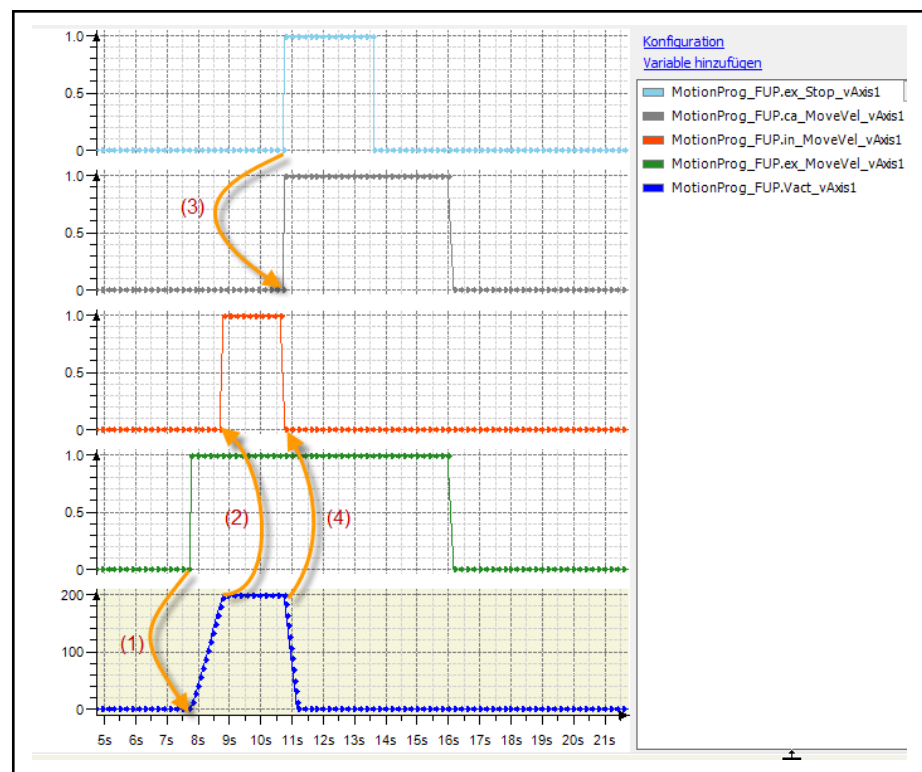


Abb. 3-73: Traceaufzeichnung für die virtuelle Achse vAxis1

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, gleichparametrierten Achse verhält (vergleiche Geschwindigkeitsverlauf, Beschleunigung mit 20 rad/s^2 und Verzögerung mit 50 rad/s^2).

3.5.5 Programmieren der realen Achse "Drive1" (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 [Kap. 3.4.2 "Programmieren der virtuellen Achse vAxis1, Programmiersprache FUP"](#) auf Seite 96.

MC_Power

Neue Zeile einfügen

"MC_Power" einfügen und eine Instanz "MC_Power_Drive1" deklarieren

Eingänge beschalten und wenn nötig Eingangsvariablen deklarieren:

"Enable" → "en_Power_Drive1" vom Typ "BOOL"

"Axis" → "MlcVarGlobal.Drive1" vom Typ "AXIS_REF"

Ausgänge:

"Status" → "st_Power_Drive1" vom Typ "BOOL"

"Error" → "er_Power_Drive1" vom Typ "BOOL"

"ErrorID" → "ID_Power_Drive1" vom Typ "ERROR_CODE"

"ErrorIdent" → "Ident_Power_Drive1" vom Typ "ERROR_STRUCT"

MC_GearIn

Neue Zeile einfügen

"MC_GearIn" einfügen und Instanz "MC_GearIn_Drive1" deklarieren

Eingänge beschalten und wenn nötig Eingangsvariablen deklarieren:

"Execute" → "ex_GearIn_Drive1" vom Typ "BOOL"

"RatioNumerator" → 5

"RatioDenominator" → 10

"MasterFineadjust" → 20 (%)

"Master" → "MlcVarGlobal.vAxis1" vom Typ "AXIS_REF"

"Slave" → "MlcVarGlobal.Drive1" vom Typ "AXIS_REF"

Ausgänge:

"InGear" → "ig_GearIn_Drive1" vom Typ "BOOL"

"Active" → "ac_GearIn_Drive1" vom Typ "BOOL"

"CommandAborted" → "ca_GearIn_Drive1" vom Typ "BOOL"

"Error" → "er_GearIn_Drive1" vom Typ "BOOL"

"ErrorID" → "ID_GearIn_Drive1" vom Typ "ERROR_CODE"

"ErrorIdent" → "Ident_GearIn_Drive1" vom Typ "ERROR_STRUCT"

MC_GearOut

Neue Zeile einfügen

"MC_GearOut" einfügen und eine Instanz "MC_GearOut_Drive1" deklarieren

Eingänge beschalten und wenn nötig Eingangsvariablen deklarieren:

"Execute" → "ex_GearOut_Drive1" vom Typ "BOOL"

"Slave" → "MlcVarGlobal.Drive1" vom Typ "AXIS_REF"

Ausgänge:

"Done" → "do_GearOut_Drive1" vom Typ "BOOL"

"Active" → "ac_GearOut_Drive1" vom Typ "BOOL"

"Error" → "er_GearOut_Drive1" vom Typ "BOOL"

Erste Schritte – Motion

"ErrorID" → "ID_GearOut_Drive1" vom Typ "ERROR_CODE"

"ErrorIdent" → "Ident_GearOut_Drive1" vom Typ "ERROR_STRUCT"

MC_Stop

Neue Zeile einfügen

"MC_Stop" einfügen und eine Instanz "MC_Stop_Drive1" deklarieren

Eingänge beschalten und wenn nötig Eingangsvariablen deklarieren:

"Execute" → "ex_Stop_Drive1" vom Typ "BOOL"

"Deceleration" → 50 (rad/s²)

"Axis" → "MlcVarGlobal.Drive1" vom Typ "AXIS_REF"

Ausgänge:

"Done" → "do_Stop_Drive1" vom Typ "BOOL"

"Active" → "ac_Stop_Drive1" vom Typ "BOOL"

"CommandAborted" → "ca_Stop_Drive1" vom Typ "BOOL"

"Error" → "er_Stop_Drive1" vom Typ "BOOL"

"ErrorID" → "ID_Stop_Drive1" vom Typ "ERROR_CODE"

"ErrorIdent" → "Ident_Stop_Drive1" vom Typ "ERROR_STRUCT"

Istgeschwindigkeit "Vact_Drive1"

Als Letztes soll ein Netzwerk, in dem die aktuelle Geschwindigkeit der Achse "Drive1" "Vact_Drive1" (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 IndraMotion 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 ("Drive1" hat Achsnummer 1, da das Scannen von "Drive1" vor der Deklaration von "vAxis1" erfolgte).

- Neue Zeile anfügen
- `Vact_Drive1 :=` einfügen, Deklaration komplettieren
- unter Nutzung der Eingabehilfe zu **Globale Variablen ▶ ML_BASE ▶ Global Variables List ▶ GVL_Base ▶ AxisData** navigieren:

⁴⁾ Die Anzahl der steuerbaren Achsen ist vom Typ der Steuerung und ihrer Firmware abhängig, aktuell für L65: 64 Achsen

Erste Schritte – Motion

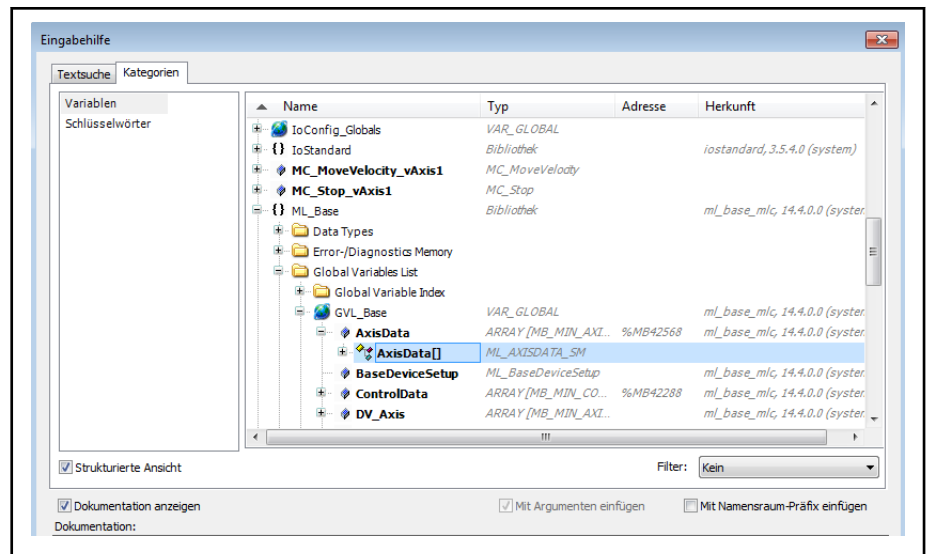


Abb. 3-74: Geschwindigkeitswert "vAxis1" aus der Globalen Variablenliste der Bibliothek "ML_Base.library"

Die weitere Eingabe wird durch die Intellisense-Funktion (automatische Vervollständigung) unterstützt; d. h., dass nach der Eingabe des "." bei "GVL_Base.AxisData[1]." alle verfügbaren Elemente aufgerollt werden und zur Auswahl stehen.

"rActualVelocity_1" mit <Enter> bestätigen.

- Abschluss der Zeile mit ";" nicht vergessen.

Erste Schritte – Motion

```

MC_Power_Drive1(
    Enable:= en_Power_Drive1,
    Status=> st_Power_Drive1,
    Error=> er_Power_Drive1,
    ErrorID=> ID_Power_Drive1,
    ErrorIdent=> Ident_Power_Drive1,
    Axis:= MlcVarGlobal.Drive1);

MC_GearIn_Drive1(
    Execute:=ex_GearIn_Drive1,
    RatioNumerator:= 5,
    RatioDenominator:= 10,
    MasterFineadjust:= 20,
    InGear=> ig_GearIn_Drive1,
    Active=> ac_GearIn_Drive1,
    CommandAborted=> ca_GearIn_Drive1,
    Error=> er_GearIn_Drive1,
    ErrorID=> ID_GearIn_Drive1,
    ErrorIdent=> Ident_GearIn_Drive1,
    Master:= MlcVarGlobal.vAxis1,
    Slave:= MlcVarGlobal.Drive1);

MC_GearOut_Drive1(
    Execute:= ex_GearOut_Drive1,
    Done=> do_GearOut_Drive1,
    Active=> ac_GearOut_Drive1,
    Error=> er_GearOut_Drive1,
    ErrorID=> ID_GearOut_Drive1,
    ErrorIdent=> Ident_GearOut_Drive1,
    Slave:= MlcVarGlobal.Drive1);

MC_Stop_Drive1(
    Execute:= ex_Stop_Drive1,
    Deceleration:= 50,
    Done=> do_Stop_Drive1,
    Active=> ac_Stop_Drive1,
    CommandAborted=> ca_Stop_Drive1,
    Error=> er_Stop_Drive1,
    ErrorID=> ID_Stop_Drive1,
    ErrorIdent=> Ident_Stop_Drive1,
    Axis:=MlcVarGlobal.Drive1 );

Vact_Drive1:=GVL_Base.AxisData[1].rActualVelocity_i;

```

Abb. 3-75: Implementation der realen Achse



AxisData[].rActualVelocity_i; und
GVL_Base.AxisData[].rActualVelocity_i;
werden vom Compiler als gleichwertig verwendet.

3.5.6 Download des kompletten SPS-Projektes (ST)

Der Ladevorgang verläuft analog zum Abschnitt [Kap. 3.5.3 "Download des SPS-Teilprojektes virtuelle Achse vAxis1 \(ST\)"](#) auf Seite 127.

Das neue Projekt überschreibt dabei das alte nach Rückfrage.

3.5.7 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_Power_Drive1" (MC_Power) durch "Werte schreiben".

st_Power_Drive1= TRUE (MC_Power).

2. Start der virtuellen Achse.

Zuschalten des "ex_MoveVel_vAxis1" (MC_MoveVelocity) durch "Werte schreiben".

Hochlauf bei Vact_vAxis1 = 200 U/min beendet.

3. Nachführen der realen Achse.

Zuschalten des "ex_GearIn_Drive1" (MC_GearIn) durch "Werte schreiben".

Das zu erwartenden Übersetzungsverhältnis des Getriebes ergibt sich zu:

$$\dot{U}_v = (\text{RatioNumerator} / \text{RatioDenominator}) \times (1 + \text{MasterFineadjust})$$

also bei Vact_vAxis1 = 200 U/min ergibt sich Vact_Drive1 = 120 U/min.



Der MasterFineadjust-Wert wird in "%" vorgegeben.

4. Befindet sich der MC_GearIn "InGear", so kann die Geschwindigkeit "Vcom_vAxis1" der virtuellen Achse durch "Werte schreiben" (Doppelklick auf die Variable, gewünschten Wert eintragen) verändert werden. Die reale Achse folgt der virtuellen.

5. Abschalten des "ex_GearIn_Drive1" (MC_GearIn) durch "Werte schreiben".

in_GearIn_Drive1= FALSE, ac_GearIn_Drive1= TRUE.

Die reale Achse folgt der virtuellen weiterhin.

6. Auflösen der Synchronität.

Zuschalten des "ex_GearOut_Drive1" (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 "Vcom_vAxis1".

Die veränderte Sollgeschwindigkeit wirkt sich nur noch auf die virtuelle Achse aus, "Vact_Drive1" bleibt unverändert.

8. Abschalten des "ex_GearOut_Drive1" durch "Werte schreiben".

do_GearOut_Drive1= FALSE

Erste Schritte – Motion

9. Stoppen der virtuellen Achse.
Zuschalten des "ex_Stop_vAxis1" (MC_Stop).
Die virtuelle Achse wird abgebremst auf Vact_vAxis1= 0 U/min.
10. Abschalten des "ex_Stop_vAxis1" (MC_Stop).
11. Stoppen der realen Achse.
Zuschalten des "ex_Stop_Drive1" (MC_Stop).
Die reale Achse wird abgebremst auf Vact_Drive1= 0 U/min. Die Achse wird nach Erreichen des Stillstandes aktiv in dieser Position gehalten.
12. Abschalten des "ex_Stop_Drive1" (MC_Stop).
Das aktive Halten der realen Achse ist beendet.
13. Leistungsabschaltung für die reale Achse.
Abschalten von "en_Power_Drive1" (MC_Power) durch "Werte schreiben".



Nur möglich im Zustand "Standstill" nach Ausführung des Stopps an "Drive1".

3.5.8 Aufzeichnung der Bewegung beider Achsen (Trace), ST

Das Trace-Werkzeug wurde analog zum Abschnitt [Kap. 3.5.4 "Aufzeichnung der Bewegung von Achse vAxis1 \(Trace\), ST" auf Seite 131](#), 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_Power_Drive1" (MC_Power).
2. Start der virtuellen Achse.
Zuschalten des "ex_MoveVel_vAxis1" (MC_MoveVelocity).
3. Nachführen der realen Achse.
Zuschalten des "ex_GearIn_Drive1" (MC_GearIn).
4. Drehzahlsollwertsprung der virtuellen Achse.
Erhöhen des Drehzahlsollwertes auf 250.
Wirksamschalten durch "ex_MoveVel_vAxis1" TRUE ⇒ FALSE ⇒ TRUE.

Die nachstehende Abbildung zeigt das Trace-Ergebnis:

Erste Schritte – Motion

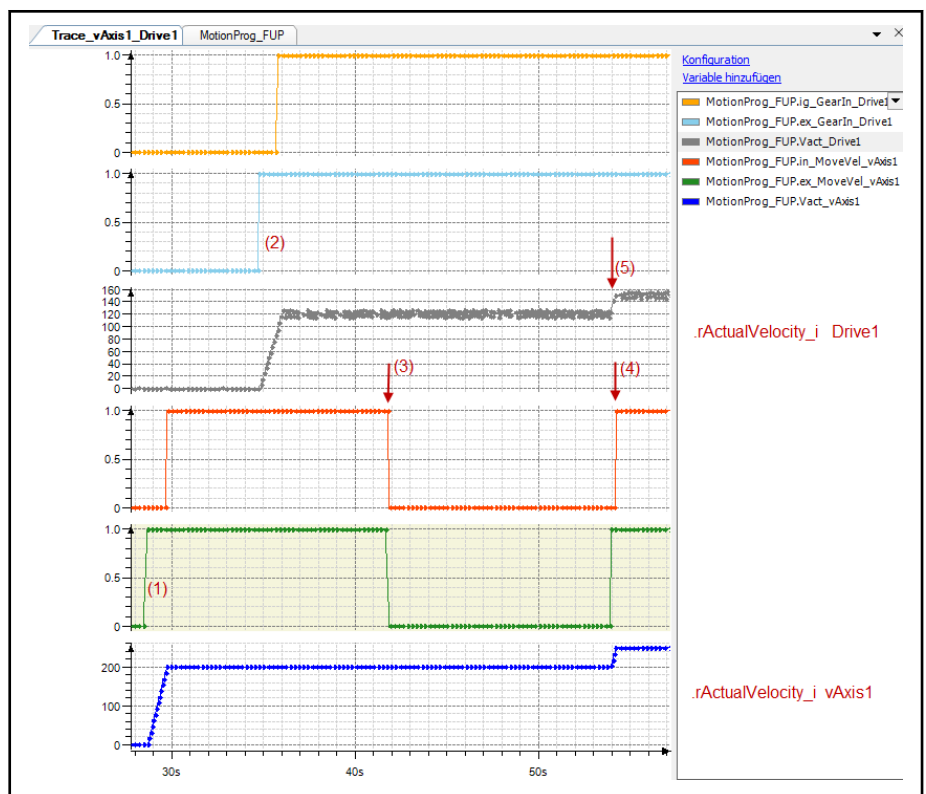


Abb. 3-76: Traceaufzeichnung für beide Achsen

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_MoveVel_vAxis1" TRUE ⇒ FALSE
- (4) - Wirksamwerden der Solldrehzahl der virtuellen Achse, "ex_MoveVel_vAxis1" FALSE ⇒ TRUE
- (5) - Sprungantwort an der realen Achse

3.6 Projekt wiederherstellen

Ablage Ausgehend vom erstellten Projekt wurde eine Exportdatei erzeugt, die unabhängig von Ihrer Steuerungshardware (IndraMotion MLC L25, L45, L65, L75, XM2, VPx) und der Antriebstechnik erlaubt, Teile des Projektes wiederherzustellen.

Die Exportdatei wurde für den Knoten "Application" erstellt und enthält folgende Bestandteile:

Erste Schritte – Motion

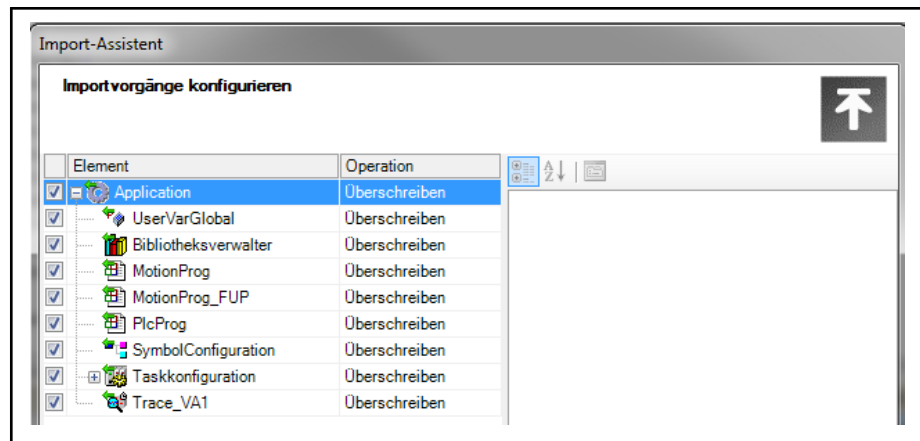


Abb. 3-77: Bestandteile der Exportdatei

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 (IndraMotion MLC L25, L45, L65, L75, XM2, VPx) lauffähig zu machen, gehen Sie wie folgt vor:

1. Löschen von Altdaten in Ihrer Steuerung.
Löschen Sie den alten Speicherinhalt Ihrer Steuerung.
Details sind unter [Kap. 3.3.1 "Steuerung zurücksetzen \(CMLx5\)" auf Seite 80](#) 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 nun über ein neues leeres Projekt.
3. Anlegen der Steuerung.
Ziehen Sie eine Steuerung Ihrer Wahl (IndraMotion MLC L25, L45, L65, L75, XM2, VPx) aus der Gerätebibliothek auf das leere Projekt. Der Steuerungsname ist beliebig, empfohlen wird "Mlc1".
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 nun 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, der Antrieb ist umzubenennen und erhält den Namen "Drive1".
5. Vereinbaren Sie eine virtuelle Achse, die Achse erhält die Achsnummer "2" und den Namen "vAxis1".

In der Globalen Variablenliste "MlcVarGlobal" werden die Achsnummern und Achsnamen als Achsreferenzen vom Typ "AXIS_REF" zur Verfügung gestellt.

Der Download des Motion-Teilobjekts kann durchgeführt werden (siehe [Kap. 3.3.8 "Download des Motion-Teilprojektes" auf Seite 91](#)).

6. Import der Exportdatei "Application_FirstSteps_IndraLogic_Motion".
Öffnen Sie im Projektnavigator den Steuerungsknoten bis **Logic ► Application**.

Wählen Sie im Kontextmenü des "Application"-Knotens **Importieren...** aus.

Wählen Sie die Exportdatei "Application_FirstSteps_IndraLogic_Motion.iwx" in "<Laufwerk>:\Programme\Rexroth\IndraWorks\Projects". Es öffnet sich der Dialog "Importvorgänge konfigurieren". Wählen Sie in diesem Import-Dialog für jedes Element die Operation "Überschreiben" aus. Starten Sie den Importvorgang mit **Fertigstellen**.

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

4 Erste Schritte – FlexProfile

4.1 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.

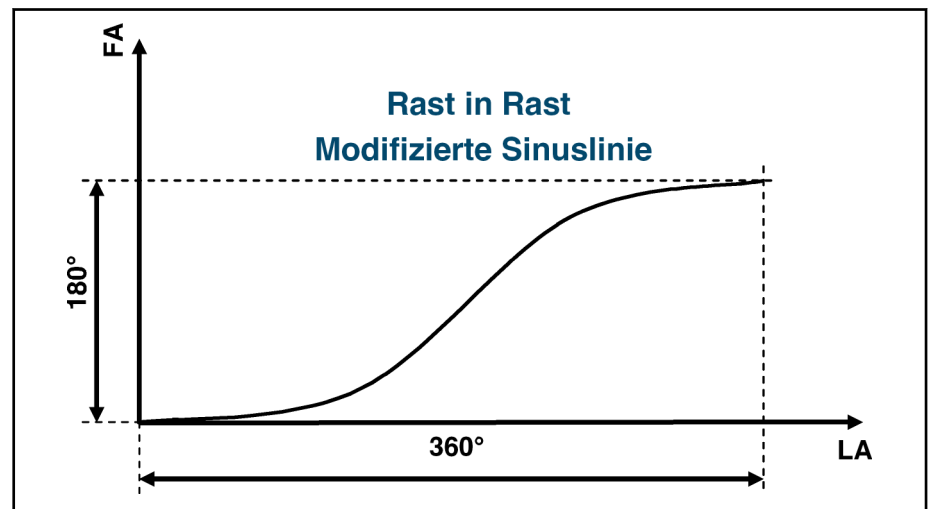


Abb. 4-1: Kopplungsprofil zwischen der Leitachse (vAxis1) und Folgeachse (Drive1)

Der Folgeachse können dabei bis zu vier Bewegungsprofile mit jeweils maximal 16 (64) Schritten zugeordnet werden.¹⁾

Wir nutzen für die "Ersten Schritte – FlexProfile" das FlexProfile 0 in "Drive1".

Ein FlexProfile kann auf zwei unterschiedlichen Wegen erstellt werden:

- IndraWorks, Dialog "FlexProfile" der Folgeachse, siehe [Kap. 4.4.2 "Erstellen des FlexProfiles über IndraWorks-Dialoge" auf Seite 150](#)
oder
- IndraLogic, SPS-Programmcode mit einem Funktionsbaustein "MB_ChangeFlexProfileSet" zur Datenübergabe vom SPS-Programm an die Folgeachse, siehe [Kap. 4.4.3 "Erstellen des FlexProfiles über die SPS" auf Seite 152](#)
 - Programmierung in ST siehe ["FlexProfile, Datenübergabe in ST" auf Seite 153](#)
 - Programmierung in FUP siehe ["FlexProfile, Datenübergabe in FUP" auf Seite 154](#)

Die eigentliche Abarbeitung wird über den Baustein "ML_FlexProfile" gestartet ("ML_FlexProfile" übernimmt damit die Aufgabe (Anbindung der Folgeach-

¹⁾ Die Anzahl der Schritte innerhalb eines FlexProfiles hängt von der Firmwareversion und vom Steuerungstyp ab.

Erste Schritte – FlexProfile

se an die Leitachse), die "MC_GearIn" in den "Ersten Schritte – Motion" erfüllt hat.

Der Baustein "ML_FlexProfile" wird durch den Baustein "MC_GearOut" deaktiviert.

- Programmierung in ST siehe ["ML_FlexProfile in ST" auf Seite 156](#)
- Programmierung in FUP siehe ["ML_FlexProfile in FUP" auf Seite 158](#)

4.2 Technische Voraussetzungen

- Software:
 - PC mit installiertem IndraWorks Engineering für IndraMotion MLC 14VRS
- Steuerung:
 - IndraMotion MLC L25
 - IndraMotion MLC L45
 - IndraMotion MLC L65
 - IndraMotion MLC L75
 - IndraMotion MLC XM2
 - IndraMotion MLC VPx

Eine Steuerung, z. B. Typ "**IndraMotion MLC L65**" mit aktueller Firmware 14VRS, muss vorhanden sein. Zusätzlich müssen noch folgende Voraussetzungen erfüllt sein:

- Die IP-Adresse ist bekannt
- Die Steuerung ist über Ethernet mit dem Programmiergerät (PC) verbunden

Antriebsregelgerät, z. B. Typ "**IndraDrive M**" mit aktueller Firmware:

- Das Antriebsregelgerät ist mit der Steuerung über Sercos III verbunden
- Die Adresse des Antriebsregelgerätes am Sercos Bus wird gescannt oder
- ein gebrückter Sercos Bus, wenn kein Antrieb vorhanden ist



Weitere Hinweise zu FlexProfile siehe Dokumentation
IndraMotion MLC 14VRS Funktionsbeschreibung, Kapitel
"Flexprofile".

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 "vAxis1" und als Folgeachse die reale Achse "Drive1", IndraDrive **mit Interpolation in der Steuerung**.

4.3 Anlegen des Motion-Teilprojektes

Um das Projekt auf einer Steuerung (IndraMotion MLC L25, L45, L65, L75, XM2, VPx) lauffähig zu machen, gehen Sie wie folgt vor:

Erste Schritte – FlexProfile

1. Löschen von Altdaten in Ihrer Steuerung.

Löschen Sie den alten Speicherinhalt Ihrer Steuerung.

Details sind in [Kap. 3.3.1 "Steuerung zurücksetzen \(CMLx5\)"](#) auf Seite 80 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 nun über ein neues leeres Projekt.

3. Anlegen der Steuerung.

Ziehen Sie eine Steuerung Ihrer Wahl (IndraMotion MLC L25, L45, L65, L75, XM2, VPx) aus der Gerätebibliothek auf das leere Projekt. Der Steuerungsname ist beliebig, empfohlen wird "Mlc1".

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 nun die gewünschte Steuerung.

4. Anlegen des Motion-Teilprojektes.

Scannen Sie den Sercos Bus (über Kontextmenü "Sercos" Knoten **Buskonfiguration scannen**).



Kontrollieren Sie, dass in Spalte "Ipo Antrieb" kein Haken gesetzt ist. Ein eventuell gesetzter Haken ist zu entfernen!

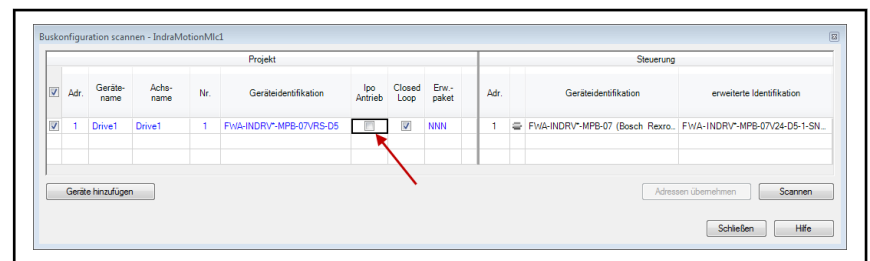


Abb. 4-2: Buskonfiguration scannen

Dem Antrieb wird die Achsnummer "1" zugewiesen.

5. Vereinbaren Sie eine virtuelle Achse. Weisen Sie der Achse die Achsnummer "2" zu.

In der Globalen Variablenliste "MlcVarGlobal" werden die Achsnummern und Achsnamen als Achsreferenzen vom Typ "AXIS_REF" zur Verfügung gestellt.

Der Download (siehe [Kap. 3.3.8 "Download des Motion-Teilprojektes"](#) auf Seite 91) kann stattfinden (die Motion-Komponente befindet sich dann im Zustand "Betriebsbereit (BB)").

6. Import der Exportdatei "Application_FirstSteps_IndraLogic_Motion.iwx".

Öffnen Sie im Projektnavigator den Steuerungsknoten bis **Logic ► Application**.

Erste Schritte – FlexProfile

Wählen Sie im Kontextmenü des "Application"-Knotens **Importieren...** aus.

Wählen Sie die Exportdatei "Application_FirstSteps_IndraLogic_Motion.iwx" in "<Laufwerk>:\Programme\Rexroth\IndraWorks\Projects" aus. Wählen Sie in diesem Import-Dialog für jedes Element die Operation "Überschreiben" aus. Starten Sie den Importvorgang mit **Fertigstellen**.

Die benötigten Dateien werden kopiert, das importierte Projekt ist auf der Steuerung lauffähig.

4.4 Erstellen eines FlexProfiles

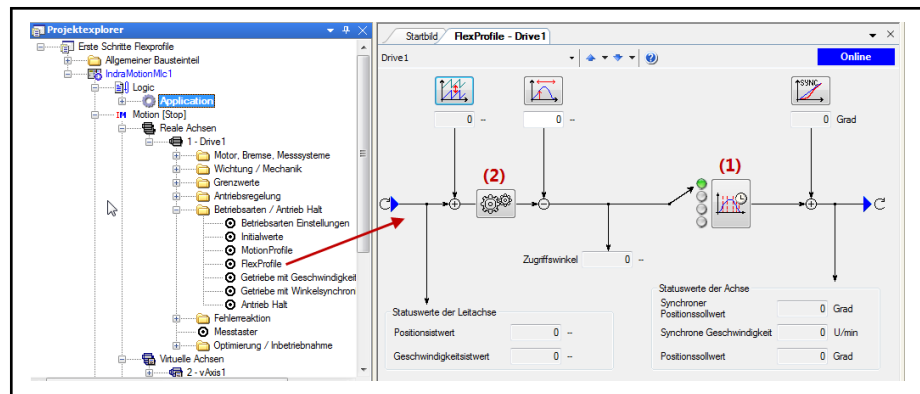
4.4.1 Erstellen eines FlexProfiles, Allgemeines



Das Erstellen oder Modifizieren eines FlexProfiles setzt voraus, dass Steuerung und PC **online** gekoppelt sind.

4.4.2 Erstellen des FlexProfiles über IndraWorks-Dialoge

Durch Anklicken des Eintrags "FlexProfile" im Projektextplorer unter "Reale Achsen" → "Drive1" öffnet sich ein Dialog, in dem die allgemeine Profilkonfiguration vorgenommen werden kann.



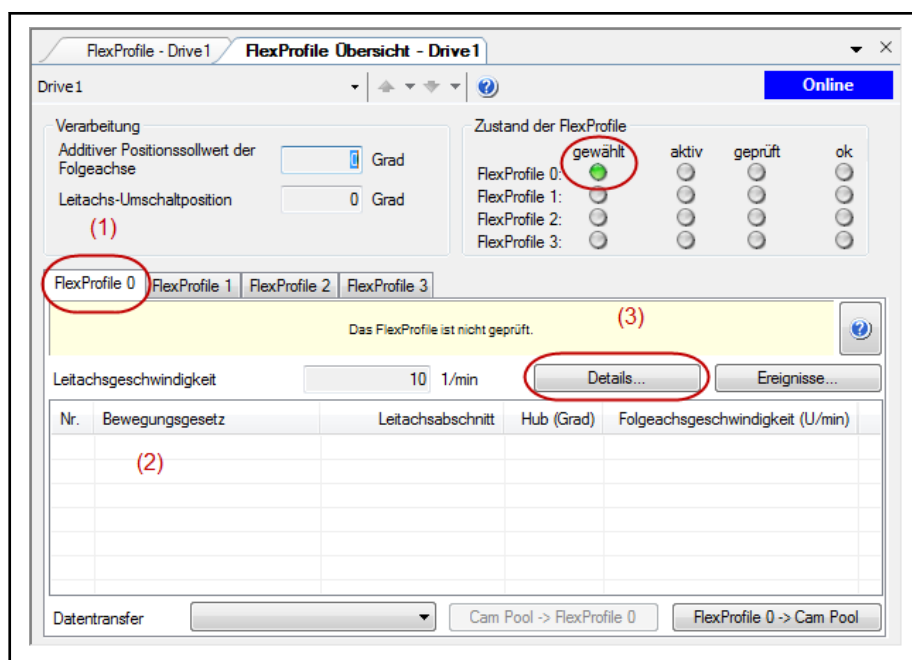
- (1) Übergang zu den Dialogfenstern "FlexProfile 0"... "FlexProfile 3"
- (2) Übergang zum Dialogfenster "Getriebeeinstellung"

Abb. 4-3: 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 FlexProfiles (Register "FlexProfile 0" bis "FlexProfile 3").

Erste Schritte – FlexProfile



- (1) Register "FlexProfile 0" bis "FlexProfile 3", ausgewählt ist "FlexProfile 0"
- (2) Übersicht über die Bewegungsschritte 0...15 (max. 64 Bewegungsschritte) des gewählten FlexProfiles (schreibgeschützt)
- (3) Übergang zu den Details des gewählten FlexProfiles

Abb. 4-4: Dialog: "FlexProfile" – Übersichtsfenster, gewählt wird FlexProfile 0

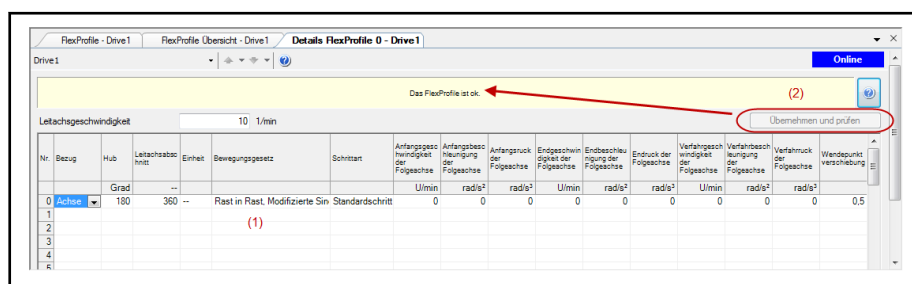
Die Erstellung eines neuen Bewegungsprofils kann über "Details" (1) vorgenommen werden.

Dazu wird automatisch der FlexProfile-Editor geöffnet.

Es wird eine leere Konfigurationstabelle gezeigt, in der jede Zeile einen Profilschritt (0..15 (max. 64 Bewegungsschritte) definiert.

Die dazu benötigten Leitachsquellen, Bewegungsgesetze, Bewegungsschritt-Typen können aus einem Dropdown-Menü ausgewählt werden.

Tragen Sie ihre Daten, wie auf unten gezeigtem Bild ein, und konfigurieren Sie so den gewünschten Bewegungsverlauf.



- (1) Konfiguration der einzelnen Bewegungsschritte 0...15 (max. 64 Bewegungsschritte) des gewählten FlexProfiles
- (2) Übernehmen der Daten des Bewegungsprofils und Prüfung, ob das Profil korrekt ist

Abb. 4-5: Dialog: "FlexProfile" – Konfigurationsfenster

Für das FlexProfile 0 wurden in obigem Dialogfenster nachstehende Daten eingetragen:

Erste Schritte – FlexProfile

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. Die Übertragung wird durch die Betätigung von **Übernehmen und Prüfen** (2) ausgelöst.



Ein Bewegungsprofil kann nur gestartet werden, wenn das Bewegungsprofil 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.

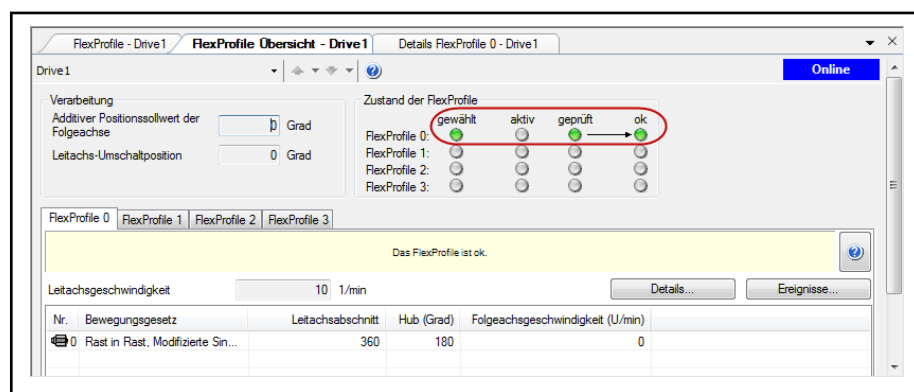


Abb. 4-6: Dialog: "FlexProfile" – Übersichtsfenster, FlexProfile 0, geprüft → ok

Das so konfigurierte Bewegungsprofil "FlexProfile 0" kann nun mit dem SPS-Baustein "ML_FlexProfile" gestartet werden.

4.4.3 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 ["FlexProfile, Datenübergabe in ST"](#) auf Seite 153 bzw. ["FlexProfile, Datenübergabe in FUP"](#) auf Seite 154. 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 (max. 64 Bewegungsschritte) 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.

Dieser Datensatz entspricht [Abb. 4-5 "Dialog: FlexProfile – Konfigurationsfenster"](#) auf Seite 151.

2. Übergabe der Daten des FlexProfiles aus dem SPS-Programm an die Steuerung (Achse Drive1).

Diese Aufgabe übernimmt die Instanz "ChangeFlexProfileSetDrive1" vom Typ "MB_ChangeFlexProfileSet".

Diese Instanz entspricht [Abb. 4-6 "Dialog: FlexProfile – Übersichtsfenster, FlexProfile 0, geprüft → ok"](#) auf Seite 152.

FlexProfile, Datenübergabe in ST

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 ▶ Data Types ▶ Flex Structs ▶ MB_PROFILE_STEP**).

Der Name der Bibliothek wird automatisch entsprechend der im Projektexplorer angelegten Steuerung ergänzt, im nachstehenden Bild "ml_base_mlc".

Name	Datentyp	Geerbt von	Adresse	Initialwert	Kommentar
Master	MB_MASTER_TYPE		MASTER_AXIS0		selection of master axis sources: time or position
Hub	LREAL				stroke
Range	LREAL				definition of master axis section
MotionLaw	MC_STEP_MODE				definition of motion law
StepType	MB_STEP_TYPE		STEP_FIX		step type: standard (fixed) or Flex
StartVel	LREAL			0.0	slave axis start velocity
StartAcc	LREAL			0.0	slave axis start acceleration
StartJrk	LREAL			0.0	slave axis start jerk
EndVel	LREAL			0.0	slave axis final velocity
EndAcc	LREAL			0.0	slave axis final acceleration
EndJrk	LREAL			0.0	slave axis final jerk
TravelVel	LREAL			0.0	slave axis traveling velocity, not taken into account by all motion laws
LimitAcc	LREAL			0.0	slave axis traveling acceleration, not taken into account by all motion laws
LimitJrk	LREAL			0.0	slave axis traveling jerk not taken into account by all motion laws
TurningPointDisplacement	LREAL			0.5	turning point displacement for symmetric motion laws

Abb. 4-7: Beschreibung der Struktur "MB_PROFILE_STEP" im Bibliotheksverwalter

Erste Schritte – FlexProfile

Nutzen Sie zur Eingabe den "."-Operator mit nachfolgender Intellisense-Funktion (automatische Vervollständigung).

```

1 PROGRAM PlcProg
2 VAR
3   profileAxis: ARRAY[0..15] OF MB_PROFILE_STEP;
4 END_VAR

1 profileAxis[0].Master:= MB_MASTER_TYPE.MASTER_AXIS0;
2 profileAxis[0].Hub:=180;
3 profileAxis[0].Range:=360;
4 profileAxis[0].MotionLaw:=MC_STEP_MODE.REST_IN_REST_MOD_SINE;
5 profileAxis[0].StepType:=MB_STEP_TYPE.STEP_FIX;
6 profileAxis[0].StartVel:=0;
7 profileAxis[0].StartAcc:=0;
8 profileAxis[0].StartJrk:=0;
9 profileAxis[0].EndVel:=0;
10 profileAxis[0].EndAcc:=0;
11 profileAxis[0].EndJrk:=0;
12 profileAxis[0].TravelVel:=0;
13 profileAxis[0].LimitAcc:=0;
14 profileAxis[0].LimitJrk:=0;
15 profileAxis[0].TurningPointDisplacement:=0.5;
16

```

Abb. 4-8: Datensatz "profileAxis" mit Wertezuordnung in ST

Ab Zeile 17 wird in gewohnter Weise die Instanz "ChangeFlexProfileSetDrive1" des Funktionsbausteins "MB_ChangeFlexProfileSet" zur Übergabe des Datenfeldes an die Steuerung eingegeben.

```

13 profileAxis[0].LimitAcc:=0;
14 profileAxis[0].LimitJrk:=0;
15 profileAxis[0].TurningPointDisplacement:=0.5;
16
17 ChangeFlexProfileSetDrive1(
18   Execute:= ex_ChangeFlexProfileSetDrive1,
19   SetNumber:= SetNum,
20   NumberOfSteps:= 1,
21   MasterAxisVelocity:= 10,
22   ProfileStepAdr:= ADR(profileAxis),
23   Done=> do_ChangeFlexProfileSetDrive1,
24   Active=> ac_ChangeFlexProfileSetDrive1,
25   Error=> er_ChangeFlexProfileSetDrive1,
26   ErrorID=> ID_ChangeFlexProfileSetDrive1,
27   ErrorIdent=> Ident_ChangeFlexProfileSetDrive1,
28   Axis:=Drive1);

```

Abb. 4-9: Funktionsbaustein zur Übergabe des Datenfeldes an die Steuerung (SetNum = 0 für FlexProfile 0)

Die Übergabe des Datensatzes erfolgt durch "Werte schreiben" am Eingang "ex_ChangeFlexProfileSetDrive1", 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, im FlexProfile-Editor vergleichen.

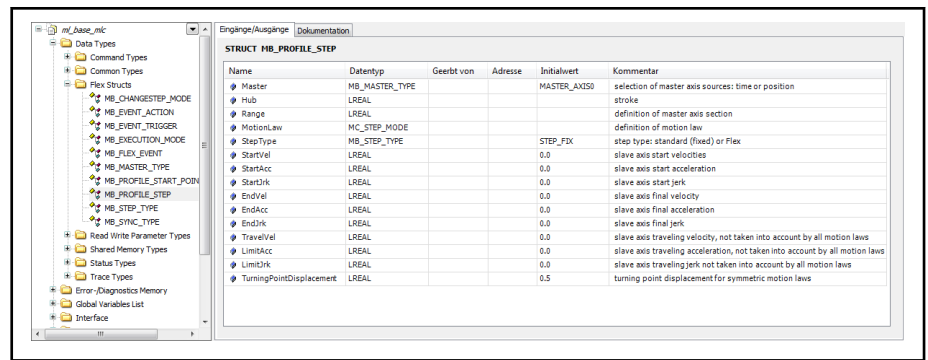
FlexProfile, Datenübergabe in FUP

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 ► Data Types ► Flex Structs ► MB_PROFILE_STEP).

Erste Schritte – FlexProfile



Name	Datentyp	Geerbt von	Adresse	Initialwert	Kommentar
Master	MB_MASTER_TYPE			MASTER_AXIS0	selection of master axis sources: time or position
Hub	LREAL				stroke
Range	LREAL				definition of master axis section
MotionLaw	MC_STEP_MODE				definition of motion law
StepType	MB_STEP_TYPE			STEP_FIX	step type: standard (fixed) or Flex
StartVel	LREAL			0.0	slave axis start velocities
StartAcc	LREAL			0.0	slave axis start acceleration
StartJrk	LREAL			0.0	slave axis start jerk
EndVel	LREAL			0.0	slave axis final velocity
EndAcc	LREAL			0.0	slave axis final acceleration
EndJrk	LREAL			0.0	slave axis final jerk
TravelVel	LREAL			0.0	slave axis traveling velocity, not taken into account by all motion laws
LimitAcc	LREAL			0.0	slave axis traveling acceleration, not taken into account by all motion laws
LimitJrk	LREAL			0.0	slave axis traveling jerk not taken into account by all motion laws
TurningPointDisplacement	LREAL			0.5	turning point displacement for symmetric motion laws

Abb. 4-10: Beschreibung der Struktur "MB_PROFILE_STEP" im Bibliotheksverwalter

Nutzen Sie zur Eingabe der Zuweisungen den "."-Operator mit nachfolgender Intellisense-Funktion (automatische Vervollständigung).

```

1  PROGRAM PlcProg_FUP
2  VAR
3  profileAxis: ARRAY[0..15] OF MB_PROFILE_STEP;
4  END_VAR

1  MB_MASTER_TYPE.MASTER_AXIS0 — profileAxis[0].Master
2  180 — profileAxis[0].Hub
3  360 — profileAxis[0].Range
4  MC_STEP_MODE.REST_IN_REST_... — profileAxis[0].MotionLaw
5  MB_STEP_TYPE.STEP_FIX — profileAxis[0].StepType
6  0 — profileAxis[0].StartVel
7  0 — profileAxis[0].StartAcc
8  0 — profileAxis[0].StartJrk
9  0 — profileAxis[0].EndVel
10 0 — profileAxis[0].EndAcc
11 0 — profileAxis[0].EndJrk
12 0 — profileAxis[0].TravelVel
13 0 — profileAxis[0].LimitAcc
14 0 — profileAxis[0].LimitJrk
15 0.5 — profileAxis[0].TurningPoin...

```

Abb. 4-11: Datensatz "profileAxis" mit Wertezuordnung in FUP

Als Netzwerk Nummer 16 wird in gewohnter Weise die Instanz "ChangeFlexProfileSetDrive1" des Funktionsbausteins "MB_ChangeFlexProfileSet" zur Übergabe des Datenfeldes an die Steuerung eingegeben.

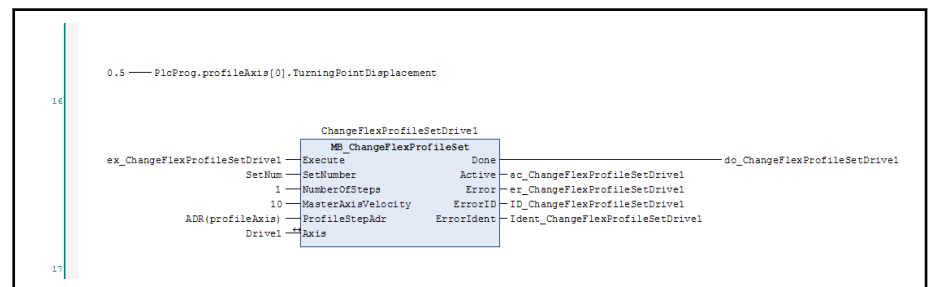


Abb. 4-12: Funktionsbaustein zur Übergabe des Datenfeldes an die Steuerung (SetNum = 0 für FlexProfile 0)

Die Übergabe des Datensatzes erfolgt durch "Werte schreiben" am Eingang "ex_ChangeFlexProfileSetDrive1", 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, im FlexProfile-Editor vergleichen.

4.5 Komplettierung 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 Funktionsbaustein stellt die 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 die Exportdatei "Application_FirstSteps_IndraLogic_Motion.iwx" aus dem Verzeichnis "<Laufwerk>:\Programme\Rexroth\IndraWorks\Projects".

Öffnen Sie zum Importieren im Projektnavigator den Steuerungsknoten bis **Logic ▶ Application** und wählen Sie im Kontextmenü des "Application"-Knotens **Importieren...** aus.

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 **VcomvAxis1** von 200 U/min auf 10 U/min.
2. **MoveVelvAxis1:**
 "Acceleration" → 100 [rad/s²]
 "Deceleration" → 100 [rad/s²]
3. Löschen Sie die Instanz "MC_GearIn_Drive1" von "MC_GearIn". Diese Instanz wird durch die Instanz "FlexProfileDrive1" 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 "FlexProfileDrive1" deklarieren:

Deklarationsteil von "MotionProg":

1. Leere Zeile vor END_VAR einfügen.
2. FlexProfileDrive1: eingeben, mit <F2> die Eingabehilfe öffnen und unter **benutzerdefinierte Typen**
ML_PLCOpen ▶ POU ▶ Motion ▶ ML_FlexProfile
 die Option "Mit Argumenten einfügen" abwählen.
3. Zeile komplettieren:

```
FlexProfileDrive1: ML_FlexProfile;
```

Leere Zeile anstelle von "GearInDrive1" anfügen

Mit <F2> die Eingabehilfe öffnen und unter **Instanzzaufrufe** ► **FlexProfileDrive1** die Option "Mit Argumenten einfügen" markieren

Eingänge beschalten und wenn nötig Eingangsvariablen deklarieren:

"Execute" → "ex_FlexProfileDrive1" 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" → "vAxis1" vom Typ "AXIS_REF"

(bereits deklarierte Variable aus dem Bereich der "Variablen", Application, MlcVarGlobal)

"Slave" → "Drive1" vom Typ "AXIS_REF"

(bereits deklarierte Variable aus dem Bereich der "Variablen", Application, MlcVarGlobal)

Ausgänge:

"InSync" → "is_FlexProfileDrive1" vom Typ "BOOL"

"Active" → "ac_FlexProfileDrive1" vom Typ "BOOL"

"CommandAborted" → "ca_FlexProfileDrive1" vom Typ "BOOL"

"Error" → "er_FlexProfileDrive1" vom Typ "BOOL"

"ErrorID" → "ID_FlexProfileDrive1" vom Typ "ERROR_CODE"

"ErrorIdent" → "Ident_FlexProfileDrive1" vom Typ "ERROR_STRUCT"

Erste Schritte – FlexProfile

```

33   Axis:= MlcVarGlobal.Drive1);
34
35   FlexProfileDrive1(
36     Execute:= ex_FlexProfileDrive1,
37     RatioNumerator:= 1,
38     RatioDenominator:= 1,
39     MasterFineadjust:= 0,
40     SetSelection:= 0,
41     UseSwitchingPosition:= FALSE,
42     SwitchingPosition:= 0,
43     ProfileEntry:= MB_PROFILE_START_POINT.SLAVE_CURRENT_MASTER_CURRENT,
44     MasterOffset:= 0,
45     SlaveOffset:= 0,
46     SyncType:= MB_SYNC_TYPE.SYNC_RAMPIN_CATCH_UP,
47     SyncVelocity:= 100,
48     SyncAcceleration:= 100,
49     ExecutionMode:= MB_EXECUTION_MODE.EXECUTE_CYCLIC,
50     InSync=> is_FlexProfileDrive1,
51     Active=> ac_FlexProfileDrive1,
52     CommandAborted=> ca_FlexProfileDrive1,
53     Error=> er_FlexProfileDrive1,
54     ErrorID=> ID_FlexProfileDrive1,
55     ErrorIdent=> Ident_FlexProfileDrive1,
56     Master:=vAxis1,
57     Slave:= Drive1);
58
59   MC_GearOut_Drive1(
60     Execute:= ex_GearOut_Drive1,
61     Done=> do_GearOut_Drive1,

```

Abb. 4-13: "ML_FlexProfile" ist eingefügt und beschaltet (ST)

"ML_FlexProfile" in FUP

Fügen Sie analog zu obiger Beschreibung einen Baustein des Typs "ML_FlexProfile" ein und beschalten Sie den Funktionsbaustein, wie nachfolgend beschrieben.

"ML_FlexProfile" einfügen und eine Instanz "FlexProfileDrive1" deklarieren:

Eingänge beschalten und wenn nötig Eingangsvariablen deklarieren:

"Execute" → "ex_FlexProfileDrive1" 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

Erste Schritte – FlexProfile

"Master" → "vAxis1" vom Typ "AXIS_REF"

(bereits deklarierte Variable aus dem Bereich der "Variablen", Application, MlcVarGlobal)

"Slave" → "Drive1" vom Typ "AXIS_REF"

(bereits deklarierte Variable aus dem Bereich der "Variablen", Application, MlcVarGlobal)

Ausgänge:

"InSync" → "is_FlexProfileDrive1" vom Typ "BOOL"

"Active" → "ac_FlexProfileDrive1" vom Typ "BOOL"

"CommandAborted" → "ca_FlexProfileDrive1" vom Typ "BOOL"

"Error" → "er_FlexProfileDrive1" vom Typ "BOOL"

"ErrorID" → "ID_FlexProfileDrive1" vom Typ "ERROR_CODE"

"ErrorIdent" → "Ident_FlexProfileDrive1" vom Typ "ERROR_STRUCT"

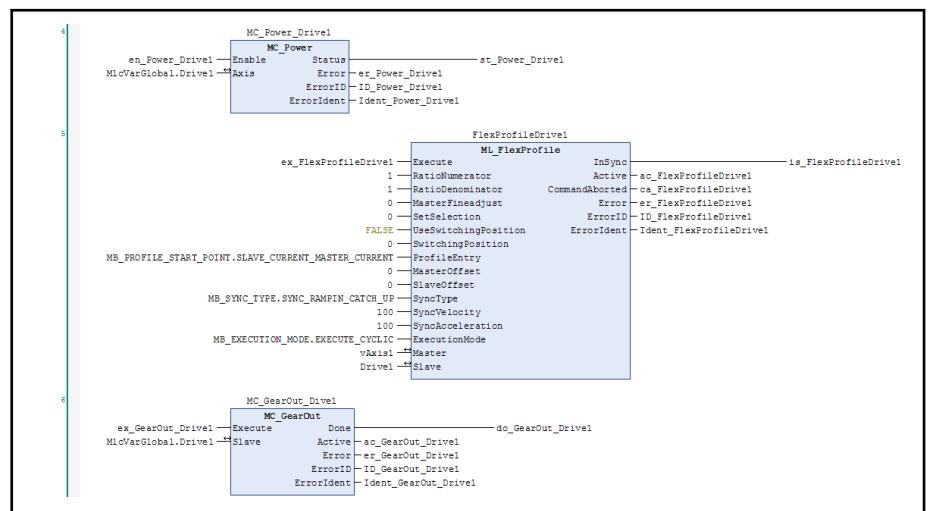


Abb. 4-14: "ML_FlexProfile" ist eingefügt und beschaltet (FUP)

4.6 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 IndraWorks-Menüpunkt

- **Erstellen ► Übersetzen**, Übersetzen des hinzu gekommenen Anteils der Applikation
- **Erstellen ► Code erzeugen**, der Befehl ermöglicht eine Code-Generierung zur Überprüfung von Übersetzungsfehlern.

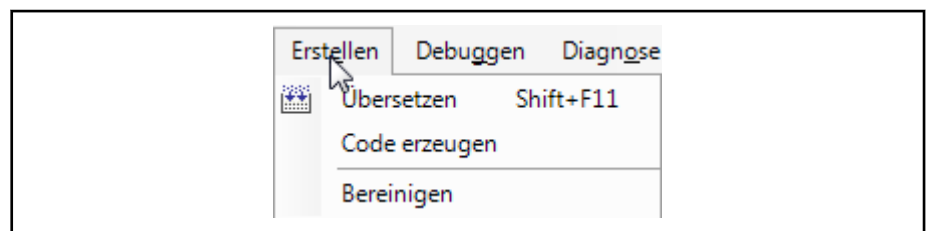


Abb. 4-15: Applikation übersetzen

Erste Schritte – FlexProfile



Die Menüpunkte **Erstellen ▶ Neu übersetzen** und **Erstellen ▶ Alles bereinigen** aus der 13VRS sind standardmäßig nicht mehr vorhanden. Diese Befehle können aber über **Extras ▶ Anpassen ▶ Befehle ▶ Erstellen** wieder hinzugefügt werden.

Zum Übertragen der kompilierten Daten in die Zielsteuerung wählen Sie in der IndraWorks-Menüleiste **Debuggen ▶ Einloggen** (siehe nachfolgende Abbildung).

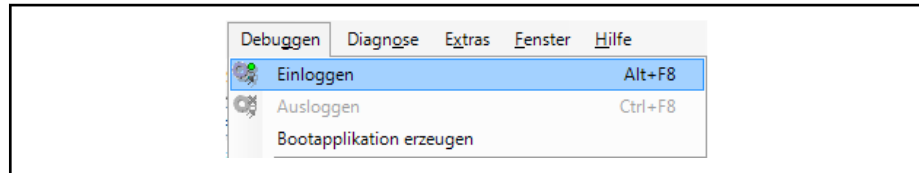


Abb. 4-16: Übertragen der kompilierten Daten in die Steuerung

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.

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

4.7 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_Power_Drive1" (MC_Power) durch "Werte schreiben".

st_Power_Drive1= TRUE (MC_Power).

2. Start der virtuellen Achse.

Zuschalten des "ex_MoveVel_vAxis1" (MC_MoveVelocity) durch "Werte schreiben".

Hochlauf bei Vact_vAxis1 = 10 U/min beendet.

3. Nachführen der realen Achse.

Zuschalten des "ex_FlexProfile_Drive1" (ML_FlexProfile) durch "Werte schreiben".

Das zu erwartenden Kopplungsverhalten ergibt sich zu:

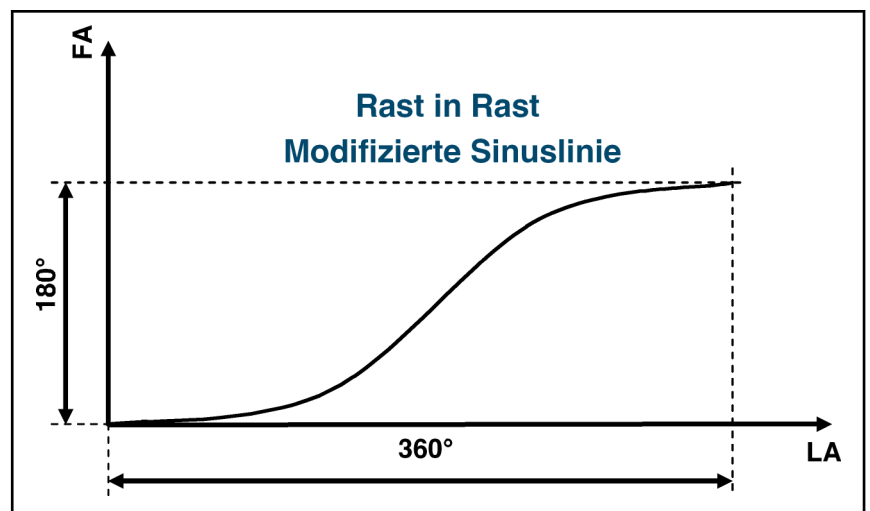


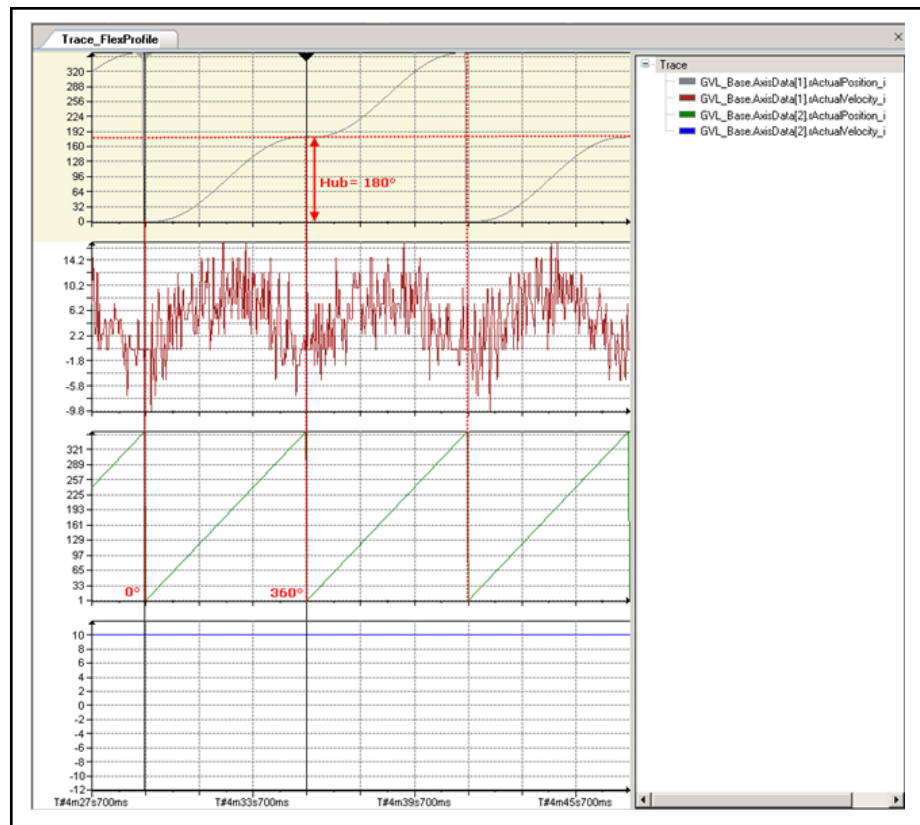
Abb. 4-17: Kopplungsprofil zwischen der Leitachse (vAxis1) und Folgeachse (Drive1)

4. Abschalten des "ex_FlexProfileDrive1" (ML_FlexProfile) durch "Werte schreiben".
 is_FlexProfileDrive1= FALSE, ac_FlexProfileDrive1= TRUE.
 Die reale Achse folgt der virtuellen weiterhin.
5. Auflösen der Synchronität.
 Zuschalten des "ex_GearOut_Drive1" (MC_GearOut) durch "Werte schreiben".
 Die Abhängigkeit der realen Achse von der virtuellen ist aufgehoben. Beide Achsen drehen jedoch mit unveränderter Geschwindigkeit weiter.
6. Änderung der Sollgeschwindigkeit "Vcom_vAxis1".
 Die veränderte Sollgeschwindigkeit wirkt sich nur noch auf die virtuelle Achse aus, "Vact_Drive1" bleibt unverändert.
7. Abschalten des "ex_GearOut_Drive1" durch "Werte schreiben" TRUE/FALSE.
 do_GearOut_Drive1= FALSE
 Die reale Achse bleibt stehen. Die Achse wird nach Erreichen des Stillstandes aktiv in dieser Position gehalten.
8. Stoppen der virtuellen Achse.
 Zuschalten des "ex_Stop_vAxis1" (MC_Stop)
 Die virtuelle Achse wird abgebremst auf Vact_vAxis1= 0 U/min.
9. Leistungsabschaltung für die reale Achse.
 Abschalten von "en_Power_Drive1" (MC_Power) durch "Werte schreiben".

4.8 Aufzeichnung der Bewegung beider Achsen (Trace)

In der Traceaufzeichnung, können die Bewegungen der Leit- und Folgeachse dargestellt werden. Es ergibt sich nachfolgender Verlauf.

Erste Schritte – FlexProfile



Grau Folgeachse (Drive1), Position
Rot Folgeachse (Drive1), Geschwindigkeit
Grün Leitachse (vAxis1), Position
Blau Leitachse (vAxis1), Geschwindigkeit

Abb. 4-18: Aufzeichnung der Bewegung beider Achsen

Die Traceaufzeichnung zeigt unten den Geschwindigkeitsverlauf der Leitachse.

Die Leitachse dreht sich mit einer konstanten Geschwindigkeit von 10 [U/min].

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.

4.9 Projekt wiederherstellen

Ausgehend vom erstellten Projekt wurde eine Exportdatei erzeugt, die unabhängig von Ihrer Steuerungshardware (IndraMotion MLC L25, L45, L65, L75, XM2, VPx) erlaubt, das Projekt wiederherzustellen. Das wiederhergestellte Projekt kann als Simulation oder unter Nutzung Ihrer Steuerung laufen.

Die Exportdatei wurde für den Knoten "Application" erstellt und enthält folgende Bestandteile:

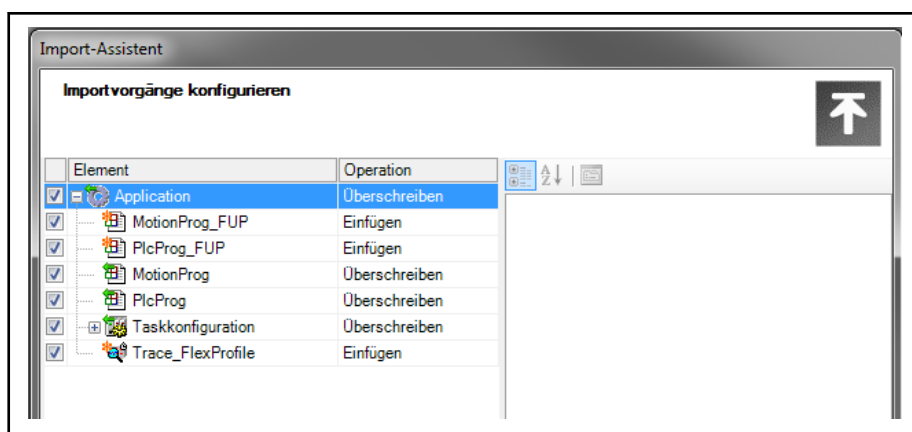


Abb. 4-19: Bestandteile der Exportdatei

Die angegebenen Komponenten werden beim Import überschrieben.

Die Exportdatei "Application_FirstSteps_IndraLogic_FlexProfile.iwx" befindet sich im Verzeichnis "<Laufwerk>:\Programme\Rexroth\IndraWorks\Projects".

5 Erste Schritte – Robot-Control V2

5.1 Ziel des Projekts "Erste Schritte – Robot-Control V2"

Anhand eines einfachen Beispiels soll die grundsätzliche Vorgehensweise beim Erstellen eines Robot-Control V2-Projekts mit IndraMotion MLC erläutert werden. Es wird vom Anlegen bis zum Ausführen eines Beispielprojektes stichpunktartig auf die hierfür minimal notwendige Konfiguration in IndraWorks und die Programmierung in der SPS eingegangen.

Im Beispiel-Projekt "RoCoV2TestV14" wird davon ausgegangen, dass nur virtuelle Achsen zur Verfügung stehen. Am Ende der Beschreibung wird kurz darauf eingegangen, wie reale Achsen eingebunden werden können.



Der Anwender kann entweder das Beispielprojekt von der IndraWorks Installations-DVD aus dem Verzeichnis "AddOns \MLC\Example-Project\RobotControl" wiederherstellen; das Beispielprojekt enthält eine ZIP-Datei mit folgenden Inhalt:

- die Parameter-Export-Datei (*.par)
- das SPS-Programm
- die Visualisierungsdaten

- oder -

der Anwender erstellt das Projekt Schritt für Schritt anhand der nachfolgenden Beschreibung. Dazu sind die im Anhang aufgeführten Detaildaten zur Unterstützung zu nutzen.

5.2 Erforderliche Hard- und Software

Erforderliche Hardware *Minimal:*

- Steuerung, Typ "IndraMotion MLC Lx5", "IndraMotion MLC XM2", "IndraMotion MLC VPx"
- 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

Erforderliche Software IndraWorks-Installation 14VRS mit IndraMotion MLC als Zielsystem.

5.3 Vorgehensweise beim Erstellen eines RoCo-Projektes

Für die Schritt-für-Schritt-Erstellung des Beispielprojektes sind folgende Schritte durchzuführen:

Allgemeine Vorgehensweise

1. IndraWorks starten und ein neues Projekt anlegen.

Nach dem Start von IndraWorks wird über **Datei ► Neu ► Projekt...** das gewünschte Projekt angelegt.

Erste Schritte – Robot-Control V2

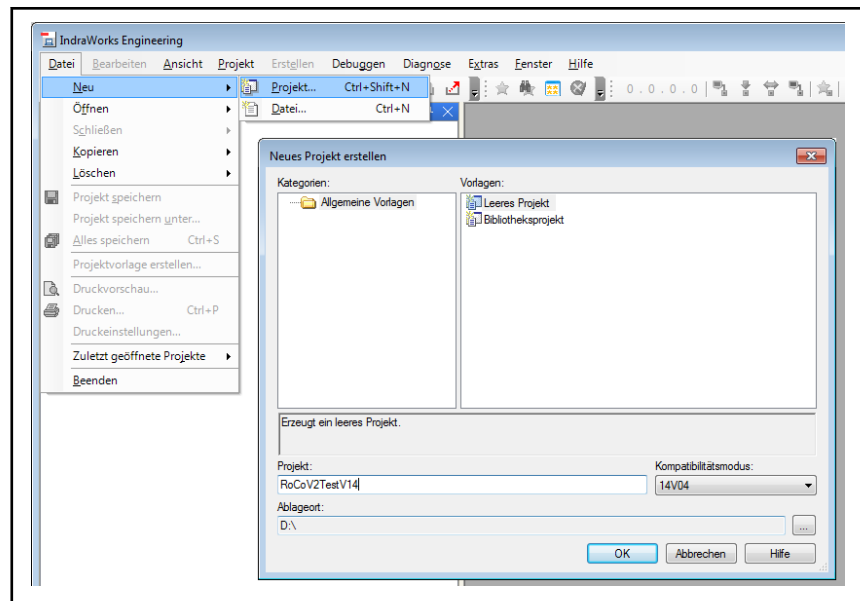


Abb. 5-1: Neues IndraWorks Projekt anlegen

Geben Sie dem Projekt einen beliebigen Namen (z. B. "Ro-coV2TestV14").

Wählen Sie einen Ablageort.

Mit der Auswahl der Projektsprache legen Sie die Sprache ihrer Anwendertexte fest.

Bestätigen Sie mit der Schaltfläche OK.

Das neu angelegte Projekt erscheint im Projektextplorer mit dem zuvor definierten Projektnamen.

2. IndraMotion MLC Steuerung anlegen und konfigurieren.

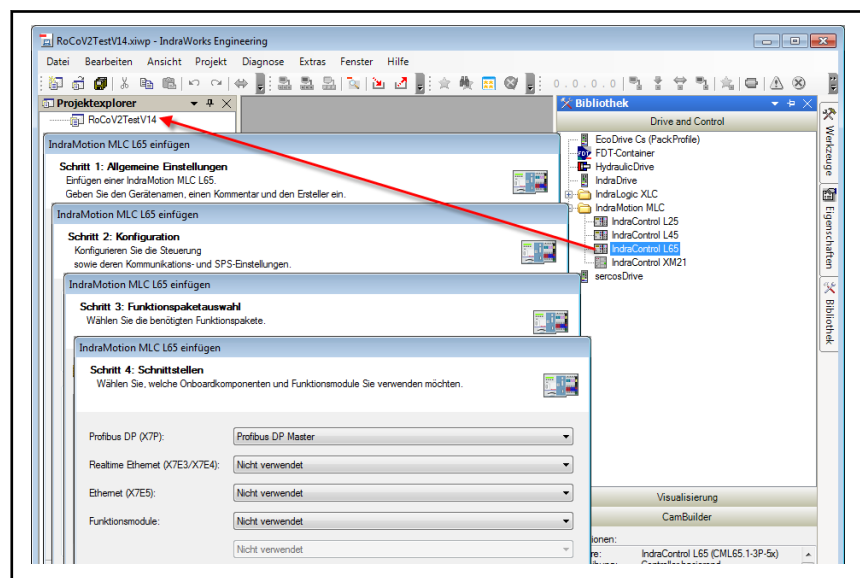


Abb. 5-2: IndraMotion MLC anlegen und konfigurieren

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

Erste Schritte – Robot-Control V2

Anschließend startet automatisch der Assistent zur Vergabe notwendiger Basiseinstellungen.

Geben Sie im Dialog "Allgemeine Einstellungen" einen beliebigen Gerätenamen an (z. B. "mlc_L65") und bestätigen Sie die Eingabe mit der Schaltfläche **Weiter**.

Definieren Sie im Dialog "Konfiguration" die Einstellungen zur Firmware und der Kommunikation zwischen IndraWorks und Ihrem Gerät.

Die Voreinstellungen für den "Gerätetyp", die "Firmware-Version" und das "Firmware-Release" können in der Regel so belassen werden.

Geben Sie unter "IP-Adresse" die IP-Adresse der Steuerung an, auf die Sie Ihre Applikation laden möchten.

Führen Sie den Verbindungstest aus und überprüfen Sie somit ob die Kommunikation zwischen PC und Steuerung erfolgreich ist.

Bestätigen Sie mit der Schaltfläche **Weiter**.

Sind keine Funktionsmodule an der Steuerung angeschlossen, sind im Dialog "Schnittstellen" keine weiteren Einstellungen notwendig.

Bestätigen Sie mit der Schaltfläche **Fertigstellen**.

3. Drei virtuelle Achsen anlegen und konfigurieren.

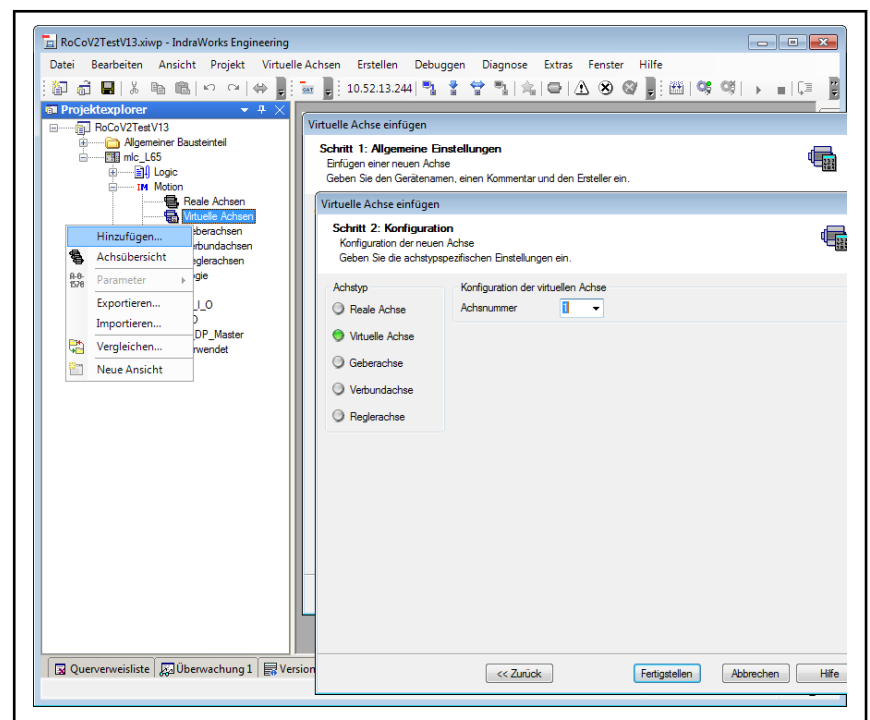


Abb. 5-3: Virtuelle Achsen anlegen

Klicken Sie mit der rechten Maustaste auf "Virtuelle Achsen" und wählen Sie im Kontextmenü "Hinzufügen" aus.

Geben Sie im Dialog "Allgemeine Einstellungen" unter "Gerätename" den Namen "Axis1" an und bestätigen Sie mit **Weiter**.

Geben Sie unter dem Dialog "Konfiguration" die Achsnummer "1" an und bestätigen Sie mit **Fertigstellen**.

Erste Schritte – Robot-Control V2

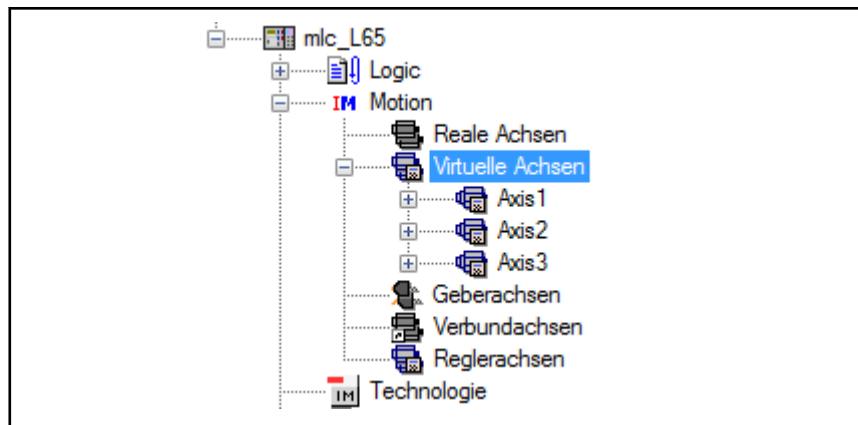


Abb. 5-4: Virtuelle Achsen angelegt

Führen Sie dies analog zur Erstellung zwei weiterer virtueller Achsen durch, sodass drei virtuelle Achsen im Knoten "Virtuelle Achsen" erscheinen.

4. Kinematik hinzufügen und konfigurieren.

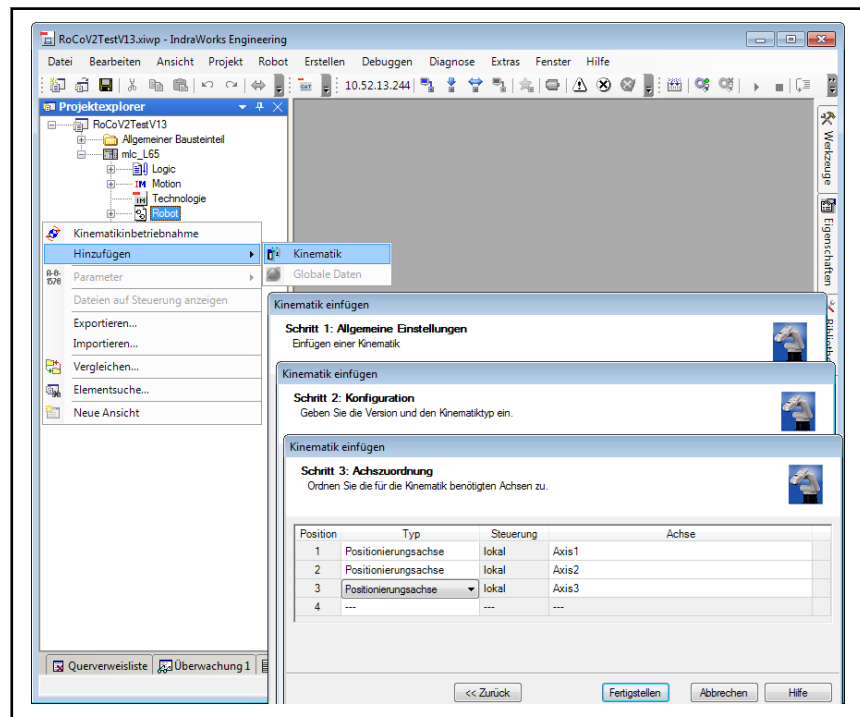


Abb. 5-5: Kinematik hinzufügen und konfigurieren

Klicken Sie mit der rechten Maustaste auf den "Robot"-Knoten und wählen Sie im Kontextmenü "Hinzufügen" und "Kinematik" aus.

Geben Sie im Dialog "Allgemeine Einstellungen" einen beliebigen "Gerätenamen" (z. B. "Kin01") ein und bestätigen Sie die Eingabe mit der Schaltfläche **Weiter**.

Wählen Sie im Dialogfeld "Konfiguration" unter Kinematik-Typ "Kartesisch" aus und bestätigen Sie mit der Schaltfläche **Weiter**.

Definieren Sie im Dialog "Achszuordnung" drei Positionierachsen und weisen Sie die virtuellen Achsen zu.

Schließen Sie den Vorgang mit der Schaltfläche **Fertigstellen** ab.

Erste Schritte – Robot-Control V2

5. Motion-Konfiguration auf das Gerät herunterladen.

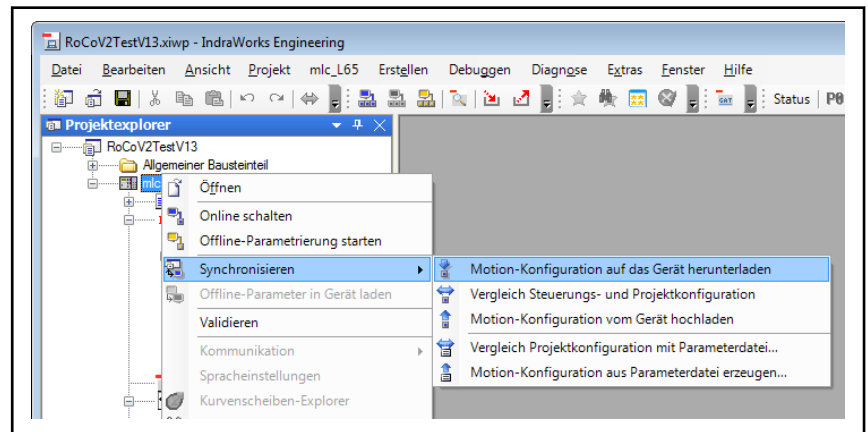


Abb. 5-6: Motionkonfiguration auf das Gerät herunterladen

Klicken Sie mit der rechten Maustaste auf den Steuerungsknoten und wählen Sie im Kontextmenü "Synchronisieren" und "Motion-Konfiguration auf das Gerät herunterladen" aus. Die bis hierher erfolgten Einstellungen werden in die Steuerung übertragen.

6. Parameter importieren.

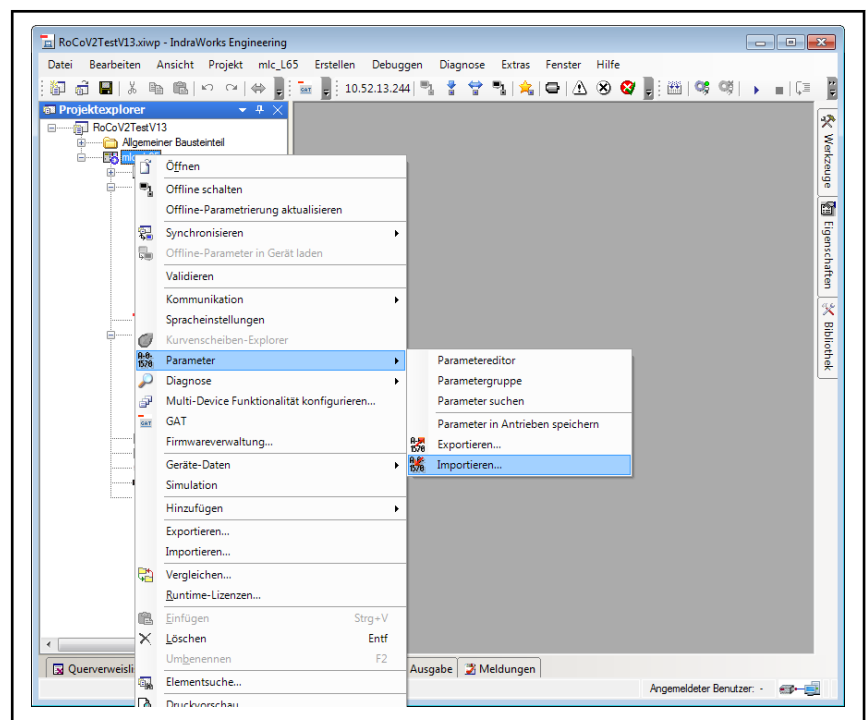


Abb. 5-7: Parameter importieren

Klicken Sie mit der rechten Maustaste auf den Steuerungsknoten und wählen Sie im Kontextmenü "Parameter" und "Importieren" aus.



Die Parameterdatei finden Sie im Projekt-Verzeichnis auf der Installations-DVD: "`\\AddOns\\MLC\\Example-Project\\RobotControl\\RoCoV2TestV14\\RoCoV2TestV14virtuell.par`".

Erste Schritte – Robot-Control V2

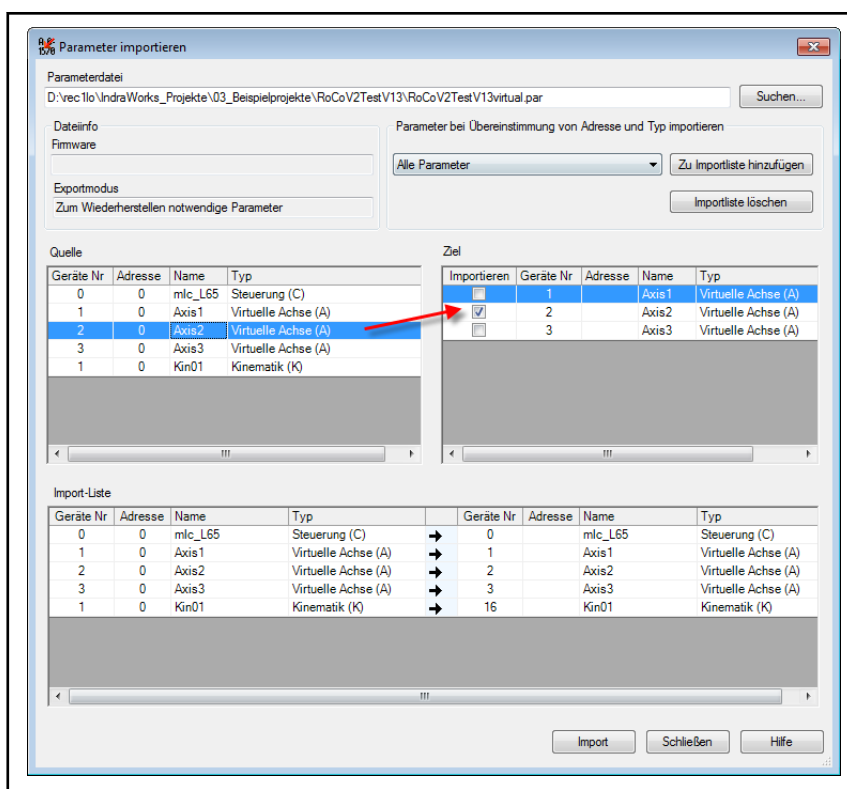


Abb. 5-8: Parameter auswählen

Nachfolgend öffnet sich automatisch der Assistent zum Import. Weisen Sie hierbei jeder Quelle ein Ziel zu, sodass die Steuerungs-, Achs- und Kinematikparameter importiert werden. Bestätigen Sie die Eingabe mit der Schaltfläche **Import**.

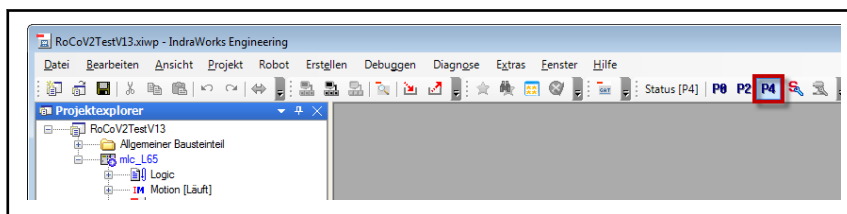


Abb. 5-9: Betriebsbereiter Zustand

Nach erfolgreichem Parameterimport sollte sich die Steuerung in den betriebsbereiten Zustand BB (P4) schalten lassen.

7. SPS-Programm erstellen.

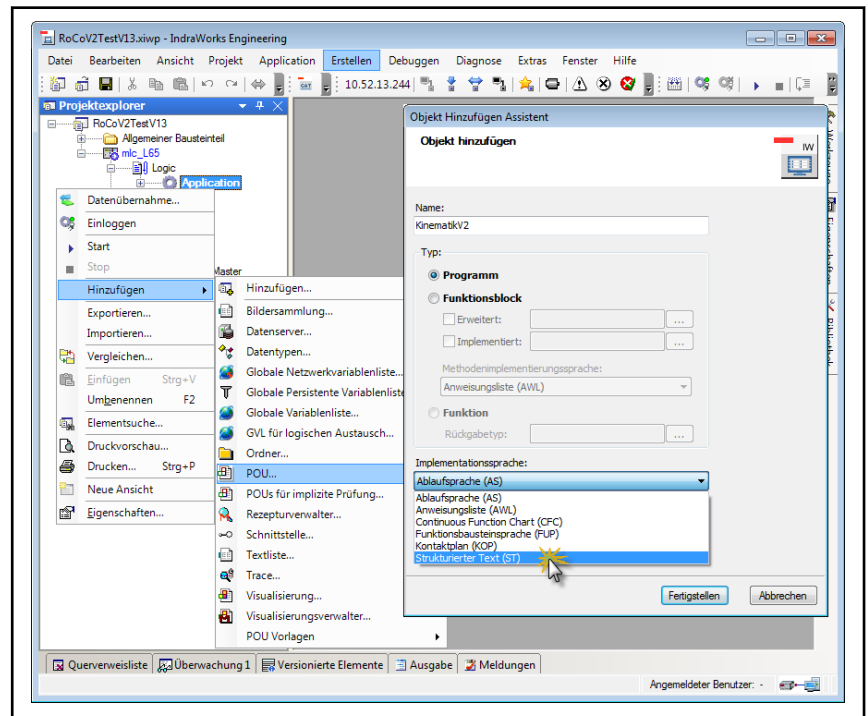


Abb. 5-10: SPS-Programm erstellen

Klicken Sie mit der rechten Maustaste auf den Knoten "Application" und wählen Sie im Kontextmenü "Hinzufügen" und nachfolgend "POU".

Geben Sie im Dialog "Objekt hinzufügen" einen beliebigen Programmnamen (z. B. "KinematikV2") an. Wählen Sie als Objekttyp "Programm" und als Implementationssprache "Strukturierter Text (ST)".

Fügen Sie das in "Detail zum Projekt RoCoV2TestV14 - Listing des SPS-Programms (KinematikV2)" hinterlegte Programm mit Deklarationsteil ein.

Fügen Sie ebenfalls im Programm "PlcProg" das in "Detail zum Projekt RoCoV2TestV14 - Listing des SPS-Programms (PlcProg)" hinterlegte Programm mit Deklarationsteil ein.

Alle Programme müssen sich an dieser Stelle über  fehlerfrei übersetzen lassen.

8. Aufruf des Programms in die MotionTask hinzufügen.

Mit Doppelklick auf die MotionTask gelangen Sie in die Taskkonfiguration.

Erste Schritte – Robot-Control V2

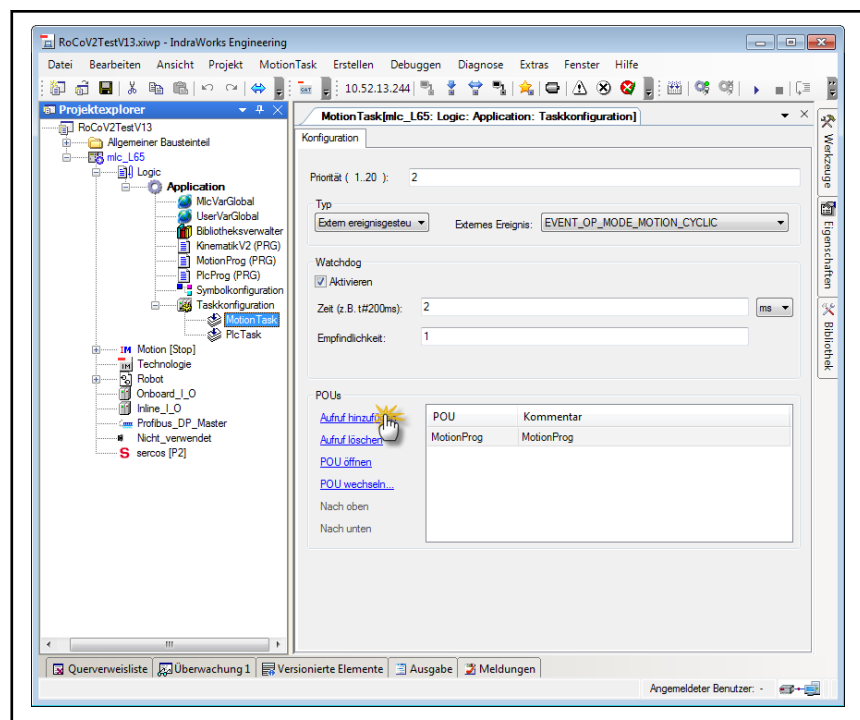


Abb. 5-11: Task konfigurieren

Fügen Sie hier den Aufruf des SPS-Programms hinzu. Wählen Sie hierzu über "Aufruf hinzufügen" das zuvor erstellte SPS-Programm aus und bestätigen Sie mit **OK**.

9. Visualisierung einbinden.

Zur Bedienung des SPS-Programms befindet sich eine Visualisierung im Projektverzeichnis auf der Installations-DVD ("AddOns\MLC\Example-Project\RobotControl\RoCoV2TestV14\Visualization.iwx"). Damit lassen sich alle nötigen Bedienaktionen durchführen.

Erste Schritte – Robot-Control V2

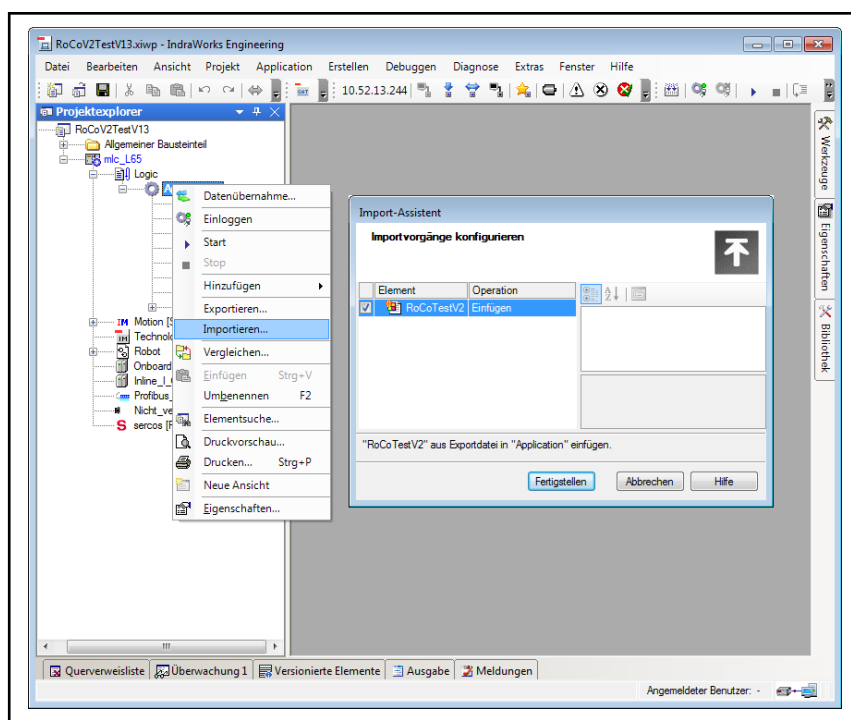


Abb. 5-12: Visualisierung importieren

Klicken Sie mit der rechten Maustaste auf "Application" und wählen Sie im Kontextmenü "Importieren" aus.

Wählen Sie aus dem Projektverzeichnis die Datei "Visualization" und bestätigen Sie mit **Öffnen**.

Bestätigen Sie anschließend den Importiervorgang mit der Schaltfläche **Fertigstellen**.

10. Download des SPS-Programms.

Durch das Einloggen auf die Steuerung wird das SPS-Programm heruntergeladen.

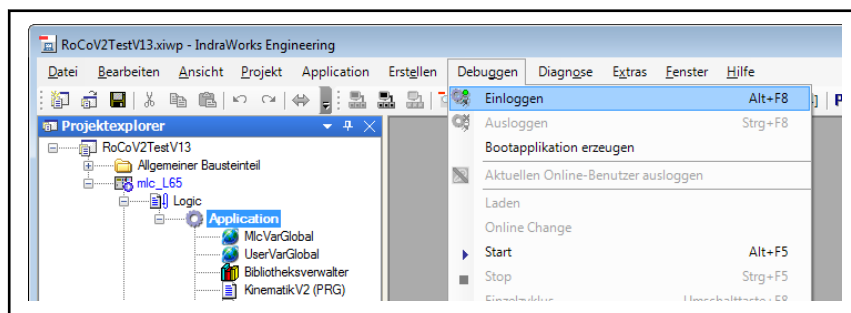


Abb. 5-13: Einloggen

Wählen Sie hierzu **Debuggen** ► **Einloggen** aus. Anschließend kann die Steuerung gestartet werden.

Erste Schritte – Robot-Control V2

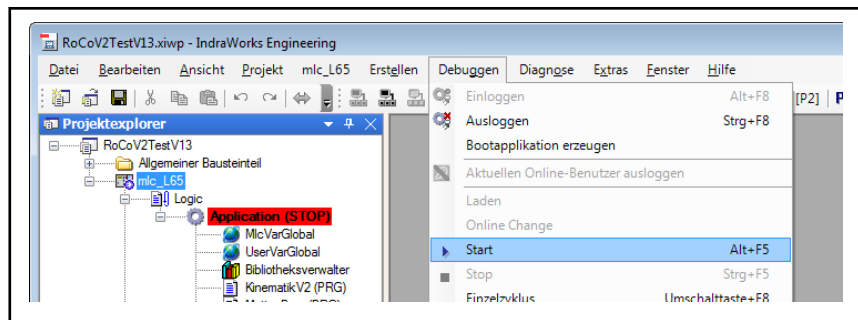


Abb. 5-14: Applikation starten

Wählen Sie hierfür **Debuggen ► Start** aus.

11. Start der Bewegung mithilfe der Visualisierung.

Mithilfe der importierten Visualisierung kann nun der Bewegungsablauf vorbereitet, gestartet und auch gestoppt werden.

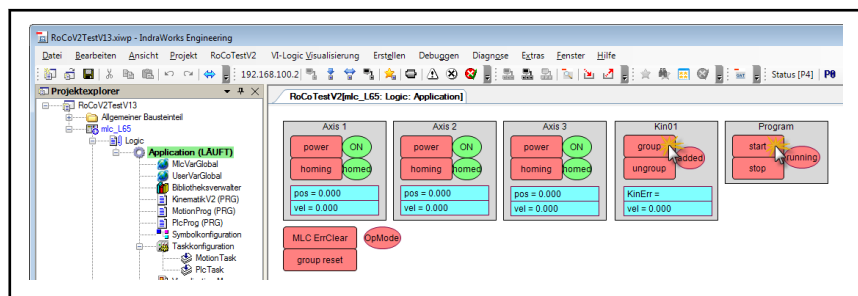


Abb. 5-15: Bewegung über Visualisierung starten

Gruppieren Sie die Achsen mithilfe des Buttons **group**. Sind die Achsen erfolgreich gruppiert wechselt das Feld "added" auf grün. In diesem Schritt werden alle Achsen, die zur definierten Kinematik gehören, gruppiert.

Starten Sie das Programm mithilfe des Buttons **start**. Das Programm wurde erfolgreich gestartet wenn das Feld "running" auf grün wechselt.

In gleicher Art und Weise kann das Programm gestoppt (Schaltfläche **stop**) und die Kinematik degruppiert (Schaltfläche **ungroup**) werden.

Für den Fehlerfall kann die Kinematik über die Schaltfläche **group reset** zurück gesetzt und alle Steuerungs- und Antriebsfehler über **MLC ErrClear** (C-0-1030) gelöscht werden.

Einbinden realer Achsen

Ebenfalls ist es möglich anstelle von virtuellen Achsen reale Achsen für die Kinematik zu nutzen. Schließen Sie hierzu drei reale Antriebe über den Sercos Bus an die Steuerung an und führen folgende Schritte aus.

Erste Schritte – Robot-Control V2

1. Löschen der virtuellen Achsen.

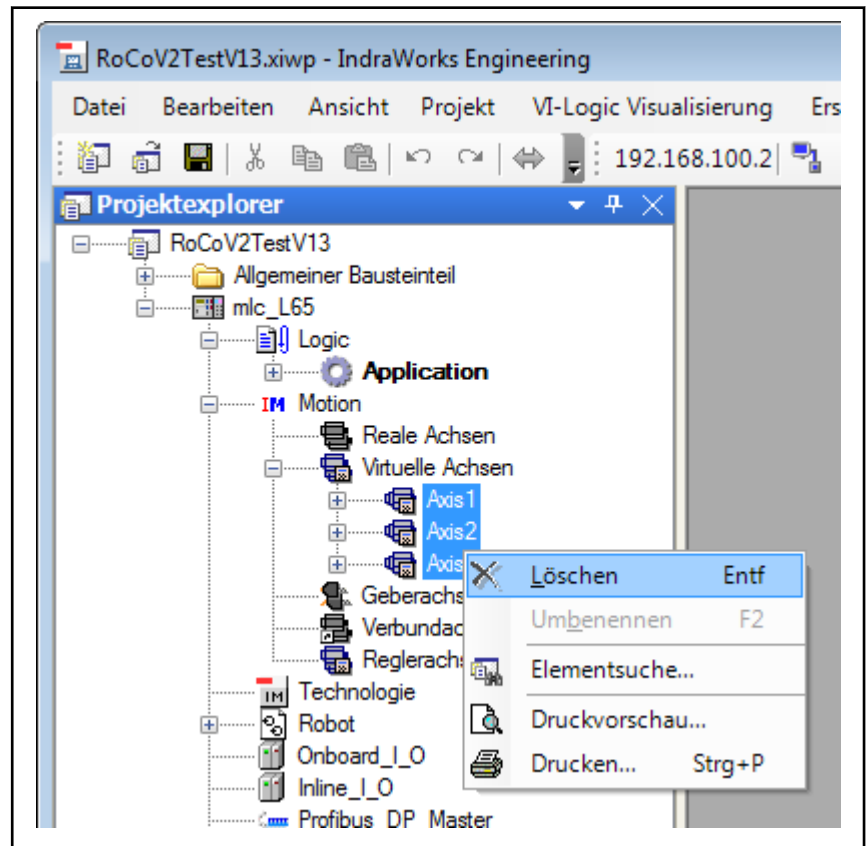


Abb. 5-16: Löschen der virtuellen Achsen

Markieren Sie alle drei virtuellen Achsen. Klicken Sie mit der rechten Maustaste auf die markierten Achsen und wählen Sie "Löschen" im Kontextmenü.

Erste Schritte – Robot-Control V2

2. Drei reale Antriebe anlegen und konfigurieren.

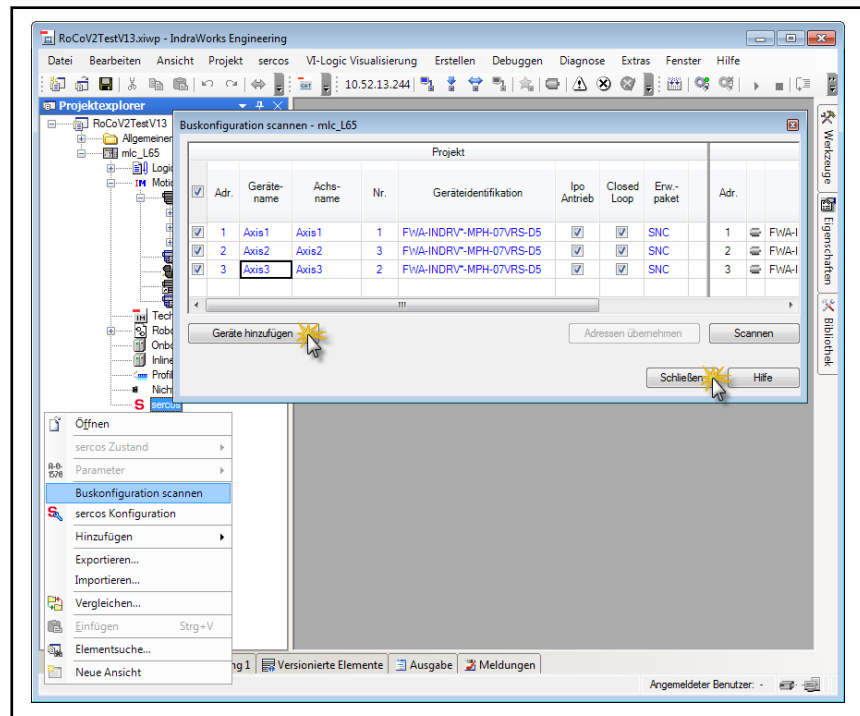


Abb. 5-17: Reale Achsen anlegen

Klicken Sie mit der rechten Maustaste auf den Sercos Knoten und wählen Sie im Kontextmenü "Buskonfiguration scannen".

Wählen Sie als Gerätenamen die Namen der gelöschten virtuellen Achsen ("Axis1", "Axis2", "Axis3"). Dadurch wird gewährleistet, dass das SPS-Programm ohne Anpassung lauffähig bleibt.

Fügen Sie die Achsen mit der Schaltfläche **Geräte hinzufügen** hinzu und schließen Sie das Fenster mit **Schließen**.

3. Anpassen der Kinematikkonfiguration.

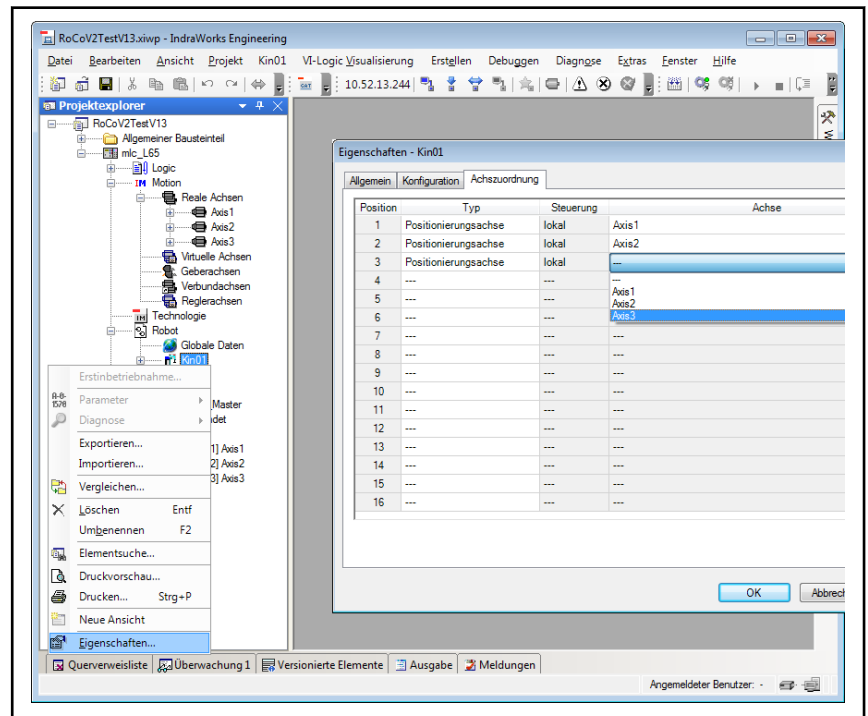


Abb. 5-18: Anpassen der Kinematikkonfiguration

Öffnen Sie den Eigenschaftendialog der Kinematik und passen Sie die Achszuordnungen an die neuen realen Achsen an.

Bestätigen Sie Ihre Eingabe mit der Schaltfläche **OK**.

4. Motion-Konfiguration herunterladen und Parameter importieren.

Laden Sie nach bekannter Methode die Motion-Konfiguration auf die Steuerung herunter und importieren Sie die Parameter für die realen Achsen. Steuerungs- und Kinematikparameter müssen nicht noch einmal importiert werden!



Die Parameterdatei für reale Achsen finden Sie im Projekt-Verzeichnis auf der Installations-DVD: "AddOns\MLC\Example-Projekt\RobotControl\RoCoV2TestV14\RoCoV2TestV14real.par".

Nun kann mit der bereits bekannten Visualisierung die Kinematik in Betrieb genommen werden. Es ist zu beachten, dass im Unterschied zu virtuellen Achsen bei realen Achsen die Leistung zugeschaltet (Schaltfläche **power**) und - insofern keine Absolutwertgeber verwendet werden - eine Referenzfahrt durchgeführt (Schaltfläche **homing**) werden muss. Erst dann kann eine Gruppierung vorgenommen werden.

5.4 Detail zum Projekt "RoCoV2TestV14"

Listing des SPS-Programms (KinematikV2), Deklaration

```
PROGRAM KinematikV2
VAR
  (* variables *)
  bPower      : ARRAY[1..3] OF BOOL := [3(FALSE)];
  bHome       : ARRAY[1..3] OF BOOL := [3(FALSE)];
  bGroup      : BOOL := FALSE;
  bUngroup    : BOOL := FALSE;
  bGroupReset : BOOL := FALSE;
  bStart      : BOOL := FALSE;
  bStop       : BOOL := FALSE;
```

Erste Schritte – Robot-Control V2

```

bRunning      : BOOL := FALSE;
bMoveDirectAbsoluteP0 : BOOL := FALSE;
bMoveDirectAbsoluteP1 : BOOL := FALSE;
bMoveDirectAbsoluteP2 : BOOL := FALSE;
bMoveDirectAbsoluteP3 : BOOL := FALSE;
bMoveDirectAbsoluteP4 : BOOL := FALSE;
bMoveDirectAbsoluteP5 : BOOL := FALSE;
bMoveDirectAbsoluteP6 : BOOL := FALSE;
rVFac: REAL := 0.0;
rAFac: REAL := 0.0;
rDFac: REAL := 0.0;
rJFac: REAL := 0.0;
arPositionP0: ARRAY[1..16] OF REAL := [0.0,0.0,0.0,13(0)];
arPositionP1: ARRAY[1..16] OF REAL := [100.0,0.0,0.0,13(0)];
arPositionP2: ARRAY[1..16] OF REAL := [100.0,100.0,0.0,13(0)];
arPositionP3: ARRAY[1..16] OF REAL := [100.0,100.0,100.0,13(0)];
arPositionP4: ARRAY[1..16] OF REAL := [400.0,400.0,400.0,13(0)];
arPositionP5: ARRAY[1..16] OF REAL := [300.0,200.0,100.0,13(0)];
arPositionP6: ARRAY[1..16] OF REAL := [111.0,222.0,333.0,13(0)];
uiState       : UINT := 0;
uiCounter     : UINT := 0;
tGroupWait    : TIME := T#0S;

(* instances *)
fbPower       : ARRAY[1..3] OF MC_Power;
fbHome        : ARRAY[1..3] OF MB_Home;
fbGroupAllAxes : ML_GroupAllAxes;
fbUngroupAllAxes: MC_UngroupAllAxes;
fbGroupReset   : MC_GroupReset;
fbMoveDirectAbsoluteP0: MC_MoveDirectAbsolute;
fbMoveDirectAbsoluteP1: MC_MoveDirectAbsolute;
fbMoveDirectAbsoluteP2: MC_MoveDirectAbsolute;
fbMoveDirectAbsoluteP3: MC_MoveDirectAbsolute;
fbMoveDirectAbsoluteP4: MC_MoveDirectAbsolute;
fbMoveDirectAbsoluteP5: MC_MoveDirectAbsolute;
fbMoveDirectAbsoluteP6: MC_MoveDirectAbsolute;
fbGroupWaitP0: ML_GroupWait;
fbGroupWaitP6: ML_GroupWait;
fbGroupStop    : MC_GroupStop;
END_VAR

VAR CONSTANT
    STATE_INIT       : UINT := 00;
    STATE_MOVE_ZERO : UINT := 10;
    STATE_MOVEMENT   : UINT := 20;
    STATE_FINISH     : UINT := 30;
END_VAR

```

Listing des SPS-Programms (KinematikV2), Implementation

```

(* example program for kinematics *)

(* set power for all real kinematic 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 kinematic axes *)
fbHome[1]( Execute := bHome[1], Axis := Axis1 );
fbHome[2]( Execute := bHome[2], Axis := Axis2 );
fbHome[3]( Execute := bHome[3], Axis := Axis3 );

(* group all axes *)
fbGroupAllAxes( Execute := bGroup, AxisGroup := Kin01 );

(* ungroup all axes *)
fbUnGroupAllAxes( Execute := bUngroup, AxisGroup := Kin01 );

(* reset group *)
fbGroupReset( Execute := bGroupReset, AxisGroup := Kin01 );

(* state machine *)
CASE uiState OF
    STATE_INIT: // initialization
        rVFac := 40; // velocity [%]
        rAFac := 100; // acceleration [%]
        rDFac := 100; // deceleration [%]
        rJFac := 100; // jerk [%]
        tGroupWait := T#3S;
        uiCounter := 0;
        bRunning := FALSE;

```

Erste Schritte – Robot-Control V2

```

IF bStart THEN
    uiState := STATE_MOVE_ZERO;
    bStart := FALSE;
    bRunning := TRUE;
END_IF;

STATE_MOVE_ZERO:    // move to zero position
    bMoveDirectAbsoluteP0 := TRUE;

    (* movement successfully - go to next state *)
    IF fbGroupWaitP0.Done THEN
        bMoveDirectAbsoluteP0 := FALSE;
        uiState := STATE_MOVEMENT;
    END_IF;

    (* movement stopped or finished with error - go to initialization *)
    IF fbMoveDirectAbsoluteP0.Error OR fbGroupWaitP0.Error OR bStop THEN
        bMoveDirectAbsoluteP0 := FALSE;
        uiState := STATE_INIT;
    END_IF;

STATE_MOVEMENT:    // move to several positions
    bMoveDirectAbsoluteP1 := TRUE;
    bMoveDirectAbsoluteP2 := TRUE;
    bMoveDirectAbsoluteP3 := TRUE;
    bMoveDirectAbsoluteP4 := TRUE;
    bMoveDirectAbsoluteP5 := TRUE;
    bMoveDirectAbsoluteP6 := TRUE;

    (* movements successfully - go to next state *)
    IF fbGroupWaitP6.Done THEN
        bMoveDirectAbsoluteP1 := FALSE;
        bMoveDirectAbsoluteP2 := FALSE;
        bMoveDirectAbsoluteP3 := FALSE;
        bMoveDirectAbsoluteP4 := FALSE;
        bMoveDirectAbsoluteP5 := FALSE;
        bMoveDirectAbsoluteP6 := FALSE;
        uiCounter := uiCounter + 1;
        uiState := STATE_FINISH;
    END_IF;

    (* movement stopped or finished with error - go to initialization *)
    IF fbGroupWaitP6.Error OR
        fbMoveDirectAbsoluteP1.Error OR
        fbMoveDirectAbsoluteP2.Error OR
        fbMoveDirectAbsoluteP3.Error OR
        fbMoveDirectAbsoluteP4.Error OR
        fbMoveDirectAbsoluteP5.Error OR
        fbMoveDirectAbsoluteP6.Error OR
        bStop THEN
        bMoveDirectAbsoluteP1 := FALSE;
        bMoveDirectAbsoluteP2 := FALSE;
        bMoveDirectAbsoluteP3 := FALSE;
        bMoveDirectAbsoluteP4 := FALSE;
        bMoveDirectAbsoluteP5 := FALSE;
        bMoveDirectAbsoluteP6 := FALSE;
        uiState := STATE_INIT;
    END_IF;

STATE_FINISH:    // movement finished
    IF uiCounter > 3 THEN
        uiState := STATE_INIT;
    ELSE
        uiState := STATE_MOVEMENT;
    END_IF;

END_CASE;

(* function block: move to PO *)
fbMoveDirectAbsoluteP0(
    Execute      := bMoveDirectAbsoluteP0,
    Position     := arPositionP0,
    Velocity     := rVFac,
    Acceleration := rAFac,
    Deceleration := rDFac,
    Jerk        := rJFac,
    CoordSystem  := CS_MACHINE_COORDINATES,
    SlopeType    := ST_BLOCK_SLOPE,
    SetMode      := SM_TO,
    BlendingRadius := 0.0,
    AxisGroup    := Kin01);

(* function block: wait 3 seconds *)

```

Erste Schritte – Robot-Control V2

```

fbGroupWaitP0 (
    Execute      := fbMoveDirectAbsoluteP0.Done,
    WaitTime     := tGroupWait,
    AxisGroup    := Kin01);

(* function block: move to P1 *)
fbMoveDirectAbsoluteP1 (
    Execute      := bMoveDirectAbsoluteP1,
    Position     := arPositionP1,
    Velocity     := rVFac,
    Acceleration := rAFac,
    Deceleration := rDFac,
    Jerk         := rJFac,
    CoordSystem  := CS_MACHINE_COORDINATES,
    SlopeType    := ST_BLOCK_SLOPE,
    SetMode      := SM_TO,
    BlendingRadius := 0.0,
    AxisGroup    := Kin01);

(* function block: move to P2 *)
fbMoveDirectAbsoluteP2 (
    Execute      := bMoveDirectAbsoluteP2,
    Position     := arPositionP2,
    Velocity     := rVFac,
    Acceleration := rAFac,
    Deceleration := rDFac,
    Jerk         := rJFac,
    CoordSystem  := CS_MACHINE_COORDINATES,
    SlopeType    := ST_BLOCK_SLOPE,
    SetMode      := SM_TO,
    BlendingRadius := 0.0,
    AxisGroup    := Kin01);

(* function block: move to P3 *)
fbMoveDirectAbsoluteP3 (
    Execute      := bMoveDirectAbsoluteP3,
    Position     := arPositionP3,
    Velocity     := rVFac,
    Acceleration := rAFac,
    Deceleration := rDFac,
    Jerk         := rJFac,
    CoordSystem  := CS_MACHINE_COORDINATES,
    SlopeType    := ST_BLOCK_SLOPE,
    SetMode      := SM_TO,
    BlendingRadius := 0.0,
    AxisGroup    := Kin01);

(* function block: move via P4 *)
fbMoveDirectAbsoluteP4 (
    Execute      := bMoveDirectAbsoluteP4,
    Position     := arPositionP4,
    Velocity     := rVFac,
    Acceleration := rAFac,
    Deceleration := rDFac,
    Jerk         := rJFac,
    CoordSystem  := CS_MACHINE_COORDINATES,
    SlopeType    := ST_BLOCK_SLOPE,
    SetMode      := SM_VIA,
    BlendingRadius := 30,
    AxisGroup    := Kin01);

(* function block: move via P5 *)
fbMoveDirectAbsoluteP5 (
    Execute      := bMoveDirectAbsoluteP5,
    Position     := arPositionP5,
    Velocity     := rVFac,
    Acceleration := rAFac,
    Deceleration := rDFac,
    Jerk         := rJFac,
    CoordSystem  := CS_MACHINE_COORDINATES,
    SlopeType    := ST_BLOCK_SLOPE,
    SetMode      := SM_VIA,
    BlendingRadius := 30,
    AxisGroup    := Kin01);

(* function block: move to P6 *)
fbMoveDirectAbsoluteP6 (
    Execute      := bMoveDirectAbsoluteP6,
    Position     := arPositionP6,
    Velocity     := rVFac,
    Acceleration := rAFac,
    Deceleration := rDFac,
    Jerk         := rJFac,

```

Erste Schritte – Robot-Control V2

```
CoordSystem      := CS_MACHINE_COORDINATES,  
SlopeType        := ST_BLOCK_SLOPE,  
SetMode          := SM_TO,  
BlendingRadius   := 0.0,  
AxisGroup        := Kin01);  
  
(* function block: wait 3 seconds *)  
fbGroupWaitP6(  
    Execute      := fbMoveDirectAbsoluteP6.Done,  
    WaitTime     := tGroupWait,  
    AxisGroup    := Kin01);  
  
(* function block: stop movement *)  
fbGroupStop(  
    Execute      := bStop,  
    Deceleration := rDFac,  
    Jerk         := rJFac,  
    AxisGroup    := Kin01);
```

Listing des SPS-Programms (PlcPrg), Deklaration

```
PROGRAM PlcProg  
VAR  
    fbReset      : MB_WriteParameter;  
    bErrClear    : BOOL;  
END_VAR
```

Listing des SPS-Programms (PlcPrg), Implementation

```
(* error clear MLC *)  
fbReset(  
    Execute      := bErrClear,  
    ParameterNumber := FP_C_0_1030,  
    Value        := 3,  
    Axis         := Axis1);
```

6 Service und Support

Für Ihre schnelle und optimale Unterstützung verfügen wir über ein dichtes weltweites Servicenetz. Unsere Experten stehen Ihnen mit Rat und Tat zur Seite. Sie erreichen uns täglich **rund um die Uhr – auch an Wochenenden und Feiertagen**.

Service Deutschland Unser technologieorientiertes Competence Center in Lohr deckt alle Belange rund um den Service für elektrische Antriebe und Steuerungen ab.

Sie erreichen unsere **Service-Hotline** und unseren **Service-Helpdesk** unter:

Telefon: **+49 9352 40 5060**
Fax: **+49 9352 18 4941**
E-Mail: service.svc@boschrexroth.de
Internet: <http://www.boschrexroth.com>

Auf unseren Internetseiten finden Sie ergänzende Hinweise zu Service, Reparatur (z. B. Anlieferadressen) und Training.

Service weltweit Außerhalb Deutschlands nehmen Sie bitte zuerst Kontakt mit Ihrem Ansprechpartner auf. Die Hotline-Rufnummern entnehmen Sie bitte den Vertriebsadressen im Internet.

Vorbereitung der Informationen Wir können Ihnen schnell und effizient helfen, wenn Sie folgende Informationen bereithalten:

- Eine detaillierte Beschreibung der Störung und der Umstände
- Angaben auf dem Typenschild der betreffenden Produkte, insbesondere Typenschlüssel und Seriennummern
- Ihre Kontaktdaten (Telefon-, Faxnummer und E-Mail-Adresse)

Index

A

Abkürzungen.....	14
Anregungen.....	14
ANSI Z535.6-2006.....	6

B

Beschwerde.....	14
Bezeichnungen.....	14
Bezeichnungen und Abkürzungen.....	14

E

Erste Schritte – FlexProfile.....	147
Bewegungsschritt ausführen,	
ML_FlexProfile.....	156
Download SPS-Teilprojekt Flexprofile.....	159
FlexProfile erstellen.....	150
FlexProfile erstellen über die SPS.....	152
FlexProfile erstellen über IndraWorks-Di-	
alog.....	150
Motion-Teilprojekt erstellen.....	148
Projekt wiederherstellen.....	162
Technische Voraussetzungen.....	148
Trace_FlexProfile.....	161
Workflow Virtuelle Achse → FlexProfile 0	
→ Reale Achse.....	160
Erste Schritte – IndraLogic.....	15
AS (SFC) editieren.....	40
Basiseinstellungen zum Gerät	18
Die Steuerung einer Ampelanlage.....	25
Element einfügen und editieren.....	70
Erstellen einer neuen Visualisierung.....	63
Funktionsbaustein einfügen.....	29
Funktionsmoduleinstellungen.....	24
FUP editieren.....	30
Geräteeinstellungen.....	18
Globale Variablen deklarieren.....	49
IndraWorks starten.....	16
Kommunikationseinstellungen.....	19
Neues Gerät in ein Projekt einfügen.....	18
Neues IndraWorks-Projekt	16
Neues Projekt erstellen.....	16
PlcProg in CFC editieren.....	56
PlcProg in ST editieren.....	53
POU einfügen.....	25, 30
Programm einfügen.....	28
Programme und Funktionsbausteine.....	25
Programme und Funktionsbausteine edi-	
tieren – zweiter Teil.....	53
Projekt wiederherstellen.....	75
SPS-Programm erstellen.....	25
ST editieren.....	38
Taskkonfiguration.....	48
Technische Voraussetzungen.....	15
Testen im Simulationsmodus.....	50
Visu-Element Anzeige von Variablenwerten..	73

Visu-Element Beschriftung.....	72
Visu-Element Ellipse.....	63
Visu-Element kopieren.....	67
Visu-Element Rechteck.....	68
Visu-Element Schaltfläche.....	71
Visualisierung editieren.....	63
Visualisierung in IndraLogic.....	62
Visualisierung testen.....	74
Visualisierung zum SPS-Programm erstellen	62
wiederherstellen.....	75
Erste Schritte – Motion.....	79
Basiseinstellungen zum Gerät.....	82
Bewegungsablauf programmieren (FUP).....	94
Bewegungsablauf programmieren (ST).....	118
Download des Motion-Teilprojektes.....	91
Download des SPS-Projektes Drive1 (FUP)	115
Download des SPS-Projektes Drive1 (ST)..	141
Download des SPS-Projektes vAxis1 (FUP)	103
Download des SPS-Projektes vAxis1 (ST)..	127
Funktionsmoduleinstellungen.....	88
Funktionspaketauswahl.....	22, 86
Geräteeinstellungen.....	82
IndraWorks starten.....	80
Kommunikationseinstellungen.....	82
Motion-Teilprojekt anlegen.....	80
Neues Gerät in ein Projekt einfügen.....	82
Neues Projekt erstellen.....	80
Programmieren der realen Achse Drive1	
(FUP).....	112
Programmieren der realen Achse Drive1	
(ST).....	136
Programmieren der virtuellen Achse vA-	
xis1 (FUP).....	96
Programmieren der virtuellen Achse vA-	
xis1 (ST).....	119
Projekt wiederherstellen.....	143
Sercos III Antriebe scannen.....	89
SPS-Eigenschaften.....	21, 85
Technische Voraussetzungen.....	79
Trace_vAxis1 (FUP).....	106
Trace_vAxis1 (ST).....	131
Trace_vAxis1_Drive1 (FUP).....	117
Trace_vAxis1_Drive1 (ST).....	142
Virtuelle Achse hinzufügen.....	90
Werte schreiben, Eingangsvariablen (FUP)..	105
Werte schreiben, Eingangsvariablen (ST)...	129
Workflow Virtuelle Achse → Elektron-	
isches Getriebe → Reale Achse.....	115, 141
Erste Schritte – Robot-Control V2.....	165
Erforderliche Hard- und Software.....	165
RoCo-Projekt erstellen.....	165
RoCoTestV2.....	177
Exportdatei	
Erste Schritte – FlexProfile.....	162
Erste Schritte – IndraLogic.....	75

Index

Erste Schritte – Motion.....	143
F	
Feedback.....	14
G	
Gefahrenhinweise.....	6
Gliederung der Dokumentation.....	6
H	
Helpdesk.....	183
Hotline.....	183
K	
Kritik.....	14
Kundenfeedback.....	14
S	
Service-Hotline.....	183
Sicherheitshinweise.....	6
Signalgrafik.....	6
Signalwörter.....	6
Support.....	183
Symbole.....	8
W	
Warnhinweise.....	6
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 9352 18 0

Fax +49 9352 18 8400

www.boschrexroth.com/electrics



R911341516