

VIPA SPEED7 Library

OPL_SP7-LIB | SW90MS0MA V10.008 | Handbuch

HB00 | OPL_SP7-LIB | SW90MS0MA V10.008 | de | 17-46

Baustein Bibliothek - Simple Motion Control



VIPA GmbH
Ohmstr. 4
91074 Herzogenaurach
Telefon: +49 9132 744-0
Telefax: +49 9132 744-1864
E-Mail: info@vipa.com
Internet: www.vipa.com

Inhaltsverzeichnis

1	Allgemeines.....	6
1.1	Copyright © VIPA GmbH	6
1.2	Über dieses Handbuch.....	7
2	Übersicht.....	8
3	Einsatz <i>Sigma-5/7</i> EtherCAT.....	9
3.1	Einsatz <i>Sigma-5</i> EtherCAT.....	9
3.1.1	Übersicht.....	9
3.1.2	Parameter am Antrieb einstellen.....	9
3.1.3	Einsatz im VIPA <i>SPEED7 Studio</i>	10
3.1.4	Einsatz im Siemens SIMATIC Manager.....	26
3.1.5	Antriebsspezifische Bausteine.....	45
3.2	Einsatz <i>Sigma-7S</i> EtherCAT.....	47
3.2.1	Übersicht.....	47
3.2.2	Parameter am Antrieb einstellen.....	48
3.2.3	Einsatz im VIPA <i>SPEED7 Studio</i>	49
3.2.4	Einsatz im Siemens SIMATIC Manager.....	65
3.2.5	Antriebsspezifische Bausteine.....	84
3.3	Einsatz <i>Sigma-7W</i> EtherCAT.....	86
3.3.1	Übersicht.....	86
3.3.2	Parameter am Antrieb einstellen.....	87
3.3.3	Einsatz im VIPA <i>SPEED7 Studio</i>	88
3.3.4	Einsatz im Siemens SIMATIC Manager.....	105
3.3.5	Antriebsspezifische Bausteine.....	126
3.4	Bausteine zur Achskontrolle.....	129
3.4.1	Übersicht.....	129
3.4.2	Einfache Bewegungsaufgaben.....	132
3.4.3	Komplexe Bewegungsaufgaben - PLCopen-Bausteine.....	136
4	Einsatz <i>Sigma-5/7</i> Pulse Train.....	201
4.1	Übersicht.....	201
4.2	Parameter am Antrieb einstellen.....	201
4.3	Beschaltung.....	202
4.4	Einsatz im VIPA <i>SPEED7 Studio</i>	204
4.4.1	Hardware-Konfiguration.....	204
4.4.2	Anwender-Programm.....	206
4.5	Einsatz im Siemens SIMATIC Manager.....	208
4.5.1	Voraussetzung.....	208
4.5.2	Hardware-Konfiguration.....	209
4.5.3	Anwender-Programm.....	211
4.6	Einsatz im Siemens TIA Portal.....	213
4.6.1	Voraussetzung.....	213
4.6.2	Hardware-Konfiguration.....	213
4.6.3	Anwender-Programm.....	216
4.7	Antriebsspezifischer Baustein.....	219
4.7.1	FB 875 - VMC_AxisControl_PT - Achskontrolle über Pulse Train.....	219
5	Einsatz Frequenzumrichter über PWM.....	227
5.1	Übersicht.....	227
5.2	Parameter am Frequenzumrichter einstellen.....	227

5.3	Beschaltung.....	229
5.3.1	Ansteuerung V1000 Eingänge.....	229
5.3.2	Ansteuerung V1000 Ausgänge.....	230
5.4	Einsatz im VIPA <i>SPEED7 Studio</i>	230
5.4.1	Hardware-Konfiguration.....	230
5.4.2	Anwender-Programm.....	233
5.5	Einsatz im Siemens SIMATIC Manager.....	235
5.5.1	Voraussetzung.....	235
5.5.2	Hardware-Konfiguration.....	236
5.5.3	Anwender-Programm.....	238
5.6	Einsatz im Siemens TIA Portal.....	240
5.6.1	Voraussetzung.....	240
5.6.2	Hardware-Konfiguration.....	240
5.6.3	Anwender-Programm.....	243
5.7	Antriebsspezifischer Baustein.....	246
5.7.1	FB 885 - VMC_AxisControlV1000_PWM - Achskontrolle über PWM.....	246
6	Einsatz Frequenzumrichter über Modbus RTU.....	250
6.1	Übersicht.....	250
6.2	Parameter am Frequenzumrichter einstellen.....	250
6.3	Beschaltung.....	252
6.4	Einsatz im VIPA <i>SPEED7 Studio</i>	254
6.4.1	Hardware-Konfiguration.....	254
6.4.2	Anwender-Programm.....	259
6.5	Einsatz im Siemens SIMATIC Manager.....	269
6.5.1	Voraussetzung.....	269
6.5.2	Hardware-Konfiguration.....	269
6.5.3	Anwender-Programm.....	275
6.6	Einsatz im Siemens TIA Portal.....	284
6.6.1	Voraussetzung.....	284
6.6.2	Hardware-Konfiguration.....	285
6.6.3	Anwender-Programm.....	293
6.7	Antriebsspezifische Bausteine.....	303
6.7.1	UDT 877 - VMC_ComSlavesRTU_REF - Modbus RTU Datenstruktur Kommunikationsdaten aller Slaves.....	303
6.7.2	UDT 878 - VMC_ComObjectRTU_REF - Modbus RTU Datenstruktur Kommunikationsdaten Slave.....	303
6.7.3	UDT 879 - VMC_AxisRTU_REF - Modbus RTU Datenstruktur Achs- daten.....	303
6.7.4	UDT 881 - VMC_ConfigV1000RTU_REF - Modbus RTU Datenstruktur Konfiguration.....	303
6.7.5	FB 876 - VMC_ConfigMaster_RTU - Modbus RTU CPU-Schnittstelle.....	303
6.7.6	FB 877 - VMC_ComManager_RTU - Modbus RTU Kommunikations- Manager.....	305
6.7.7	FB 878 - VMC_RWParameterSys_RTU - Modbus RTU Parameter System lesen/schreiben.....	305
6.7.8	FB 879 - VMC_ReadParameter_RTU - Modbus RTU Parameter lesen...	305
6.7.9	FB 880 - VMC_WriteParameter_RTU - Modbus RTU Parameter schreiben.....	306
6.7.10	FB 881 - VMC_InitV1000_RTU - Modbus RTU Initialisierung.....	307
6.7.11	FB 882 - VMC_AxisControlV1000_RTU - Modbus RTU Achskontrolle...	309

- 7 Antrieb über HMI steuern..... 313**
 - 7.1 Neues Projekt erstellen..... 314
 - 7.2 Projekt in Movicon anpassen..... 318
 - 7.3 Inbetriebnahme..... 329
 - 7.3.1 VMC_AxisControl über das Panel ansteuern..... 330
- 8 Zustände und Verhalten der Ausgänge..... 334**
 - 8.1 Zustände..... 334
 - 8.2 Ablöseverhalten von Bewegungsaufträgen..... 336
 - 8.3 Verhalten der Ein- und Ausgänge..... 338
- 9 ErrorID - Zusätzliche Fehlerinformationen..... 340**

1 Allgemeines

1.1 Copyright © VIPA GmbH

All Rights Reserved

Dieses Dokument enthält geschützte Informationen von VIPA und darf außer in Übereinstimmung mit anwendbaren Vereinbarungen weder offengelegt noch benutzt werden.

Dieses Material ist durch Urheberrechtsgesetze geschützt. Ohne schriftliches Einverständnis von VIPA und dem Besitzer dieses Materials darf dieses Material weder reproduziert, verteilt, noch in keiner Form von keiner Einheit (sowohl VIPA-intern als auch -extern) geändert werden, es sei denn in Übereinstimmung mit anwendbaren Vereinbarungen, Verträgen oder Lizenzen.

Zur Genehmigung von Vervielfältigung oder Verteilung wenden Sie sich bitte an: VIPA, Gesellschaft für Visualisierung und Prozessautomatisierung mbH Ohmstraße 4, D-91074 Herzogenaurach, Germany

Tel.: +49 9132 744 -0

Fax.: +49 9132 744-1864

E-Mail: info@vipa.de

<http://www.vipa.com>



Es wurden alle Anstrengungen unternommen, um sicherzustellen, dass die in diesem Dokument enthaltenen Informationen zum Zeitpunkt der Veröffentlichung vollständig und richtig sind. Das Recht auf Änderungen der Informationen bleibt jedoch vorbehalten.

Die vorliegende Kundendokumentation beschreibt alle heute bekannten Hardware-Einheiten und Funktionen. Es ist möglich, dass Einheiten beschrieben sind, die beim Kunden nicht vorhanden sind. Der genaue Lieferumfang ist im jeweiligen Kaufvertrag beschrieben.

EG-Konformitätserklärung

Hiermit erklärt VIPA GmbH, dass die Produkte und Systeme mit den grundlegenden Anforderungen und den anderen relevanten Vorschriften übereinstimmen. Die Übereinstimmung ist durch CE-Zeichen gekennzeichnet.

Informationen zur Konformitätserklärung

Für weitere Informationen zur CE-Kennzeichnung und Konformitätserklärung wenden Sie sich bitte an Ihre Landesvertretung der VIPA GmbH.

Warenzeichen

VIPA, SLIO, System 100V, System 200V, System 300V, System 300S, System 400V, System 500S und Commander Compact sind eingetragene Warenzeichen der VIPA Gesellschaft für Visualisierung und Prozessautomatisierung mbH.

SPEED7 ist ein eingetragenes Warenzeichen der profichip GmbH.

SIMATIC, STEP, SINEC, TIA Portal, S7-300 und S7-400 sind eingetragene Warenzeichen der Siemens AG.

Microsoft und Windows sind eingetragene Warenzeichen von Microsoft Inc., USA.

Portable Document Format (PDF) und Postscript sind eingetragene Warenzeichen von Adobe Systems, Inc.

Alle anderen erwähnten Firmennamen und Logos sowie Marken- oder Produktnamen sind Warenzeichen oder eingetragene Warenzeichen ihrer jeweiligen Eigentümer.

Dokument-Support

Wenden Sie sich an Ihre Landesvertretung der VIPA GmbH, wenn Sie Fehler anzeigen oder inhaltliche Fragen zu diesem Dokument stellen möchten. Ist eine solche Stelle nicht erreichbar, können Sie VIPA über folgenden Kontakt erreichen:

VIPA GmbH, Ohmstraße 4, 91074 Herzogenaurach, Germany

Telefax: +49 9132 744-1204

E-Mail: documentation@vipa.de

Technischer Support

Wenden Sie sich an Ihre Landesvertretung der VIPA GmbH, wenn Sie Probleme mit dem Produkt haben oder Fragen zum Produkt stellen möchten. Ist eine solche Stelle nicht erreichbar, können Sie VIPA über folgenden Kontakt erreichen:

VIPA GmbH, Ohmstraße 4, 91074 Herzogenaurach, Germany

Telefon: +49 9132 744-1150 (Hotline)

E-Mail: support@vipa.de

1.2 Über dieses Handbuch

Zielsetzung und Inhalt

Das Handbuch beschreibt die Baustein-Bibliothek *"Simple Motion Control"* von VIPA:

- Beschrieben wird Aufbau, Projektierung und Anwendung in verschiedenen Programmiersystemen.
- Das Handbuch ist geschrieben für Anwender mit Grundkenntnissen in der Automatisierungstechnik.
- Das Handbuch ist in elektronischer Form als PDF-Datei verfügbar. Hierzu ist der Adobe Acrobat Reader erforderlich.
- Das Handbuch ist in Kapitel gegliedert. Jedes Kapitel beschreibt eine abgeschlossene Thematik.
- Als Orientierungshilfe stehen im Handbuch zur Verfügung:
 - Gesamt-Inhaltsverzeichnis am Anfang des Handbuchs
 - Verweise mit Seitenangabe

Piktogramme Signalwörter

Besonders wichtige Textteile sind mit folgenden Piktogrammen und Signalworten ausgezeichnet:

**GEFAHR!**

Unmittelbar drohende oder mögliche Gefahr. Personenschäden sind möglich.

**VORSICHT!**

Bei Nichtbefolgen sind Sachschäden möglich.



Zusätzliche Informationen und nützliche Tipps.

2 Übersicht

Leistungsmerkmale

Mit den Bausteinen der *Simple Motion Control Library* können Sie auf einfache Weise ohne Detailwissen Antriebe in Ihre Applikationen integrieren. Hierbei werden verschiedene Antriebe und Bussystem unterstützt. Mittels der PLCopen-Bausteine können Sie einfache Antriebsaufgaben in Ihrer Steuerung realisieren. Dieses System bietet folgenden Leistungsumfang:

- Einsetzbar im VIPA *SPEED7 Studio* und Siemens SIMATIC Manager
- Umsetzung von einfachen Antriebsfunktionen
 - Einschalten bzw. Ausschalten
 - Drehzahlvorgabe
 - Relative bzw. absolute Positionierung
 - Referenzfahrt (Homing)
 - Lesen und Schreiben von Parametern
 - Abfrage von Achsposition und Status
- Einfache Inbetriebnahme und Diagnose ohne detaillierte Kenntnisse der Antriebe
- Unterstützung verschiedener Antriebe und Feldbusse
- Visualisierung einzelner Achsen
- Skalierbar durch Einsatz von PLCopen-Bausteinen

Struktur

Die *Simple Motion Control Library* ist in folgende Gruppen gegliedert:

- Axis Control
 - Allgemeine Bausteine zur Steuerung der Antriebe.
- Sigma5 EtherCAT
 - Spezifische Bausteine für den Einsatz von *Sigma-5*-Antrieben, welche über EtherCAT angebunden sind.
- Sigma7 EtherCAT
 - Spezifische Bausteine für den Einsatz von *Sigma-7S*-Antrieben, welche über EtherCAT angebunden sind.
 - Spezifische Bausteine für den Einsatz von *Sigma-7W*-Antrieben, welche über EtherCAT angebunden sind.
- Sigma5+7 PulseTrain
 - Spezifischer Baustein für den Einsatz von *Sigma-5* bzw. *Sigma-7*-Antrieben, welche über Pulse Train angebunden sind.
- V1000 PWM
 - Spezifischer Baustein für den Einsatz von *V1000*-Frequenzumrichter, welche über PWM angebunden sind.
- V1000 Modbus RTU
 - Spezifische Bausteine für den Einsatz von *V1000*-Frequenzumrichter, welche über Modbus-RTU angebunden sind.

3 Einsatz Sigma-5/7 EtherCAT

3.1 Einsatz Sigma-5 EtherCAT

3.1.1 Übersicht

Voraussetzung

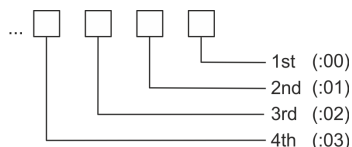
- SPEED7 Studio ab V1.6.1
oder
- Siemens SIMATIC Manager ab V 5.5 SP2 & *SPEED7 EtherCAT Manager & Simple Motion Control Library*
- CPU mit EtherCAT-Master wie z.B. CPU 015-CEFN00
- Sigma-5-Antrieb mit EtherCAT-Optionskarte

Schritte der Projektierung

1. ➤ Parameter am Antrieb einstellen
 - Die Einstellung der Parameter hat mit dem Softwaretool *Sigma Win+* zu erfolgen.
2. ➤ Hardwarekonfiguration im VIPA *SPEED7 Studio* oder Siemens SIMATIC Manager
 - Projektierung einer CPU mit EtherCAT-Master-Funktionalität.
 - Projektierung eines Sigma-5 EtherCAT-Antriebs.
 - Projektierung der EtherCAT-Anbindung über *SPEED7 EtherCAT Manager*.
3. ➤ Programmierung im VIPA *SPEED7 Studio* oder Siemens SIMATIC Manager
 - *Init*-Baustein zur Konfiguration der Achse beschalten.
 - *Kernel*-Baustein zur Kommunikation mit der Achse beschalten.
 - Bausteine für die Bewegungsabläufe beschalten.

3.1.2 Parameter am Antrieb einstellen

Parameter-Digits



VORSICHT!

Vor der Erstinbetriebnahme müssen Sie Ihren Antrieb mit dem Softwaretool *Sigma Win+* an Ihre Applikation anpassen! Näheres hierzu finden Sie im Handbuch zu Ihrem Antrieb.

Zur Abstimmung auf die *Simple Motion Control Library* sind folgende Parameter über *Sigma Win+* einzustellen:

Sigma-5 (20Bit Encoder)

Servopack Parameter	Adresse:Digit	Name	Wert
Pn205	(2205h)	Multiturn Limit Setting	65535
Pn20E	(220Eh)	Electronic Gear Ratio (Numerator)	1
Pn210	(2210h)	Electronic Gear Ratio (Denominator)	1
PnB02	(2701h:01)	Position User Unit (Numerator)	1
PnB04	(2701h:02)	Position User Unit (Denominator)	1
PnB06	(2702h:01)	Velocity User Unit (Numerator)	1
PnB08	(2702h:02)	Velocity User Unit (Denominator)	1

Servopack Parameter	Adresse:Digit	Name	Wert
PnB0A	(2703h:01)	Acceleration User Unit (Numerator)	1
PnB0C	(2703h:02)	Acceleration User Unit (Denominator)	1

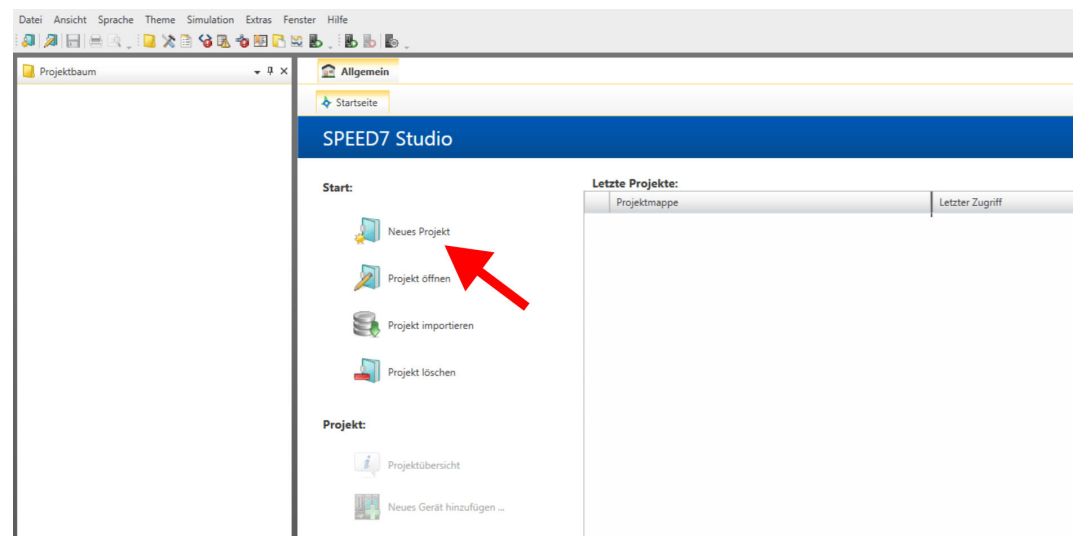
3.1.3 Einsatz im VIPA SPEED7 Studio

3.1.3.1 Hardware-Konfiguration

CPU im Projekt anlegen

Bitte verwenden Sie für die Projektierung das *SPEED7 Studio* ab V1.6.1.

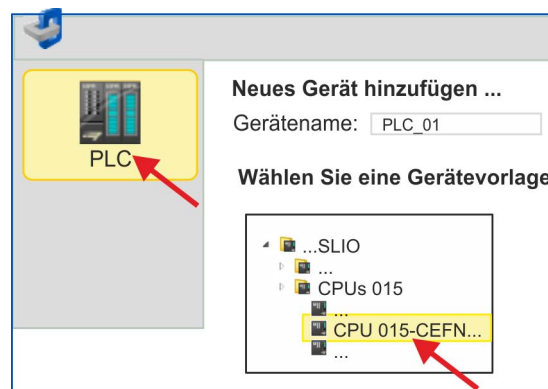
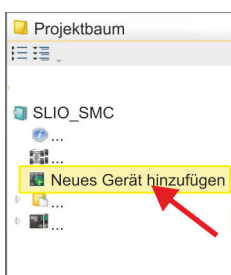
1. Starten Sie das *SPEED7 Studio*.



2. Erstellen sie auf der Startseite mit "Neues Projekt" ein neues Projekt und vergeben Sie einen "Projektnamen".

⇒ Ein neues Projekt wird angelegt und in die Sicht "Geräte und Netze" gewechselt.

3. Klicken Sie im *Projektbaum* auf "Neues Gerät hinzufügen ...".



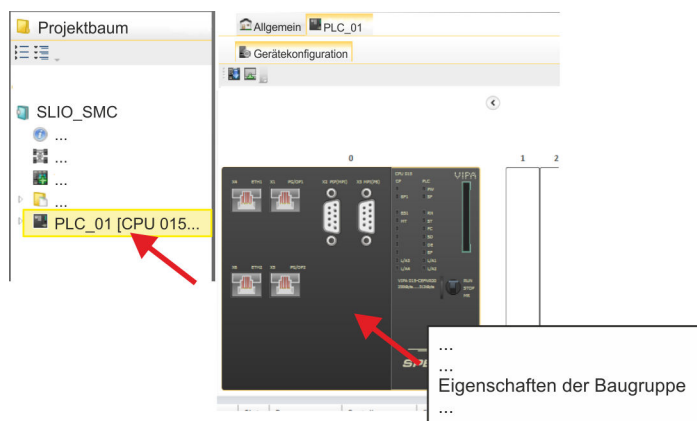
⇒ Es öffnet sich ein Dialog für die Geräteauswahl.

4. Wählen Sie unter den "Gerätevorlagen" eine CPU mit EtherCAT-Master-Funktionalität wie z.B. die CPU 015-CEFN00 und klicken Sie auf [OK].

⇒ Die CPU wird in "Geräte und Netze" eingefügt und die "Gerätekonfiguration" geöffnet.

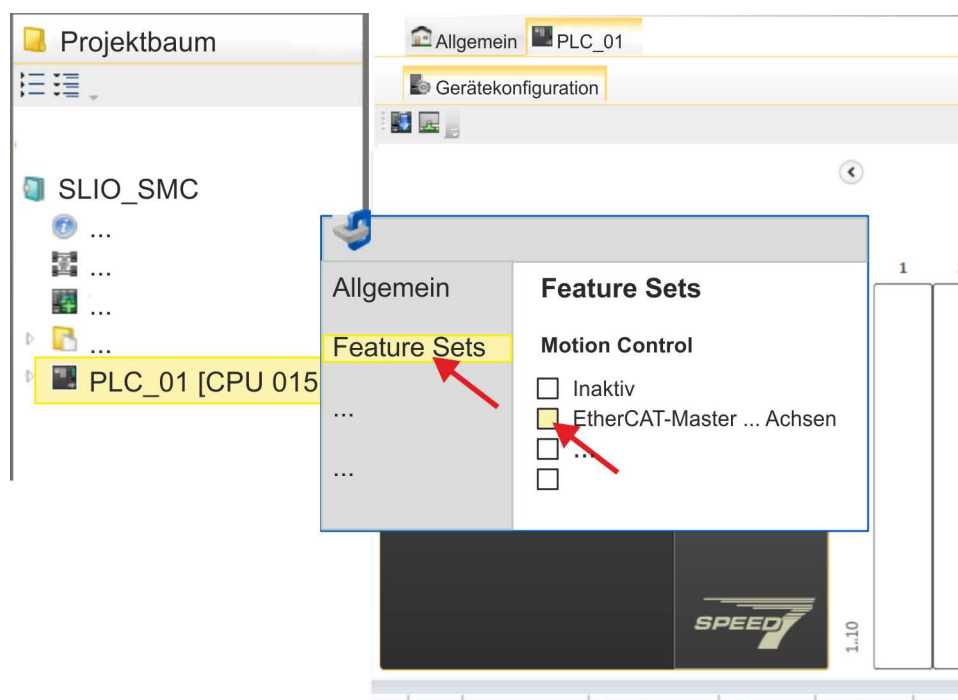
Motion-Control-Funktionen aktivieren

Sofern bei Ihrer CPU die EtherCAT-Master-Funktionalität noch nicht aktiviert ist, erfolgt die Aktivierung nach folgenden Vorgehensweise:



1. Klicken Sie in der "Gerätekonfiguration" auf die CPU und wählen Sie "Kontextmenü → Eigenschaften der Baugruppe".

⇒ Es öffnet sich der Eigenschaften-Dialog der CPU.



2. Klicken Sie auf "Feature Sets" und aktivieren Sie unter "Motion Control" einen der Parameter "EtherCAT-Master ... Achsen". Die Anzahl der Achsen ist in diesem Beispiel nicht relevant.

3. Bestätigen Sie Ihre Angaben mit [OK].

⇒ Die Motion-Control-Funktionen stehen Ihnen nun in Ihrem Projekt zur Verfügung.

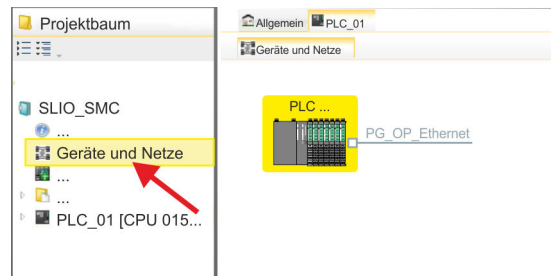


VORSICHT!

Bitte beachten Sie, dass bei jeder Änderung der Feature-Set-Einstellungen systembedingt das EtherCAT-Feldbus-System zusammen mit der Motion-Control-Konfiguration aus Ihrem Projekt gelöscht werden!

Ethernet-PG/OP-Kanal parametrieren

1. ➤ Klicken Sie im *Projektbaum* auf "*Geräte und Netze*".
⇒ Sie erhalten eine grafische Objekt-Ansicht Ihrer CPU.



2. ➤ Klicken Sie auf das Netzwerk "*PG_OP_Ethernet*".
3. ➤ Wählen Sie "*Kontextmenü* ➔ *Eigenschaften der Schnittstelle*".
⇒ Es öffnet sich ein Dialogfenster. Hier können Sie IP-Adressdaten für Ihren Ethernet-PG/OP-Kanal angeben. Gültige IP-Adress-Parameter erhalten Sie von Ihrem Systemadministrator.
4. ➤ Bestätigen Sie Ihre Eingabe mit [OK].
⇒ Die IP-Adressdaten werden in Ihr Projekt übernommen und in "*Geräte und Netze*" unter "*Lokale Baugruppen*" aufgelistet.
Nach der Übertragung Ihres Projekts ist Ihre CPU über die angegebenen IP-Adressdaten via Ethernet-PG/OP-Kanal erreichbar.

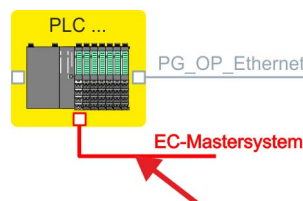
ESI-Datei installieren

Damit der Sigma-5 EtherCAT Antrieb im *SPEED7 EtherCAT Manager* konfiguriert werden kann, muss die entsprechende ESI-Datei installiert sein. In der Regel wird das *SPEED7 Studio* mit aktuellen ESI-Dateien ausgeliefert und Sie können diesen Teil überspringen. Sollte Ihre ESI-Datei veraltet sein, finden Sie die aktuellste ESI-Datei für den Sigma-5 EtherCAT Antrieb unter www.yaskawa.eu.com unter "*Service* ➔ *Drives & Motion Software*".

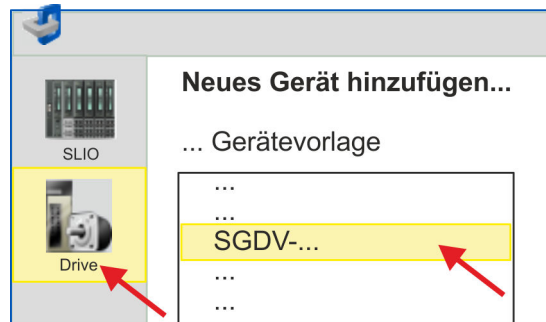
1. ➤ Laden Sie die zu Ihrem Antrieb passende ESI-Datei herunter. Entpacken Sie diese falls erforderlich.
2. ➤ Gehen Sie in Ihr *SPEED7 Studio*.
3. ➤ Öffnen Sie mit "*Extras* ➔ *Gerätebeschreibungsdatei installieren (EtherCAT - ESI)*" das zugehörige Dialogfenster.
4. ➤ Geben Sie unter "*Quellpfad*" die ESI-Datei an und installieren Sie diese mit [Installieren].
⇒ Die Geräte der ESI-Datei steht Ihnen nun zur Verfügung.

Sigma-5 Antrieb hinzufügen

1. ➤ Klicken Sie im Projektbaum auf "*Geräte und Netze*".
2. ➤ Klicken Sie hier auf "*EC-Mastersystem*" und wählen sie "*Kontextmenü* ➔ *Neues Gerät hinzufügen*".



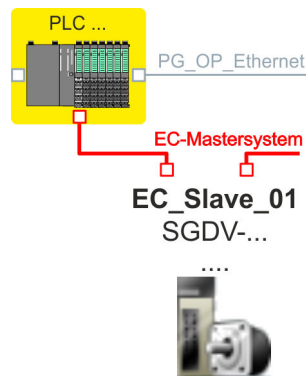
- ⇒ Es öffnet sich die Gerätevorlage zur Auswahl eines EtherCAT-Devices.



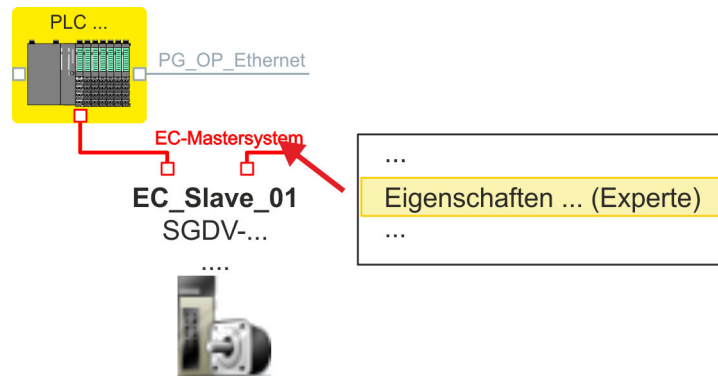
3. Wählen Sie Ihren *Sigma-5* Antrieb aus:

- SGDV-xxxxE5...
- SGDV-xxxxE1...

Bestätigen Sie Ihre Angaben mit [OK]. Sollte Ihr Antrieb nicht vorhanden sein, müssen Sie die entsprechende ESI-Datei wie weiter oben beschrieben installieren.



⇒ Der *Sigma-5* Antrieb wird an Ihr EC-Mastersystem angebunden.

Sigma-5 Antrieb konfigurieren

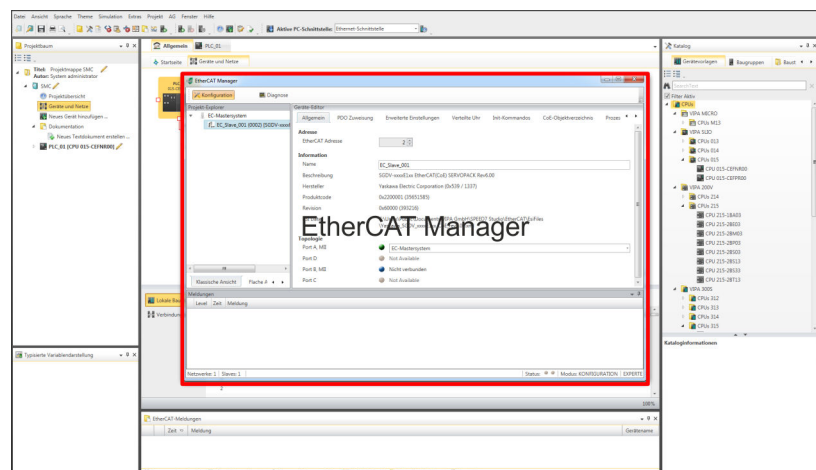
1. Klicken Sie auf "EC-Mastersystem" und wählen sie "Kontextmenü → Eigenschaft des Bussystems (Experte)".



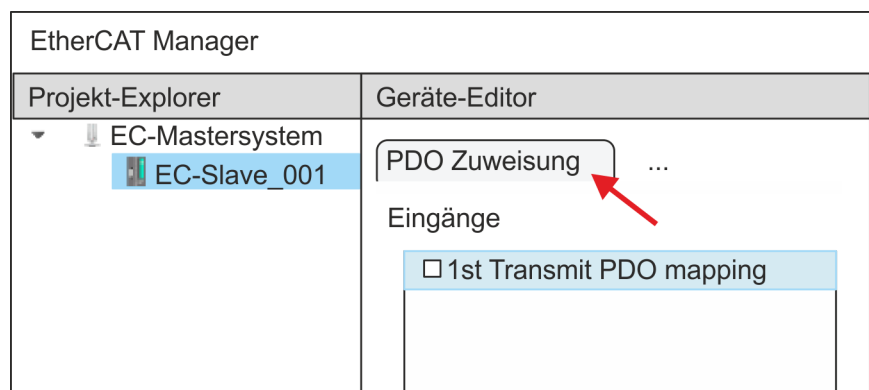
PDOs können Sie nur im "Experten-Modus" bearbeiten! Ansonsten werden die Schaltflächen ausgeblendet.

- ⇒ Der **SPEED7 EtherCAT Manager** wird gestartet. Hier können Sie die EtherCAT-Kommunikation zu Ihrem Sigma-5 Antrieb konfigurieren.

Näheres zum Einsatz des **SPEED7 EtherCAT Manager** finden Sie in der Onlinehilfe zum **SPEED7 Studio**.



2. Klicken Sie im **SPEED7 EtherCAT Manager** auf den Slave und wählen Sie im "Geräte-Editor" den Reiter "PDO-Zuweisung" an.



- ⇒ Dieser Dialog zeigt eine Auflistung aller PDOs.

3. ➔ Durch Anwahl des entsprechenden PDO-Mappings können Sie mit [Bearbeiten] die PDOs bearbeiten. Wählen Sie das Mapping "1st Transmit PDO mapping" an und klicken Sie auf [Bearbeiten].



Bitte beachten Sie, dass aufgrund der Voreinstellung manche PDOs nicht bearbeitet werden können. Durch Deaktivierung bereits aktivierter PDOs können Sie die Bearbeitung von gesperrten PDOs frei geben.

Geräte-Editor

PDO Zuweisung ...

Eingänge	Ausgänge
<input type="checkbox"/> 1st Transmit PDO mapping	<input type="checkbox"/> 1st Receive PDO mapping
<input type="checkbox"/> 2nd Transmit PDO mapping	<input type="checkbox"/> 2nd Receive PDO mapping
...	...

... Bearbeiten ...

- ⇒ Es öffnet sich der Dialog *"PDO bearbeiten"*. Bitte überprüfen Sie hier die aufgeführten PDO-Einstellungen und passen Sie diese ggf. an. Bitte berücksichtigen Sie hierbei auch die Reihenfolge der *"Einträge"* und ergänzen Sie diese entsprechend.

The dialog box 'PDO bearbeiten' contains the following elements:

- Allgemein:**
 - Name: 1st Transmit PDO mapping
 - Index: 0x1A00 (with 'Dez' and 'Hex' radio buttons, 'Hex' is selected)
 - Flags:
 - Zwingend: ☐
 - Schreibgeschützt: ☐
 - Virtuell: ☐
 - Richtung:
 - TxPdo (Eingang): ☒
 - RxPdo (Ausgang): ☐
- Optional:**
 - Ausschließen:
 - 1A01: ☐
 - 1A02: ☒
 - 1A03: ☒
- Einträge:** A table listing various data objects.

Name	Index	Bitlänge	Kommentar
Status word	0x6041:00	16	
Position actual internal value	0x6063:00	32	
Position actual value	0x6064:00	32	
Torque actual value	0x6077:00	16	
Following error actual value	0x60F4:00	32	
Modes of operation display	0x6061:00	8	
---	---	8	---
Digital inputs	0x60FD:00	32	

At the bottom of the dialog are five buttons: Neu, Löschen, Bearbeiten, Nach oben, and Nach unten.

Für die Bearbeitung der *"Einträge"* stehen folgende Funktionen zur Verfügung:

- Neu
 - Hiermit können Sie in einem Dialogfenster einen neuen Eintrag anlegen, indem Sie aus dem *"CoE-Objektverzeichnis"* den entsprechenden Eintrag auswählen und Ihre Einstellungen vornehmen. Mit [OK] wird der Eintrag übernommen und in der Liste der Einträge aufgeführt.
- Löschen
 - Hiermit können Sie den angewählte Eintrag löschen.
- Bearbeiten
 - Hiermit können Sie allgemeinen Daten eines Eintrags bearbeiten.
- Nach oben/unten
 - Hiermit können Sie den angewählten Eintrag in der Liste nach oben bzw. nach unten bewegen.

4. ➔ Führen Sie folgende Einstellungen durch:**Eingänge: 1st Transmit PDO 0x1A00**

- Allgemein
 - Name: 1st Transmit PDO mapping
 - Index: 0x1A00
- Flags
 - Alles deaktiviert
- Richtung
 - TxPdo (Eingang): aktiviert
- Ausschließen

Bitte diese Einstellungen beachten, da ansonsten die PDO-Mappings nicht zeitgleich aktiviert werden können!

 - 1A01: deaktiviert
- Einträge

Name	Index	Bitlänge
Status word	0x6041:00	16Bit
Position actual internal value	0x6063:00	32Bit
Position actual value	0x6064:00	32Bit
Torque actual value	0x6077:00	16Bit
Following error actual value	0x60F4:00	32Bit
Modes of operation display	0x6061:00	8Bit
---	---	8Bit
Digital inputs	0x60FD:00	32Bit

Schließen Sie den Dialog *"PDO bearbeiten"* mit [OK].

5. ➔ Wählen Sie das Mapping *"2nd Transmit PDO mapping"* an und klicken Sie auf [Bearbeiten]. Führen Sie folgende Einstellungen durch:

Eingänge: 2nd Transmit PDO 0x1A01

- Allgemein
 - Name: 2nd Transmit PDO mapping
 - Index: 0x1A01
- Flags
 - Alles deaktiviert
- Richtung
 - TxPdo (Eingang): aktiviert
- Ausschließen

Bitte diese Einstellungen beachten, da ansonsten die PDO-Mappings nicht zeitgleich aktiviert werden können!

 - 1A00: deaktiviert
 - 1A02: deaktiviert
 - 1A03: deaktiviert
- Einträge

Name	Index	Bitlänge
Touch probe status	0x60B9:00	16Bit
Touch probe 1 position value	0x60BA:00	32Bit
Touch probe 2 position value	0x60BC:00	32Bit
Velocity actual value	0x606C:00	32Bit

Schließen Sie den Dialog *"PDO bearbeiten"* mit [OK].

6. ➔ Wählen Sie das Mapping *"1st Receive PDO mapping"* an und klicken Sie auf [Bearbeiten]. Führen Sie folgende Einstellungen durch:

Ausgänge: 1st Receive PDO 0x1600

- Allgemein
 - Name: 1st Receive PDO mapping
 - Index: 0x1600
- Flags
 - Alles deaktiviert
- Richtung
 - RxPdo (Ausgang): aktiviert
- Ausschließen

Bitte diese Einstellungen beachten, da ansonsten die PDO-Mappings nicht zeitgleich aktiviert werden können!

 - 1601: deaktiviert
 - 1602: deaktiviert
 - 1603: deaktiviert
- Einträge

Name	Index	Bitlänge
Control word	0x6040:00	16Bit
Target position	0x607A:00	32Bit
Target velocity	0x60FF:00	32Bit
Modes of operation	0x6060:00	8Bit
---	---	8Bit
Touch probe function	0x60B8:00	16Bit

Schließen Sie den Dialog *"PDO bearbeiten"* mit [OK].

7. ➔ Wählen Sie das Mapping *"2nd Receive PDO mapping"* an und klicken Sie auf [Bearbeiten]. Führen Sie folgende Einstellungen durch:

Ausgänge: 2nd Receive PDO 0x1601

- Allgemein
 - Name: 2nd Receive PDO mapping
 - Index: 0x1601
- Flags
 - Alles deaktiviert
- Richtung
 - RxPdo (Ausgang): aktiviert
- Ausschließen

Bitte diese Einstellungen beachten, da ansonsten die PDO-Mappings nicht zeitgleich aktiviert werden können!

 - 1600: deaktiviert
 - 1602: aktiviert
 - 1603: aktiviert
- Einträge

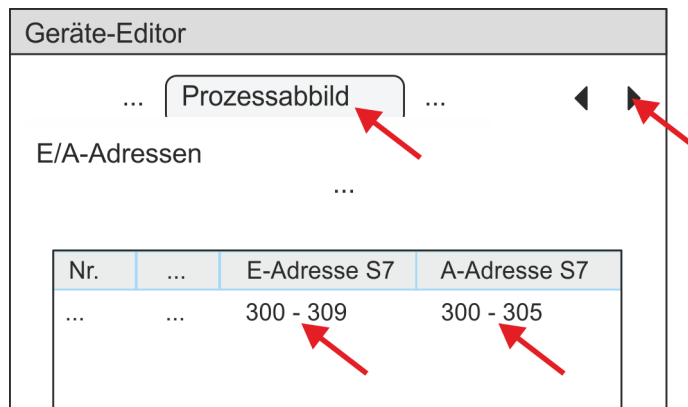
Name	Index	Bitlänge
Profile velocity	0x6081:00	32Bit
Profile acceleration	0x6083:00	32Bit
Profile deceleration	0x6084:00	32Bit

Schließen Sie den Dialog *"PDO bearbeiten"* mit [OK].

8. ➔ Aktivieren Sie in PDO-Zuweisung die PDOs 1 und 2 für die Ein und Ausgänge. Alle nachfolgenden PDOs müssen deaktiviert bleiben. Sollte dies nicht möglich sein, überprüfen Sie bitte den jeweiligen PDO-Parameter *"Ausschließen"*.

9. ➔ Wählen Sie im *"Geräte-Editor"* des *SPEED7 EtherCAT Manager* den Reiter *"Verteilte Uhren"* an und stellen Sie *"DC unused"* als *"Betriebsart"* ein.

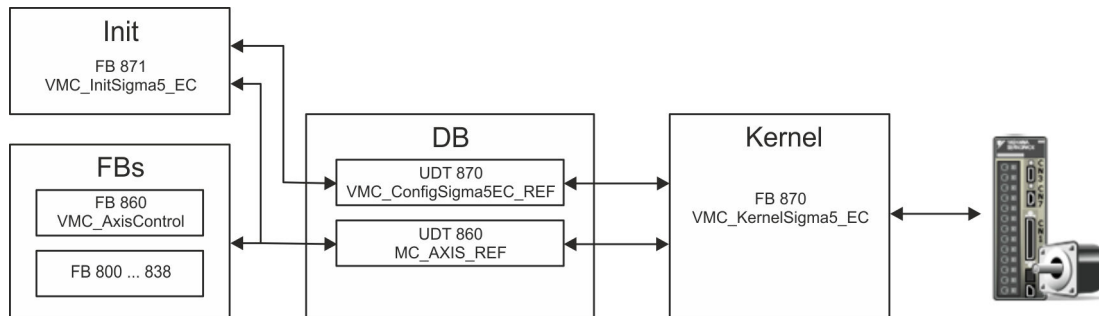
- 10.** ➤ Wählen Sie im "Geräte-Editor" über die Pfeiltaste den Reiter "Prozessabbild" an und notieren Sie sich für die Parameter des Bausteins FB 871 - VMC_InitSigma5_EC folgende PDO-Anfangsadressen:
- "E-Adresse S7" → "InputsStartAddressPDO"
 - "A-Adresse S7" → "OutputsStartAddressPDO"



- 11.** ➤ Indem Sie den Dialog des *SPEED7 EtherCAT Manager* mit [X] schließen, wird die Konfiguration in das *SPEED7 Studio* übernommen.

3.1.3.2 Anwender-Programm

3.1.3.2.1 Programmstruktur



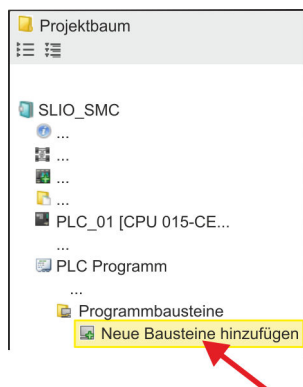
- **DB**

Für jede Achse ist ein Datenbaustein (Achs-DB) für Konfiguration und Statusdaten anzulegen. Der Datenbaustein besteht aus folgenden Datenstrukturen:

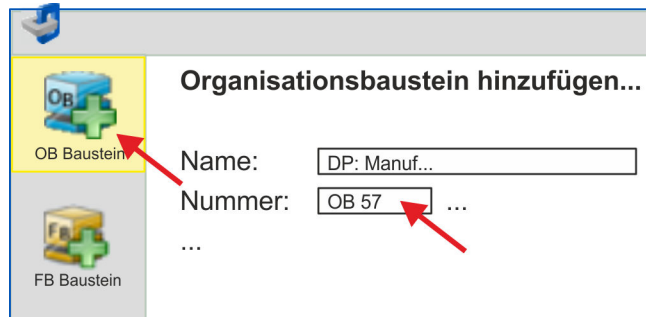
 - UDT 870 - *VMC_ConfigSigma5EC_REF*
Die Datenstruktur beschreibt den Aufbau der Konfiguration des Antriebs. Spezifische Datenstruktur für *Sigma-5* EtherCAT.
 - UDT 860 - *MC_AXIS_REF*
Die Datenstruktur beschreibt den Aufbau der Parameter und Statusinformationen von Antrieben. Allgemeine Datenstruktur für alle Antriebe und Bussysteme.
- **FB 871 - VMC_InitSigma5_EC**
 - Der *Init*-Baustein dient zur Konfiguration einer Achse.
 - Spezifischer Baustein für *Sigma-5* EtherCAT.
 - Die Konfigurationsdaten für die Initialisierung sind im *Achs-DB* abzulegen.
- **FB 870 - VMC_KernelSigma5_EC**
 - Der *Kernel*-Baustein kommuniziert mit dem Antrieb über das entsprechende Bussystem, verarbeitet die Benutzeraufträge und liefert Statusmeldungen zurück.
 - Spezifischer Baustein für *Sigma-5* EtherCAT.
 - Der Austausch der Daten erfolgt mittels des *Achs-DB*.
- **FB 860 - VMC_AxisControl**
 - Universal-Baustein für alle Antriebe und Bussysteme.
 - Unterstützt einfache Bewegungskommandos und liefert alle relevanten Statusmeldungen.
 - Der Austausch der Daten erfolgt mittels des *Achs-DB*.
 - Über die Instanzdaten des Bausteins können Sie zur Bewegungssteuerung und Statusabfrage eine Visualisierung anbinden.
 - Zusätzlich zum FB 860 - *VMC_AxisControl* haben Sie die Möglichkeit *PLCopen*-Bausteine zu nutzen.
- **FB 800 ... FB 838 - PLCopen**
 - Die *PLCopen*-Bausteine dienen zur Programmierung von Bewegungsabläufen und Statusabfragen.
 - Allgemeine Bausteine für alle Antriebe und Bussysteme.

3.1.3.2.2 Programmierung

Bausteine in Projekt kopieren

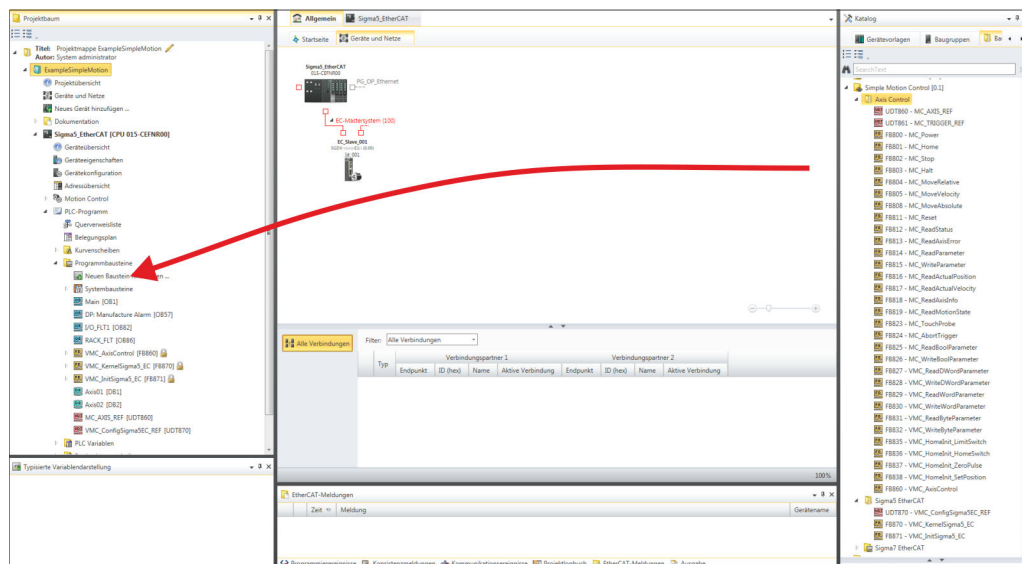


1. Klicken Sie im *Projektbaum* innerhalb der CPU unter "*PLC-Programm*", "*Programmbausteine*" auf "*Neuen Baustein hinzufügen*".



⇒ Das Dialogfenster "*Baustein hinzufügen*" öffnet sich.

2. Wählen Sie den Bausteintyp "*OB Baustein*" und fügen Sie nacheinander OB 57, OB 82 und OB 86 Ihrem Projekt hinzu.



3. Öffnen Sie im "*Katalog*" unter "*Bausteine*" "*Simple Motion Control*" und ziehen Sie per Drag&Drop folgende Bausteine in "*Programmbausteine*" des *Projektbaums*:

- Sigma-5 EtherCAT:
 - UDT 870 - VMC_ConfigSigma5EC_REF
 - FB 870 - VMC_KernelSigma5_EC
 - FB 871 - VMC_InitSigma5_EC
- Axis Control
 - UDT 860 - MC_AXIS_REF
 - Bausteine für die gewünschten Bewegungsabläufe

Achs-DB anlegen

1. Fügen Sie Ihrem Projekt einen neuen DB als *Achs-DB* hinzu. Klicken Sie hierzu im *Projektbaum* innerhalb der CPU unter "*PLC-Programm*", "*Programmbausteine*" auf "*Neuen Baustein hinzufügen*", wählen Sie den Bausteintyp "*DB Baustein*" und vergeben Sie diesem den Namen "*Axis01*". Die DB-Nr. können Sie frei wählen wie z.B. DB 10.

⇒ Der Baustein wird angelegt und geöffnet.

2. ➔ ■ Legen Sie in "Axis01" die Variable "Config" vom Typ UDT 870 an. Dies sind spezifische Achs-Konfigurationsdaten.
- Legen Sie in "Axis01" die Variable "Axis" vom Typ UDT 860 an. Während des Betriebs werden hier alle Betriebsdaten der Achse abgelegt.

Axis01 [DB10]
Bausteinstruktur

	Adr...	Name	Datentyp	...
	...	Config	UDT	[870]
	...	Axis	UDT	[860]

OB 1

Konfiguration der Achse

Öffnen Sie den OB 1 und programmieren Sie folgende FB-Aufrufe mit zugehörigen DBs:

- ➔ FB 871 - VMC_InitSigma5_EC, DB 871 ↗ Kapitel 3.1.5.3 "FB 871 - VMC_InitSigma5_EC - Sigma-5 EtherCAT Initialisierung" auf Seite 45

Geben Sie unter *InputsStartAddressPDO* bzw. *OutputsStartAddressPDO* die Adresse aus dem *SPEED7 EtherCAT Manager* an. ↗ 21

```
⇒ CALL "VMC_InitSigma5_EC" , "DI_InitSgm5ETC01"
   Enable           := "InitS5EC1_Enable"
   LogicalAddress    := 300
   InputsStartAddressPDO := 300 (EtherCAT-Man.: E-Adresse S7)
   OutputsStartAddressPDO := 300 (EtherCAT-Man.: A-Adresse S7)
   EncoderType       := 1
   EncoderResolutionBits := 20
   FactorPosition     := 1.048576e+006
   FactorVelocity     := 1.048576e+006
   FactorAcceleration := 1.048576e+002
   OffsetPosition     := 0.000000e+000
   MaxVelocityApp     := 5.000000e+001
   MaxAccelerationApp := 1.000000e+002
   MaxDecelerationApp := 1.000000e+002
   MaxVelocityDrive   := 6.000000e+001
   MaxAccelerationDrive := 1.500000e+002
   MaxDecelerationDrive := 1.500000e+002
   MaxPosition        := 1.048500e+003
   MinPosition        := -1.048514e+003
   EnableMaxPosition  := TRUE
   EnableMinPosition  := TRUE
   MinUserPosition    := "InitS5EC1_MinUserPos"
   MaxUserPosition    := "InitS5EC1_MaxUserPos"
   Valid              := "InitS5EC1_Valid"
   Error               := "InitS5EC1_Error"
   ErrorID             := "InitS5EC1_ErrorID"
   Config              := "Axis01".Config
   Axis                := "Axis01".Axis
```

Kernel für Achse beschalten

Der *Kernel* verarbeitet die Benutzerkommandos und gibt sie entsprechend aufbereitet an den Antrieb über das jeweilige Bussystem weiter.

- ➔ FB 870 - VMC_KernelSigma5_EC, DB 870 ↗ Kapitel 3.1.5.2 "FB 870 - VMC_KernelSigma5_EC - Sigma-5 EtherCAT Kernel" auf Seite 45

```
⇒ CALL "VMC_KernelSigma5_EC" , "DI_KernelSgm5ETC01"
   Init := "KernelS5EC1_Init"
   Config := "Axis01".Config
   Axis := "Axis01".Axis
```

Baustein für Bewegungsabläufe beschalten

Zur Vereinfachung soll hier die Beschaltung des FB 860 - VMC_AxisControl gezeigt werden. Dieser Universalbaustein unterstützt einfache Bewegungskommandos und liefert Statusmeldungen zurück. Die Ein- und Ausgänge können Sie individuell beschalten. Bitte geben Sie unter "Axis" die Referenz zu den entsprechenden Achsdaten im *Achs-DB* an.

➔ FB 860 - VMC_AxisControl, DB 860 ↗ *Kapitel 3.4.2.2 "FB 860 - VMC_AxisControl - Control-Baustein Achskontrolle" auf Seite 132*

```
⇒ CALL "VMC_AxisControl" , "DI_AxisControl01"
    AxisEnable      := "AxCtrl1_AxisEnable"
    AxisReset       := "AxCtrl1_AxisReset"
    HomeExecute     := "AxCtrl1_HomeExecute"
    HomePosition    := "AxCtrl1_HomePosition"
    StopExecute     := "AxCtrl1_StopExecute"
    MvVelocityExecute := "AxCtrl1_MvVelExecute"
    MvRelativeExecute := "AxCtrl1_MvRelExecute"
    MvAbsoluteExecute := "AxCtrl1_MvAbsExecute"
    PositionDistance := "AxCtrl1_PositionDistance"
    Velocity        := "AxCtrl1_Velocity"
    Acceleration     := "AxCtrl1_Acceleration"
    Deceleration     := "AxCtrl1_Deceleration"
    JogPositive      := "AxCtrl1_JogPositive"
    JogNegative      := "AxCtrl1_JogNegative"
    JogVelocity      := "AxCtrl1_JogVelocity"
    JogAcceleration  := "AxCtrl1_JogAcceleration"
    JogDeceleration  := "AxCtrl1_JogDeceleration"
    AxisReady        := "AxCtrl1_AxisReady"
    AxisEnabled      := "AxCtrl1_AxisEnabled"
    AxisError        := "AxCtrl1_AxisError"
    AxisErrorID      := "AxCtrl1_AxisErrorID"
    DriveWarning     := "AxCtrl1_DriveWarning"
    DriveError       := "AxCtrl1_DriveError"
    DriveErrorID     := "AxCtrl1_DriveErrorID"
    IsHomed          := "AxCtrl1_IsHomed"
    ModeOfOperation  := "AxCtrl1_ModeOfOperation"
    PLCOpenState     := "AxCtrl1_PLCOpenState"
    ActualPosition   := "AxCtrl1_ActualPosition"
    ActualVelocity   := "AxCtrl1_ActualVelocity"
    CmdDone          := "AxCtrl1_CmdDone"
    CmdBusy          := "AxCtrl1_CmdBusy"
    CmdAborted       := "AxCtrl1_CmdAborted"
    CmdError         := "AxCtrl1_CmdError"
    CmdErrorID       := "AxCtrl1_CmdErrorID"
    DirectionPositive := "AxCtrl1_DirectionPos"
    DirectionNegative := "AxCtrl1_DirectionNeg"
    SWLimitMinActive := "AxCtrl1_SWLimitMinActive"
    SWLimitMaxActive := "AxCtrl1_SWLimitMaxActive"
    HWLimitMinActive := "AxCtrl1_HWLimitMinActive"
    HWLimitMaxActive := "AxCtrl1_HWLimitMaxActive"
    Axis             := "Axis01".Axis
```



Für komplexe Bewegungsaufgaben können Sie die PLCOpen-Bausteine verwenden. Hier müssen Sie ebenfalls unter Axis die Referenz zu den Achsdaten im Achs-DB angeben.

Ihr Projekt beinhaltet nun folgende Bausteine:

- OB 1 - Main
- OB 57 - DP Manufacturer Alarm
- OB 82 - I/O_FLT1
- OB 86 - Rack_FLT
- FB 860 - VMC_AxisControl mit Instanz-DB

- FB 870 - VMC_KernelSigma5_EC mit Instanz-DB
- FB 871 - VMC_InitSigma5_EC mit Instanz-DB
- UDT 860 - MC_Axis_REF
- UDT 870 - VMC_ConfigSigma5EC_REF

Zeitlicher Ablauf

1. ➤ Wählen Sie *"Projekt ➔ Alles übersetzen"* und übertragen Sie das Projekt in Ihre CPU. Näheres zur Übertragung Ihres Projekt finden Sie in der Onlinehilfe zum *SPEED7 Studio*.

⇒ Sie können jetzt Ihre Applikation in Betrieb nehmen.



VORSICHT!

Bitte beachten Sie immer die Sicherheitshinweise zu ihrem Antrieb, insbesondere bei der Inbetriebnahme!

2. ➤ Bevor eine Achse gesteuert werden kann, muss diese initialisiert werden. Rufen Sie hierzu den *Init*-Baustein FB 871 - VMC_InitSigma5_EC mit *Enable* = TRUE auf.

⇒ Der Ausgang *Valid* meldet TRUE zurück. Im Fehlerfall können Sie durch Auswertung der *ErrorID* den Fehler ermitteln.

Den *Init*-Baustein müssen Sie erneut aufrufen, wenn Sie einen neuen Achs-DB laden oder Parameter am *Init*-Baustein geändert wurden.



Fahren Sie erst fort, wenn der Init-Baustein keinen Fehler meldet!

3. ➤ Stellen Sie sicher, dass der *Kernel*-Baustein FB 870 - VMC_KernelSigma5_EC zyklisch aufgerufen wird. Auf diese Weise werden Steuersignale an den Antrieb übergeben und Statusmeldungen übermittelt.
4. ➤ Programmieren Sie Ihre Applikation mit dem FB 860 - VMC_AxisControl oder mit den PLCopen Bausteinen auf.

Steuerung des Antriebs über HMI

Sie haben die Möglichkeit über ein HMI Ihren Antrieb zu steuern. Hierzu gibt es für Movicon eine vorgefertigte Symbolbibliothek für den Zugriff auf den VMC_AxisControl Funktionsbaustein. ➔ *Kapitel 7 "Antrieb über HMI steuern" auf Seite 313*

3.1.4 Einsatz im Siemens SIMATIC Manager

3.1.4.1 Voraussetzung

Übersicht

- Bitte verwenden Sie für die Projektierung den Siemens SIMATIC Manager ab V 5.5 SP2.
- Die Projektierung der System SLIO CPU erfolgt im Siemens SIMATIC Manager in Form des virtuellen PROFINET IO Devices *"VIPA SLIO CPU"*. Das *"VIPA SLIO System"* ist mittels GSDML im Hardware-Katalog zu installieren.
- Die Projektierung des EtherCAT-Masters erfolgt im Siemens SIMATIC Manager in Form des virtuellen PROFINET IO Devices *"EtherCAT-Netzwerk"*. Das *"EtherCAT-Netzwerk"* ist mittels GSDML im Hardware-Katalog zu installieren.
- Das *"EtherCAT-Netzwerk"* kann mit dem VIPA-Tool *SPEED7 EtherCAT Manager* konfiguriert werden.
- Für die Projektierung des Antriebs im *SPEED7 EtherCAT Manager* ist die Installation der zugehörigen ESI-Datei erforderlich.

IO Device "VIPA SLIO System" installieren

Die Installation des PROFINET IO Devices "VIPA SLIO CPU" im Hardware-Katalog erfolgt nach folgender Vorgehensweise:

1. ➤ Gehen Sie in den Service-Bereich von www.vipa.com.
2. ➤ Laden Sie aus dem Downloadbereich unter "*Config Dateien* ➔ *PROFINET*" die Konfigurationsdatei für Ihre CPU.
3. ➤ Extrahieren Sie die Datei in Ihr Arbeitsverzeichnis.
4. ➤ Starten Sie den Hardware-Konfigurator von Siemens.
5. ➤ Schließen Sie alle Projekte.
6. ➤ Gehen Sie auf "*Extras* ➔ *GSD-Dateien installieren*".
7. ➤ Navigieren Sie in Ihr Arbeitsverzeichnis und installieren Sie die entsprechende GSDML-Datei.
 - ⇒ Nach der Installation finden Sie das entsprechende PROFINET IO Device unter "*PROFINET IO* ➔ *Weitere Feldgeräte* ➔ *I/O* ➔ *VIPA SLIO System*".

IO Device EtherCAT-Netzwerk installieren

Die Installation des PROFINET IO Devices "EtherCAT-Netzwerk" im Hardware-Katalog erfolgt nach folgender Vorgehensweise:

1. ➤ Gehen Sie in den Service-Bereich von www.vipa.com
2. ➤ Laden Sie aus dem Downloadbereich unter "*Config Dateien* ➔ *EtherCAT*" die GSDML-Datei für Ihren EtherCAT-Master.
3. ➤ Extrahieren Sie die Dateien in Ihr Arbeitsverzeichnis.
4. ➤ Starten Sie den Hardware-Konfigurator von Siemens.
5. ➤ Schließen Sie alle Projekte.
6. ➤ Gehen Sie auf "*Extras* ➔ *GSD-Dateien installieren*".
7. ➤ Navigieren Sie in Ihr Arbeitsverzeichnis und installieren Sie die entsprechende GSDML-Datei.
 - ⇒ Nach der Installation finden Sie das "EtherCAT-Netzwerk" unter "*PROFINET IO* ➔ *Weitere Feldgeräte* ➔ *I/O* ➔ *VIPA EtherCAT System*".

SPEED7 EtherCAT Manager installieren

Die Konfiguration des PROFINET IO Devices "EtherCAT-Netzwerk" erfolgt mit dem *SPEED7 EtherCAT Manager* von VIPA. Sie finden diesen Im Servicebereich von www.vipa.com unter "*Service/Support* ➔ *Downloads* ➔ *SPEED7*".

Die Installation erfolgt nach folgender Vorgehensweise:

1. ➤ Schließen Sie den Siemens SIMATIC Manager.
2. ➤ Gehen Sie in den Service-Bereich von www.vipa.com
3. ➤ Laden Sie den *SPEED7 EtherCAT Manager* und entpacken Sie diesen auf Ihren PC.
4. ➤ Zur Installation starten Sie die Datei *EtherCATManager_v... .exe*.
5. ➤ Wählen Sie die Sprache für die Installation aus.
6. ➤ Stimmen Sie dem Lizenzvertrag zu.
7. ➤ Wählen Sie das Installationsverzeichnis und starten Sie die Installation.
8. ➤ Nach der Installation müssen Sie Ihren PC neu starten
 - ⇒ Der *SPEED7 EtherCAT Manager* ist installiert und kann jetzt über das Kontextmenü des Siemens SIMATIC Manager aufgerufen werden.

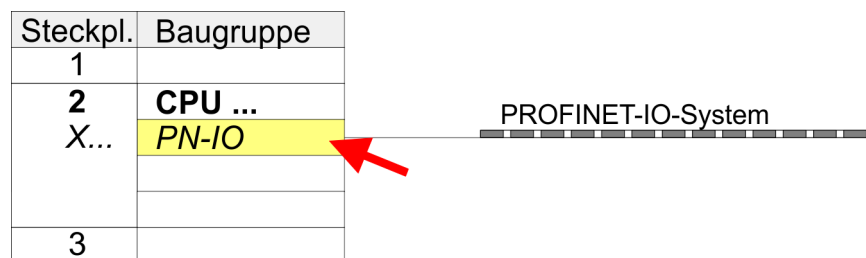
3.1.4.2 Hardware-Konfiguration

CPU im Projekt anlegen

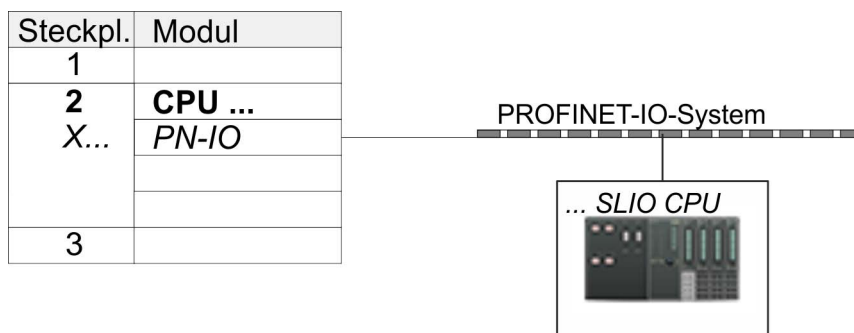
Steckp..	Baugruppe
1	
2	CPU 315-2 PN/DP
X1	MPI/DP
X2	PN-IO
X2...	Port 1
X2...	Port 2
3	

Um kompatibel mit dem Siemens SIMATIC Manager zu sein, sind folgende Schritte durchzuführen:

1. Starten Sie den Hardware-Konfigurator von Siemens mit einem neuen Projekt.
2. Fügen Sie aus dem Hardware-Katalog eine Profilschiene ein.
3. Platzieren Sie auf "Slot"-Nummer 2 die CPU 315-2 PN/DP (6ES7 315-2EH14 V3.2).
4. Über das Submodul "X1 MPI/DP" projektieren und vernetzen Sie den integrierten PROFIBUS-DP-Master (Buchse X3).
5. Über das Submodul "X2 PN-IO" projektieren Sie den EtherCAT-Master als virtuelles PROFINET-Netzwerk.
6. Klicken Sie auf das Submodul "PN-IO" der CPU.
7. Wählen Sie "Kontextmenü → PROFINET IO-System einfügen".



8. Legen Sie mit [Neu] ein neues Subnetz an und vergeben Sie gültige IP-Adress-Daten
9. Klicken Sie auf das Submodul "PN-IO" der CPU und öffnen Sie mit "Kontextmenü → Objekteigenschaften" den Eigenschafts-Dialog.
10. Geben Sie unter "Allgemein" einen "Gerätenamen" an. Der Geräte name muss eindeutig am Ethernet-Subnetz sein.



Steckpl.	Baugruppe	Bestellnummer	
0	... SLIO CPU ...	015-...	
X2	015-...		
1			
2			
3			
...			

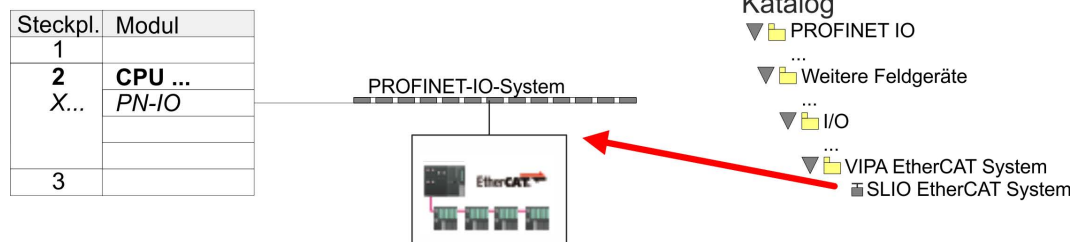
11. Navigieren Sie im Hardware-Katalog in das Verzeichnis **"PROFINET IO → Weitere Feldgeräte → I/O → VIPA SLIO System"** und binden Sie das IO-Device **"015-CEFNR00 CPU"** an Ihr PROFINET-System an.
- ⇒ In der Steckplatzübersicht des PROFINET-IO-Device **"VIPA SLIO CPU"** ist auf Steckplatz 0 die CPU bereits vorplatziert. Ab Steckplatz 1 können Sie Ihre System SLIO Module platzieren.

Ethernet-PG/OP-Kanal parametrieren

Steckpl.	Modul
1	
2	CPU ...
X...	PN-IO
3	
4	343-1EX30
5	
...	

- Platzieren Sie für den Ethernet-PG/OP-Kanal auf Steckplatz 4 den Siemens CP 343-1 (SIMATIC 300 \ CP 300 \ Industrial Ethernet \ CP 343-1 \ 6GK7 343-1EX30 0XE0 V3.0).
- Öffnen Sie durch Doppelklick auf den CP 343-1EX30 den Eigenschaften-Dialog und geben Sie für den CP unter **"Eigenschaften"** IP-Adress-Daten an. Gültige IP-Adress-Parameter erhalten Sie von Ihrem Systemadministrator.
- Ordnen Sie den CP einem **"Subnetz"** zu. Ohne Zuordnung werden die IP-Adress-Daten nicht übernommen!

"EtherCAT-Netzwerk" einfügen

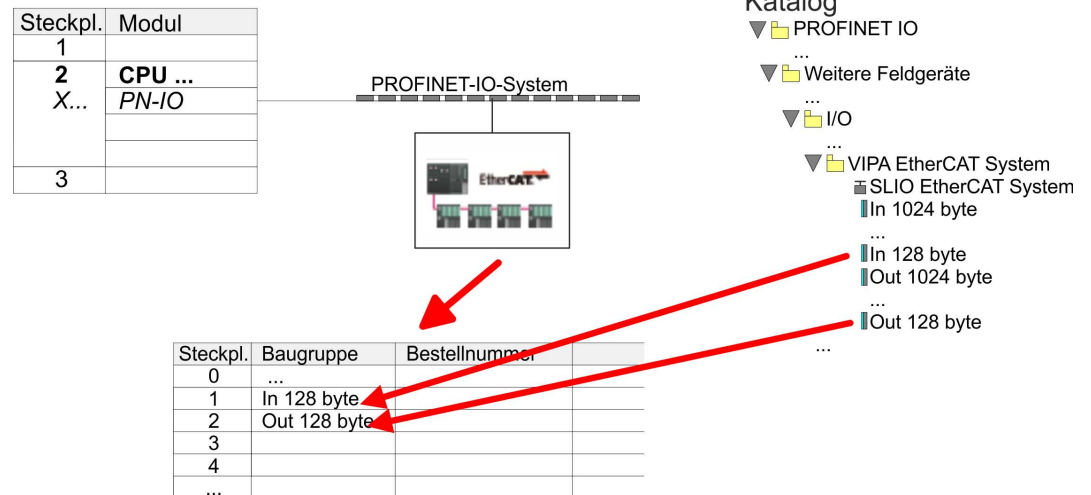


1. Navigieren Sie im Hardware-Katalog in das Verzeichnis **"PROFINET IO → Weitere Feldgeräte → I/O → VIPA EtherCAT System"** und binden Sie das IO Device **"SLIO EtherCAT System"** an Ihr PROFINET-System an.

2. ➔ Klicken Sie auf das eingefügte IO Device *"EtherCAT-Netzwerk"* und definieren Sie die Bereiche für Ein- und Ausgabe, indem Sie den entsprechenden *"Out"*- bzw. *"In"*-Bereich auf einen Steckplatz ziehen.

Legen Sie folgende Bereiche an:

- In 128Byte
- Out 128Byte



3. ➔ Wählen Sie *"Station → Speichern und übersetzen"*

Sigma-5 EtherCAT Antrieb konfigurieren

Die Konfiguration des Antriebs erfolgt im *SPEED7 EtherCAT Manager*.



Vor dem Aufruf des **SPEED7 EtherCAT Manager** müssen Sie immer Ihr Projekt mit "Station → Speichern und übersetzen" speichern.

Steckpl.	Modul
1	
2	CPU ...
X...	PN-IO
3	

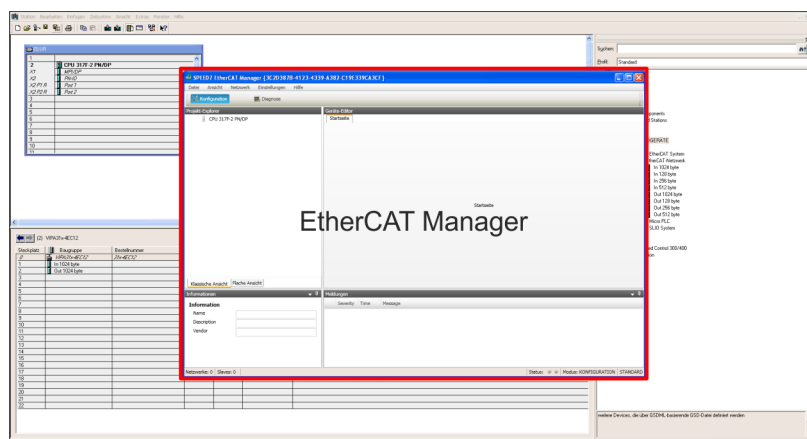
PROFINET-IO-System



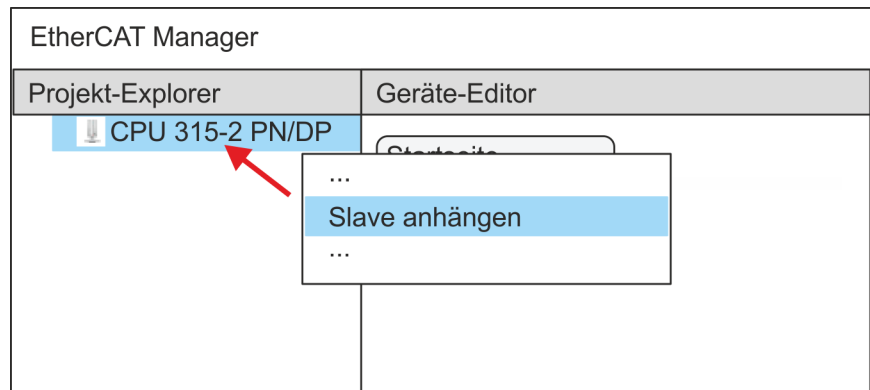
1. Klicken Sie auf das eingefügte IO Device "EtherCAT-Netzwerk" und wählen Sie "Kontextmenü → Device Tool starten → **SPEED7 EtherCAT Manager**".

⇒ Der **SPEED7 EtherCAT Manager** wird gestartet. Hier können Sie die EtherCAT-Kommunikation zu Ihrem **Sigma-5** Antrieb konfigurieren.

Näheres zum Einsatz des **SPEED7 EtherCAT Manager** finden Sie im zugehörigen Handbuch bzw. in der Onlinehilfe.



3. Damit der **Sigma-5** EtherCAT Antrieb im **SPEED7 EtherCAT Manager** konfiguriert werden kann, ist die entsprechende ESI-Datei zu installieren. Die ESI-Datei für den **Sigma-5** EtherCAT Antrieb finden Sie unter www.yaskawa.eu.com unter "Service → Drives & Motion Software". Laden Sie die zu Ihrem Antrieb passende ESI-Datei herunter. Entpacken Sie diese falls erforderlich.
 4. Öffnen Sie im **SPEED7 EtherCAT Manager** über "Datei → ESI-Verwaltung" das Dialogfenster "ESI-Manager".
 5. Klicken Sie im "ESI-Manager" auf [Datei hinzufügen] und wählen Sie Ihre ESI-Datei aus. Mit [Öffnen] wird die ESI-Datei im **SPEED7 EtherCAT Manager** installiert.
 6. Schließen Sie den "ESI-Manager".
- ⇒ Ihr **Sigma-5** EtherCAT Antrieb steht Ihnen nun zur Konfiguration zur Verfügung.



7. Klicken Sie im EtherCAT Manager auf ihre CPU und öffnen Sie über "Kontextmenü → Slave anhängen" das Dialogfenster zum Hinzufügen eines EtherCAT-Slave.

⇒ Das Dialogfenster zur Auswahl eines EtherCAT-Slave wird geöffnet.

8. Wählen Sie Ihren Sigma-5 EtherCAT Antrieb und bestätigen Sie Ihre Auswahl mit [OK].

⇒ Der Sigma-5 EtherCAT Antrieb wird an den Master angebunden und kann nun konfiguriert werden.

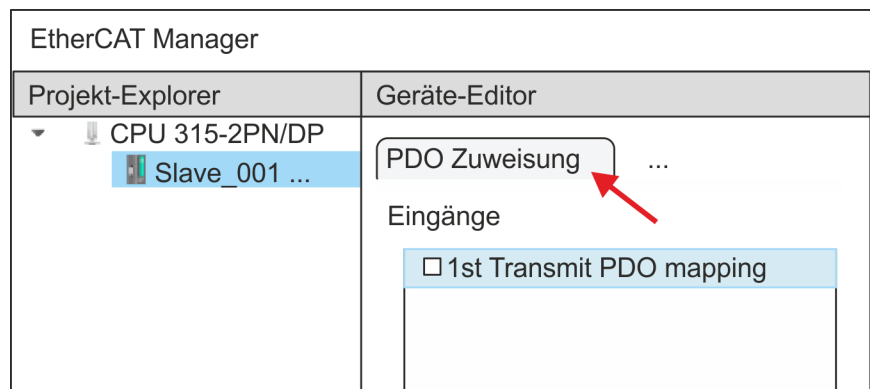
- 9.



PDOs können Sie nur im "Experten-Modus" bearbeiten! Ansonsten werden die Schaltflächen ausgeblendet. Durch Aktivierung des "Experten-Modus" können Sie in die erweiterte Bearbeitung umschalten.

Aktivieren Sie den Experten-Modus durch Aktivierung von "Ansicht → Experte".

10. Klicken Sie im SPEED7 EtherCAT Manager auf den Sigma-5 EtherCAT Slave und wählen Sie im "Geräte-Editor" den Reiter "PDO-Zuweisung" an.



⇒ Dieser Dialog zeigt eine Auflistung aller PDOs.

Geräte-Editor

PDO Zuweisung ...

Eingänge	Ausgänge
<input type="checkbox"/> 1st Transmit PDO mapping	<input type="checkbox"/> 1st Receive PDO mapping
<input type="checkbox"/> 2nd Transmit PDO mapping	<input type="checkbox"/> 2nd Receive PDO mapping

... Bearbeiten ...

- 11.** Durch Anwahl des entsprechenden PDO-Mappings können Sie mit [Bearbeiten] die PDOs bearbeiten. Wählen Sie das Mapping "1st Transmit PDO mapping" an und klicken Sie auf [Bearbeiten].



Bitte beachten Sie, dass aufgrund der Voreinstellung manche PDOs nicht bearbeitet werden können. Durch Deaktivierung bereits aktivierter PDOs können Sie die Bearbeitung von gesperrten PDOs frei geben.

Eintrag: 1st Transmit PDO mapping

Index: 0x1A00 (Dez) Hex

Flags:

- ☐ Zwingend
- ☐ Schreibgeschützt
- ☐ Virtuell

Richtung:

- ☒ TxPdo (Eingang)
- ☐ RxPdo (Ausgang)

Optional:

Ausschließen:

- ☐ 1A01
- ☒ 1A02
- ☒ 1A03

Name	Index	Bitlänge	Kommentar
Status word	0x6041:00	16	
Position actual internal value	0x6063:00	32	
Position actual value	0x6064:00	32	
Torque actual value	0x6077:00	16	
Following error actual value	0x60F4:00	32	
Modes of operation display	0x6061:00	8	
---	---	8	---
Digital inputs	0x60FD:00	32	

Neu Löschen Bearbeiten Nach oben Nach unten

- ⇒ Es öffnet sich der Dialog *"PDO bearbeiten"*. Bitte überprüfen Sie hier die aufgeführten PDO-Einstellungen und passen Sie diese ggf. an. Bitte berücksichtigen Sie hierbei auch die Reihenfolge der *"Einträge"* und ergänzen Sie diese entsprechend.

Für die Bearbeitung der *"Einträge"* stehen folgende Funktionen zur Verfügung:

- Neu
 - Hiermit können Sie in einem Dialogfenster einen neuen Eintrag anlegen, indem Sie aus dem *"CoE-Objektverzeichnis"* den entsprechenden Eintrag auswählen und Ihre Einstellungen vornehmen. Mit [OK] wird der Eintrag übernommen und in der Liste der Einträge aufgeführt.
- Löschen
 - Hiermit können Sie den angewählte Eintrag löschen.
- Bearbeiten
 - Hiermit können Sie allgemeinen Daten eines Eintrags bearbeiten.
- Nach oben/unten
 - Hiermit können Sie den angewählten Eintrag in der List nach oben bzw. nach unten bewegen.

12. Führen Sie folgende Einstellungen durch:

Eingänge: 1st Transmit PDO 0x1A00

- Allgemein
 - Name: 1st Transmit PDO mapping
 - Index: 0x1A00
- Flags
 - Alles deaktiviert
- Richtung
 - TxPdo (Eingang): aktiviert
- Ausschließen

Bitte diese Einstellungen beachten, da ansonsten die PDO-Mappings nicht zeitgleich aktiviert werden können!

 - 1A01: deaktiviert
- Einträge

Name	Index	Bitlänge
Status word	0x6041:00	16Bit
Position actual internal value	0x6063:00	32Bit
Position actual value	0x6064:00	32Bit
Torque actual value	0x6077:00	16Bit
Following error actual value	0x60F4:00	32Bit
Modes of operation display	0x6061:00	8Bit
---	---	8Bit
Digital inputs	0x60FD:00	32Bit

Schließen Sie den Dialog *"PDO bearbeiten"* mit [OK].

- 13.** Wählen Sie das Mapping *"2nd Transmit PDO mapping"* an und klicken Sie auf [Bearbeiten]. Führen Sie folgende Einstellungen durch:

Eingänge: 2nd Transmit PDO 0x1A01

- Allgemein
 - Name: 2nd Transmit PDO mapping
 - Index: 0x1A01
- Flags
 - Alles deaktiviert
- Richtung
 - TxPdo (Eingang): aktiviert
- Ausschließen

Bitte diese Einstellungen beachten, da ansonsten die PDO-Mappings nicht zeitgleich aktiviert werden können!

 - 1A00: deaktiviert
 - 1A02: deaktiviert
 - 1A03: deaktiviert
- Einträge

Name	Index	Bitlänge
Touch probe status	0x60B9:00	16Bit
Touch probe 1 position value	0x60BA:00	32Bit
Touch probe 2 position value	0x60BC:00	32Bit
Velocity actual value	0x606C:00	32Bit

Schließen Sie den Dialog *"PDO bearbeiten"* mit [OK].

- 14.** ➤ Wählen Sie das Mapping *"1st Receive PDO mapping"* an und klicken Sie auf [Bearbeiten]. Führen Sie folgende Einstellungen durch:

Ausgänge: 1st Receive PDO 0x1600

- Allgemein
 - Name: 1st Receive PDO mapping
 - Index: 0x1600
- Flags
 - Alles deaktiviert
- Richtung
 - RxPdo (Ausgang): aktiviert
- Ausschließen

Bitte diese Einstellungen beachten, da ansonsten die PDO-Mappings nicht zeitgleich aktiviert werden können!

 - 1601: deaktiviert
 - 1602: deaktiviert
 - 1603: deaktiviert
- Einträge

Name	Index	Bitlänge
Control word	0x6040:00	16 Bit
Target position	0x607A:00	32 Bit
Target velocity	0x60FF:00	32 Bit
Modes of operation	0x6060:00	8 Bit
---	---	8 Bit
Touch probe function	0x60B8:00	16 Bit

Schließen Sie den Dialog *"PDO bearbeiten"* mit [OK].

15. Wählen Sie das Mapping *"2nd Receive PDO mapping"* an und klicken Sie auf [Bearbeiten]. Führen Sie folgende Einstellungen durch:

Ausgänge: 2nd Receive PDO 0x1601

- Allgemein
 - Name: 2nd Receive PDO mapping
 - Index: 0x1601
- Flags
 - Alles deaktiviert
- Richtung
 - RxPdo (Ausgang): aktiviert
- Ausschließen

Bitte diese Einstellungen beachten, da ansonsten die PDO-Mappings nicht zeitgleich aktiviert werden können!

 - 1600: deaktiviert
 - 1602: aktiviert
 - 1603: aktiviert
- Einträge

Name	Index	Bitlänge
Profile velocity	0x6081:00	32Bit
Profile acceleration	0x6083:00	32Bit
Profile deceleration	0x6084:00	32Bit

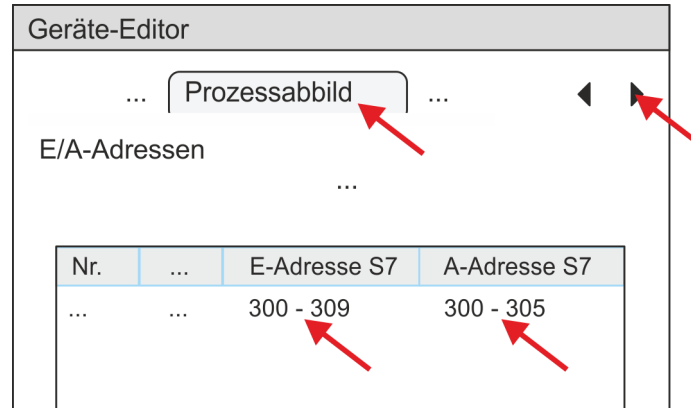
Schließen Sie den Dialog *"PDO bearbeiten"* mit [OK].

16. Aktivieren Sie in PDO-Zuweisung die PDOs 1 und 2 für die Ein und Ausgänge. Alle nachfolgenden PDOs müssen deaktiviert bleiben. Sollte dies nicht möglich sein, überprüfen Sie bitte den jeweiligen PDO-Parameter *"Ausschließen"*.

17. Wählen Sie im *"Geräte-Editor"* des *SPEED7 EtherCAT Manager* den Reiter *"Verteilte Uhren"* an und stellen Sie *"DC unused"* als *"Betriebsart"* ein.

18. Wählen Sie im "Geräte-Editor" über die Pfeiltaste den Reiter "Prozessabbild" an und notieren Sie sich für die Parameter des Bausteins FB 871 - VMC_InitSigma5_EC folgende PDO-Anfangsadressen:

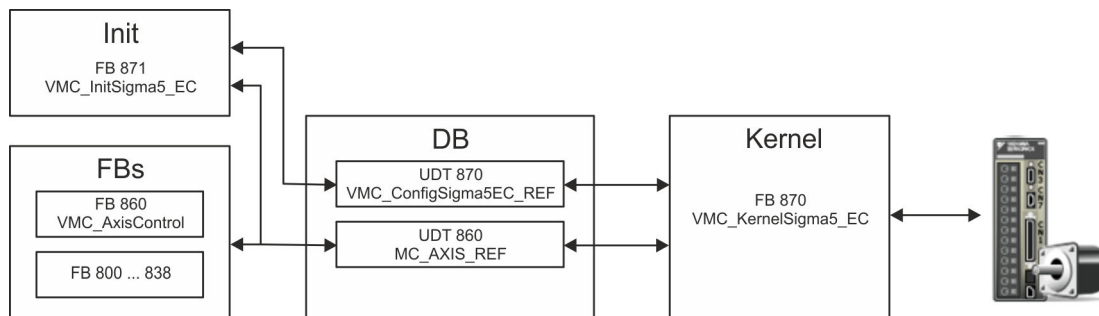
- "E-Adresse S7" → "InputsStartAddressPDO"
- "A-Adresse S7" → "OutputsStartAddressPDO"



19. Indem Sie den Dialog des *SPEED7 EtherCAT Manager* mit [X] schließen, wird die Konfiguration in die Projektierung übernommen. Sie können Ihre EtherCAT-Konfiguration jederzeit im *SPEED7 EtherCAT Manager* wieder bearbeiten, da die Konfiguration in Ihrem Projekt gespeichert wird.
20. Speichern und übersetzen Sie Ihre Konfiguration

3.1.4.3 Anwender-Programm

3.1.4.3.1 Programmstruktur



- DB

Für jede Achse ist ein Datenbaustein (Achs-DB) für Konfiguration und Statusdaten anzulegen. Der Datenbaustein besteht aus folgenden Datenstrukturen:

 - UDT 870 - *VMC_ConfigSigma5EC_REF*
Die Datenstruktur beschreibt den Aufbau der Konfiguration des Antriebs. Spezifische Datenstruktur für *Sigma-5* EtherCAT.
 - UDT 860 - *MC_AXIS_REF*
Die Datenstruktur beschreibt den Aufbau der Parameter und Statusinformationen von Antrieben. Allgemeine Datenstruktur für alle Antriebe und Bussysteme.
- FB 871 - *VMC_InitSigma5_EC*
 - Der *Init*-Baustein dient zur Konfiguration einer Achse.
 - Spezifischer Baustein für *Sigma-5* EtherCAT.
 - Die Konfigurationsdaten für die Initialisierung sind im *Achs-DB* abzulegen.

- FB 870 - *VMC_KernelSigma5_EC*
 - Der *Kernel*-Baustein kommuniziert mit dem Antrieb über das entsprechende Bussystem, verarbeitet die Benutzeraufträge und liefert Statusmeldungen zurück.
 - Spezifischer Baustein für *Sigma-5* EtherCAT.
 - Der Austausch der Daten erfolgt mittels des *Achs-DB*.
- FB 860 - *VMC_AxisControl*
 - Universal-Baustein für alle Antriebe und Bussysteme.
 - Unterstützt einfache Bewegungskommandos und liefert alle relevanten Statusmeldungen.
 - Der Austausch der Daten erfolgt mittels des *Achs-DB*.
 - Über die Instanzdaten des Bausteins können Sie zur Bewegungssteuerung und Statusabfrage eine Visualisierung anbinden.
 - Zusätzlich zum FB 860 - *VMC_AxisControl* haben Sie die Möglichkeit *PLCopen*-Bausteine zu nutzen.
- FB 800 ... FB 838 - *PLCopen*
 - Die *PLCopen*-Bausteine dienen zur Programmierung von Bewegungsabläufen und Statusabfragen.
 - Allgemeine Bausteine für alle Antriebe und Bussysteme.

3.1.4.3.2 Programmierung

Bibliothek einbinden

1. ➞ Gehen Sie in den Service-Bereich von www.vipa.com.
2. ➞ Laden Sie aus dem Downloadbereich unter "*VIPA Lib*" die *Simple Motion Control Library*.
3. ➞ Öffnen Sie mit "*Datei ➞ Dearchivieren*" das Dialogfenster zur Auswahl der ZIP-Datei.
4. ➞ Wählen Sie die entsprechende ZIP-Datei an und klicken Sie auf [Öffnen].
5. ➞ Geben Sie ein Zielverzeichnis an, in dem die Bausteine abzulegen sind und starten Sie den Entpackvorgang mit [OK].

Bausteine in Projekt kopieren

- ➞ Öffnen Sie die Bibliothek nach dem Entpackvorgang und ziehen Sie per Drag&Drop folgende Bausteine in "*Bausteine*" Ihres Projekts:
 - *Sigma-5* EtherCAT:
 - UDT 870 - *VMC_ConfigSigma5EC_REF*
 - FB 870 - *VMC_KernelSigma5_EC*
 - FB 871 - *VMC_InitSigma5_EC*
 - *Axis Control*
 - UDT 860 - *MC_AXIS_REF*
 - Bausteine für die gewünschten Bewegungsabläufe

Alarm-OBs anlegen

1. ➞ Klicken Sie in Ihrem Projekt auf "*Bausteine*" und wählen Sie "*Kontextmenü ➞ Neues Objekt einfügen ➞ Organisationsbaustein*".
 - ⇒ Das Dialogfenster "*Eigenschaften Organisationsbaustein*" öffnet sich.
2. ➞ Fügen Sie nacheinander OB 57, OB 82 und OB 86 Ihrem Projekt hinzu.

Achs-DB anlegen

- 1.** ➤ Klicken Sie in Ihrem Projekt auf *"Bausteine"* und wählen Sie *"Kontextmenü ➔ Neues Objekt einfügen ➔ Datenbaustein"*.

Geben Sie folgende Parameter an:

- Name und Typ
 - Die DB-Nr. als *"Name"* können Sie frei wählen wie z.B. DB 10.
 - Stellen Sie *"Global-DB"* als *"Typ"* ein.
- Symbolischer Name
 - Geben Sie *"Axis01"* an.

Bestätigen Sie Ihre Eingaben mit [OK].

⇒ Der Baustein wird angelegt.

- 2.** ➤ Öffnen Sie DB 10 "Axis01" durch Doppelklick.

- Legen Sie in "Axis01" die Variable "Config" vom Typ UDT 870 an. Dies sind spezifische Achs-Konfigurationsdaten.
- Legen Sie in "Axis01" die Variable "Axis" vom Typ UDT 860 an. Während des Betriebs werden hier alle Betriebsdaten der Achse abgelegt.

DB10

Adresse	Name	Typ	...
		Struct	
...	Config	"VMC_ConfigSigma5EC_REF"	
...	Axis	"MC_AXIS_REF"	
...		END_STRUCT	

OB 1**Konfiguration der Achse**

Öffnen Sie den OB 1 und programmieren Sie folgende FB-Aufrufe mit zugehörigen DBs:

➔ FB 871 - VMC_InitSigma5_EC, DB 871 ↪ *Kapitel 3.1.5.3 "FB 871 - VMC_InitSigma5_EC - Sigma-5 EtherCAT Initialisierung" auf Seite 45*

Geben Sie unter *InputsStartAddressPDO* bzw. *OutputsStartAddressPDO* die Adresse aus dem *SPEED7 EtherCAT Manager* an. ↪ 38

```
⇒ CALL "VMC_InitSigma5_EC" , "DI_InitSgm5ETC01"
   Enable           := "InitS5EC1_Enable"
   LogicalAddress    := 300
   InputsStartAddressPDO := 300 (EtherCAT-Man.: E-Adresse S7)
   OutputsStartAddressPDO := 300 (EtherCAT-Man.: A-Adresse S7)
   EncoderType       := 1
   EncoderResolutionBits := 20
   FactorPosition     := 1.048576e+006
   FactorVelocity     := 1.048576e+006
   FactorAcceleration := 1.048576e+002
   OffsetPosition     := 0.000000e+000
   MaxVelocityApp     := 5.000000e+001
   MaxAccelerationApp := 1.000000e+002
   MaxDecelerationApp := 1.000000e+002
   MaxVelocityDrive   := 6.000000e+001
   MaxAccelerationDrive := 1.500000e+002
   MaxDecelerationDrive := 1.500000e+002
   MaxPosition        := 1.048500e+003
   MinPosition        := -1.048514e+003
   EnableMaxPosition  := TRUE
   EnableMinPosition  := TRUE
   MinUserPosition    := "InitS5EC1_MinUserPos"
   MaxUserPosition    := "InitS5EC1_MaxUserPos"
   Valid              := "InitS5EC1_Valid"
   Error              := "InitS5EC1_Error"
   ErrorID            := "InitS5EC1_ErrorID"
   Config             := "Axis01".Config
   Axis               := "Axis01".Axis
```

**Kernel für Achse
beschalten**

Der *Kernel* verarbeitet die Benutzerkommandos und gibt sie entsprechend aufbereitet an den Antrieb über das jeweilige Bussystem weiter.

➔ FB 870 - VMC_KernelSigma5_EC, DB 870 ↪ *Kapitel 3.1.5.2 "FB 870 - VMC_KernelSigma5_EC - Sigma-5 EtherCAT Kernel" auf Seite 45*

```
⇒ CALL "VMC_KernelSigma5_EC" , "DI_KernelSgm5ETC01"
   Init := "KernelS5EC1_Init"
   Config := "Axis01".Config
   Axis := "Axis01".Axis
```

Baustein für Bewegungsabläufe beschalten

Zur Vereinfachung soll hier die Beschaltung des FB 860 - VMC_AxisControl gezeigt werden. Dieser Universalbaustein unterstützt einfache Bewegungskommandos und liefert Statusmeldungen zurück. Die Ein- und Ausgänge können Sie individuell beschalten. Bitte geben Sie unter "Axis" die Referenz zu den entsprechenden Achsdaten im *Achs-DB* an.

➔ FB 860 - VMC_AxisControl, DB 860 ↗ *Kapitel 3.4.2.2 "FB 860 - VMC_AxisControl - Control-Baustein Achskontrolle" auf Seite 132*

```
⇒ CALL "VMC_AxisControl" , "DI_AxisControl01"
    AxisEnable      := "AxCtrl1_AxisEnable"
    AxisReset       := "AxCtrl1_AxisReset"
    HomeExecute     := "AxCtrl1_HomeExecute"
    HomePosition    := "AxCtrl1_HomePosition"
    StopExecute     := "AxCtrl1_StopExecute"
    MvVelocityExecute := "AxCtrl1_MvVelExecute"
    MvRelativeExecute := "AxCtrl1_MvRelExecute"
    MvAbsoluteExecute := "AxCtrl1_MvAbsExecute"
    PositionDistance := "AxCtrl1_PositionDistance"
    Velocity        := "AxCtrl1_Velocity"
    Acceleration     := "AxCtrl1_Acceleration"
    Deceleration     := "AxCtrl1_Deceleration"
    JogPositive      := "AxCtrl1_JogPositive"
    JogNegative      := "AxCtrl1_JogNegative"
    JogVelocity      := "AxCtrl1_JogVelocity"
    JogAcceleration  := "AxCtrl1_JogAcceleration"
    JogDeceleration  := "AxCtrl1_JogDeceleration"
    AxisReady       := "AxCtrl1_AxisReady"
    AxisEnabled      := "AxCtrl1_AxisEnabled"
    AxisError        := "AxCtrl1_AxisError"
    AxisErrorID      := "AxCtrl1_AxisErrorID"
    DriveWarning     := "AxCtrl1_DriveWarning"
    DriveError       := "AxCtrl1_DriveError"
    DriveErrorID     := "AxCtrl1_DriveErrorID"
    IsHomed          := "AxCtrl1_IsHomed"
    ModeOfOperation  := "AxCtrl1_ModeOfOperation"
    PLCOpenState     := "AxCtrl1_PLCOpenState"
    ActualPosition   := "AxCtrl1_ActualPosition"
    ActualVelocity   := "AxCtrl1_ActualVelocity"
    CmdDone          := "AxCtrl1_CmdDone"
    CmdBusy          := "AxCtrl1_CmdBusy"
    CmdAborted       := "AxCtrl1_CmdAborted"
    CmdError         := "AxCtrl1_CmdError"
    CmdErrorID       := "AxCtrl1_CmdErrorID"
    DirectionPositive := "AxCtrl1_DirectionPos"
    DirectionNegative := "AxCtrl1_DirectionNeg"
    SWLimitMinActive := "AxCtrl1_SWLimitMinActive"
    SWLimitMaxActive := "AxCtrl1_SWLimitMaxActive"
    HWLimitMinActive := "AxCtrl1_HWLimitMinActive"
    HWLimitMaxActive := "AxCtrl1_HWLimitMaxActive"
    Axis             := "Axis01".Axis
```



Für komplexe Bewegungsaufgaben können Sie die PLCopen-Bausteine verwenden. Hier müssen Sie ebenfalls unter Axis die Referenz zu den Achsdaten im Achs-DB angeben.

Ihr Projekt beinhaltet nun folgende Bausteine:

- OB 1 - Main
- OB 57 - DP Manufacturer Alarm
- OB 82 - I/O_FLT1
- OB 86 - Rack_FLT
- FB 860 - VMC_AxisControl mit Instanz-DB

- FB 870 - VMC_KernelSigma5_EC mit Instanz-DB
- FB 871 - VMC_InitSigma5_EC mit Instanz-DB
- UDT 860 - MC_Axis_REF
- UDT 870 - VMC_ConfigSigma5EC_REF

Zeitlicher Ablauf

1. ➔ Wechseln Sie in den Siemens SIMATIC Manager und übertragen Sie Ihr Projekt in die CPU.

Die Übertragung kann ausschließlich aus dem Siemens SIMATIC Manager erfolgen - nicht Hardware-Konfigurator!



Da Slave- und Modulparameter mittels SDO-Zugriff bzw. SDO-Init-Kommando übertragen werden, bleibt die Parametrierung solange bestehen, bis ein Power-Cycle durchgeführt wird oder neue Parameter für die gleichen SDO-Objekte übertragen werden.

Beim Urlöschen werden Slave- und Modul-Parameter nicht zurückgesetzt!

⇒ Sie können jetzt Ihre Applikation in Betrieb nehmen.



VORSICHT!

Bitte beachten Sie immer die Sicherheitshinweise zu ihrem Antrieb, insbesondere bei der Inbetriebnahme!

2. ➔ Bevor eine Achse gesteuert werden kann, muss diese initialisiert werden. Rufen Sie hierzu den *Init*-Baustein FB 871 - VMC_InitSigma5_EC mit *Enable* = TRUE auf.

⇒ Der Ausgang *Valid* meldet TRUE zurück. Im Fehlerfall können Sie durch Auswertung der *ErrorID* den Fehler ermitteln.

Den *Init*-Baustein müssen Sie erneut aufrufen, wenn Sie einen neuen Achs-DB laden oder Parameter am *Init*-Baustein geändert wurden.



Fahren Sie erst fort, wenn der Init-Baustein keinen Fehler meldet!

3. ➔ Stellen Sie sicher, dass der *Kernel*-Baustein FB 870 - VMC_KernelSigma5_EC zyklisch aufgerufen wird. Auf diese Weise werden Steuersignale an den Antrieb übergeben und Statusmeldungen übermittelt.
4. ➔ Programmieren Sie Ihre Applikation mit dem FB 860 - VMC_AxisControl oder mit den PLCopen Bausteinen auf.

Steuerung des Antriebs über HMI

Sie haben die Möglichkeit über ein HMI Ihren Antrieb zu steuern. Hierzu gibt es für Movicon eine vorgefertigte Symbolbibliothek für den Zugriff auf den VMC_AxisControl Funktionsbaustein. ➔ Kapitel 7 "Antrieb über HMI steuern" auf Seite 313

3.1.4.4 Projekt kopieren

Vorgehensweise

Im Beispiel wird die Station "Source" kopiert und als "Target" gespeichert.

1. Öffnen Sie die Hardware-Konfiguration der "Source"-CPU und starten Sie hier den *SPEED7 EtherCAT Manager*.
2. Speichern Sie im *SPEED7 EtherCAT Manager* über "Datei → Speichern unter" die Konfiguration in Ihrem Arbeitsverzeichnis.
3. Schließen Sie den *SPEED7 EtherCAT Manager* und den Hardware-Konfigurator wieder.
4. Kopieren Sie die Station "Source" mit Strg+C und fügen Sie diese mit Strg+V als "Target" in Ihr Projekt ein.
5. Wechseln Sie in den "Baustein"-Ordner der "Target"-CPU löschen Sie die "Systemdaten".
6. Öffnen Sie die Hardware-Konfiguration der "Target"-CPU. Passen Sie die IP-Adressdaten an oder vernetzen Sie die CPU bzw. den CP neu.



Vor dem Aufruf des SPEED7 EtherCAT Manager müssen Sie immer Ihr Projekt mit "Station → Speichern und übersetzen" speichern.

7. Speichern Sie Ihr Projekt mit "Station → Speichern und übersetzen".
8. Öffnen Sie den *SPEED7 EtherCAT Manager*.
9. Laden Sie mit "Datei → Öffnen" die Konfiguration aus Ihrem Arbeitsverzeichnis.
10. Schließen Sie den *SPEED7 EtherCAT Manager* wieder.
11. Speichern und übersetzen Sie Ihre Konfiguration.

3.1.5 Antriebsspezifische Bausteine

3.1.5.1 UDT 870 - VMC_ConfigSigma5EC_REF - Sigma-5 EtherCAT Datenstruktur Achskonfiguration

Dies ist eine benutzerdefinierte Datenstruktur, die Informationen zu den Konfigurationsdaten beinhaltet. Die UDT ist speziell angepasst an die Verwendung eines *Sigma-5*-Antriebs, welcher über EtherCAT angebunden ist.

3.1.5.2 FB 870 - VMC_KernelSigma5_EC - Sigma-5 EtherCAT Kernel

Beschreibung

Dieser Baustein setzt die Antriebskommandos für eine *Sigma-5* Achse über EtherCAT um und kommuniziert mit dem Antrieb. Je *Sigma-5* Achse ist eine Instanz dieses FBs zyklisch aufzurufen.



Bitte beachten Sie, dass dieser Baustein intern den SFB 238 aufruft.

Im SPEED7 Studio wird dieser Baustein automatisch in Ihr Projekt eingefügt.

Im Siemens SIMATIC Manager müssen Sie den SFB 238 aus der Motion Control Library in Ihr Projekt kopieren.

Parameter	Deklaration	Datentyp	Beschreibung
Init	INPUT	BOOL	Mit einer Flanke 0-1 wird der Baustein intern zurückgesetzt. Hierbei werden bestehende Bewegungskommandos abgebrochen und der Baustein wird initialisiert.
Config	IN_OUT	UDT870	Datenstruktur zur Übergabe von achsabhängigen Konfigurationsdaten an den <i>AxisKernel</i> .
Axis	IN_OUT	MC_AXIS_REF	Datenstruktur zur Übergabe von achsabhängigen Informationen an <i>AxisKernel</i> und PLCopen-Bausteine.

3.1.5.3 FB 871 - VMC_InitSigma5_EC - Sigma-5 EtherCAT Initialisierung

Beschreibung

Dieser Baustein dient zur Konfiguration der Achse. Der Baustein ist speziell angepasst an die Verwendung eines *Sigma-5*-Antriebs, welcher über EtherCAT angebunden ist.

Parameter	Deklaration	Datentyp	Beschreibung
Config	IN_OUT	UDT870	Datenstruktur zur Übergabe von achsabhängigen Konfigurationsdaten an den <i>AxisKernel</i> .
Axis	IN_OUT	MC_AXIS_REF	Datenstruktur zur Übergabe von achsabhängigen Informationen an <i>AxisKernel</i> und PLCopen-Bausteine.
Enable	INPUT	BOOL	Freigabe der Initialisierung
LogicalAddress	INPUT	INT	Startadresse der PDO-Eingangsdaten
InputsStartAddressPDO	INPUT	INT	Startadresse der Eingabe-PDOs
OutputsStartAddressPDO	INPUT	INT	Startadresse der Ausgabe-PDOs
EncoderType	INPUT	INT	Encoder-Typ <ul style="list-style-type: none"> ■ 1: Absolut-Encoder ■ 2: Inkremental-Encoder

Parameter	Deklaration	Datentyp	Beschreibung
EncoderResolutionBits	INPUT	INT	Anzahl der Bits, die einer Geber-Umdrehung entsprechen. Default: 20
FactorPosition	INPUT	REAL	Faktor zur Umrechnung der Position von Benutzereinheiten [u] in Antriebseinheiten [Inkmente] und zurück. Es gilt: $p_{[\text{Inkmente}]} = p_{[u]} \times \text{FactorPosition}$ Bitte berücksichtigen sie auch den Faktor, welchen Sie am Antrieb über die Objekte 0x2701:1 und 0x2701:2 vorgeben können. Dieser sollte 1 sein.
FactorVelocity	INPUT	REAL	Faktor zur Umrechnung der Geschwindigkeit von Benutzereinheiten [u/s] in Antriebseinheiten [Inkmente/s] und zurück. Es gilt: $v_{[\text{Inkmente/s}]} = v_{[u/s]} \times \text{FactorVelocity}$ Bitte berücksichtigen sie auch den Faktor, welchen Sie am Antrieb über die Objekte 0x2702:1 und 0x2702:2 vorgeben können. Dieser sollte 1 sein.
FactorAcceleration	INPUT	REAL	Faktor zur Umrechnung der Beschleunigung von Benutzereinheiten [u/s ²] in Antriebseinheiten [10 ⁻⁴ x Inkmente/s ²] und zurück. Es gilt: $10^{-4} \times a_{[\text{Inkmente/s}^2]} = a_{[u/s^2]} \times \text{FactorAcceleration}$ Bitte berücksichtigen sie auch den Faktor, welchen Sie am Antrieb über die Objekte 0x2703:1 und 0x2703:2 vorgeben können. Dieser sollte 1 sein.
OffsetPosition	INPUT	REAL	Offset für die Nullposition [u].
MaxVelocityApp	INPUT	REAL	Maximale Geschwindigkeit der Applikation [u/s]. Die Kommandoeingaben werden vor Ausführung auf den Maximalwert überprüft.
MaxAccelerationApp	INPUT	REAL	Maximale Beschleunigung der Applikation [u/s ²]. Die Kommandoeingaben werden vor Ausführung auf den Maximalwert überprüft.
MaxDecelerationApp	INPUT	REAL	Maximale Verzögerung der Applikation [u/s ²]. Die Kommandoeingaben werden vor Ausführung auf den Maximalwert überprüft.
MaxPosition	INPUT	REAL	Maximale Position für die Überwachung der Softwarelimits [u].
MinPosition	INPUT	REAL	Minimale Position für die Überwachung der Softwarelimits [u].
EnableMaxPosition	INPUT	BOOL	Überwachung maximale Position ■ TRUE: Aktivierung der Überwachung der maximalen Position.
EnableMinPosition	INPUT	BOOL	Überwachung minimale Position ■ TRUE: Aktivierung der Überwachung der minimalen Position.
MinUserPosition	OUTPUT	REAL	Minimale Benutzerposition basierend auf dem minimalen Encoder Wert von 0x80000000 und dem <i>FactorPosition</i> [u].

Parameter	Deklaration	Datentyp	Beschreibung
MaxUserPosition	OUTPUT	REAL	Maximale Benutzerposition basierend auf dem maximalen Encoder Wert von 0x7FFFFFFF und dem <i>Factor-Position</i> [u].
Valid	OUTPUT	BOOL	Initialisierung ■ TRUE: Initialisierung ist gültig.
Error	OUTPUT	BOOL	■ Fehler – TRUE: Ein Fehler ist aufgetreten. Zusätzliche Fehlerinformationen können dem Parameter <i>ErrorID</i> entnommen werden. Die Achse wird gesperrt.
ErrorID	OUTPUT	WORD	Zusätzliche Fehlerinformationen 🔗 Kapitel 9 "ErrorID - Zusätzliche Fehlerinformationen" auf Seite 340

3.2 Einsatz Sigma-7S EtherCAT

3.2.1 Übersicht

Einsatz des Doppelachs-Antriebs 🔗 Kapitel 3.3 "Einsatz Sigma-7W EtherCAT" auf Seite 86

Voraussetzung

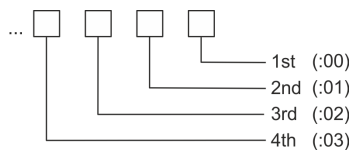
- SPEED7 Studio ab V1.6.1 oder
- Siemens SIMATIC Manager ab V 5.5 SP2 & *SPEED7 EtherCAT Manager & Simple Motion Control Library*
- CPU mit EtherCAT-Master wie z.B. CPU 015-CEFNR00
- *Sigma-7S*-Antrieb mit EtherCAT-Optionskarte

Schritte der Projektierung

1. ➞ Parameter am Antrieb einstellen
 - Die Einstellung der Parameter hat mit dem Softwaretool *Sigma Win+* zu erfolgen.
2. ➞ Hardwarekonfiguration im VIPA *SPEED7 Studio* oder Siemens SIMATIC Manager
 - Projektierung einer CPU mit EtherCAT-Master-Funktionalität.
 - Projektierung eines *Sigma-7S* EtherCAT-Antriebs.
 - Projektierung der EtherCAT-Anbindung über *SPEED7 EtherCAT Manager*.
3. ➞ Programmierung im VIPA *SPEED7 Studio* oder Siemens SIMATIC Manager
 - *Init*-Baustein zur Konfiguration der Achse beschalten.
 - *Kernel*-Baustein zur Kommunikation mit der Achse beschalten.
 - Bausteine für die Bewegungsabläufe beschalten.

3.2.2 Parameter am Antrieb einstellen

Parameter-Digits



VORSICHT!

Vor der Erstinbetriebnahme müssen Sie Ihren Antrieb mit dem Software-tool *Sigma Win+* an Ihre Applikation anpassen! Näheres hierzu finden Sie im Handbuch zu ihrem Antrieb.

Zur Abstimmung auf die *Simple Motion Control Library* sind folgende Parameter über *Sigma Win+* einzustellen:

Sigma-7S (24Bit Encoder)

Servopack Parameter	Adresse:Digit	Name	Wert
Pn205	(2205h)	Multiturn Limit Setting	65535
Pn20E	(220Eh)	ElectronicGear Ratio (Numerator)	16
Pn210	(2210h)	Electronic Gear Ratio (Denominator)	1
PnB02	(2701h:01)	Position User Unit (Numerator)	1
PnB04	(2701h:02)	Position User Unit (Denominator)	1
PnB06	(2702h:01)	Velocity User Unit (Numerator)	1
PnB08	(2702h:02)	Velocity User Unit (Denominator)	1
PnB0A	(2703h:01)	Acceleration User Unit (Numerator)	1
PnB0C	(2703h:02)	Acceleration User Unit (Denominator)	1

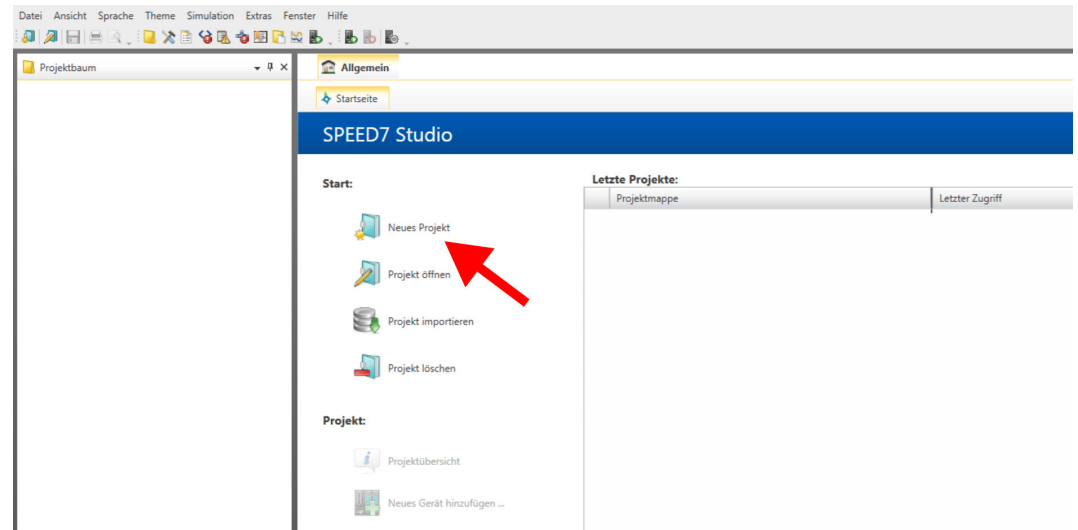
3.2.3 Einsatz im VIPA *SPEED7 Studio*

3.2.3.1 Hardware-Konfiguration

CPU im Projekt anlegen

Bitte verwenden Sie für die Projektierung das *SPEED7 Studio* ab V1.6.1.

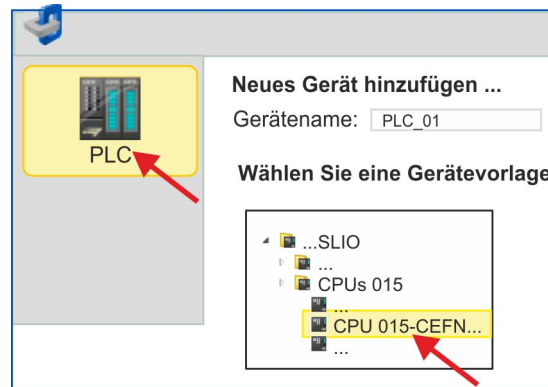
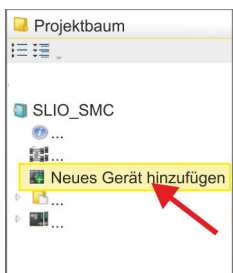
1. ➔ Starten Sie das *SPEED7 Studio*.



2. ➔ Erstellen sie auf der Startseite mit "*Neues Projekt*" ein neues Projekt und vergeben Sie einen "*Projektnamen*".

⇒ Ein neues Projekt wird angelegt und in die Sicht "*Geräte und Netze*" gewechselt.

3. ➔ Klicken Sie im *Projektbaum* auf "*Neues Gerät hinzufügen ...*".



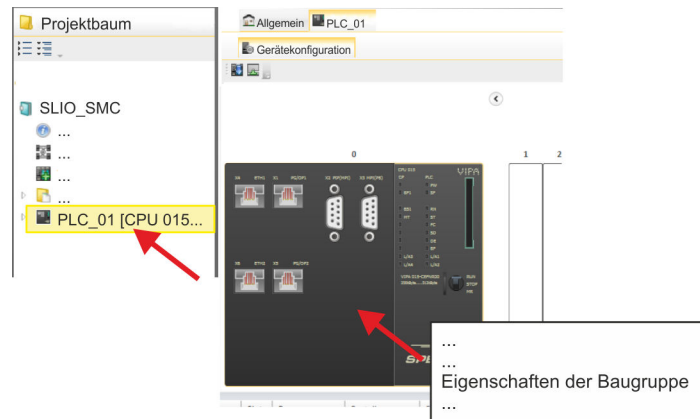
⇒ Es öffnet sich ein Dialog für die Geräteauswahl.

4. ➔ Wählen Sie unter den "*Gerätevorlagen*" eine CPU mit EtherCAT-Master-Funktionalität wie z.B. die CPU 015-CEFN00 und klicken Sie auf [OK].

⇒ Die CPU wird in "*Geräte und Netze*" eingefügt und die "*Gerätekonfiguration*" geöffnet.

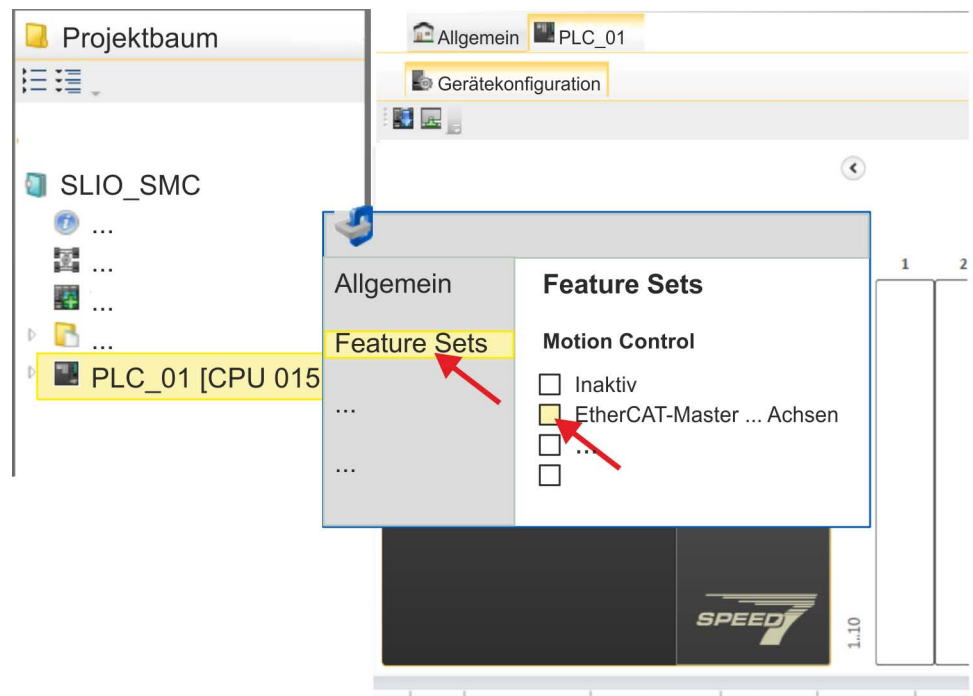
Motion-Control-Funktionen aktivieren

Sofern bei Ihrer CPU die EtherCAT-Master-Funktionalität noch nicht aktiviert ist, erfolgt die Aktivierung nach folgenden Vorgehensweise:



1. Klicken Sie in der "Gerätekonfiguration" auf die CPU und wählen Sie "Kontextmenü → Eigenschaften der Baugruppe".

⇒ Es öffnet sich der Eigenschaften-Dialog der CPU.



2. Klicken Sie auf "Feature Sets" und aktivieren Sie unter "Motion Control" einen der Parameter "EtherCAT-Master ... Achsen". Die Anzahl der Achsen ist in diesem Beispiel nicht relevant.

3. Bestätigen Sie Ihre Angaben mit [OK].

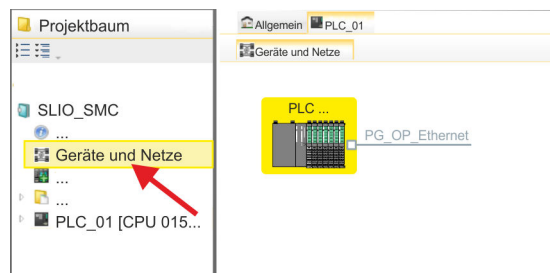
⇒ Die Motion-Control-Funktionen stehen Ihnen nun in Ihrem Projekt zur Verfügung.

**VORSICHT!**

Bitte beachten Sie, dass bei jeder Änderung der Feature-Set-Einstellungen systembedingt das EtherCAT-Feldbus-System zusammen mit der Motion-Control-Konfiguration aus Ihrem Projekt gelöscht werden!

Ethernet-PG/OP-Kanal parametrieren

1. ➤ Klicken Sie im *Projektbaum* auf "*Geräte und Netze*".
⇒ Sie erhalten eine grafische Objekt-Ansicht Ihrer CPU.



2. ➤ Klicken Sie auf das Netzwerk "*PG_OP_Ethernet*".
3. ➤ Wählen Sie "*Kontextmenü → Eigenschaften der Schnittstelle*".
⇒ Es öffnet sich ein Dialogfenster. Hier können Sie IP-Adressdaten für Ihren Ethernet-PG/OP-Kanal angeben. Gültige IP-Adress-Parameter erhalten Sie von Ihrem Systemadministrator.
4. ➤ Bestätigen Sie Ihre Eingabe mit [OK].
⇒ Die IP-Adressdaten werden in Ihr Projekt übernommen und in "*Geräte und Netze*" unter "*Lokale Baugruppen*" aufgelistet.
Nach der Übertragung Ihres Projekts ist Ihre CPU über die angegebenen IP-Adressdaten via Ethernet-PG/OP-Kanal erreichbar.

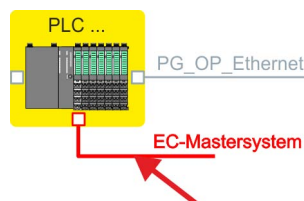
ESI-Datei installieren

Damit der *Sigma-7* EtherCAT Antrieb im *SPEED7 EtherCAT Manager* konfiguriert werden kann, muss die entsprechende ESI-Datei installiert sein. In der Regel wird das *SPEED7 Studio* mit aktuellen ESI-Dateien ausgeliefert und Sie können diesen Teil überspringen. Sollte Ihre ESI-Datei veraltet sein, finden Sie die aktuellste ESI-Datei für den *Sigma-7* EtherCAT Antrieb unter www.yaskawa.eu.com unter "*Service → Drives & Motion Software*".

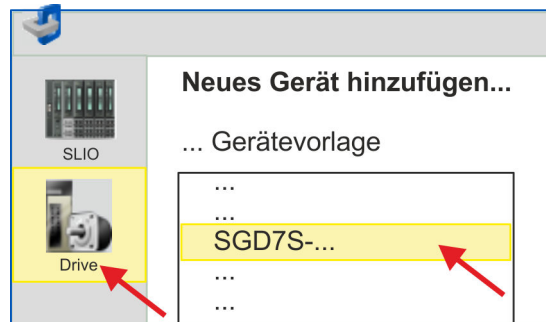
1. ➤ Laden Sie die zu Ihrem Antrieb passende ESI-Datei herunter. Entpacken Sie diese falls erforderlich.
2. ➤ Gehen Sie in Ihr *SPEED7 Studio*.
3. ➤ Öffnen Sie mit "*Extras → Gerätebeschreibungsdatei installieren (EtherCAT - ESI)*" das zugehörige Dialogfenster.
4. ➤ Geben Sie unter "*Quellpfad*" die ESI-Datei an und installieren Sie diese mit [Installieren].
⇒ Die Geräte der ESI-Datei steht Ihnen nun zur Verfügung.

Sigma-7S Singleachs-Antrieb hinzufügen

1. ➤ Klicken Sie im Projektbaum auf "*Geräte und Netze*".
2. ➤ Klicken Sie hier auf "*EC-Mastersystem*" und wählen sie "*Kontextmenü → Neues Gerät hinzufügen*".



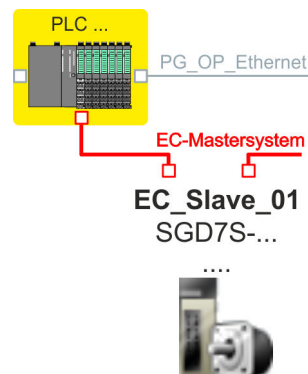
- ⇒ Es öffnet sich die Gerätevorlage zur Auswahl eines EtherCAT-Devices.



3. Wählen Sie Ihren *Sigma-7* Antrieb aus:

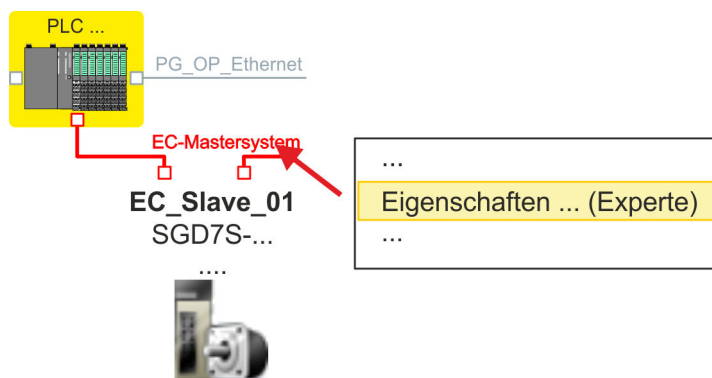
- SGD7S-xxxAA0...
- SGD7S-xxxDA0...
- SGD7S-xxxxA0...

Bestätigen Sie Ihre Angaben mit [OK]. Sollte Ihr Antrieb nicht vorhanden sein, müssen Sie die entsprechende ESI-Datei wie weiter oben beschrieben installieren.



⇒ Der *Sigma-7* Antrieb wird an Ihr EC-Mastersystem angebunden.

Sigma-7S Singleachs-Antrieb konfigurieren



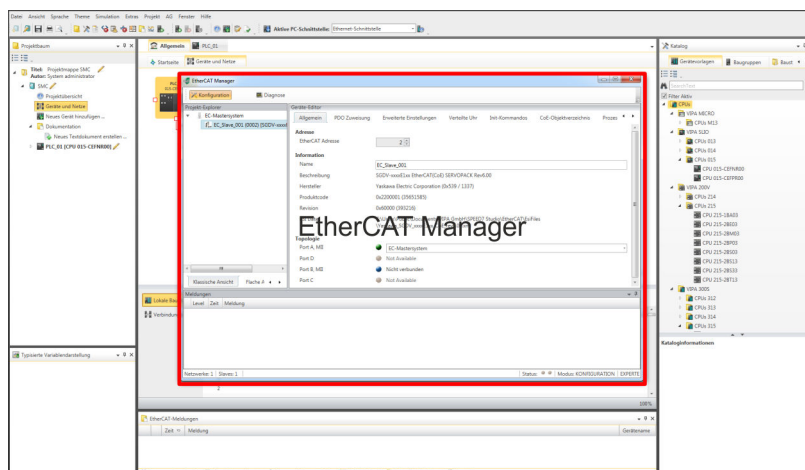
1. Klicken Sie auf "EC-Mastersystem" und wählen sie "Kontextmenü → Eigenschaft des Busystems (Experte)".



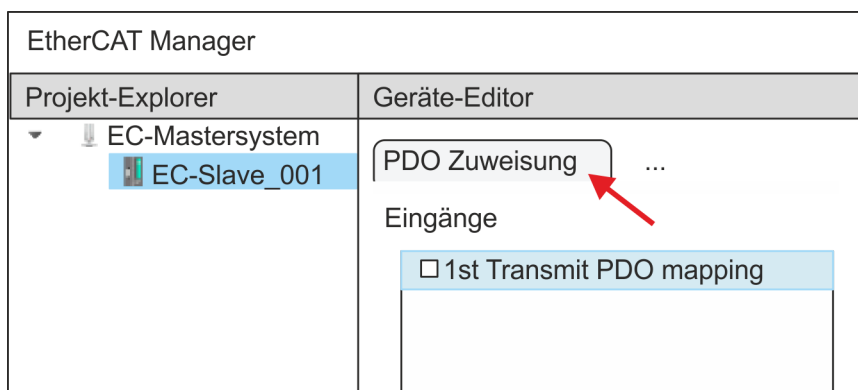
PDOs können Sie nur im "Experten-Modus" bearbeiten! Ansonsten werden die Schaltflächen ausgeblendet.

- ⇒ Der **SPEED7 EtherCAT Manager** wird gestartet. Hier können Sie die EtherCAT-Kommunikation zu Ihrem Sigma-7 Antrieb konfigurieren.

Näheres zum Einsatz des **SPEED7 EtherCAT Manager** finden Sie in der Onlinehilfe zum **SPEED7 Studio**.



2. Klicken Sie im **SPEED7 EtherCAT Manager** auf den Slave und wählen Sie im "Geräte-Editor" den Reiter "PDO-Zuweisung" an.



- ⇒ Dieser Dialog zeigt eine Auflistung aller PDOs.

3. → Durch Auswahl des entsprechenden PDO-Mappings können Sie mit [Bearbeiten] die PDOs bearbeiten. Wählen Sie das Mapping "1st Transmit PDO mapping" an und klicken Sie auf [Bearbeiten].



Bitte beachten Sie, dass aufgrund der Voreinstellung manche PDOs nicht bearbeitet werden können. Durch Deaktivierung bereits aktivierter PDOs können Sie die Bearbeitung von gesperrten PDOs frei geben.

Geräte-Editor

PDO Zuweisung ...

Eingänge	Ausgänge
<input type="checkbox"/> 1st Transmit PDO mapping	<input type="checkbox"/> 1st Receive PDO mapping
<input type="checkbox"/> 2nd Transmit PDO mapping	<input type="checkbox"/> 2nd Receive PDO mapping
...	...

... Bearbeiten ...

- ⇒ Es öffnet sich der Dialog *"PDO bearbeiten"*. Bitte überprüfen Sie hier die aufgeführten PDO-Einstellungen und passen Sie diese ggf. an. Bitte berücksichtigen Sie hierbei auch die Reihenfolge der *"Einträge"* und ergänzen Sie diese entsprechend.

The dialog box 'PDO bearbeiten' contains the following elements:

- Allgemein:**
 - Name: 1st Transmit PDO mapping
 - Index: 0x1A00 (with 'Dez' and 'Hex' buttons)
 - Flags:
 - Zwingend: ☐
 - Schreibgeschützt: ☐
 - Virtuell: ☐
 - Richtung:
 - TxPdo (Eingang): ☒
 - RxPdo (Ausgang): ☐
- Optional:**
 - Ausschließen:
 - 1A01: ☐
 - 1A02: ☒
 - 1A03: ☒
- Einträge:**

Name	Index	Bitlänge	Kommentar
Status word	0x6041:00	16	
Position actual internal value	0x6063:00	32	
Position actual value	0x6064:00	32	
Torque actual value	0x6077:00	16	
Following error actual value	0x60F4:00	32	
Modes of operation display	0x6061:00	8	
---	---	8	---
Digital inputs	0x60FD:00	32	

Buttons at the bottom: Neu, Löschen, Bearbeiten, Nach oben, Nach unten

Für die Bearbeitung der *"Einträge"* stehen folgende Funktionen zur Verfügung:

- **Neu**
 - Hiermit können Sie in einem Dialogfenster einen neuen Eintrag anlegen, indem Sie aus dem *"CoE-Objektverzeichnis"* den entsprechenden Eintrag auswählen und Ihre Einstellungen vornehmen. Mit [OK] wird der Eintrag übernommen und in der Liste der Einträge aufgeführt.
- **Löschen**
 - Hiermit können Sie den angewählte Eintrag löschen.
- **Bearbeiten**
 - Hiermit können Sie allgemeinen Daten eines Eintrags bearbeiten.
- **Nach oben/unten**
 - Hiermit können Sie den angewählten Eintrag in der Liste nach oben bzw. nach unten bewegen

4. ➔ Führen Sie folgende Einstellungen durch:**Eingänge: 1st Transmit PDO 0x1A00**

- Allgemein
 - Name: 1st Transmit PDO mapping
 - Index: 0x1A00
- Flags
 - Alles deaktiviert
- Richtung
 - TxPdo (Eingang): aktiviert
- Ausschließen

Bitte diese Einstellungen beachten, da ansonsten die PDO-Mappings nicht zeitgleich aktiviert werden können!

 - 1A01: deaktiviert
- Einträge

Name	Index	Bitlänge
Status word	0x6041:00	16Bit
Position actual internal value	0x6063:00	32Bit
Position actual value	0x6064:00	32Bit
Torque actual value	0x6077:00	16Bit
Following error actual value	0x60F4:00	32Bit
Modes of operation display	0x6061:00	8Bit
---	---	8Bit
Digital inputs	0x60FD:00	32Bit

Schließen Sie den Dialog *"PDO bearbeiten"* mit [OK].

5. ➔ Wählen Sie das Mapping *"2nd Transmit PDO mapping"* an und klicken Sie auf [Bearbeiten]. Führen Sie folgende Einstellungen durch:

Eingänge: 2nd Transmit PDO 0x1A01

- Allgemein
 - Name: 2nd Transmit PDO mapping
 - Index: 0x1A01
- Flags
 - Alles deaktiviert
- Richtung
 - TxPdo (Eingang): aktiviert
- Ausschließen

Bitte diese Einstellungen beachten, da ansonsten die PDO-Mappings nicht zeitgleich aktiviert werden können!

 - 1A00: deaktiviert
 - 1A02: deaktiviert
 - 1A03: deaktiviert
- Einträge

Name	Index	Bitlänge
Touch probe status	0x60B9:00	16Bit
Touch probe 1 position value	0x60BA:00	32Bit
Touch probe 2 position value	0x60BC:00	32Bit
Velocity actual value	0x606C:00	32Bit

Schließen Sie den Dialog *"PDO bearbeiten"* mit [OK].

6. ➔ Wählen Sie das Mapping *"1st Receive PDO mapping"* an und klicken Sie auf [Bearbeiten]. Führen Sie folgende Einstellungen durch:

Ausgänge: 1st Receive PDO 0x1600

- Allgemein
 - Name: 1st Receive PDO mapping
 - Index: 0x1600
- Flags
 - Alles deaktiviert
- Richtung
 - RxPdo (Ausgang): aktiviert
- Ausschließen

Bitte diese Einstellungen beachten, da ansonsten die PDO-Mappings nicht zeitgleich aktiviert werden können!

 - 1601: deaktiviert
 - 1602: deaktiviert
 - 1603: deaktiviert
- Einträge

Name	Index	Bitlänge
Control word	0x6040:00	16Bit
Target position	0x607A:00	32Bit
Target velocity	0x60FF:00	32Bit
Modes of operation	0x6060:00	8Bit
---	---	8Bit
Touch probe function	0x60B8:00	16Bit

Schließen Sie den Dialog *"PDO bearbeiten"* mit [OK].

7. ➔ Wählen Sie das Mapping *"2nd Receive PDO mapping"* an und klicken Sie auf [Bearbeiten]. Führen Sie folgende Einstellungen durch:

Ausgänge: 2nd Receive PDO 0x1601

- Allgemein
 - Name: 2nd Receive PDO mapping
 - Index: 0x1601
- Flags
 - Alles deaktiviert
- Richtung
 - RxPdo (Ausgang): aktiviert
- Ausschließen

Bitte diese Einstellungen beachten, da ansonsten die PDO-Mappings nicht zeitgleich aktiviert werden können!

 - 1600: deaktiviert
 - 1602: aktiviert
 - 1603: aktiviert
- Einträge

Name	Index	Bitlänge
Profile velocity	0x6081:00	32Bit
Profile acceleration	0x6083:00	32Bit
Profile deceleration	0x6084:00	32Bit

Schließen Sie den Dialog *"PDO bearbeiten"* mit [OK].

8. ➔ Aktivieren Sie in PDO-Zuweisung die PDOs 1 und 2 für die Ein und Ausgänge. Alle nachfolgenden PDOs müssen deaktiviert bleiben. Sollte dies nicht möglich sein, überprüfen Sie bitte den jeweiligen PDO-Parameter *"Ausschließen"*.

Geräte-Editor

PDO Zuweisung ...

Eingänge	Ausgänge
<input checked="" type="checkbox"/> 1st Transmit PDO mapping	<input checked="" type="checkbox"/> 1st Receive PDO mapping
<input checked="" type="checkbox"/> 2nd Transmit PDO mapping	<input checked="" type="checkbox"/> 2nd Receive PDO mapping

9. ➔ Wählen Sie im *"Geräte-Editor"* des *SPEED7 EtherCAT Manager* den Reiter *"Verteilte Uhren"* an und stellen Sie *"DC unused"* als *"Betriebsart"* ein.

Geräte-Editor

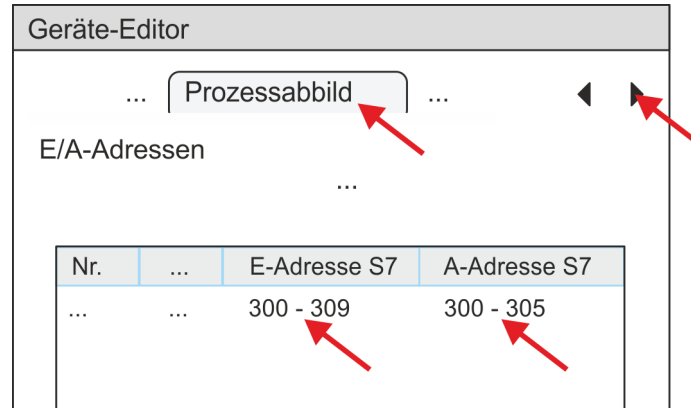
Verteilte Uhren ...

Verteilte Uhren

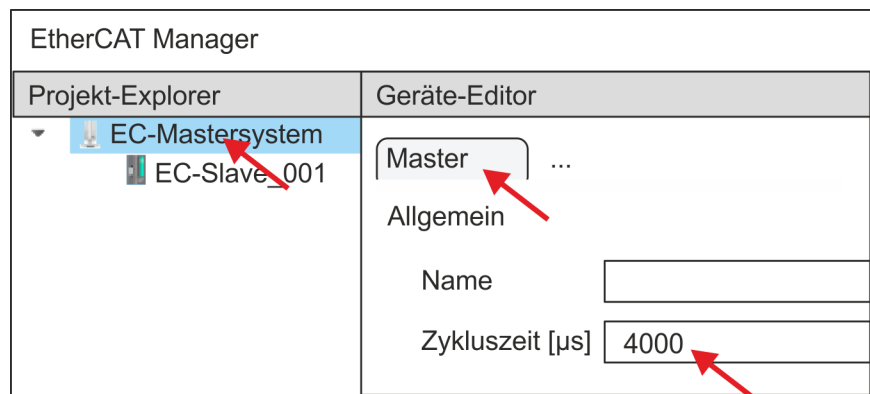
Betriebsart: DC unused

10. Wählen Sie im "Geräte-Editor" über die Pfeiltaste den Reiter "Prozessabbild" an und notieren Sie sich für die Parameter des Bausteins FB 873 - VMC_InitSigma7S_EC folgende PDO-Anfangsadressen:

- "E-Adresse S7" → "InputsStartAddressPDO"
- "A-Adresse S7" → "OutputsStartAddressPDO"



11. Klicken Sie im *SPEED7 EtherCAT Manager* auf "EC-Mastersystem" und wählen Sie im "Geräte-Editor" den Reiter "Master" an.

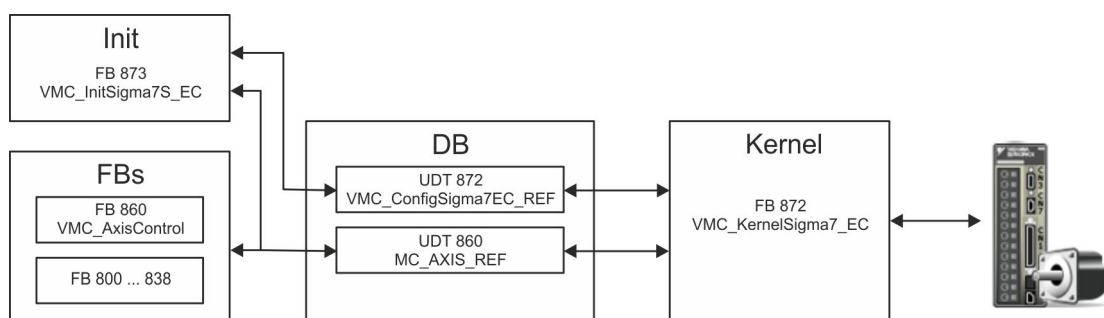


⇒ Stellen Sie für Sigma-7S (400V) Antriebe (SGD7S-xxxDA0... und SGD7S-xxxxA0...) eine Zykluszeit von mindestens 4ms ein. Ansonsten lassen Sie den Wert bei 1ms.

12. Indem Sie den Dialog des *SPEED7 EtherCAT Manager* mit [X] schließen, wird die Konfiguration in das *SPEED7 Studio* übernommen.

3.2.3.2 Anwender-Programm

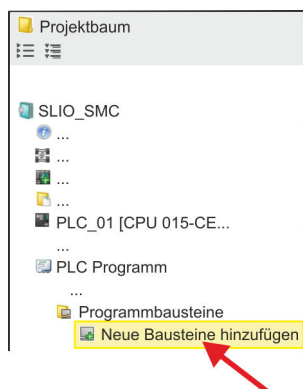
3.2.3.2.1 Programmstruktur



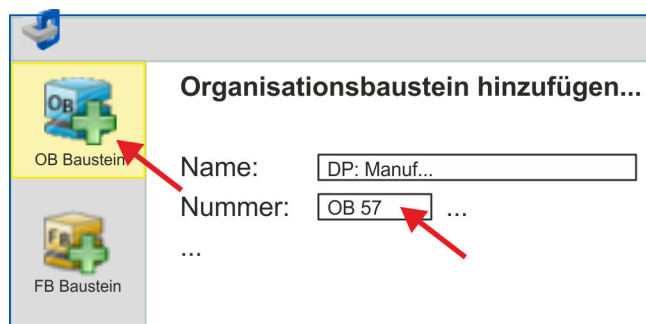
- **DB**
Für jede Achse ist ein Datenbaustein (Achs-DB) für Konfiguration und Statusdaten anzulegen. Der Datenbaustein besteht aus folgenden Datenstrukturen:
 - UDT 872 - *VMC_ConfigSigma7EC_REF*
Die Datenstruktur beschreibt den Aufbau der Konfiguration des Antriebs. Spezifische Datenstruktur für *Sigma-7* EtherCAT.
 - UDT 860 - *MC_AXIS_REF*
Die Datenstruktur beschreibt den Aufbau der Parameter und Statusinformationen von Antrieben. Allgemeine Datenstruktur für alle Antriebe und Bussysteme.
- **FB 873 - *VMC_InitSigma7S_EC***
 - Der *Init*-Baustein dient zur Konfiguration einer Achse.
 - Spezifischer Baustein für *Sigma-7S* EtherCAT.
 - Die Konfigurationsdaten für die Initialisierung sind im *Achs-DB* abzulegen.
- **FB 872 - *VMC_KernelSigma7_EC***
 - Der *Kernel*-Baustein kommuniziert mit dem Antrieb über das entsprechende Bussystem, verarbeitet die Benutzeraufträge und liefert Statusmeldungen zurück.
 - Spezifischer Baustein für *Sigma-7* EtherCAT.
 - Der Austausch der Daten erfolgt mittels des *Achs-DB*.
- **FB 860 - *VMC_AxisControl***
 - Universal-Baustein für alle Antriebe und Bussysteme.
 - Unterstützt einfache Bewegungskommandos und liefert alle relevanten Statusmeldungen.
 - Der Austausch der Daten erfolgt mittels des *Achs-DB*.
 - Über die Instanzdaten des Bausteins können Sie zur Bewegungssteuerung und Statusabfrage eine Visualisierung anbinden.
 - Zusätzlich zum FB 860 - *VMC_AxisControl* haben Sie die Möglichkeit *PLCopen*-Bausteine zu nutzen.
- **FB 800 ... FB 838 - *PLCopen***
 - Die *PLCopen*-Bausteine dienen zur Programmierung von Bewegungsabläufen und Statusabfragen.
 - Allgemeine Bausteine für alle Antriebe und Bussysteme.

3.2.3.2.2 Programmierung

Bausteine in Projekt kopieren

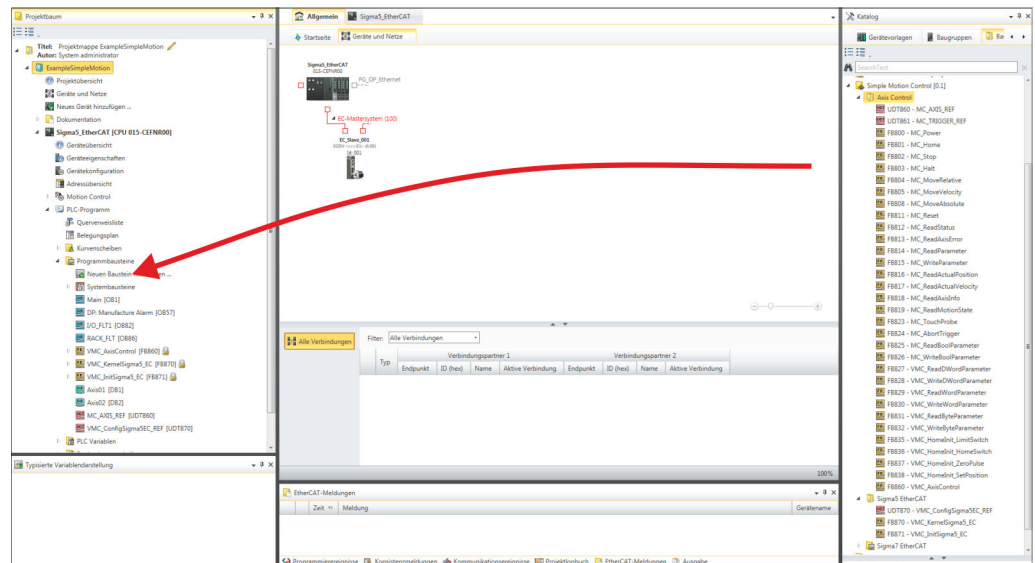


1. ➔ Klicken Sie im *Projektbaum* innerhalb der CPU unter "*PLC-Programm*", "*Programmbausteine*" auf "*Neuen Baustein hinzufügen*".



⇒ Das Dialogfenster "*Baustein hinzufügen*" öffnet sich.

2. ➔ Wählen Sie den Bausteintyp "*OB Baustein*" und fügen Sie nacheinander OB 57, OB 82 und OB 86 Ihrem Projekt hinzu.



3. Öffnen Sie im "Katalog" unter "Bausteine" "Simple Motion Control" und ziehen Sie per Drag&Drop folgende Bausteine in "Programmbausteine" des Projektbaums:

- Sigma-7 EtherCAT:
 - UDT 872 - VMC_ConfigSigma7EC_REF
 - FB 872 - VMC_KernelSigma7_EC
 - FB 873 - VMC_InitSigma7S_EC
- Axis Control
 - UDT 860 - MC_AXIS_REF
 - Bausteine für die gewünschten Bewegungsabläufe

Achs-DB anlegen

1. Fügen Sie Ihrem Projekt einen neuen DB als *Achs-DB* hinzu. Klicken Sie hierzu im *Projektbaum* innerhalb der CPU unter "PLC-Programm", "Programmbausteine" auf "Neuen Baustein hinzufügen", wählen Sie den Bausteintyp "DB Baustein" und vergeben Sie diesem den Namen "Axis01". Die DB-Nr. können Sie frei wählen wie z.B. DB10.

⇒ Der Baustein wird angelegt und geöffnet.

2. ■ Legen Sie in "Axis01" die Variable "Config" vom Typ UDT 872 an. Dies sind spezifische Achs-Konfigurationsdaten.
- Legen Sie in "Axis01" die Variable "Axis" vom Typ UDT 860 an. Während des Betriebs werden hier alle Betriebsdaten der Achse abgelegt.

Axis01 [DB10]

Bausteinstruktur

	Adr...	Name	Datentyp	...
	...	Config	UDT	[872]
	...	Axis	UDT	[860]

OB 1**Konfiguration der Achse**

Öffnen Sie den OB 1 und programmieren Sie folgende FB-Aufrufe mit zugehörigen DBs:

➔ FB 873 - VMC_InitSigma7S_EC, DB 873 ↪ *Kapitel 3.2.5.3 "FB 873 - VMC_InitSigma7S_EC - Sigma-7S EtherCAT Initialisierung" auf Seite 84*

Geben Sie unter *InputsStartAddressPDO* bzw. *OutputsStartAddressPDO* die Adresse aus dem *SPEED7 EtherCAT Manager* an. ↪ 60

```
⇒ CALL "VMC_InitSigma7S_EC" , "DI_InitSgm7SETC01"
   Enable           := "InitS7SEC1_Enable"
   LogicalAddress    := 300
   InputsStartAddressPDO := 300 (EtherCAT-Man.: E-Adresse S7)
   OutputsStartAddressPDO := 300 (EtherCAT-Man.: A-Adresse S7)
   EncoderType       := 1
   EncoderResolutionBits := 20
   FactorPosition     := 1.048576e+006
   FactorVelocity     := 1.048576e+006
   FactorAcceleration := 1.048576e+002
   OffsetPosition     := 0.000000e+000
   MaxVelocityApp     := 5.000000e+001
   MaxAccelerationApp := 1.000000e+002
   MaxDecelerationApp := 1.000000e+002
   MaxVelocityDrive   := 6.000000e+001
   MaxAccelerationDrive := 1.500000e+002
   MaxDecelerationDrive := 1.500000e+002
   MaxPosition        := 1.048500e+003
   MinPosition        := -1.048514e+003
   EnableMaxPosition  := TRUE
   EnableMinPosition  := TRUE
   MinUserPosition    := "InitS7SEC1_MinUserPos"
   MaxUserPosition    := "InitS7SEC1_MaxUserPos"
   Valid              := "InitS7SEC1_Valid"
   Error              := "InitS7SEC1_Error"
   ErrorID            := "InitS7SEC1_ErrorID"
   Config             := "Axis01".Config
   Axis               := "Axis01".Axis
```

**Kernel für Achse
beschalten**

Der *Kernel* verarbeitet die Benutzerkommandos und gibt sie entsprechend aufbereitet an den Antrieb über das jeweilige Bussystem weiter.

➔ FB 872 - VMC_KernelSigma7_EC, DB 872 ↪ *Kapitel 3.2.5.2 "FB 872 - VMC_KernelSigma7_EC - Sigma-7 EtherCAT Kernel" auf Seite 84*

```
⇒ CALL "VMC_KernelSigma7_EC" , "DI_KernelSgm5ETC01"
   Init := "KernelS7SEC1_Init"
   Config := "Axis01".Config
   Axis := "Axis01".Axis
```

Baustein für Bewegungsabläufe beschalten

Zur Vereinfachung soll hier die Beschaltung des FB 860 - VMC_AxisControl gezeigt werden. Dieser Universalbaustein unterstützt einfache Bewegungskommandos und liefert Statusmeldungen zurück. Die Ein- und Ausgänge können Sie individuell beschalten. Bitte geben Sie unter "Axis" die Referenz zu den entsprechenden Achsdaten im *Achs-DB* an.

➔ FB 860 - VMC_AxisControl, DB 860 ↗ *Kapitel 3.4.2.2 "FB 860 - VMC_AxisControl - Control-Baustein Achskontrolle" auf Seite 132*

```
⇒ CALL "VMC_AxisControl" , "DI_AxisControl01"
    AxisEnable      := "AxCtrl1_AxisEnable"
    AxisReset       := "AxCtrl1_AxisReset"
    HomeExecute     := "AxCtrl1_HomeExecute"
    HomePosition    := "AxCtrl1_HomePosition"
    StopExecute     := "AxCtrl1_StopExecute"
    MvVelocityExecute := "AxCtrl1_MvVelExecute"
    MvRelativeExecute := "AxCtrl1_MvRelExecute"
    MvAbsoluteExecute := "AxCtrl1_MvAbsExecute"
    PositionDistance := "AxCtrl1_PositionDistance"
    Velocity        := "AxCtrl1_Velocity"
    Acceleration     := "AxCtrl1_Acceleration"
    Deceleration     := "AxCtrl1_Deceleration"
    JogPositive      := "AxCtrl1_JogPositive"
    JogNegative      := "AxCtrl1_JogNegative"
    JogVelocity      := "AxCtrl1_JogVelocity"
    JogAcceleration  := "AxCtrl1_JogAcceleration"
    JogDeceleration  := "AxCtrl1_JogDeceleration"
    AxisReady        := "AxCtrl1_AxisReady"
    AxisEnabled      := "AxCtrl1_AxisEnabled"
    AxisError        := "AxCtrl1_AxisError"
    AxisErrorID      := "AxCtrl1_AxisErrorID"
    DriveWarning     := "AxCtrl1_DriveWarning"
    DriveError       := "AxCtrl1_DriveError"
    DriveErrorID     := "AxCtrl1_DriveErrorID"
    IsHomed          := "AxCtrl1_IsHomed"
    ModeOfOperation  := "AxCtrl1_ModeOfOperation"
    PLCOpenState     := "AxCtrl1_PLCOpenState"
    ActualPosition    := "AxCtrl1_ActualPosition"
    ActualVelocity    := "AxCtrl1_ActualVelocity"
    CmdDone          := "AxCtrl1_CmdDone"
    CmdBusy          := "AxCtrl1_CmdBusy"
    CmdAborted       := "AxCtrl1_CmdAborted"
    CmdError         := "AxCtrl1_CmdError"
    CmdErrorID       := "AxCtrl1_CmdErrorID"
    DirectionPositive := "AxCtrl1_DirectionPos"
    DirectionNegative := "AxCtrl1_DirectionNeg"
    SWLimitMinActive := "AxCtrl1_SWLimitMinActive"
    SWLimitMaxActive := "AxCtrl1_SWLimitMaxActive"
    HWLimitMinActive := "AxCtrl1_HWLimitMinActive"
    HWLimitMaxActive := "AxCtrl1_HWLimitMaxActive"
    Axis             := "Axis01".Axis
```



Für komplexe Bewegungsaufgaben können Sie die PLCopen-Bausteine verwenden. Hier müssen Sie ebenfalls unter Axis die Referenz zu den Achsdaten im Achs-DB angeben.

Ihr Projekt beinhaltet nun folgende Bausteine:

- OB 1 - Main
- OB 57 - DP Manufacturer Alarm
- OB 82 - I/O_FLT1
- OB 86 - Rack_FLT
- FB 860 - VMC_AxisControl mit Instanz-DB

- FB 872 - VMC_KernelSigma7_EC mit Instanz-DB
- FB 873 - VMC_InitSigma7S_EC mit Instanz-DB
- UDT 860 - MC_Axis_REF
- UDT 872 - VMC_ConfigSigma7EC_REF

Zeitlicher Ablauf

1. ➤ Wählen Sie *"Projekt ➔ Alles übersetzen"* und übertragen Sie das Projekt in Ihre CPU. Näheres zur Übertragung Ihres Projekt finden Sie in der Onlinehilfe zum *SPEED7 Studio*.
⇒ Sie können jetzt Ihre Applikation in Betrieb nehmen.



VORSICHT!

Bitte beachten Sie immer die Sicherheitshinweise zu ihrem Antrieb, insbesondere bei der Inbetriebnahme!

2. ➤ Bevor eine Achse gesteuert werden kann, muss diese initialisiert werden. Rufen Sie hierzu den *Init*-Baustein FB 873 - VMC_InitSigma7S_EC mit *Enable* = TRUE auf.
⇒ Der Ausgang *Valid* meldet TRUE zurück. Im Fehlerfall können Sie durch Auswertung der *ErrorID* den Fehler ermitteln.

Den *Init*-Baustein müssen Sie erneut aufrufen, wenn Sie einen neuen Achs-DB laden oder Parameter am *Init*-Baustein geändert wurden.



Fahren Sie erst fort, wenn der Init-Baustein keinen Fehler meldet!

3. ➤ Stellen Sie sicher, dass der *Kernel*-Baustein FB 872 - VMC_KernelSigma7_EC zyklisch aufgerufen wird. Auf diese Weise werden Steuersignale an den Antrieb übergeben und Statusmeldungen übermittelt.
4. ➤ Programmieren Sie Ihre Applikation mit dem FB 860 - VMC_AxisControl oder mit den PLCopen Bausteinen.

Steuerung des Antriebs über HMI

Sie haben die Möglichkeit über ein HMI Ihren Antrieb zu steuern. Hierzu gibt es für Movicon eine vorgefertigte Symbolbibliothek für den Zugriff auf den VMC_AxisControl Funktionsbaustein. ➔ *Kapitel 7 "Antrieb über HMI steuern" auf Seite 313*

3.2.4 Einsatz im Siemens SIMATIC Manager

3.2.4.1 Voraussetzung

Übersicht

- Bitte verwenden Sie für die Projektierung den Siemens SIMATIC Manager ab V 5.5 SP2.
- Die Projektierung der System SLIO CPU erfolgt im Siemens SIMATIC Manager in Form des virtuellen PROFINET IO Devices *"VIPA SLIO CPU"*. Das *"VIPA SLIO System"* ist mittels GSDML im Hardware-Katalog zu installieren.
- Die Projektierung des EtherCAT-Masters erfolgt im Siemens SIMATIC Manager in Form des virtuellen PROFINET IO Devices *"EtherCAT-Netzwerk"*. Das *"EtherCAT-Netzwerk"* ist mittels GSDML im Hardware-Katalog zu installieren.
- Das *"EtherCAT-Netzwerk"* kann mit dem VIPA-Tool *SPEED7 EtherCAT Manager* konfiguriert werden.
- Für die Projektierung des Antriebs im *SPEED7 EtherCAT Manager* ist die Installation der zugehörigen ESI-Datei erforderlich.

IO Device "VIPA SLIO System" installieren

Die Installation des PROFINET IO Devices "*VIPA SLIO CPU*" im Hardware-Katalog erfolgt nach folgender Vorgehensweise:

1. ➤ Gehen Sie in den Service-Bereich von www.vipa.com.
2. ➤ Laden Sie aus dem Downloadbereich unter "*Config Dateien ➔ PROFINET*" die Konfigurationsdatei für Ihre CPU.
3. ➤ Extrahieren Sie die Datei in Ihr Arbeitsverzeichnis.
4. ➤ Starten Sie den Hardware-Konfigurator von Siemens.
5. ➤ Schließen Sie alle Projekte.
6. ➤ Gehen Sie auf "*Extras ➔ GSD-Dateien installieren*".
7. ➤ Navigieren Sie in Ihr Arbeitsverzeichnis und installieren Sie die entsprechende GSDML-Datei.
 - ⇒ Nach der Installation finden Sie das entsprechende PROFINET IO Device unter "*PROFINET IO ➔ Weitere Feldgeräte ➔ I/O ➔ VIPA SLIO System*".

IO Device EtherCAT-Netzwerk installieren

Die Installation des PROFINET IO Devices "*EtherCAT-Netzwerk*" im Hardware-Katalog erfolgt nach folgender Vorgehensweise:

1. ➤ Gehen Sie in den Service-Bereich von www.vipa.com
2. ➤ Laden Sie aus dem Downloadbereich unter "*Config Dateien ➔ EtherCAT*" die GSDML-Datei für Ihren EtherCAT-Master.
3. ➤ Extrahieren Sie die Dateien in Ihr Arbeitsverzeichnis.
4. ➤ Starten Sie den Hardware-Konfigurator von Siemens.
5. ➤ Schließen Sie alle Projekte.
6. ➤ Gehen Sie auf "*Extras ➔ GSD-Dateien installieren*".
7. ➤ Navigieren Sie in Ihr Arbeitsverzeichnis und installieren Sie die entsprechende GSDML-Datei.
 - ⇒ Nach der Installation finden Sie das "*EtherCAT-Netzwerk*" unter "*PROFINET IO ➔ Weitere Feldgeräte ➔ I/O ➔ VIPA EtherCAT System*".

SPEED7 EtherCAT Manager installieren

Die Konfiguration des PROFINET IO Devices "*EtherCAT-Netzwerk*" erfolgt mit dem *SPEED7 EtherCAT Manager* von VIPA. Sie finden diesen Im Servicebereich von www.vipa.com unter "*Service/Support ➔ Downloads ➔ SPEED7*".

Die Installation erfolgt nach folgender Vorgehensweise:

1. ➤ Schließen Sie den Siemens SIMATIC Manager.
2. ➤ Gehen Sie in den Service-Bereich von www.vipa.com
3. ➤ Laden Sie den *SPEED7 EtherCAT Manager* und entpacken Sie diesen auf Ihren PC.
4. ➤ Zur Installation starten Sie die Datei *EtherCATManager_v... .exe*.
5. ➤ Wählen Sie die Sprache für die Installation aus.
6. ➤ Stimmen Sie dem Lizenzvertrag zu.
7. ➤ Wählen Sie das Installationsverzeichnis und starten Sie die Installation.
8. ➤ Nach der Installation müssen Sie Ihren PC neu starten
 - ⇒ Der *SPEED7 EtherCAT Manager* ist installiert und kann jetzt über das Kontextmenü des Siemens SIMATIC Manager aufgerufen werden.

3.2.4.2 Hardware-Konfiguration


CPU im Projekt anlegen

Steckpl..	Baugruppe
1	
2	CPU 315-2 PN/DP
X1	MPI/DP
X2	PN-IO
X2...	Port 1
X2...	Port 2
3	

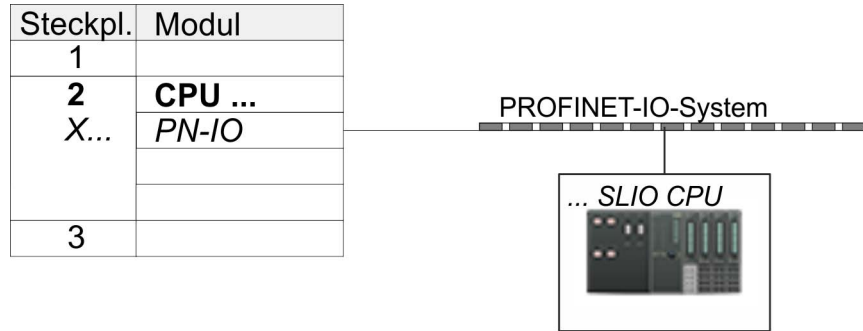
Um kompatibel mit dem Siemens SIMATIC Manager zu sein, sind folgende Schritte durchzuführen:

1. ➤ Starten Sie den Hardware-Konfigurator von Siemens mit einem neuen Projekt.
2. ➤ Fügen Sie aus dem Hardware-Katalog eine Profilschiene ein.
3. ➤ Platzieren Sie auf "Slot"-Nummer 2 die CPU 315-2 PN/DP (6ES7 315-2EH14 V3.2).
4. ➤ Über das Submodul "X1 MPI/DP" projektieren und vernetzen Sie den integrierten PROFIBUS-DP-Master (Buchse X3).
5. ➤ Über das Submodul "X2 PN-IO" projektieren Sie den EtherCAT-Master als virtuelles PROFINET-Netzwerk.
6. ➤ Klicken Sie auf das Submodul "PN-IO" der CPU.
7. ➤ Wählen Sie "Kontextmenü ➔ PROFINET IO-System einfügen".

Steckpl.	Baugruppe
1	
2	CPU ...
X...	PN-IO
3	



8. ➤ Legen Sie mit [Neu] ein neues Subnetz an und vergeben Sie gültige IP-Adress-Daten
9. ➤ Klicken Sie auf das Submodul "PN-IO" der CPU und öffnen Sie mit "Kontextmenü ➔ Objekteigenschaften" den Eigenschafts-Dialog.
10. ➤ Geben Sie unter "Allgemein" einen "Gerätenamen" an. Der Geräte name muss eindeutig am Ethernet-Subnetz sein.



Steckpl.	Baugruppe	Bestellnummer	
0	... SLIO CPU ...	015-...	
X2	015-...		
1			
2			
3			
...			

1. Navigieren Sie im Hardware-Katalog in das Verzeichnis *"PROFINET IO → Weitere Feldgeräte → I/O → VIPA SLIO System"* und binden Sie das IO-Device *"015-CEFNR00 CPU"* an Ihr PROFINET-System an.
⇒ In der Steckplatzübersicht des PROFINET-IO-Device *"VIPA SLIO CPU"* ist auf Steckplatz 0 die CPU bereits vorplatziert. Ab Steckplatz 1 können Sie Ihre System SLIO Module platzieren.

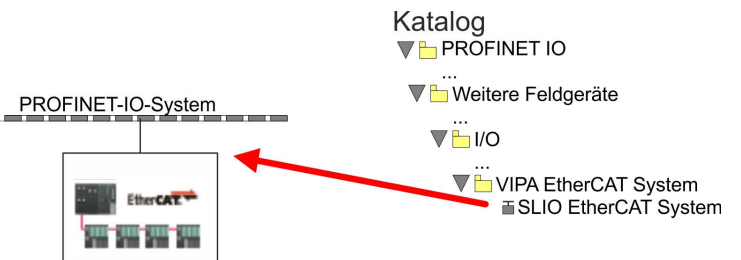
Ethernet-PG/OP-Kanal parametrieren

Steckpl.	Modul
1	
2	CPU ...
X...	PN-IO
3	
4	343-1EX30
5	
...	

1. Platzieren Sie für den Ethernet-PG/OP-Kanal auf Steckplatz 4 den Siemens CP 343-1 (SIMATIC 300 \ CP 300 \ Industrial Ethernet \ CP 343-1 \ 6GK7 343-1EX30 0XE0 V3.0).
2. Öffnen Sie durch Doppelklick auf den CP 343-1EX30 den Eigenschaften-Dialog und geben Sie für den CP unter *"Eigenschaften"* IP-Adress-Daten an. Gültige IP-Adress-Parameter erhalten Sie von Ihrem Systemadministrator.
3. Ordnen Sie den CP einem *"Subnetz"* zu. Ohne Zuordnung werden die IP-Adress-Daten nicht übernommen!

"EtherCAT-Netzwerk" einfügen

Steckpl.	Modul
1	
2	CPU ...
X...	PN-IO
3	



1. Navigieren Sie im Hardware-Katalog in das Verzeichnis *"PROFINET IO → Weitere Feldgeräte → I/O → VIPA EtherCAT System"* und binden Sie das IO Device *"SLIO EtherCAT System"* an Ihr PROFINET-System an.

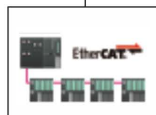
2. ➔ Klicken Sie auf das eingefügte IO Device *"EtherCAT-Netzwerk"* und definieren Sie die Bereiche für Ein- und Ausgabe, indem Sie den entsprechenden *"Out"*- bzw. *"In"*-Bereich auf einen Steckplatz ziehen.

Legen Sie folgende Bereiche an:

- In 128Byte
- Out 128Byte

Steckpl.	Modul
1	
2	CPU ...
X...	PN-IO
3	

PROFINET-IO-System



Katalog

- ▼ PROFINET IO
- ▼ Weitere Feldgeräte
- ▼ I/O
- ▼ VIPA EtherCAT System
 - SLIO EtherCAT System
 - In 1024 byte
 - In 128 byte
 - Out 1024 byte
 - Out 128 byte

Steckpl.	Baugruppe	Bestellnummer
0	...	
1	In 128 byte	
2	Out 128 byte	
3		
4		
...		

3. ➔ Wählen Sie *"Station → Speichern und übersetzen"*

Sigma-7S EtherCAT Antrieb konfigurieren

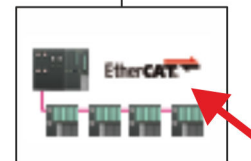
Die Konfiguration des Antriebs erfolgt im *SPEED7 EtherCAT Manager*.



Vor dem Aufruf des **SPEED7 EtherCAT Manager** müssen Sie immer Ihr Projekt mit "Station → Speichern und übersetzen" speichern.

Steckpl.	Modul
1	
2	CPU ...
X...	PN-IO
3	

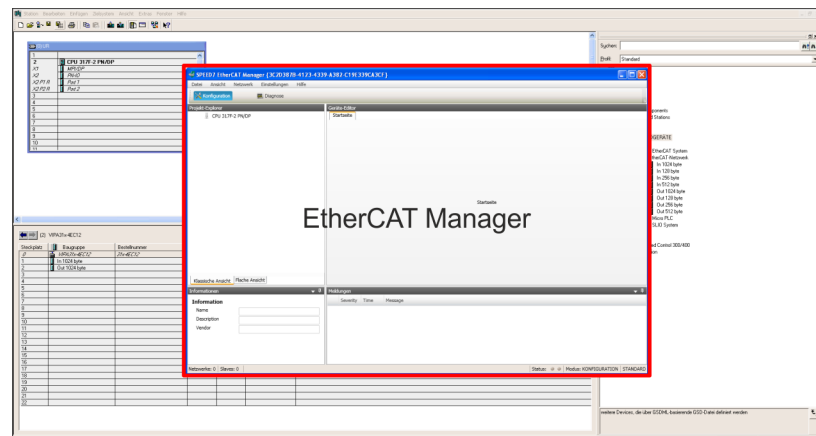
PROFINET-IO-System



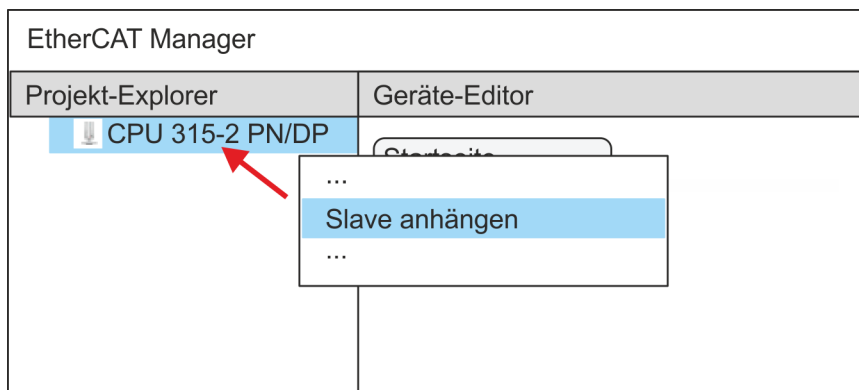
1. Klicken Sie auf das eingefügte IO Device "EtherCAT-Netzwerk" und wählen Sie "Kontextmenü → Device Tool starten → **SPEED7 EtherCAT Manager**".

⇒ Der **SPEED7 EtherCAT Manager** wird gestartet. Hier können Sie die EtherCAT-Kommunikation zu Ihrem **Sigma-7S** Antrieb konfigurieren.


Näheres zum Einsatz des **SPEED7 EtherCAT Manager** finden Sie im zugehörigen Handbuch bzw. in der Onlinehilfe.



3. Damit der **Sigma-7S** EtherCAT Antrieb im **SPEED7 EtherCAT Manager** konfiguriert werden kann, ist die entsprechende ESI-Datei zu installieren. Die ESI-Datei für den **Sigma-7S** EtherCAT Antrieb finden Sie unter www.yaskawa.eu.com unter "Service → Drives & Motion Software". Laden Sie die zu Ihrem Antrieb passende ESI-Datei herunter. Entpacken Sie diese falls erforderlich.
4. Öffnen Sie im **SPEED7 EtherCAT Manager** über "Datei → ESI-Verwaltung" das Dialogfenster "ESI-Manager".
5. Klicken Sie im "ESI-Manager" auf [Datei hinzufügen] und wählen Sie Ihre ESI-Datei aus. Mit [Öffnen] wird die ESI-Datei im **SPEED7 EtherCAT Manager** installiert.
6. Schließen Sie den "ESI-Manager".
 - ⇒ Ihr **Sigma-7S** EtherCAT Antrieb steht Ihnen nun zur Konfiguration zur Verfügung.

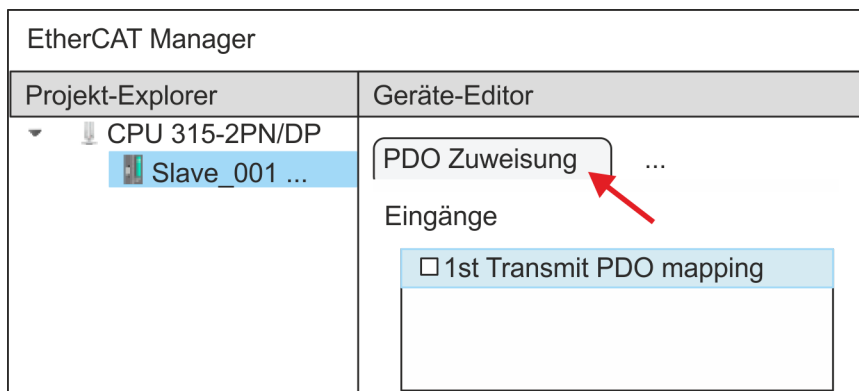


7. ➤ Klicken Sie im EtherCAT Manager auf ihre CPU und öffnen Sie über "Kontextmenü ➔ *Slave anhängen*" das Dialogfenster zum Hinzufügen eines EtherCAT-Slave.
- ⇒ Das Dialogfenster zur Auswahl eines EtherCAT-Slave wird geöffnet.
8. ➤ Wählen Sie Ihren *Sigma-7S* EtherCAT Antrieb und bestätigen Sie Ihre Auswahl mit [OK].
- ⇒ Der *Sigma-7S* EtherCAT Antrieb wird an den Master angebunden und kann nun konfiguriert werden.

9. ➤  *PDOs können Sie nur im "Experten-Modus" bearbeiten! Ansonsten werden die Schaltflächen ausgeblendet. Durch Aktivierung des "Experten-Modus" können Sie in die erweiterte Bearbeitung umschalten.*

Aktivieren Sie den *Experten-Modus* durch Aktivierung von "Ansicht ➔ Experte".

10. ➤ Klicken Sie im *SPEED7 EtherCAT Manager* auf den *Sigma-7S* EtherCAT Slave und wählen Sie im "Geräte-Editor" den Reiter "PDO-Zuweisung" an.



- ⇒ Dieser Dialog zeigt eine Auflistung aller PDOs.

Geräte-Editor

PDO Zuweisung ...

Eingänge	Ausgänge
<input type="checkbox"/> 1st Transmit PDO mapping	<input type="checkbox"/> 1st Receive PDO mapping
<input type="checkbox"/> 2nd Transmit PDO mapping	<input type="checkbox"/> 2nd Receive PDO mapping

... Bearbeiten ...

- 11.** Durch Anwahl des entsprechenden PDO-Mappings können Sie mit [Bearbeiten] die PDOs bearbeiten. Wählen Sie das Mapping "1st Transmit PDO mapping" an und klicken Sie auf [Bearbeiten].



Bitte beachten Sie, dass aufgrund der Voreinstellung manche PDOs nicht bearbeitet werden können. Durch Deaktivierung bereits aktivierter PDOs können Sie die Bearbeitung von gesperrten PDOs frei geben.

Allgemein

Name: 1st Transmit PDO mapping

Index: 0x1A00 Dez Hex

Flags

☐ Zwingend ☒ Richtung

☐ Schreibgeschützt ☐ TxPdo (Eingang)

☐ Virtuell ☐ RxPdo (Ausgang)

Optional

Ausschließen:

☐ 1A01

☒ 1A02

☒ 1A03

Einträge

Name	Index	Bitlänge	Kommentar
Status word	0x6041:00	16	
Position actual internal value	0x6063:00	32	
Position actual value	0x6064:00	32	
Torque actual value	0x6077:00	16	
Following error actual value	0x60F4:00	32	
Modes of operation display	0x6061:00	8	
---	---	8	---
Digital inputs	0x60FD:00	32	

Neu Löschen Bearbeiten Nach oben Nach unten

- ⇒ Es öffnet sich der Dialog *"PDO bearbeiten"*. Bitte überprüfen Sie hier die aufgeführten PDO-Einstellungen und passen Sie diese ggf. an. Bitte berücksichtigen Sie hierbei auch die Reihenfolge der *"Einträge"* und ergänzen Sie diese entsprechend.

Für die Bearbeitung der *"Einträge"* stehen folgende Funktionen zur Verfügung:

- Neu
 - Hiermit können Sie in einem Dialogfenster einen neuen Eintrag anlegen, indem Sie aus dem *"CoE-Objektverzeichnis"* den entsprechenden Eintrag auswählen und Ihre Einstellungen vornehmen. Mit [OK] wird der Eintrag übernommen und in der Liste der Einträge aufgeführt.
- Löschen
 - Hiermit können Sie den angewählte Eintrag löschen.
- Bearbeiten
 - Hiermit können Sie allgemeinen Daten eines Eintrags bearbeiten.
- Nach oben/unten
 - Hiermit können Sie den angewählten Eintrag in der List nach oben bzw. nach unten bewegen.

12. Führen Sie folgende Einstellungen durch:

Eingänge: 1st Transmit PDO 0x1A00

- Allgemein
 - Name: 1st Transmit PDO mapping
 - Index: 0x1A00
- Flags
 - Alles deaktiviert
- Richtung
 - TxPdo (Eingang): aktiviert
- Ausschließen

Bitte diese Einstellungen beachten, da ansonsten die PDO-Mappings nicht zeitgleich aktiviert werden können!

 - 1A01: deaktiviert
- Einträge

Name	Index	Bitlänge
Status word	0x6041:00	16Bit
Position actual internal value	0x6063:00	32Bit
Position actual value	0x6064:00	32Bit
Torque actual value	0x6077:00	16Bit
Following error actual value	0x60F4:00	32Bit
Modes of operation display	0x6061:00	8Bit
---	---	8Bit
Digital inputs	0x60FD:00	32Bit

Schließen Sie den Dialog *"PDO bearbeiten"* mit [OK].

- 13.** Wählen Sie das Mapping *"2nd Transmit PDO mapping"* an und klicken Sie auf [Bearbeiten]. Führen Sie folgende Einstellungen durch:

Eingänge: 2nd Transmit PDO 0x1A01

- Allgemein
 - Name: 2nd Transmit PDO mapping
 - Index: 0x1A01
- Flags
 - Alles deaktiviert
- Richtung
 - TxPdo (Eingang): aktiviert
- Ausschließen

Bitte diese Einstellungen beachten, da ansonsten die PDO-Mappings nicht zeitgleich aktiviert werden können!

 - 1A00: deaktiviert
 - 1A02: deaktiviert
 - 1A03: deaktiviert
- Einträge

Name	Index	Bitlänge
Touch probe status	0x60B9:00	16Bit
Touch probe 1 position value	0x60BA:00	32Bit
Touch probe 2 position value	0x60BC:00	32Bit
Velocity actual value	0x606C:00	32Bit

Schließen Sie den Dialog *"PDO bearbeiten"* mit [OK].

- 14.** ➤ Wählen Sie das Mapping "*1st Receive PDO mapping*" an und klicken Sie auf [Bearbeiten]. Führen Sie folgende Einstellungen durch:

Ausgänge: 1st Receive PDO 0x1600

- Allgemein
 - Name: 1st Receive PDO mapping
 - Index: 0x1600
- Flags
 - Alles deaktiviert
- Richtung
 - RxPdo (Ausgang): aktiviert
- Ausschließen

Bitte diese Einstellungen beachten, da ansonsten die PDO-Mappings nicht zeitgleich aktiviert werden können!

 - 1601: deaktiviert
 - 1602: deaktiviert
 - 1603: deaktiviert
- Einträge

Name	Index	Bitlänge
Control word	0x6040:00	16 Bit
Target position	0x607A:00	32 Bit
Target velocity	0x60FF:00	32 Bit
Modes of operation	0x6060:00	8 Bit
---	---	8 Bit
Touch probe function	0x60B8:00	16 Bit

Schließen Sie den Dialog "*PDO bearbeiten*" mit [OK].

- 15.** Wählen Sie das Mapping *"2nd Receive PDO mapping"* an und klicken Sie auf [Bearbeiten]. Führen Sie folgende Einstellungen durch:

Ausgänge: 2nd Receive PDO 0x1601

- Allgemein
 - Name: 2nd Receive PDO mapping
 - Index: 0x1601
- Flags
 - Alles deaktiviert
- Richtung
 - RxPdo (Ausgang): aktiviert
- Ausschließen

Bitte diese Einstellungen beachten, da ansonsten die PDO-Mappings nicht zeitgleich aktiviert werden können!

 - 1600: deaktiviert
 - 1602: aktiviert
 - 1603: aktiviert
- Einträge

Name	Index	Bitlänge
Profile velocity	0x6081:00	32Bit
Profile acceleration	0x6083:00	32Bit
Profile deceleration	0x6084:00	32Bit

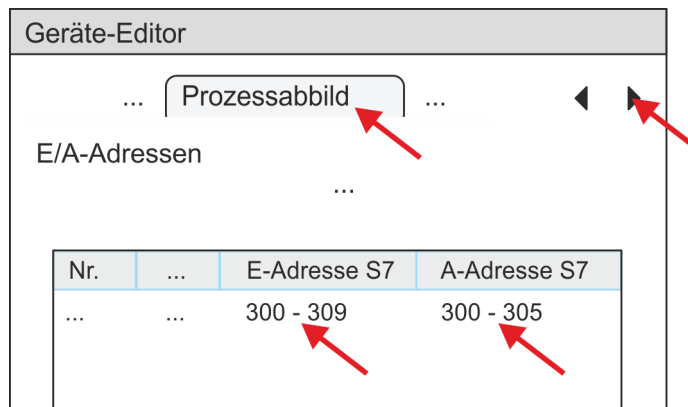
Schließen Sie den Dialog *"PDO bearbeiten"* mit [OK].

- 16.** Aktivieren Sie in PDO-Zuweisung die PDOs 1 und 2 für die Ein und Ausgänge. Alle nachfolgenden PDOs müssen deaktiviert bleiben. Sollte dies nicht möglich sein, überprüfen Sie bitte den jeweiligen PDO-Parameter *"Ausschließen"*.

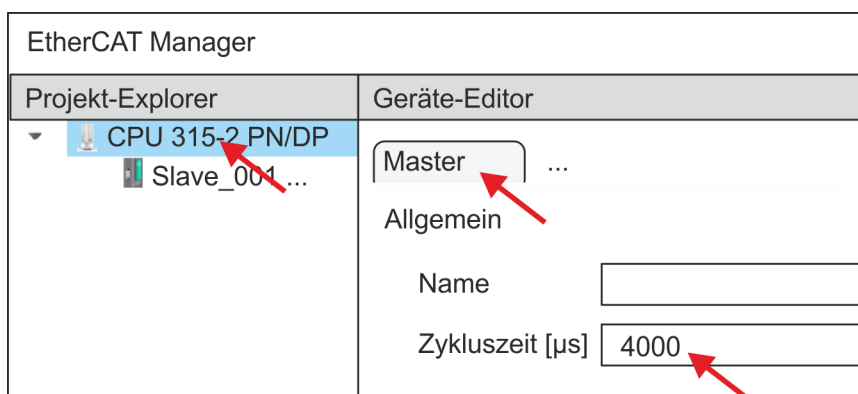
- 17.** Wählen Sie im *"Geräte-Editor"* des *SPEED7 EtherCAT Manager* den Reiter *"Verteilte Uhren"* an und stellen Sie *"DC unused"* als *"Betriebsart"* ein.

- 18.** Wählen Sie im "Geräte-Editor" über die Pfeiltaste den Reiter "Prozessabbild" an und notieren Sie sich für die Parameter des Bausteins FB 873 - VMC_InitSigma7S_EC folgende PDO-Anfangsadressen:

- "E-Adresse S7" → "InputsStartAddressPDO"
- "A-Adresse S7" → "OutputsStartAddressPDO"



- 19.** Klicken Sie im *SPEED7 EtherCAT Manager* auf ihre CPU und wählen Sie im "Geräte-Editor" den Reiter "Master" an.

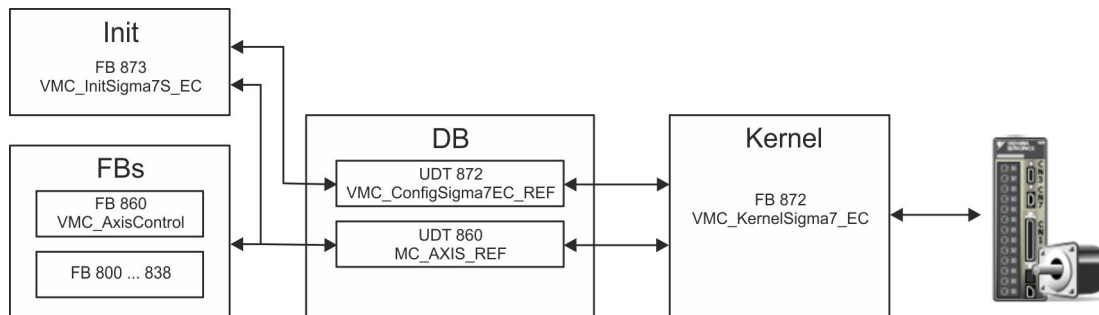


- ⇒ Stellen Sie für Sigma-7S (400V) Antriebe (SGD7S-xxxDA0... und SGD7S-xxxxA0...) eine Zykluszeit von mindestens 4ms ein. Ansonsten lassen Sie den Wert bei 1ms.

- 20.** Indem Sie den Dialog des *SPEED7 EtherCAT Manager* mit [X] schließen, wird die Konfiguration in die Projektierung übernommen. Sie können Ihre EtherCAT-Konfiguration jederzeit im *SPEED7 EtherCAT Manager* wieder bearbeiten, da die Konfiguration in Ihrem Projekt gespeichert wird.
- 21.** Speichern und übersetzen Sie Ihre Konfiguration.

3.2.4.3 Anwender-Programm

3.2.4.3.1 Programmstruktur



- **DB**
Für jede Achse ist ein Datenbaustein (Achs-DB) für Konfiguration und Statusdaten anzulegen. Der Datenbaustein besteht aus folgenden Datenstrukturen:
 - UDT 872 - *VMC_ConfigSigma7EC_REF*
Die Datenstruktur beschreibt den Aufbau der Konfiguration des Antriebs. Spezifische Datenstruktur für *Sigma-7* EtherCAT.
 - UDT 860 - *MC_AXIS_REF*
Die Datenstruktur beschreibt den Aufbau der Parameter und Statusinformationen von Antrieben. Allgemeine Datenstruktur für alle Antriebe und Bussysteme.
- **FB 873 - *VMC_InitSigma7S_EC***
 - Der *Init*-Baustein dient zur Konfiguration einer Achse.
 - Spezifischer Baustein für *Sigma-7S* EtherCAT.
 - Die Konfigurationsdaten für die Initialisierung sind im *Achs-DB* abzulegen.
- **FB 872 - *VMC_KernelSigma7_EC***
 - Der *Kernel*-Baustein kommuniziert mit dem Antrieb über das entsprechende Bussystem, verarbeitet die Benutzeraufträge und liefert Statusmeldungen zurück.
 - Spezifischer Baustein für *Sigma-7* EtherCAT.
 - Der Austausch der Daten erfolgt mittels des *Achs-DB*.
- **FB 860 - *VMC_AxisControl***
 - Universal-Baustein für alle Antriebe und Bussysteme.
 - Unterstützt einfache Bewegungskommandos und liefert alle relevanten Statusmeldungen.
 - Der Austausch der Daten erfolgt mittels des *Achs-DB*.
 - Über die Instanzdaten des Bausteins können Sie zur Bewegungssteuerung und Statusabfrage eine Visualisierung anbinden.
 - Zusätzlich zum FB 860 - *VMC_AxisControl* haben Sie die Möglichkeit *PLCopen*-Bausteine zu nutzen.
- **FB 800 ... FB 838 - *PLCopen***
 - Die *PLCopen*-Bausteine dienen zur Programmierung von Bewegungsabläufen und Statusabfragen.
 - Allgemeine Bausteine für alle Antriebe und Bussysteme.

3.2.4.3.2 Programmierung

Bibliothek einbinden

1. ➤ Gehen Sie in den Service-Bereich von www.vipa.com.
2. ➤ Laden Sie aus dem Downloadbereich unter "*VIPA Lib*" die *Simple Motion Control Library*.

3. ➔ Öffnen Sie mit *"Datei → Dearchivieren"* das Dialogfenster zur Auswahl der ZIP-Datei.
4. ➔ Wählen Sie die entsprechende ZIP-Datei an und klicken Sie auf [Öffnen].
5. ➔ Geben Sie ein Zielverzeichnis an, in dem die Bausteine abzulegen sind und starten Sie den Entpackvorgang mit [OK].

Bausteine in Projekt kopieren

- ➔ Öffnen Sie die Bibliothek nach dem Entpackvorgang und ziehen Sie per Drag&Drop folgende Bausteine in *"Bausteine"* Ihres Projekts:
- *Sigma-7S EtherCAT*:
 - UDT 872 - VMC_ConfigSigma7EC_REF
 - FB 872 - VMC_KernelSigma7_EC
 - FB 873 - VMC_InitSigma7S_EC
 - *Axis Control*
 - UDT 860 - MC_AXIS_REF
 - Bausteine für die gewünschten Bewegungsabläufe

Alarm-OBs anlegen

1. ➔ Klicken Sie in Ihrem Projekt auf *"Bausteine"* und wählen Sie *"Kontextmenü → Neues Objekt einfügen → Organisationsbaustein"*.
⇒ Das Dialogfenster *"Eigenschaften Organisationsbaustein"* öffnet sich.
2. ➔ Fügen Sie nacheinander OB 57, OB 82 und OB 86 Ihrem Projekt hinzu.

Achs-DB anlegen

1. ➔ Klicken Sie in Ihrem Projekt auf *"Bausteine"* und wählen Sie *"Kontextmenü → Neues Objekt einfügen → Datenbaustein"*.
Geben Sie folgende Parameter an:
 - Name und Typ
 - Die DB-Nr. als *"Name"* können Sie frei wählen wie z.B. DB10.
 - Stellen Sie *"Global-DB"* als *"Typ"* ein.
 - Symbolischer Name
 - Geben Sie *"Axis01"* an.
 Bestätigen Sie Ihre Eingaben mit [OK].
⇒ Der Baustein wird angelegt.
2. ➔ Öffnen Sie DB10 "Axis01" durch Doppelklick.
 - Legen Sie in "Axis01" die Variable "Config" vom Typ UDT 872 an. Dies sind spezifische Achs-Konfigurationsdaten.
 - Legen Sie in "Axis01" die Variable "Axis" vom Typ UDT 860 an. Während des Betriebs werden hier alle Betriebsdaten der Achse abgelegt.

DB10

Adresse	Name	Typ	...
		Struct	
...	Config	"VMC_ConfigSigma7EC_REF"	
...	Axis	"MC_AXIS_REF"	
...		END_STRUCT	

OB 1**Konfiguration der Achse**

Öffnen Sie den OB 1 und programmieren Sie folgende FB-Aufrufe mit zugehörigen DBs:

➔ FB 873 - VMC_InitSigma7S_EC, DB 873 ↪ *Kapitel 3.2.5.3 "FB 873 - VMC_InitSigma7S_EC - Sigma-7S EtherCAT Initialisierung" auf Seite 84*

Geben Sie unter *InputsStartAddressPDO* bzw. *OutputsStartAddressPDO* die Adresse aus dem *SPEED7 EtherCAT Manager* an. ↪ 78

```
⇒ CALL "VMC_InitSigma7S_EC" , "DI_InitSgm7SETC01"
   Enable           := "InitS7SEC1_Enable"
   LogicalAddress    := 300
   InputsStartAddressPDO := 300 (EtherCAT-Man.: E-Adresse S7)
   OutputsStartAddressPDO := 300 (EtherCAT-Man.: A-Adresse S7)
   EncoderType       := 1
   EncoderResolutionBits := 20
   FactorPosition     := 1.048576e+006
   FactorVelocity     := 1.048576e+006
   FactorAcceleration := 1.048576e+002
   OffsetPosition     := 0.000000e+000
   MaxVelocityApp     := 5.000000e+001
   MaxAccelerationApp := 1.000000e+002
   MaxDecelerationApp := 1.000000e+002
   MaxVelocityDrive   := 6.000000e+001
   MaxAccelerationDrive := 1.500000e+002
   MaxDecelerationDrive := 1.500000e+002
   MaxPosition        := 1.048500e+003
   MinPosition        := -1.048514e+003
   EnableMaxPosition  := TRUE
   EnableMinPosition  := TRUE
   MinUserPosition     := "InitS5EC1_MinUserPos"
   MaxUserPosition     := "InitS5EC1_MaxUserPos"
   Valid              := "InitS5EC1_Valid"
   Error              := "InitS5EC1_Error"
   ErrorID            := "InitS5EC1_ErrorID"
   Config             := "Axis01".Config
   Axis               := "Axis01".Axis
```

**Kernel für Achse
beschalten**

Der *Kernel* verarbeitet die Benutzerkommandos und gibt sie entsprechend aufbereitet an den Antrieb über das jeweilige Bussystem weiter.

➔ FB 872 - VMC_KernelSigma7_EC, DB 872 ↪ *Kapitel 3.2.5.2 "FB 872 - VMC_KernelSigma7_EC - Sigma-7 EtherCAT Kernel" auf Seite 84*

```
⇒ CALL "VMC_KernelSigma7_EC" , "DI_KernelSgm7ETC01"
   Init := "KernelS7EC1_Init"
   Config := "Axis01".Config
   Axis := "Axis01".Axis
```

Baustein für Bewegungsabläufe beschalten

Zur Vereinfachung soll hier die Beschaltung des FB 860 - VMC_AxisControl gezeigt werden. Dieser Universalbaustein unterstützt einfache Bewegungskommandos und liefert Statusmeldungen zurück. Die Ein- und Ausgänge können Sie individuell beschalten. Bitte geben Sie unter "Axis" die Referenz zu den entsprechenden Achsdaten im *Achs-DB* an.

➔ FB 860 - VMC_AxisControl, DB 860 ↗ *Kapitel 3.4.2.2 "FB 860 - VMC_AxisControl - Control-Baustein Achskontrolle" auf Seite 132*

```

⇒      CALL "VMC_AxisControl" , "DI_AxisControl01"
        AxisEnable       := "AxCtrl1_AxisEnable"
        AxisReset        := "AxCtrl1_AxisReset"
        HomeExecute      := "AxCtrl1_HomeExecute"
        HomePosition     := "AxCtrl1_HomePosition"
        StopExecute      := "AxCtrl1_StopExecute"
        MvVelocityExecute := "AxCtrl1_MvVelExecute"
        MvRelativeExecute := "AxCtrl1_MvRelExecute"
        MvAbsoluteExecute := "AxCtrl1_MvAbsExecute"
        PositionDistance := "AxCtrl1_PositionDistance"
        Velocity         := "AxCtrl1_Velocity"
        Acceleration     := "AxCtrl1_Acceleration"
        Deceleration     := "AxCtrl1_Deceleration"
        JogPositive      := "AxCtrl1_JogPositive"
        JogNegative      := "AxCtrl1_JogNegative"
        JogVelocity      := "AxCtrl1_JogVelocity"
        JogAcceleration  := "AxCtrl1_JogAcceleration"
        JogDeceleration  := "AxCtrl1_JogDeceleration"
        AxisReady       := "AxCtrl1_AxisReady"
        AxisEnabled      := "AxCtrl1_AxisEnabled"
        AxisError        := "AxCtrl1_AxisError"
        AxisErrorID      := "AxCtrl1_AxisErrorID"
        DriveWarning     := "AxCtrl1_DriveWarning"
        DriveError       := "AxCtrl1_DriveError"
        DriveErrorID     := "AxCtrl1_DriveErrorID"
        IsHomed          := "AxCtrl1_IsHomed"
        ModeOfOperation := "AxCtrl1_ModeOfOperation"
        PLCOpenState     := "AxCtrl1_PLCOpenState"
        ActualPosition   := "AxCtrl1_ActualPosition"
        ActualVelocity   := "AxCtrl1_ActualVelocity"
        CmdDone          := "AxCtrl1_CmdDone"
        CmdBusy          := "AxCtrl1_CmdBusy"
        CmdAborted       := "AxCtrl1_CmdAborted"
        CmdError         := "AxCtrl1_CmdError"
        CmdErrorID       := "AxCtrl1_CmdErrorID"
        DirectionPositive := "AxCtrl1_DirectionPos"
        DirectionNegative := "AxCtrl1_DirectionNeg"
        SWLimitMinActive := "AxCtrl1_SWLimitMinActive"
        SWLimitMaxActive := "AxCtrl1_SWLimitMaxActive"
        HWLimitMinActive := "AxCtrl1_HWLimitMinActive"
        HWLimitMaxActive := "AxCtrl1_HWLimitMaxActive"
        Axis             := "Axis01".Axis
  
```



Für komplexe Bewegungsaufgaben können Sie die PLCopen-Bausteine verwenden. Hier müssen Sie ebenfalls unter Axis die Referenz zu den Achsdaten im Achs-DB angeben.

Ihr Projekt beinhaltet nun folgende Bausteine:

- OB 1 - Main
- OB 57 - DP Manufacturer Alarm
- OB 82 - I/O_FLT1
- OB 86 - Rack_FLT
- FB 860 - VMC_AxisControl mit Instanz-DB

- FB 872 - VMC_KernelSigma7_EC mit Instanz-DB
- FB 873 - VMC_InitSigma7S_EC mit Instanz-DB
- UDT 860 - MC_Axis_REF
- UDT 872 - VMC_ConfigSigma7EC_REF

Zeitlicher Ablauf

1. ➔ Wechseln Sie in den Siemens SIMATIC Manager und übertragen Sie Ihr Projekt in die CPU.

Die Übertragung kann ausschließlich aus dem Siemens SIMATIC Manager erfolgen - nicht Hardware-Konfigurator!



Da Slave- und Modulparameter mittels SDO-Zugriff bzw. SDO-Init-Kommando übertragen werden, bleibt die Parametrierung solange bestehen, bis ein Power-Cycle durchgeführt wird oder neue Parameter für die gleichen SDO-Objekte übertragen werden.

Beim Urlöschen werden Slave- und Modul-Parameter nicht zurückgesetzt!

⇒ Sie können jetzt Ihre Applikation in Betrieb nehmen.



VORSICHT!

Bitte beachten Sie immer die Sicherheitshinweise zu ihrem Antrieb, insbesondere bei der Inbetriebnahme!

2. ➔ Bevor eine Achse gesteuert werden kann, muss diese initialisiert werden. Rufen Sie hierzu den *Init*-Baustein FB 873 - VMC_InitSigma7S_EC mit *Enable* = TRUE auf.

⇒ Der Ausgang *Valid* meldet TRUE zurück. Im Fehlerfall können Sie durch Auswertung der *ErrorID* den Fehler ermitteln.

Den *Init*-Baustein müssen Sie erneut aufrufen, wenn Sie einen neuen Achs-DB laden oder Parameter am *Init*-Baustein geändert wurden.



Fahren Sie erst fort, wenn der Init-Baustein keinen Fehler meldet!

3. ➔ Stellen Sie sicher, dass der *Kernel*-Baustein FB 872 - VMC_KernelSigma7_EC zyklisch aufgerufen wird. Auf diese Weise werden Steuersignale an den Antrieb übergeben und Statusmeldungen übermittelt.
4. ➔ Programmieren Sie Ihre Applikation mit dem FB 860 - VMC_AxisControl oder mit den PLCopen Bausteinen auf.

Steuerung des Antriebs über HMI

Sie haben die Möglichkeit über ein HMI Ihren Antrieb zu steuern. Hierzu gibt es für Movicon eine vorgefertigte Symbolbibliothek für den Zugriff auf den VMC_AxisControl Funktionsbaustein. ➔ Kapitel 7 "Antrieb über HMI steuern" auf Seite 313

3.2.4.4 Projekt kopieren

Vorgehensweise

Im Beispiel wird die Station "Source" kopiert und als "Target" gespeichert.

1. Öffnen Sie die Hardware-Konfiguration der "Source"-CPU und starten Sie hier den *SPEED7 EtherCAT Manager*.
2. Speichern Sie im *SPEED7 EtherCAT Manager* über "Datei → Speichern unter" die Konfiguration in Ihrem Arbeitsverzeichnis.
3. Schließen Sie den *SPEED7 EtherCAT Manager* und den Hardware-Konfigurator wieder.
4. Kopieren Sie die Station "Source" mit Strg+C und fügen Sie diese mit Strg+V als "Target" in Ihr Projekt ein.
5. Wechseln Sie in den "Baustein"-Ordner der "Target"-CPU löschen Sie die "Systemdaten".
6. Öffnen Sie die Hardware-Konfiguration der "Target"-CPU. Passen Sie die IP-Adressdaten an oder vernetzen Sie die CPU bzw. den CP neu.



Vor dem Aufruf des SPEED7 EtherCAT Manager müssen Sie immer Ihr Projekt mit "Station → Speichern und übersetzen" speichern.

7. Speichern Sie Ihr Projekt mit "Station → Speichern und übersetzen".
8. Öffnen Sie den *SPEED7 EtherCAT Manager*.
9. Laden Sie mit "Datei → Öffnen" die Konfiguration aus Ihrem Arbeitsverzeichnis.
10. Schließen Sie den *SPEED7 EtherCAT Manager* wieder.
11. Speichern und übersetzen Sie Ihre Konfiguration.

3.2.5 Antriebsspezifische Bausteine

3.2.5.1 UDT 872 - VMC_ConfigSigma7EC_REF - Sigma-7 EtherCAT Datenstruktur Achskonfiguration

Dies ist eine benutzerdefinierte Datenstruktur, die Informationen zu den Konfigurationsdaten beinhaltet. Die UDT ist speziell angepasst an die Verwendung eines *Sigma-7*-Antriebs, welcher über EtherCAT angebunden ist.

3.2.5.2 FB 872 - VMC_KernelSigma7_EC - Sigma-7 EtherCAT Kernel

Beschreibung

Dieser Baustein setzt die Antriebskommandos für eine *Sigma-7* Achse über EtherCAT um und kommuniziert mit dem Antrieb. Je *Sigma-7* Achse ist eine Instanz dieses FBs zyklisch aufzurufen.



Bitte beachten Sie, dass dieser Baustein intern den SFB 238 aufruft.

Im SPEED7 Studio wird dieser Baustein automatisch in Ihr Projekt eingefügt.

Im Siemens SIMATIC Manager müssen Sie den SFB 238 aus der Motion Control Library in Ihr Projekt kopieren.

Parameter	Deklaration	Datentyp	Beschreibung
Init	INPUT	BOOL	Mit einer Flanke 0-1 wird der Baustein intern zurückgesetzt. Hierbei werden bestehende Bewegungskommandos abgebrochen und der Baustein wird initialisiert.
Config	IN_OUT	UDT872	Datenstruktur zur Übergabe von achsabhängigen Konfigurationsdaten an den <i>AxisKernel</i> .
Axis	IN_OUT	MC_AXIS_REF	Datenstruktur zur Übergabe von achsabhängigen Informationen an <i>AxisKernel</i> und PLCopen-Bausteine.

3.2.5.3 FB 873 - VMC_InitSigma7S_EC - Sigma-7S EtherCAT Initialisierung

Beschreibung

Dieser Baustein dient zur Konfiguration der Achse. Der Baustein ist speziell angepasst an die Verwendung eines *Sigma-7*-Antriebs, welcher über EtherCAT angebunden ist.

Parameter	Deklaration	Datentyp	Beschreibung
Config	IN_OUT	UDT872	Datenstruktur zur Übergabe von achsabhängigen Konfigurationsdaten an den <i>AxisKernel</i> .
Axis	IN_OUT	MC_AXIS_REF	Datenstruktur zur Übergabe von achsabhängigen Informationen an <i>AxisKernel</i> und PLCopen-Bausteine.
Enable	INPUT	BOOL	Freigabe der Initialisierung
LogicalAddress	INPUT	INT	Startadresse der PDO-Eingangsdaten
InputsStartAddressPDO	INPUT	INT	Startadresse der Eingabe-PDOs
OutputsStartAddressPDO	INPUT	INT	Startadresse der Ausgabe-PDOs
EncoderType	INPUT	INT	Encoder-Typ <ul style="list-style-type: none"> 1: Absolut-Encoder 2: Inkremental-Encoder

Parameter	Deklaration	Datentyp	Beschreibung
EncoderResolutionBits	INPUT	INT	Anzahl der Bits, die einer Geber-Umdrehung entsprechen. Default: 20
FactorPosition	INPUT	REAL	Faktor zur Umrechnung der Position von Benutzereinheiten [u] in Antriebseinheiten [Inkrement] und zurück. Es gilt: $p_{[\text{Inkrement}]} = p_{[u]} \times \text{FactorPosition}$ Bitte berücksichtigen sie auch den Faktor, welchen Sie am Antrieb über die Objekte 0x2701:1 und 0x2701:2 vorgeben können. Dieser sollte 1 sein.
FactorVelocity	INPUT	REAL	Faktor zur Umrechnung der Geschwindigkeit von Benutzereinheiten [u/s] in Antriebseinheiten [Inkrement/s] und zurück. Es gilt: $v_{[\text{Inkrement/s}]} = v_{[u/s]} \times \text{FactorVelocity}$ Bitte berücksichtigen sie auch den Faktor, welchen Sie am Antrieb über die Objekte 0x2702:1 und 0x2702:2 vorgeben können. Dieser sollte 1 sein.
FactorAcceleration	INPUT	REAL	Faktor zur Umrechnung der Beschleunigung von Benutzereinheiten [u/s ²] in Antriebseinheiten [10 ⁻⁴ x Inkrement/s ²] und zurück. Es gilt: $10^{-4} \times a_{[\text{Inkrement/s}^2]} = a_{[u/s^2]} \times \text{FactorAcceleration}$ Bitte berücksichtigen sie auch den Faktor, welchen Sie am Antrieb über die Objekte 0x2703:1 und 0x2703:2 vorgeben können. Dieser sollte 1 sein.
OffsetPosition	INPUT	REAL	Offset für die Nullposition [u].
MaxVelocityApp	INPUT	REAL	Maximale Geschwindigkeit der Applikation [u/s]. Die Kommandoeingaben werden vor Ausführung auf den Maximalwert überprüft.
MaxAccelerationApp	INPUT	REAL	Maximale Beschleunigung der Applikation [u/s ²]. Die Kommandoeingaben werden vor Ausführung auf den Maximalwert überprüft.
MaxDecelerationApp	INPUT	REAL	Maximale Verzögerung der Applikation [u/s ²]. Die Kommandoeingaben werden vor Ausführung auf den Maximalwert überprüft.
MaxPosition	INPUT	REAL	Maximale Position für die Überwachung der Softwarelimits [u].
MinPosition	INPUT	REAL	Minimale Position für die Überwachung der Softwarelimits [u].
EnableMaxPosition	INPUT	BOOL	Überwachung maximale Position ■ TRUE: Aktivierung der Überwachung der maximalen Position.
EnableMinPosition	INPUT	BOOL	Überwachung minimale Position ■ TRUE: Aktivierung der Überwachung der minimalen Position.
MinUserPosition	OUTPUT	REAL	Minimale Benutzerposition basierend auf dem minimalen Encoder Wert von 0x80000000 und dem <i>FactorPosition</i> [u].

Parameter	Deklaration	Datentyp	Beschreibung
MaxUserPosition	OUTPUT	REAL	Maximale Benutzerposition basierend auf dem maximalen Encoder Wert von 0x7FFFFFFF und dem <i>Factor-Position</i> [u].
Valid	OUTPUT	BOOL	Initialisierung ■ TRUE: Initialisierung ist gültig.
Error	OUTPUT	BOOL	■ Fehler – TRUE: Ein Fehler ist aufgetreten. Zusätzliche Fehlerinformationen können dem Parameter <i>ErrorID</i> entnommen werden. Die Achse wird gesperrt.
ErrorID	OUTPUT	WORD	Zusätzliche Fehlerinformationen 🔗 Kapitel 9 "ErrorID - Zusätzliche Fehlerinformationen" auf Seite 340

3.3 Einsatz *Sigma-7W EtherCAT*

3.3.1 Übersicht

Einsatz des Singelachs-Antriebs 🔗 Kapitel 3.2 "Einsatz *Sigma-7S EtherCAT*" auf Seite 47

Voraussetzung

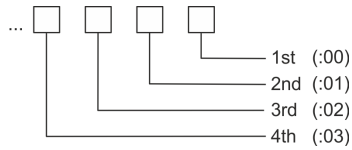
- SPEED7 Studio ab V1.6.1
oder
- Siemens SIMATIC Manager ab V 5.5 SP2 & *SPEED7 EtherCAT Manager & Simple Motion Control Library*
- CPU mit EtherCAT-Master wie z.B. CPU 015-CEFNR00
- *Sigma-7W* Doppelachs-Antrieb mit EtherCAT-Optionskarte

Schritte der Projektierung

1. ➤ Parameter am Antrieb einstellen
 - Die Einstellung der Parameter hat mit dem Softwaretool *Sigma Win+* zu erfolgen.
2. ➤ Hardwarekonfiguration im VIPA *SPEED7 Studio* oder Siemens SIMATIC Manager
 - Projektierung einer CPU mit EtherCAT-Master-Funktionalität.
 - Projektierung der *Sigma-7W* EtherCAT Doppelachsen.
 - Projektierung der EtherCAT-Anbindung über *SPEED7 EtherCAT Manager*.
3. ➤ Programmierung im VIPA *SPEED7 Studio* oder Siemens SIMATIC Manager
 - *Init*-Baustein zur Konfiguration der Doppel-Achsen beschalten.
 - *Kernel*-Baustein zur Kommunikation mit je einer Achse beschalten.
 - Bausteine für die Bewegungsabläufe beschalten.

3.3.2 Parameter am Antrieb einstellen

Parameter-Digits



VORSICHT!

Vor der Erstinbetriebnahme müssen Sie Ihren Antrieb mit dem Software-tool *Sigma Win+* an Ihre Applikation anpassen! Näheres hierzu finden Sie im Handbuch zu Ihrem Antrieb.

Zur Abstimmung auf die *Simple Motion Control Library* sind folgende Parameter über *Sigma Win+* einzustellen:

Achse 1 - Module 1 (24Bit Encoder)

Servopack Parameter	Adresse:Digit	Name	Wert
Pn205	(2205h)	Multiturn Limit Setting	65535
Pn20E	(220Eh)	ElectronicGear Ratio (Numerator)	16
Pn210	(2210h)	Electronic Gear Ratio (Denominator)	1
PnB02	(2701h:01)	Position User Unit (Numerator)	1
PnB04	(2701h:02)	Position User Unit (Denominator)	1
PnB06	(2702h:01)	Velocity User Unit (Numerator)	1
PnB08	(2702h:02)	Velocity User Unit (Denominator)	1
PnB0A	(2703h:01)	Acceleration User Unit (Numerator)	1
PnB0C	(2703h:02)	Acceleration User Unit (Denominator)	1

Achse 2 - Module 2 (24Bit Encoder)

Servopack Parameter	Adresse:Digit	Name	Wert
Pn205	(2A05h)	Multiturn Limit Setting	65535
Pn20E	(2A0Eh)	ElectronicGear Ratio (Numerator)	16
Pn210	(2A10h)	Electronic Gear Ratio (Denominator)	1
PnB02	(2F01h:01)	Position User Unit (Numerator)	1
PnB04	(2F01h:02)	Position User Unit (Denominator)	1
PnB06	(2F02h:01)	Velocity User Unit (Numerator)	1
PnB08	(2F02h:02)	Velocity User Unit (Denominator)	1
PnB0A	(2F03h:01)	Acceleration User Unit (Numerator)	1
PnB0C	(2F03h:02)	Acceleration User Unit (Denominator)	1

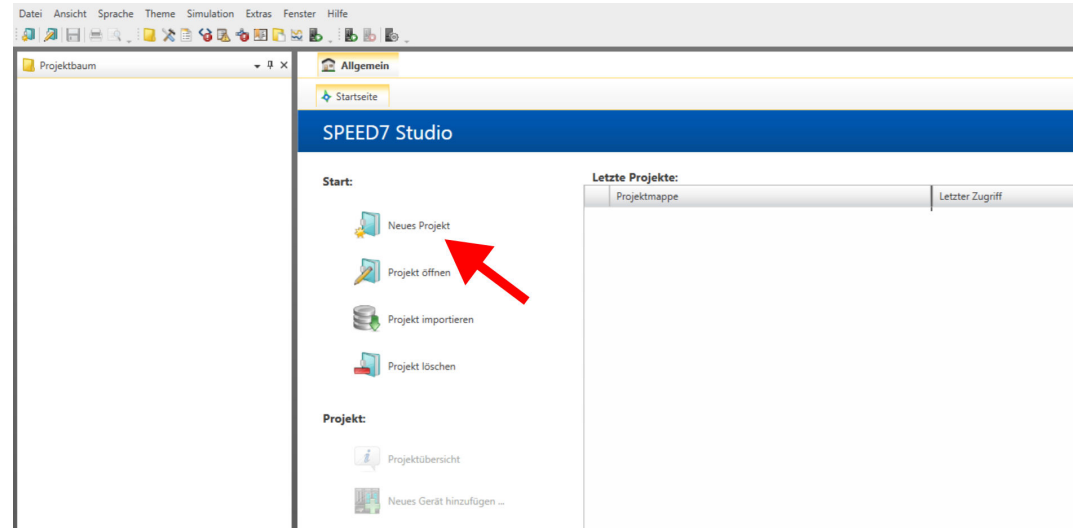
3.3.3 Einsatz im VIPA SPEED7 Studio

3.3.3.1 Hardware-Konfiguration

CPU im Projekt anlegen

Bitte verwenden Sie für die Projektierung das *SPEED7 Studio* ab V1.6.1.

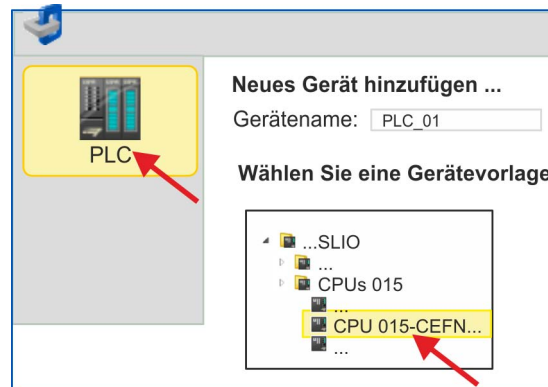
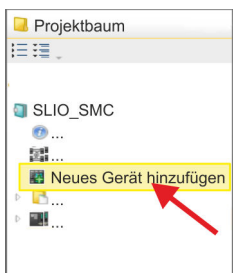
1. Starten Sie das *SPEED7 Studio*.



2. Erstellen sie auf der Startseite mit "Neues Projekt" ein neues Projekt und vergeben Sie einen "Projektnamen".

⇒ Ein neues Projekt wird angelegt und in die Sicht "Geräte und Netze" gewechselt.

3. Klicken Sie im Projektbaum auf "Neues Gerät hinzufügen ...".



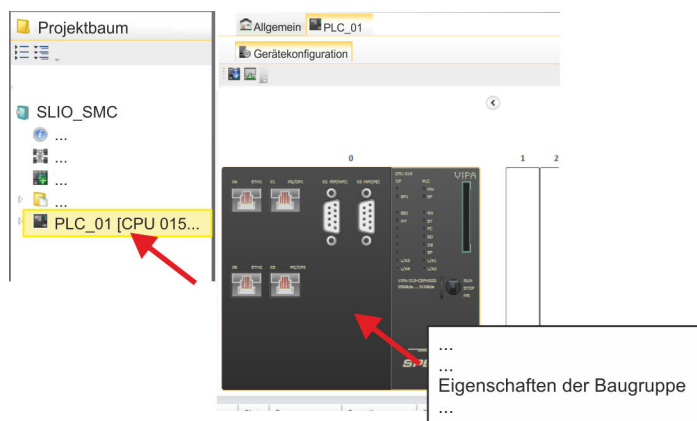
⇒ Es öffnet sich ein Dialog für die Geräteauswahl.

4. Wählen Sie unter den "Gerätevorlagen" eine CPU mit EtherCAT-Master-Funktionalität wie z.B. die CPU 015-CEFN00 und klicken Sie auf [OK].

⇒ Die CPU wird in "Geräte und Netze" eingefügt und die "Gerätekonfiguration" geöffnet.

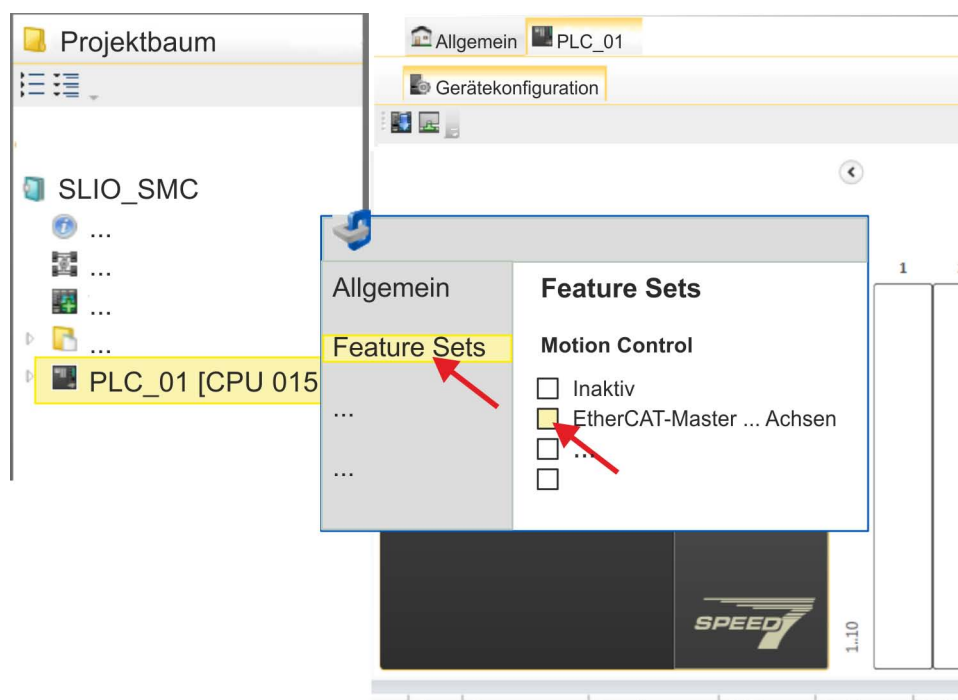
Motion-Control-Funktionen aktivieren

Sofern bei Ihrer CPU die EtherCAT-Master-Funktionalität noch nicht aktiviert ist, erfolgt die Aktivierung nach folgenden Vorgehensweise:



1. Klicken Sie in der "Gerätekonfiguration" auf die CPU und wählen Sie "Kontextmenü → Eigenschaften der Baugruppe".

⇒ Es öffnet sich der Eigenschaften-Dialog der CPU.



2. Klicken Sie auf "Feature Sets" und aktivieren Sie unter "Motion Control" einen der Parameter "EtherCAT-Master ... Achsen". Die Anzahl der Achsen ist in diesem Beispiel nicht relevant.

3. Bestätigen Sie Ihre Angaben mit [OK].

⇒ Die Motion-Control-Funktionen stehen Ihnen nun in Ihrem Projekt zur Verfügung.

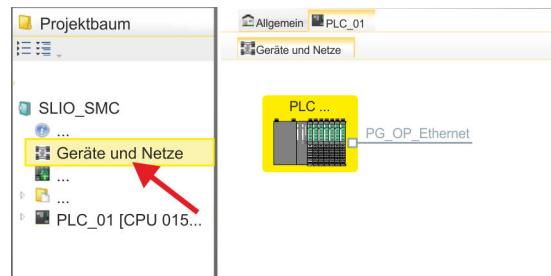


VORSICHT!

Bitte beachten Sie, dass bei jeder Änderung der Feature-Set-Einstellungen systembedingt das EtherCAT-Feldbus-System zusammen mit der Motion-Control-Konfiguration aus Ihrem Projekt gelöscht werden!

Ethernet-PG/OP-Kanal parametrieren

1. ➤ Klicken Sie im *Projektbaum* auf "*Geräte und Netze*".
⇒ Sie erhalten eine grafische Objekt-Ansicht Ihrer CPU.



2. ➤ Klicken Sie auf das Netzwerk "*PG_OP_Ethernet*".
3. ➤ Wählen Sie "*Kontextmenü* ➔ *Eigenschaften der Schnittstelle*".
⇒ Es öffnet sich ein Dialogfenster. Hier können Sie IP-Adressdaten für Ihren Ethernet-PG/OP-Kanal angeben. Gültige IP-Adress-Parameter erhalten Sie von Ihrem Systemadministrator.
4. ➤ Bestätigen Sie Ihre Eingabe mit [OK].
⇒ Die IP-Adressdaten werden in Ihr Projekt übernommen und in "*Geräte und Netze*" unter "*Lokale Baugruppen*" aufgelistet.
Nach der Übertragung Ihres Projekts ist Ihre CPU über die angegebenen IP-Adressdaten via Ethernet-PG/OP-Kanal erreichbar.

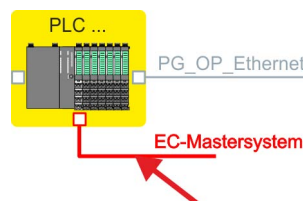
ESI-Datei installieren

Damit der *Sigma-7* EtherCAT Antrieb im *SPEED7 EtherCAT Manager* konfiguriert werden kann, muss die entsprechende ESI-Datei installiert sein. In der Regel wird das *SPEED7 Studio* mit aktuellen ESI-Dateien ausgeliefert und Sie können diesen Teil überspringen. Sollte Ihre ESI-Datei veraltet sein, finden Sie die aktuellste ESI-Datei für den *Sigma-7* EtherCAT Antrieb unter www.yaskawa.eu.com unter "*Service* ➔ *Drives & Motion Software*".

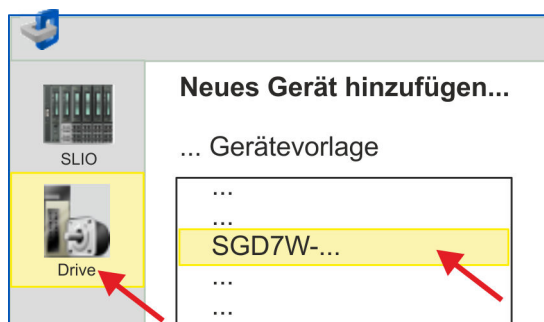
1. ➤ Laden Sie die zu Ihrem Antrieb passende ESI-Datei herunter. Entpacken Sie diese falls erforderlich.
2. ➤ Gehen Sie in Ihr *SPEED7 Studio*.
3. ➤ Öffnen Sie mit "*Extras* ➔ *Gerätebeschreibungsdatei installieren (EtherCAT - ESI)*" das zugehörige Dialogfenster.
4. ➤ Geben Sie unter "*Quellpfad*" die ESI-Datei an und installieren Sie diese mit [Installieren].
⇒ Die Geräte der ESI-Datei steht Ihnen nun zur Verfügung.

Sigma-7W Doppelachs-Antrieb hinzufügen

1. ➤ Klicken Sie im Projektbaum auf "*Geräte und Netze*".
2. ➤ Klicken Sie hier auf "*EC-Mastersystem*" und wählen sie "*Kontextmenü* ➔ *Neues Gerät hinzufügen*".



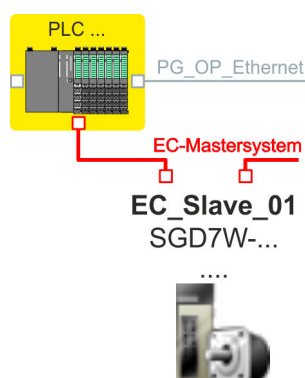
- ⇒ Es öffnet sich die Gerätevorlage zur Auswahl eines EtherCAT-Devices.



3. ➤ Wählen Sie Ihren *Sigma-7W* Doppelachs-Antrieb aus:

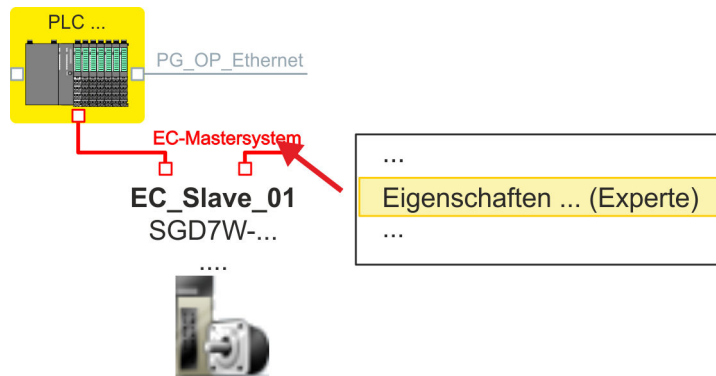
- SGD7W-xxxxA0...

Bestätigen Sie Ihre Angaben mit [OK]. Sollte Ihr Antrieb nicht vorhanden sein, müssen Sie die entsprechende ESI-Datei wie weiter oben beschrieben installieren.



⇒ Der *Sigma-7W* Doppelachs-Antrieb wird an Ihr EC-Mastersystem angebunden.

Sigma-7W Doppelachs-Antrieb konfigurieren



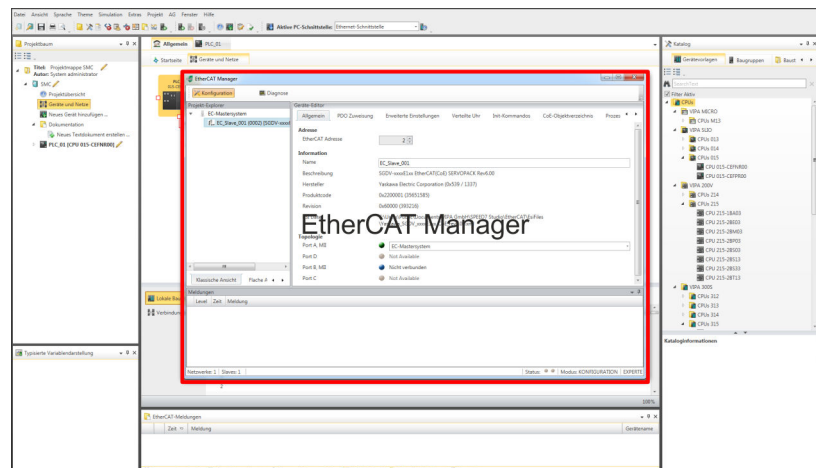
1. Klicken Sie auf "EC-Mastersystem" und wählen sie "Kontextmenü → Eigenschaft des Busystems (Experte)".



PDOs können Sie nur im "Experten-Modus" bearbeiten! Ansonsten werden die Schaltflächen ausgeblendet.

- ⇒ Der **SPEED7 EtherCAT Manager** wird gestartet. Hier können Sie die EtherCAT-Kommunikation zu Ihrem Sigma-7W Doppelachs-Antrieb konfigurieren.

Näheres zum Einsatz des **SPEED7 EtherCAT Manager** finden Sie in der Onlinehilfe zum **SPEED7 Studio**.



2. Klicken Sie im **SPEED7 EtherCAT Manager** auf den Slave und wählen Sie im "Geräte-Editor" den Reiter "PDO-Zuweisung" an.

EtherCAT Manager	
Projekt-Explorer	Geräte-Editor
<ul style="list-style-type: none"> EC-Mastersystem <ul style="list-style-type: none"> EC-Slave_001 <ul style="list-style-type: none"> 001: Module 1 002: Module 2 	<div>PDO Zuweisung ...</div> <div>Eingänge</div> <div> <input type="checkbox"/> Module 1 (SGD7). 1st Transmit PDO mapping </div>

- ⇒ Dieser Dialog zeigt eine Auflistung aller PDOs für "Module 1" (Achse 1) und "Module 2" (Achse 2).

3. → Durch Auswahl des entsprechenden PDO-Mappings können Sie mit [Bearbeiten] die PDOs bearbeiten. Wählen Sie das Mapping "Module 1 (SGD7). 1st Transmit PDO mapping" an und klicken Sie auf [Bearbeiten].



Bitte beachten Sie, dass aufgrund der Voreinstellung manche PDOs nicht bearbeitet werden können. Durch Deaktivierung bereits aktivierter PDOs können Sie die Bearbeitung von gesperrten PDOs frei geben.

Geräte-Editor

PDO Zuweisung ...

Eingänge	Ausgänge
<input type="checkbox"/> Module 1 (SGD7). 1st Transmit PDO mapping	<input type="checkbox"/> Module 1 (SGD7). 1st Receive PDO mapping
<input type="checkbox"/> Module 1 (SGD7). 2nd Transmit PDO mapping	<input type="checkbox"/> Module 1 (SGD7). 2nd Receive PDO mapping
<input type="checkbox"/> Module 2 (SGD7). 1st Transmit PDO mapping	<input type="checkbox"/> Module 2 (SGD7). 1st Receive PDO mapping
<input type="checkbox"/> Module 2 (SGD7). 2nd Transmit PDO mapping	<input type="checkbox"/> Module 2 (SGD7). 2nd Receive PDO mapping

... Bearbeiten ...

- ⇒ Es öffnet sich der Dialog *"PDO bearbeiten"*. Bitte überprüfen Sie hier die aufgeführten PDO-Einstellungen und passen Sie diese ggf. an. Bitte berücksichtigen Sie hierbei auch die Reihenfolge der *"Einträge"* und ergänzen Sie diese entsprechend.

Allgemein

Name: Module 1 (SGD7).1st Transmit PDO

Index: 0x1A00 Dez Hex

Flags:

- ☐ Zwingend
- ☐ Schreibgeschützt
- ☐ Virtuell

Richtung:

- ☒ TxPdo (Eingang)
- ☐ RxPdo (Ausgang)

Optional

Ausschließen:

- ☒ 1A01
- ☐ 1A02
- ☐ 1A03
- ☐ 1A10
- ☐ 1A11
- ☐ 1A12
- ☐ 1A13

Einträge

Name	Index	Bitlänge	Kommentar
Status word	0x6041:00	16	
Position actual internal value	0x6063:00	32	
Position actual value	0x6064:00	32	
Torque actual value	0x6077:00	16	
Following error actual value	0x60F4:00	32	
Modes of operation display	0x6061:00	8	
---	---	8	---
Digital inputs	0x60FD:00	32	

Neu Löschen Bearbeiten Nach oben Nach unten

Für die Bearbeitung der *"Einträge"* stehen folgende Funktionen zur Verfügung:

- Neu
 - Hiermit können Sie in einem Dialogfenster einen neuen Eintrag anlegen, indem Sie aus dem *"CoE-Objektverzeichnis"* den entsprechenden Eintrag auswählen und Ihre Einstellungen vornehmen. Mit [OK] wird der Eintrag übernommen und in der Liste der Einträge aufgeführt.
- Löschen
 - Hiermit können Sie den angewählte Eintrag löschen.
- Bearbeiten
 - Hiermit können Sie allgemeinen Daten eines Eintrags bearbeiten.
- Nach oben/unten
 - Hiermit können Sie den angewählten Eintrag in der Liste nach oben bzw. nach unten bewegen.

4. ➔ Führen Sie für die Transmit PDOs folgende Einstellungen durch:

Eingänge: 1st Transmit PDO

Module 1 (SGD7). 1st Transmit PDO mapping	Module 2 (SGD7). 1st Transmit PDO mapping
Name: Module 1 (SGD7). 1st Transmit PDO mapping	Name: Module 2 (SGD7). 1st Transmit PDO mapping
Index: 0x1A00	Index: 0x1A10
Flags: Alles deaktiviert	
Richtung: TxPdo (Eingang): aktiviert	
Ausschließen: 1A01: deaktiviert	1A11: deaktiviert
Bitte diese Einstellungen beachten, da ansonsten die PDO-Mappings nicht zeitgleich aktiviert werden können!	

Einträge	Modul 1 (Achse 1)	Modul 2 (Achse 2)	Bitlänge
Name	Index	Index	
Status word	0x6041:00	0x6841:00	16Bit
Position actual internal value	0x6063:00	0x6863:00	32Bit
Position actual value	0x6064:00	0x6864:00	32Bit
Torque actual value	0x6077:00	0x6877:00	16Bit
Following error actual value	0x60F4:00	0x68F4:00	32Bit
Modes of operation display	0x6061:00	0x6861:00	8Bit
---	---	---	8Bit
Digital inputs	0x60FD:00	0x68FD:00	32Bit

Eingänge: 2nd Transmit PDO

Module 1 (SGD7). 2nd Transmit PDO mapping	Module 2 (SGD7). 2st Transmit PDO mapping
Name: Module 1 (SGD7). 2nd Transmit PDO mapping	Name: Module 2 (SGD7). 2st Transmit PDO mapping
Index: 0x1A01	Index: 0x1A11
Flags: Alles deaktiviert	
Richtung: TxPdo (Eingang): aktiviert	
Ausschließen: 1A00, 1A02, 1A03: deaktiviert	1A10, 1A12, 1A13: deaktiviert
Bitte diese Einstellungen beachten, da ansonsten die PDO-Mappings nicht zeitgleich aktiviert werden können!	

Einträge	Modul 1 (Achse 1)	Modul 2 (Achse 2)	Bitlänge
Name	Index	Index	
Touch probe status	0x60B9:00	0x68B9:00	16Bit
Touch probe 1 position value	0x60BA:00	0x68BA:00	32Bit
Touch probe 2 position value	0x60BC:00	0x68BC:00	32Bit
Velocity actual value	0x606C:00	0x686C:00	32Bit

5. ➔ Führen Sie für die Receive PDOs folgende Einstellungen durch:

Ausgänge: 1st Receive PDO

Module 1 (SGD7). 1st Receive PDO	Module 2 (SGD7). 1st Receive PDO
Name: Module 1 (SGD7). 1st Receive PDO mapping	Name: Module 2 (SGD7). 1st Receive PDO mapping
Index: 0x1600	Index: 0x1610
Flags: Alles deaktiviert	
Richtung: RxPdo (Ausgang): aktiviert	
Ausschließen: 1601, 1602, 1603: deaktiviert	1611, 1612, 1613: deaktiviert
Bitte diese Einstellungen beachten, da ansonsten die PDO-Mappings nicht zeitgleich aktiviert werden können!	

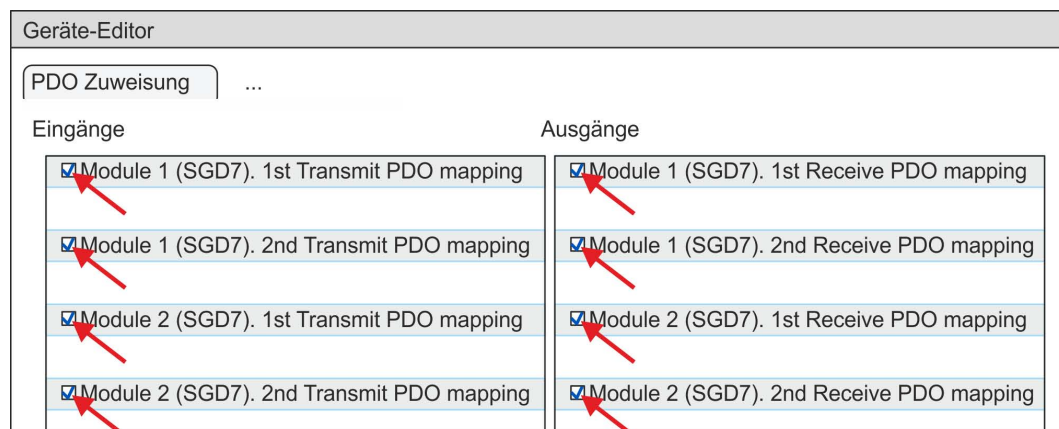
Einträge	Modul 1 (Achse 1)	Modul 2 (Achse 2)	Bitlänge
Name	Index	Index	
Control word	0x6040:00	0x6840:00	16Bit
Target position	0x607A:00	0x687A:00	32Bit
Target velocity	0x60FF:00	0x68FF:00	32Bit
Modes of operation	0x6060:00	0x6860:00	8Bit
---	---	---	8Bit
Touch probe function	0x60B8:00	0x68B8:00	16Bit

Ausgänge: 2nd Receive PDO

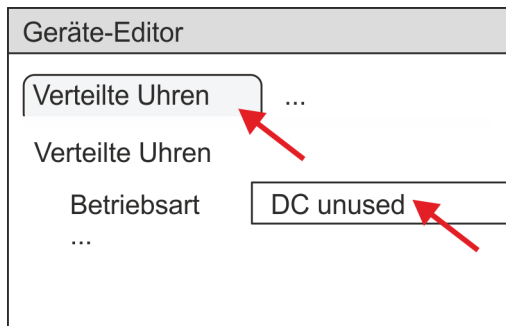
Module 1 (SGD7). 2nd Receive PDO	Module 2 (SGD7). 2nd Receive PDO
Name: Module 1 (SGD7). 2nd Receive PDO mapping	Name: Module 2 (SGD7). 2nd Receive PDO mapping
Index: 0x1601	Index: 0x1611
Flags: Alles deaktiviert	
Richtung: RxPdo (Ausgang): aktiviert	
Ausschließen: 1600, 1602, 1603: deaktiviert	1610, 1612, 1613: deaktiviert
Bitte diese Einstellungen beachten, da ansonsten die PDO-Mappings nicht zeitgleich aktiviert werden können!	

Einträge	Modul 1 (Achse 1)	Modul 2 (Achse 2)	Bitlänge
Name	Index	Index	
Profile velocity	0x6081:00	0x6881:00	32Bit
Profile acceleration	0x6083:00	0x6883:00	32Bit
Profile deceleration	0x6084:00	0x6884:00	32Bit

6. ➔ Aktivieren Sie für "Module 1" und "Module 2" in PDO-Zuweisung die PDOs 1 und 2 für die Ein- und Ausgänge. Alle nachfolgenden PDOs müssen deaktiviert bleiben. Sollte dies nicht möglich sein, überprüfen Sie bitte den jeweiligen PDO-Parameter "Ausschließen".

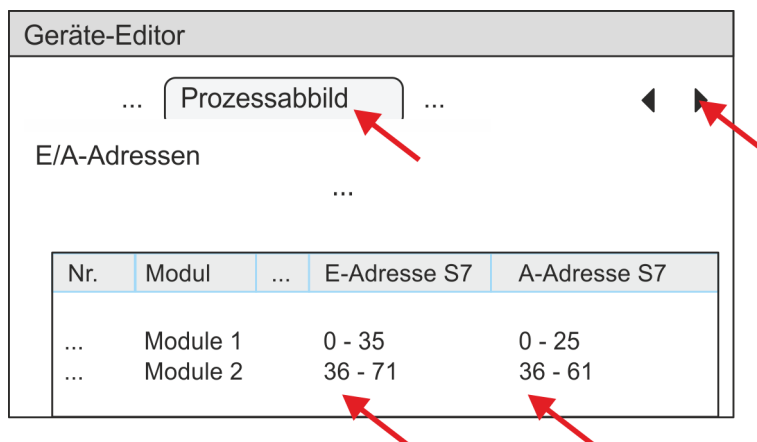


7. ➔ Wählen Sie im "Geräte-Editor" des SPEED7 EtherCAT Manager den Reiter "Verteilte Uhren" an und stellen Sie "DC unused" als "Betriebsart" ein.

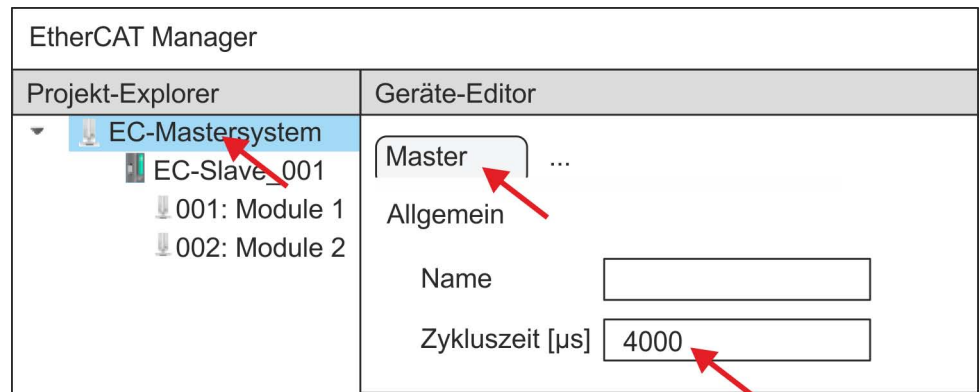


8. ➔ Wählen Sie im "Geräte-Editor" über die Pfeiltaste den Reiter "Prozessabbild" an und notieren Sie sich für die Parameter des Bausteins FB 874 - VMC_InitSigma7W_EC folgende PDO-Anfangsadressen:

- Module 1: "E-Adresse S7" → "M1_PdoInputs" (hier 0)
- Module 2: "E-Adresse S7" → "M2_PdoInputs" (hier 36)
- Module 1: "A-Adresse S7" → "M1_PdoOutputs" (hier 0)
- Module 2: "A-Adresse S7" → "M2_PdoOutputs" (hier 36)



9. ➔ Klicken Sie im *SPEED7 EtherCAT Manager* auf "EC-Mastersystem" und wählen Sie im "Geräte-Editor" den Reiter "Master" an.

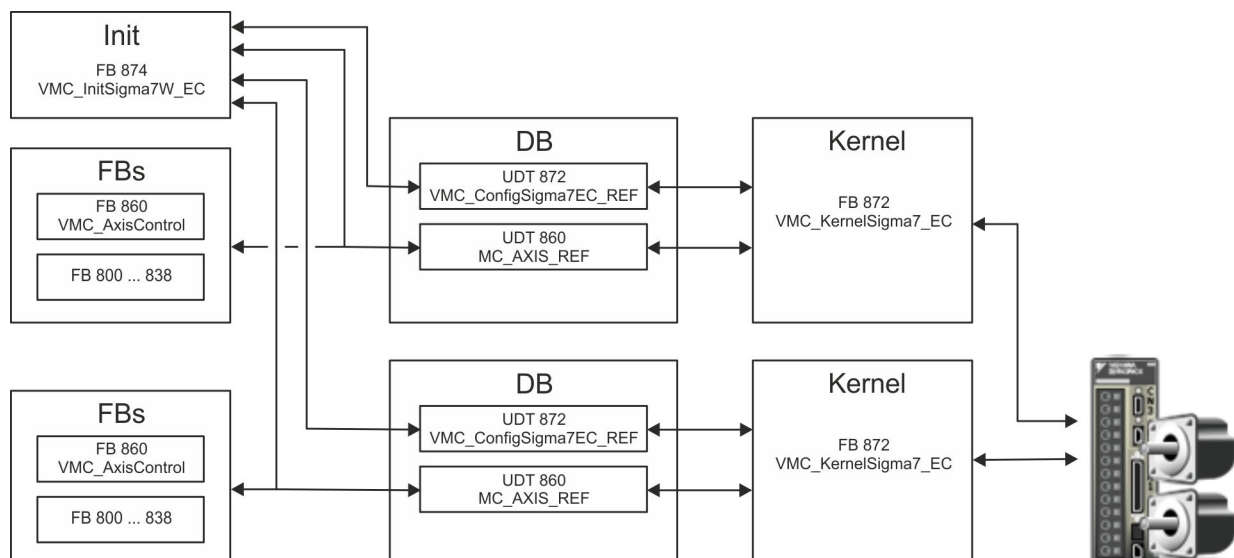


⇒ Stellen Sie für Sigma-7W (400V) Antriebe eine Zykluszeit von mindestens 4ms ein.

10. ➔ Indem Sie den Dialog des *SPEED7 EtherCAT Manager* mit [X] schließen, wird die Konfiguration in das *SPEED7 Studio* übernommen.

3.3.3.2 Anwender-Programm

3.3.3.2.1 Programmstruktur



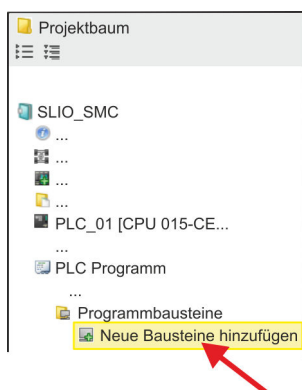
- DB

Für jede Achse ist ein Datenbaustein (Achs-DB) für Konfiguration und Statusdaten anzulegen. Der Datenbaustein besteht aus folgenden Datenstrukturen:

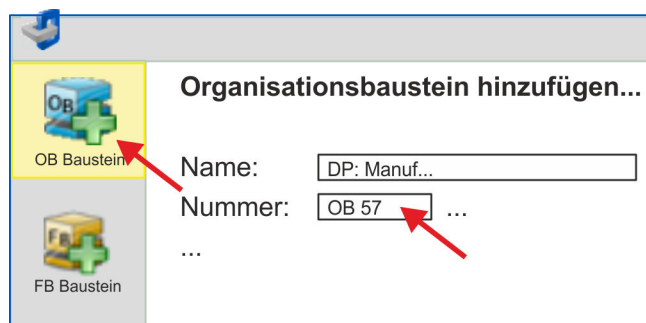
 - UDT 872 - *VMC_ConfigSigma7EC_REF*
Die Datenstruktur beschreibt den Aufbau der Konfiguration des Antriebs. Spezifische Datenstruktur für *Sigma-7* EtherCAT.
 - UDT 860 - *MC_AXIS_REF*
Die Datenstruktur beschreibt den Aufbau der Parameter und Statusinformationen von Antrieben. Allgemeine Datenstruktur für alle Antriebe und Bussysteme.
- FB 874 - *VMC_InitSigma7W_EC*
 - Der *Init*-Baustein dient zur Konfiguration des Doppelachs-Antriebs.
 - Spezifischer Baustein für *Sigma-7W* EtherCAT.
 - Die Konfigurationsdaten für die Initialisierung sind im *Achs-DB* abzulegen.
- FB 872 - *VMC_KernelSigma7_EC*
 - Der *Kernel*-Baustein kommuniziert mit dem Antrieb über das entsprechende Bussystem, verarbeitet die Benutzeraufträge und liefert Statusmeldungen zurück.
 - Je Achse ist der FB 872 - *VMC_KernelSigma7_EC* aufzurufen.
 - Spezifischer Baustein für *Sigma-7* EtherCAT.
 - Der Austausch der Daten erfolgt mittels des *Achs-DB*.
- FB 860 - *VMC_AxisControl*
 - Universal-Baustein für alle Antriebe und Bussysteme.
 - Je Achse ist der FB 860 - *VMC_AxisControl* aufzurufen.
 - Unterstützt einfache Bewegungskommandos und liefert alle relevanten Statusmeldungen.
 - Der Austausch der Daten erfolgt mittels des *Achs-DB*.
 - Über die Instanzdaten des Bausteins können Sie zur Bewegungssteuerung und Statusabfrage eine Visualisierung anbinden.
 - Zusätzlich zum FB 860 - *VMC_AxisControl* haben Sie die Möglichkeit *PLCopen*-Bausteine zu nutzen.
- FB 800 ... FB 838 - *PLCopen*
 - Die *PLCopen*-Bausteine dienen zur Programmierung von Bewegungsabläufen und Statusabfragen.
 - Je Achse sind die *PLCopen*-Bausteine aufzurufen.

3.3.3.2.2 Programmierung

Bausteine in Projekt kopieren

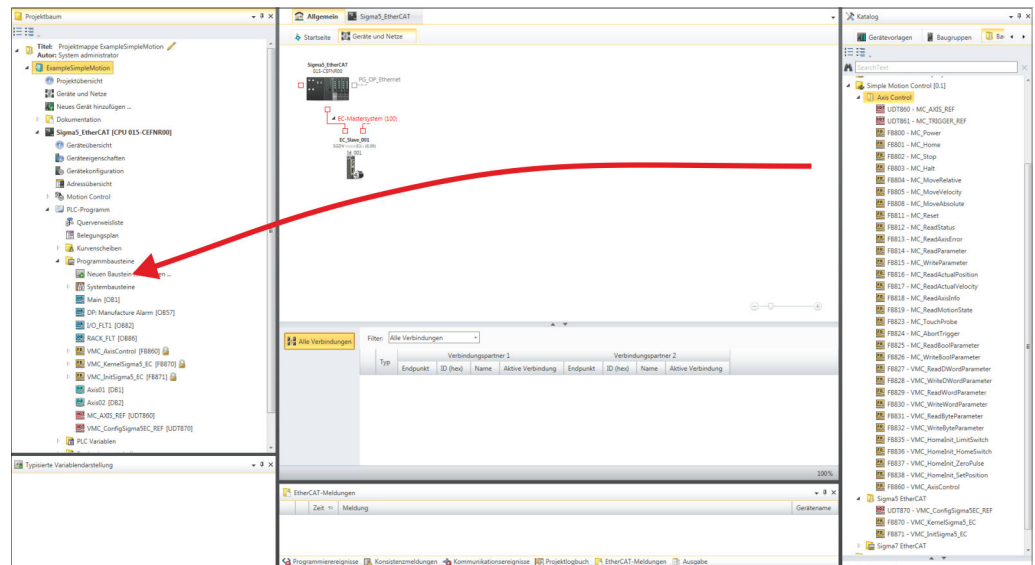


1. Klicken Sie im *Projektbaum* innerhalb der CPU unter "*PLC-Programm*", "*Programmbausteine*" auf "*Neuen Baustein hinzufügen*".



⇒ Das Dialogfenster "*Baustein hinzufügen*" öffnet sich.

2. ➤ Wählen Sie den Bausteintyp **"OB Baustein"** und fügen Sie nacheinander OB 57, OB 82 und OB 86 Ihrem Projekt hinzu.



3. ➤ Öffnen Sie im **"Katalog"** unter **"Bausteine"** **"Simple Motion Control"** und ziehen Sie per Drag&Drop folgende Bausteine in **"Programmbausteine"** des **Projektbaums**:

- **Sigma-7 EtherCAT:**
 - UDT 872 - VMC_ConfigSigma7EC_REF
 - FB 872 - VMC_KernelSigma7_EC
 - FB 874 - VMC_InitSigma7W_EC
- **Axis Control**
 - UDT 860 - MC_AXIS_REF
 - Bausteine für die gewünschten Bewegungsabläufe

Achs-DB für **"Module 1"** anlegen

1. ➤ Fügen Sie Ihrem Projekt einen neuen DB als **Achs-DB** hinzu. Klicken Sie hierzu im **Projektbaum** innerhalb der CPU unter **"PLC-Programm"**, **"Programmbausteine"** auf **"Neuen Baustein hinzufügen"**, wählen Sie den Bausteintyp **"DB Baustein"** und vergeben Sie diesem den Namen **"Axis01"**. Die DB-Nr. können Sie frei wählen wie z.B. DB 10.

⇒ Der Baustein wird angelegt und geöffnet.

2. ➤ ■ Legen Sie in **"Axis01"** die Variable **"Config"** vom Typ UDT 872 an. Dies sind spezifische Achs-Konfigurationsdaten.
- Legen Sie in **"Axis01"** die Variable **"Axis"** vom Typ UDT 860 an. Während des Betriebs werden hier alle Betriebsdaten der Achse abgelegt.

Axis01 [DB10]
Bausteinstruktur

	Adr...	Name	Datentyp	...
	...	Config	UDT	[872]
	...	Axis	UDT	[860]

Achs-DB für **"Module 2"** anlegen

1. ➤ Fügen Sie Ihrem Projekt einen weiteren DB als **Achs-DB** hinzu und vergeben Sie diesem den Namen **"Axis02"**. Die DB-Nr. können Sie frei wählen wie z.B. DB 11.

⇒ Der Baustein wird angelegt und geöffnet.

2. ➔
- Legen Sie in "Axis02" die Variable "Config" vom Typ UDT 872 an. Dies sind spezifische Achs-Konfigurationsdaten.
 - Legen Sie in "Axis02" die Variable "Axis" vom Typ UDT 860 an. Während des Betriebs werden hier alle Betriebsdaten der Achse abgelegt.

Axis02 [DB11]

Bausteinstruktur

	Adr...	Name	Datentyp	...
	...	Config	UDT	[872]
	...	Axis	UDT	[860]

OB 1

Konfiguration der Doppel-
achse

Öffnen Sie den OB 1 und programmieren Sie folgende FB-Aufrufe mit zugehörigen DBs:

→ FB 874 - VMC_InitSigma7W_EC, DB 874 ↪ Kapitel 3.3.5.3 "FB 874 - VMC_InitSigma7W_EC - Sigma-7W EtherCAT Initialisierung" auf Seite 126

Geben Sie unter *M1/M2_PdoInputs* bzw. *M1/M2_PdoOutputs* die Adresse aus dem *SPEED7 EtherCAT Manager* für die entsprechende Achse an. ↪ 98

```
⇒ CALL "VMC_InitSigma7W_EC" , "DI_InitSgm7WETC01"
   Enable                                     :=TRUE
   LogicalAddress                             :=0
   M1_PdoInputs                               :=0 (EtherCAT-Manager
                                                Module1: E-Adresse S7)

   M1_PdoOutputs                             :=0 (EtherCAT-Manager
                                                Module1: A-Adresse S7)

   M1_EncoderType                             :=2
   M1_EncoderResolutionBits                   :=20
   M1_FactorPosition                         :=1.048576e+006
   M1_FactorVelocity                         :=1.048576e+006
   M1_FactorAcceleration                     :=1.048576e+002
   M1_OffsetPosition                         :=0.000000e+000
   M1_MaxVelocityApp                         :=5.000000e+001
   M1_MaxAccelerationApp                     :=1.000000e+002
   M1_MaxDecelerationApp                     :=1.000000e+002
   M1_MaxVelocityDrive                       :=6.000000e+001
   M1_MaxAccelerationDrive                   :=1.500000e+002
   M1_MaxDecelerationDrive                   :=1.500000e+002
   M1_MaxPosition                           :=1.048500e+003
   M1_MinPosition                           :=-1.048514e+003
   M1_EnableMaxPosition                       :=TRUE
   M1_EnableMinPosition                       :=TRUE
   M2_PdoInputs                               :=36 (EtherCAT-Manager
                                                Module2: E-Adresse S7)

   M2_PdoOutputs                             :=36 (EtherCAT-Manager
                                                Module2: A-Adresse S7)

   M2_EncoderType                             :=2
   M2_EncoderResolutionBits                   :=20
   M2_FactorPosition                         :=1.048576e+006
   M2_FactorVelocity                         :=1.048576e+006
   M2_FactorAcceleration                     :=1.048576e+002
   M2_OffsetPosition                         :=0.000000e+000
   M2_MaxVelocityApp                         :=5.000000e+001
   M2_MaxAccelerationApp                     :=1.000000e+002
   M2_MaxDecelerationApp                     :=1.000000e+002
   M2_MaxVelocityDrive                       :=6.000000e+001
   M2_MaxAccelerationDrive                   :=1.500000e+002
   M2_MaxDecelerationDrive                   :=1.500000e+002
   M2_MaxPosition                           :=1.048500e+003
   M2_MinPosition                           :=-1.048514e+003
   M2_EnableMaxPosition                       :=TRUE
   M2_EnableMinPosition                       :=TRUE
   M1_MinUserPosition                         :=-1000.0
   M1_MaxUserPosition                         :=1000.0
   M2_MinUserPosition                         :=-1000.0
   M2_MaxUserPosition                         :=1000.0
   Valid                                       :="InitS7WEC1_Valid"
   Error                                       :="InitS7WEC1_Error"
```

```

ErrorID                := "InitS7WEC1_ErrorID"
M1_Config               := "Axis01".Config
M1_Axis                 := "Axis01".Axis
M2_Config               := "Axis02".Config
M2_Axis                 := "Axis02".Axis

```

Kernel für die jeweilige Achse beschalten

Der *Kernel* verarbeitet die Benutzerkommandos und gibt sie entsprechend aufbereitet an den Antrieb über das jeweilige Bussystem weiter.

➔ FB 872 - VMC_KernelSigma7_EC, DB 872 für Achse 1

FB 872 - VMC_KernelSigma7_EC, DB 1872 für Achse 2 ➔ *Kapitel 3.2.5.2 "FB 872 - VMC_KernelSigma7_EC - Sigma-7 EtherCAT Kernel" auf Seite 84*

```

⇒ CALL "VMC_KernelSigma7_EC" , DB 872
   Init := "KernelS7WEC1_Init"
   Config := "Axis01".Config
   Axis := "Axis01".Axis

CALL "VMC_KernelSigma7_EC" , DB 1872
   Init := "KernelS7WEC2_Init"
   Config := "Axis02".Config
   Axis := "Axis02".Axis

```

Baustein für Bewegungsabläufe beschalten

Zur Vereinfachung soll hier die Beschaltung des FB 860 - VMC_AxisControl gezeigt werden. Dieser Universalbaustein unterstützt einfache Bewegungskommandos und liefert Statusmeldungen zurück. Die Ein- und Ausgänge können Sie individuell beschalten. Bitte geben Sie unter "Axis" die Referenz zu den entsprechenden Achsdaten im *Achs-DB* an.

➔ FB 860 - VMC_AxisControl, DB 860 ↗ *Kapitel 3.4.2.2 "FB 860 - VMC_AxisControl - Control-Baustein Achskontrolle" auf Seite 132*

```
⇒ CALL "VMC_AxisControl" , "DI_AxisControl01"
   AxisEnable      := "AxCtrl1_AxisEnable"
   AxisReset       := "AxCtrl1_AxisReset"
   HomeExecute     := "AxCtrl1_HomeExecute"
   HomePosition    := "AxCtrl1_HomePosition"
   StopExecute     := "AxCtrl1_StopExecute"
   MvVelocityExecute := "AxCtrl1_MvVelExecute"
   MvRelativeExecute := "AxCtrl1_MvRelExecute"
   MvAbsoluteExecute := "AxCtrl1_MvAbsExecute"
   PositionDistance := "AxCtrl1_PositionDistance"
   Velocity        := "AxCtrl1_Velocity"
   Acceleration     := "AxCtrl1_Acceleration"
   Deceleration     := "AxCtrl1_Deceleration"
   JogPositive      := "AxCtrl1_JogPositive"
   JogNegative      := "AxCtrl1_JogNegative"
   JogVelocity      := "AxCtrl1_JogVelocity"
   JogAcceleration  := "AxCtrl1_JogAcceleration"
   JogDeceleration  := "AxCtrl1_JogDeceleration"
   AxisReady       := "AxCtrl1_AxisReady"
   AxisEnabled      := "AxCtrl1_AxisEnabled"
   AxisError        := "AxCtrl1_AxisError"
   AxisErrorID      := "AxCtrl1_AxisErrorID"
   DriveWarning     := "AxCtrl1_DriveWarning"
   DriveError       := "AxCtrl1_DriveError"
   DriveErrorID     := "AxCtrl1_DriveErrorID"
   IsHomed          := "AxCtrl1_IsHomed"
   ModeOfOperation  := "AxCtrl1_ModeOfOperation"
   PLCOpenState     := "AxCtrl1_PLCOpenState"
   ActualPosition    := "AxCtrl1_ActualPosition"
   ActualVelocity    := "AxCtrl1_ActualVelocity"
   CmdDone          := "AxCtrl1_CmdDone"
   CmdBusy          := "AxCtrl1_CmdBusy"
   CmdAborted       := "AxCtrl1_CmdAborted"
   CmdError         := "AxCtrl1_CmdError"
   CmdErrorID       := "AxCtrl1_CmdErrorID"
   DirectionPositive := "AxCtrl1_DirectionPos"
   DirectionNegative := "AxCtrl1_DirectionNeg"
   SWLimitMinActive := "AxCtrl1_SWLimitMinActive"
   SWLimitMaxActive := "AxCtrl1_SWLimitMaxActive"
   HWLimitMinActive := "AxCtrl1_HWLimitMinActive"
   HWLimitMaxActive := "AxCtrl1_HWLimitMaxActive"
   Axis             := "Axis..."_Axis
```

Geben Sie unter *Axis* für die Achse 1 "Axis01" und für die Achse 2 "Axis02" an.



Für komplexe Bewegungsaufgaben können Sie die PLCOpen-Bausteine verwenden. Hier müssen Sie ebenfalls unter Axis die Referenz zu den Achsdaten im Achs-DB angeben.

Ihr Projekt beinhaltet nun folgende Bausteine:

- OB 1 - Main
- OB 57 - DP Manufacturer Alarm
- OB 82 - I/O_FLT1
- OB 86 - Rack_FLT

- FB 860 - VMC_AxisControl mit Instanz-DB
- FB 872 - VMC_KernelSigma7_EC mit Instanz-DB
- FB 874 - VMC_InitSigma7W_EC mit Instanz-DB
- UDT 860 - MC_Axis_REF
- UDT 872 - VMC_ConfigSigma7EC_REF

Zeitlicher Ablauf

1. ➤ Wählen Sie *"Projekt ➔ Alles übersetzen"* und übertragen Sie das Projekt in Ihre CPU. Näheres zur Übertragung Ihres Projekt finden Sie in der Onlinehilfe zum *SPEED7 Studio*.
⇒ Sie können jetzt Ihre Applikation in Betrieb nehmen.



VORSICHT!

Bitte beachten Sie immer die Sicherheitshinweise zu ihrem Antrieb, insbesondere bei der Inbetriebnahme!

2. ➤ Bevor der Doppelachs-Antrieb gesteuert werden kann, muss diese initialisiert werden. Rufen Sie hierzu den *Init*-Baustein FB 874 - VMC_InitSigma7W_EC mit *Enable* = TRUE auf.
⇒ Der Ausgang *Valid* meldet TRUE zurück. Im Fehlerfall können Sie durch Auswertung der *ErrorID* den Fehler ermitteln.
Den *Init*-Baustein müssen Sie erneut aufrufen, wenn Sie einen neuen Achs-DB laden oder Parameter am *Init*-Baustein geändert wurden.



Fahren Sie erst fort, wenn der Init-Baustein keinen Fehler meldet!

3. ➤ Stellen Sie sicher, dass für jede Achse der *Kernel*-Baustein FB 872 - VMC_KernelSigma7_EC zyklisch aufgerufen wird. Auf diese Weise werden Steuersignale an den Antrieb übergeben und Statusmeldungen übermittelt.
4. ➤ Programmieren Sie für jede Achse Ihre Applikation mit dem FB 860 - VMC_AxisControl oder mit den PLCopen Bausteinen auf.

Steuerung des Antriebs über HMI

Sie haben die Möglichkeit über ein HMI Ihren Antrieb zu steuern. Hierzu gibt es für Movicon eine vorgefertigte Symbolbibliothek für den Zugriff auf den VMC_AxisControl Funktionsbaustein. ➔ *Kapitel 7 "Antrieb über HMI steuern" auf Seite 313*

3.3.4 Einsatz im Siemens SIMATIC Manager

3.3.4.1 Voraussetzung

Übersicht

- Bitte verwenden Sie für die Projektierung den Siemens SIMATIC Manager ab V 5.5 SP2.
- Die Projektierung der System SLIO CPU erfolgt im Siemens SIMATIC Manager in Form des virtuellen PROFINET IO Devices *"VIPA SLIO CPU"*. Das *"VIPA SLIO System"* ist mittels GSDML im Hardware-Katalog zu installieren.
- Die Projektierung des EtherCAT-Masters erfolgt im Siemens SIMATIC Manager in Form des virtuellen PROFINET IO Devices *"EtherCAT-Netzwerk"*. Das *"EtherCAT-Netzwerk"* ist mittels GSDML im Hardware-Katalog zu installieren.

- Das "*EtherCAT-Netzwerk*" kann mit dem VIPA-Tool *SPEED7 EtherCAT Manager* konfiguriert werden.
- Für die Projektierung des Antriebs im *SPEED7 EtherCAT Manager* ist die Installation der zugehörigen ESI-Datei erforderlich.

IO Device "*VIPA SLIO System*" installieren

Die Installation des PROFINET IO Devices "*VIPA SLIO CPU*" im Hardware-Katalog erfolgt nach folgender Vorgehensweise:

1. ➤ Gehen Sie in den Service-Bereich von www.vipa.com.
 2. ➤ Laden Sie aus dem Downloadbereich unter "*Config Dateien* ➔ *PROFINET*" die Konfigurationsdatei für Ihre CPU.
 3. ➤ Extrahieren Sie die Datei in Ihr Arbeitsverzeichnis.
 4. ➤ Starten Sie den Hardware-Konfigurator von Siemens.
 5. ➤ Schließen Sie alle Projekte.
 6. ➤ Gehen Sie auf "*Extras* ➔ *GSD-Dateien installieren*".
 7. ➤ Navigieren Sie in Ihr Arbeitsverzeichnis und installieren Sie die entsprechende GSDML-Datei.
- ⇒ Nach der Installation finden Sie das entsprechende PROFINET IO Device unter "*PROFINET IO* ➔ *Weitere Feldgeräte* ➔ *I/O* ➔ *VIPA SLIO System*".

IO Device EtherCAT-Netzwerk installieren

Die Installation des PROFINET IO Devices "*EtherCAT-Netzwerk*" im Hardware-Katalog erfolgt nach folgender Vorgehensweise:

1. ➤ Gehen Sie in den Service-Bereich von www.vipa.com
 2. ➤ Laden Sie aus dem Downloadbereich unter "*Config Dateien* ➔ *EtherCAT*" die GSDML-Datei für Ihren EtherCAT-Master.
 3. ➤ Extrahieren Sie die Dateien in Ihr Arbeitsverzeichnis.
 4. ➤ Starten Sie den Hardware-Konfigurator von Siemens.
 5. ➤ Schließen Sie alle Projekte.
 6. ➤ Gehen Sie auf "*Extras* ➔ *GSD-Dateien installieren*".
 7. ➤ Navigieren Sie in Ihr Arbeitsverzeichnis und installieren Sie die entsprechende GSDML-Datei.
- ⇒ Nach der Installation finden Sie das "*EtherCAT-Netzwerk*" unter "*PROFINET IO* ➔ *Weitere Feldgeräte* ➔ *I/O* ➔ *VIPA EtherCAT System*".

SPEED7 EtherCAT Manager installieren

Die Konfiguration des PROFINET IO Devices "*EtherCAT-Netzwerk*" erfolgt mit dem *SPEED7 EtherCAT Manager* von VIPA. Sie finden diesen Im Servicebereich von www.vipa.com unter "*Service/Support* ➔ *Downloads* ➔ *SPEED7*".

Die Installation erfolgt nach folgender Vorgehensweise:

1. ➤ Schließen Sie den Siemens SIMATIC Manager.
2. ➤ Gehen Sie in den Service-Bereich von www.vipa.com
3. ➤ Laden Sie den *SPEED7 EtherCAT Manager* und entpacken Sie diesen auf Ihren PC.
4. ➤ Zur Installation starten Sie die Datei *EtherCATManager_v... .exe*.
5. ➤ Wählen Sie die Sprache für die Installation aus.
6. ➤ Stimmen Sie dem Lizenzvertrag zu.
7. ➤ Wählen Sie das Installationsverzeichnis und starten Sie die Installation.

8. ➤ Nach der Installation müssen Sie Ihren PC neu starten
 ⇒ Der *SPEED7 EtherCAT Manager* ist installiert und kann jetzt über das Kontextmenü des Siemens SIMATIC Manager aufgerufen werden.

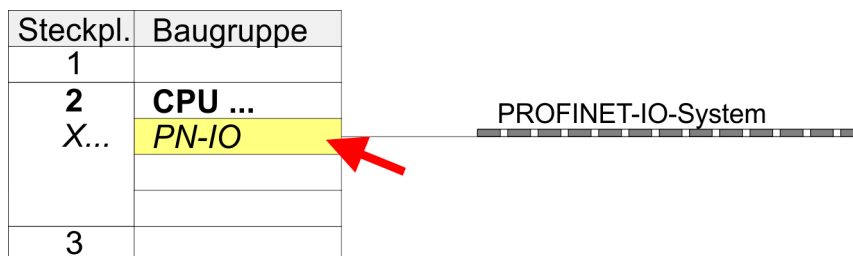
3.3.4.2 Hardware-Konfiguration

CPU im Projekt anlegen

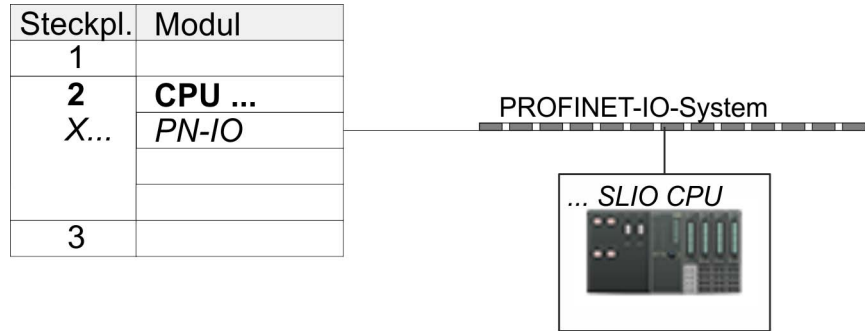
Steckp..	Baugruppe
1	
2	CPU 315-2 PN/DP
X1	MPI/DP
X2	PN-IO
X2...	Port 1
X2...	Port 2
3	

Um kompatibel mit dem Siemens SIMATIC Manager zu sein, sind folgende Schritte durchzuführen:

1. ➤ Starten Sie den Hardware-Konfigurator von Siemens mit einem neuen Projekt.
2. ➤ Fügen Sie aus dem Hardware-Katalog eine Profilschiene ein.
3. ➤ Platzieren Sie auf "Slot"-Nummer 2 die CPU 315-2 PN/DP (6ES7 315-2EH14 V3.2).
4. ➤ Über das Submodul "X1 MPI/DP" projektieren und vernetzen Sie den integrierten PROFIBUS-DP-Master (Buchse X3).
5. ➤ Über das Submodul "X2 PN-IO" projektieren Sie den EtherCAT-Master als virtuelles PROFINET-Netzwerk.
6. ➤ Klicken Sie auf das Submodul "PN-IO" der CPU.
7. ➤ Wählen Sie "Kontextmenü → PROFINET IO-System einfügen".



8. ➤ Legen Sie mit [Neu] ein neues Subnetz an und vergeben Sie gültige IP-Adress-Daten
9. ➤ Klicken Sie auf das Submodul "PN-IO" der CPU und öffnen Sie mit "Kontextmenü → Objekteigenschaften" den Eigenschafts-Dialog.
10. ➤ Geben Sie unter "Allgemein" einen "Gerätenamen" an. Der Geräte name muss eindeutig am Ethernet-Subnetz sein.



Steckpl.	Baugruppe	Bestellnummer	
0	... SLIO CPU ...	015-...	
X2	015-...		
1			
2			
3			
...			

11. Navigieren Sie im Hardware-Katalog in das Verzeichnis *"PROFINET IO → Weitere Feldgeräte → I/O → VIPA SLIO System"* und binden Sie das IO-Device *"015-CEFNR00 CPU"* an Ihr PROFINET-System an.

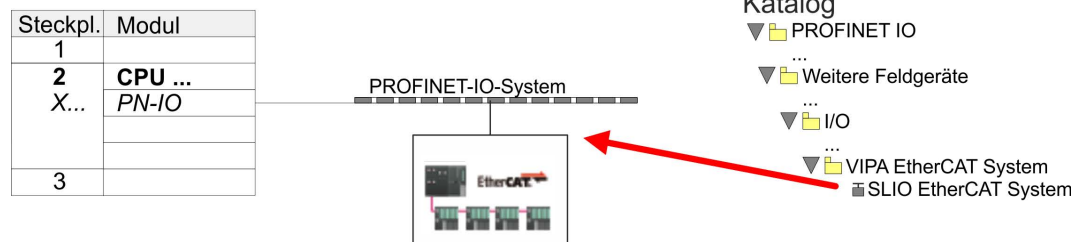
⇒ In der Steckplatzübersicht des PROFINET-IO-Device *"VIPA SLIO CPU"* ist auf Steckplatz 0 die CPU bereits vorplatziert. Ab Steckplatz 1 können Sie Ihre System SLIO Module platzieren.

Ethernet-PG/OP-Kanal parametrieren

Steckpl.	Modul
1	
2	CPU ...
X...	PN-IO
3	
4	343-1EX30
5	
...	

- Platzieren Sie für den Ethernet-PG/OP-Kanal auf Steckplatz 4 den Siemens CP 343-1 (SIMATIC 300 \ CP 300 \ Industrial Ethernet \ CP 343-1 \ 6GK7 343-1EX30 0XE0 V3.0).
- Öffnen Sie durch Doppelklick auf den CP 343-1EX30 den Eigenschaften-Dialog und geben Sie für den CP unter *"Eigenschaften"* IP-Adress-Daten an. Gültige IP-Adress-Parameter erhalten Sie von Ihrem Systemadministrator.
- Ordnen Sie den CP einem *"Subnetz"* zu. Ohne Zuordnung werden die IP-Adress-Daten nicht übernommen!

"EtherCAT-Netzwerk" einfügen



1. Navigieren Sie im Hardware-Katalog in das Verzeichnis *"PROFINET IO → Weitere Feldgeräte → I/O → VIPA EtherCAT System"* und binden Sie das IO Device *"SLIO EtherCAT System"* an Ihr PROFINET-System an.

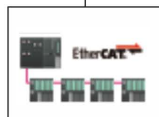
2. ➔ Klicken Sie auf das eingefügte IO Device *"EtherCAT-Netzwerk"* und definieren Sie die Bereiche für Ein- und Ausgabe, indem Sie den entsprechenden *"Out"*- bzw. *"In"*-Bereich auf einen Steckplatz ziehen.

Legen Sie folgende Bereiche an:

- In 128Byte
- Out 128Byte

Steckpl.	Modul
1	
2	CPU ...
X...	PN-IO
3	

PROFINET-IO-System



Katalog

- ▼ PROFINET IO
- ▼ Weitere Feldgeräte
- ▼ I/O
- ▼ VIPA EtherCAT System
 - SLIO EtherCAT System
 - In 1024 byte
 - In 128 byte
 - Out 1024 byte
 - Out 128 byte

Steckpl.	Baugruppe	Bestellnummer
0	...	
1	In 128 byte	
2	Out 128 byte	
3		
4		
...		

3. ➔ Wählen Sie *"Station → Speichern und übersetzen"*

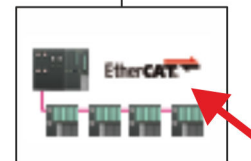
Sigma-7W EtherCAT Doppelachs-Antrieb konfigurieren



Die Konfiguration des Doppelachs-Antriebs erfolgt im *SPEED7 EtherCAT Manager*.



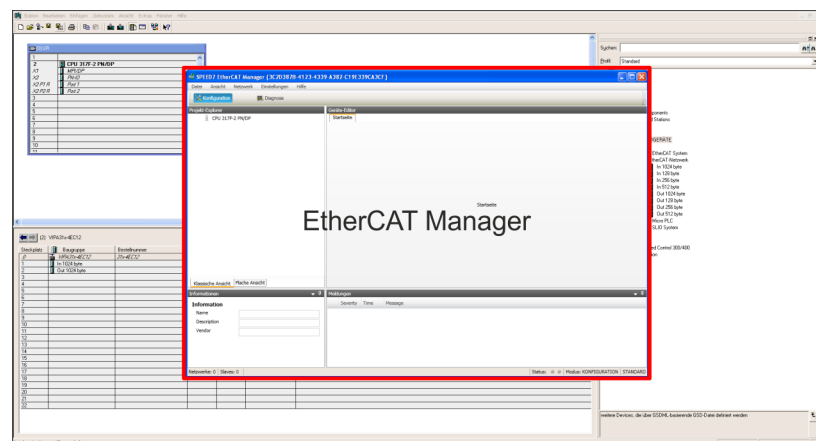
Vor dem Aufruf des SPEED7 EtherCAT Manager müssen Sie immer Ihr Projekt mit "Station → Speichern und übersetzen" speichern.

Steckpl.	Modul
1	
2	CPU ...
X...	<i>PN-IO</i>
3	

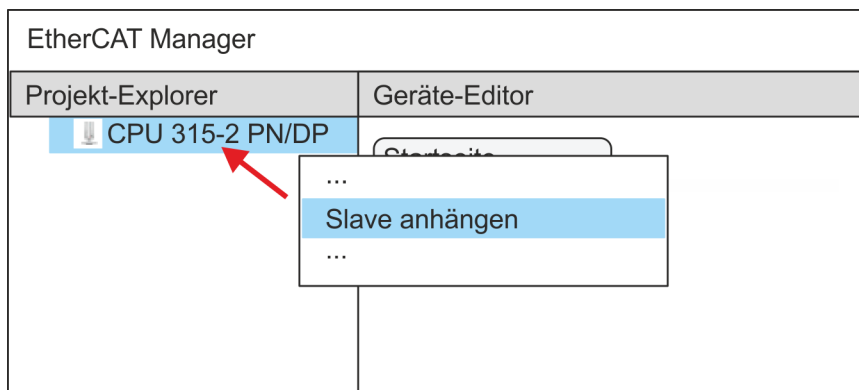


1.  Klicken Sie auf das eingefügte IO Device "*EtherCAT-Netzwerk*" und wählen Sie "*Kontextmenü → Device Tool starten → SPEED7 EtherCAT Manager*".
 Der *SPEED7 EtherCAT Manager* wird gestartet. Hier können Sie die EtherCAT-Kommunikation zu Ihrem *Sigma-7W* EtherCAT Doppelachs-Antrieb konfigurieren.

Näheres zum Einsatz des *SPEED7 EtherCAT Manager* finden Sie im zugehörigen Handbuch bzw. in der Onlinehilfe.



3. ➤ Damit der *Sigma-7W* EtherCAT Doppelachs-Antrieb im *SPEED7 EtherCAT Manager* konfiguriert werden kann, ist die entsprechende ESI-Datei zu installieren. Die ESI-Datei für den *Sigma-7W* EtherCAT Doppelachs-Antrieb finden Sie unter www.yaskawa.eu.com unter "*Service ➔ Drives & Motion Software*". Laden Sie die zu Ihrem Antrieb passende ESI-Datei herunter. Entpacken Sie diese falls erforderlich.
4. ➤ Öffnen Sie im *SPEED7 EtherCAT Manager* über "*Datei ➔ ESI-Verwaltung*" das Dialogfenster "*ESI-Manager*".
5. ➤ Klicken Sie im "*ESI-Manager*" auf [Datei hinzufügen] und wählen Sie Ihre ESI-Datei aus. Mit [Öffnen] wird die ESI-Datei im *SPEED7 EtherCAT Manager* installiert.
6. ➤ Schließen Sie den "*ESI-Manager*".
⇒ Ihr *Sigma-7W* EtherCAT Doppelachs-Antrieb steht Ihnen nun zur Konfiguration zur Verfügung.



7. ➤ Klicken Sie im EtherCAT Manager auf ihre CPU und öffnen Sie über "Kontextmenü ➔ *Slave anhängen*" das Dialogfenster zum Hinzufügen eines EtherCAT-Slave.
- ⇒ Das Dialogfenster zur Auswahl eines EtherCAT-Slave wird geöffnet.
8. ➤ Wählen Sie Ihren *Sigma-7W* EtherCAT Doppelachs-Antrieb und bestätigen Sie Ihre Auswahl mit [OK].
- ⇒ Der *Sigma-7W* EtherCAT Doppelachs-Antrieb wird an den Master angebunden und kann nun konfiguriert werden.

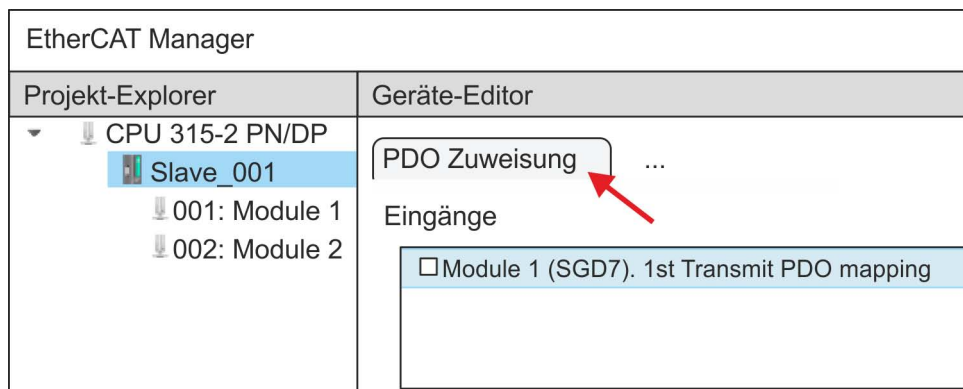
9. ➤



PDOs können Sie nur im "Experten-Modus" bearbeiten! Ansonsten werden die Schaltflächen ausgeblendet. Durch Aktivierung des "Experten-Modus" können Sie in die erweiterte Bearbeitung umschalten.

Aktivieren Sie den *Experten-Modus* durch Aktivierung von "Ansicht ➔ Experte".

10. ➤ Klicken Sie im *SPEED7 EtherCAT Manager* auf den *Sigma-7W* EtherCAT Slave und wählen Sie im "Geräte-Editor" den Reiter "PDO-Zuweisung" an.



⇒ Dieser Dialog zeigt eine Auflistung aller PDOs.

11. Durch Anwahl des entsprechenden PDO-Mappings können Sie mit [Bearbeiten] die PDOs bearbeiten. Wählen Sie das Mapping "Module 1 (SGD7). 1st Transmit PDO mapping" an und klicken Sie auf [Bearbeiten].



Bitte beachten Sie, dass aufgrund der Voreinstellung manche PDOs nicht bearbeitet werden können. Durch Deaktivierung bereits aktivierter PDOs können Sie die Bearbeitung von gesperrten PDOs frei geben.

Geräte-Editor

PDO Zuweisung ...

Eingänge	Ausgänge
<input type="checkbox"/> Module 1 (SGD7). 1st Transmit PDO mapping	<input type="checkbox"/> Module 1 (SGD7). 1st Receive PDO mapping
<input type="checkbox"/> Module 1 (SGD7). 2nd Transmit PDO mapping	<input type="checkbox"/> Module 1 (SGD7). 2nd Receive PDO mapping
<input type="checkbox"/> Module 2 (SGD7). 1st Transmit PDO mapping	<input type="checkbox"/> Module 2 (SGD7). 1st Receive PDO mapping
<input type="checkbox"/> Module 2 (SGD7). 2nd Transmit PDO mapping	<input type="checkbox"/> Module 2 (SGD7). 2nd Receive PDO mapping

... Bearbeiten ...

- ⇒ Es öffnet sich der Dialog *"PDO bearbeiten"*. Bitte überprüfen Sie hier die aufgeführten PDO-Einstellungen und passen Sie diese ggf. an. Bitte berücksichtigen Sie hierbei auch die Reihenfolge der *"Einträge"* und ergänzen Sie diese entsprechend.

The dialog box 'PDO bearbeiten' contains the following elements:

- Allgemein:**
 - Name: Module 1 (SGD7).1st Transmit PDO
 - Index: 0x1A00 (with 'Dez' and 'Hex' buttons)
 - Flags:
 - ☐ Zwingend
 - ☐ Schreibgeschützt
 - ☐ Virtuell
 - Richtung:
 - ☒ TxPdo (Eingang)
 - ☐ RxPdo (Ausgang)
- Optional:**
 - Ausschließen:
 - ☒ 1A01
 - ☐ 1A02
 - ☐ 1A03
 - ☐ 1A10
 - ☐ 1A11
 - ☐ 1A12
 - ☐ 1A13
- Einträge:**

Name	Index	Bitlänge	Kommentar
Status word	0x6041:00	16	
Position actual internal value	0x6063:00	32	
Position actual value	0x6064:00	32	
Torque actual value	0x6077:00	16	
Following error actual value	0x60F4:00	32	
Modes of operation display	0x6061:00	8	
---	---	8	---
Digital inputs	0x60FD:00	32	

At the bottom, there are buttons: Neu, Löschen, Bearbeiten, Nach oben, Nach unten.

Für die Bearbeitung der *"Einträge"* stehen folgende Funktionen zur Verfügung:

- **Neu**
 - Hiermit können Sie in einem Dialogfenster einen neuen Eintrag anlegen, indem Sie aus dem *"CoE-Objektverzeichnis"* den entsprechenden Eintrag auswählen und Ihre Einstellungen vornehmen. Mit [OK] wird der Eintrag übernommen und in der Liste der Einträge aufgeführt.
- **Löschen**
 - Hiermit können Sie den angewählte Eintrag löschen.
- **Bearbeiten**
 - Hiermit können Sie allgemeinen Daten eines Eintrags bearbeiten.
- **Nach oben/unten**
 - Hiermit können Sie den angewählten Eintrag in der Liste nach oben bzw. nach unten bewegen.

12. Führen Sie für die Transmit PDOs folgende Einstellungen durch:**Eingänge: 1st Transmit PDO**

Module 1 (SGD7). 1st Transmit PDO mapping	Module 2 (SGD7). 1st Transmit PDO mapping
Name: Module 1 (SGD7). 1st Transmit PDO mapping	Name: Module 2 (SGD7). 1st Transmit PDO mapping
Index: 0x1A00	Index: 0x1A10
Flags: Alles deaktiviert	
Richtung: TxPdo (Eingang): aktiviert	
Ausschließen: 1A01: deaktiviert	1A11: deaktiviert
Bitte diese Einstellungen beachten, da ansonsten die PDO-Mappings nicht zeitgleich aktiviert werden können!	

Einträge	Modul 1 (Achse 1)	Modul 2 (Achse 2)	Bitlänge
Name	Index	Index	
Status word	0x6041:00	0x6841:00	16Bit
Position actual internal value	0x6063:00	0x6863:00	32Bit
Position actual value	0x6064:00	0x6864:00	32Bit
Torque actual value	0x6077:00	0x6877:00	16Bit
Following error actual value	0x60F4:00	0x68F4:00	32Bit
Modes of operation display	0x6061:00	0x6861:00	8Bit
---	---	---	8Bit
Digital inputs	0x60FD:00	0x68FD:00	32Bit

Eingänge: 2nd Transmit PDO

Module 1 (SGD7). 2nd Transmit PDO mapping	Module 2 (SGD7). 2st Transmit PDO mapping
Name: Module 1 (SGD7). 2nd Transmit PDO mapping	Name: Module 2 (SGD7). 2st Transmit PDO mapping
Index: 0x1A01	Index: 0x1A11
Flags: Alles deaktiviert	
Richtung: TxPdo (Eingang): aktiviert	
Ausschließen: 1A00, 1A02, 1A03: deaktiviert	1A10, 1A12, 1A13: deaktiviert
Bitte diese Einstellungen beachten, da ansonsten die PDO-Mappings nicht zeitgleich aktiviert werden können!	

Einträge	Modul 1 (Achse 1)	Modul 2 (Achse 2)	Bitlänge
Name	Index	Index	
Touch probe status	0x60B9:00	0x68B9:00	16Bit
Touch probe 1 position value	0x60BA:00	0x68BA:00	32Bit
Touch probe 2 position value	0x60BC:00	0x68BC:00	32Bit
Velocity actual value	0x606C:00	0x686C:00	32Bit

13. Führen Sie für die Receive PDOs folgende Einstellungen durch:**Ausgänge: 1st Receive PDO**

Module 1 (SGD7). 1st Receive PDO	Module 2 (SGD7). 1st Receive PDO
Name: Module 1 (SGD7). 1st Receive PDO mapping	Name: Module 2 (SGD7). 1st Receive PDO mapping
Index: 0x1600	Index: 0x1610
Flags: Alles deaktiviert	
Richtung: RxPdo (Ausgang): aktiviert	
Ausschließen: 1601, 1602, 1603: deaktiviert	1611, 1612, 1613: deaktiviert
Bitte diese Einstellungen beachten, da ansonsten die PDO-Mappings nicht zeitgleich aktiviert werden können!	

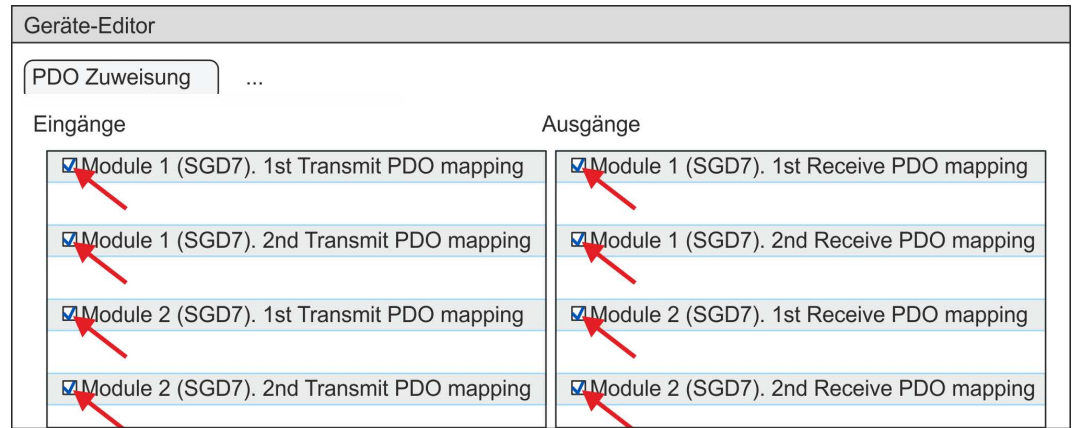
Einträge	Modul 1 (Achse 1)	Modul 2 (Achse 2)	Bitlänge
Name	Index	Index	
Control word	0x6040:00	0x6840:00	16Bit
Target position	0x607A:00	0x687A:00	32Bit
Target velocity	0x60FF:00	0x68FF:00	32Bit
Modes of operation	0x6060:00	0x6860:00	8Bit
---	---	---	8Bit
Touch probe function	0x60B8:00	0x68B8:00	16Bit

Ausgänge: 2nd Receive PDO

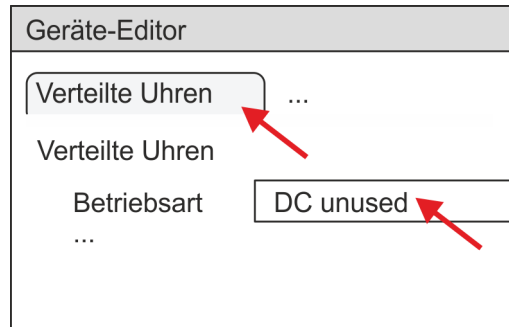
Module 1 (SGD7). 2nd Receive PDO	Module 2 (SGD7). 2nd Receive PDO
Name: Module 1 (SGD7). 2nd Receive PDO mapping	Name: Module 2 (SGD7). 2nd Receive PDO mapping
Index: 0x1601	Index: 0x1611
Flags: Alles deaktiviert	
Richtung: RxPdo (Ausgang): aktiviert	
Ausschließen: 1600, 1602, 1603: deaktiviert	1610, 1612, 1613: deaktiviert
Bitte diese Einstellungen beachten, da ansonsten die PDO-Mappings nicht zeitgleich aktiviert werden können!	

Einträge	Modul 1 (Achse 1)	Modul 2 (Achse 2)	Bitlänge
Name	Index	Index	
Profile velocity	0x6081:00	0x6881:00	32Bit
Profile acceleration	0x6083:00	0x6883:00	32Bit
Profile deceleration	0x6084:00	0x6884:00	32Bit

- 14.** Aktivieren Sie für "Module 1" und "Module 2" in PDO-Zuweisung die PDOs 1 und 2 für die Ein- und Ausgänge. Alle nachfolgenden PDOs müssen deaktiviert bleiben. Sollte dies nicht möglich sein, überprüfen Sie bitte den jeweiligen PDO-Parameter "Ausschließen".

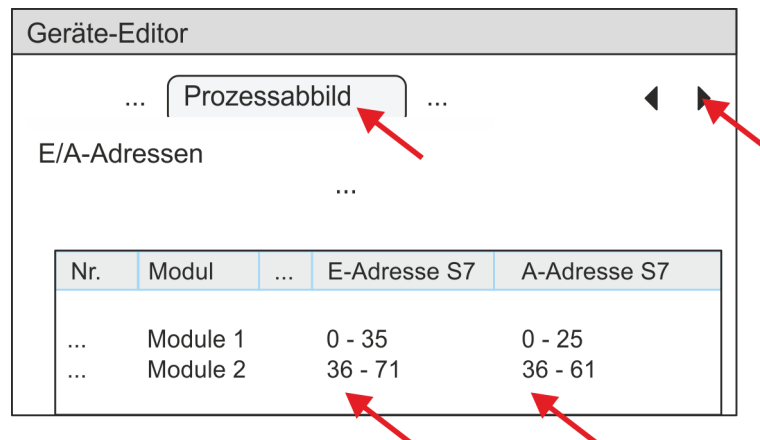


- 15.** Wählen Sie im "Geräte-Editor" des SPEED7 EtherCAT Manager den Reiter "Verteilte Uhren" an und stellen Sie "DC unused" als "Betriebsart" ein.

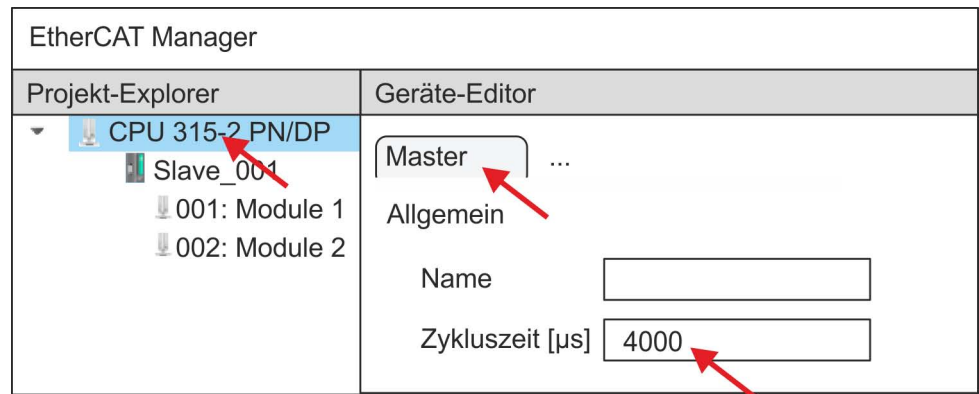


- 16.** Wählen Sie im "Geräte-Editor" über die Pfeiltaste den Reiter "Prozessabbild" an und notieren Sie sich für die Parameter des Bausteins FB 874 - VMC_InitSigma7W_EC folgende PDO-Anfangsadressen:

- Module 1: "E-Adresse S7" → "M1_PdoInputs" (hier 0)
- Module 2: "E-Adresse S7" → "M2_PdoInputs" (hier 36)
- Module 1: "A-Adresse S7" → "M1_PdoOutputs" (hier 0)
- Module 2: "A-Adresse S7" → "M2_PdoOutputs" (hier 36)



- 17.** Klicken Sie im *SPEED7 EtherCAT Manager* auf Ihre CPU und wählen Sie im "Geräte-Editor" den Reiter "Master" an.

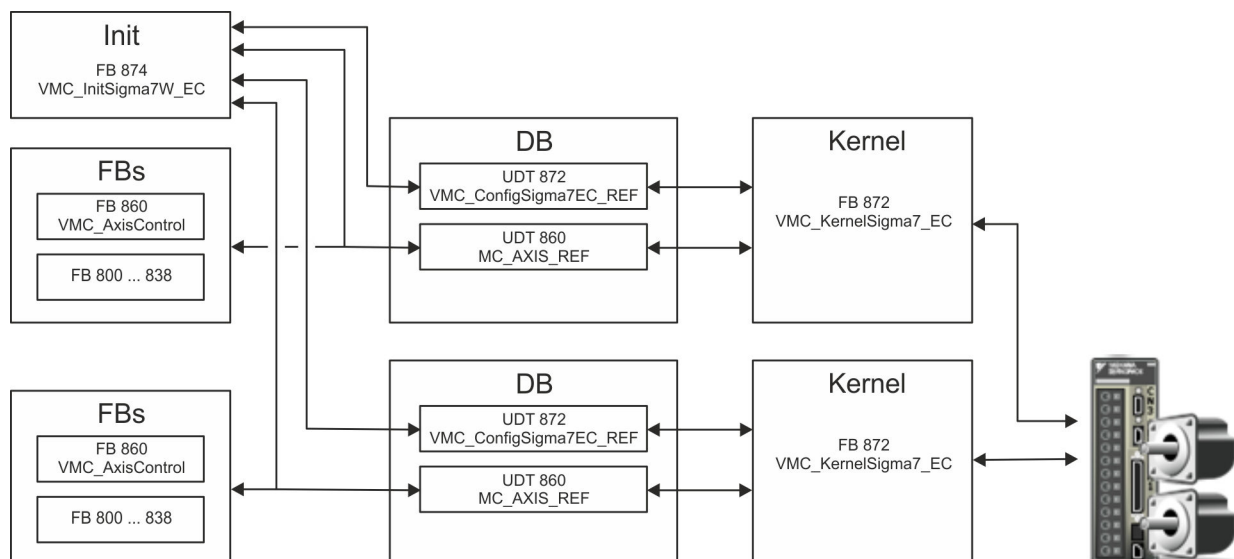


⇒ Stellen Sie für Sigma-7W (400V) Antriebe eine Zykluszeit von mindestens 4ms ein.

- 18.** Indem Sie den Dialog des *SPEED7 EtherCAT Manager* mit [X] schließen, wird die Konfiguration in die Projektierung übernommen. Sie können Ihre EtherCAT-Konfiguration jederzeit im *SPEED7 EtherCAT Manager* wieder bearbeiten, da die Konfiguration in Ihrem Projekt gespeichert wird.
- 19.** Speichern und übersetzen Sie Ihre Konfiguration.

3.3.4.3 Anwender-Programm

3.3.4.3.1 Programmstruktur



- **DB**
Für jede Achse ist ein Datenbaustein (Achs-DB) für Konfiguration und Statusdaten anzulegen. Der Datenbaustein besteht aus folgenden Datenstrukturen:
 - UDT 872 - *VMC_ConfigSigma7EC_REF*
Die Datenstruktur beschreibt den Aufbau der Konfiguration des Antriebs. Spezifische Datenstruktur für *Sigma-7* EtherCAT.
 - UDT 860 - *MC_AXIS_REF*
Die Datenstruktur beschreibt den Aufbau der Parameter und Statusinformationen von Antrieben. Allgemeine Datenstruktur für alle Antriebe und Bussysteme.
- **FB 874 - *VMC_InitSigma7W_EC***
 - Der *Init*-Baustein dient zur Konfiguration des Doppelachs-Antriebs.
 - Spezifischer Baustein für *Sigma-7W* EtherCAT.
 - Die Konfigurationsdaten für die Initialisierung sind im *Achs-DB* abzulegen.
- **FB 872 - *VMC_KernelSigma7_EC***
 - Der *Kernel*-Baustein kommuniziert mit dem Antrieb über das entsprechende Bussystem, verarbeitet die Benutzeraufträge und liefert Statusmeldungen zurück.
 - Je Achse ist der FB 872 - *VMC_KernelSigma7_EC* aufzurufen.
 - Spezifischer Baustein für *Sigma-7* EtherCAT.
 - Der Austausch der Daten erfolgt mittels des *Achs-DB*.
- **FB 860 - *VMC_AxisControl***
 - Universal-Baustein für alle Antriebe und Bussysteme.
 - Je Achse ist der FB 860 - *VMC_AxisControl* aufzurufen.
 - Unterstützt einfache Bewegungskommandos und liefert alle relevanten Statusmeldungen.
 - Der Austausch der Daten erfolgt mittels des *Achs-DB*.
 - Über die Instanzdaten des Bausteins können Sie zur Bewegungssteuerung und Statusabfrage eine Visualisierung anbinden.
 - Zusätzlich zum FB 860 - *VMC_AxisControl* haben Sie die Möglichkeit *PLCopen*-Bausteine zu nutzen.
- **FB 800 ... FB 838 - *PLCopen***
 - Die *PLCopen*-Bausteine dienen zur Programmierung von Bewegungsabläufen und Statusabfragen.
 - Je Achse sind die *PLCopen*-Bausteine aufzurufen.

3.3.4.3.2 Programmierung

Bibliothek einbinden

1. ➞ Gehen Sie in den Service-Bereich von www.vipa.com.
2. ➞ Laden Sie aus dem Downloadbereich unter "*VIPA Lib*" die *Simple Motion Control Library*.
3. ➞ Öffnen Sie mit "*Datei ➞ Dearchivieren*" das Dialogfenster zur Auswahl der ZIP-Datei.
4. ➞ Wählen Sie die entsprechende ZIP-Datei an und klicken Sie auf [Öffnen].
5. ➞ Geben Sie ein Zielverzeichnis an, in dem die Bausteine abzulegen sind und starten Sie den Entpackvorgang mit [OK].

Bausteine in Projekt kopieren

➔ Öffnen Sie die Bibliothek nach dem Entpackvorgang und ziehen Sie per Drag&Drop folgende Bausteine in *"Bausteine"* Ihres Projekts:

- *Sigma-7W EtherCAT*:
 - UDT 872 - VMC_ConfigSigma7EC_REF
 - FB 872 - VMC_KernelSigma7_EC
 - FB 874 - VMC_InitSigma7W_EC
- *Axis Control*
 - UDT 860 - MC_AXIS_REF
 - Bausteine für die gewünschten Bewegungsabläufe

Alarm-OBs anlegen

1. ➔ Klicken Sie in Ihrem Projekt auf *"Bausteine"* und wählen Sie *"Kontextmenü ➔ Neues Objekt einfügen ➔ Organisationsbaustein"*.
⇒ Das Dialogfenster *"Eigenschaften Organisationsbaustein"* öffnet sich.
2. ➔ Fügen Sie nacheinander OB 57, OB 82 und OB 86 Ihrem Projekt hinzu.

Achs-DB für "Module 1" anlegen

1. ➔ Klicken Sie in Ihrem Projekt auf *"Bausteine"* und wählen Sie *"Kontextmenü ➔ Neues Objekt einfügen ➔ Datenbaustein"*.

Geben Sie folgende Parameter an:

- **Name und Typ**
 - Die DB-Nr. als *"Name"* können Sie frei wählen wie z.B. DB 10.
 - Stellen Sie *"Global-DB"* als *"Typ"* ein.
- **Symbolischer Name**
 - Geben Sie *"Axis01"* an.

Bestätigen Sie Ihre Eingaben mit [OK].

⇒ Der Baustein wird angelegt.

2. ➔ Öffnen Sie DB 10 "Axis01" durch Doppelklick.
 - Legen Sie in "Axis01" die Variable "Config" vom Typ UDT 872 an. Dies sind spezifische Achs-Konfigurationsdaten.
 - Legen Sie in "Axis01" die Variable "Axis" vom Typ UDT 860 an. Während des Betriebs werden hier alle Betriebsdaten der Achse abgelegt.

DB10

Adresse	Name	Typ	...
		Struct	
...	Config	"VMC_ConfigSigma7EC_REF"	
...	Axis	"MC_AXIS_REF"	
...		END_STRUCT	

Achs-DB für "Module 2" anlegen

1. ➔ Fügen Sie Ihrem Projekt einen weiteren DB als *Achs-DB* hinzu und vergeben Sie diesem den Namen *"Axis02"*. Die DB-Nr. können Sie frei wählen wie z.B. DB11.
⇒ Der Baustein wird angelegt.

2. ➔ Öffnen Sie DB 11 "Axis02" durch Doppelklick.

- Legen Sie in "Axis02" die Variable "Config" vom Typ UDT 872 an. Dies sind spezifische Achs-Konfigurationsdaten.
- Legen Sie in "Axis02" die Variable "Axis" vom Typ UDT 860 an. Während des Betriebs werden hier alle Betriebsdaten der Achse abgelegt.

DB 11

Adresse	Name	Typ	...
		Struct	
...	Config	"VMC_ConfigSigma7EC_REF"	
...	Axis	"MC_AXIS_REF"	
...		END_STRUCT	

OB 1

Konfiguration der Doppel-
achse

Öffnen Sie den OB 1 und programmieren Sie folgende FB-Aufrufe mit zugehörigen DBs:

→ FB 874 - VMC_InitSigma7W_EC, DB 874 ↪ *Kapitel 3.3.5.3 "FB 874 - VMC_InitSigma7W_EC - Sigma-7W EtherCAT Initialisierung" auf Seite 126*

Geben Sie unter *M1/M2_PdoInputs* bzw. *M1/M2_PdoOutputs* die Adresse aus dem *SPEED7 EtherCAT Manager* für die entsprechende Achse an. ↪ 117

```
⇒ CALL "VMC_InitSigma7W_EC" , "DI_InitSgm7WETC01"
   Enable                                     :=TRUE
   LogicalAddress                             :=0
   M1_PdoInputs                               :=0 (EtherCAT-Manager
                                                Module1: E-Adresse S7)

   M1_PdoOutputs                             :=0 (EtherCAT-Manager
                                                Module1: A-Adresse S7)

   M1_EncoderType                             :=2
   M1_EncoderResolutionBits                   :=20
   M1_FactorPosition                         :=1.048576e+006
   M1_FactorVelocity                         :=1.048576e+006
   M1_FactorAcceleration                     :=1.048576e+002
   M1_OffsetPosition                         :=0.000000e+000
   M1_MaxVelocityApp                         :=5.000000e+001
   M1_MaxAccelerationApp                     :=1.000000e+002
   M1_MaxDecelerationApp                     :=1.000000e+002
   M1_MaxVelocityDrive                       :=6.000000e+001
   M1_MaxAccelerationDrive                   :=1.500000e+002
   M1_MaxDecelerationDrive                   :=1.500000e+002
   M1_MaxPosition                           :=1.048500e+003
   M1_MinPosition                           :=-1.048514e+003
   M1_EnableMaxPosition                       :=TRUE
   M1_EnableMinPosition                       :=TRUE
   M2_PdoInputs                               :=36 (EtherCAT-Manager
                                                Module2: E-Adresse S7)

   M2_PdoOutputs                             :=36 (EtherCAT-Manager
                                                Module2: A-Adresse S7)

   M2_EncoderType                             :=2
   M2_EncoderResolutionBits                   :=20
   M2_FactorPosition                         :=1.048576e+006
   M2_FactorVelocity                         :=1.048576e+006
   M2_FactorAcceleration                     :=1.048576e+002
   M2_OffsetPosition                         :=0.000000e+000
   M2_MaxVelocityApp                         :=5.000000e+001
   M2_MaxAccelerationApp                     :=1.000000e+002
   M2_MaxDecelerationApp                     :=1.000000e+002
   M2_MaxVelocityDrive                       :=6.000000e+001
   M2_MaxAccelerationDrive                   :=1.500000e+002
   M2_MaxDecelerationDrive                   :=1.500000e+002
   M2_MaxPosition                           :=1.048500e+003
   M2_MinPosition                           :=-1.048514e+003
   M2_EnableMaxPosition                       :=TRUE
   M2_EnableMinPosition                       :=TRUE
   M1_MinUserPosition                         :=-1000.0
   M1_MaxUserPosition                         :=1000.0
   M2_MinUserPosition                         :=-1000.0
   M2_MaxUserPosition                         :=1000.0
   Valid                                     := "InitS7WEC1_Valid"
   Error                                     := "InitS7WEC1_Error"
```

```

ErrorID           := "InitS7WEC1_ErrorID"
M1_Config         := "Axis01".Config
M1_Axis           := "Axis01".Axis
M2_Config         := "Axis02".Config
M2_Axis           := "Axis02".Axis

```

Kernel für die jeweilige Achse beschalten

Der *Kernel* verarbeitet die Benutzerkommandos und gibt sie entsprechend aufbereitet an den Antrieb über das jeweilige Bussystem weiter.

➔ FB 872 - VMC_KernelSigma7_EC, DB 872 für Achse 1

FB 872 - VMC_KernelSigma7_EC, DB 1872 für Achse 2 ↪ *Kapitel 3.2.5.2 "FB 872 - VMC_KernelSigma7_EC - Sigma-7 EtherCAT Kernel" auf Seite 84*

```

⇒ CALL "VMC_KernelSigma7_EC" , DB 872
   Init := "KernelS7WEC1_Init"
   Config := "Axis01".Config
   Axis  := "Axis01".Axis

CALL "VMC_KernelSigma7_EC" , DB 1872
   Init := "KernelS7WEC2_Init"
   Config := "Axis02".Config
   Axis  := "Axis02".Axis

```

Baustein für Bewegungsabläufe beschalten

Zur Vereinfachung soll hier die Beschaltung des FB 860 - VMC_AxisControl gezeigt werden. Dieser Universalbaustein unterstützt einfache Bewegungskommandos und liefert Statusmeldungen zurück. Die Ein- und Ausgänge können Sie individuell beschalten. Bitte geben Sie unter "Axis" die Referenz zu den entsprechenden Achsdaten im *Achs-DB* an.

➔ FB 860 - VMC_AxisControl, DB 860 ↗ *Kapitel 3.4.2.2 "FB 860 - VMC_AxisControl - Control-Baustein Achskontrolle" auf Seite 132*

```
⇒ CALL "VMC_AxisControl" , "DI_AxisControl01"
    AxisEnable      := "AxCtrl1_AxisEnable"
    AxisReset       := "AxCtrl1_AxisReset"
    HomeExecute     := "AxCtrl1_HomeExecute"
    HomePosition    := "AxCtrl1_HomePosition"
    StopExecute     := "AxCtrl1_StopExecute"
    MvVelocityExecute := "AxCtrl1_MvVelExecute"
    MvRelativeExecute := "AxCtrl1_MvRelExecute"
    MvAbsoluteExecute := "AxCtrl1_MvAbsExecute"
    PositionDistance := "AxCtrl1_PositionDistance"
    Velocity        := "AxCtrl1_Velocity"
    Acceleration     := "AxCtrl1_Acceleration"
    Deceleration     := "AxCtrl1_Deceleration"
    JogPositive      := "AxCtrl1_JogPositive"
    JogNegative      := "AxCtrl1_JogNegative"
    JogVelocity      := "AxCtrl1_JogVelocity"
    JogAcceleration  := "AxCtrl1_JogAcceleration"
    JogDeceleration  := "AxCtrl1_JogDeceleration"
    AxisReady       := "AxCtrl1_AxisReady"
    AxisEnabled      := "AxCtrl1_AxisEnabled"
    AxisError        := "AxCtrl1_AxisError"
    AxisErrorID      := "AxCtrl1_AxisErrorID"
    DriveWarning     := "AxCtrl1_DriveWarning"
    DriveError       := "AxCtrl1_DriveError"
    DriveErrorID     := "AxCtrl1_DriveErrorID"
    IsHomed          := "AxCtrl1_IsHomed"
    ModeOfOperation  := "AxCtrl1_ModeOfOperation"
    PLCOpenState     := "AxCtrl1_PLCOpenState"
    ActualPosition   := "AxCtrl1_ActualPosition"
    ActualVelocity   := "AxCtrl1_ActualVelocity"
    CmdDone          := "AxCtrl1_CmdDone"
    CmdBusy          := "AxCtrl1_CmdBusy"
    CmdAborted       := "AxCtrl1_CmdAborted"
    CmdError         := "AxCtrl1_CmdError"
    CmdErrorID       := "AxCtrl1_CmdErrorID"
    DirectionPositive := "AxCtrl1_DirectionPos"
    DirectionNegative := "AxCtrl1_DirectionNeg"
    SWLimitMinActive := "AxCtrl1_SWLimitMinActive"
    SWLimitMaxActive := "AxCtrl1_SWLimitMaxActive"
    HWLimitMinActive := "AxCtrl1_HWLimitMinActive"
    HWLimitMaxActive := "AxCtrl1_HWLimitMaxActive"
    Axis             := "Axis..." . Axis
```

Geben Sie unter *Axis* für die Achse 1 "Axis01" und für die Achse 2 "Axis02" an.



Für komplexe Bewegungsaufgaben können Sie die PLCOpen-Bausteine verwenden. Hier müssen Sie ebenfalls unter Axis die Referenz zu den Achsdaten im Achs-DB angeben.

Ihr Projekt beinhaltet nun folgende Bausteine:

- OB 1 - Main
- OB 57 - DP Manufacturer Alarm
- OB 82 - I/O_FLT1
- OB 86 - Rack_FLT

- FB 860 - VMC_AxisControl mit Instanz-DB
- FB 872 - VMC_KernelSigma7_EC mit Instanz-DB
- FB 874 - VMC_InitSigma7W_EC mit Instanz-DB
- UDT 860 - MC_Axis_REF
- UDT 872 - VMC_ConfigSigma7EC_REF

Zeitlicher Ablauf

1. ➔ Wechseln Sie in den Siemens SIMATIC Manager und übertragen Sie Ihr Projekt in die CPU.

Die Übertragung kann ausschließlich aus dem Siemens SIMATIC Manager erfolgen - nicht Hardware-Konfigurator!



Da Slave- und Modulparameter mittels SDO-Zugriff bzw. SDO-Init-Kommando übertragen werden, bleibt die Parametrierung solange bestehen, bis ein Power-Cycle durchgeführt wird oder neue Parameter für die gleichen SDO-Objekte übertragen werden.

Beim Utlöschen werden Slave- und Modul-Parameter nicht zurückgesetzt!

⇒ Sie können jetzt Ihre Applikation in Betrieb nehmen.



VORSICHT!

Bitte beachten Sie immer die Sicherheitshinweise zu ihrem Antrieb, insbesondere bei der Inbetriebnahme!

2. ➔ Bevor der Doppelachs-Antrieb gesteuert werden kann, muss diese initialisiert werden. Rufen Sie hierzu den *Init*-Baustein FB 874 - VMC_InitSigma7W_EC mit *Enable* = TRUE auf.

⇒ Der Ausgang *Valid* meldet TRUE zurück. Im Fehlerfall können Sie durch Auswertung der *ErrorID* den Fehler ermitteln.

Den *Init*-Baustein müssen Sie erneut aufrufen, wenn Sie einen neuen Achs-DB laden oder Parameter am *Init*-Baustein geändert wurden.



Fahren Sie erst fort, wenn der Init-Baustein keinen Fehler meldet!

3. ➔ Stellen Sie sicher, dass für jede Achse der *Kernel*-Baustein FB 872 - VMC_KernelSigma7_EC zyklisch aufgerufen wird. Auf diese Weise werden Steuersignale an den Antrieb übergeben und Statusmeldungen übermittelt.
4. ➔ Programmieren Sie für jede Achse Ihre Applikation mit dem FB 860 - VMC_AxisControl oder mit den PLCopen Bausteinen auf.

Steuerung des Antriebs über HMI

Sie haben die Möglichkeit über ein HMI Ihren Antrieb zu steuern. Hierzu gibt es für Movicon eine vorgefertigte Symbolbibliothek für den Zugriff auf den VMC_AxisControl Funktionsbaustein. ➔ Kapitel 7 "Antrieb über HMI steuern" auf Seite 313

3.3.4.4 Projekt kopieren

Vorgehensweise

Im Beispiel wird die Station "Source" kopiert und als "Target " gespeichert.

1. ➤ Öffnen Sie die Hardware-Konfiguration der "Source"-CPU und starten Sie hier den *SPEED7 EtherCAT Manager*.
2. ➤ Speichern Sie im *SPEED7 EtherCAT Manager* über "*Datei ➔ Speichern unter*" die Konfiguration in Ihrem Arbeitsverzeichnis.
3. ➤ Schließen Sie den *SPEED7 EtherCAT Manager* und den Hardware-Konfigurator wieder.
4. ➤ Kopieren Sie die Station "Source" mit Strg+C und fügen Sie diese mit Strg+V als "Target" in Ihr Projekt ein.
5. ➤ Wechseln Sie in den "Baustein"-Ordner der "Target"-CPU löschen Sie die "Systemdaten".
6. ➤ Öffnen Sie die Hardware-Konfiguration der "Target"-CPU. Passen Sie die IP-Adressdaten an oder vernetzen Sie die CPU bzw. den CP neu.



Vor dem Aufruf des SPEED7 EtherCAT Manager müssen Sie immer Ihr Projekt mit "Station ➔ Speichern und übersetzen" speichern.

7. ➤ Speichern Sie Ihr Projekt mit "*Station ➔ Speichern und übersetzen*".
8. ➤ Öffnen Sie den *SPEED7 EtherCAT Manager*.
9. ➤ Laden Sie mit "*Datei ➔ Öffnen*" die Konfiguration aus Ihrem Arbeitsverzeichnis.
10. ➤ Schließen Sie den *SPEED7 EtherCAT Manager* wieder.
11. ➤ Speichern und übersetzen Sie Ihre Konfiguration.

3.3.5 Antriebsspezifische Bausteine

3.3.5.1 UDT 872 - VMC_ConfigSigma7EC_REF - Sigma-7 EtherCAT Datenstruktur Achskonfiguration

Dies ist eine benutzerdefinierte Datenstruktur, die Informationen zu den Konfigurationsdaten beinhaltet. Die UDT ist speziell angepasst an die Verwendung eines *Sigma-7*-Antriebs, welcher über EtherCAT angebunden ist.

3.3.5.2 FB 872 - VMC_KernelSigma7_EC - Sigma-7 EtherCAT Kernel

Beschreibung

Dieser Baustein setzt die Antriebskommandos für eine *Sigma-7* Achse über EtherCAT um und kommuniziert mit dem Antrieb. Je *Sigma-7* Achse ist eine Instanz dieses FBs zyklisch aufzurufen.



Bitte beachten Sie, dass dieser Baustein intern den SFB 238 aufruft.

Im SPEED7 Studio wird dieser Baustein automatisch in Ihr Projekt eingefügt.

Im Siemens SIMATIC Manager müssen Sie den SFB 238 aus der Motion Control Library in Ihr Projekt kopieren.

Parameter	Deklaration	Datentyp	Beschreibung
Init	INPUT	BOOL	Mit einer Flanke 0-1 wird der Baustein intern zurückgesetzt. Hierbei werden bestehende Bewegungskommandos abgebrochen und der Baustein wird initialisiert.
Config	IN_OUT	UDT872	Datenstruktur zur Übergabe von achsabhängigen Konfigurationsdaten an den <i>AxisKernel</i> .
Axis	IN_OUT	MC_AXIS_REF	Datenstruktur zur Übergabe von achsabhängigen Informationen an <i>AxisKernel</i> und PLCopen-Bausteine.

3.3.5.3 FB 874 - VMC_InitSigma7W_EC - Sigma-7W EtherCAT Initialisierung

Beschreibung

Dieser Baustein dient zur Konfiguration der Doppelachse eines *Sigma-7W*-Antriebs. Der Baustein ist speziell angepasst an die Verwendung eines *Sigma-7W*-Antriebs, welcher über EtherCAT angebunden ist.

Parameter	Deklaration	Datentyp	Beschreibung
M1_Config	IN_OUT	UDT872	Datenstruktur zur Übergabe von achsabhängigen Konfigurationsdaten an den <i>AxisKernel</i> für Achse 1.
M1_Axis	IN_OUT	MC_AXIS_REF	Datenstruktur zur Übergabe von achsabhängigen Informationen an <i>AxisKernel</i> und PLCopen-Bausteine für Achse 1.
M2_Config	IN_OUT	UDT872	Datenstruktur zur Übergabe von achsabhängigen Konfigurationsdaten an den <i>AxisKernel</i> für Achse 2.
M2_Axis	IN_OUT	MC_AXIS_REF	Datenstruktur zur Übergabe von achsabhängigen Informationen an <i>AxisKernel</i> und PLCopen-Bausteine für Achse 2.
Enable	INPUT	BOOL	Freigabe der Initialisierung
LogicalAddress	INPUT	INT	Startadresse der PDO-Eingangsdaten

Parameter	Deklaration	Datentyp	Beschreibung
M1_PdoInputs	INPUT	INT	Startadresse der Eingabe-PDOs für Achse 1
M1_PdoOutputs	INPUT	INT	Startadresse der Ausgabe-PDOs für Achse 1
M1_EncoderType	INPUT	INT	Encoder-Typ von Achse 1 <ul style="list-style-type: none"> ■ 1: Absolut-Encoder ■ 2: Inkremental-Encoder
M1_EncoderResolutionBits	INPUT	INT	Anzahl der Bits, die einer Geber-Umdrehung von Achse 1 entsprechen. Default: 20
M1_FactorPosition	INPUT	REAL	Faktor zur Umrechnung der Position von Benutzereinheiten [u] in Antriebseinheiten [Inkrement] und zurück von Achse 1. <p>Es gilt: $p_{[\text{Inkrement}]} = p_{[u]} \times \text{FactorPosition}$</p> <p>Bitte berücksichtigen sie auch den Faktor, welchen Sie am Antrieb über die Objekte 0x2701:1 und 0x2701:2 vorgeben können. Dieser sollte 1 sein.</p>
M1_FactorVelocity	INPUT	REAL	Faktor zur Umrechnung der Geschwindigkeit von Benutzereinheiten [u/s] in Antriebseinheiten [Inkrement/s] und zurück von Achse 1. <p>Es gilt: $v_{[\text{Inkrement/s}]} = v_{[u/s]} \times \text{FactorVelocity}$</p> <p>Bitte berücksichtigen sie auch den Faktor, welchen Sie am Antrieb über die Objekte 0x2702:1 und 0x2702:2 vorgeben können. Dieser sollte 1 sein.</p>
M1_FactorAcceleration	INPUT	REAL	Faktor zur Umrechnung der Beschleunigung von Benutzereinheiten [u/s^2] in Antriebseinheiten [$10^{-4} \times \text{Inkrement/s}^2$] und zurück von Achse 1. <p>Es gilt: $10^{-4} \times a_{[\text{Inkrement/s}^2]} = a_{[u/s^2]} \times \text{FactorAcceleration}$</p> <p>Bitte berücksichtigen sie auch den Faktor, welchen Sie am Antrieb über die Objekte 0x2703:1 und 0x2703:2 vorgeben können. Dieser sollte 1 sein.</p>
M1_OffsetPosition	INPUT	REAL	Offset für die Nullposition von Achse 1 [u].
M1_MaxVelocityApp	INPUT	REAL	Maximale Geschwindigkeit der Applikation von Achse 1 [u/s]. <p>Die Kommandoeingaben werden vor Ausführung auf den Maximalwert überprüft.</p>
M1_MaxAccelerationApp	INPUT	REAL	Maximale Beschleunigung der Applikation von Achse 1 [u/s^2]. <p>Die Kommandoeingaben werden vor Ausführung auf den Maximalwert überprüft.</p>
M1_MaxDecelerationApp	INPUT	REAL	Maximale Verzögerung der Applikation von Achse 1 [u/s^2]. <p>Die Kommandoeingaben werden vor Ausführung auf den Maximalwert überprüft.</p>
M1_MaxPosition	INPUT	REAL	Maximale Position für die Überwachung der Softwarelimits von Achse 1 [u].
M1_MinPosition	INPUT	REAL	Minimale Position für die Überwachung der Softwarelimits von Achse 1 [u].

Parameter	Deklaration	Datentyp	Beschreibung
M1_EnableMaxPosition	INPUT	BOOL	Überwachung maximale Position von Achse 1 ■ TRUE: Aktivierung der Überwachung der maximalen Position.
M1_EnableMinPosition	INPUT	BOOL	Überwachung minimale Position von Achse 1 ■ TRUE: Aktivierung der Überwachung der minimalen Position.
M2_PdoInputs	INPUT	INT	Startadresse der Eingabe-PDOs für Achse 2
M2_PdoOutputs	INPUT	INT	Startadresse der Ausgabe-PDOs für Achse 2
M2_EncoderType	INPUT	INT	Encoder-Typ von Achse 2 ■ 1: Absolut-Encoder ■ 2: Inkremental-Encoder
M2_EncoderResolutionBits	INPUT	INT	Anzahl der Bits, die einer Geber-Umdrehung von Achse 2 entsprechen. Default: 20
M2_FactorPosition	INPUT	REAL	Faktor zur Umrechnung der Position von Benutzereinheiten [u] in Antriebseinheiten [Inkrement] und zurück von Achse 2. Es gilt: $p_{[\text{Inkrement}]} = p_{[u]} \times \text{FactorPosition}$ Bitte berücksichtigen sie auch den Faktor, welchen Sie am Antrieb über die Objekte 0x2701:1 und 0x2701:2 vorgeben können. Dieser sollte 1 sein.
M2_FactorVelocity	INPUT	REAL	Faktor zur Umrechnung der Geschwindigkeit von Benutzereinheiten [u/s] in Antriebseinheiten [Inkrement/s] und zurück von Achse 2. Es gilt: $v_{[\text{Inkrement/s}]} = v_{[u/s]} \times \text{FactorVelocity}$ Bitte berücksichtigen sie auch den Faktor, welchen Sie am Antrieb über die Objekte 0x2702:1 und 0x2702:2 vorgeben können. Dieser sollte 1 sein.
M2_FactorAcceleration	INPUT	REAL	Faktor zur Umrechnung der Beschleunigung von Benutzereinheiten [u/s^2] in Antriebseinheiten [$10^{-4} \times \text{Inkrement}/s^2$] und zurück von Achse 2. Es gilt: $10^{-4} \times a_{[\text{Inkrement}/s^2]} = a_{[u/s^2]} \times \text{FactorAcceleration}$ Bitte berücksichtigen sie auch den Faktor, welchen Sie am Antrieb über die Objekte 0x2703:1 und 0x2703:2 vorgeben können. Dieser sollte 1 sein.
M2_OffsetPosition	INPUT	REAL	Offset für die Nullposition von Achse 2 [u].
M2_MaxVelocityApp	INPUT	REAL	Maximale Geschwindigkeit der Applikation von Achse 2 [u/s]. Die Kommandoeingaben werden vor Ausführung auf den Maximalwert überprüft.
M2_MaxAccelerationApp	INPUT	REAL	Maximale Beschleunigung der Applikation von Achse 2 [u/s^2]. Die Kommandoeingaben werden vor Ausführung auf den Maximalwert überprüft.



Parameter	Deklaration	Datentyp	Beschreibung
M2_MaxDecelerationApp	INPUT	REAL	Maximale Verzögerung der Applikation von Achse 2 [u/s ²]. Die Kommandoeingaben werden vor Ausführung auf den Maximalwert überprüft.
M2_MaxPosition	INPUT	REAL	Maximale Position für die Überwachung der Softwarelimits von Achse 2 [u].
M2_MinPosition	INPUT	REAL	Minimale Position für die Überwachung der Softwarelimits von Achse 2 [u].
M2_EnableMaxPosition	INPUT	BOOL	Überwachung maximale Position von Achse 2 ■ TRUE: Aktivierung der Überwachung der maximalen Position.
M2_EnableMinPosition	INPUT	BOOL	Überwachung minimale Position von Achse 2 ■ TRUE: Aktivierung der Überwachung der minimalen Position.
M1_MinUserPosition	OUTPUT	REAL	Minimale Benutzerposition für Achse 1 basierend auf dem minimalen Encoder Wert von 0x80000000 und dem <i>FactorPosition</i> [u].
M1_MaxUserPosition	OUTPUT	REAL	Maximale Benutzerposition für Achse 1 basierend auf dem maximalen Encoder Wert von 0x7FFFFFFF und dem <i>FactorPosition</i> [u].
M2_MinUserPosition	OUTPUT	REAL	Minimale Benutzerposition für Achse 2 basierend auf dem minimalen Encoder Wert von 0x80000000 und dem <i>FactorPosition</i> [u].
M2_MaxUserPosition	OUTPUT	REAL	Maximale Benutzerposition für Achse 2 basierend auf dem maximalen Encoder Wert von 0x7FFFFFFF und dem <i>FactorPosition</i> [u].
Valid	OUTPUT	BOOL	Initialisierung ■ TRUE: Initialisierung ist gültig.
Error	OUTPUT	BOOL	■ Fehler – TRUE: Ein Fehler ist aufgetreten. Zusätzliche Fehlerinformationen können dem Parameter <i>ErrorID</i> entnommen werden. Die Achse wird gesperrt.
ErrorID	OUTPUT	WORD	Zusätzliche Fehlerinformationen 🔗 Kapitel 9 "ErrorID - Zusätzliche Fehlerinformationen" auf Seite 340

3.4 Bausteine zur Achskontrolle

3.4.1 Übersicht

Unter *Axis Control* finden Sie die Bausteine zur Programmierung von Bewegungsaufgaben und Statusabfragen.

Einfache Bewegungsaufgaben

Baustein	
UDT 860 - MC_AXIS_REF - Datenstruktur für Achse	 132
FB 860 - VMC_AxisControl - Steuerung von Antriebsefunktionen und Auslesen von Antriebszuständen	 132

Komplexe Bewegungsaufgaben - PLCopen-Bausteine

Baustein	Siehe Seite
UDT 860 - MC_AXIS_REF - Datenstruktur für Achse	 136
UDT 861 - MC_TRIGGER_REF - Datenstruktur	 136
FB 800 - MC_Power Achse freigeben bzw. sperren	 137
FB 801 - MC_Home Achse referenzieren	 139
FB 802 - MC_Stop Achse stoppen	 141
FB 803 - MC_Halt Achse anhalten	 143
FB 804 - MC_MoveRelative Achse relativ verfahren	 145
FB 805 - MC_MoveVelocity Achse verfahren mit konstanter Geschwindigkeit	 147
FB 808 - MC_MoveAbsolute Achse auf absolute Position verfahren	 149
FB 811 - MC_Reset Achse zurücksetzen	 151
FB 812 - MC_ReadStatus PLCopen-State der Achse lesen	 153
FB 813 - MC_ReadAxisError Fehler von Achse lesen	 155
FB 814 - MC_ReadParameter Parameter der Achse lesen	 157
FB 815 - MC_WriteParameter Parameter an Achse schreiben	 159
FB 816 - MC_ReadActualPosition Aktuelle Position der Achse lesen	 161
FB 817 - MC_ReadActualVelocity Aktuelle Geschwindigkeit der Achse lesen	 162

Baustein	Siehe Seite
FB 818 - MC_ReadAxisInfo Zusatzinformationen der Achse lesen	163
FB 819 - MC_ReadMotionState Zustand Bewegungsauftrag lesen	165
FB 823 - MC_TouchProbe Achsposition erfassen	167
FB 824 - MC_AbortTrigger Achsposition erfassen abbrechen	169
FB 825 - MC_ReadBoolParameter Boolean-Parameter von Achse lesen	170
FB 826 - MC_WriteBoolParameter Boolean-Parameter an Achse schreiben	172
FB 827 - VMC_ReadDWordParameter Doppelwort-Parameter von Achse lesen	174
FB 828 - VMC_WriteDWordParameter Doppelwort-Parameter an Achse schreiben	176
FB 829 - VMC_ReadWordParameter Wort-Parameter von Achse lesen	178
FB 830 - VMC_WriteWordParameter Wort-Parameter an Achse schreiben	180
FB 831 - VMC_ReadByteParameter Byte-Parameter von Achse lesen	182
FB 832 - VMC_WriteByteParameter Byte-Parameter an Achse schreiben	184
FB 833 - VMC_ReadDriveParameter Antriebsparameter lesen	186
FB 834 - VMC_WriteDriveParameter Antriebsparameter schreiben	188
FB 835 - VMC_HomeInit_LimitSwitch Initialisierung Referenzfahrt auf Endschalter	190
FB 836 - VMC_HomeInit_HomeSwitch Initialisierung Referenzfahrt auf Referenzschalter	192
FB 837 - VMC_HomeInit_ZeroPulse Initialisierung Referenzfahrt auf Nullimpuls	195
FB 838 - VMC_HomeInit_SetPosition Initialisierung Referenzfahrt setze Position	197


3.4.2 Einfache Bewegungsaufgaben

3.4.2.1 UDT 860 - MC_AXIS_REF - Datenstruktur Achsdaten

Dies ist eine benutzerdefinierte Datenstruktur, die Statusinformationen der Achse beinhaltet.

3.4.2.2 FB 860 - VMC_AxisControl - Control-Baustein Achskontrolle

Beschreibung

Mit dem FB `VMC_AxisControl` können Sie die angebundene Achse steuern. Sie können den Status des Antriebs abrufen, den Antrieb ein- bzw. ausschalten oder verschiedene Bewegungskommandos ausführen. In den Instanzdaten des Bausteins befindet sich ein gesonderter Speicherbereich. Über diesen können Sie mittels eines HMI Ihre Achse steuern.  Kapitel 7 "Antrieb über HMI steuern" auf Seite 313



Der Baustein VMC_AxisControl sollte nie gleichzeitig mit dem PLCopen-Baustein MC_Power verwendet werden. Da der VMC_AxisControl Funktionalitäten des MC_Power beinhaltet und immer der aktuellste Befehl vom VMC_Kernel-Baustein ausgeführt wird, kann dies zu einem Fehlverhalten des Antriebs führen.

Parameter

Parameter	Deklaration	Datentyp	Beschreibung
AxisEnable	INPUT	BOOL	<ul style="list-style-type: none"> ■ Achsenfreigabe <ul style="list-style-type: none"> – TRUE: Die Achse wird freigegeben. – FALSE: Die Achse wird gesperrt.
AxisReset	INPUT	BOOL	<ul style="list-style-type: none"> ■ Reset Achse <ul style="list-style-type: none"> – Flanke 0-1: Reset der Achse wird durchgeführt.
HomeExecute	INPUT	BOOL	<ul style="list-style-type: none"> ■ Referenzfahrt <ul style="list-style-type: none"> – Flanke 0-1: Referenzfahrt wird gestartet.
HomePosition	INPUT	REAL	Bei erfolgreicher Referenzierung wird die Istposition der Achse einmalig gleich Position gesetzt. Die Position ist in der verwendeten Anwendereinheit anzugeben.
StopExecute	INPUT	BOOL	<ul style="list-style-type: none"> ■ Achse stoppen <ul style="list-style-type: none"> – Flanke 0-1: Stoppen der Achse wird gestartet.
MvVelocityExecute	INPUT	BOOL	<ul style="list-style-type: none"> ■ Verfahren der Achse starten <ul style="list-style-type: none"> – Flanke 0-1: Die Achse wird auf die angegebene Geschwindigkeit beschleunigt / abgebremst.
MvRelativeExecute	INPUT	BOOL	<ul style="list-style-type: none"> ■ Verfahren der Achse starten <ul style="list-style-type: none"> – Flanke 0-1: Die relative Positionierung der Achse wird gestartet.
MvAbsoluteExecute	INPUT	BOOL	<ul style="list-style-type: none"> ■ Verfahren der Achse starten <ul style="list-style-type: none"> – Flanke 0-1: Die absolute Positionierung der Achse wird gestartet.
Direction*	INPUT	BYTE	Modus für absolute Positionierung: <ul style="list-style-type: none"> ■ 0: kürzester Weg ■ 1: positive Richtung ■ 2: negative Richtung ■ 3: aktuelle Richtung

Parameter	Deklaration	Datentyp	Beschreibung
PositionDistance	INPUT	REAL	Absolute Position bzw. relative Wegstrecke je nach Kommando in [Anwendereinheiten].
Velocity	INPUT	REAL	Geschwindigkeitsvorgabe (vorzeichenbehafteter Wert) in [Anwendereinheiten/s].
Acceleration	INPUT	REAL	Beschleunigung in [Anwendereinheiten/s ²].
Deceleration	INPUT	REAL	Verzögerung in [Anwendereinheiten/s ²].
JogPositive	INPUT	BOOL	<ul style="list-style-type: none"> ■ Achse mit konstanter Geschwindigkeit in positive Richtung verfahren <ul style="list-style-type: none"> – Flanke 0-1: Das Verfahren der Achse mit konstanter Geschwindigkeit wird gestartet. – Flanke 1-0: Die Achse wird gestoppt.
JogNegative	INPUT	BOOL	<ul style="list-style-type: none"> ■ Achse mit konstanter Geschwindigkeit in negative Richtung verfahren <ul style="list-style-type: none"> – Flanke 0-1: Das Verfahren der Achse mit konstanter Geschwindigkeit wird gestartet. – Flanke 1-0: Die Achse wird gestoppt.
JogVelocity	INPUT	REAL	Geschwindigkeitsvorgabe für Jogging (positiver Wert) in [Anwendereinheiten/s].
JogAcceleration	INPUT	REAL	Beschleunigung in [Anwendereinheiten/s ²].
JogDeceleration	INPUT	REAL	Verzögerung für Jogging in [Anwendereinheiten/s ²].
AxisReady	OUTPUT	BOOL	<ul style="list-style-type: none"> ■ AxisReady <ul style="list-style-type: none"> – TRUE: Die Achse ist einschaltbereit. – FALSE: Die Achse ist nicht einschaltbereit. <ul style="list-style-type: none"> → Prüfe und behebe <i>AxisError</i> (siehe AxisErrorID). → Prüfe und behebe <i>DriveError</i> (siehe DriveErrorID). → Prüfe Initialisierungs FB (Input- und Output Adressen bzw. PDO Mapping richtig?)
AxisEnabled	OUTPUT	BOOL	<ul style="list-style-type: none"> ■ Status Achse <ul style="list-style-type: none"> – TRUE: Achse ist eingeschaltet und nimmt Bewegungsaufträge an. – FALSE: Achse ist nicht eingeschaltet und nimmt keine Bewegungsaufträge an.
AxisError	OUTPUT	BOOL	<ul style="list-style-type: none"> ■ Fehler bei Motion Achse <ul style="list-style-type: none"> – TRUE: Ein Fehler ist aufgetreten. <p>Zusätzliche Fehlerinformationen können dem Parameter <i>AxisErrorID</i> entnommen werden.</p> <p>→ Die Achse wird gesperrt.</p>
AxisErrorID	OUTPUT	WORD	<p>Zusätzliche Fehlerinformationen</p> <p>↳ Kapitel 9 "ErrorID - Zusätzliche Fehlerinformationen" auf Seite 340</p>
DriveWarning	OUTPUT	BOOL	<ul style="list-style-type: none"> ■ Warnung <ul style="list-style-type: none"> – TRUE: Der Antrieb liefert eine Warnung. <p>Zusätzliche Informationen sind aus dem entsprechenden Handbuch des Herstellers zu entnehmen.</p>

Parameter	Deklaration	Datentyp	Beschreibung
DriveError	OUTPUT	BOOL	<ul style="list-style-type: none"> ■ Fehler direkt am Antrieb <ul style="list-style-type: none"> – TRUE: Ein Fehler ist aufgetreten. <p>Zusätzliche Fehlerinformationen können dem Parameter <i>DriveErrorID</i> entnommen werden.</p> <p>→ Die Achse wird gesperrt.</p>
DriveErrorID	OUTPUT	WORD	<ul style="list-style-type: none"> ■ Fehler <ul style="list-style-type: none"> – TRUE: Der Antrieb liefert einen Fehler. <p>Zusätzliche Informationen sind aus dem entsprechenden Handbuch des Herstellers zu entnehmen.</p>
IsHomed	OUTPUT	BOOL	<ul style="list-style-type: none"> ■ Information Achse: referenziert <ul style="list-style-type: none"> – TRUE: Die Achse ist referenziert.
ModeOfOperation	OUTPUT	INT	<p>Antriebspezifischer Modus. Weitere Infos siehe Antriebsmanual.</p> <p>Beispiel <i>Sigma-5</i>:</p> <p>0: No mode changed/no mode assigned 1: Profile Position mode 2: Reserved (keep last mode) 3: Profile Velocity mode 4: Torque Profile mode 6: Homing mode 7: Interpolated Position mode 8: Cyclic Sync Position mode 9: Cyclic Sync Velocity mode 10: Cyclic Sync Torque mode Other Reserved (keep last mode)</p>
PLCopenState	OUTPUT	INT	<p>Aktueller PLCopenState:</p> <p>1: Disabled 2: Standstill 3: Homing 4: Discrete Motion 5: Continuous Motion 7: Stopping 8: Errorstop</p>
ActualPosition	OUTPUT	REAL	Position der Achse in [Anwendereinheit].
ActualVelocity	OUTPUT	REAL	Geschwindigkeit der Achse in [Anwendereinheit/s].
CmdDone	OUTPUT	BOOL	<ul style="list-style-type: none"> ■ Status <ul style="list-style-type: none"> – TRUE: Auftrag wurde ohne Fehler beendet.
CmdBusy	OUTPUT	BOOL	<ul style="list-style-type: none"> ■ Status <ul style="list-style-type: none"> – TRUE: Auftrag ist in Bearbeitung.

Parameter	Deklaration	Datentyp	Beschreibung
CmdAborted	OUTPUT	BOOL	<ul style="list-style-type: none"> ■ Status <ul style="list-style-type: none"> – TRUE: Der Auftrag wurde während der Bearbeitung von einem anderen Auftrag abgebrochen.
CmdError	OUTPUT	BOOL	<ul style="list-style-type: none"> ■ Status <ul style="list-style-type: none"> – TRUE: Ein Fehler ist aufgetreten. <p>Zusätzliche Fehlerinformationen können dem Parameter <i>CmdErrorID</i> entnommen werden.</p>
CmdErrorID	OUTPUT	WORD	<p>Zusätzliche Fehlerinformationen</p> <p>🔗 <i>Kapitel 9 "ErrorID - Zusätzliche Fehlerinformationen" auf Seite 340</i></p>
DirectionPositive	OUTPUT	BOOL	<ul style="list-style-type: none"> ■ Zustand Bewegungsauftrag: Position zunehmend <ul style="list-style-type: none"> – TRUE: Die Position der Achse nimmt zu.
DirectionNegative	OUTPUT	BOOL	<ul style="list-style-type: none"> ■ Zustand Bewegungsauftrag: Position abnehmend <ul style="list-style-type: none"> – TRUE: Die Position der Achse nimmt ab.
SWLimitMinActive	OUTPUT	BOOL	<ul style="list-style-type: none"> ■ Software Endschalter <ul style="list-style-type: none"> – TRUE: Software Endschalter Minimum aktiv (Minimale Position in negative Richtung überschritten).
SWLimitMaxActive	OUTPUT	BOOL	<ul style="list-style-type: none"> ■ Software Endschalter <ul style="list-style-type: none"> – TRUE: Software Endschalter Maximum aktiv (Maximale Position in positive Richtung überschritten).
HWLimitMinActive	OUTPUT	BOOL	<ul style="list-style-type: none"> ■ Hardware Endschalter <ul style="list-style-type: none"> – TRUE: Negativer Hardware Endschalter am Antrieb aktiv (NOT- Negative Overtravel).
HWLimitMaxActive	OUTPUT	BOOL	<ul style="list-style-type: none"> ■ Hardware Endschalter <ul style="list-style-type: none"> – TRUE: Positiver Hardware Endschalter am Antrieb aktiv (POT- Positive Overtravel).
Axis	IN_OUT	MC_AXIS_REF	Referenz zur Achse.

*) Dieser Parameter wird nicht von allen Antrieben unterstützt, z.B. *Sigma 5 über EtherCAT* unterstützt diesen Parameter nicht.

3.4.3 Komplexe Bewegungsaufgaben - PLCopen-Bausteine



Die Ansteuerung eines Pulse Train Antriebs erfolgt ausschließlich mit dem FB 875 VMC_AxisControl_PT. PLCopen-Bausteine werden nicht unterstützt!

3.4.3.1 UDT 860 - MC_AXIS_REF - Datenstruktur Achsdaten

Dies ist eine benutzerdefinierte Datenstruktur, die Statusinformationen der Achse beinhaltet.

3.4.3.2 UDT 861 - MC_TRIGGER_REF - Datenstruktur Triggersignal

Diese ist eine benutzerdefinierte Datenstruktur, die Informationen zum Triggersignal beinhaltet.

3.4.3.3 FB 800 - MC_Power - Achsenfreigabe

Beschreibung Mit MC_Power kann eine Achse freigegeben bzw. gesperrt werden.

Parameter

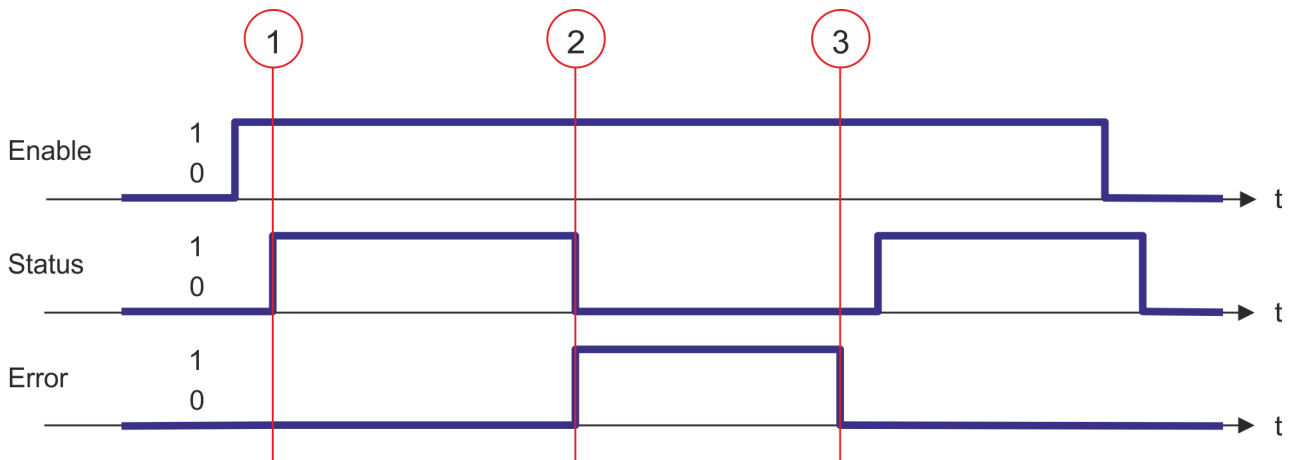
Parameter	Deklaration	Datentyp	Beschreibung
Axis	IN_OUT	MC_AXIS_REF	Referenz zur Achse
Enable	INPUT	BOOL	<ul style="list-style-type: none"> ■ Achsenfreigabe <ul style="list-style-type: none"> – TRUE: Die Achse wird freigegeben – FALSE: Die Achse wird gesperrt
EnablePositive	INPUT	BOOL	Parameter aktuell nicht unterstützt; Aufruf mit FALSE
EnableNegative	INPUT	BOOL	Parameter aktuell nicht unterstützt; Aufruf mit FALSE
Status	OUTPUT	BOOL	<ul style="list-style-type: none"> ■ Status Achse <ul style="list-style-type: none"> – TRUE: Achse nimmt Bewegungsaufträge an – FALSE: Achse nimmt keine Bewegungsaufträge an
Valid	OUTPUT	BOOL	Immer FALSE
Error	OUTPUT	BOOL	<ul style="list-style-type: none"> ■ Fehler <ul style="list-style-type: none"> – TRUE: Ein Fehler ist aufgetreten. Zusätzliche Fehlerinformationen können dem Parameter <i>ErrorID</i> entnommen werden. Die Achse wird gesperrt.
ErrorID	OUTPUT	WORD	Zusätzliche Fehlerinformationen <i>↳ Kapitel 9 "ErrorID - Zusätzliche Fehlerinformationen" auf Seite 340</i>

Achse freigegeben

Aufruf von MC_Power mit *Enable* = TRUE. Sobald *Status* den Wert TRUE zeigt, ist die Achse freigegeben. In diesem Zustand können Bewegungsaufträge aktiviert werden.

Achse sperren

Aufruf von MC_Power mit *Enable* = FALSE. Sobald *Status* den Wert FALSE zeigt, ist die Achse gesperrt. Bei Sperren der Achse wird ein ggf. aktiver Bewegungsauftrag abgebrochen und die Achse gestoppt.

Zustandsdiagramm der Bausteinparameter

- (1) Die Achse wird mit *Enable* = TRUE freigegeben. Zum Zeitpunkt (1) ist die Freigabe erfolgt. Anschließend können Bewegungsaufträge aktiviert werden.
- (2) Zum Zeitpunkt (2) tritt ein Fehler auf, der das Sperren der Achse zur Folge hat. Ein ggf. aktiver Bewegungsauftrag wird abgebrochen und die Achse gestoppt.
- (3) Der Fehler wird beseitigt und zum Zeitpunkt (3) quittiert. Da *Enable* weiterhin gesetzt ist, wird die Achse wieder freigegeben. Zuletzt wird die Achse mit *Enable* = FALSE gesperrt.

3.4.3.4 FB 801 - MC_Home - Achse referenzieren

Beschreibung

Mit MC_Home kann eine Achse referenziert werden. Dadurch kann ein Bezug zwischen der Position der Achse und der mechanischen Stellung hergestellt werden. Die Referenzfahrt-Methode und die zugehörigen Parameter müssen Sie direkt am Antrieb konfigurieren. Verwenden Sie hierzu die VMC_HomeInit... Bausteine.

Parameter

Parameter	Deklaration	Datentyp	Beschreibung
Axis	IN_OUT	MC_AXIS_REF	Referenz zur Achse
Execute	INPUT	BOOL	<ul style="list-style-type: none"> Referenzfahrt <ul style="list-style-type: none"> Flanke 0-1: Referenzfahrt wird gestartet
Position	INPUT	REAL	<p>Bei erfolgreicher Referenzierung wird die Istposition der Achse einmalig gleich <i>Position</i> gesetzt.</p> <p><i>Position</i> ist in der verwendeten Anwendereinheit anzugeben.</p>
BufferMode	INPUT	BYTE	Parameter aktuell nicht unterstützt; Aufruf mit B#16#0
Done	OUTPUT	BOOL	<ul style="list-style-type: none"> Status <ul style="list-style-type: none"> TRUE: Auftrag erfolgreich durchgeführt
Busy	OUTPUT	BOOL	<ul style="list-style-type: none"> Status <ul style="list-style-type: none"> TRUE: Auftrag ist in Bearbeitung
CommandAborted	OUTPUT	BOOL	<ul style="list-style-type: none"> Status <ul style="list-style-type: none"> TRUE: Der Auftrag wurde während der Bearbeitung von einem anderen Auftrag abgebrochen.
Error	OUTPUT	BOOL	<ul style="list-style-type: none"> Status <ul style="list-style-type: none"> TRUE: Ein Fehler ist aufgetreten. Zusätzliche Fehlerinformationen können dem Parameter <i>ErrorID</i> entnommen werden.
ErrorID	OUTPUT	WORD	<p>Zusätzliche Fehlerinformationen</p> <p>🔗 Kapitel 9 "ErrorID - Zusätzliche Fehlerinformationen" auf Seite 340</p>

PLCopen-State

Start des Auftrags nur im PLCopen-State *Standstill* möglich.

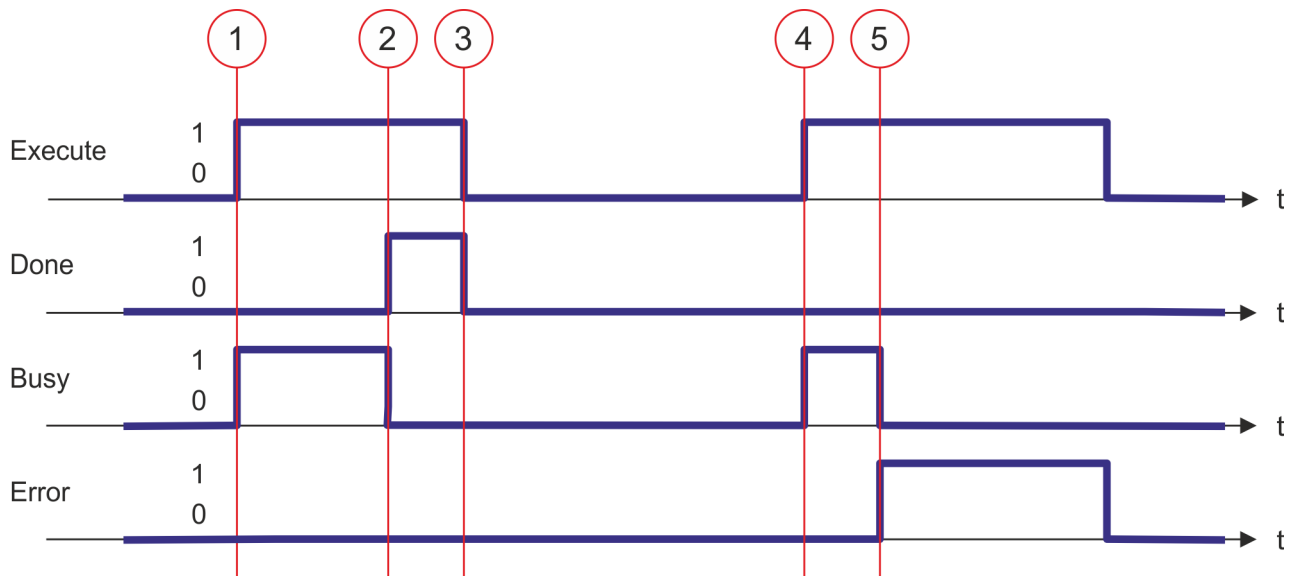
Achse referenzieren

Mit einer Flanke 0-1 an *Execute* wird die Referenzierung gestartet. Solange die Referenzierung läuft zeigt *Busy* den Wert TRUE. Sobald *Done* den Wert TRUE hat, ist die Referenzierung erfolgreich abgeschlossen. Die Istposition der Achse wurde auf den Wert von *Position* gesetzt.



- Ein laufender Auftrag wird auch beim Setzen von *Execute* gleich FALSE weiterhin ausgeführt.
- Ein laufender Auftrag kann durch einen Bewegungsauftrag (z.B. MC_MoveRelative) nicht abgebrochen werden.

Zustandsdiagramm der Bausteinparameter



- (1) Mit Flanke 0-1 an *Execute* zum Zeitpunkt (1) wird die Referenzierung gestartet und *Busy* liefert den Wert TRUE.
- (2) Zum Zeitpunkt (2) ist die Referenzierung abgeschlossen. *Busy* liefert den Wert FALSE und *Done* den Wert TRUE.
- (3) Zum Zeitpunkt (3) ist der Auftrag abgeschlossen und *Execute* wird gleich FALSE gesetzt und dadurch sämtliche Ausgangsparameter auf FALSE bzw. 0 gesetzt.
- (4) Zum Zeitpunkt (4) wird erneut die Referenzierung mit einer Flanke 0-1 an *Execute* gestartet und *Busy* liefert den Wert TRUE.
- (5) Zum Zeitpunkt (5) tritt ein Fehler bei der Referenzierung auf. *Busy* liefert den Wert FALSE und *Error* den Wert TRUE.

3.4.3.5 FB 802 - MC_Stop - Achse stoppen

Beschreibung Mit MC_Stop wird die Achse gestoppt. Mit dem Parameter *Deceleration* kann das dynamische Verhalten beim Stoppvorgang bestimmt werden.

Parameter

Parameter	Deklaration	Datentyp	Beschreibung
Axis	IN_OUT	MC_AXIS_REF	Referenz zur Achse
Execute	INPUT	BOOL	<ul style="list-style-type: none"> Achse stoppen <ul style="list-style-type: none"> Flanke 0-1: Stoppen der Achse wird gestartet
Deceleration	INPUT	REAL	Verzögerung beim Stoppen in [Anwendereinheiten/s ²]
Jerk	INPUT	REAL	Parameter aktuell nicht unterstützt; Aufruf mit 0.0
Done	OUTPUT	BOOL	<ul style="list-style-type: none"> Status <ul style="list-style-type: none"> TRUE: Auftrag erfolgreich durchgeführt
Busy	OUTPUT	BOOL	<ul style="list-style-type: none"> Status <ul style="list-style-type: none"> TRUE: Auftrag ist in Bearbeitung
CommandAborted	OUTPUT	BOOL	<ul style="list-style-type: none"> Status <ul style="list-style-type: none"> TRUE: Der Auftrag wurde während der Bearbeitung von einem anderen Auftrag abgebrochen
Error	OUTPUT	BOOL	<ul style="list-style-type: none"> Status <ul style="list-style-type: none"> TRUE: Ein Fehler ist aufgetreten. Zusätzliche Fehlerinformationen können dem Parameter <i>ErrorID</i> entnommen werden.
ErrorID	OUTPUT	WORD	Zusätzliche Fehlerinformationen <i>☞ Kapitel 9 "ErrorID - Zusätzliche Fehlerinformationen" auf Seite 340</i>

PLCopen-State

- Start des Auftrags in den PLCopen-States *Standstill*, *Homing*, *Discrete Motion* und *Continuous Motion* möglich.
- MC_Stop führt die Achse in den PLCopen-State *Stopping* über. In *Stopping* können keine Bewegungsaufträge gestartet werden. Solange *Execute* gleich TRUE ist, bleibt die Achse im PLCopen-State *Stopping*. Wird *Execute* gleich FALSE gesetzt, geht die Achse in den PLCopen-State *Standstill* über. In *Standstill* können Bewegungsaufträge gestartet werden.

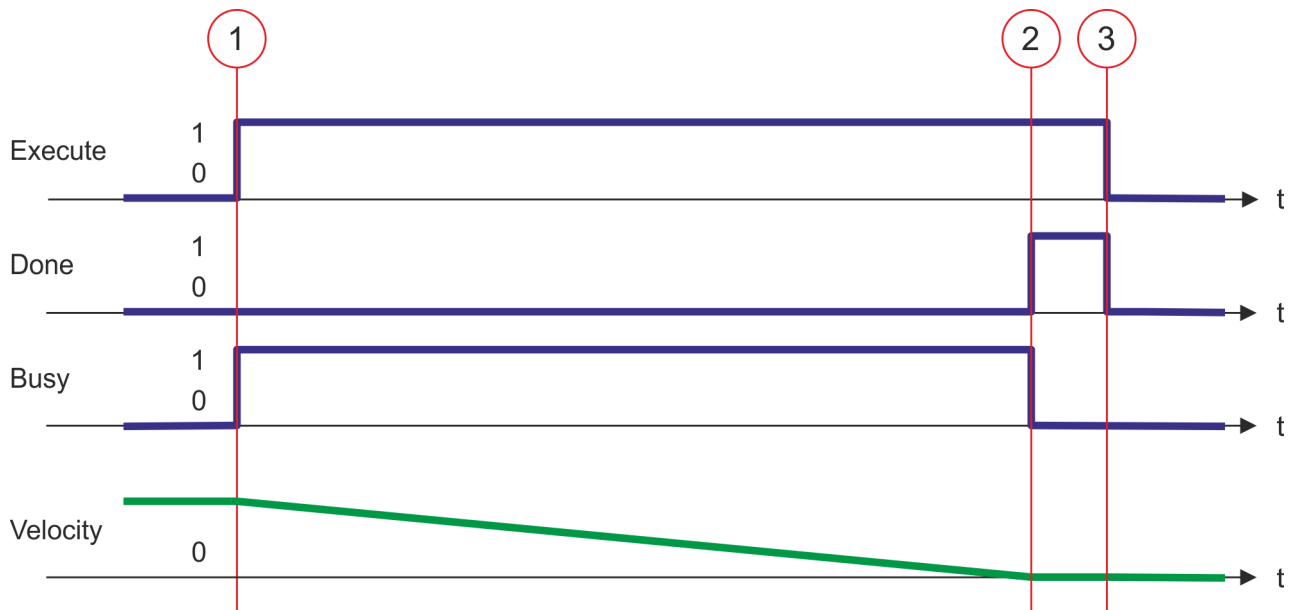
Achse stoppen

Mit einer Flanke 0-1 an *Execute* wird das Stoppen der Achse gestartet. Solange das Stoppen der Achse läuft, zeigt *Busy* den Wert TRUE. Nachdem die Achse gestoppt wurde und somit die Geschwindigkeit 0 erreicht hat, wird *Busy* gleich FALSE und *Done* gleich TRUE geliefert.



- Ein laufender Auftrag wird auch beim Setzen von *Execute* gleich FALSE bis zum Stopp der Achse ausgeführt.
- Ein laufender Auftrag kann durch einen Bewegungsauftrag (z.B. MC_MoveRelative) nicht abgebrochen werden.

Zustandsdiagramm der Bausteinparameter



- (1) Mit der Flanke 0-1 an *Execute* zum Zeitpunkt (1) wird das Stoppen der Achse gestartet und *Busy* liefert den Wert TRUE. Die Geschwindigkeit der Achse wird unter Berücksichtigung des Parameters *Deceleration* bis auf null verringert.
- (2) Zum Zeitpunkt (2) ist das Stoppen der Achse abgeschlossen, die Achse ist gestoppt. *Busy* liefert den Wert FALSE und *Done* den Wert TRUE.
- (3) Zum Zeitpunkt (3) ist der Auftrag abgeschlossen und *Execute* wird gleich FALSE gesetzt und dadurch sämtliche Ausgangsparameter auf FALSE bzw. 0 gesetzt.

3.4.3.6 FB 803 - MC_Halt - Achse anhalten

Beschreibung

Mit MC_Halt wird die Achse bis zum Stillstand abgebremst. Mit dem Parameter *Deceleration* kann das dynamische Verhalten beim Bremsvorgang bestimmt werden.

Parameter

Parameter	Deklaration	Datentyp	Beschreibung
Axis	IN_OUT	MC_AXIS_REF	Referenz zur Achse
Execute	INPUT	BOOL	<ul style="list-style-type: none"> ■ Achse anhalten <ul style="list-style-type: none"> – Flanke 0-1: Anhalten der Achse wird gestartet
Deceleration	INPUT	REAL	Verzögerung beim Bremsen in [Anwendereinheiten/s ²]
Jerk	INPUT	REAL	Parameter aktuell nicht unterstützt; Aufruf mit 0.0
BufferMode	INPUT	BYTE	Parameter aktuell nicht unterstützt; Aufruf mit B#16#0
Done	OUTPUT	BOOL	<ul style="list-style-type: none"> ■ Status <ul style="list-style-type: none"> – TRUE: Auftrag erfolgreich durchgeführt
Busy	OUTPUT	BOOL	<ul style="list-style-type: none"> ■ Status <ul style="list-style-type: none"> – TRUE: Auftrag ist in Bearbeitung
Active	OUTPUT	BOOL	<ul style="list-style-type: none"> ■ Status <ul style="list-style-type: none"> – TRUE: Baustein hat die Kontrolle über die Achse
CommandAborted	OUTPUT	BOOL	<ul style="list-style-type: none"> ■ Status <ul style="list-style-type: none"> – TRUE: Der Auftrag wurde während der Bearbeitung von einem anderen Auftrag abgebrochen
Error	OUTPUT	BOOL	<ul style="list-style-type: none"> ■ Status <ul style="list-style-type: none"> – TRUE: Ein Fehler ist aufgetreten. Zusätzliche Fehlerinformationen können dem Parameter <i>ErrorID</i> entnommen werden.
ErrorID	OUTPUT	WORD	Zusätzliche Fehlerinformationen <i>🔗 Kapitel 9 "ErrorID - Zusätzliche Fehlerinformationen" auf Seite 340</i>

PLCopen-State

- Start des Auftrags in den PLCopen-States *Discrete Motion* und *Continuous Motion* möglich.
- MC_Halt führt die Achse in den PLCopen-State *Discrete Motion* über.

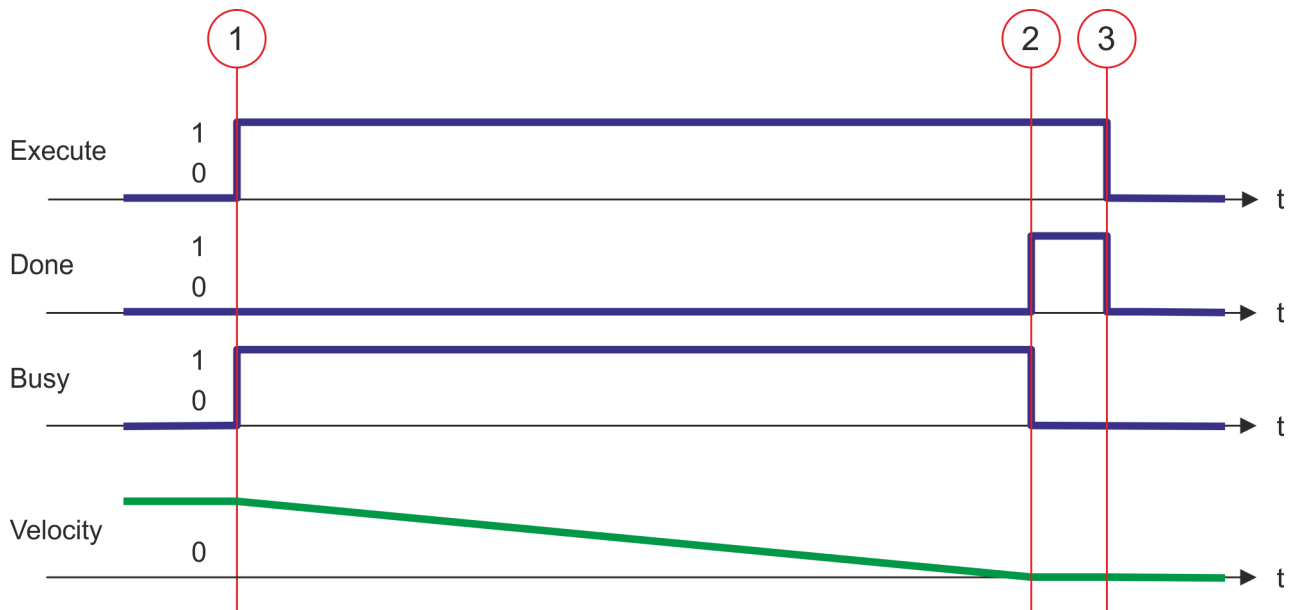
Achse anhalten

Mit einer Flanke 0-1 an *Execute* wird das Anhalten der Achse gestartet. Solange das Anhalten der Achse läuft, zeigt *Busy* den Wert TRUE. Nachdem die Achse angehalten wurde und somit die Geschwindigkeit 0 erreicht hat, wird *Busy* gleich FALSE und *Done* gleich TRUE geliefert.



- Ein laufender Auftrag wird auch beim Setzen von *Execute* gleich FALSE bis zum Anhalten der Achse ausgeführt.
- Ein laufender Auftrag kann durch einen anderen Bewegungsauftrag (z.B. MC_MoveRelative) abgebrochen werden.

Zustandsdiagramm der Bausteinparameter




- (1) Mit der Flanke 0-1 an *Execute* zum Zeitpunkt (1) wird das Anhalten der Achse gestartet und *Busy* liefert den Wert TRUE. Die Geschwindigkeit der Achse wird unter Berücksichtigung des Parameters *Deceleration* bis auf 0 verringert.
- (2) Zum Zeitpunkt (2) ist das Anhalten der Achse abgeschlossen, die Achse steht. *Busy* liefert den Wert FALSE und *Done* den Wert TRUE.
- (3) Zum Zeitpunkt (3) ist der Auftrag abgeschlossen und *Execute* wird gleich FALSE gesetzt und dadurch sämtliche Ausgangsparameter auf FALSE bzw. 0 gesetzt.

3.4.3.7 FB 804 - MC_MoveRelative - Achse relativ verfahren

Beschreibung

Mit MC_MoveRelative wird die Achse relativ zu der Position bei Auftragsstart um eine spezifizierte Distanz verfahren. Mit den Parametern *Velocity*, *Acceleration* und *Deceleration* wird das dynamische Verhalten beim Bewegungsvorgang bestimmt.

Parameter

Parameter	Deklaration	Datentyp	Beschreibung
Axis	IN_OUT	MC_AXIS_REF	Referenz zur Achse
Execute	INPUT	BOOL	<ul style="list-style-type: none"> ■ Achse relativ verfahren <ul style="list-style-type: none"> – Flanke 0-1: Das relative Verfahren der Achse wird gestartet
ContinuousUpdate	INPUT	BOOL	Parameter aktuell nicht unterstützt; Aufruf mit FALSE
Distance	INPUT	REAL	Relative Wegstrecke in [Anwendereinheiten]
Velocity	INPUT	REAL	Max. Geschwindigkeit (muss nicht zwingend erreicht werden) in [Anwendereinheiten/s]
Acceleration	INPUT	REAL	Beschleunigung in [Anwendereinheiten/s ²]
Deceleration	INPUT	REAL	Verzögerung in [Anwendereinheiten/s ²]
Jerk	INPUT	REAL	Parameter aktuell nicht unterstützt; Aufruf mit 0.0
BufferMode	INPUT	BYTE	Parameter aktuell nicht unterstützt; Aufruf mit B#16#0
Done	OUTPUT	BOOL	<ul style="list-style-type: none"> ■ Status <ul style="list-style-type: none"> – TRUE: Auftrag erfolgreich durchgeführt; Zielposition erreicht
Busy	OUTPUT	BOOL	<ul style="list-style-type: none"> ■ Status <ul style="list-style-type: none"> – TRUE: Auftrag ist in Bearbeitung
Active	OUTPUT	BOOL	<ul style="list-style-type: none"> ■ Status <ul style="list-style-type: none"> – TRUE: Baustein hat die Kontrolle über die Achse
CommandAborted	OUTPUT	BOOL	<ul style="list-style-type: none"> ■ Status <ul style="list-style-type: none"> – TRUE: Der Auftrag wurde während der Bearbeitung von einem anderen Auftrag abgebrochen
Error	OUTPUT	BOOL	<ul style="list-style-type: none"> ■ Status <ul style="list-style-type: none"> – TRUE: Ein Fehler ist aufgetreten. Zusätzliche Fehlerinformationen können dem Parameter <i>ErrorID</i> entnommen werden.
ErrorID	OUTPUT	WORD	Zusätzliche Fehlerinformationen  Kapitel 9 "ErrorID - Zusätzliche Fehlerinformationen" auf Seite 340

PLCopen-State

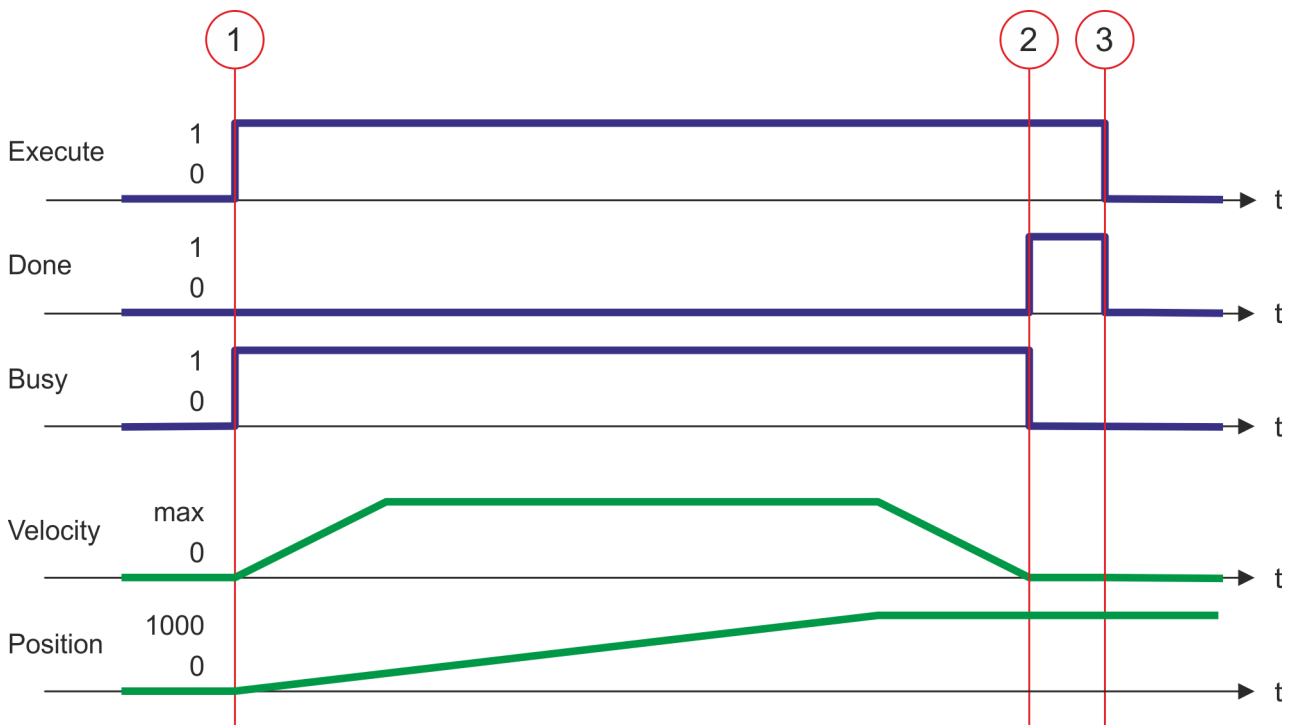
- Start des Auftrags in den PLCopen-States *Standstill*, *Discrete Motion* und *Continuous Motion* möglich.
- MC_MoveRelative führt die Achse in den PLCopen-State *Discrete Motion* über.

Achse relativ verfahren

Mit einer Flanke 0-1 an *Execute* wird das Verfahren der Achse gestartet. Solange das Verfahren der Achse läuft, zeigt *Busy* den Wert TRUE. Nachdem die Zielposition erreicht wurde, wird *Busy* gleich FALSE und *Done* gleich TRUE geliefert. Die Geschwindigkeit der Achse ist dann gleich 0.



- Ein laufender Auftrag wird auch beim Setzen von *Execute* gleich FALSE bis die Achse die Zielposition erreicht hat, ausgeführt.
- Ein laufender Auftrag kann durch einen anderen Bewegungsauftrag (z.B. MC_MoveAbsolute) abgebrochen werden.

Zustandsdiagramm der Bausteinparameter

- (1) Die Achse wird mit MC_MoveRelative um eine *Distance* = 1000.0 verfahren (Startposition bei Auftragsstart gleich 0.0). Mit der Flanke 0-1 an *Execute* zum Zeitpunkt (1) wird das Verfahren der Achse gestartet und *Busy* liefert den Wert TRUE.
- (2) Zum Zeitpunkt (2) wurde die Achse um die *Distance* = 1000.0 verfahren, d.h. die Zielposition wurde erreicht. *Busy* liefert den Wert FALSE und *Done* den Wert TRUE.
- (3) Zum Zeitpunkt (3) ist der Auftrag abgeschlossen und *Execute* wird gleich FALSE gesetzt und dadurch sämtliche Ausgangsparameter auf FALSE bzw. 0 gesetzt.

3.4.3.8 FB 805 - MC_MoveVelocity - Achse verfahren mit konstanter Geschwindigkeit

Beschreibung

Mit MC_MoveVelocity wird die Achse mit einer konstanten Geschwindigkeit verfahren. Mit den Parametern *Velocity*, *Acceleration* und *Deceleration* wird das dynamische Verhalten beim Bewegungsvorgang bestimmt.

Parameter

Parameter	Deklaration	Datentyp	Beschreibung
Axis	IN_OUT	MC_AXIS_REF	Referenz zur Achse
Execute	INPUT	BOOL	<ul style="list-style-type: none"> Achse mit konstanter Geschwindigkeit verfahren <ul style="list-style-type: none"> Flanke 0-1: Das Verfahren der Achse mit konstanter Geschwindigkeit wird gestartet
ContinuousUpdate	INPUT	BOOL	Parameter aktuell nicht unterstützt; Aufruf mit FALSE
Velocity	INPUT	REAL	Geschwindigkeitsvorgabe (vorzeichenbehafteter Wert) in [Anwendereinheiten/s]
Acceleration	INPUT	REAL	Beschleunigung in [Anwendereinheiten/s ²]
Deceleration	INPUT	REAL	Verzögerung in [Anwendereinheiten/s ²]
Jerk	INPUT	REAL	Parameter aktuell nicht unterstützt; Aufruf mit 0.0
BufferMode	INPUT	BYTE	Parameter aktuell nicht unterstützt; Aufruf mit B#16#0
InVelocity	OUTPUT	BOOL	<ul style="list-style-type: none"> Geschwindigkeitsvorgabe <ul style="list-style-type: none"> TRUE: Geschwindigkeitsvorgabe erreicht
Busy	OUTPUT	BOOL	<ul style="list-style-type: none"> Status <ul style="list-style-type: none"> TRUE: Auftrag ist in Bearbeitung
Active	OUTPUT	BOOL	<ul style="list-style-type: none"> Status <ul style="list-style-type: none"> TRUE: Baustein hat die Kontrolle über die Achse
CommandAborted	OUTPUT	BOOL	<ul style="list-style-type: none"> Status <ul style="list-style-type: none"> TRUE: Der Auftrag wurde während der Bearbeitung von einem anderen Auftrag abgebrochen
Error	OUTPUT	BOOL	<ul style="list-style-type: none"> Status <ul style="list-style-type: none"> TRUE: Ein Fehler ist aufgetreten. Zusätzliche Fehlerinformationen können dem Parameter <i>ErrorID</i> entnommen werden.
ErrorID	OUTPUT	WORD	Zusätzliche Fehlerinformationen <i>↳ Kapitel 9 "ErrorID - Zusätzliche Fehlerinformationen" auf Seite 340</i>

PLCopen-State

- Start des Auftrags in den PLCopen-States *Standstill*, *Discrete Motion* und *Continuous Motion* möglich.
- MC_MoveVelocity führt die Achse in den PLCopen-State *Continuous Motion* über.

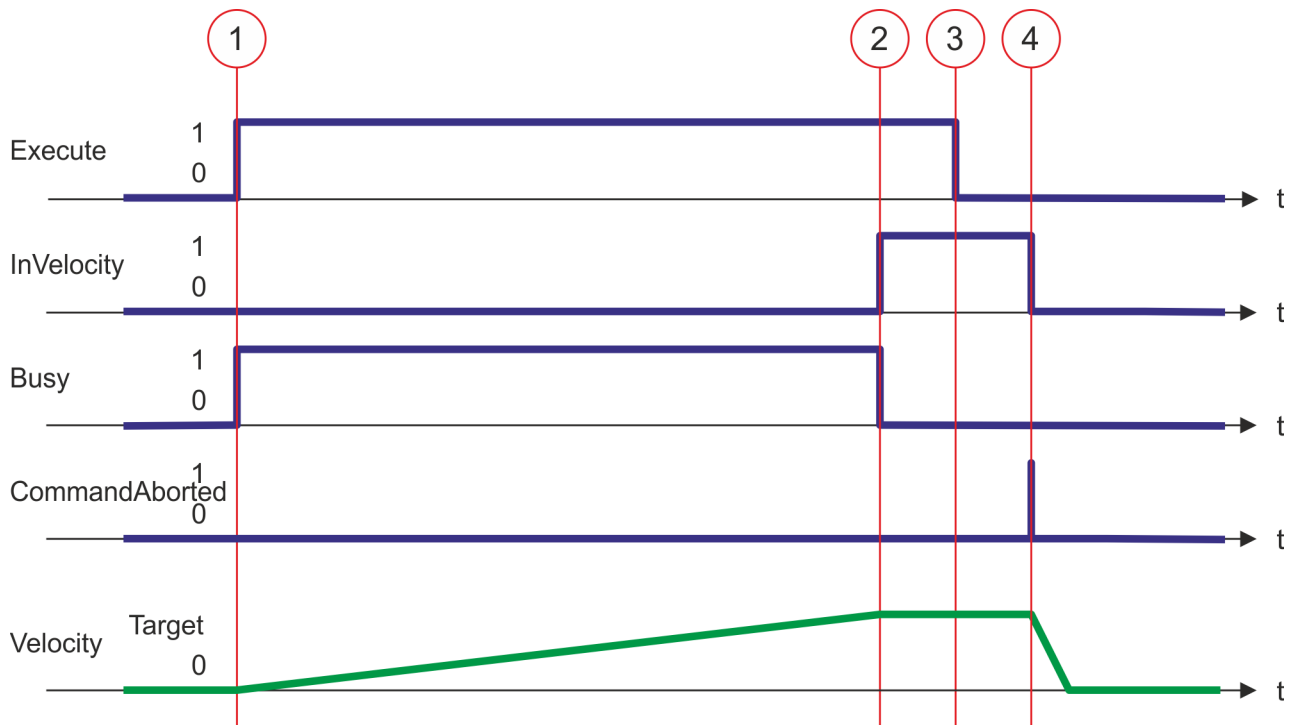
Achse mit Geschwindigkeitsvorgabe verfahren

Mit einer Flanke 0-1 an *Execute* wird das Verfahren der Achse mit Geschwindigkeitsvorgabe gestartet. Solange die Geschwindigkeitsvorgabe nicht erreicht ist, zeigt *Busy* den Wert TRUE und *InVelocity* den Wert FALSE. Ist die Geschwindigkeitsvorgabe erreicht, wird *Busy* gleich FALSE und *InVelocity* gleich TRUE. Die Achse wird mit dieser Geschwindigkeit konstant weiter verfahren.



- Ein laufender Auftrag wird auch beim Setzen von Execute gleich FALSE weiterhin ausgeführt, auch wenn die Geschwindigkeitsvorgabe erreicht wurde.
- Ein laufender Auftrag kann durch einen anderen Bewegungsauftrag (z.B. MC_MoveAbsolute) abgebrochen werden.

Zustandsdiagramm der Bausteinparameter



- (1) Mit der Flanke 0-1 an Execute zum Zeitpunkt (1) wird das Verfahren der Achse mit Geschwindigkeitsvorgabe gestartet und *Busy* liefert den Wert TRUE.
- (2) Zum Zeitpunkt (2) erreicht die Achse die Geschwindigkeitsvorgabe und *Busy* liefert den Wert FALSE und *InVelocity* den Wert TRUE.
- (3) Das Rücksetzen von Execute auf FALSE zum Zeitpunkt (3) hat keine Auswirkung auf die Achse. Die Achse wird weiterhin konstant mit der Geschwindigkeitsvorgabe verfahren und *InVelocity* liefert weiterhin den Wert TRUE.
- (4) Zum Zeitpunkt (4) wird der MC_Velocity-Auftrag durch einen MC_Halt-Auftrag abgebrochen. Die Achse wird bis zum Halt abgebremst.

3.4.3.9 FB 808 - MC_MoveAbsolute - Achse auf absolute Position verfahren

Beschreibung Mit MC_MoveAbsolute wird die Achse auf eine absolute Position verfahren. Mit den Parametern *Velocity*, *Acceleration* und *Deceleration* wird das dynamische Verhalten beim Bewegungsvorgang bestimmt.

Parameter

Parameter	Deklaration	Datentyp	Beschreibung
Axis	IN_OUT	MC_AXIS_REF	Referenz zur Achse
Execute	INPUT	BOOL	<ul style="list-style-type: none"> ■ Verfahren der Achse starten <ul style="list-style-type: none"> – Flanke 0-1: Das Verfahren der Achse wird gestartet
ContinuousUpdate	INPUT	BOOL	Parameter aktuell nicht unterstützt; Aufruf mit FALSE
Position	INPUT	REAL	Absolute Position in [Anwendereinheiten]
Velocity	INPUT	REAL	Maximale Geschwindigkeit (muss nicht zwingend erreicht werden) vorzeichenbehafteter Wert in [Anwendereinheiten/s]
Acceleration	INPUT	REAL	Beschleunigung in [Anwendereinheiten/s ²]
Deceleration	INPUT	REAL	Verzögerung in [Anwendereinheiten/s ²]
Jerk	INPUT	REAL	Parameter aktuell nicht unterstützt; Aufruf mit 0.0
Direction	INPUT	Byte	<ul style="list-style-type: none"> ■ Richtung <ul style="list-style-type: none"> – 0: Kürzeste Entfernung – 1: Positive Richtung – 2: Negative Richtung – 3: Aktuelle Richtung
BufferMode	INPUT	BYTE	Parameter aktuell nicht unterstützt; Aufruf mit B#16#0
Done	OUTPUT	BOOL	<ul style="list-style-type: none"> ■ Status <ul style="list-style-type: none"> – TRUE: Auftrag erfolgreich durchgeführt. Die Zielposition wurde erreicht.
Busy	OUTPUT	BOOL	<ul style="list-style-type: none"> ■ Status <ul style="list-style-type: none"> – TRUE: Auftrag ist in Bearbeitung
Active	OUTPUT	BOOL	<ul style="list-style-type: none"> ■ Status <ul style="list-style-type: none"> – TRUE: Baustein hat die Kontrolle über die Achse
CommandAborted	OUTPUT	BOOL	<ul style="list-style-type: none"> ■ Status <ul style="list-style-type: none"> – TRUE: Der Auftrag wurde während der Bearbeitung von einem anderen Auftrag abgebrochen
Error	OUTPUT	BOOL	<ul style="list-style-type: none"> ■ Status <ul style="list-style-type: none"> – TRUE: Ein Fehler ist aufgetreten. Zusätzliche Fehlerinformationen können dem Parameter <i>ErrorID</i> entnommen werden.
ErrorID	OUTPUT	WORD	Zusätzliche Fehlerinformationen <i>↳ Kapitel 9 "ErrorID - Zusätzliche Fehlerinformationen" auf Seite 340</i>

PLCopen-State

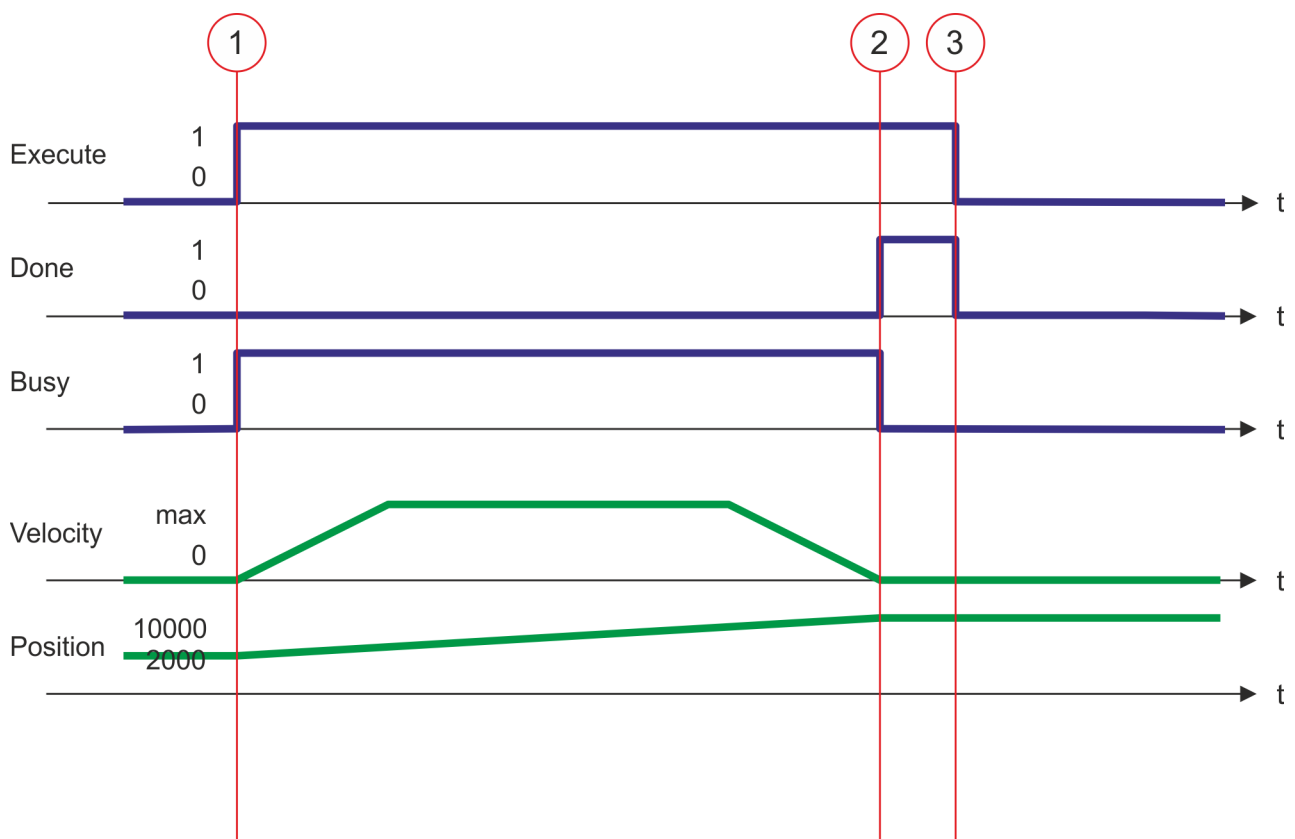
- Start des Auftrags in den PLCopen-States *Standstill*, *Discrete Motion* und *Continuous Motion* möglich.
- MC_MoveVelocity führt die Achse in den PLCopen-State *Discrete Motion* über.

Achse absolute verfahren

Mit einer Flanke 0-1 an *Execute* wird das Verfahren der Achse gestartet. Solange das Verfahren der Achse läuft, zeigt *Busy* den Wert TRUE. Nachdem die Zielposition erreicht wurde, wird *Busy* = FALSE und *Done* = TRUE geliefert. Die Geschwindigkeit der Achse ist dann gleich 0.



- Mit Sigma-5 EtherCAT wird die Zielposition immer über den Weg angefahren, welcher am kürzesten ist.
- Ein laufender Auftrag wird auch beim Setzen von *Execute* gleich FALSE, bis die Achse die Zielposition erreicht hat, ausgeführt.
- Ein laufender Auftrag kann durch einen anderen Bewegungsauftrag (z.B. MC_MoveVelocity) abgebrochen werden.

Zustandsdiagramm der Bausteinparameter

- (1) Die Achse wird mit MC_MoveAbsolute auf die absolute Position = 10000.0 verfahren (Startposition bei Auftragsstart gleich 2000.0). Mit der Flanke 0-1 an *Execute* zum Zeitpunkt (1) wird das Verfahren der Achse gestartet und *Busy* liefert den Wert TRUE.
- (2) Zum Zeitpunkt (2) hat die Achse die Zielposition erreicht. *Busy* liefert den Wert FALSE und *Done* den Wert TRUE.
- (3) Zum Zeitpunkt (3) ist der Auftrag abgeschlossen und *Execute* wird gleich FALSE gesetzt und dadurch sämtliche Ausgangsparameter auf FALSE bzw. 0 gesetzt.

3.4.3.10 FB 811 - MC_Reset - Reset Achse

Beschreibung

Mit MC_Reset wird ein Reset (Neuinitialisieren) der Achse durchgeführt. Dabei werden alle internen Fehler der Achse zurückgesetzt.

Parameter

Parameter	Deklaration	Datentyp	Beschreibung
Axis	IN_OUT	MC_AXIS_REF	Referenz zur Achse
Execute	INPUT	BOOL	<ul style="list-style-type: none"> Reset Achse <ul style="list-style-type: none"> Flanke 0-1: Reset der Achse wird durchgeführt
Done	OUTPUT	BOOL	<ul style="list-style-type: none"> Status <ul style="list-style-type: none"> TRUE: Auftrag erfolgreich durchgeführt. Reset wurde durchgeführt
Busy	OUTPUT	BOOL	<ul style="list-style-type: none"> Status <ul style="list-style-type: none"> TRUE: Auftrag ist in Bearbeitung
Error	OUTPUT	BOOL	<ul style="list-style-type: none"> Status <ul style="list-style-type: none"> TRUE: Ein Fehler ist aufgetreten. Zusätzliche Fehlerinformationen können dem Parameter <i>ErrorID</i> entnommen werden.
ErrorID	OUTPUT	WORD	Zusätzliche Fehlerinformationen 🔗 Kapitel 9 "ErrorID - Zusätzliche Fehlerinformationen" auf Seite 340

PLCopen-State

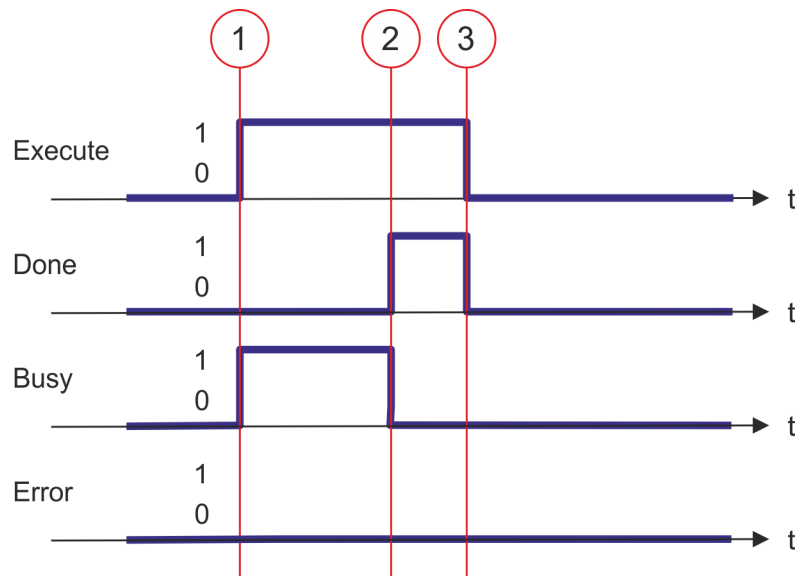
- Start des Auftrags im PLCopen-State *ErrorStop* möglich.
- MC_Reset führt die Achse in Abhängigkeit von MC_Power entweder in den PLCopen-State *Standstill* (Aufruf von MC_Power mit *Enable* = TRUE) oder *Disabled* (Aufruf von MC_Power mit *Enable* = FALSE) über.

Reset an Achse durchführen

Mit einer Flanke 0-1 an *Execute* wird der Reset der Achse gestartet. Solange der Reset der Achse läuft, zeigt *Busy* den Wert TRUE. Nachdem die Achse neu initialisiert wurde, wird *Busy* gleich FALSE und *Done* gleich TRUE geliefert.



Ein laufender Auftrag wird auch beim Setzen von *Execute* gleich FALSE ausgeführt, bis der Auftrag abgeschlossen ist.

Zustandsdiagramm der Bausteinparameter

- (1) Mit der Flanke 0-1 an *Execute* zum Zeitpunkt (1) wird der Reset der Achse gestartet und *Busy* liefert den Wert TRUE.
- (2) Zum Zeitpunkt (2) ist der Reset erfolgreich abgeschlossen. *Busy* liefert den Wert FALSE und *Done* den Wert TRUE.
- (3) Zum Zeitpunkt (3) ist der Auftrag abgeschlossen und *Execute* wird gleich FALSE gesetzt und dadurch sämtliche Ausgangsparameter auf FALSE bzw. 0 gesetzt.

3.4.3.11 FB 812 - MC_ReadStatus - PLCopen Status

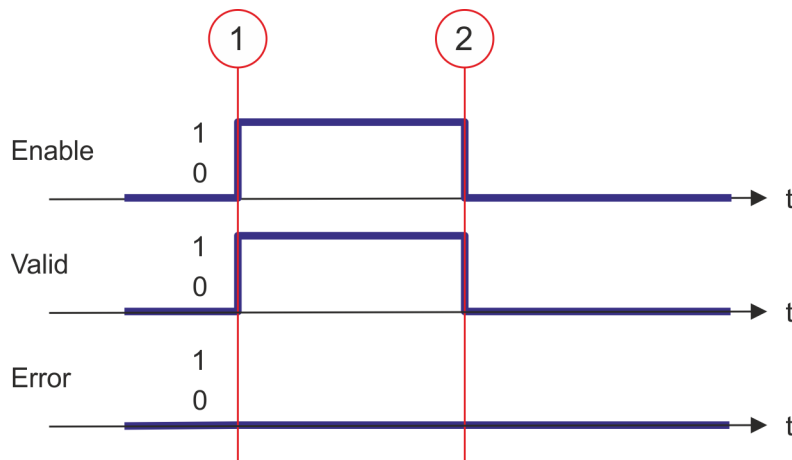
Beschreibung Mit MC_ReadStatus kann der PLCopen-State der Achse ermittelt werden.

Parameter

Parameter	Deklaration	Datentyp	Beschreibung
Axis	IN_OUT	MC_AXIS_REF	Referenz zur Slave-Achse
Enable	INPUT	BOOL	<ul style="list-style-type: none"> ■ Statusanzeige <ul style="list-style-type: none"> – TRUE: Der Status wird an den Ausgängen permanent angezeigt – FALSE: Alle Ausgänge werden gleich FALSE bzw. 0 geliefert
Valid	OUTPUT	BOOL	<ul style="list-style-type: none"> ■ Status gültig <ul style="list-style-type: none"> – TRUE: Der angezeigte Status ist gültig
Error	OUTPUT	BOOL	<ul style="list-style-type: none"> ■ Status <ul style="list-style-type: none"> – TRUE: Ein Fehler ist aufgetreten. Zusätzliche Fehlerinformationen können dem Parameter <i>ErrorID</i> entnommen werden.
ErrorID	OUTPUT	WORD	Zusätzliche Fehlerinformationen <i>↪ Kapitel 9 "ErrorID - Zusätzliche Fehlerinformationen" auf Seite 340</i>
ErrorStop	OUTPUT	BOOL	<ul style="list-style-type: none"> ■ Achsfehler <ul style="list-style-type: none"> – TRUE: Achsfehler aufgetreten; ein Bewegungsauftrag kann nicht aktiviert werden
Disabled	OUTPUT	BOOL	<ul style="list-style-type: none"> ■ Status Achse: Sperrung <ul style="list-style-type: none"> – TRUE: Achse ist gesperrt; ein Bewegungsauftrag kann nicht aktiviert werden
Stopping	OUTPUT	BOOL	<ul style="list-style-type: none"> ■ Status Achse: Stop <ul style="list-style-type: none"> – TRUE: Achse wird gestoppt (MC_Stop ist aktiv)
Homing	OUTPUT	BOOL	<ul style="list-style-type: none"> ■ Status Achse: Referenzierung <ul style="list-style-type: none"> – TRUE: Achse wird referenziert (MC_Homing ist aktiv)
Standstill	OUTPUT	BOOL	<ul style="list-style-type: none"> ■ Status Bewegungsauftrag <ul style="list-style-type: none"> – TRUE: Kein Bewegungsauftrag aktiv; Bewegungsauftrag kann aktiviert werden
DiscreteMotion	OUTPUT	BOOL	<ul style="list-style-type: none"> ■ Status Achsbewegung: Diskret <ul style="list-style-type: none"> – TRUE: Achse wird durch eine diskrete Bewegung verfahren (MC_MoveRelative, MC_MoveAbsolute oder MC_Halt ist aktiv)
ContinuousMotion	OUTPUT	BOOL	<ul style="list-style-type: none"> ■ Status Achsbewegung: Kontinuierlich <ul style="list-style-type: none"> – TRUE: Achse wird durch eine kontinuierliche Bewegung verfahren (MC_MoveVelocity ist aktiv)

PLCopen-State ■ Start des Auftrags in jedem PLCopen-State möglich.

Status der Achse ermitteln Mit *Enable* = TRUE wird an den Ausgängen der Zustand der Achse entsprechend dem Zustandsdiagramm nach PLCopen geliefert.

Zustandsdiagramm der Bausteinparameter

- (1) Zum Zeitpunkt (1) wird *Enable* = TRUE gesetzt. Damit liefert *Valid* den Wert TRUE und an den Ausgängen wird der Zustand entsprechend des PLCopen-Zustandsdiagramms angezeigt.
- (2) Zum Zeitpunkt (2) wird *Enable* = FALSE gesetzt. Damit werden sämtliche Ausgänge gleich FALSE bzw. 0 gesetzt.

3.4.3.12 FB 813 - MC_ReadAxisError - Fehler von Achse lesen

Beschreibung Mit MC_ReadAxisError wird der aktuell anstehende Fehler direkt vom Antrieb gelesen.

Parameter

Parameter	Deklaration	Datentyp	Beschreibung
Axis	IN_OUT	MC_AXIS_REF	Referenz zur Achse
Execute	INPUT	BOOL	<ul style="list-style-type: none"> ■ Reset Achse <ul style="list-style-type: none"> – Flanke 0-1: Achsfehler wird gelesen.
Done	OUTPUT	BOOL	<ul style="list-style-type: none"> ■ Status <ul style="list-style-type: none"> – TRUE: Auftrag erfolgreich durchgeführt. Achsfehler ausgelesen.
Busy	OUTPUT	BOOL	<ul style="list-style-type: none"> ■ Status <ul style="list-style-type: none"> – TRUE: Auftrag ist in Bearbeitung.
Error	OUTPUT	BOOL	<ul style="list-style-type: none"> ■ Status <ul style="list-style-type: none"> – TRUE: Ein Fehler ist aufgetreten. Zusätzliche Fehlerinformationen können dem Parameter <i>ErrorID</i> entnommen werden.
ErrorID	OUTPUT	WORD	Zusätzliche Fehlerinformationen <i>🔗 Kapitel 9 "ErrorID - Zusätzliche Fehlerinformationen" auf Seite 340</i>
AxisErrorID	OUTPUT	WORD	Achsfehler-ID; der gelieferte Wert ist Hersteller-spezifisch kodiert.

PLCopen-State

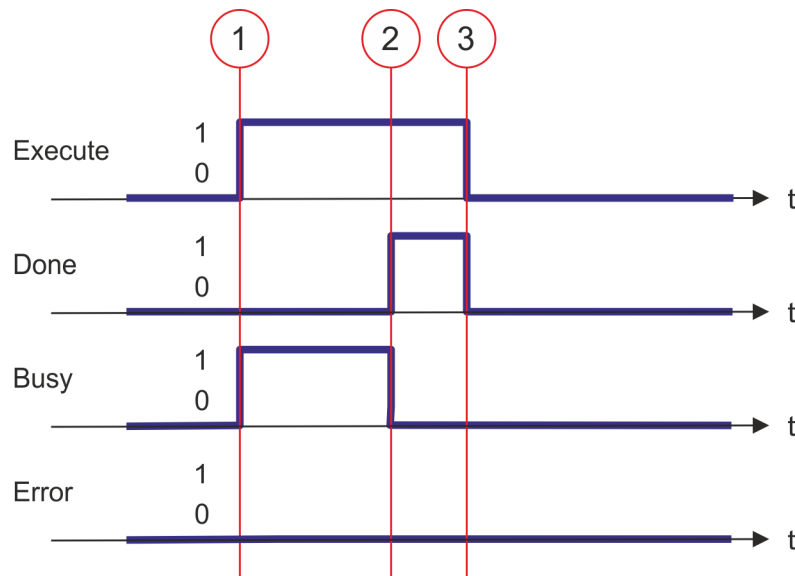
- Start des Auftrags in jedem PLCopen-State möglich.

Fehler der Achse lesen

Mit einer Flanke 0-1 an *Execute* wird das Lesen des Achsfehlers gestartet. Solange das Lesen des Achsfehlers läuft, zeigt *Busy* den Wert TRUE. Nachdem der Achsfehler gelesen wurde, wird *Busy* gleich FALSE und *Done* gleich TRUE geliefert. Der Ausgang *AxisErrorID* zeigt den aktuell anstehenden Achsfehler an.



Ein laufender Auftrag wird auch beim Setzen von Execute gleich FALSE weiterhin ausgeführt.

Zustandsdiagramm der Bausteinparameter

- (1) Mit der Flanke 0-1 an *Execute* zum Zeitpunkt (1) wird das Lesen des Achsfehlers gestartet und *Busy* liefert den Wert TRUE.
- (2) Zum Zeitpunkt (2) ist das Lesen des Achsfehlers erfolgreich abgeschlossen. *Busy* liefert den Wert FALSE und *Done* den Wert TRUE.
- (3) Zum Zeitpunkt (3) ist der Auftrag abgeschlossen und *Execute* wird gleich FALSE gesetzt und dadurch sämtliche Ausgangsparameter auf FALSE bzw. 0 gesetzt.

3.4.3.13 FB 814 - MC_ReadParameter - Parameter der Achse lesen

Beschreibung

Mit MC_ReadParameter wird der Parameter, der über die Parameter-Nummer festgelegt ist, von der Achse gelesen. ↗ *Kapitel 3.4.3.35 "PLCopen Parameter" auf Seite 197*

Parameter

Parameter	Deklaration	Datentyp	Beschreibung
Axis	IN_OUT	MC_AXIS_REF	Referenz zur Achse
Execute	INPUT	BOOL	<ul style="list-style-type: none"> ■ Parameter Achse lesen <ul style="list-style-type: none"> – Flanke 0-1: Das Lesen des Parameters wird durchgeführt
Parameter Number	INPUT	INT	Nummer des Parameters, welcher gelesen werden soll. ↗ <i>Kapitel 3.4.3.35 "PLCopen Parameter" auf Seite 197</i>
Done	OUTPUT	BOOL	<ul style="list-style-type: none"> ■ Status <ul style="list-style-type: none"> – TRUE: Auftrag erfolgreich durchgeführt. Parameter wurde ausgelesen
Busy	OUTPUT	BOOL	<ul style="list-style-type: none"> ■ Status <ul style="list-style-type: none"> – TRUE: Auftrag ist in Bearbeitung
Error	OUTPUT	BOOL	<ul style="list-style-type: none"> ■ Status <ul style="list-style-type: none"> – TRUE: Ein Fehler ist aufgetreten. Zusätzliche Fehlerinformationen können dem Parameter <i>ErrorID</i> entnommen werden.
ErrorID	OUTPUT	WORD	Zusätzliche Fehlerinformationen ↗ <i>Kapitel 9 "ErrorID - Zusätzliche Fehlerinformationen" auf Seite 340</i>
Value	OUTPUT	REAL	Wert des gelesenen Parameters

PLCopen-State

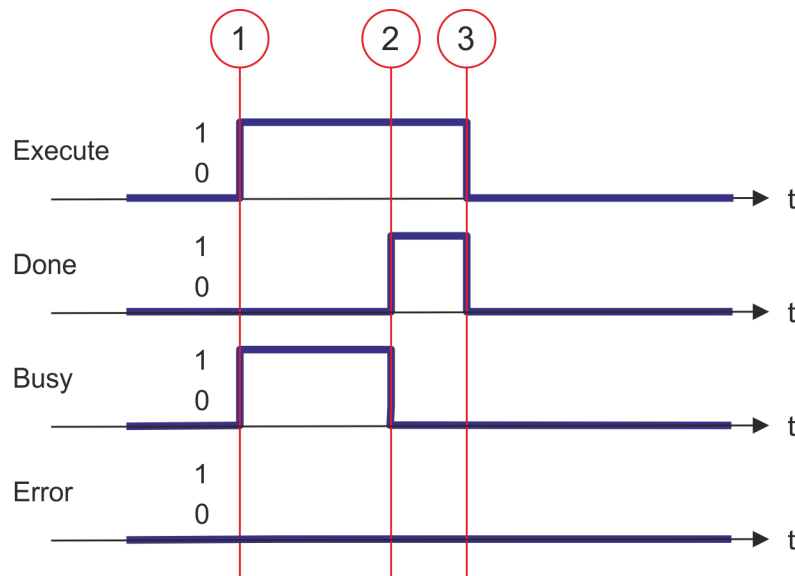
- Start des Auftrags in jedem PLCopen-State möglich.

Parameter der Achse lesen

Mit einer Flanke 0-1 an *Execute* wird das Lesen des Parameters gestartet. Solange das Lesen des Parameters läuft, zeigt *Busy* den Wert TRUE. Nachdem der Parameter gelesen wurde, wird *Busy* gleich FALSE und *Done* gleich TRUE geliefert. Der Ausgang *Value* zeigt den Wert des Parameters an.



Ein laufender Auftrag wird auch beim Setzen von Execute gleich FALSE weiterhin ausgeführt.

Zustandsdiagramm der Bausteinparameter

- (1) Mit der Flanke 0-1 an *Execute* zum Zeitpunkt (1) wird das Lesen des Parameters gestartet und *Busy* liefert den Wert TRUE.
- (2) Zum Zeitpunkt (2) ist das Lesen des Parameters erfolgreich abgeschlossen. *Busy* liefert den Wert FALSE und *Done* den Wert TRUE.
- (3) Zum Zeitpunkt (3) ist der Auftrag abgeschlossen und *Execute* wird gleich FALSE gesetzt und dadurch sämtliche Ausgangsparameter auf FALSE bzw. 0 gesetzt.

3.4.3.14 FB 815 - MC_WriteParameter - Parameter an Achse schreiben

Beschreibung

Mit MC_WriteParameter wird der Wert des Parameters, der über die Parameter-Nummer festgelegt ist, zur Achse geschrieben. ↗ *Kapitel 3.4.3.35 "PLCopen Parameter" auf Seite 197*

Parameter

Parameter	Deklaration	Datentyp	Beschreibung
Axis	IN_OUT	MC_AXIS_REF	Referenz zur Achse
Execute	INPUT	BOOL	<ul style="list-style-type: none"> ■ Parameter Achse schreiben <ul style="list-style-type: none"> – Flanke 0-1: Das Schreiben des Parameters wird durchgeführt
Parameter Number	INPUT	INT	Nummer des Parameters, welcher geschrieben werden soll. ↗ <i>Kapitel 3.4.3.35 "PLCopen Parameter" auf Seite 197</i>
Value	INPUT	REAL	Wert des geschriebenen Parameters
Done	OUTPUT	BOOL	<ul style="list-style-type: none"> ■ Status <ul style="list-style-type: none"> – TRUE: Auftrag erfolgreich durchgeführt. Parameter wurde geschrieben
Busy	OUTPUT	BOOL	<ul style="list-style-type: none"> ■ Status <ul style="list-style-type: none"> – TRUE: Auftrag ist in Bearbeitung
Error	OUTPUT	BOOL	<ul style="list-style-type: none"> ■ Status <ul style="list-style-type: none"> – TRUE: Ein Fehler ist aufgetreten. Zusätzliche Fehlerinformationen können dem Parameter <i>ErrorID</i> entnommen werden.
ErrorID	OUTPUT	WORD	Zusätzliche Fehlerinformationen ↗ <i>Kapitel 9 "ErrorID - Zusätzliche Fehlerinformationen" auf Seite 340</i>

PLCopen-State

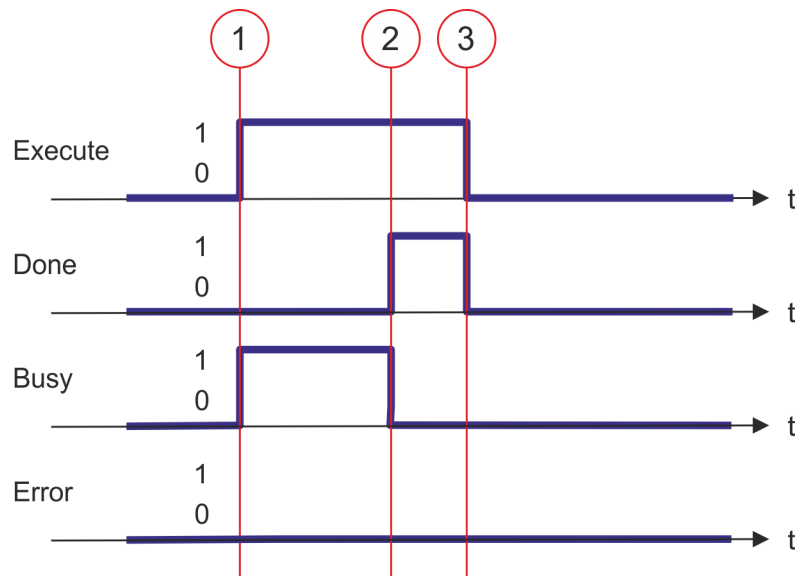
- Start des Auftrags in jedem PLCopen-State möglich.

Parameter der Achse schreiben

Mit einer Flanke 0-1 an *Execute* wird das Schreiben des Parameters gestartet. Solange das Schreiben des Parameters läuft, zeigt *Busy* den Wert TRUE. Nachdem der Parameter geschrieben wurde, wird *Busy* gleich FALSE und *Done* gleich TRUE geliefert.



Ein laufender Auftrag wird auch beim Setzen von Execute gleich FALSE weiterhin ausgeführt.

Zustandsdiagramm der Bausteinparameter

- (1) Mit der Flanke 0-1 an *Execute* zum Zeitpunkt (1) wird das Schreiben des Parameters gestartet und *Busy* liefert den Wert TRUE.
- (2) Zum Zeitpunkt (2) ist das Schreiben des Parameters erfolgreich abgeschlossen. *Busy* liefert den Wert FALSE und *Done* den Wert TRUE.
- (3) Zum Zeitpunkt (3) ist der Auftrag abgeschlossen und *Execute* wird gleich FALSE gesetzt und dadurch sämtliche Ausgangsparameter auf FALSE bzw. 0 gesetzt.

3.4.3.15 FB 816 - MC_ReadActualPosition - Aktuelle Position der Achse lesen

Beschreibung Mit MC_ReadActualPosition wird die aktuelle Position der Achse gelesen.

Parameter

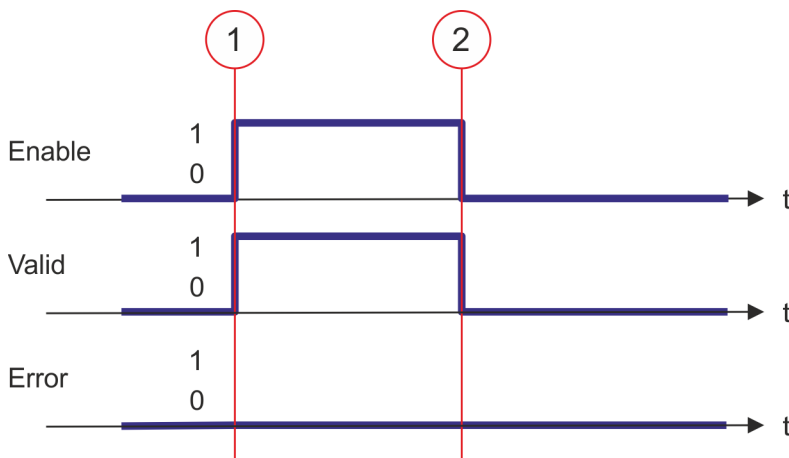
Parameter	Deklaration	Datentyp	Beschreibung
Axis	IN_OUT	MC_AXIS_REF	Referenz zur Achse
Enable	INPUT	BOOL	<ul style="list-style-type: none"> ■ Position Achse lesen <ul style="list-style-type: none"> – TRUE: Die Position der Achse wird kontinuierlich gelesen – FALSE: Alle Ausgänge werden gleich FALSE bzw. 0 geliefert
Valid	OUTPUT	BOOL	<ul style="list-style-type: none"> ■ Position gültig <ul style="list-style-type: none"> – TRUE: Die gelesene Position ist gültig
Error	OUTPUT	BOOL	<ul style="list-style-type: none"> ■ Status <ul style="list-style-type: none"> – TRUE: Ein Fehler ist aufgetreten. Zusätzliche Fehlerinformationen können dem Parameter <i>ErrorID</i> entnommen werden.
ErrorID	OUTPUT	WORD	Zusätzliche Fehlerinformationen <i>↪ Kapitel 9 "ErrorID - Zusätzliche Fehlerinformationen" auf Seite 340</i>
Position	OUTPUT	REAL	Position der Achse in [Anwendereinheit]

PLCopen-State

- Start des Auftrags in jedem PLCopen-State möglich.

Position der Achse lesen

Mit *Enable* gleich TRUE wird die aktuelle Position der Achse ermittelt und unter *Position* abgelegt.

Zustandsdiagramm der Bausteinparameter

- (1) Zum Zeitpunkt (1) wird *Enable* = TRUE gesetzt. Damit liefert *Valid* den Wert TRUE und am Ausgang *Position* wird die aktuelle Position der Achse angezeigt.
- (2) Zum Zeitpunkt (2) wird *Enable* = FALSE gesetzt. Damit werden sämtliche Ausgänge gleich FALSE bzw. 0 gesetzt.

3.4.3.16 FB 817 - MC_ReadActualVelocity - Aktuelle Geschwindigkeit der Achse lesen

Beschreibung Mit MC_ReadActualVelocity wird die aktuelle Geschwindigkeit der Achse gelesen.

Parameter

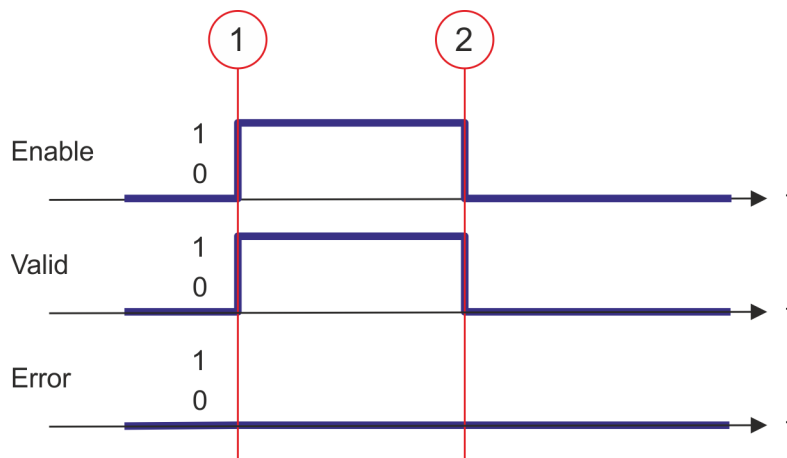
Parameter	Deklaration	Datentyp	Beschreibung
Axis	IN_OUT	MC_AXIS_REF	Referenz zur Achse
Enable	INPUT	BOOL	<ul style="list-style-type: none"> ■ Geschwindigkeit Achse lesen <ul style="list-style-type: none"> – TRUE: Die Geschwindigkeit der Achse wird kontinuierlich gelesen – FALSE: Alle Ausgänge werden gleich FALSE bzw. 0 geliefert
Valid	OUTPUT	BOOL	<ul style="list-style-type: none"> ■ Geschwindigkeit gültig <ul style="list-style-type: none"> – TRUE: Die gelesene Geschwindigkeit ist gültig
Error	OUTPUT	BOOL	<ul style="list-style-type: none"> ■ Status <ul style="list-style-type: none"> – TRUE: Ein Fehler ist aufgetreten. Zusätzliche Fehlerinformationen können dem Parameter <i>ErrorID</i> entnommen werden.
ErrorID	OUTPUT	WORD	Zusätzliche Fehlerinformationen <i>↪ Kapitel 9 "ErrorID - Zusätzliche Fehlerinformationen" auf Seite 340</i>
Velocity	OUTPUT	REAL	Geschwindigkeit der Achse in [Anwendereinheit/s]

PLCopen-State

- Start des Auftrags in jedem PLCopen-State möglich.

Geschwindigkeit der Achse lesen

Mit *Enable* gleich TRUE wird die aktuelle Geschwindigkeit der Achse ermittelt und unter *Velocity* abgelegt.

Zustandsdiagramm der Bausteinparameter

- (1) Zum Zeitpunkt (1) wird *Enable* = TRUE gesetzt. Damit liefert *Valid* den Wert TRUE und am Ausgang *Velocity* wird die aktuelle Geschwindigkeit der Achse angezeigt.
- (2) Zum Zeitpunkt (2) wird *Enable* = FALSE gesetzt. Damit werden sämtliche Ausgänge gleich FALSE bzw. 0 gesetzt.

3.4.3.17 FB 818 - MC_ReadAxisInfo - Zusatzinformationen der Achse lesen

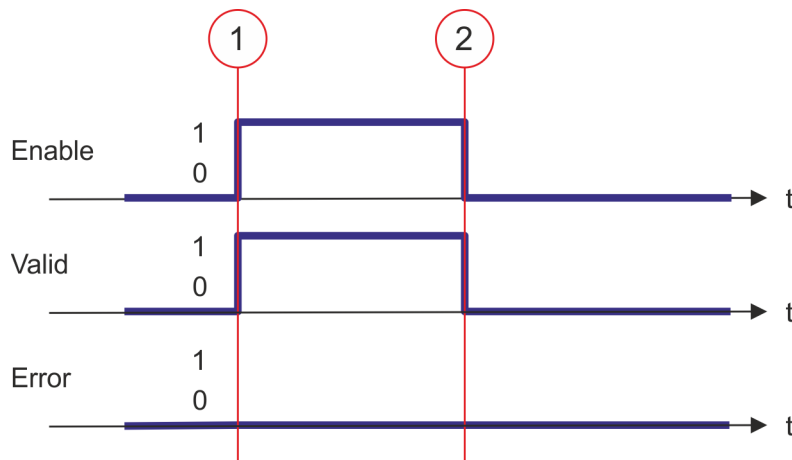
Beschreibung Mit MC_ReadAxisInfo werden einige Zusatzinformationen der Achse angezeigt.

Parameter

Parameter	Deklaration	Datentyp	Beschreibung
Axis	IN_OUT	MC_AXIS_REF	Referenz zur Achse
Enable	INPUT	BOOL	<ul style="list-style-type: none"> ■ Zusatzinformationen Achse lesen <ul style="list-style-type: none"> – TRUE: Die Zusatzinformationen der Achse werden kontinuierlich gelesen – FALSE: Alle Ausgänge werden gleich FALSE bzw. 0 geliefert
Valid	OUTPUT	BOOL	<ul style="list-style-type: none"> ■ Zusatzinformationen gültig <ul style="list-style-type: none"> – TRUE: Die gelesene Zusatzinformationen sind gültig
Error	OUTPUT	BOOL	<ul style="list-style-type: none"> ■ Status <ul style="list-style-type: none"> – TRUE: Ein Fehler ist aufgetreten. Zusätzliche Fehlerinformationen können dem Parameter <i>ErrorID</i> entnommen werden.
ErrorID	OUTPUT	WORD	Zusätzliche Fehlerinformationen <i>↪ Kapitel 9 "ErrorID - Zusätzliche Fehlerinformationen" auf Seite 340</i>
HomeAbsSwitch	OUTPUT	BOOL	Referenzschalter <ul style="list-style-type: none"> ■ TRUE: Der Referenzschalter ist aktiviert
LimitSwitchPos	OUTPUT	BOOL	Endschalter positive Richtung <ul style="list-style-type: none"> ■ TRUE: Endschalter positive Richtung ist aktiviert
LimitSwitchNeg	OUTPUT	BOOL	Endschalter negative Richtung (NOT-Bit am Antrieb) <ul style="list-style-type: none"> ■ TRUE: Endschalter negative Richtung ist aktiviert
Simulation	OUTPUT	BOOL	Parameter aktuell nicht unterstützt; immer FALSE
Communication-Ready	OUTPUT	BOOL	<ul style="list-style-type: none"> ■ Information Achse: Datenaustausch <ul style="list-style-type: none"> – TRUE: Datenaustausch mit der Achse initialisiert; Achse ist kommunikationsbereit
ReadyForPowerOn	OUTPUT	BOOL	<ul style="list-style-type: none"> ■ Information Achse: Freigabe möglich <ul style="list-style-type: none"> – TRUE: Die Freigabe der Achse ist möglich
PowerOn	OUTPUT	BOOL	<ul style="list-style-type: none"> ■ Information Achse: freigegeben <ul style="list-style-type: none"> – TRUE: Die Freigabe der Achse ist erfolgt
IsHomed	OUTPUT	BOOL	<ul style="list-style-type: none"> ■ Information Achse: referenziert <ul style="list-style-type: none"> – TRUE: Die Achse ist referenziert
AxisWarning	OUTPUT	BOOL	<ul style="list-style-type: none"> ■ Information Achse: Fehler <ul style="list-style-type: none"> – TRUE: Mindestens 1 Fehler wird von der Achse gemeldet

PLCopen-State ■ Start des Auftrags in jedem PLCopen-State möglich.

Status der Achse ermitteln Mit *Enable* gleich TRUE werden an den Ausgängen die Zusatzinformationen zur Achse geliefert.

**Zustandsdiagramm der
Bausteinparameter**

- (1) Zum Zeitpunkt (1) wird *Enable* = TRUE gesetzt. Damit liefert *Valid* den Wert TRUE und an den Ausgängen werden die Zusatzinformationen zur Achse angezeigt.
- (2) Zum Zeitpunkt (2) wird *Enable* = FALSE gesetzt. Damit werden sämtliche Ausgänge gleich FALSE bzw. 0 gesetzt.

3.4.3.18 FB 819 - MC_ReadMotionState - Zustand Bewegungsauftrag lesen

Beschreibung Mit MC_ReadMotionState wird der aktuelle Zustand des Bewegungsauftrags angezeigt.

Parameter

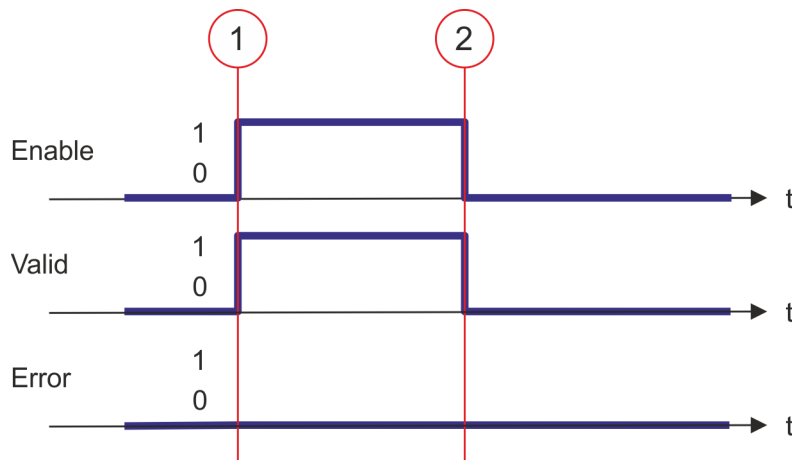
Parameter	Deklaration	Datentyp	Beschreibung
Axis	IN_OUT	MC_AXIS_REF	Referenz zur Achse
Enable	INPUT	BOOL	<ul style="list-style-type: none"> ■ Zustand Bewegungsauftrag lesen <ul style="list-style-type: none"> – TRUE: Zustand Bewegungsauftrag wird kontinuierlich gelesen – FALSE: Alle Ausgänge werden gleich FALSE bzw. 0 geliefert
Source	INPUT	Byte	Nur Source = 0 wird unterstützt; an den Ausgängen werden die Istzustände des Bewegungsauftrags angezeigt.
Valid	OUTPUT	BOOL	<ul style="list-style-type: none"> ■ Zustand gültig <ul style="list-style-type: none"> – TRUE: Die gelesene Zustand des Bewegungsauftrags ist gültig
Error	OUTPUT	BOOL	<ul style="list-style-type: none"> ■ Status <ul style="list-style-type: none"> – TRUE: Ein Fehler ist aufgetreten. Zusätzliche Fehlerinformationen können dem Parameter <i>ErrorID</i> entnommen werden.
ErrorID	OUTPUT	WORD	Zusätzliche Fehlerinformationen <i>↳ Kapitel 9 "ErrorID - Zusätzliche Fehlerinformationen" auf Seite 340</i>
ConstantVelocity	OUTPUT	BOOL	<ul style="list-style-type: none"> ■ Zustand Bewegungsauftrag: Geschwindigkeit <ul style="list-style-type: none"> – TRUE: Geschwindigkeit ist konstant
Accelerating	OUTPUT	BOOL	<ul style="list-style-type: none"> ■ Zustand Bewegungsauftrag: Beschleunigung <ul style="list-style-type: none"> – TRUE: Achse wird beschleunigt; die Geschwindigkeit der Achse erhöht sich.
Decelerating	OUTPUT	BOOL	<ul style="list-style-type: none"> ■ Zustand Bewegungsauftrag: Bremsvorgang <ul style="list-style-type: none"> – TRUE: Achse wird gebremst; die Geschwindigkeit der Achse wird geringer.
DirectionPositive	OUTPUT	BOOL	<ul style="list-style-type: none"> ■ Zustand Bewegungsauftrag: Position zunehmend <ul style="list-style-type: none"> – TRUE: Die Position der Achse nimmt zu
DirectionNegative	OUTPUT	BOOL	<ul style="list-style-type: none"> ■ Zustand Bewegungsauftrag: Position abnehmend <ul style="list-style-type: none"> – TRUE: Die Position der Achse nimmt ab

PLCopen-State

- Start des Auftrags in jedem PLCopen-State möglich.

Zustand des Bewegungsauftrags lesen

Mit *Enable* gleich TRUE wird an den Ausgängen der Zustand des Bewegungsauftrags der Achse geliefert.

**Zustandsdiagramm der
Bausteinparameter**

- (1) Zum Zeitpunkt (1) wird *Enable* = TRUE gesetzt. Damit liefert *Valid* den Wert TRUE und an den Ausgängen wird der Zustand des Bewegungsauftrags angezeigt.
- (2) Zum Zeitpunkt (2) wird *Enable* = FALSE gesetzt. Damit werden sämtliche Ausgänge gleich FALSE bzw. 0 gesetzt.

3.4.3.19 FB 823 - MC_TouchProbe - Achsposition erfassen**Beschreibung**

Dieser Baustein erfasst einmalig die Achsposition in Abhängigkeit eines Trigger-Signals. Das Trigger-Signal kann über die am Eingang *TriggerInput* angegebene Variable konfiguriert werden. Als Trigger-Signal kann z.B. ein Digitaleingang oder die Gebernulldspur dienen.

Parameter

Parameter	Deklaration	Datentyp	Beschreibung
Axis	IN_OUT	MC_AXIS_REF	Referenz zur Achse.
TriggerInput	IN_OUT	MC_TRIGGER_REF	Referenz zum Trigger-Eingang. Struktur <ul style="list-style-type: none"> ■ .Probe <ul style="list-style-type: none"> – 01: TouchProbe-Register 1 – 02: TouchProbe-Register 2 ■ .TriggerSource <ul style="list-style-type: none"> – 00: Eingang – 00: Encoder Nullimpuls ■ .Triggermode <ul style="list-style-type: none"> – 00: SingleTrigger (fix) ■ .Reserved (0 fix)
Execute	IN	BOOL	Mit einer Flanke 0-1 an <i>Execute</i> wird die Erfassung der Achsposition aktiviert.
Done	OUT	BOOL	■ Status <ul style="list-style-type: none"> – TRUE: Auftrag erfolgreich durchgeführt. Die Achsposition wurde erfasst.
Busy	OUT	BOOL	■ Status <ul style="list-style-type: none"> – TRUE: Auftrag ist in Bearbeitung.
CommandAborted	OUT	BOOL	■ Status <ul style="list-style-type: none"> – TRUE: Der Auftrag wurde während der Bearbeitung von einem anderen Auftrag abgebrochen.
Error	OUT	BOOL	■ Status <ul style="list-style-type: none"> – TRUE: Ein Fehler ist aufgetreten. Zusätzliche Fehlerinformationen können dem Parameter <i>ErrorID</i> entnommen werden.
ErrorID	OUT	WORD	Zusätzliche Fehlerinformationen 🔗 Kapitel 9 "ErrorID - Zusätzliche Fehlerinformationen" auf Seite 340
RecordedPosition	OUT	REAL	Erfasste Achsposition zum Zeitpunkt des Trigger-Signals [Anwendereinheiten].



- Ein laufender Auftrag wird auch beim Setzen von *Execute* gleich *FALSE*, solange ausgeführt, bis dieser abgearbeitet ist. Die erfasste Achsposition wird dann für einen Zyklus am Ausgang *RecordedPosition* ausgegeben. ↗ Kapitel 8.3 "Verhalten der Ein- und Ausgänge" auf Seite 338
- Damit der Befehl ausgeführt werden kann, muss die Kommunikation mit der Achse OK und der PLCopen-State ungleich *Homing* sein.
- Ein laufender Auftrag kann durch einen neuen *MC_TouchProbe* auf der gleichen Achse abgebrochen werden.
- Ein laufender Auftrag kann durch den Befehl *MC_AbortTrigger* abgebrochen werden.
- Ein laufender Auftrag kann durch den Befehl *MC_Home* abgebrochen werden.

Achsposition erfassen

Mit einer Flanke 0-1 an *Execute* wird die Erfassung der Achsposition aktiviert. Solange der Befehl abgearbeitet wird, zeigt *Busy* den Wert *TRUE*. Nach Abarbeitung des Befehls, wird *Busy* gleich *FALSE* und *Done* gleich *TRUE* geliefert. Der erfasste Wert wird in *RecordedPosition* ausgegeben.

3.4.3.20 FB 824 - MC_AbortTrigger - Achsposition erfassen abbrechen

Beschreibung Dieser Baustein bricht die durch MC_TouchProbe gestartete Erfassung der Achsposition ab.

Parameter

Parameter	Deklaration	Datentyp	Beschreibung
Axis	IN_OUT	MC_AXIS_REF	Referenz zur Achse.
TriggerInput	IN_OUT	MC_TRIGGER_REF	Referenz zum Trigger-Eingang. Struktur <ul style="list-style-type: none"> ■ .Probe <ul style="list-style-type: none"> – 01: TouchProbe-Register 1 – 02: TouchProbe-Register 2 ■ .TriggerSource <ul style="list-style-type: none"> – 00: Eingang – 00: Encoder Nullimpuls ■ .Triggermode <ul style="list-style-type: none"> – 00: SingleTrigger (fix) ■ .Reserved (0 fix)
Execute	IN	BOOL	Mit einer Flanke 0-1 an <i>Execute</i> wird die Erfassung der Achsposition abgebrochen.
Done	OUT	BOOL	■ Status <ul style="list-style-type: none"> – TRUE: Auftrag erfolgreich durchgeführt. Die Erfassung der Achsposition wurde abgebrochen.
Busy	OUT	BOOL	■ Status <ul style="list-style-type: none"> – TRUE: Auftrag ist in Bearbeitung.
Error	OUT	BOOL	■ Status <ul style="list-style-type: none"> – TRUE: Ein Fehler ist aufgetreten. Zusätzliche Fehlerinformationen können dem Parameter <i>ErrorID</i> entnommen werden.
ErrorID	OUT	WORD	Zusätzliche Fehlerinformationen 🔗 Kapitel 9 "ErrorID - Zusätzliche Fehlerinformationen" auf Seite 340




Damit der Befehl ausgeführt werden kann, muss die Kommunikation mit der Achse OK sein.



Erfassung der Achsposition abbrechen

Mit einer Flanke 0-1 an *Execute* wird die Erfassung der Achsposition abgebrochen. Solange der Befehl abgearbeitet wird, zeigt *Busy* den Wert TRUE. Nach Abarbeitung des Befehls, wird *Busy* gleich FALSE und *Done* gleich TRUE geliefert.

3.4.3.21 FB 825 - MC_ReadBoolParameter - Boolean-Parameter von Achse lesen**Beschreibung**

Mit MC_ReadBoolParameter wird der Parameter, der über die Parameter-Nummer festgelegt ist, vom Datentyp BOOL von der Achse gelesen.  Kapitel 3.4.3.35 "PLCopen Parameter" auf Seite 197

Parameter

Parameter	Deklaration	Datentyp	Beschreibung
Axis	IN_OUT	MC_AXIS_REF	Referenz zur Achse
Execute	INPUT	BOOL	<ul style="list-style-type: none"> ■ Parameter Achse lesen <ul style="list-style-type: none"> – Flanke 0-1: Das Lesen des Parameters wird durchgeführt
Parameter Number	INPUT	INT	Nummer des Parameters, welcher gelesen werden soll.  Kapitel 3.4.3.35 "PLCopen Parameter" auf Seite 197
Done	OUTPUT	BOOL	<ul style="list-style-type: none"> ■ Status <ul style="list-style-type: none"> – TRUE: Auftrag erfolgreich durchgeführt. Parameter wurde ausgelesen
Busy	OUTPUT	BOOL	<ul style="list-style-type: none"> ■ Status <ul style="list-style-type: none"> – TRUE: Auftrag ist in Bearbeitung
Error	OUTPUT	BOOL	<ul style="list-style-type: none"> ■ Status <ul style="list-style-type: none"> – TRUE: Ein Fehler ist aufgetreten. Zusätzliche Fehlerinformationen können dem Parameter <i>ErrorID</i> entnommen werden.
ErrorID	OUTPUT	WORD	Zusätzliche Fehlerinformationen  Kapitel 9 "ErrorID - Zusätzliche Fehlerinformationen" auf Seite 340
Value	OUTPUT	BOOL	Wert des gelesenen Parameters

PLCopen-State

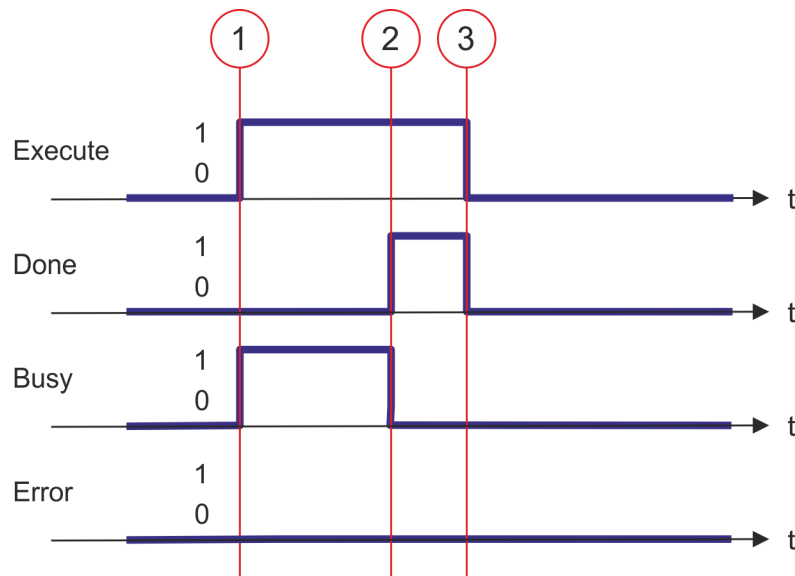
- Start des Auftrags in jedem PLCopen-State möglich.

Parameter der Achse lesen

Mit einer Flanke 0-1 an *Execute* wird das Lesen des Parameters gestartet. Solange das Lesen des Parameters läuft, zeigt *Busy* den Wert TRUE. Nachdem der Parameter gelesen wurde, wird *Busy* gleich FALSE und *Done* gleich TRUE geliefert. Der Ausgang *Value* zeigt den Wert des Parameters an.



Ein laufender Auftrag wird auch beim Setzen von *Execute* gleich FALSE weiterhin ausgeführt.

Zustandsdiagramm der Bausteinparameter

- (1) Mit der Flanke 0-1 an *Execute* zum Zeitpunkt (1) wird das Lesen des Parameters gestartet und *Busy* liefert den Wert TRUE.
- (2) Zum Zeitpunkt (2) ist das Lesen des Parameters erfolgreich abgeschlossen. *Busy* liefert den Wert FALSE und *Done* den Wert TRUE.
- (3) Zum Zeitpunkt (3) ist der Auftrag abgeschlossen und *Execute* wird gleich FALSE gesetzt und dadurch sämtliche Ausgangsparameter auf FALSE bzw. 0 gesetzt.

3.4.3.22 FB 826 - MC_WriteBoolParameter - Boolean-Parameter an Achse schreiben**Beschreibung**

Mit MC_WriteBoolParameter wird der Wert des Parameters, der über die Parameter-Nummer festgelegt ist, vom Datentyp BOOL zur Achse geschrieben. [↗ Kapitel 3.4.3.35 "PLCopen Parameter" auf Seite 197](#)

Parameter

Parameter	Deklaration	Datentyp	Beschreibung
Axis	IN_OUT	MC_AXIS_REF	Referenz zur Achse
Execute	INPUT	BOOL	<ul style="list-style-type: none"> ■ Parameter Achse schreiben <ul style="list-style-type: none"> – Flanke 0-1: Das Schreiben des Parameters wird durchgeführt
Parameter Number	INPUT	INT	Nummer des Parameters, welcher geschrieben werden soll. ↗ Kapitel 3.4.3.35 "PLCopen Parameter" auf Seite 197
Value	INPUT	BOOL	Wert des geschriebenen Parameters
Done	OUTPUT	BOOL	<ul style="list-style-type: none"> ■ Status <ul style="list-style-type: none"> – TRUE: Auftrag erfolgreich durchgeführt. Parameter wurde geschrieben
Busy	OUTPUT	BOOL	<ul style="list-style-type: none"> ■ Status <ul style="list-style-type: none"> – TRUE: Auftrag ist in Bearbeitung
Error	OUTPUT	BOOL	<ul style="list-style-type: none"> ■ Status <ul style="list-style-type: none"> – TRUE: Ein Fehler ist aufgetreten. Zusätzliche Fehlerinformationen können dem Parameter <i>ErrorID</i> entnommen werden.
ErrorID	OUTPUT	WORD	Zusätzliche Fehlerinformationen ↗ Kapitel 9 "ErrorID - Zusätzliche Fehlerinformationen" auf Seite 340

PLCopen-State

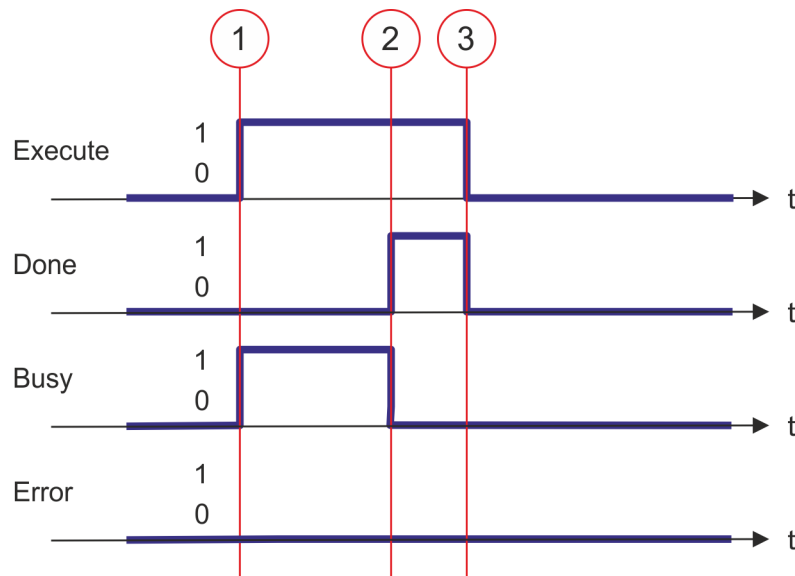
- Start des Auftrags in jedem PLCopen-State möglich.

Parameter der Achse schreiben

Mit einer Flanke 0-1 an *Execute* wird das Schreiben des Parameters gestartet. Solange das Schreiben des Parameters läuft, zeigt *Busy* den Wert TRUE. Nachdem der Parameter geschrieben wurde, wird *Busy* gleich FALSE und *Done* gleich TRUE geliefert.



Ein laufender Auftrag wird auch beim Setzen von Execute gleich FALSE weiterhin ausgeführt.

**Zustandsdiagramm der
Bausteinparameter**

- (1) Mit der Flanke 0-1 an *Execute* zum Zeitpunkt (1) wird das Schreiben des Parameters gestartet und *Busy* liefert den Wert TRUE.
- (2) Zum Zeitpunkt (2) ist das Schreiben des Parameters erfolgreich abgeschlossen. *Busy* liefert den Wert FALSE und *Done* den Wert TRUE.
- (3) Zum Zeitpunkt (3) ist der Auftrag abgeschlossen und *Execute* wird gleich FALSE gesetzt und dadurch sämtliche Ausgangsparameter auf FALSE bzw. 0 gesetzt.

3.4.3.23 FB 827 - VMC_ReadDWordParameter - Doppelwort-Parameter von Achse lesen**Beschreibung**

Mit VMC_ReadDWordParameter wird der Parameter, der über die Parameter-Nummer festgelegt ist, vom Datentyp DWORD von der Achse gelesen. ↗ *Kapitel 3.4.3.35 "PLCopen Parameter" auf Seite 197*

Parameter

Parameter	Deklaration	Datentyp	Beschreibung
Axis	IN_OUT	MC_AXIS_REF	Referenz zur Achse
Execute	INPUT	BOOL	<ul style="list-style-type: none"> ■ Parameter Achse lesen <ul style="list-style-type: none"> – Flanke 0-1: Das Lesen des Parameters wird durchgeführt
ParameterN-umber	INPUT	INT	Nummer des Parameters, welcher gelesen werden soll. ↗ <i>Kapitel 3.4.3.35 "PLCopen Parameter" auf Seite 197</i>
Done	OUTPUT	BOOL	<ul style="list-style-type: none"> ■ Status <ul style="list-style-type: none"> – TRUE: Auftrag erfolgreich durchgeführt. Parameter wurde ausgelesen
Busy	OUTPUT	BOOL	<ul style="list-style-type: none"> ■ Status <ul style="list-style-type: none"> – TRUE: Auftrag ist in Bearbeitung
Error	OUTPUT	BOOL	<ul style="list-style-type: none"> ■ Status <ul style="list-style-type: none"> – TRUE: Ein Fehler ist aufgetreten. Zusätzliche Fehlerinformationen können dem Parameter <i>ErrorID</i> entnommen werden.
ErrorID	OUTPUT	WORD	Zusätzliche Fehlerinformationen ↗ <i>Kapitel 9 "ErrorID - Zusätzliche Fehlerinformationen" auf Seite 340</i>
Value	OUTPUT	DWORD	Wert des gelesenen Parameters

PLCopen-State

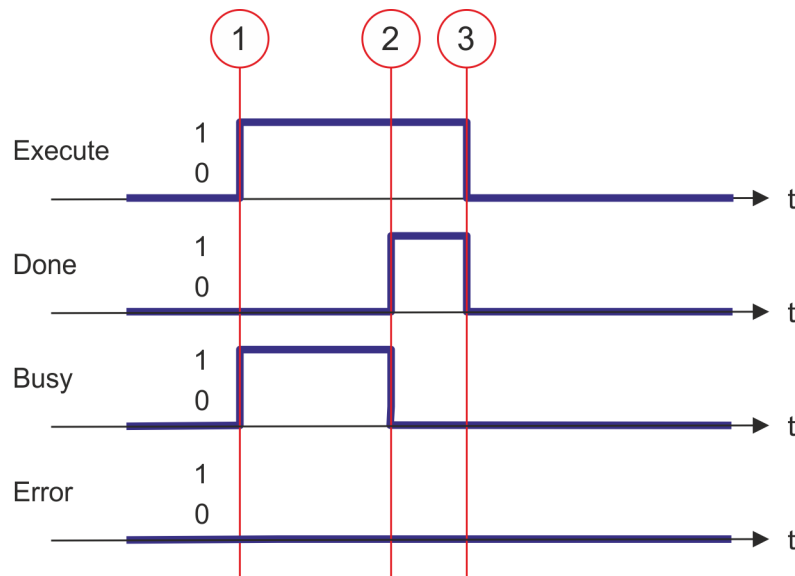
- Start des Auftrags in jedem PLCopen-State möglich.

Parameter der Achse lesen

Mit einer Flanke 0-1 an *Execute* wird das Lesen des Parameters gestartet. Solange das Lesen des Parameters läuft, zeigt *Busy* den Wert TRUE. Nachdem der Parameter gelesen wurde, wird *Busy* gleich FALSE und *Done* gleich TRUE geliefert. Der Ausgang *Value* zeigt den Wert des Parameters an.



Ein laufender Auftrag wird auch beim Setzen von Execute gleich FALSE weiterhin ausgeführt.

Zustandsdiagramm der Bausteinparameter

- (1) Mit der Flanke 0-1 an *Execute* zum Zeitpunkt (1) wird das Lesen des Parameters gestartet und *Busy* liefert den Wert TRUE.
- (2) Zum Zeitpunkt (2) ist das Lesen des Parameters erfolgreich abgeschlossen. *Busy* liefert den Wert FALSE und *Done* den Wert TRUE.
- (3) Zum Zeitpunkt (3) ist der Auftrag abgeschlossen und *Execute* wird gleich FALSE gesetzt und dadurch sämtliche Ausgangsparameter auf FALSE bzw. 0 gesetzt.

3.4.3.24 FB 828 - VMC_WriteDWordParameter - Doppelwort-Parameter an Achse schreiben**Beschreibung**

Mit VMC_WriteDWordParameter wird der Wert des Parameters, der über die Parameter-Nummer festgelegt ist, vom Datentyp DWORD zur Achse geschrieben. ↗ *Kapitel 3.4.3.35 "PLCopen Parameter" auf Seite 197*

Parameter

Parameter	Deklaration	Datentyp	Beschreibung
Axis	IN_OUT	MC_AXIS_REF	Referenz zur Achse
Execute	INPUT	BOOL	<ul style="list-style-type: none"> ■ Parameter Achse schreiben <ul style="list-style-type: none"> – Flanke 0-1: Das Schreiben des Parameters wird durchgeführt
Parameter Number	INPUT	INT	Nummer des Parameters, welcher geschrieben werden soll. ↗ <i>Kapitel 3.4.3.35 "PLCopen Parameter" auf Seite 197</i>
Value	INPUT	DWORD	Wert des geschriebenen Parameters
Done	OUTPUT	BOOL	<ul style="list-style-type: none"> ■ Status <ul style="list-style-type: none"> – TRUE: Auftrag erfolgreich durchgeführt. Parameter wurde geschrieben
Busy	OUTPUT	BOOL	<ul style="list-style-type: none"> ■ Status <ul style="list-style-type: none"> – TRUE: Auftrag ist in Bearbeitung
Error	OUTPUT	BOOL	<ul style="list-style-type: none"> ■ Status <ul style="list-style-type: none"> – TRUE: Ein Fehler ist aufgetreten. Zusätzliche Fehlerinformationen können dem Parameter <i>ErrorID</i> entnommen werden.
ErrorID	OUTPUT	WORD	Zusätzliche Fehlerinformationen ↗ <i>Kapitel 9 "ErrorID - Zusätzliche Fehlerinformationen" auf Seite 340</i>

PLCopen-State

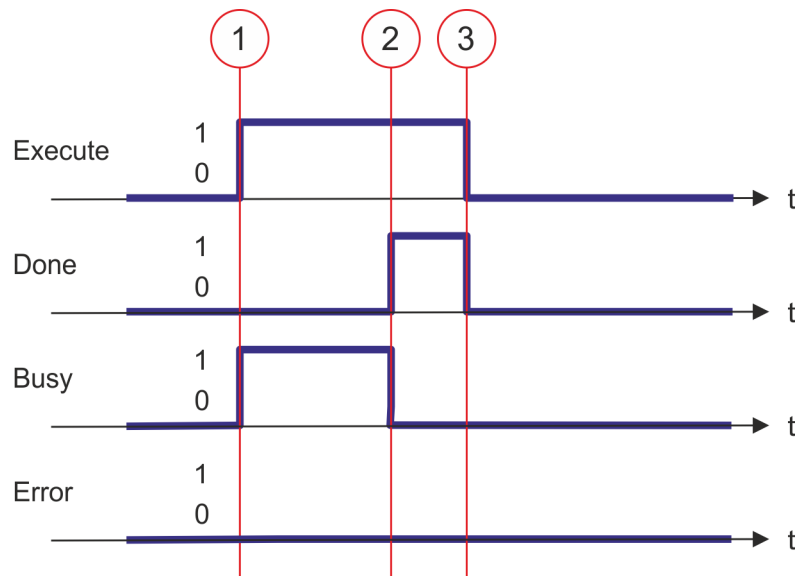
- Start des Auftrags in jedem PLCopen-State möglich.

Parameter der Achse schreiben

Mit einer Flanke 0-1 an *Execute* wird das Schreiben des Parameters gestartet. Solange das Schreiben des Parameters läuft, zeigt *Busy* den Wert TRUE. Nachdem der Parameter geschrieben wurde, wird *Busy* gleich FALSE und *Done* gleich TRUE geliefert.



Ein laufender Auftrag wird auch beim Setzen von Execute gleich FALSE weiterhin ausgeführt.

Zustandsdiagramm der Bausteinparameter

- (1) Mit der Flanke 0-1 an *Execute* zum Zeitpunkt (1) wird das Schreiben des Parameters gestartet und *Busy* liefert den Wert TRUE.
- (2) Zum Zeitpunkt (2) ist das Schreiben des Parameters erfolgreich abgeschlossen. *Busy* liefert den Wert FALSE und *Done* den Wert TRUE.
- (3) Zum Zeitpunkt (3) ist der Auftrag abgeschlossen und *Execute* wird gleich FALSE gesetzt und dadurch sämtliche Ausgangsparameter auf FALSE bzw. 0 gesetzt.

3.4.3.25 FB 829 - VMC_ReadWordParameter - Wort-Parameter von Achse lesen**Beschreibung**

Mit VMC_ReadWordParameter wird der Parameter, der über die Parameter-Nummer festgelegt ist, vom Datentyp WORD von der Achse gelesen. ↗ *Kapitel 3.4.3.35 "PLCopen Parameter" auf Seite 197*

Parameter

Parameter	Deklaration	Datentyp	Beschreibung
Axis	IN_OUT	MC_AXIS_REF	Referenz zur Achse
Execute	INPUT	BOOL	<ul style="list-style-type: none"> ■ Parameter Achse lesen <ul style="list-style-type: none"> – Flanke 0-1: Das Lesen des Parameters wird durchgeführt
Parameter Number	INPUT	INT	Nummer des Parameters, welcher gelesen werden soll. ↗ <i>Kapitel 3.4.3.35 "PLCopen Parameter" auf Seite 197</i>
Done	OUTPUT	BOOL	<ul style="list-style-type: none"> ■ Status <ul style="list-style-type: none"> – TRUE: Auftrag erfolgreich durchgeführt. Parameter wurde ausgelesen
Busy	OUTPUT	BOOL	<ul style="list-style-type: none"> ■ Status <ul style="list-style-type: none"> – TRUE: Auftrag ist in Bearbeitung
Error	OUTPUT	BOOL	<ul style="list-style-type: none"> ■ Status <ul style="list-style-type: none"> – TRUE: Ein Fehler ist aufgetreten. Zusätzliche Fehlerinformationen können dem Parameter <i>ErrorID</i> entnommen werden.
ErrorID	OUTPUT	WORD	Zusätzliche Fehlerinformationen ↗ <i>Kapitel 9 "ErrorID - Zusätzliche Fehlerinformationen" auf Seite 340</i>
Value	OUTPUT	WORD	Wert des gelesenen Parameters

PLCopen-State

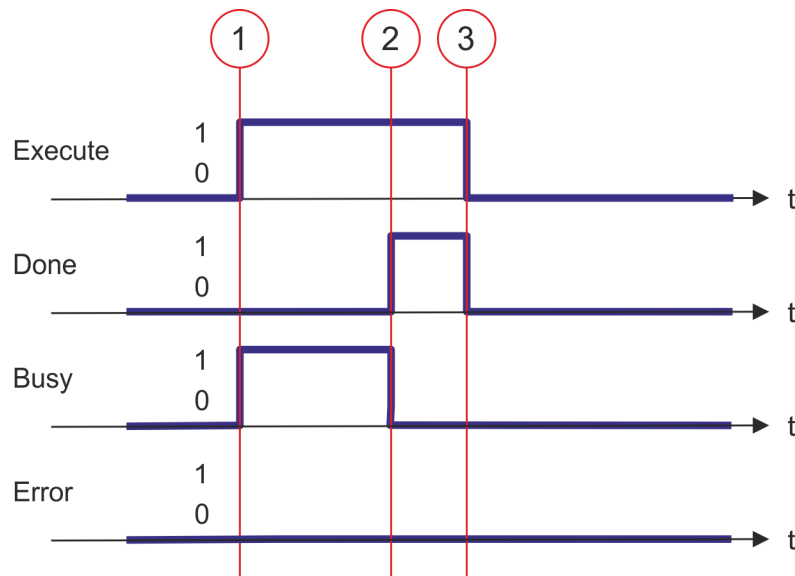
- Start des Auftrags in jedem PLCopen-State möglich.

Parameter der Achse lesen

Mit einer Flanke 0-1 an *Execute* wird das Lesen des Parameters gestartet. Solange das Lesen des Parameters läuft, zeigt *Busy* den Wert TRUE. Nachdem der Parameter gelesen wurde, wird *Busy* gleich FALSE und *Done* gleich TRUE geliefert. Der Ausgang *Value* zeigt den Wert des Parameters an.



Ein laufender Auftrag wird auch beim Setzen von Execute gleich FALSE weiterhin ausgeführt.

Zustandsdiagramm der Bausteinparameter

- (1) Mit der Flanke 0-1 an *Execute* zum Zeitpunkt (1) wird das Lesen des Parameters gestartet und *Busy* liefert den Wert TRUE.
- (2) Zum Zeitpunkt (2) ist das Lesen des Parameters erfolgreich abgeschlossen. *Busy* liefert den Wert FALSE und *Done* den Wert TRUE.
- (3) Zum Zeitpunkt (3) ist der Auftrag abgeschlossen und *Execute* wird gleich FALSE gesetzt und dadurch sämtliche Ausgangsparameter auf FALSE bzw. 0 gesetzt.

3.4.3.26 FB 830 - VMC_WriteWordParameter - Wort-Parameter an Achse schreiben**Beschreibung**

Mit VMC_WriteWordParameter wird der Wert des Parameters, der über die Parameter-Nummer festgelegt ist, vom Datentyp WORD zur Achse geschrieben. ↗ *Kapitel 3.4.3.35 "PLCopen Parameter" auf Seite 197*

Parameter

Parameter	Deklaration	Datentyp	Beschreibung
Axis	IN_OUT	MC_AXIS_REF	Referenz zur Achse
Execute	INPUT	BOOL	<ul style="list-style-type: none"> Parameter Achse schreiben <ul style="list-style-type: none"> Flanke 0-1: Das Schreiben des Parameters wird durchgeführt
Parameter Number	INPUT	INT	Nummer des Parameters, welcher geschrieben werden soll. ↗ <i>Kapitel 3.4.3.35 "PLCopen Parameter" auf Seite 197</i>
Value	INPUT	WORD	Wert des geschriebenen Parameters
Done	OUTPUT	BOOL	<ul style="list-style-type: none"> Status <ul style="list-style-type: none"> TRUE: Auftrag erfolgreich durchgeführt. Parameter wurde geschrieben
Busy	OUTPUT	BOOL	<ul style="list-style-type: none"> Status <ul style="list-style-type: none"> TRUE: Auftrag ist in Bearbeitung
Error	OUTPUT	BOOL	<ul style="list-style-type: none"> Status <ul style="list-style-type: none"> TRUE: Ein Fehler ist aufgetreten. Zusätzliche Fehlerinformationen können dem Parameter <i>ErrorID</i> entnommen werden.
ErrorID	OUTPUT	WORD	Zusätzliche Fehlerinformationen ↗ <i>Kapitel 9 "ErrorID - Zusätzliche Fehlerinformationen" auf Seite 340</i>

PLCopen-State

- Start des Auftrags in jedem PLCopen-State möglich.

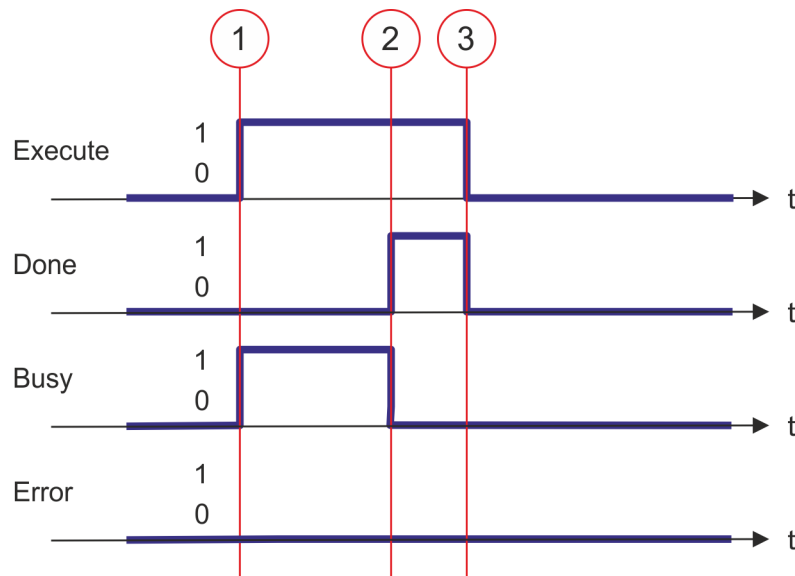
Parameter der Achse schreiben

Mit einer Flanke 0-1 an *Execute* wird das Schreiben des Parameters gestartet. Solange das Schreiben des Parameters läuft, zeigt *Busy* den Wert TRUE. Nachdem der Parameter geschrieben wurde, wird *Busy* gleich FALSE und *Done* gleich TRUE geliefert.




Ein laufender Auftrag wird auch beim Setzen von Execute gleich FALSE weiterhin ausgeführt.

Zustandsdiagramm der Bausteinparameter





- (1) Mit der Flanke 0-1 an *Execute* zum Zeitpunkt (1) wird das Schreiben des Parameters gestartet und *Busy* liefert den Wert TRUE.
- (2) Zum Zeitpunkt (2) ist das Schreiben des Parameters erfolgreich abgeschlossen. *Busy* liefert den Wert FALSE und *Done* den Wert TRUE.
- (3) Zum Zeitpunkt (3) ist der Auftrag abgeschlossen und *Execute* wird gleich FALSE gesetzt und dadurch sämtliche Ausgangsparameter auf FALSE bzw. 0 gesetzt.

3.4.3.27 FB 831 - VMC_ReadByteParameter - Byte-Parameter von Achse lesen**Beschreibung**

Mit VMC_ReadByteParameter wird der Parameter, der über die Parameter-Nummer festgelegt ist, vom Datentyp BYTE von der Achse gelesen.  *Kapitel 3.4.3.35 "PLCopen Parameter" auf Seite 197*

Parameter

Parameter	Deklaration	Datentyp	Beschreibung
Axis	IN_OUT	MC_AXIS_REF	Referenz zur Achse
Execute	INPUT	BOOL	<ul style="list-style-type: none"> ■ Parameter Achse lesen <ul style="list-style-type: none"> – Flanke 0-1: Das Lesen des Parameters wird durchgeführt
Parameter Number	INPUT	INT	Nummer des Parameters, welcher gelesen werden soll.  <i>Kapitel 3.4.3.35 "PLCopen Parameter" auf Seite 197</i>
Done	OUTPUT	BOOL	<ul style="list-style-type: none"> ■ Status <ul style="list-style-type: none"> – TRUE: Auftrag erfolgreich durchgeführt. Parameter wurde ausgelesen
Busy	OUTPUT	BOOL	<ul style="list-style-type: none"> ■ Status <ul style="list-style-type: none"> – TRUE: Auftrag ist in Bearbeitung
Error	OUTPUT	BOOL	<ul style="list-style-type: none"> ■ Status <ul style="list-style-type: none"> – TRUE: Ein Fehler ist aufgetreten. Zusätzliche Fehlerinformationen können dem Parameter <i>ErrorID</i> entnommen werden.
ErrorID	OUTPUT	WORD	Zusätzliche Fehlerinformationen  <i>Kapitel 9 "ErrorID - Zusätzliche Fehlerinformationen" auf Seite 340</i>
Value	OUTPUT	BYTE	Wert des gelesenen Parameters

PLCopen-State

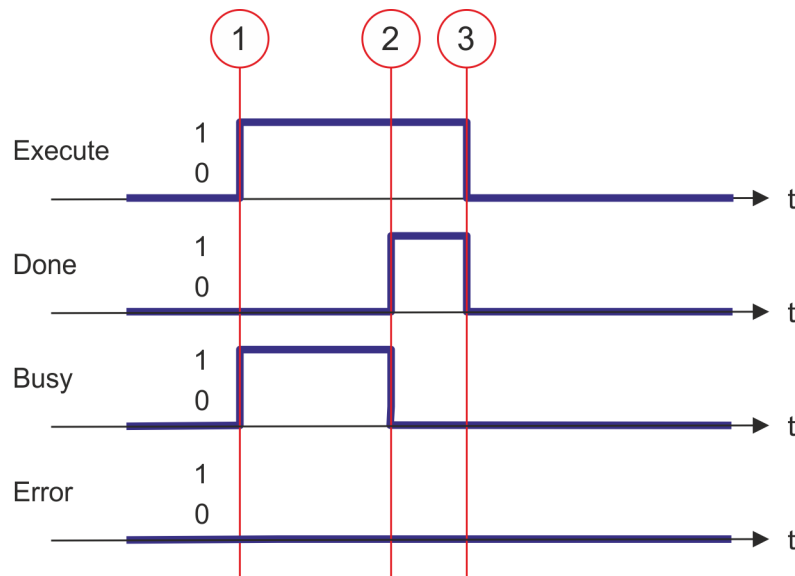
- Start des Auftrags in jedem PLCopen-State möglich.

Parameter der Achse lesen

Mit einer Flanke 0-1 an *Execute* wird das Lesen des Parameters gestartet. Solange das Lesen des Parameters läuft, zeigt *Busy* den Wert TRUE. Nachdem der Parameter gelesen wurde, wird *Busy* gleich FALSE und *Done* gleich TRUE geliefert. Der Ausgang *Value* zeigt den Wert des Parameters an.



Ein laufender Auftrag wird auch beim Setzen von Execute gleich FALSE weiterhin ausgeführt.

Zustandsdiagramm der Bausteinparameter

- (1) Mit der Flanke 0-1 an *Execute* zum Zeitpunkt (1) wird das Lesen des Parameters gestartet und *Busy* liefert den Wert TRUE.
- (2) Zum Zeitpunkt (2) ist das Lesen des Parameters erfolgreich abgeschlossen. *Busy* liefert den Wert FALSE und *Done* den Wert TRUE.
- (3) Zum Zeitpunkt (3) ist der Auftrag abgeschlossen und *Execute* wird gleich FALSE gesetzt und dadurch sämtliche Ausgangsparameter auf FALSE bzw. 0 gesetzt.

3.4.3.28 FB 832 - VMC_WriteByteParameter - Byte-Parameter an Achse schreiben

Beschreibung Mit VMC_WriteByteParameter wird der Wert des Parameters, der über die Parameter-Nummer festgelegt ist, vom Datentyp BYTE zur Achse geschrieben.

Parameter

Parameter	Deklaration	Datentyp	Beschreibung
Axis	IN_OUT	MC_AXIS_REF	Referenz zur Achse
Execute	INPUT	BOOL	<ul style="list-style-type: none"> ■ Parameter Achse schreiben <ul style="list-style-type: none"> – Flanke 0-1: Das Schreiben des Parameters wird durchgeführt
Parameter Number	INPUT	INT	Nummer des Parameters, welcher geschrieben werden soll. ↗ Kapitel 3.4.3.35 "PLCopen Parameter" auf Seite 197
Value	INPUT	BYTE	Wert des geschriebenen Parameters
Done	OUTPUT	BOOL	<ul style="list-style-type: none"> ■ Status <ul style="list-style-type: none"> – TRUE: Auftrag erfolgreich durchgeführt. Parameter wurde geschrieben
Busy	OUTPUT	BOOL	<ul style="list-style-type: none"> ■ Status <ul style="list-style-type: none"> – TRUE: Auftrag ist in Bearbeitung
Error	OUTPUT	BOOL	<ul style="list-style-type: none"> ■ Status <ul style="list-style-type: none"> – TRUE: Ein Fehler ist aufgetreten. Zusätzliche Fehlerinformationen können dem Parameter <i>ErrorID</i> entnommen werden.
ErrorID	OUTPUT	WORD	Zusätzliche Fehlerinformationen ↗ Kapitel 9 "ErrorID - Zusätzliche Fehlerinformationen" auf Seite 340

PLCopen-State

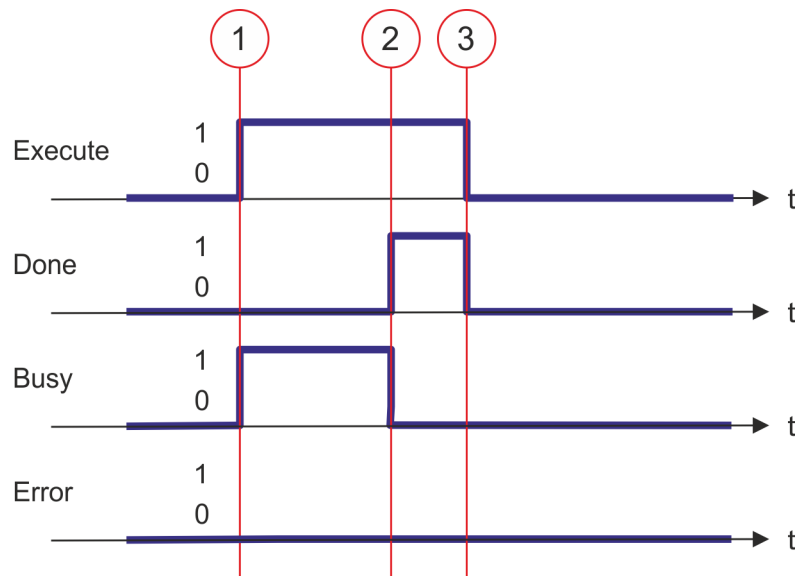
- Start des Auftrags in jedem PLCopen-State möglich.

Parameter der Achse schreiben

Mit einer Flanke 0-1 an *Execute* wird das Schreiben des Parameters gestartet. Solange das Schreiben des Parameters läuft, zeigt *Busy* den Wert TRUE. Nachdem der Parameter geschrieben wurde, wird *Busy* gleich FALSE und *Done* gleich TRUE geliefert.



Ein laufender Auftrag wird auch beim Setzen von Execute gleich FALSE weiterhin ausgeführt.

Zustandsdiagramm der Bausteinparameter

- (1) Mit der Flanke 0-1 an *Execute* zum Zeitpunkt (1) wird das Schreiben des Parameters gestartet und *Busy* liefert den Wert TRUE.
- (2) Zum Zeitpunkt (2) ist das Schreiben des Parameters erfolgreich abgeschlossen. *Busy* liefert den Wert FALSE und *Done* den Wert TRUE.
- (3) Zum Zeitpunkt (3) ist der Auftrag abgeschlossen und *Execute* wird gleich FALSE gesetzt und dadurch sämtliche Ausgangsparameter auf FALSE bzw. 0 gesetzt.

3.4.3.29 FB 833 - VMC_ReadDriveParameter - Antriebsparameter lesen

Beschreibung Mit dem VMC_ReadDriveParameter wird ein Parameterwert aus dem angeordneten Antrieb gelesen.

Parameter

Parameter	Deklaration	Datentyp	Beschreibung
Axis	IN_OUT	MC_AXIS_REF	Referenz zur Achse
Execute	INPUT	BOOL	<ul style="list-style-type: none"> Antriebsparameter lesen <ul style="list-style-type: none"> Flanke 0-1: Das Lesen des Antriebsparameters wird durchgeführt.
Index	INPUT	WORD	Index des Antriebsparameters
Subindex	INPUT	BYTE	Subindex des Antriebsparameters
Length	INPUT	BYTE	Datenlänge <ul style="list-style-type: none"> 1: BYTE 2: WORD 4: DWORD
Done	OUTPUT	BOOL	<ul style="list-style-type: none"> Status <ul style="list-style-type: none"> TRUE: Auftrag erfolgreich durchgeführt. Parameter wurde ausgelesen
Busy	OUTPUT	BOOL	<ul style="list-style-type: none"> Status <ul style="list-style-type: none"> TRUE: Auftrag ist in Bearbeitung
Error	OUTPUT	BOOL	<ul style="list-style-type: none"> Status <ul style="list-style-type: none"> TRUE: Ein Fehler ist aufgetreten. Zusätzliche Fehlerinformationen können dem Parameter <i>ErrorID</i> entnommen werden.
ErrorID	OUTPUT	WORD	Zusätzliche Fehlerinformationen <i>↪ Kapitel 9 "ErrorID - Zusätzliche Fehlerinformationen" auf Seite 340</i>
Value	OUTPUT	DWORD	Wert des gelesenen Parameters

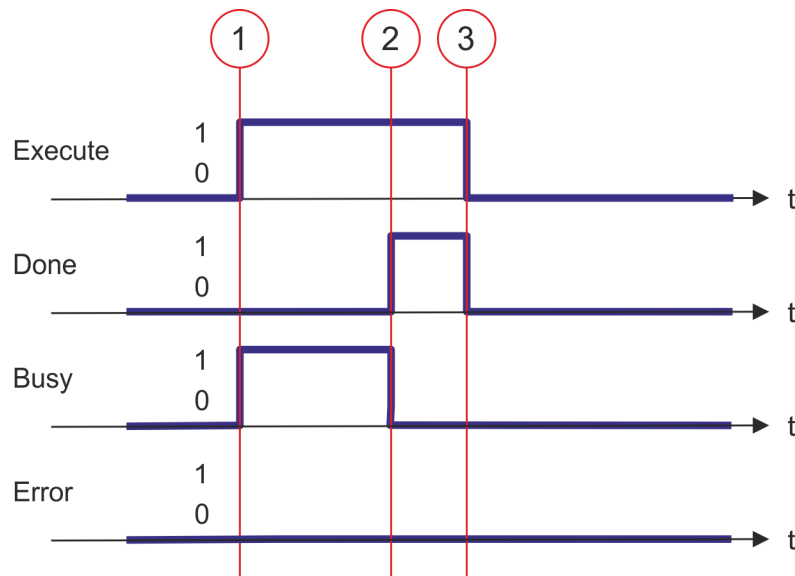
PLCopen-State

- Start des Auftrags in jedem PLCopen-State möglich.

Antriebsparameter lesen Mit einer Flanke 0-1 an *Execute* wird das Lesen des Antriebsparameters gestartet. Solange das Lesen des Parameters läuft, zeigt *Busy* den Wert TRUE. Nachdem der Parameter gelesen wurde, wird *Busy* gleich FALSE und *Done* gleich TRUE geliefert. Der Ausgang *Value* zeigt den Wert des Parameters an.



Ein laufender Auftrag wird auch beim Setzen von Execute gleich FALSE weiterhin ausgeführt.

Zustandsdiagramm der Bausteinparameter

- (1) Mit der Flanke 0-1 an *Execute* zum Zeitpunkt (1) wird das Lesen des Parameters gestartet und *Busy* liefert den Wert TRUE.
- (2) Zum Zeitpunkt (2) ist das Lesen des Parameters erfolgreich abgeschlossen. *Busy* liefert den Wert FALSE und *Done* den Wert TRUE.
- (3) Zum Zeitpunkt (3) ist der Auftrag abgeschlossen und *Execute* wird gleich FALSE gesetzt und dadurch sämtliche Ausgangsparameter auf FALSE bzw. 0 gesetzt.

3.4.3.30 FB 834 - VMC_WriteDriveParameter - Antriebsparameter schreiben

Beschreibung Mit dem VMC_WriteDriveParameter wird ein Parameterwert in den angeordneten Antrieb geschrieben.

Parameter

Parameter	Deklaration	Datentyp	Beschreibung
Axis	IN_OUT	MC_AXIS_REF	Referenz zur Achse
Execute	INPUT	BOOL	<ul style="list-style-type: none"> ■ Antriebsparameter schreiben <ul style="list-style-type: none"> – Flanke 0-1: Das Schreiben des Antriebsparameters wird durchgeführt.
Index	INPUT	WORD	Index des Antriebsparameters
Subindex	INPUT	BYTE	Subindex des Antriebsparameters
Length	INPUT	BYTE	Datenlänge: 1=BYTE; 2=WORD; 4=DWORD
Value	INPUT	DWORD	Wert des geschriebenen Parameters
Done	OUTPUT	BOOL	<ul style="list-style-type: none"> ■ Status <ul style="list-style-type: none"> – TRUE: Auftrag erfolgreich durchgeführt. Parameter wurde ausgelesen
Busy	OUTPUT	BOOL	<ul style="list-style-type: none"> ■ Status <ul style="list-style-type: none"> – TRUE: Auftrag ist in Bearbeitung
Error	OUTPUT	BOOL	<ul style="list-style-type: none"> ■ Status <ul style="list-style-type: none"> – TRUE: Ein Fehler ist aufgetreten. Zusätzliche Fehlerinformationen können dem Parameter <i>ErrorID</i> entnommen werden.
ErrorID	OUTPUT	WORD	Zusätzliche Fehlerinformationen <i>↳ Kapitel 9 "ErrorID - Zusätzliche Fehlerinformationen" auf Seite 340</i>

PLCopen-State

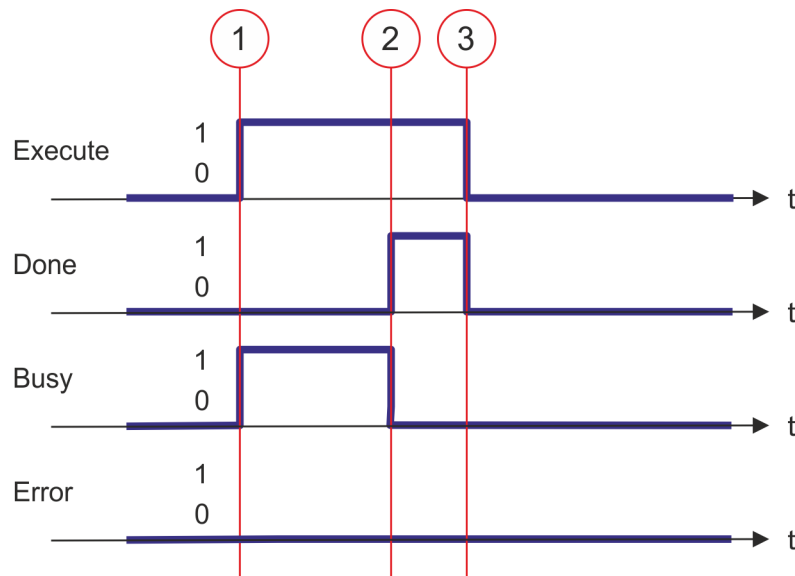
- Start des Auftrags in jedem PLCopen-State möglich.

Antriebsparameter schreiben

Mit einer Flanke 0-1 an *Execute* wird das Schreiben des Parameters gestartet. Solange das Schreiben des Parameters läuft, zeigt *Busy* den Wert TRUE. Nachdem der Parameter geschrieben wurde, wird *Busy* gleich FALSE und *Done* gleich TRUE geliefert.



Ein laufender Auftrag wird auch beim Setzen von Execute gleich FALSE weiterhin ausgeführt.

Zustandsdiagramm der Bausteinparameter

- (1) Mit der Flanke 0-1 an *Execute* zum Zeitpunkt (1) wird das Schreiben des Parameters gestartet und *Busy* liefert den Wert TRUE.
- (2) Zum Zeitpunkt (2) ist das Schreiben des Parameters erfolgreich abgeschlossen. *Busy* liefert den Wert FALSE und *Done* den Wert TRUE.
- (3) Zum Zeitpunkt (3) ist der Auftrag abgeschlossen und *Execute* wird gleich FALSE gesetzt und dadurch sämtliche Ausgangsparameter auf FALSE bzw. 0 gesetzt.

3.4.3.31 FB 835 - VMC_HomeInit_LimitSwitch - Initialisierung Referenzfahrt auf Endschalter

Beschreibung Dieser Baustein initialisiert die Referenzfahrt (Homing) auf den Endschalter.

Parameter

Parameter	Deklaration	Datentyp	Beschreibung
Execute	IN	BOOL	<ul style="list-style-type: none"> ■ Initialisierung der Referenzfahrt Methode <ul style="list-style-type: none"> – Flanke 0-1: Werte der Eingangsparameter werden übernommen und die Initialisierung der Referenzfahrt Methode gestartet.
Direction	IN	BOOL	<ul style="list-style-type: none"> ■ Richtung der Referenzfahrt <ul style="list-style-type: none"> – TRUE: auf positiven Endschalter – FALSE: auf negativen Endschalter
Velocity-SearchSwitch	IN	REAL	Geschwindigkeit für die Suche nach dem Schalter in [Anwendereinheiten/s]
VelocitySearch-Zero	IN	REAL	Geschwindigkeit für die Suche nach dem Index in [Anwendereinheiten/s]
Acceleration	IN	REAL	Beschleunigung in [Anwendereinheiten/s ²]
Done	OUT	BOOL	<ul style="list-style-type: none"> ■ Status <ul style="list-style-type: none"> – TRUE: Initialisierung wurde ohne Fehler beendet.
Busy	OUT	BOOL	<ul style="list-style-type: none"> ■ Status <ul style="list-style-type: none"> – TRUE: Initialisierung ist aktiv
Error	OUT	BOOL	<ul style="list-style-type: none"> ■ Status <ul style="list-style-type: none"> – TRUE: Ein Fehler ist aufgetreten. Zusätzliche Fehlerinformationen können dem Parameter <i>ErrorID</i> entnommen werden.
ErrorID	OUT	WORD	Zusätzliche Fehlerinformationen <i>↪ Kapitel 9 "ErrorID - Zusätzliche Fehlerinformationen" auf Seite 340</i>
AXIS	IN_OUT	MC_AXIS_REF	Referenz zur Achse.

Initialisierung Referenzfahrt auf Endschalter

Mit einer Flanke 0-1 an *Execute* werden die Werte der Eingangsparameter übernommen und die Initialisierung der Referenzfahrt Methode gestartet. So lange die Initialisierung aktiv ist, wird der Ausgang *Busy* auf True gesetzt. Ist die Initialisierung ohne Fehler beendet worden, wird der Ausgang *Done* auf True gesetzt. Tritt bei der Initialisierung ein Fehler auf, wird der Ausgang *Error* auf True gesetzt und am Ausgang *ErrorID* eine Fehlernummer ausgegeben.

Initialisierung der Referenzfahrt Methode

1. ➤ Überprüfen der Kommunikation zur Achse.
2. ➤ Prüfen auf erlaubte PLCopen Zustände.
3. ➤ Prüfung der Eingangswerte:
 - Eingang VelocitySearchSwitch [UserUnits] > 0.0
 - VelocitySearchSwitch [InternalUnits] > 0
 - VelocitySearchSwitch [InternalUnits] ≤ VelocityMax
 - Eingang VelocitySearchZero [UserUnits] > 0.0
 - VelocitySearchZero [InternalUnits] > 0
 - VelocitySearchZero [InternalUnits] ≤ VelocityMax
 - Eingang Acceleration [UserUnits] > 0.0
 - Acceleration [InternalUnits] > 0
 - Acceleration [InternalUnits] ≤ AccelerationMax
4. ➤ Übertragung der Antriebsparameter:
 - "Homing Method" Referenzfahrtmethode in Abhängigkeit vom Eingang "Direction"
Siehe Tabelle unten!
 - "Homing Speed during search for switch" [Inc/s]
Geschwindigkeit für die Schaltersuche
 - "Homing Speed during search for zero" [Inc/s]
Geschwindigkeit für die Indexsuche
 - "Homing Acceleration" [Inc/s²]
Anfahr- und Bremsbeschleunigung für die Referenzfahrt

Homing Method	Direction
1	false
2	true

3.4.3.32 FB 836 - VMC_HomeInit_HomeSwitch - Initialisierung Referenzfahrt auf Referenzschalter

Beschreibung Dieser Baustein initialisiert die Referenzfahrt (Homing) auf den Referenzschalter.

Parameter

Parameter	Deklaration	Datentyp	Beschreibung
Execute	IN	BOOL	<ul style="list-style-type: none"> ■ Initialisierung der Referenzfahrt Methode <ul style="list-style-type: none"> – Flanke 0-1: Werte der Eingangsparameter werden übernommen und die Initialisierung der Referenzfahrt Methode gestartet.
InitialDirection	IN	BOOL	<ul style="list-style-type: none"> ■ Startrichtung der Referenzfahrt <ul style="list-style-type: none"> – TRUE: auf positiven Endschalter – FALSE: auf negativen Endschalter
WithIndexPulse	IN	BOOL	<ul style="list-style-type: none"> ■ Referenzfahrt <ul style="list-style-type: none"> – TRUE: mit Nullimpuls – FALSE: ohne Nullimpuls
OnRisingEdge	IN	BOOL	<ul style="list-style-type: none"> ■ Flanke Referenzschalter <ul style="list-style-type: none"> – TRUE: Flanke 0-1 – FALSE: Flanke 1-0
SameDirIndexPulse	IN	BOOL	<ul style="list-style-type: none"> ■ Nullimpulssuche <ul style="list-style-type: none"> – TRUE: Nach dem Erkennen des Referenzschalters ohne Richtungswechsel den Nullimpuls suchen. – FALSE: Nach dem Erkennen des Referenzschalters einen Richtungswechsel zur Nullimpulssuche durchführen.
VelocitySearchSwitch	IN	REAL	Geschwindigkeit für die Schaltersuche in [Anwendereinheiten/s]
VelocitySearchZero	IN	REAL	Geschwindigkeit für die Indexsuche in [Anwendereinheiten/s]
Acceleration	IN	REAL	Beschleunigung in [Anwendereinheiten/s ²]
Done	OUT	BOOL	<ul style="list-style-type: none"> ■ Status <ul style="list-style-type: none"> – TRUE: Initialisierung wurde ohne Fehler beendet.
Busy	OUT	BOOL	<ul style="list-style-type: none"> ■ Status <ul style="list-style-type: none"> – TRUE: Initialisierung ist aktiv
Error	OUT	BOOL	<ul style="list-style-type: none"> ■ Status <ul style="list-style-type: none"> – TRUE: Ein Fehler ist aufgetreten. Zusätzliche Fehlerinformationen können dem Parameter <i>ErrorID</i> entnommen werden.
ErrorID	OUT	WORD	Zusätzliche Fehlerinformationen <i>☞ Kapitel 9 "ErrorID - Zusätzliche Fehlerinformationen" auf Seite 340</i>
AXIS	IN_OUT	MC_AXIS_REF	Referenz zur Achse.

Initialisierung Referenzfahrt auf Referenzschalter

Mit einer Flanke 0-1 an *Execute* werden die Werte der Eingangsparameter übernommen und die Initialisierung der Referenzfahrt Methode gestartet. So lange die Initialisierung aktiv ist, wird der Ausgang *Busy* auf True gesetzt. Ist die Initialisierung ohne Fehler beendet worden, wird der Ausgang *Done* auf True gesetzt. Tritt bei der Initialisierung ein Fehler auf, wird der Ausgang *Error* auf True gesetzt und am Ausgang *ErrorID* eine Fehlernummer ausgegeben.

Initialisierung der Referenzfahrt Methode

1. ➤ Überprüfen der Kommunikation zur Achse.
2. ➤ Prüfen auf erlaubte PLCopen Zustände.
3. ➤ Prüfung der Eingangswerte:
 - Eingang VelocitySearchSwitch [UserUnits] > 0.0
 - VelocitySearchSwitch [InternalUnits] > 0
 - VelocitySearchSwitch [InternalUnits] ≤ VelocityMax
 - Eingang VelocitySearchZero [UserUnits] > 0.0
 - VelocitySearchZero [InternalUnits] > 0
 - VelocitySearchZero [InternalUnits] ≤ VelocityMax
 - Eingang Acceleration [UserUnits] > 0.0
 - Acceleration [InternalUnits] > 0
 - Acceleration [InternalUnits] ≤ AccelerationMax
4. ➤ Übertragung der Antriebsparameter:
 - "Homing Method" Referenzfahrtmethode in Abhängigkeit vom Eingang "Direction"
Siehe Tabelle unten!
 - "Homing Speed during search for switch" [Inc/s]
Geschwindigkeit für die Schaltersuche
 - "Homing Speed during search for zero" [Inc/s]
Geschwindigkeit für die Indexsuche
 - "Homing Acceleration" [Inc/s²]
Anfahr- und Bremsbeschleunigung für die Referenzfahrt

Homing Method	InitialDirection	WithIndexPulse	OnRisingEdge	SameDirIndexPulse
7	positive	true	true	false
8	positive	true	true	true
9	positive	true	false	false
10	positive	true	false	true
11	negative	true	true	false
12	negative	true	true	true
13	negative	true	false	false
14	negative	true	false	true
24	positive	false	true	false
24	positive	false	true	true
24	positive	false	false	false
24	positive	false	false	true
28	negative	false	true	false

Homing Method	InitialDirection	WithIndexPulse	OnRisingEdge	SameDirIndexPulse
28	negative	false	true	true
28	negative	false	false	false
28	negative	false	false	true

3.4.3.33 FB 837 - VMC_HomeInit_ZeroPulse - Initialisierung Referenzfahrt auf Null Impuls

Beschreibung Dieser Baustein initialisiert die Referenzfahrt (Homing) auf Null Impuls.

Parameter

Parameter	Deklaration	Datentyp	Beschreibung
Execute	IN	BOOL	<ul style="list-style-type: none"> ■ Initialisierung der Referenzfahrt Methode <ul style="list-style-type: none"> – Flanke 0-1: Werte der Eingangsparameter werden übernommen und die Initialisierung der Referenzfahrt Methode gestartet.
Direction	IN	BOOL	<ul style="list-style-type: none"> ■ Richtung der Referenzfahrt <ul style="list-style-type: none"> – TRUE: Positive Richtung – FALSE: Negative Richtung
VelocitySearchZero	IN	REAL	Geschwindigkeit für die Indexsuche in [Anwendereinheiten/s]
Acceleration	IN	REAL	Beschleunigung in [Anwendereinheiten/s ²]
Done	OUT	BOOL	<ul style="list-style-type: none"> ■ Status <ul style="list-style-type: none"> – TRUE: Initialisierung wurde ohne Fehler beendet.
Busy	OUT	BOOL	<ul style="list-style-type: none"> ■ Status <ul style="list-style-type: none"> – TRUE: Initialisierung ist aktiv
Error	OUT	BOOL	<ul style="list-style-type: none"> ■ Status <ul style="list-style-type: none"> – TRUE: Ein Fehler ist aufgetreten. Zusätzliche Fehlerinformationen können dem Parameter <i>ErrorID</i> entnommen werden.
ErrorID	OUT	WORD	Zusätzliche Fehlerinformationen <i>↳ Kapitel 9 "ErrorID - Zusätzliche Fehlerinformationen" auf Seite 340</i>
AXIS	IN_OUT	MC_AXIS_REF	Referenz zur Achse.

Initialisierung Referenzfahrt auf Nullimpuls

Mit einer Flanke 0-1 an *Execute* werden die Werte der Eingangsparameter übernommen und die Initialisierung der Referenzfahrt Methode gestartet. So lange die Initialisierung aktiv ist, wird der Ausgang *Busy* auf True gesetzt. Ist die Initialisierung ohne Fehler beendet worden, wird der Ausgang *Done* auf True gesetzt. Tritt bei der Initialisierung ein Fehler auf, wird der Ausgang *Error* auf True gesetzt und am Ausgang *ErrorID* eine Fehlernummer ausgegeben.

Initialisierung der Referenzfahrt Methode

1. ➤ Überprüfen der Kommunikation zur Achse.
2. ➤ Prüfen auf erlaubte PLCopen Zustände.
3. ➤ Prüfung der Eingangswerte:
 - Eingang VelocitySearchZero [UserUnits] > 0.0
 - VelocitySearchZero [InternalUnits] > 0
 - VelocitySearchZero [InternalUnits] ≤ VelocityMax
 - Eingang Acceleration [UserUnits] > 0.0
 - Acceleration [InternalUnits] > 0
 - Acceleration [InternalUnits] ≤ AccelerationMax

4. ➔ Übertragung der Antriebsparameter:

- "Homing Method" Referenzfahrtmethode in Abhängigkeit vom Eingang "Direction"
Siehe Tabelle unten!
- "Homing Speed during search for switch" [Inc/s]
Geschwindigkeit für die Schaltersuche
- "Homing Speed during search for zero" [Inc/s]
Geschwindigkeit für die Indexsuche
- "Homing Acceleration" [Inc/s²]
Anfahr- und Bremsbeschleunigung für die Referenzfahrt

Homing Method	Direction
33	false
34	true

3.4.3.34 FB 838 - VMC_HomeInit_SetPosition - Initialisierung Referenzfahrt auf aktuelle Position

Beschreibung Dieser Baustein initialisiert die Referenzfahrt (Homing) auf die aktuelle Position.

Parameter

Parameter	Deklaration	Datentyp	Beschreibung
Execute	IN	BOOL	<ul style="list-style-type: none"> ■ Initialisierung der Referenzfahrt Methode <ul style="list-style-type: none"> – Flanke 0-1: Werte der Eingangsparameter werden übernommen und die Initialisierung der Referenzfahrt Methode gestartet.
Done	OUT	BOOL	<ul style="list-style-type: none"> ■ Status <ul style="list-style-type: none"> – TRUE: Initialisierung wurde ohne Fehler beendet.
Busy	OUT	BOOL	<ul style="list-style-type: none"> ■ Status <ul style="list-style-type: none"> – TRUE: Initialisierung ist aktiv
Error	OUT	BOOL	<ul style="list-style-type: none"> ■ Status <ul style="list-style-type: none"> – TRUE: Ein Fehler ist aufgetreten. Zusätzliche Fehlerinformationen können dem Parameter <i>ErrorID</i> entnommen werden.
ErrorID	OUT	WORD	Zusätzliche Fehlerinformationen <i>↳ Kapitel 9 "ErrorID - Zusätzliche Fehlerinformationen" auf Seite 340</i>
AXIS	IN_OUT	MC_AXIS_REF	Referenz zur Achse.

Initialisierung Referenzfahrt auf Endschalter

Mit einer Flanke 0-1 an *Execute* werden die Werte der Eingangsparameter übernommen und die Initialisierung der Referenzfahrt Methode gestartet. So lange die Initialisierung aktiv ist, wird der Ausgang *Busy* auf TRUE gesetzt. Ist die Initialisierung ohne Fehler beendet worden, wird der Ausgang *Done* auf TRUE gesetzt. Tritt bei der Initialisierung ein Fehler auf, wird der Ausgang *Error* auf TRUE gesetzt und am Ausgang *ErrorID* eine Fehlernummer ausgegeben.

Initialisierung der Referenzfahrt Methode

1. ➤ Überprüfen der Kommunikation zur Achse.
2. ➤ Prüfen auf erlaubte PLCopen Zustände.
3. ➤ Übertragung der Antriebsparameter:
 - "Homing Method" = 35

3.4.3.35 PLCopen Parameter

PN	Name	Datentyp	R/W	Beschreibung
1	CommandedPosition	REAL	R	Sollposition Zugriff auf: <code>#Axis.Status.Positioning.SetValues.CommandedPosition</code>
2	SWLimitPos	REAL	R/W	Positive Software Endschalter Position Zugriff auf: <code>"Axis".AxisConfiguration.PositionLimits.MaxPosition</code>

PN	Name	Datentyp	R/W	Beschreibung
3	SWLimitNeg	REAL	R/W	Negative Software Endschalter Position Zugriff auf: "Axis".AxisConfiguration.PositionLimits.MinPosition
4	EnableLimitPos	BOOL	R/W	Positiven Endschalter aktivieren Zugriff auf: "Axis".AxisConfiguration.PositionLimits.EnableMaxPos
5	EnableLimitNeg	BOOL	R/W	Negativen Endschalter aktivieren Zugriff auf: "Axis".AxisConfiguration.PositionLimits.EnableMinPos
6	EnablePosLagMonitoring	BOOL	R/W	Überwachung Schleppabstand aktivieren Funktion wird nicht unterstützt
7	MaxPositionLag	REAL	R/W	Maximaler Schleppabstand Funktion wird nicht unterstützt
8	MaxVelocitySystem	REAL	R	Maximal zulässige Geschwindigkeit der Achse im Bewegungssystem Dieser Parameter wird aktuell nicht unterstützt
9	MaxVelocityAppl	REAL	R/W	Maximal zulässige Geschwindigkeit der Achse in der Anwendung Zugriff auf: #Axis.AxisConfiguration.DynamicLimits.MaxVelocityApp
10	ActualVelocity	REAL	R	Aktuelle Geschwindigkeit Zugriff auf: #Axis.Status.Positioning.ActValues.Velocity
11	CommandedVelocity	REAL	R	Sollgeschwindigkeit Zugriff auf: #Axis.Status.Positioning.SetValues.Velocity
12	MaxAccelerationSystem	REAL	R	Maximal zulässige Beschleunigung der Achse im Bewegungssystem Dieser Parameter wird aktuell nicht unterstützt
13	MaxAccelerationAppl	REAL	R/W	Maximal zulässige Beschleunigung der Achse in der Anwendung Zugriff auf: #Axis.AxisConfiguration.DynamicLimits.MaxAccelerationApp
14	MaxDecelerationSystem	REAL	R	Maximal zulässige Verzögerung der Achse im Bewegungssystem Dieser Parameter wird aktuell nicht unterstützt
15	MaxDecelerationAppl	REAL	R/W	Maximal zulässige Verzögerung der Achse in der Anwendung Zugriff auf: #Axis.AxisConfiguration.DynamicLimits.MaxDecelerationApp

PN	Name	Datentyp	R/W	Beschreibung
16	MaxJerkSystem	REAL	R	Maximal zulässiger Ruck der Achse im Bewegungssystem Dieser Parameter wird aktuell nicht unterstützt
17	MaxJerkAppl	REAL	R/W	Maximal zulässiger Ruck der Achse in der Anwendung Dieser Parameter wird aktuell nicht unterstützt

3.4.3.36 VIPA-spezifische Parameter

Positionierachse: Yaskawa Sigma-5 / Sigma-7 über EtherCAT

Nr.	Name	Datentyp	Index	Subindex	Zugriff
900	HomingDone	BOOL	-	-	R/W ^{1, 2}
901	PositiveTorqueLimit	BOOL	-	-	R/W ^{1, 2}
902	NegativeTorqueLimit	BOOL	-	-	R/W ^{1, 2}
1000	ErrorCode	WORD	603F	0	R ³
1001	HomeOffset	DWORD	607C	0	R/W ^{5, 6}
1002	HomingMethod	WORD	6098	0	R/W ^{3, 4}
1003	SpeedSearchSwitch	DWORD	6099	1	R/W ^{5, 6}
1004	SpeedSearchZero	DWORD	6099	2	R/W ^{5, 6}
1005	HomingAcceleration	DWORD	609A	0	R/W ^{5, 6}
1006	PositiveTorqueLimit	WORD	60E0	0	R/W ^{3, 4}
1007	NegativeTorqueLimit	WORD	0x60E1	0	R/W ^{3, 4}
1008	MotorRatedTorque	DWORD	0x6076	0	R/W ^{5, 6}
1009	FollowingErrorWindow	DWORD	0x6065	0	R/W ^{5, 6}
1010	FollowingErrorTimeOut	WORD	0x6066	0	R/W ^{3, 4}
1011	PositionWindow	DWORD	0x6067	0	R/W ^{5, 6}
1012	PositionTime	WORD	0x6068	0	R/W ^{3, 4}
1013	Min Position Limit	DWORD	0x607D	1	R/W ^{5, 6}
1014	Max Position Limit	DWORD	0x607D	2	R/W ^{5, 6}
1015	Digital outputs/ physical outputs	DWORD	0x60FE	1	R/W ^{5, 6}
1016	Digital outputs/ mask	DWORD	0x60FE	2	R/W ^{5, 6}
1017	Quick stop deceleration	DWORD	0x6085	0	R/W ^{5, 6}

1) Zugriff über § Kapitel 3.4.3.21 "FB 825 - MC_ReadBoolParameter - Boolean-Parameter von Achse lesen" auf Seite 170

2) Zugriff über § Kapitel 3.4.3.22 "FB 826 - MC_WriteBoolParameter - Boolean-Parameter an Achse schreiben" auf Seite 172

3) Zugriff über § Kapitel 3.4.3.25 "FB 829 - VMC_ReadWordParameter - Wort-Parameter von Achse lesen" auf Seite 178

4) Zugriff über § Kapitel 3.4.3.26 "FB 830 - VMC_WriteWordParameter - Wort-Parameter an Achse schreiben" auf Seite 180

5) Zugriff über § Kapitel 3.4.3.23 "FB 827 - VMC_ReadDWordParameter - Doppelwort-Parameter von Achse lesen" auf Seite 174

6) Zugriff über § Kapitel 3.4.3.24 "FB 828 - VMC_WriteDWordParameter - Doppelwort-Parameter an Achse schreiben" auf Seite 176

Nr.	Name	Datentyp	Index	Subindex	Zugriff
1018	Forward external torque limit	WORD	0x2404	0	R/W ^{3, 4}
1019	Reverse external torque limit	WORD	0x2405	0	R/W ^{3, 4}

- 1) Zugriff über [☞ Kapitel 3.4.3.21 "FB 825 - MC_ReadBoolParameter - Boolean-Parameter von Achse lesen" auf Seite 170](#)
- 2) Zugriff über [☞ Kapitel 3.4.3.22 "FB 826 - MC_WriteBoolParameter - Boolean-Parameter an Achse schreiben" auf Seite 172](#)
- 3) Zugriff über [☞ Kapitel 3.4.3.25 "FB 829 - VMC_ReadWordParameter - Wort-Parameter von Achse lesen" auf Seite 178](#)
- 4) Zugriff über [☞ Kapitel 3.4.3.26 "FB 830 - VMC_WriteWordParameter - Wort-Parameter an Achse schreiben" auf Seite 180](#)
- 5) Zugriff über [☞ Kapitel 3.4.3.23 "FB 827 - VMC_ReadDWordParameter - Doppelwort-Parameter von Achse lesen" auf Seite 174](#)
- 6) Zugriff über [☞ Kapitel 3.4.3.24 "FB 828 - VMC_WriteDWordParameter - Doppelwort-Parameter an Achse schreiben" auf Seite 176](#)

4 Einsatz *Sigma-5/7* Pulse Train

4.1 Übersicht

Voraussetzung

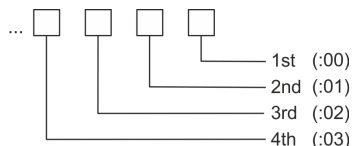
- SPEED7 Studio ab V1.7
oder
- Siemens SIMATIC Manager ab V 5.5 SP2 & *Simple Motion Control Library*
oder
- Siemens TIA Portal V 14 & *Simple Motion Control Library*
- System MICRO bzw. System SLIO CPU mit Pulse Train Ausgabe wie z.B. CPU M13-CCF0000 bzw. CPU 013-CCF0R00.
- *Sigma-5-* bzw. *Sigma-7*-Antrieb mit Pulse Train Optionskarte

Schritte der Projektierung

1. ➤ Parameter am Antrieb einstellen
 - Die Einstellung der Parameter hat mit dem Softwaretool *Sigma Win+* zu erfolgen.
2. ➤ Hardwarekonfiguration im VIPA *SPEED7 Studio*, Siemens SIMATIC Manager oder Siemens TIA Portal.
 - Projektierung der CPU.
3. ➤ Programmierung im VIPA *SPEED7 Studio*, Siemens SIMATIC Manager oder Siemens TIA Portal.
 - *VMC_AxisControl_PT*-Baustein zur Konfiguration und zur Kommunikation mit der Achse, welche über Pulse Train angebunden ist.

4.2 Parameter am Antrieb einstellen

Parameter-Digits



VORSICHT!

Vor der Erstinbetriebnahme müssen Sie Ihren Antrieb mit dem Softwaretool *Sigma Win+* an Ihre Applikation anpassen! Näheres hierzu finden Sie im Handbuch zu ihrem Antrieb.

Die nachfolgende Tabelle zeigt alle Parameter auf, die nicht den Standardwerten entsprechen. Zur Abstimmung auf die *Simple Motion Control Library* sind diese über *Sigma Win+* einzustellen:

Sigma-5/7

Servopack Parameter	Adresse:Digit	Name	Wert
Pn000	(2000h:01)	Basic Function Selection Switch 0	1: Position control (pulse train reference)
Pn002	(2002h:02)	Application Function Select Switch 2	1: Uses absolute encoder as incremental encoder
Pn200	(2200h:03)	Position Control Reference From Selection Switch	1: Uses reference input filter for open collector signal
Pn20E	(220Eh)	Electronic Gear Ratio (Numerator)	1024
Pn216	(2216h)	Position Reference Acceleration / Deceleration Time Constant	0
Pn217	(2217h)	Average Movement Time of Position Reference	0

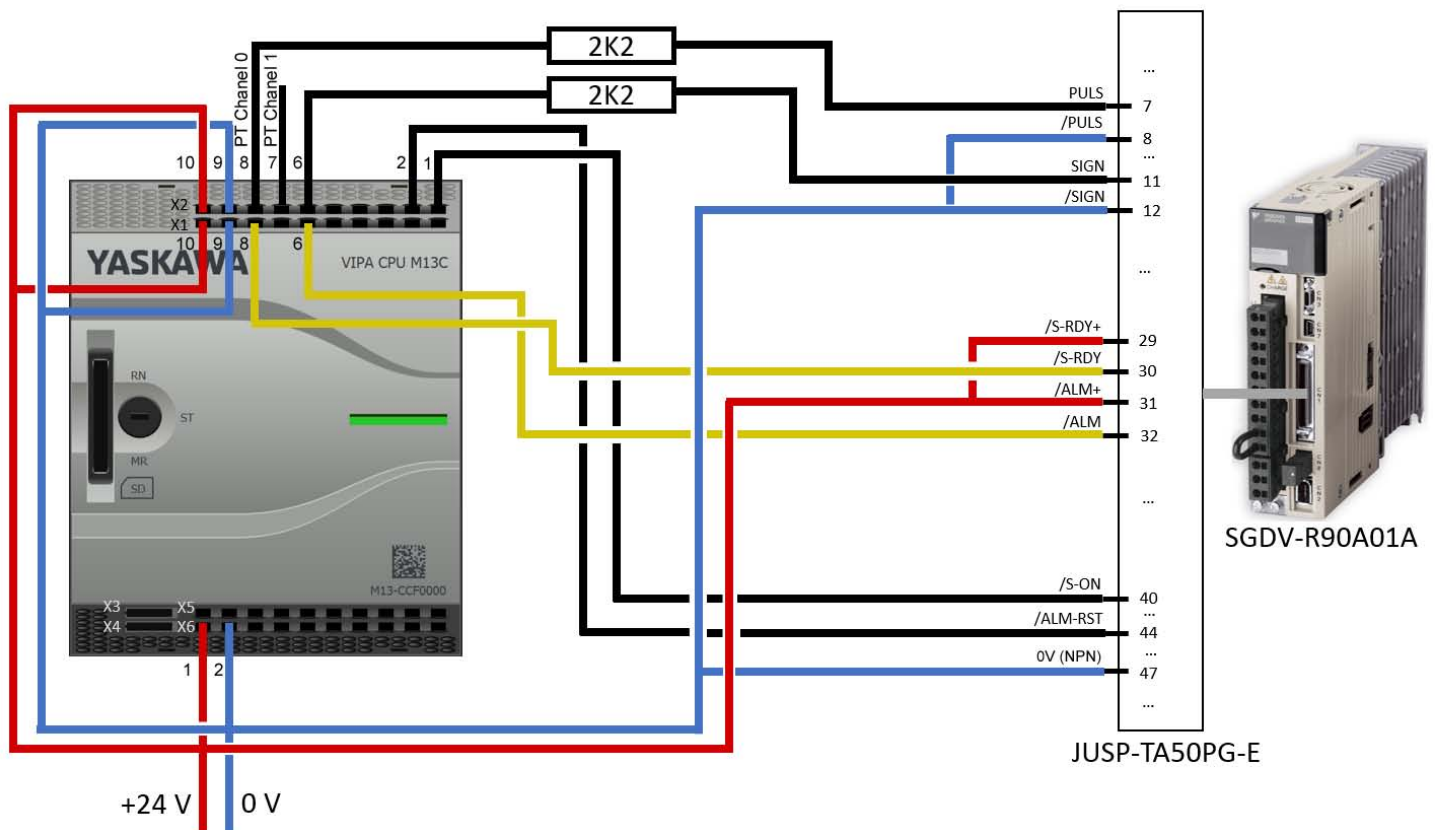
Beschaltung

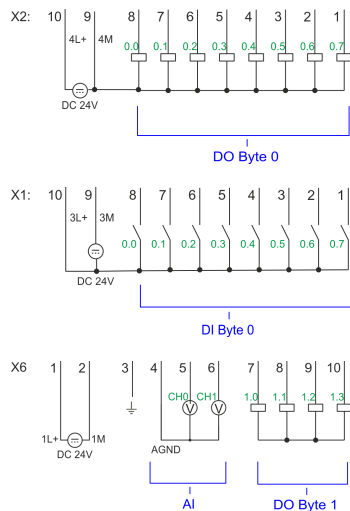
Servopack Parameter	Adresse:Digit	Name	Wert
Pn50A	(250Ah:02)	/P-CON Signal Mapping	8: Sets signal off
Pn50A	(250Ah:03)	P-OT Signal Mapping	8: Forward run allowed
Pn50B	(250Bh:00)	N-OT Signal Mapping	8: Reverse run allowed
Pn50B	(250Bh:02)	/P-CL Signal Mapping	8: Sets signal off
Pn50B	(250Bh:03)	/N-CL Signal Mapping	8: Sets signal off

4.3 Beschaltung

Beispielapplikation

Die nachfolgende Abbildung zeigt den Anschluss eines Sigma-5 Servo-Antriebs über Pulse Train an eine System MICRO CPU M13C. In diesem Beispiel ist der Pulse Train Kanal 0 (X2 - Pin 8) beschaltet. Zur Beschaltung von Kanal 1 ist X2 - Pin 7 zu verwenden.





X2	Funktion	Typ	LED grün rot	Beschreibung
1	DO 0.7	A	grün	Digital Ausgang DO 7
2	DO 0.6	A	grün	Digital Ausgang DO 6
6	DO 0.2	A	grün	Digital Ausgang DO 2
7	DO 0.1	A	grün	Pulse Train Kanal 1
8	DO 0.0	A	grün	Pulse Train Kanal 0
9	0 V	E	rot	4M: GND für Pulse Train LED leuchtet bei Fehler, Überlast bzw. Kurzschluss an den Ausgängen
10	DC 24V	E	grün	4L+: DC 24V Leistungsverorgung für Pulse Train

X1	Funktion	Typ	LED grün	Beschreibung
6	DI 0.2	E	grün	Digitaler Eingang DI 2
8	DI 0.0	E	grün	Digitaler Eingang DI 0
9	0 V	E		3M: GND Leistungsverorgung für Onboard DI
10	DC 24V	E	grün	3L+: DC 24V Leistungsverorgung für Onboard DI

X6	Funktion	Typ	LED grün	Beschreibung
1	Sys DC 24V	E	grün	1L+: DC 24V für Elektronikversorgung
2	Sys 0V	E		1M: GND für Elektronikversorgung

4.4 Einsatz im VIPA SPEED7 Studio

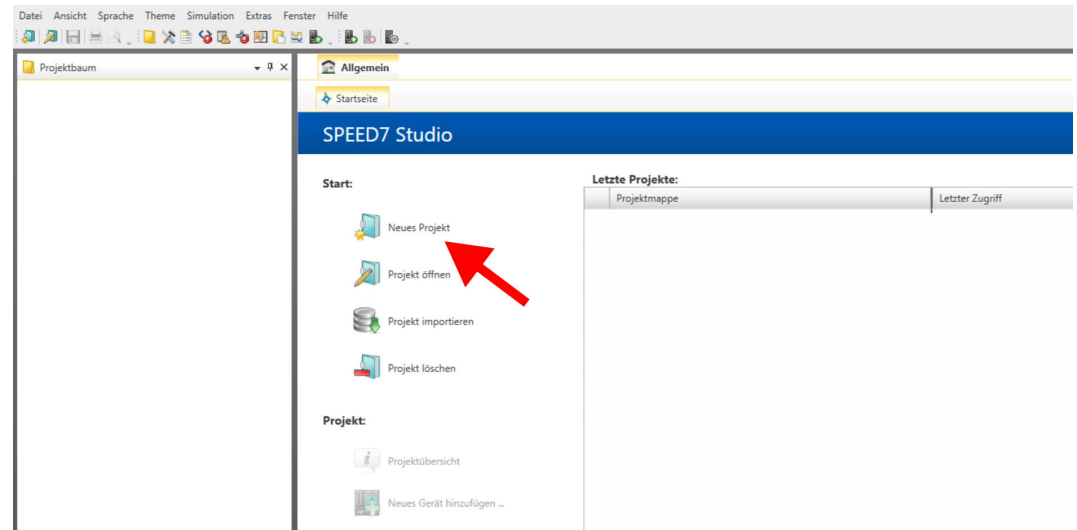
4.4.1 Hardware-Konfiguration

CPU im Projekt anlegen

Bitte verwenden Sie für die Projektierung das *SPEED7 Studio* ab V1.7.

Sollten Sie einen anderen Kanal als Kanal 0 verwenden, müssen Sie diesen in der Hardware-Konfiguration und in Ihrem Anwenderprogramm entsprechend anpassen.

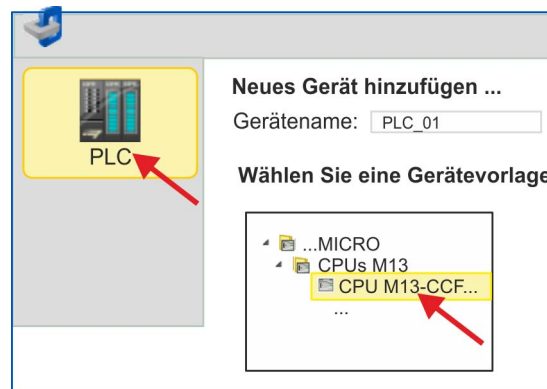
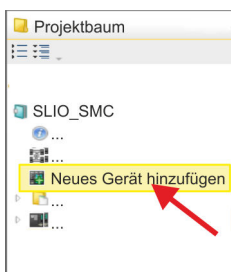
1. Starten Sie das *SPEED7 Studio*.



2. Erstellen sie auf der Startseite mit *"Neues Projekt"* ein neues Projekt und vergeben Sie einen *"Projektnamen"*.

⇒ Ein neues Projekt wird angelegt und in die Sicht *"Geräte und Netze"* gewechselt.

3. Klicken Sie im *Projektbaum* auf *"Neues Gerät hinzufügen ..."*.



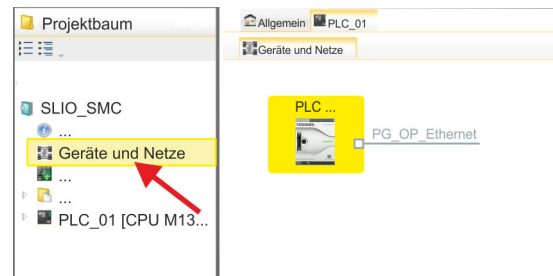
⇒ Es öffnet sich ein Dialog für die Geräteauswahl.

4. Wählen Sie unter den *"Gerätevorlagen"* Ihre CPU mit Pulse Train Funktionalität wie z.B. die System MICRO CPU M13-CCF0000 und klicken Sie auf [OK].

⇒ Die CPU wird in *"Geräte und Netze"* eingefügt und die *"Gerätekonfiguration"* geöffnet.

Ethernet-PG/OP-Kanal parametrieren

1. ➤ Klicken Sie im *Projektbaum* auf "*Geräte und Netze*".
⇒ Sie erhalten eine grafische Objekt-Ansicht Ihrer CPU.



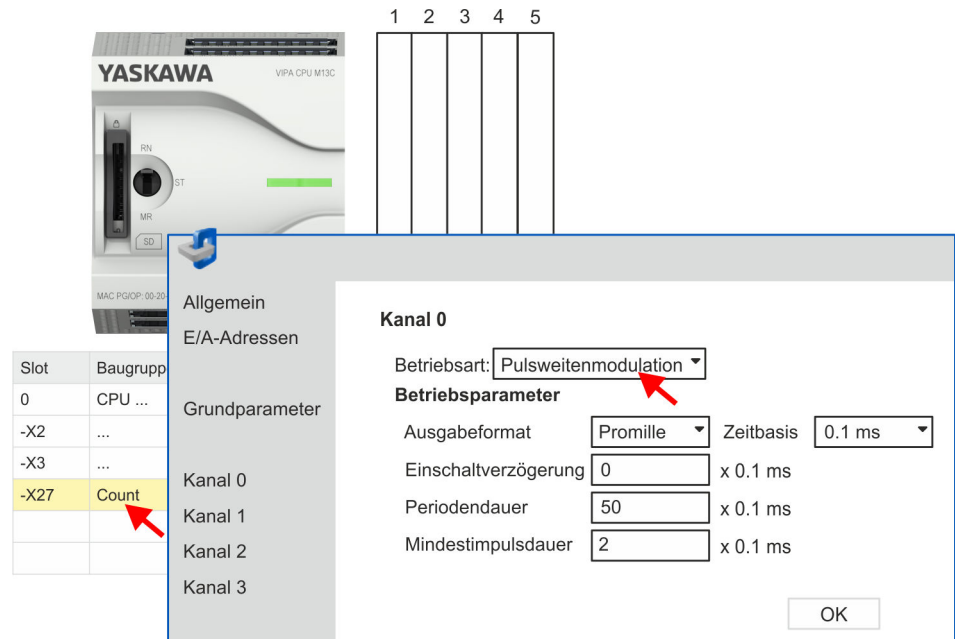
2. ➤ Klicken Sie auf das Netzwerk "*PG_OP_Ethernet*".
3. ➤ Wählen Sie "*Kontextmenü* ➔ *Eigenschaften der Schnittstelle*".
⇒ Es öffnet sich ein Dialogfenster. Hier können Sie IP-Adressdaten für Ihren Ethernet-PG/OP-Kanal angeben. Gültige IP-Adress-Parameter erhalten Sie von Ihrem Systemadministrator.
4. ➤ Bestätigen Sie Ihre Eingabe mit [OK].
⇒ Die IP-Adressdaten werden in Ihr Projekt übernommen und in "*Geräte und Netze*" unter "*Lokale Baugruppen*" aufgelistet.
Nach der Übertragung Ihres Projekts ist Ihre CPU über die angegebenen IP-Adressdaten via Ethernet-PG/OP-Kanal erreichbar.

E/A-Peripherie auf Pulse Train umschalten

Für die Parametrierung der Ein-/Ausgabeperipherie und der *Technologischen Funktionen* sind die entsprechenden Submodule der CPU zu verwenden. Für die Pulse Train Ausgabe ist das Submodul Count auf "*Pulsweitenmodulation*" umzuschalten.

1. ➤ Klicken Sie im *Projektbaum* auf "*PLC... > Gerätekonfiguration*".
2. ➤ Klicken Sie in der "*Gerätekonfiguration*" auf "*-X27 Count*" und wählen Sie "*Kontextmenü* ➔ *Eigenschaften der Baugruppe*".
⇒ Es öffnet sich der Eigenschaften-Dialog.
3. ➤ Klicken Sie auf den entsprechenden Kanal wie z.B. "*Kanal 0*" und stellen Sie unter "*Betriebsart*" die Funktion "*Pulsweitenmodulation*" ein.

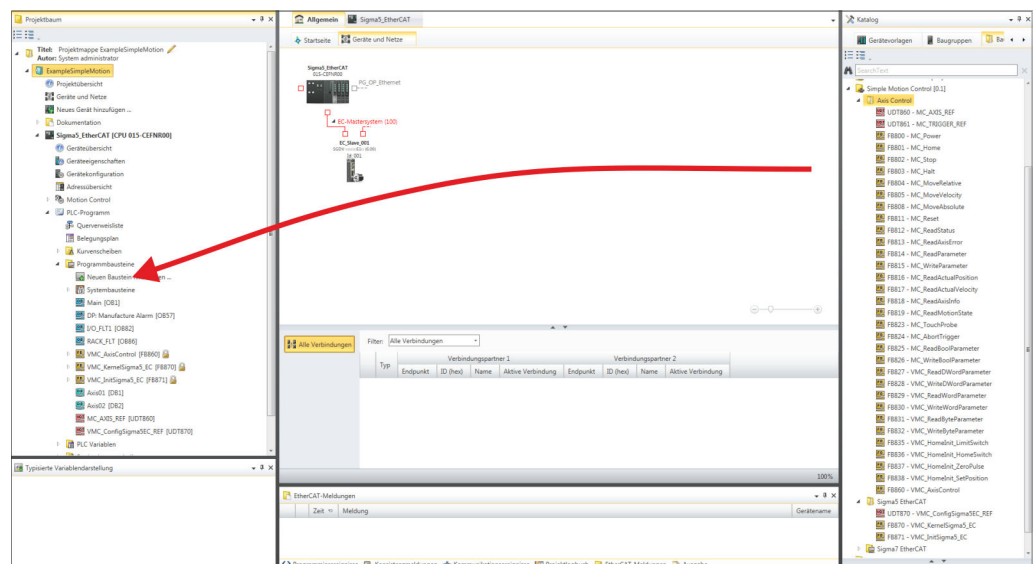
4. Die für Pulse Train erforderlichen Betriebsparameter werden intern auf die entsprechenden Werte angepasst. Lassen Sie alle Werte unverändert.



5. Schließen Sie den Dialog mit [OK].
6. Wählen Sie "Projekt → Alles übersetzen".

4.4.2 Anwender-Programm

Baustein in Projekt kopieren



- Öffnen Sie im "Katalog" unter "Bausteine" "Simple Motion Control" und ziehen Sie per Drag&Drop folgenden Baustein in "Programmbausteine" des Projektbaums:

- Sigma5+7 Pulse Train
 - FB 875 - VMC_AxisControl_PT ↗ Kapitel 4.7.1 "FB 875 - VMC_AxisControl_PT - Achskontrolle über Pulse Train" auf Seite 219

OB 1

Konfiguration der Achse

Sollten Sie einen anderen Kanal als Kanal 0 verwenden, müssen Sie diesen in der Hardware-Konfiguration und in Ihrem Anwenderprogramm entsprechend anpassen.

1. Öffnen Sie im *Projektbaum* innerhalb der CPU unter *"PLC-Programm"*, *"Programmbausteine"* den OB 1 und programmieren Sie den Call FB 875, DB 875.

⇒ Es öffnet sich der Dialog *"Instanz-Datenbaustein hinzufügen"*.

2. Stellen Sie, wenn nicht schon geschehen, die Nummer für den Instanz-Datenbaustein ein und schließen Sie den Dialog mit [OK].

⇒ Der Bausteinaufruf wird angelegt und die Parameter werden aufgelistet

3. Vergeben Sie für das Beispielprojekt folgende Parameter. Beachten Sie hier insbesondere die beiden Umrechnungsfaktoren *FactorPosition* und *FactorVelocity*:

```
⇒ CALL FB      "VMC_AxisControl_PT" , "DI_AxisControl_PT"
      S_ChannelNumberPWM      := 0
      S_Ready                  := E 136.0
      S_Alarm                  := E 136.2
      FactorPosition            := 1024.0
      FactorVelocity            := 976.5625
      AxisEnable                := M 100.1
      AxisReset                 := M 100.2
      StopExecute               := M 100.3
      MvVelocityExecute         := M 100.4
      MvRelativeExecute         := M 100.5
      JogPositive               := M 100.6
      JogNegative               := M 100.7
      PositionDistance          := MD 102
      Velocity                  := MD 106
      S_On                     := A 136.7
      S_Direction               := A 136.2
      S_AlarmReset              := A 136.6
      MinUserDistance           := MD 110
      MaxUserDistance           := MD 114
      MinUserVelocity           := MD 118
      MaxUserVelocity           := MD 122
      AxisReady                 := M 101.3
      AxisEnabled               := M 101.4
      AxisError                 := M 101.5
      AxisErrorID               := MW 126
      DriveError                := M 101.6
      CmdActive                  := MB 128
      CmdDone                    := M 130.0
      CmdBusy                    := M 130.1
      CmdAborted                := M 130.2
      CmdError                  := M 130.3
      CmdErrorID                := MW 132
```

Die Adressen von *S_Ready* und *S_Alarm* ergeben sich aus den Adressen der Eingänge, welche mit den Digitalausgängen des Antrieb verbunden sind. Diese können über das Submodul *"X25 DI/DIO"* der CPU ermittelt werden.

Die Adressen von *S_On*, *S_Direction* und *S_AlarmReset* ergeben sich aus den Adressen der Ausgänge, welche mit den Digitaleingängen des Antriebs verbunden sind. Diese können über das Submodul *"X25 DI/DIO"* der CPU ermittelt werden.

Zeitlicher Ablauf

1. ➤ Wählen Sie *"Projekt ➔ Alles übersetzen"* und übertragen Sie das Projekt in Ihre CPU. Näheres zur Übertragung Ihres Projekt finden Sie in der Onlinehilfe zum *SPEED7 Studio*.

⇒ Sie können jetzt Ihre Applikation in Betrieb nehmen.

**VORSICHT!**

Bitte beachten Sie immer die Sicherheitshinweise zu ihrem Antrieb, insbesondere bei der Inbetriebnahme!

2. ➤ Bringen Sie Ihre CPU in RUN und schalten Sie Ihren Antrieb ein.
⇒ Der FB 875 - VMC_AxisControl_PT wird zyklisch abgearbeitet.
3. ➤ Sobald *AxisReady* = TRUE meldet, können Sie mit *AxisEnable* den Antrieb freigegeben.
4. ➤ Sie haben jetzt die Möglichkeit über die entsprechenden Parameter Ihren Antrieb zu steuern und dessen Status abzufragen. ➔ *Kapitel 4.7.1 "FB 875 - VMC_AxisControl_PT - Achskontrolle über Pulse Train" auf Seite 219*

Steuerung des Antriebs über HMI

Sie haben die Möglichkeit über ein HMI Ihren Antrieb zu steuern. Hierzu gibt es für Movicon eine vorgefertigte Symbolbibliothek für den Zugriff auf den VMC_AxisControl_PT Funktionsbaustein. ➔ *Kapitel 7 "Antrieb über HMI steuern" auf Seite 313*

4.5 Einsatz im Siemens SIMATIC Manager

4.5.1 Voraussetzung

Übersicht

- Bitte verwenden Sie für die Projektierung den Siemens SIMATIC Manager ab V 5.5 SP2.
- Die Projektierung der VIPA CPU mit Pulse Train Funktionalität erfolgt im Siemens SIMATIC Manager in Form eines virtuellen PROFINET IO Devices.
- Das PROFINET IO Device ist mittels GSDML im Hardware-Katalog zu installieren.

VIPA IO Device installieren

Die Installation des PROFINET VIPA IO Device im Hardware-Katalog erfolgt nach folgender Vorgehensweise:

1. ➤ Gehen Sie in den Service-Bereich von www.vipa.com.
2. ➤ Laden Sie aus dem Downloadbereich unter *"Config Dateien ➔ PROFINET"* die Konfigurationsdatei für Ihre CPU.
3. ➤ Extrahieren Sie die Datei in Ihr Arbeitsverzeichnis.
4. ➤ Starten Sie den Hardware-Konfigurator von Siemens.
5. ➤ Schließen Sie alle Projekte.
6. ➤ Gehen Sie auf *"Extras ➔ GSD-Dateien installieren"*.
7. ➤ Navigieren Sie in Ihr Arbeitsverzeichnis und installieren Sie die entsprechende GSDML-Datei.

⇒ Nach der Installation finden Sie das entsprechende PROFINET IO Device unter *"PROFINET IO ➔ Weitere Feldgeräte ➔ I/O ➔ VIPA ..."*.

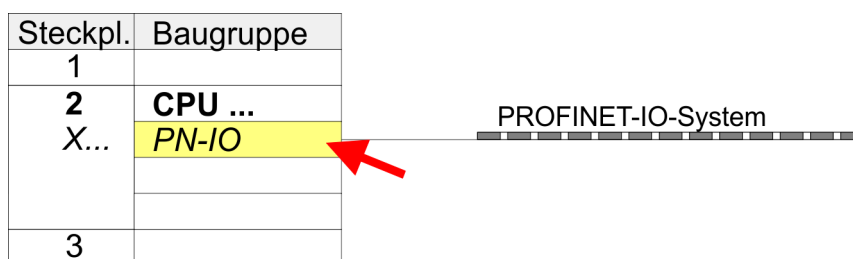
4.5.2 Hardware-Konfiguration

CPU im Projekt anlegen

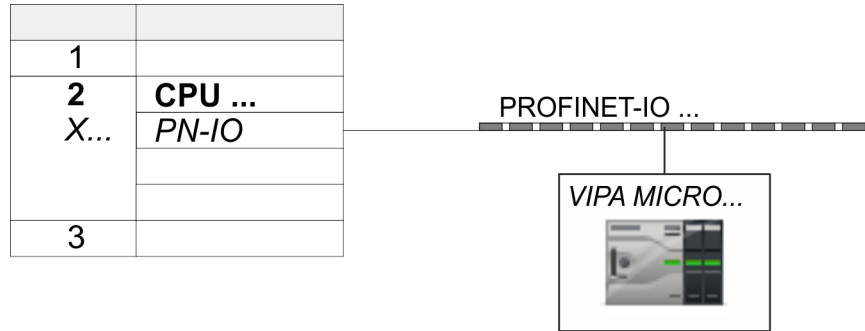
Steckp..	Baugruppe
1	
2	CPU 314C-2PN/DP
X1	MPI/DP
X2	PN-IO
X2...	Port 1
X2...	Port 2
...	...
3	

Um kompatibel mit dem Siemens SIMATIC Manager zu sein, sind folgende Schritte durchzuführen:

1. Starten Sie den Hardware-Konfigurator von Siemens mit einem neuen Projekt.
2. Fügen Sie aus dem Hardware-Katalog eine Profilschiene ein.
3. Platzieren Sie auf "Slot"-Nummer 2 die CPU 314C-2 PN/DP (314-6EH04-0AB0 V3.3).
4. Klicken Sie auf das Submodul "PN-IO" der CPU.
5. Wählen Sie "Kontextmenü → PROFINET IO-System einfügen".



6. Legen Sie mit [Neu] ein neues Subnetz an und vergeben Sie gültige IP-Adress-Daten.
7. Klicken Sie auf das Submodul "PN-IO" der CPU und öffnen Sie mit "Kontextmenü → Objekteigenschaften" den Eigenschafts-Dialog.
8. Geben Sie unter "Allgemein" einen "Gerätenamen" an. Der Geräte name muss eindeutig am Ethernet-Subnetz sein.



0	VIPA MICRO ...	M13-CCF0000	
X2	M13-CCF0000		
1			
2			
3			
...			

9. Navigieren Sie im Hardware-Katalog in das Verzeichnis *"PROFINET IO → Weitere Feldgeräte → I/O → VIPA ..."* und binden Sie z.B. für das System MICRO das IO-Device *"M13-CCF0000"* an Ihr PROFINET-System an.
- ⇒ In der *Steckplatzübersicht* des PROFINET-IO-Device *"VIPA MICRO PLC"* ist auf Steckplatz 0 die CPU bereits vorplatziert.

Ethernet-PG/OP-Kanal parametrieren

Steckpl.	Modul
1	
2	CPU ...
X...	PN-IO
3	
4	343-1EX30
5	
...	

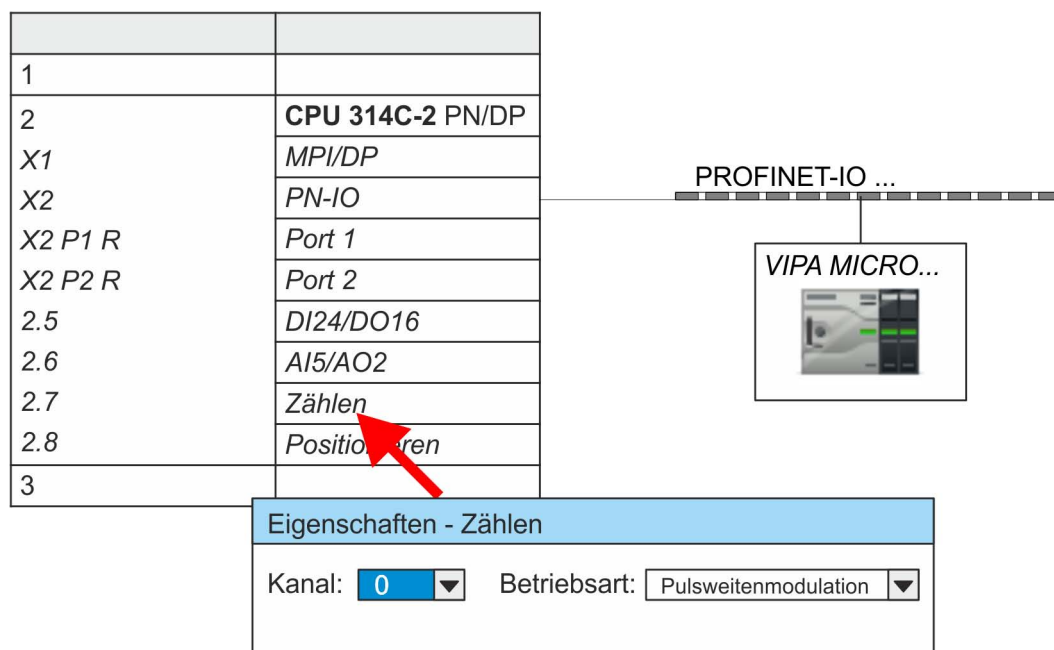
- Platzieren Sie für den Ethernet-PG/OP-Kanal auf Steckplatz 4 den Siemens CP 343-1 (SIMATIC 300 \ CP 300 \ Industrial Ethernet \ CP 343-1 \ 6GK7 343-1EX30 0XE0 V3.0).
- Öffnen Sie durch Doppelklick auf den CP 343-1EX30 den Eigenschaften-Dialog und geben Sie für den CP unter *"Eigenschaften"* IP-Adress-Daten an. Gültige IP-Adress-Parameter erhalten Sie von Ihrem Systemadministrator.
- Ordnen Sie den CP einem *"Subnetz"* zu. Ohne Zuordnung werden die IP-Adress-Daten nicht übernommen!

E/A-Peripherie auf Pulse Train umschalten

Für die Parametrierung der Ein-/Ausgabeperipherie und der *Technologischen Funktionen* sind die entsprechenden Submodule der Siemens CPU 314C-2 PN/DP (314-6EH04-0AB0 V3.3) zu verwenden. Für die Pulse Train Ausgabe ist das Submodul Count auf *"Pulsweitenmodulation"* umzuschalten. Sollten Sie einen anderen Kanal als Kanal 0 verwenden, müssen Sie diesen in der Hardware-Konfiguration und in Ihrem Anwenderprogramm entsprechend anpassen.

- Doppelklicken Sie auf das Zähler-Submodul der Siemens CPU 314C-2 PN/DP.
⇒ Sie gelangen in das Dialogfeld *"Eigenschaften"*.
- Wählen Sie z.B. *"Kanal 0"* und stellen Sie unter *"Betriebsart"* die Funktion *"Pulsweitenmodulation"* ein.

3. ➤ Lassen Sie alle Werte unverändert.



4. ➤ Schließen Sie den Dialog mit [OK].

5. ➤ Wählen Sie "Station ➔ Speichern und übersetzen".

6. ➤ Schließen Sie den Hardware-Konfigurator.

4.5.3 Anwender-Programm

Bibliothek einbinden

1. ➤ Gehen Sie in den Service-Bereich von www.vipa.com.
2. ➤ Laden Sie aus dem Downloadbereich unter "VIPA Lib" die *Simple Motion Control Library*.
3. ➤ Öffnen Sie mit "Datei ➔ Dearchivieren" das Dialogfenster zur Auswahl der ZIP-Datei.
4. ➤ Wählen Sie die entsprechende ZIP-Datei an und klicken Sie auf [Öffnen].
5. ➤ Geben Sie ein Zielverzeichnis an, in dem die Bausteine abzulegen sind und starten Sie den Entpackvorgang mit [OK].

Bausteine in Projekt kopieren

- Öffnen Sie die Bibliothek nach dem Entpackvorgang und ziehen Sie per Drag&Drop folgenden Baustein in "Bausteine" Ihres Projekts:
 - Sigma5+7 Pulse Train
 - FB 875 - VMC_AxisControl_PT ↗ Kapitel 4.7.1 "FB 875 - VMC_AxisControl_PT - Achskontrolle über Pulse Train" auf Seite 219

OB 1

Konfiguration der Achse

1. ➤ Öffnen Sie den OB 1 und programmieren Sie den Call FB 875, DB 875.
 - ⇒ Der Bausteinaufruf wird angelegt und die Parameter werden aufgelistet

- 2.** ➤ Vergeben Sie für das Beispielprojekt folgende Parameter. Beachten Sie hier insbesondere die beiden Umrechnungsfaktoren *FactorPosition* und *FactorVelocity*:

```
⇒ CALL FB      "VMC_AxisControl_PT" , "DI_AxisControl_PT"
      S_ChannelNumberPWM      := 0
      S_Ready                 := E 136.0
      S_Alarm                 := E 136.2
      FactorPosition          := 1024.0
      FactorVelocity          := 976.5625
      AxisEnable              := M 100.1
      AxisReset               := M 100.2
      StopExecute             := M 100.3
      MvVelocityExecute       := M 100.4
      MvRelativeExecute       := M 100.5
      JogPositive             := M 100.6
      JogNegative             := M 100.7
      PositionDistance        := MD 102
      Velocity                := MD 106
      S_On                   := A 136.7
      S_Direction             := A 136.2
      S_AlarmReset            := A 136.6
      MinUserDistance         := MD 110
      MaxUserDistance         := MD 114
      MinUserVelocity         := MD 118
      MaxUserVelocity         := MD 122
      AxisReady               := M 101.3
      AxisEnabled             := M 101.4
      AxisError               := M 101.5
      AxisErrorID             := MW 126
      DriveError              := M 101.6
      CmdActive               := MB 128
      CmdDone                 := M 130.0
      CmdBusy                 := M 130.1
      CmdAborted              := M 130.2
      CmdError                := M 130.3
      CmdErrorID              := MW 132
```

Die Adressen von *S_Ready* und *S_Alarm* ergeben sich aus den Adressen der Eingänge, welche mit den Digitalausgängen des Antrieb verbunden sind. Diese können über das Submodul "DI24/DO16" der CPU ermittelt werden.

Die Adressen von *S_On*, *S_Direction* und *S_AlarmReset* ergeben sich aus den Adressen der Ausgänge, welche mit den Digitaleingängen des Antriebs verbunden sind. Diese können über das Submodul "DI24/DO16" der CPU ermittelt werden.

Zeitlicher Ablauf

- 1.** ➤ Wechseln Sie in den Siemens SIMATIC Manager und übertragen Sie Ihr Projekt in die CPU.

⇒ Sie können jetzt Ihre Applikation in Betrieb nehmen.



VORSICHT!

Bitte beachten Sie immer die Sicherheitshinweise zu ihrem Antrieb, insbesondere bei der Inbetriebnahme!

- 2.** ➤ Bringen Sie Ihre CPU in RUN und schalten Sie Ihren Antrieb ein.
⇒ Der FB 875 - VMC_AxisControl_PT wird zyklisch abgearbeitet.
- 3.** ➤ Sobald *AxisReady* = TRUE meldet, können Sie mit *AxisEnable* den Antrieb freigeben.

4. Sie haben jetzt die Möglichkeit über die entsprechenden Parameter Ihren Antrieb zu steuern und dessen Status abzufragen. *↪ Kapitel 4.7.1 "FB 875 - VMC_AxisControl_PT - Achskontrolle über Pulse Train" auf Seite 219*

Steuerung des Antriebs über HMI

Sie haben die Möglichkeit über ein HMI Ihren Antrieb zu steuern. Hierzu gibt es für Movicon eine vorgefertigte Symbolbibliothek für den Zugriff auf den VMC_AxisControl_PT Funktionsbaustein. *↪ Kapitel 7 "Antrieb über HMI steuern" auf Seite 313*

4.6 Einsatz im Siemens TIA Portal

4.6.1 Voraussetzung

Übersicht

- Bitte verwenden Sie für die Projektierung das Siemens TIA Portal ab V 14.
- Die Projektierung der VIPA CPU mit Pulse Train Funktionalität erfolgt im Siemens TIA Portal in Form eines virtuellen PROFINET IO Devices.
- Das PROFINET IO Device ist mittels GSDML im Hardware-Katalog zu installieren.

VIPA IO Device installieren

Die Installation des PROFINET VIPA IO Device im Hardware-Katalog erfolgt nach folgender Vorgehensweise:

1. Gehen Sie in den Service-Bereich von www.vipa.com.
2. Laden Sie aus dem Downloadbereich unter *"Config Dateien → PROFINET"* die entsprechende Datei für Ihr System - hier System MICRO.
3. Extrahieren Sie die Datei in Ihr Arbeitsverzeichnis.
4. Starten das Siemens TIA Portal.
5. Schließen Sie alle Projekte.
6. Wechseln Sie in die *Projektansicht*.
7. Gehen Sie auf *"Extras → Gerätebeschreibungsdatei (GSD) installieren"*.
8. Navigieren Sie in Ihr Arbeitsverzeichnis und installieren Sie die entsprechende GSDML-Datei.

⇒ Nach der Installation wird der Hardware-Katalog aktualisiert und das Siemens TIA Portal beendet.

Nach einem Neustart des Siemens TIA Portals finden Sie das entsprechende PROFINET-IO-Device unter *Weitere Feldgeräte > PROFINET > IO > VIPA GmbH > VIPA MICRO PLC*.



Damit die VIPA-Komponenten angezeigt werden können, müssen Sie im Hardware-Katalog bei "Filter" den Haken entfernen.

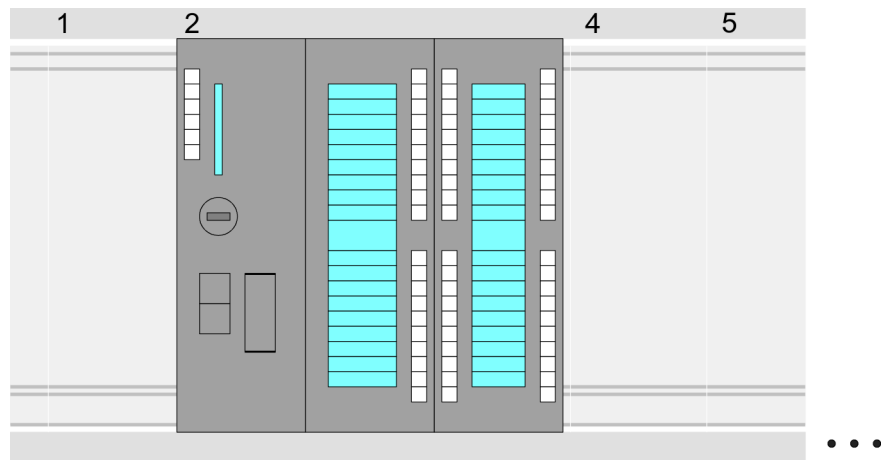
4.6.2 Hardware-Konfiguration

CPU im Projekt anlegen

Um kompatibel mit dem Siemens TIA Portal zu sein, sind folgende Schritte durchzuführen:

1. Starten Sie das Siemens TIA Portal mit einem neuen Projekt.
2. Wechseln Sie in die *Projektansicht*.
3. Klicken Sie in der *Projektnavigation* auf *"Neues Gerät hinzufügen"*.

4. Wählen Sie im Eingabedialog folgende CPU aus:
SIMATIC S7-300 > CPU 314C-2 PN/DP (314-6EH04-0AB0 V3.3)
⇒ Die CPU wird mit einer Profilschiene eingefügt.

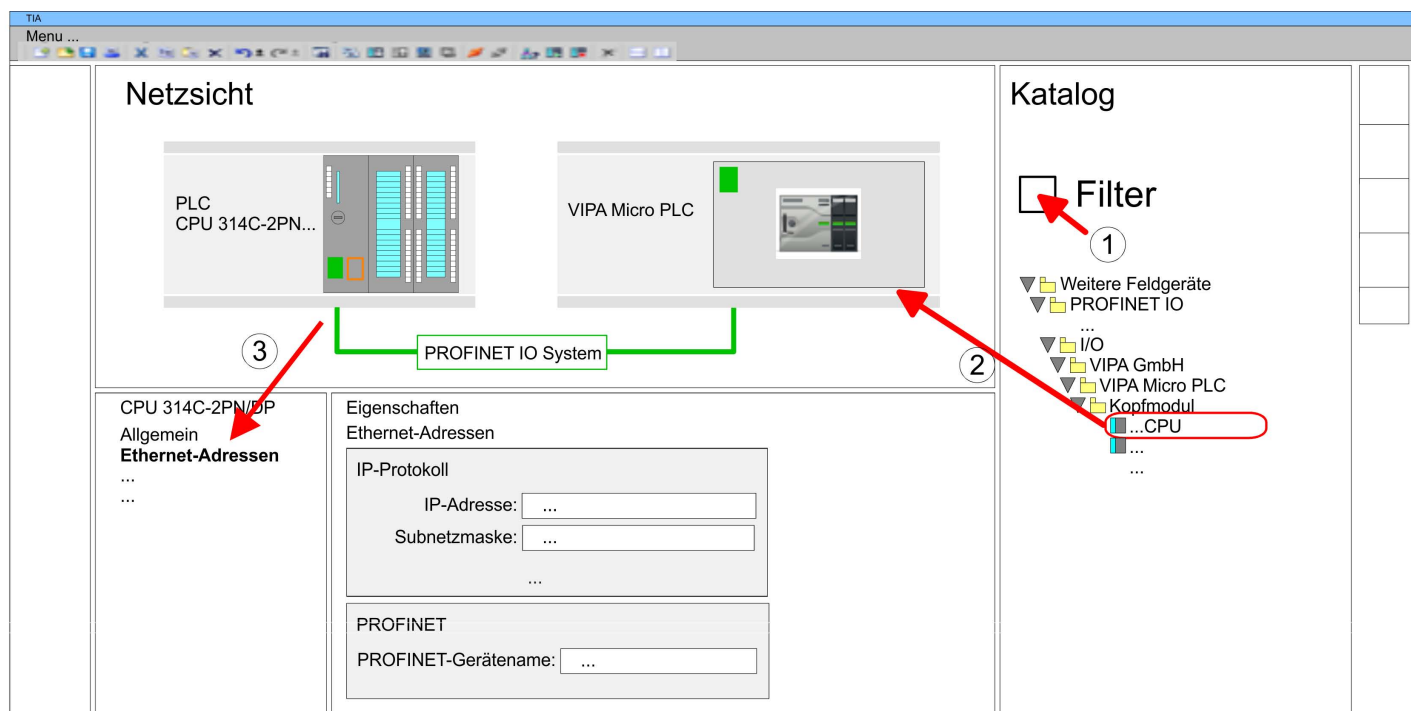


Geräteübersicht:

Baugruppe	...	Steckplatz	...	Typ	...
PLC...		2		CPU 314C-2PN/DP	
MPI-Schnittstelle...		2 X1		MPI/DP-Schnittstelle	
PROFINET-Schnitt...		2 X2		PROFINET-Schnittstelle	
DI24/DO16...		2 5		DI24/DO16	
AI5/AO2...		2 6		AI5/AO2	
Zählen...		2 7		Zählen	
...					

Anbindung CPU als PROFINET-IO-Device

1. Wechseln Sie im *Projektbereich* in die *"Netzsicht"*.
2. Nach der Installation der GSDML finden Sie das IO-Device für die CPU im Hardware-Katalog unter *Weitere Feldgeräte > PROFINET > IO > VIPA GmbH > VIPA MICRO PLC*. Binden Sie das Slave-System an die CPU an, indem Sie dies aus dem Hardware-Katalog in die *Netzsicht* ziehen und dieses über PROFINET an die CPU anbinden.
3. Klicken Sie in der *Netzsicht* auf den PROFINET-Teil der Siemens CPU und geben Sie in *"Eigenschaften"* unter *"Ethernet-Adressen"* im Bereich *"IP-Protokoll"* gültige IP-Adressdaten an.
4. Geben Sie unter *"PROFINET"* einen *"PROFINET Gerätenamen"* an. Der Geräte-name muss eindeutig am Ethernet-Subnetz sein.

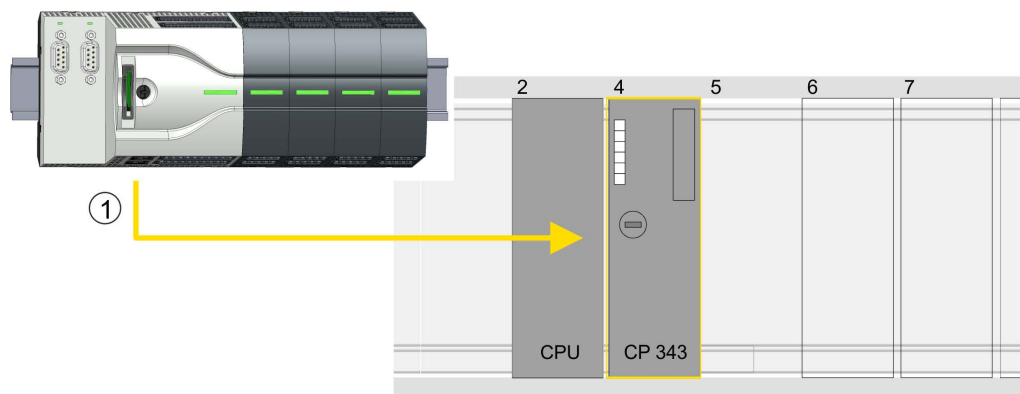


5. ➤ Wählen Sie in der *Netzwerkansicht* das IO-Device "*VIPA MICRO PLC*" an und wechseln Sie in die *Geräteübersicht*.

⇒ In der *Geräteübersicht* des PROFINET-IO-Device "*VIPA MICRO PLC*" ist auf Steckplatz 0 die CPU bereits vorplatziert.

Ethernet-PG/OP-Kanal parametrieren

- Platzieren Sie für den Ethernet-PG/OP-Kanal auf Steckplatz 4 den Siemens CP 343-1 (6GK7 343-1EX30 0XE0 V3.0).
- Öffnen Sie durch Klick auf den CP 343-1EX30 den "Eigenschaften"-Dialog und geben Sie für den CP in den "Eigenschaften" unter "Ethernet-Adresse" IP-Adress-Daten an. Gültige IP-Adress-Parameter erhalten Sie von Ihrem Systemadministrator.



1 Ethernet-PG/OP-Kanal

Geräteübersicht

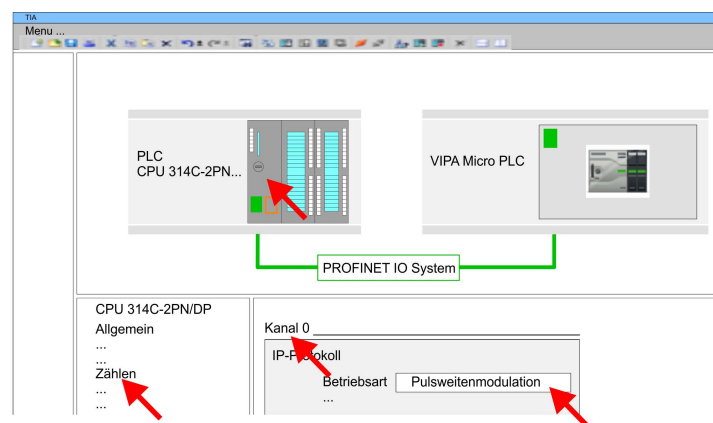
Baugruppe	...	Steckplatz	...	Typ	...
PLC ...		2		CPU 314C-2PN/DP	

MPI/DP-Schnittstelle		2 X1		MPI/DP-Schnittstelle	
PROFINET-Schnittstelle		2 X2		PROFINET-Schnittstelle	
...		
CP 343-1		4		CP 343-1	
...		

E/A-Peripherie auf Pulse Train umschalten

Für die Parametrierung der Ein-/Ausgabeperipherie und der *Technologischen Funktionen* sind die entsprechenden Submodule der Siemens CPU 314C-2 PN/DP (314-6EH04-0AB0 V3.3) zu verwenden. Für die Pulse Train Ausgabe ist das Submodul Count auf "Pulsweitenmodulation" umzuschalten. Sollten Sie einen anderen Kanal als Kanal 0 verwenden, müssen Sie diesen in der Hardware-Konfiguration und in Ihrem Anwenderprogramm entsprechend anpassen.

1. ➤ Doppelklicken Sie auf das Zähler-Submodul der Siemens CPU 314C-2 PN/DP.
⇒ Sie gelangen in das Dialogfeld "Eigenschaften".
2. ➤ Wählen Sie z.B. "Kanal 0" und stellen Sie unter "Betriebsart" die Funktion "Pulsweitenmodulation" ein.
3. ➤ Lassen Sie alle Werte unverändert.



4. ➤ Klicken Sie in der Projektnavigation auf Ihre CPU und wählen Sie "Kontextmenü ➔ Übersetzen ➔ Alles".

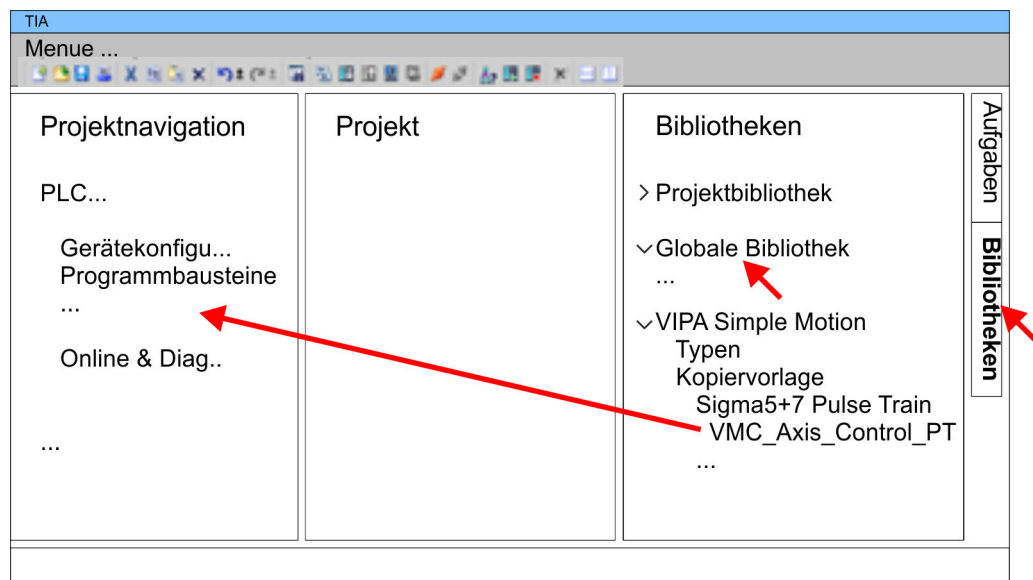
4.6.3 Anwender-Programm

Bibliothek einbinden

1. ➤ Gehen Sie in den Service-Bereich von www.vipa.com.
2. ➤ Laden Sie aus dem Downloadbereich unter "VIPA Lib" die *Simple Motion Control Library*.
Die Bibliothek liegt für die entsprechende TIA Portal Version als gepackte zip-Datei vor.
3. ➤ Starten Sie mit einem Doppelklick auf die Datei ...TIA_Vxx.zip ihr Unzip-Programm, entpacken Sie Dateien und Ordner in ein Arbeitsverzeichnis für das Siemens TIA Portal.
4. ➤ Wechseln Sie im Siemens TIA Portal in die *Projektansicht*.
5. ➤ Wählen Sie auf der rechten Seite die Task-Card "Bibliotheken".

6. ➔ Klicken Sie auf "Globale Bibliothek".
7. ➔ Klicken Sie innerhalb der "*Globalen Bibliothek*" auf die freie Fläche und wählen Sie "*Kontextmenü ➔ Bibliothek dearchivieren*".
8. ➔ Navigieren Sie zu ihrem Arbeitsverzeichnis und laden Sie die Datei ...Simple Motion.zalxx.

Bausteine in Projekt kopieren



- ➔ Kopieren Sie folgenden Baustein aus der Bibliothek in das Verzeichnis "Programmbausteine" in der *Projektnavigation* Ihres Projekts.

- **Sigma5+7 Pulse Train**
 - FB 875 - VMC_AxisControl_PT ➔ Kapitel 4.7.1 "FB 875 - VMC_Axis-Control_PT - Ächskontrolle über Pulse Train" auf Seite 219

OB 1

Konfiguration der Achse

1. ➔ Öffnen Sie in der *Projektnavigation* innerhalb der CPU unter "*Programmbausteine*" den OB 1 und programmieren Sie den Call FB 875, DB 875
 - ⇒ Es öffnet sich der Dialog "*Instanz-Datenbaustein hinzufügen*".
2. ➔ Stellen Sie, wenn nicht schon geschehen, die Nummer für den Instanz-Datenbaustein ein und schließen Sie den Dialog mit [OK].
 - ⇒ Der Bausteinaufruf wird angelegt und die Parameter werden aufgelistet

3. ➔ Vergeben Sie für das Beispielprojekt folgende Parameter. Beachten Sie hier insbesondere die beiden Umrechnungsfaktoren *FactorPosition* und *FactorVelocity*:

```
⇒ CALL FB      "VMC_AxisControl_PT" , "DI_AxisControl_PT"
      S_ChannelNumberPWM      := 0
      S_Ready                 := E 136.0
      S_Alarm                 := E 136.2
      FactorPosition          := 1024.0
      FactorVelocity          := 976.5625
      AxisEnable              := M 100.1
      AxisReset               := M 100.2
      StopExecute             := M 100.3
      MvVelocityExecute       := M 100.4
      MvRelativeExecute       := M 100.5
      JogPositive             := M 100.6
      JogNegative             := M 100.7
      PositionDistance        := MD 102
      Velocity                := MD 106
      S_On                    := A 136.7
      S_Direction             := A 136.2
      S_AlarmReset            := A 136.6
      MinUserDistance         := MD 110
      MaxUserDistance         := MD 114
      MinUserVelocity         := MD 118
      MaxUserVelocity         := MD 122
      AxisReady               := M 101.3
      AxisEnabled             := M 101.4
      AxisError               := M 101.5
      AxisErrorID             := MW 126
      DriveError              := M 101.6
      CmdActive               := MB 128
      CmdDone                 := M 130.0
      CmdBusy                 := M 130.1
      CmdAborted              := M 130.2
      CmdError                := M 130.3
      CmdErrorID              := MW 132
```

Die Adressen von *S_Ready* und *S_Alarm* ergeben sich aus den Adressen der Eingänge, welche mit den Digitalausgängen des Antrieb verbunden sind. Diese können über das Submodul "DI24/DO16" der CPU ermittelt werden.

Die Adressen von *S_On*, *S_Direction* und *S_AlarmReset* ergeben sich aus den Adressen der Ausgänge, welche mit den Digitaleingängen des Antriebs verbunden sind. Diese können über das Submodul "DI24/DO16" der CPU ermittelt werden.

Zeitlicher Ablauf

1. ➔ Wählen Sie "Bearbeiten ➔ Übersetzen" und übertragen Sie das Projekt in Ihre CPU. Näheres zur Übertragung Ihres Projekt finden Sie in der Onlinehilfe zum Siemens TIA Portal.

⇒ Sie können jetzt Ihre Applikation in Betrieb nehmen.



VORSICHT!

Bitte beachten Sie immer die Sicherheitshinweise zu ihrem Antrieb, insbesondere bei der Inbetriebnahme!

2. ➔ Bringen Sie Ihre CPU in RUN und schalten Sie Ihren Antrieb ein.
⇒ Der FB 875 - VMC_AxisControl_PT wird zyklisch abgearbeitet
3. ➔ Sobald *AxisReady* = TRUE meldet, können Sie mit *AxisEnable* den Antrieb freigeben.

4. ➔ Sie haben jetzt die Möglichkeit über die entsprechenden Parameter Ihren Antrieb zu steuern und dessen Status abzufragen. ➔ *Kapitel 4.7.1 "FB 875 - VMC_AxisControl_PT - Achskontrolle über Pulse Train" auf Seite 219*

Steuerung des Antriebs über HMI

Sie haben die Möglichkeit über ein HMI Ihren Antrieb zu steuern. Hierzu gibt es für Movicon eine vorgefertigte Symbolbibliothek für den Zugriff auf den VMC_AxisControl_PT Funktionsbaustein. ➔ *Kapitel 7 "Antrieb über HMI steuern" auf Seite 313*

4.7 Antriebsspezifischer Baustein

4.7.1 FB 875 - VMC_AxisControl_PT - Achskontrolle über Pulse Train

Beschreibung

Mit dem FB *VMC_AxisControl_PT* können Sie eine über Pulse Train angebundene Achse steuern. Sie können den Status des Antriebs abrufen, den Antrieb ein- bzw. ausschalten oder verschiedene Bewegungskommandos ausführen. In den Instanzdaten des Bausteins befindet sich ein gesonderter Speicherbereich. Über diesen können Sie mittels eines HMI Ihre Achse steuern. ➔ *Kapitel 7 "Antrieb über HMI steuern" auf Seite 313*



Die Ansteuerung eines Pulse Train Antriebs erfolgt ausschließlich mit dem FB 875 *VMC_AxisControl_PT*. PLCopen-Bausteine werden nicht unterstützt!

Parameter

Parameter	Deklaration	Datentyp	Beschreibung
S_Channel-NumberPWM	INPUT	INT	Kanalnummer vom PWM-Ausgang, der für die Ansteuerung vom Pulse-Train-Eingang des Servos (Signal PULS) verwendet wird.
S_Ready	INPUT	BOOL	<ul style="list-style-type: none"> Digitaleingang zur Anbindung des S_Ready-Signals (S-RDY) <ul style="list-style-type: none"> TRUE: Servo ist bereit für das S_On-Signal.
S_Alarm	INPUT	BOOL	<ul style="list-style-type: none"> Digitaleingang zur Anbindung des S_Alarm-Signals (ALM) <ul style="list-style-type: none"> FALSE, wenn der Servo einen Fehler erkannt hat.
FactorPosition	INPUT	REAL	Faktor zur Umrechnung der Position von Benutzer- in Antriebseinheiten (Inkrement) und zurück. ➔ <i>"FactorPosition" auf Seite 222</i>
FactorVelocity	INPUT	REAL	Faktor zur Umrechnung der Geschwindigkeit von Benutzer- in Antriebseinheiten (Inkrement/s) und zurück. ➔ <i>"FactorVelocity" auf Seite 223</i>
AxisEnable	INPUT	BOOL	<ul style="list-style-type: none"> Achsenfreigabe <ul style="list-style-type: none"> TRUE: Die Achse wird freigegeben. FALSE: Die Achse wird gesperrt.
AxisReset	INPUT	BOOL	<ul style="list-style-type: none"> Reset Achse <ul style="list-style-type: none"> Flanke 0-1: Reset der Achse wird durchgeführt. Der Zustand eines mit <i>AxisReset</i> gestarteten Reset wird nicht an den Ausgängen <i>CmdActive</i>, <i>CmdDone</i>, <i>CmdBusy</i>, <i>CmdAborted</i>, <i>CmdError</i> und <i>CmdErrorID</i> ausgegeben.
StopExecute	INPUT	BOOL	<ul style="list-style-type: none"> Achse stoppen <ul style="list-style-type: none"> Flanke 0-1: Stoppen der Achse wird gestartet. <p>Hinweis: StopExecute = 1: Kein anderes Kommando kann gestartet werden!</p>

Parameter	Deklaration	Datentyp	Beschreibung
MvVelocityExecute	INPUT	BOOL	<ul style="list-style-type: none"> ■ Verfahren der Achse starten <ul style="list-style-type: none"> – Flanke 0-1: Die Achse wird auf die angegebene Geschwindigkeit beschleunigt / abgebremst.
MvRelativeExecute	INPUT	BOOL	<ul style="list-style-type: none"> ■ Verfahren der Achse starten <ul style="list-style-type: none"> – Flanke 0-1: Die relative Positionierung der Achse wird gestartet.
JogPositive	INPUT	BOOL	<p>Tipp-Betrieb positiv</p> <ul style="list-style-type: none"> ■ Achse mit konstanter Geschwindigkeit in positive Richtung verfahren <ul style="list-style-type: none"> – Flanke 0-1: Das Verfahren der Achse mit konstanter Geschwindigkeit wird gestartet. – Flanke 1-0: Die Achse wird gestoppt.
JogNegative	INPUT	BOOL	<p>Tipp-Betrieb negativ</p> <ul style="list-style-type: none"> ■ Achse mit konstanter Geschwindigkeit in negative Richtung verfahren <ul style="list-style-type: none"> – Flanke 0-1: Das Verfahren der Achse mit konstanter Geschwindigkeit wird gestartet. – Flanke 1-0: Die Achse wird gestoppt.
PositionDistance	INPUT	REAL	Absolute Position bzw. relative Wegstrecke für <i>MvRelativeExecute</i> [Anwendereinheiten].
Velocity	INPUT	REAL	Geschwindigkeitsvorgabe (vorzeichenbehafteter Wert) in [Anwendereinheiten/s].
S_On	OUTPUT	BOOL	<ul style="list-style-type: none"> ■ Digitalausgang zur Ansteuerung des S_On-Signals (S-ON) <ul style="list-style-type: none"> – TRUE: schaltet den Servo ein. – FALSE: schaltet den Servo aus.
S_Direction	OUTPUT	BOOL	<ul style="list-style-type: none"> ■ Digitalausgang zur Ansteuerung des S_Direction-Signals (SIGN) <ul style="list-style-type: none"> – TRUE: Vorgabe der Drehrichtung positive Richtung für den Servo. – FALSE: Vorgabe der Drehrichtung negative Richtung für den Servo.
S_AlarmReset	OUTPUT	BOOL	<ul style="list-style-type: none"> ■ Digitalausgang zur Ansteuerung des S_AlarmReset-Signals (ALM-RST) <ul style="list-style-type: none"> – TRUE: Alarime werden im Servo zurückgesetzt. – FALSE: Alarime im Servo bleiben bestehen.
MinUserDistance	OUTPUT	REAL	Minimaler Verfahrweg (1 Inkrement) des Servos [Anwendereinheiten].
MaxUserDistance	OUTPUT	REAL	Maximaler Verfahrweg (8388607 Inkremente = maximale Anzahl Impulse des PWM-Ausgangs) des Servos [Anwendereinheiten].
MinUserVelocity	OUTPUT	REAL	Minimale Geschwindigkeit (Periodendauer = 65535µs = maximale Periodendauer des PWM-Ausgangs) des Servos [Anwendereinheiten].
MaxUserVelocity	OUTPUT	REAL	Maximale Geschwindigkeit (Periodendauer = 20µs = minimale Periodendauer des PWM-Ausgangs) des Servos [Anwendereinheiten].

Parameter	Deklaration	Datentyp	Beschreibung
AxisReady	OUTPUT	BOOL	<ul style="list-style-type: none"> ■ AxisReady <ul style="list-style-type: none"> – TRUE: Die Achse ist einschaltbereit. – FALSE: Die Achse ist nicht einschaltbereit. → Prüfe und behebe <i>AxisError</i> (siehe <i>AxisErrorID</i>). → Prüfe und behebe <i>DriveError</i>.
AxisEnabled	OUTPUT	BOOL	<ul style="list-style-type: none"> ■ Status Achse <ul style="list-style-type: none"> – TRUE: Achse ist eingeschaltet und nimmt Bewegungsaufträge an. – FALSE: Achse ist nicht eingeschaltet und nimmt keine Bewegungsaufträge an. ■ Bedingungen für <i>AxisEnabled</i> = TRUE <ul style="list-style-type: none"> – <i>AxisEnable</i> = TRUE – <i>S_Ready</i> = TRUE – <i>S_Alarm</i> = TRUE
AxisError	OUTPUT	BOOL	<ul style="list-style-type: none"> ■ Fehler bei Motion Achse <ul style="list-style-type: none"> – TRUE: Ein Fehler ist aufgetreten. <p>Zusätzliche Fehlerinformationen können dem Parameter <i>AxisErrorID</i> entnommen werden.</p> <p>→ Die Achse wird gesperrt (<i>S_On</i> = FALSE und <i>AxisEnabled</i> = FALSE). Kommando wird nicht ausgeführt.</p>
AxisErrorID	OUTPUT	WORD	<p>Zusätzliche Fehlerinformationen</p> <p>↪ <i>Kapitel 9 "ErrorID - Zusätzliche Fehlerinformationen" auf Seite 340</i></p>
DriveError	OUTPUT	BOOL	<ul style="list-style-type: none"> ■ Fehler direkt am Antrieb <ul style="list-style-type: none"> – TRUE: Ein Fehler ist aufgetreten. – → Die Achse wird gesperrt.
CmdActive	OUTPUT	BYTE	<ul style="list-style-type: none"> ■ Kommando <ul style="list-style-type: none"> – 0: kein Cmd aktiv – 1: STOP – 2: MvVelocity – 3: MvRelative – 4: JogPos – 5: JogNeg
CmdDone	OUTPUT	BOOL	<ul style="list-style-type: none"> ■ Status Done <ul style="list-style-type: none"> – TRUE: Auftrag wurde ohne Fehler beendet.
CmdBusy	OUTPUT	BOOL	<ul style="list-style-type: none"> ■ Status Busy <ul style="list-style-type: none"> – TRUE: Auftrag ist in Bearbeitung.
CmdAborted	OUTPUT	BOOL	<ul style="list-style-type: none"> ■ Status Aborted <ul style="list-style-type: none"> – TRUE: Der Auftrag wurde während der Bearbeitung von einem anderen Auftrag abgebrochen. <p>Hinweis: <i>CmdAborted</i> wird beim Start eines Cmd zurückgesetzt</p>

Parameter	Deklaration	Datentyp	Beschreibung
CmdError	OUTPUT	BOOL	<div> <div>■</div> Status Error <ul style="list-style-type: none"> – TRUE: Ein Fehler ist aufgetreten. Die Achse wird gesperrt </div> Zusätzliche Fehlerinformationen können dem Parameter <i>CmdErrorID</i> entnommen werden.
CmdErrorID	OUTPUT	WORD	Zusätzliche Fehlerinformationen 🔗 <i>Kapitel 9 "ErrorID - Zusätzliche Fehlerinformationen" auf Seite 340</i>

4.7.1.1 Umrechnungsfaktoren

FactorPosition



Die Berechnung von *FactorPosition* ist nur gültig, wenn der Servo-Parameter *Reference Pulse Multiplier (Pn218)* = 1 gesetzt ist.

$$FactorPosition = \frac{Resolution}{Numerator} \cdot Denominator$$

FactorPosition - Faktor zur Umrechnung der Position von Benutzer- in Antriebseinheiten (Inkrement) und zurück.

Resolution - Anzahl Inkremente pro Anwendereinheit

Numerator - Zähler: Electronic Gear Ratio (Pn20E) der Servo-Parameter

Denominator - Nenner: Electronic Gear Ratio (Pn210) der Servo-Parameter

Beispiel Anwendereinheit für Position = 1 Umdrehung

FactorPosition - Faktor zur Umrechnung der Position von Benutzer- in Antriebseinheiten (Inkrement) und zurück.

Resolution - Anzahl Inkremente pro Anwendereinheit

$$Resolution = 2^{20} = 1048576$$

Numerator - Zähler: Electronic Gear Ratio (Pn20E) der Servo-Parameter

$$Numerator = 1024$$

Denominator - Nenner: Electronic Gear Ratio (Pn210) der Servo-Parameter

$$Denominator = 1$$

$$FactorPosition = \frac{Resolution}{Numerator} \cdot Denominator$$

$$FactorPosition = \frac{1048576}{1024} \cdot 1 = 1024$$

Beispiel minimaler Verfahrenweg

MinPos - Minimaler Verfahrenweg in Umdrehungen

Resolution - Anzahl Inkremente pro Anwendereinheit

$$Resolution = 2^{20} = 1048576$$

Numerator - Zähler: Electronic Gear Ratio (Pn20E) der Servo-Parameter

$$Numerator = 1024$$

Period - Minimale Periode

$$Period = 1$$

$$MinPos = Numerator \cdot \frac{Period}{Resolution}$$

$$MinPos = 1024 \cdot \frac{1}{1048576} = \frac{1}{1024}$$

Beispiel maximaler Verfahrenweg

MaxPos - Maximaler Verfahrenweg in Umdrehungen

Resolution - Anzahl Inkremente pro Anwendereinheit

$$Resolution = 2^{20} = 1048576$$

Numerator - Zähler: Electronic Gear Ratio (Pn20E) der Servo-Parameter

$$Numerator = 1024$$

Period - Maximale Periode

$$Period = 8388607$$

$$MaxPos = Numerator \cdot \frac{Period}{Resolution}$$

$$MaxPos = 1024 \cdot \frac{8388607}{1048576} = 8192$$

FactorVelocity

Die Berechnung von FactorVelocity ist nur gültig, wenn der Servo-Parameter Reference Pulse Multiplier (Pn218) = 1 gesetzt ist.

$$FactorVelocity = Time \cdot \frac{\frac{Numerator}{Denominator}}{Resolution}$$

Time - Zeit für 1 Umdrehung in µs

Numerator - Zähler: Electronic Gear Ratio (Pn20E) der Servo-Parameter

Denominator - Nenner: Electronic Gear Ratio (Pn210) der Servo-Parameter

Resolution - Anzahl Inkremente pro Anwendereinheit

Beispiel Anwendereinheit für Geschwindigkeit = Umdrehung/min

FactorVelocity - Faktor zur Umrechnung von Benutzer- in Antriebseinheiten (Inkrement/ sec) und zurück.

Time - Zeit für 1 Umdrehung in μs

$$Time = 1\text{min} = 60 \cdot 10^6 \mu\text{s}$$

Numerator - Zähler: Electronic Gear Ratio (Pn20E) der Servo-Parameter

$$Numerator = 1024$$

Denominator - Nenner: Electronic Gear Ratio (Pn210) der Servo-Parameter

$$Denominator = 1$$

Resolution - Anzahl Inkremente pro Anwendereinheit

$$Resolution = 2^{20} = 1048576$$

$$FactorVelocity = Time \cdot \frac{\frac{Numerator}{Denominator}}{Resolution}$$

$$FactorVelocity = 60 \cdot 10^6 \cdot \frac{\frac{1024}{1}}{1048576} = \frac{60 \cdot 10^6}{1024} = 58593,75$$

Beispiel Anwendereinheit für Geschwindigkeit = Umdrehung/s

FactorVelocity - Faktor zur Umrechnung von Benutzer- in Antriebseinheiten (Inkrement/ sec) und zurück.

Time - Zeit für 1 Umdrehung in μs

$$Time = 1\text{s} = 10^6 \mu\text{s}$$

Numerator - Zähler: Electronic Gear Ratio (Pn20E) der Servo-Parameter

$$Numerator = 1024$$

Denominator - Nenner: Electronic Gear Ratio (Pn210) der Servo-Parameter

$$Denominator = 1$$

Resolution - Anzahl Inkremente pro Anwendereinheit

$$Resolution = 2^{20} = 1048576$$

$$FactorVelocity = Time \cdot \frac{\frac{Numerator}{Denominator}}{Resolution}$$

$$FactorVelocity = 10^6 \cdot \frac{\frac{1024}{1}}{1048576} = \frac{10^6}{1024} = 976,5625$$

Minimale Geschwindigkeit für Umdrehungen/min

MinVel - Minimale Geschwindigkeit in Umdrehungen/min

FactorVelocity - Faktor zur Umrechnung von Benutzer- in Antriebseinheiten (Inkrement/s) und zurück.

$$MinVel = \frac{FactorVelocity}{65535} = \frac{58593,75}{65535} = 0,89$$

Maximale Geschwindigkeit für Umdrehungen/min

MaxVel - Maximale Geschwindigkeit in Umdrehungen/min

FactorVelocity - Faktor zur Umrechnung von Benutzer- in Antriebseinheiten (Inkrement/s) und zurück.

$$MaxVel = \frac{FactorVelocity}{20} = \frac{58593,75}{20} = 2929,69$$

4.7.1.2 Funktionalität**Antrieb ein- bzw. ausschalten**

- Der Eingang *AxisEnable* dient zum Ein- und Ausschalten einer Achse.
- Das Einschalten ist nur möglich, wenn *AxisReady* = TRUE meldet, d.h. die Achse einschaltbereit ist.
- Sobald die Achse eingeschaltet ist, wird dies durch die Statusinformation *AxisEnabled* angezeigt.
- Hat die Achse einen Fehler, wird dies durch die Statusinformation *AxisError* angezeigt. Weitere Informationen liefert *AxisErrorID*.

Antriebsfehler quittieren

- Mit *AxisReset* können Sie Fehler am Antrieb quittieren.
- Anliegende Fehler werden über *DriveError* zurück gemeldet.

Achse stoppen - MC_Stop

- Eine Achse in Bewegung können Sie durch Setzen von *StopExecute* stoppen.
- Solange *StopExecute* gesetzt ist, werden keine weiteren Pulse generiert und alle Kommandos sind gesperrt.

Geschwindigkeitsmodus - MC_MoveVelocity

- Voraussetzung: Der Antrieb ist eingeschaltet und *AxisReady* = TRUE.
- Mit *MvVelocityExecute* können Sie die Achse zum Drehen mit konstanter Geschwindigkeit bringen.
- Die Geschwindigkeit geben Sie über *Velocity* vor.
- Durch Vorgabe von 0 stoppt die Achse ebenso wie mit *StopExecute*.
- Die Drehrichtung bestimmen Sie über das Vorzeichen von *Velocity*.
- Der Wert für *Velocity* darf 0 sein oder $MinUserVelocity \leq Velocity \leq MaxUserVelocity$.

Relative Positionierung - MC_MoveRelative

- Voraussetzung: Der Antrieb ist eingeschaltet und *AxisReady* = TRUE.
- Die relative Positionierung erfolgt über *MvRelativeExecute*.
- Die Distanz können Sie über *PositionDistance* in Anwendereinheiten vorgeben.
- Die Drehrichtung bestimmen Sie über das Vorzeichen von *PositionDistance*.
- Die Geschwindigkeit geben Sie über *Velocity* vor.
- Durch Setzen von *StopExecute* können Sie ein laufendes Kommando stoppen.

Tipp-Betrieb - Jogging

- Voraussetzung: Der Antrieb ist eingeschaltet und *AxisReady* = TRUE.
- Mit einer Flanke 0-1 an *JogPositive* oder *JogNegative* können Sie Ihren Antrieb im Tipp-Betrieb steuern. Hierbei wird ein Jogging Kommando in die entsprechende Drehrichtung ausgeführt.
- Die Geschwindigkeit geben Sie über *Velocity* vor. Das Vorzeichen ist nicht relevant.
- Mit einer Flanke 1-0 an *JogPositive* oder *JogNegative* bzw. durch Setzen von *StopExecute* wird die Achse gestoppt.

5 Einsatz Frequenzumrichter über PWM

5.1 Übersicht

Voraussetzung

- SPEED7 Studio ab V1.7.1
oder
- Siemens SIMATIC Manager ab V 5.5 SP2 & *Simple Motion Control Library*
oder
- Siemens TIA Portal V 14 & *Simple Motion Control Library*
- System MICRO bzw. System SLIO CPU mit PWM-Ausgabe wie z.B. CPU M13-CCF0000 bzw. CPU 013-CCF0R00.
- Frequenzumrichter mit PWM-Eingang z.B. V1000.

Schritte der Projektierung

1. ➔ Parameter am Frequenzumrichter einstellen
 - Die Einstellung der Parameter hat mit dem Softwaretool *Drive Wizard+* zu erfolgen.
2. ➔ Hardwarekonfiguration im VIPA *SPEED7 Studio*, Siemens SIMATIC Manager oder Siemens TIA Portal.
 - Projektierung der CPU.
3. ➔ Programmierung im VIPA *SPEED7 Studio*, Siemens SIMATIC Manager oder Siemens TIA Portal.
 - *VMC_AxisControlV1000PWM*-Baustein zur Konfiguration und zur Kommunikation mit der Achse, welche über PWM angebunden ist.

5.2 Parameter am Frequenzumrichter einstellen



VORSICHT!

Vor der Erstinbetriebnahme müssen Sie Ihren Frequenzumrichter mit dem Softwaretool *Drive Wizard+* an Ihre Applikation anpassen! Näheres hierzu finden Sie im Handbuch zu Ihrem Frequenzumrichter.

Die nachfolgende Tabelle zeigt alle Parameter auf, die nicht den Standardwerten entsprechen. Zur Abstimmung auf die *Simple Motion Control Library* sind diese über *Drive Wizard+* einzustellen. Danach folgt eine Tabelle mit Parameter, welche in Abhängigkeit von der Applikation angepasst werden können.

Nr.	Parameter, die vom Standard abweichen	Einstellung für <i>Simple Motion Control Library</i>
B1-01	Auswahl Frequenzsollwert	■ 4: Pulse train input
B1-02	Auswahl Start-Befehl	■ 1: Control circuit terminal
H1-01	Funktionsauswahl für Digitaleingänge S1	■ 0040: Forward Run Command
H1-02	Funktionsauswahl für Digitaleingänge S2	■ 0041: Reverse Run Command
H2-01	Funktionswahl Klemme MA, MB und MC (Relais)	■ 000E: Fault
H2-02	Funktionswahl für Klemme P1 (Open-Collector-Ausgang)	■ 0006
H6-01	Funktionsauswahl für die Impulsfolgeeingangsklemmen RP	■ 0: Frequency reference

Parameter am Frequenzumrichter einstellen

Nr.	Parameter, die vom Standard abweichen	Einstellung für <i>Simple Motion Control Library</i>
H6-02	Skalierung für Impulsfolgeeingang	■ 20000Hz
H6-03	Verstärkung für Impulsfolgeeingang	■ 100.0%
H6-04	Vorspannung für Impulsfolgeeingang	■ 0.0%
H6-05	Filterzeit für Impulsfolgeeingang	■ 0.10s
H6-06	Auswahl der Überwachung für Impulsfolgeausgangsklemme MP	■ 102: Output frequency
H6-07	Überwachungsskalierung für Impulsfolgeeingang	■ 20000Hz

Nr.	Parameter in Abhängigkeit von der Applikation	Beispiel
C1-01	Hochlaufzeit 1	■ 10.00s
C1-02	Tieflaufzeit 1	■ 10.00s
C1-10	Einstellschritte für Hochlauf-/Tieflaufzeit	■ 0: 0.01- second units
C1-11	Umschaltfrequenz für Hochlauf-/Tieflaufzeit	■ 0.0Hz
O1-02	Auswahl Anwender-Überwachungsparameter nach dem Einschalten	■ 1: Frequency reference
O1-03	Auswahl Anzeige am digitalen Bedienteil	■ 2: min-1 unit



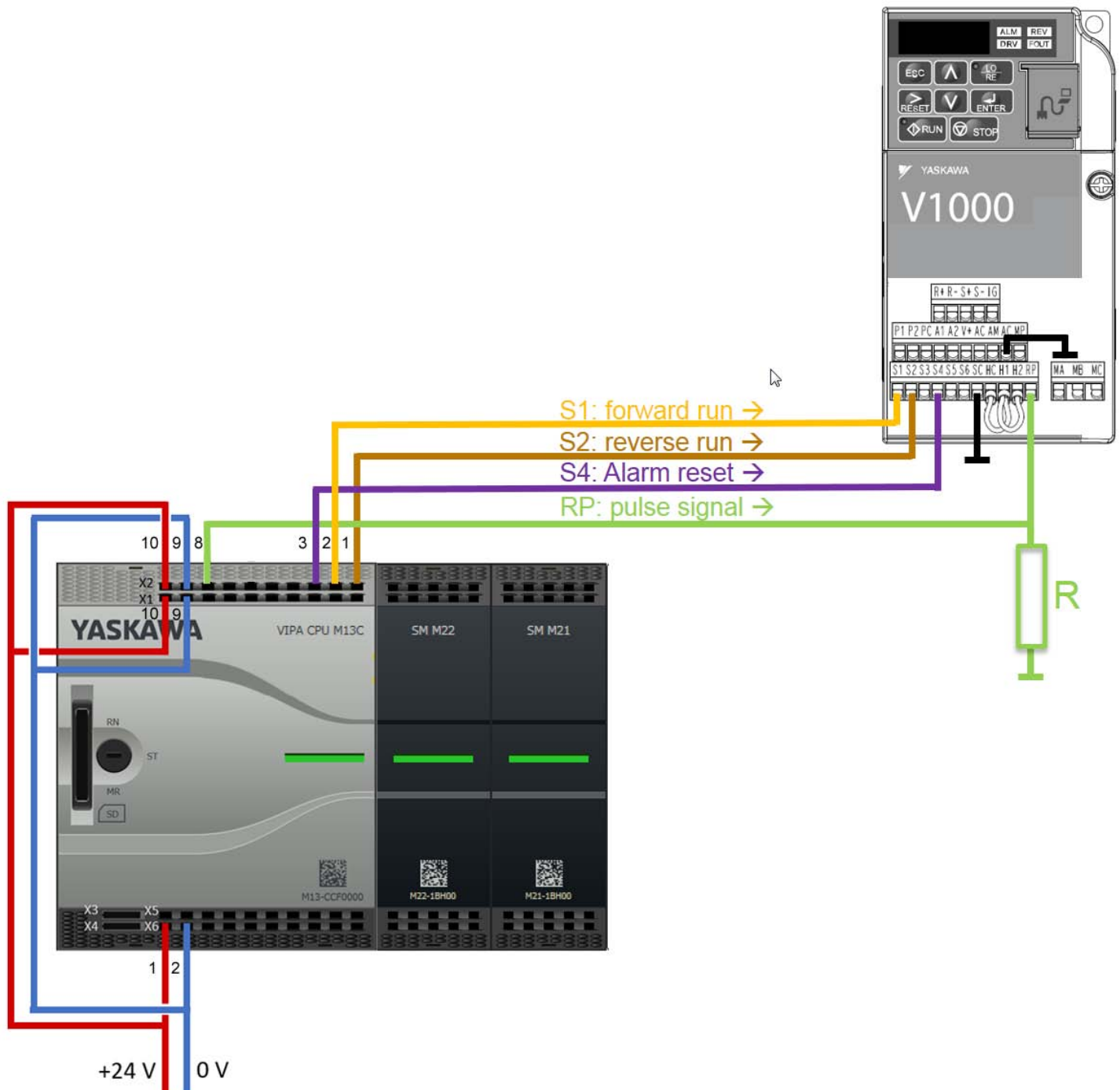
Damit alle Einstellungen übernommen werden, müssen Sie den Frequenzumrichter nach der Parametrierung neu starten!

5.3 Beschaltung

5.3.1 Ansteuerung V1000 Eingänge

Beispielapplikation

Die nachfolgende Abbildung zeigt eine Beispielapplikation zum Anschluss der Eingänge eines V1000 Frequenzumrichters über PWM an eine System MICRO CPU M13C. In diesem Beispiel ist der PWM Kanal 0 (X2 - Pin 8) beschaltet. Zur Beschaltung von Kanal 1 ist X2 - Pin 7 zu verwenden.

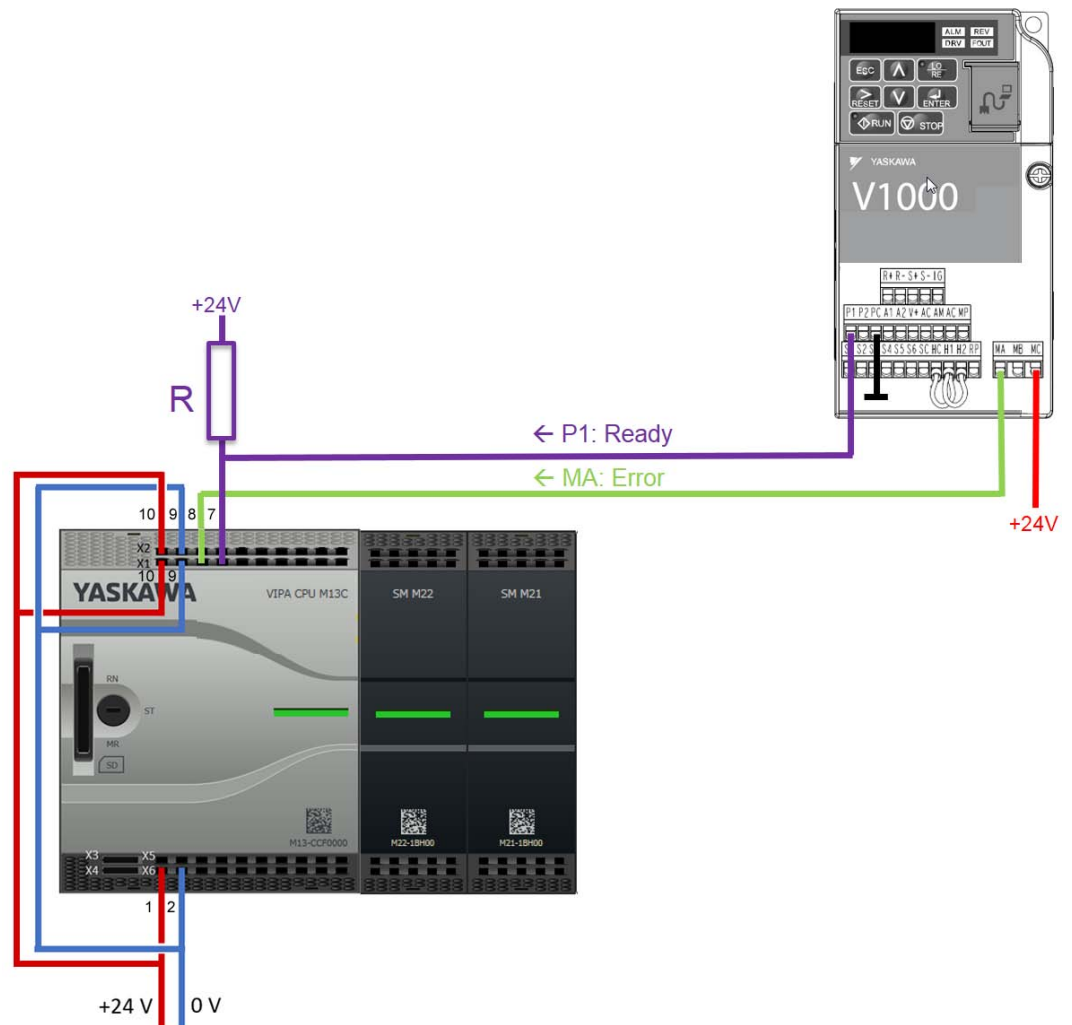


- R Widerstand
 Wert: max. 470Ω
 Verlustleistung: min. 0,6W
 Widerstandsbeispiel: Metallschichtwiderstand 0207
 bedrahtet mit 0,6W Verlustleistung
 Kabellänge max. 20m

5.3.2 Ansteuerung V1000 Ausgänge

Beispielapplikation

Die nachfolgende Abbildung zeigt eine Beispielapplikation zum Anschluss der Ausgänge eines V1000 Frequenzumrichters an eine System MICRO CPU M13C.



R Widerstand
Wert: 4,7kΩ
Verlustleistung: min. 0,25W
Widerstandsbeispiel: Kohleschichtwiderstand 0207 bedrahtet mit 0,25W Verlustleistung

5.4 Einsatz im VIPA SPEED7 Studio

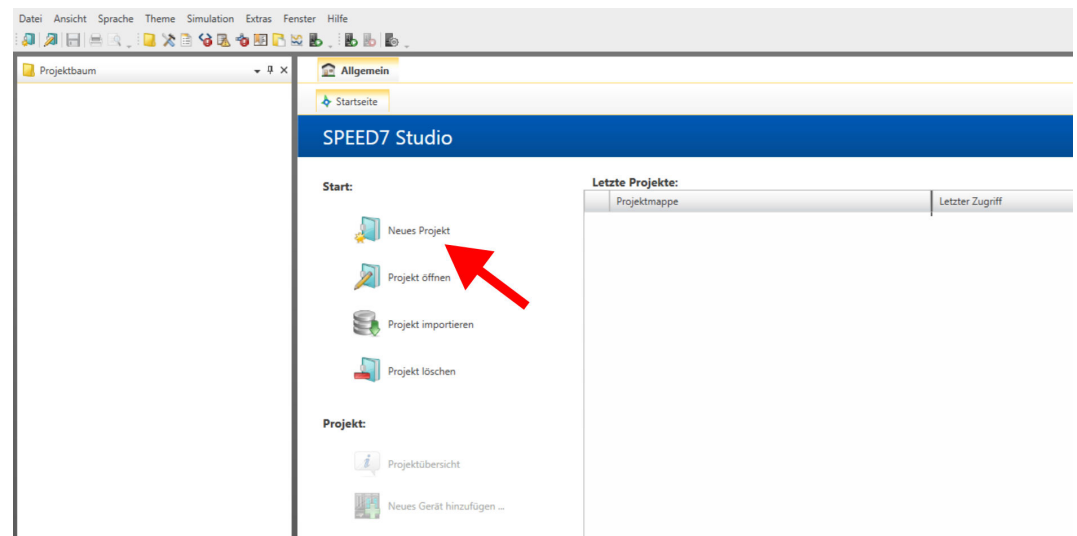
5.4.1 Hardware-Konfiguration

CPU im Projekt anlegen

Bitte verwenden Sie für die Projektierung das *SPEED7 Studio* ab V1.7.1

Sollten Sie einen anderen Kanal als Kanal 0 verwenden, müssen Sie diesen in der Hardware-Konfiguration und in Ihrem Anwenderprogramm entsprechend anpassen.

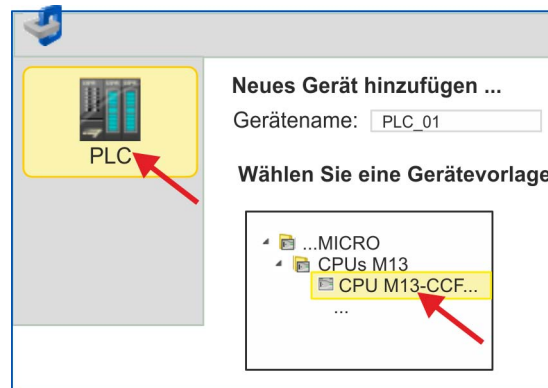
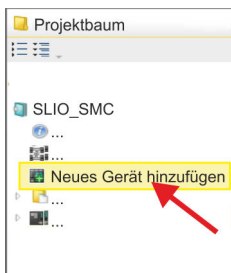
1. Starten Sie das *SPEED7 Studio*.



2. Erstellen sie auf der Startseite mit *"Neues Projekt"* ein neues Projekt und vergeben Sie einen *"Projektnamen"*.

⇒ Ein neues Projekt wird angelegt und in die Sicht *"Geräte und Netze"* gewechselt.

3. Klicken Sie im *Projektbaum* auf *"Neues Gerät hinzufügen ..."*.



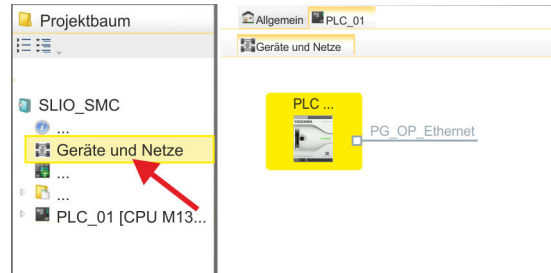
⇒ Es öffnet sich ein Dialog für die Geräteauswahl.

4. Wählen Sie unter den *"Gerätevorlagen"* Ihre CPU mit PWM Funktionalität wie z.B. die System MICRO CPU M13-CCF0000 und klicken Sie auf [OK].

⇒ Die CPU wird in *"Geräte und Netze"* eingefügt und die *"Gerätekonfiguration"* geöffnet.

Ethernet-PG/OP-Kanal parametrieren

1. ➤ Klicken Sie im *Projektbaum* auf "*Geräte und Netze*".
⇒ Sie erhalten eine grafische Objekt-Ansicht Ihrer CPU.



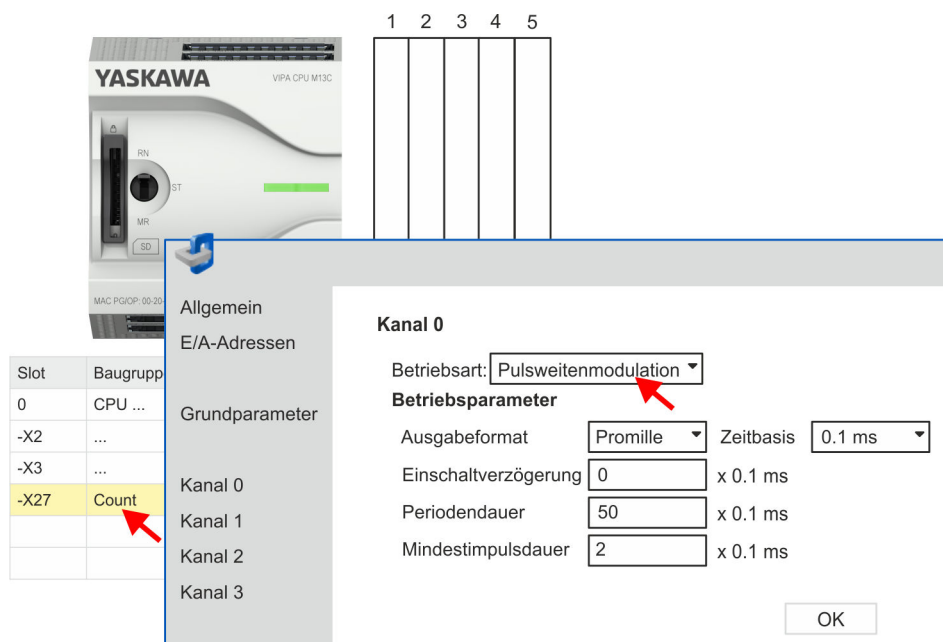
2. ➤ Klicken Sie auf das Netzwerk "*PG_OP_Ethernet*".
3. ➤ Wählen Sie "*Kontextmenü ➔ Eigenschaften der Schnittstelle*".
⇒ Es öffnet sich ein Dialogfenster. Hier können Sie IP-Adressdaten für Ihren Ethernet-PG/OP-Kanal angeben. Gültige IP-Adress-Parameter erhalten Sie von Ihrem Systemadministrator.
4. ➤ Bestätigen Sie Ihre Eingabe mit [OK].
⇒ Die IP-Adressdaten werden in Ihr Projekt übernommen und in "*Geräte und Netze*" unter "*Lokale Baugruppen*" aufgelistet.
Nach der Übertragung Ihres Projekts ist Ihre CPU über die angegebenen IP-Adressdaten via Ethernet-PG/OP-Kanal erreichbar.

E/A-Peripherie auf PWM umschalten

Für die Parametrierung der Ein-/Ausgabeperipherie und der *Technologischen Funktionen* sind die entsprechenden Submodule der CPU zu verwenden. Für die PWM Ausgabe ist das Submodul Count auf "*Pulsweitenmodulation*" umzuschalten.

1. ➤ Klicken Sie im *Projektbaum* auf "*PLC... > Gerätekonfiguration*".
2. ➤ Klicken Sie in der "*Gerätekonfiguration*" auf "*-X27 Count*" und wählen Sie "*Kontextmenü ➔ Eigenschaften der Baugruppe*".
⇒ Es öffnet sich der Eigenschaften-Dialog.
3. ➤ Klicken Sie auf den entsprechenden Kanal wie z.B. "*Kanal 0*" und stellen Sie unter "*Betriebsart*" die Funktion "*Pulsweitenmodulation*" ein.

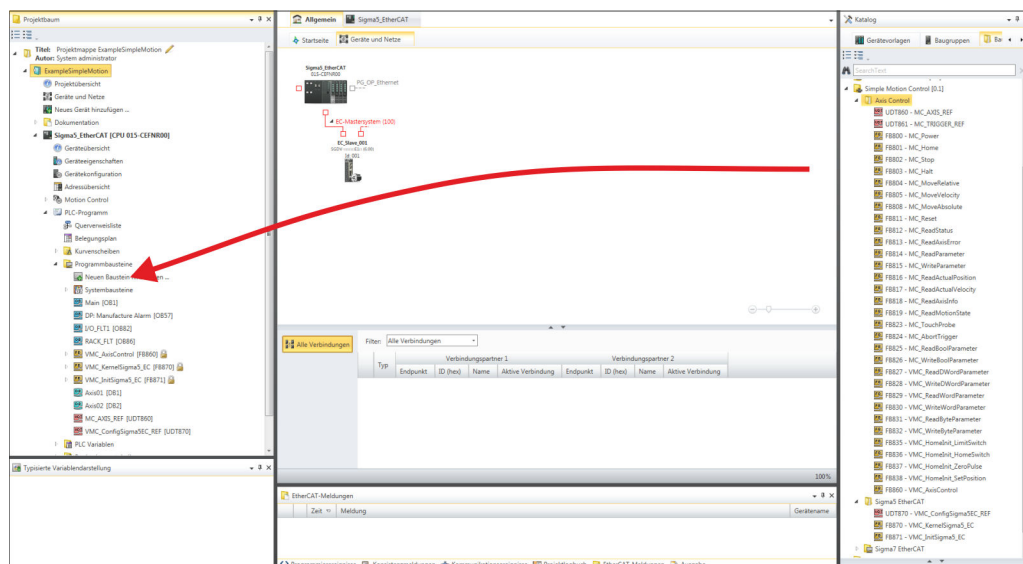
4. Die für PWM erforderlichen Betriebsparameter werden intern auf die entsprechenden Werte angepasst. Lassen Sie alle Werte unverändert.



5. Schließen Sie den Dialog mit [OK].
6. Wählen Sie "Projekt → Alles übersetzen".

5.4.2 Anwender-Programm

Baustein in Projekt kopieren



- Öffnen Sie im "Katalog" unter "Bausteine" "Simple Motion Control" und ziehen Sie per Drag&Drop folgenden Baustein in "Programmbausteine" des Projektbaums:

- V1000 PWM
 - FB885 – VMC_AxisControlV1000PWM ↗ Kapitel 5.7.1 "FB 885 - VMC_AxisControlV1000_PWM - Achskontrolle über PWM" auf Seite 246

OB 1

Konfiguration der Achse

Sollten Sie einen anderen Kanal als Kanal 0 verwenden, müssen Sie diesen in der Hardware-Konfiguration und in Ihrem Anwenderprogramm entsprechend anpassen.

1. Öffnen Sie in "Projektbaum → ...CPU... → PLC-Programm → Programmbausteine" den OB 1 und programmieren Sie den Call FB 885, DB 885.

⇒ Es öffnet sich der Dialog "Instanz-Datenbaustein hinzufügen".

2. Stellen Sie, wenn nicht schon geschehen, die Nummer für den Instanz-Datenbaustein ein und schließen Sie den Dialog mit [OK].

⇒ Der Bausteinaufruf wird angelegt und die Parameter werden aufgelistet.

3. Vergeben Sie für das Beispielprojekt folgende Parameter:

```
CALL FB "VMC_AxisControlV1000PWM" ,
      "VMC_AxisCtrlV1000PWM_885"
      I_ChannelNumberPWM := "Ax1_I_ChannelNumberPWM"
      I_MA_Alarm          := "Ax1_MA_Alarm"
      I_P1_Ready          := "I_P1_Ready"
      MaxVelocityDrive    := 1.000000e+002
      AxisEnable          := "Ax1_AxisEnable"
      AxisReset           := "Ax1_AxisReset"
      StopExecute         := "Ax1_StopExecute"
      MvVelocityExecute   := "Ax1_MvVelExecute"
      JogPositive         := "Ax1_JogPositive"
      JogNegative         := "Ax1_JogNegative"
      Velocity            := "Ax1_Velocity"
      I_S1_ForwardRun     := "Ax1_S1_ForwardRun"
      I_S2_ReverseRun     := "Ax1_S2_ReverseRun"
      I_S4_AlarmReset     := "Ax1_S4_AlarmReset"
      MinUserVelocity     := "Ax1_MinUserVelocity"
      MaxUserVelocity     := "Ax1_MaxUserVelocity"
      AxisReady           := "Ax1_AxisReady"
      AxisEnabled         := "Ax1_AxisEnabled"
      AxisError           := "Ax1_AxisError"
      AxisErrorID         := "Ax1_AxisErrorID"
      DriveError          := "Ax1_DriveError"
      CmdActive           := "Ax1_CmdActive"
      CmdDone             := "Ax1_CmdDone"
      CmdBusy             := "Ax1_CmdBusy"
      CmdAborted          := "Ax1_CmdAborted"
      CmdError            := "Ax1_CmdError"
      CmdErrorID          := "Ax1_CmdErrorID"
```

Die Adressen von *I_P1_Ready* und *I_MA_Alarm* ergeben sich aus den Adressen der Eingänge, welche mit den Digitalausgängen des Frequenzumrichters verbunden sind. Diese können über das Submodul "-X25 DI/DIO" der CPU ermittelt werden.

Die Adressen von *I_S1_ForwardRun*, *I_S2_ReverseRun* und *I_S4_AlarmReset* ergeben sich aus den Adressen der Ausgänge, welche mit den Digitaleingängen des Frequenzumrichters verbunden sind. Diese können über das Submodul "-X25 DI/DIO" der CPU ermittelt werden.

Zeitlicher Ablauf

1. ➤ Wählen Sie *"Projekt ➔ Alles übersetzen"* und übertragen Sie das Projekt in Ihre CPU. Näheres zur Übertragung Ihres Projekt finden Sie in der Onlinehilfe zum *SPEED7 Studio*.
⇒ Sie können jetzt Ihre Applikation in Betrieb nehmen.

**VORSICHT!**

Bitte beachten Sie immer die Sicherheitshinweise zu ihrem Frequenzumrichter, insbesondere bei der Inbetriebnahme!

2. ➤ Bringen Sie Ihre CPU in RUN und schalten Sie Ihren Frequenzumrichter ein.
⇒ Der FB 885 - VMC_AxisControlV1000PWM wird zyklisch abgearbeitet.
3. ➤ Sobald *AxisReady* = TRUE meldet, können Sie mit *AxisEnable* die Achse freigeben.
4. ➤ Sie haben jetzt die Möglichkeit über die entsprechenden Parameter Ihre Achse zu steuern und deren Status abzufragen. ➔ *Kapitel 5.7.1 "FB 885 - VMC_AxisControlV1000_PWM - Achskontrolle über PWM" auf Seite 246*

5.5 Einsatz im Siemens SIMATIC Manager

5.5.1 Voraussetzung

Übersicht

- Bitte verwenden Sie für die Projektierung den Siemens SIMATIC Manager ab V 5.5 SP2.
- Die Projektierung der VIPA CPU mit PWM Funktionalität erfolgt im Siemens SIMATIC Manager in Form eines virtuellen PROFINET IO Devices.
- Das PROFINET IO Device ist mittels GSDML im Hardware-Katalog zu installieren.

VIPA IO Device installieren

Die Installation des PROFINET VIPA IO Device im Hardware-Katalog erfolgt nach folgender Vorgehensweise:

1. ➤ Gehen Sie in den Service-Bereich von www.vipa.com.
2. ➤ Laden Sie aus dem Downloadbereich unter *"Config Dateien ➔ PROFINET"* die Konfigurationsdatei für Ihre CPU.
3. ➤ Extrahieren Sie die Datei in Ihr Arbeitsverzeichnis.
4. ➤ Starten Sie den Hardware-Konfigurator von Siemens.
5. ➤ Schließen Sie alle Projekte.
6. ➤ Gehen Sie auf *"Extras ➔ GSD-Dateien installieren"*.
7. ➤ Navigieren Sie in Ihr Arbeitsverzeichnis und installieren Sie die entsprechende GSDML-Datei.
⇒ Nach der Installation finden Sie das entsprechende PROFINET IO Device unter *"PROFINET IO ➔ Weitere Feldgeräte ➔ I/O ➔ VIPA ..."*.

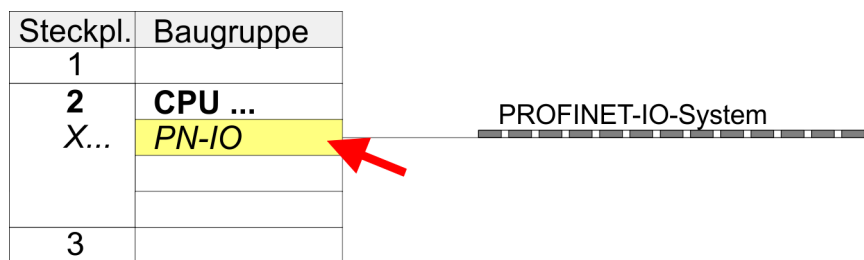
5.5.2 Hardware-Konfiguration

CPU im Projekt anlegen

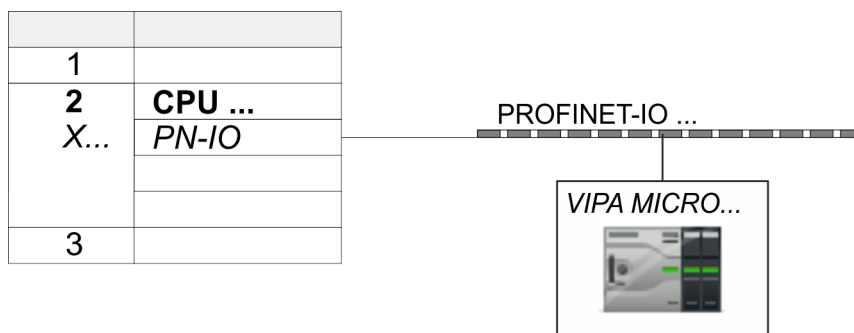
Steckp..	Baugruppe
1	
2	CPU 314C-2PN/DP
X1	MPI/DP
X2	PN-IO
X2...	Port 1
X2...	Port 2
...	...
3	

Um kompatibel mit dem Siemens SIMATIC Manager zu sein, sind folgende Schritte durchzuführen:

1. ➤ Starten Sie den Hardware-Konfigurator von Siemens mit einem neuen Projekt.
2. ➤ Fügen Sie aus dem Hardware-Katalog eine Profilschiene ein.
3. ➤ Platzieren Sie auf "Slot"-Nummer 2 die CPU 314C-2 PN/DP (314-6EH04-0AB0 V3.3).
4. ➤ Klicken Sie auf das Submodul "PN-IO" der CPU.
5. ➤ Wählen Sie "Kontextmenü ➔ PROFINET IO-System einfügen".



6. ➤ Legen Sie mit [Neu] ein neues Subnetz an und vergeben Sie gültige IP-Adress-Daten.
7. ➤ Klicken Sie auf das Submodul "PN-IO" der CPU und öffnen Sie mit "Kontextmenü ➔ Objekteigenschaften" den Eigenschafts-Dialog.
8. ➤ Geben Sie unter "Allgemein" einen "Gerätenamen" an. Der Geräte name muss eindeutig am Ethernet-Subnetz sein.



0	VIPA MICRO ...	M13-CCF0000	
X2	M13-CCF0000		
1			
2			
3			
...			

9. Navigieren Sie im Hardware-Katalog in das Verzeichnis *"PROFINET IO → Weitere Feldgeräte → I/O → VIPA ..."* und binden Sie z.B. für das System MICRO das IO-Device *"M13-CCF0000"* an Ihr PROFINET-System an.
- ⇒ In der *Steckplatzübersicht* des PROFINET-IO-Device *"VIPA MICRO PLC"* ist auf Steckplatz 0 die CPU bereits vorplatziert.

Ethernet-PG/OP-Kanal parametrieren

Steckpl.	Modul
1	
2	CPU ...
X...	PN-IO
3	
4	343-1EX30
5	
...	

- Platzieren Sie für den Ethernet-PG/OP-Kanal auf Steckplatz 4 den Siemens CP 343-1 (SIMATIC 300 \ CP 300 \ Industrial Ethernet \ CP 343-1 \ 6GK7 343-1EX30 0XE0 V3.0).
- Öffnen Sie durch Doppelklick auf den CP 343-1EX30 den Eigenschaften-Dialog und geben Sie für den CP unter *"Eigenschaften"* IP-Adress-Daten an. Gültige IP-Adress-Parameter erhalten Sie von Ihrem Systemadministrator.
- Ordnen Sie den CP einem *"Subnetz"* zu. Ohne Zuordnung werden die IP-Adress-Daten nicht übernommen!

E/A-Peripherie auf PWM umschalten

Für die Parametrierung der Ein-/Ausgabeperipherie und der *Technologischen Funktionen* sind die entsprechenden Submodule der Siemens CPU 314C-2 PN/DP (314-6EH04-0AB0 V3.3) zu verwenden. Für die PWM Ausgabe ist das Submodul Count auf *"Pulsweitenmodulation"* umzuschalten. Sollten Sie einen anderen Kanal als Kanal 0 verwenden, müssen Sie diesen in der Hardware-Konfiguration und in Ihrem Anwenderprogramm entsprechend anpassen.

- Doppelklicken Sie auf das Zähler-Submodul der Siemens CPU 314C-2 PN/DP.
⇒ Sie gelangen in das Dialogfeld *"Eigenschaften"*.
- Wählen Sie z.B. *"Kanal 0"* und stellen Sie unter *"Betriebsart"* die Funktion *"Pulsweitenmodulation"* ein.

3. Lassen Sie alle Werte unverändert.

1	
2	CPU 314C-2 PN/DP
X1	MPI/DP
X2	PN-IO
X2 P1 R	Port 1
X2 P2 R	Port 2
2.5	DI24/DO16
2.6	AI5/AO2
2.7	Zählen
2.8	Positionieren
3	

PROFINET-IO ...

VIPA MICRO...

Eigenschaften - Zählen

Kanal: 0 Betriebsart: Pulsweitenmodulation

4. Schließen Sie den Dialog mit [OK].

5. Wählen Sie "Station → Speichern und übersetzen".

6. Schließen Sie den Hardware-Konfigurator.

5.5.3 Anwender-Programm

Bibliothek einbinden

1. Gehen Sie in den Service-Bereich von www.vipa.com.
2. Laden Sie aus dem Downloadbereich unter "VIPA Lib" die *Simple Motion Control Library*.
3. Öffnen Sie mit "Datei → Dearchivieren" das Dialogfenster zur Auswahl der ZIP-Datei.
4. Wählen Sie die entsprechende ZIP-Datei an und klicken Sie auf [Öffnen].
5. Geben Sie ein Zielverzeichnis an, in dem die Bausteine abzulegen sind und starten Sie den Entpackvorgang mit [OK].

Bausteine in Projekt kopieren

- Öffnen Sie die Bibliothek nach dem Entpackvorgang und ziehen Sie per Drag&Drop folgenden Baustein in "Bausteine" Ihres Projekts:
 - V1000 PWM
 - FB885 – VMC_AxisControlV1000PWM ↗ Kapitel 5.7.1 "FB 885 - VMC_AxisControlV1000_PWM - Achskontrolle über PWM" auf Seite 246

OB 1

Konfiguration der Achse

Sollten Sie einen anderen Kanal als Kanal 0 verwenden, müssen Sie diesen in der Hardware-Konfiguration und in Ihrem Anwenderprogramm entsprechend anpassen.

1. Öffnen Sie im *Projektbaum* innerhalb der CPU unter "PLC-Programm", "Programmbausteine" den OB 1 und programmieren Sie den Call FB 885, DB 885.
 - ⇒ Es öffnet sich der Dialog "Instanz-Datenbaustein hinzufügen".

2. ➤ Stellen Sie, wenn nicht schon geschehen, die Nummer für den Instanz-Datenbaustein ein und schließen Sie den Dialog mit [OK].

⇒ Der Bausteinaufruf wird angelegt und die Parameter werden aufgelistet

3. ➤ Vergeben Sie für das Beispielprojekt folgende Parameter:

```
⇒ CALL FB    "VMC_AxisControlV1000PWM" ,
   "VMC_AxisCtrlV1000PWM_885"
      I_ChannelNumberPWM    := "Ax1_I_ChannelNumberPWM"
      I_MA_Alarm             := "Ax1_MA_Alarm"
      I_P1_Ready             := "I_P1_Ready"
      MaxVelocityDrive       := 1.000000e+002
      AxisEnable              := "Ax1_AxisEnable"
      AxisReset               := "Ax1_AxisReset"
      StopExecute             := "Ax1_StopExecute"
      MvVelocityExecute       := "Ax1_MvVelExecute"
      JogPositive              := "Ax1_JogPositive"
      JogNegative              := "Ax1_JogNegative"
      Velocity                 := "Ax1_Velocity"
      I_S1_ForwardRun         := "Ax1_S1_ForwardRun"
      I_S2_ReverseRun         := "Ax1_S2_ReverseRun"
      I_S4_AlarmReset         := "Ax1_S4_AlarmReset"
      MinUserVelocity         := "Ax1_MinUserVelocity"
      MaxUserVelocity         := "Ax1_MaxUserVelocity"
      AxisReady               := "Ax1_AxisReady"
      AxisEnabled              := "Ax1_AxisEnabled"
      AxisError                := "Ax1_AxisError"
      AxisErrorID              := "Ax1_AxisErrorID"
      DriveError               := "Ax1_DriveError"
      CmdActive                := "Ax1_CmdActive"
      CmdDone                  := "Ax1_CmdDone"
      CmdBusy                  := "Ax1_CmdBusy"
      CmdAborted               := "Ax1_CmdAborted"
      CmdError                 := "Ax1_CmdError"
      CmdErrorID               := "Ax1_CmdErrorID"
```

Die Adressen von *I_P1_Ready* und *I_MA_Alarm* ergeben sich aus den Adressen der Eingänge, welche mit den Digitalausgängen des Frequenzumrichters verbunden sind. Diese können über das Submodul "-X25 DI/DIO" der CPU ermittelt werden.

Die Adressen von *I_S1_ForwardRun*, *I_S2_ReverseRun* und *I_S4_AlarmReset* ergeben sich aus den Adressen der Ausgänge, welche mit den Digitaleingängen des Frequenzumrichters verbunden sind. Diese können über das Submodul "-X25 DI/DIO" der CPU ermittelt werden.

Zeitlicher Ablauf

1. ➤ Wechseln Sie in den Siemens SIMATIC Manager und übertragen Sie Ihr Projekt in die CPU.

⇒ Sie können jetzt Ihre Applikation in Betrieb nehmen.



VORSICHT!

Bitte beachten Sie immer die Sicherheitshinweise zu ihrem Frequenzumrichter, insbesondere bei der Inbetriebnahme!

2. ➤ Bringen Sie Ihre CPU in RUN und schalten Sie Ihren Frequenzumrichter ein.

⇒ Der FB 885 - VMC_AxisControlV1000PWM wird zyklisch abgearbeitet.

3. ➤ Sobald *AxisReady* = TRUE meldet, können Sie mit *AxisEnable* die Achse freigeben.

4. ➔ Sie haben jetzt die Möglichkeit über die entsprechenden Parameter Ihre Achse zu steuern und deren Status abzufragen. ➔ *Kapitel 5.7.1 "FB 885 - VMC_AxisControlV1000_PWM - Achskontrolle über PWM" auf Seite 246*

5.6 Einsatz im Siemens TIA Portal

5.6.1 Voraussetzung

Übersicht

- Bitte verwenden Sie für die Projektierung das Siemens TIA Portal ab V 14.
- Die Projektierung der VIPA CPU mit PWM Funktionalität erfolgt im Siemens TIA Portal in Form eines virtuellen PROFINET IO Devices.
- Das PROFINET IO Device ist mittels GSDML im Hardware-Katalog zu installieren.

VIPA IO Device installieren

Die Installation des PROFINET VIPA IO Device im Hardware-Katalog erfolgt nach folgender Vorgehensweise:

1. ➔ Gehen Sie in den Service-Bereich von www.vipa.com.
2. ➔ Laden Sie aus dem Downloadbereich unter "*Config Dateien ➔ PROFINET*" die entsprechende Datei für Ihr System - hier System MICRO.
3. ➔ Extrahieren Sie die Datei in Ihr Arbeitsverzeichnis.
4. ➔ Starten das Siemens TIA Portal.
5. ➔ Schließen Sie alle Projekte.
6. ➔ Wechseln Sie in die *Projektansicht*.
7. ➔ Gehen Sie auf "*Extras ➔ Gerätebeschreibungsdatei (GSD) installieren*".
8. ➔ Navigieren Sie in Ihr Arbeitsverzeichnis und installieren Sie die entsprechende GSDML-Datei.

⇒ Nach der Installation wird der Hardware-Katalog aktualisiert und das Siemens TIA Portal beendet.

Nach einem Neustart des Siemens TIA Portals finden Sie das entsprechende PROFINET-IO-Device unter *Weitere Feldgeräte > PROFINET > IO > VIPA GmbH > VIPA MICRO PLC*.



Damit die VIPA-Komponenten angezeigt werden können, müssen Sie im Hardware-Katalog bei "Filter" den Haken entfernen.

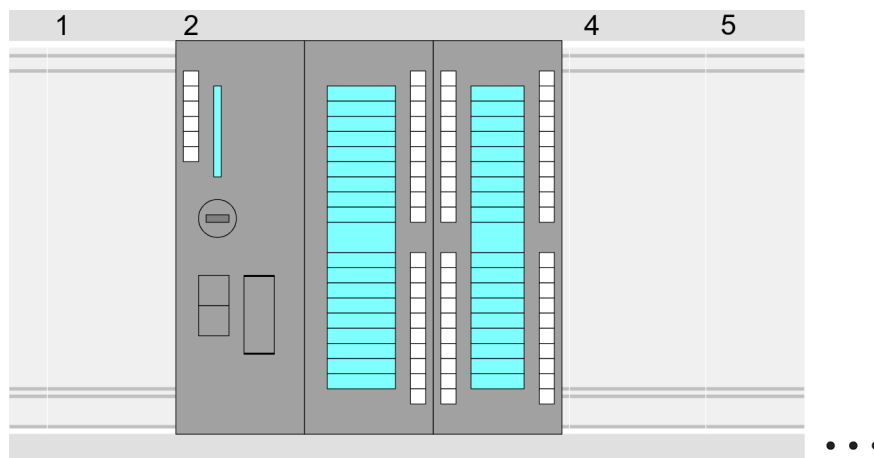
5.6.2 Hardware-Konfiguration

CPU im Projekt anlegen

Um kompatibel mit dem Siemens TIA Portal zu sein, sind folgende Schritte durchzuführen:

1. ➔ Starten Sie das Siemens TIA Portal mit einem neuen Projekt.
2. ➔ Wechseln Sie in die *Projektansicht*.
3. ➔ Klicken Sie in der *Projektnavigation* auf "*Neues Gerät hinzufügen*".

- 4.** Wählen Sie im Eingabedialog folgende CPU aus:
SIMATIC S7-300 > CPU 314C-2 PN/DP (314-6EH04-0AB0 V3.3)
⇒ Die CPU wird mit einer Profilschiene eingefügt.

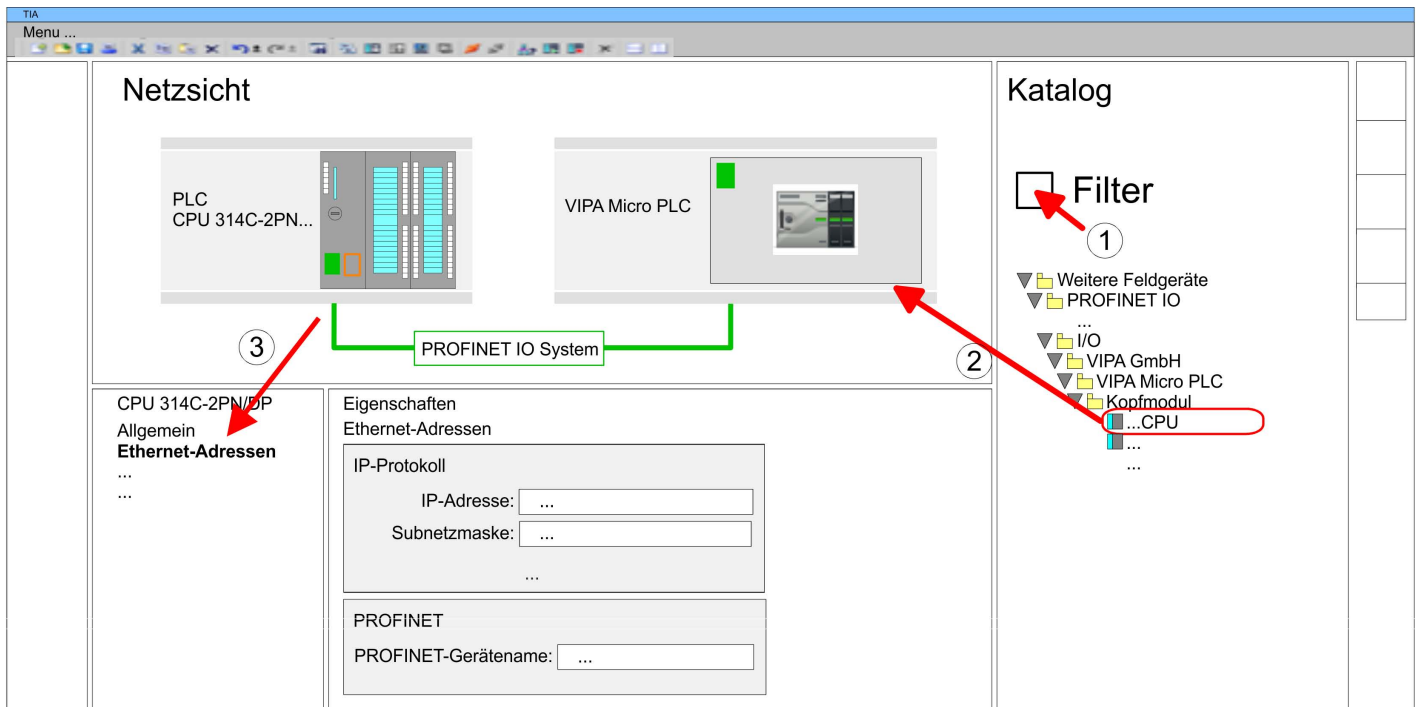


Geräteübersicht:

Baugruppe	...	Steckplatz	...	Typ	...
PLC...		2		CPU 314C-2PN/DP	
MPI-Schnittstelle...		2 X1		MPI/DP-Schnittstelle	
PROFINET-Schnitt...		2 X2		PROFINET-Schnittstelle	
DI24/DO16...		2 5		DI24/DO16	
AI5/AO2...		2 6		AI5/AO2	
Zählen...		2 7		Zählen	
...					

Anbindung CPU als PROFINET-IO-Device

1. ➤ Wechseln Sie im *Projektbereich* in die *"Netzsicht"*.
2. ➤ Nach der Installation der GSDML finden Sie das IO-Device für die CPU im Hardware-Katalog unter *Weitere Feldgeräte > PROFINET > IO > VIPA GmbH > VIPA MICRO PLC*. Binden Sie das Slave-System an die CPU an, indem Sie dies aus dem Hardware-Katalog in die *Netzsicht* ziehen und dieses über PROFINET an die CPU anbinden.
3. ➤ Klicken Sie in der *Netzsicht* auf den PROFINET-Teil der Siemens CPU und geben Sie in *"Eigenschaften"* unter *"Ethernet-Adressen"* im Bereich *"IP-Protokoll"* gültige IP-Adressdaten an.
4. ➤ Geben Sie unter *"PROFINET"* einen *"PROFINET Gerätenamen"* an. Der Geräte-name muss eindeutig am Ethernet-Subnetz sein.

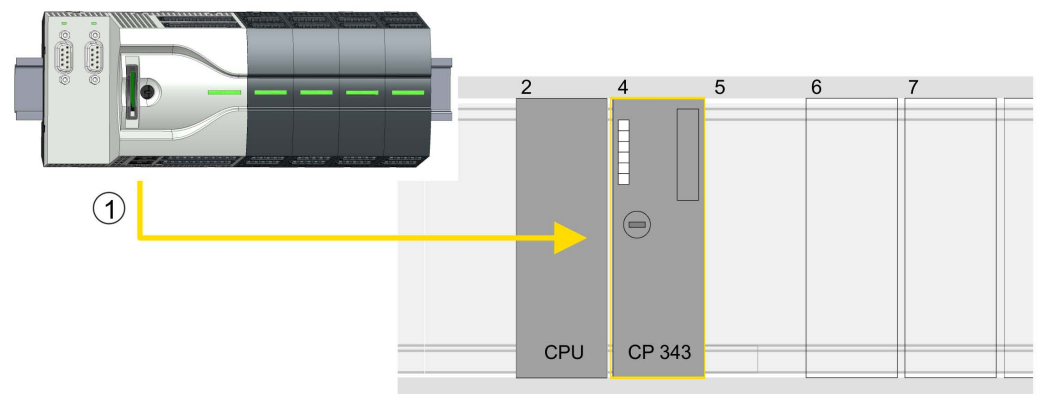


5. ➤ Wählen Sie in der *Netzansicht* das IO-Device "*VIPA MICRO PLC*" an und wechseln Sie in die *Geräteübersicht*.

⇒ In der *Geräteübersicht* des PROFINET-IO-Device "*VIPA MICRO PLC*" ist auf Steckplatz 0 die CPU bereits vorplatziert.

Ethernet-PG/OP-Kanal parametrieren

1. ➤ Platzieren Sie für den Ethernet-PG/OP-Kanal auf Steckplatz 4 den Siemens CP 343-1 (6GK7 343-1EX30 0XE0 V3.0).
2. ➤ Öffnen Sie durch Klick auf den CP 343-1EX30 den "Eigenschaften"-Dialog und geben Sie für den CP in den "Eigenschaften" unter "Ethernet-Adresse" IP-Adress-Daten an. Gültige IP-Adress-Parameter erhalten Sie von Ihrem Systemadministrator.



1 Ethernet-PG/OP-Kanal

Geräteübersicht

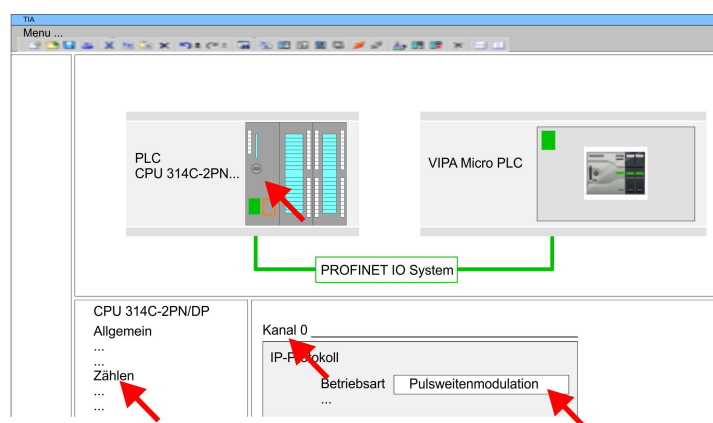
Baugruppe	...	Steckplatz	...	Typ	...
PLC ...		2		CPU 314C-2PN/DP	

MPI/DP-Schnittstelle		2 X1		MPI/DP-Schnittstelle	
PROFINET-Schnittstelle		2 X2		PROFINET-Schnittstelle	
...		
CP 343-1		4		CP 343-1	
...		

E/A-Peripherie auf PWM umschalten

Für die Parametrierung der Ein-/Ausgabeperipherie und der *Technologischen Funktionen* sind die entsprechenden Submodule der Siemens CPU 314C-2 PN/DP (314-6EH04-0AB0 V3.3) zu verwenden. Für die PWM Ausgabe ist das Submodul Count auf "Pulsweitenmodulation" umzuschalten. Sollten Sie einen anderen Kanal als Kanal 0 verwenden, müssen Sie diesen in der Hardware-Konfiguration und in Ihrem Anwenderprogramm entsprechend anpassen.

1. Doppelklicken Sie auf das Zähler-Submodul der Siemens CPU 314C-2 PN/DP.
⇒ Sie gelangen in das Dialogfeld "Eigenschaften".
2. Wählen Sie z.B. "Kanal 0" und stellen Sie unter "Betriebsart" die Funktion "Pulsweitenmodulation" ein.
3. Lassen Sie alle Werte unverändert.



4. Klicken Sie in der Projektnavigation auf Ihre CPU und wählen Sie "Kontextmenü → Übersetzen → Alles".

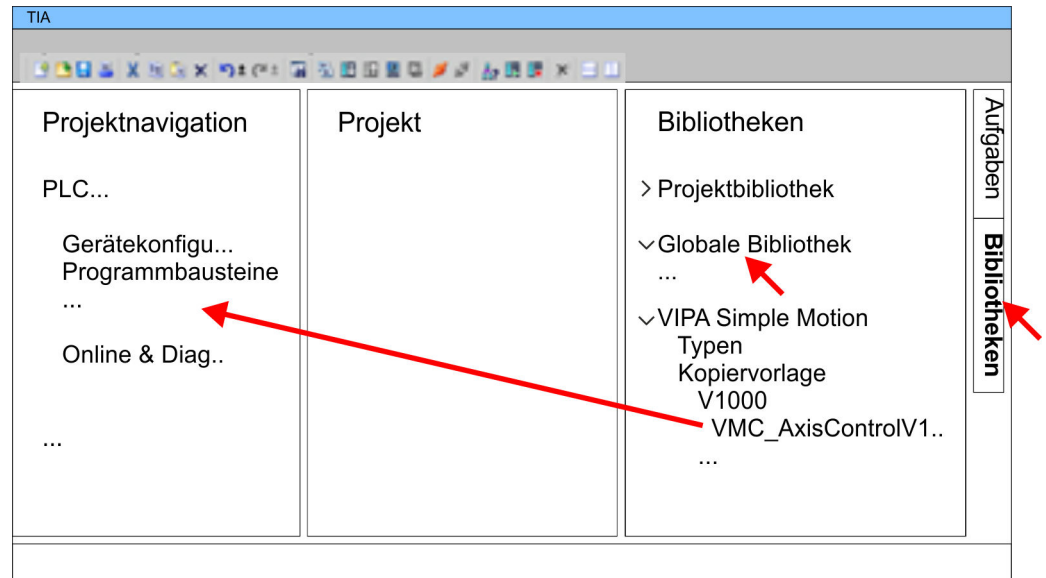
5.6.3 Anwender-Programm

Bibliothek einbinden

1. Gehen Sie in den Service-Bereich von www.vipa.com.
2. Laden Sie aus dem Downloadbereich unter "VIPA Lib" die *Simple Motion Control Library*.
Die Bibliothek liegt für die entsprechende TIA Portal Version als gepackte zip-Datei vor.
3. Starten Sie mit einem Doppelklick auf die Datei ...TIA_Vxx.zip ihr Unzip-Programm, entpacken Sie Dateien und Ordner in ein Arbeitsverzeichnis für das Siemens TIA Portal.
4. Wechseln Sie im Siemens TIA Portal in die *Projektansicht*.
5. Wählen Sie auf der rechten Seite die Task-Card "Bibliotheken".

6. ➤ Klicken Sie auf "Globale Bibliothek".
7. ➤ Klicken Sie innerhalb der "*Globalen Bibliothek*" auf die freie Fläche und wählen Sie "*Kontextmenü* ➔ *Bibliothek dearchivieren*".
8. ➤ Navigieren Sie zu ihrem Arbeitsverzeichnis und laden Sie die Datei ...Simple Motion.zalxx.

Bausteine in Projekt kopieren



- Kopieren Sie folgenden Baustein aus der Bibliothek in das Verzeichnis "Programmbausteine" in der *Projektnavigation* Ihres Projekts.
 - V1000 PWM
 - FB885 – VMC_AxisControlV1000PWM ↗ Kapitel 5.7.1 "FB 885 - VMC_AxisControlV1000_PWM - Achskontrolle über PWM" auf Seite 246

OB 1

Konfiguration der Achse

Sollten Sie einen anderen Kanal als Kanal 0 verwenden, müssen Sie diesen in der Hardware-Konfiguration und in Ihrem Anwenderprogramm entsprechend anpassen.

1. ➤ Öffnen Sie in der *Projektnavigation* innerhalb der CPU unter "*Programmbausteine*" den OB 1 und programmieren Sie den Call FB 885, DB 885
 - ⇒ Es öffnet sich der Dialog "*Instanz-Datenbaustein hinzufügen*".
2. ➤ Stellen Sie, wenn nicht schon geschehen, die Nummer für den Instanz-Datenbaustein ein und schließen Sie den Dialog mit [OK].
 - ⇒ Der Bausteinaufruf wird angelegt und die Parameter werden aufgelistet

3. ➤ Vergeben Sie für das Beispielprojekt folgende Parameter:

```

⇒ CALL FB    "VMC_AxisControlV1000PWM" ,
   "VMC_AxisCtrlV1000PWM_885"
      I_ChannelNumberPWM    := "Ax1_I_ChannelNumberPWM"
      I_MA_Alarm             := "Ax1_MA_Alarm"
      I_P1_Ready             := "I_P1_Ready"
      MaxVelocityDrive       := 1.000000e+002
      AxisEnable             := "Ax1_AxisEnable"
      AxisReset              := "Ax1_AxisReset"
      StopExecute            := "Ax1_StopExecute"
      MvVelocityExecute      := "Ax1_MvVelExecute"
      JogPositive             := "Ax1_JogPositive"
      JogNegative            := "Ax1_JogNegative"
      Velocity               := "Ax1_Velocity"
      I_S1_ForwardRun        := "Ax1_S1_ForwardRun"
      I_S2_ReverseRun        := "Ax1_S2_ReverseRun"
      I_S4_AlarmReset        := "Ax1_S4_AlarmReset"
      MinUserVelocity        := "Ax1_MinUserVelocity"
      MaxUserVelocity        := "Ax1_MaxUserVelocity"
      AxisReady              := "Ax1_AxisReady"
      AxisEnabled            := "Ax1_AxisEnabled"
      AxisError              := "Ax1_AxisError"
      AxisErrorID            := "Ax1_AxisErrorID"
      DriveError             := "Ax1_DriveError"
      CmdActive              := "Ax1_CmdActive"
      CmdDone                := "Ax1_CmdDone"
      CmdBusy                := "Ax1_CmdBusy"
      CmdAborted             := "Ax1_CmdAborted"
      CmdError               := "Ax1_CmdError"
      CmdErrorID             := "Ax1_CmdErrorID"

```

Die Adressen von *I_P1_Ready* und *I_MA_Alarm* ergeben sich aus den Adressen der Eingänge, welche mit den Digitalausgängen des Frequenzumrichters verbunden sind. Diese können über das Submodul "X25 DI/DIO" der CPU ermittelt werden.

Die Adressen von *I_S1_ForwardRun*, *I_S2_ReverseRun* und *I_S4_AlarmReset* ergeben sich aus den Adressen der Ausgänge, welche mit den Digitaleingängen des Frequenzumrichters verbunden sind. Diese können über das Submodul "X25 DI/DIO" der CPU ermittelt werden.

Zeitlicher Ablauf

- 1. ➤** Wählen Sie "Bearbeiten ➔ Übersetzen" und übertragen Sie das Projekt in Ihre CPU. Näheres zur Übertragung Ihres Projekt finden Sie in der Onlinehilfe zum Siemens TIA Portal.

⇒ Sie können jetzt Ihre Applikation in Betrieb nehmen.

**VORSICHT!**

Bitte beachten Sie immer die Sicherheitshinweise zu ihrem Frequenzumrichter, insbesondere bei der Inbetriebnahme!

- 2. ➤** Bringen Sie Ihre CPU in RUN und schalten Sie Ihren Frequenzumrichter ein.

⇒ Der FB 875 - VMC_AxisControl_PT wird zyklisch abgearbeitet
- 3. ➤** Sobald *AxisReady* = TRUE meldet, können Sie mit *AxisEnable* die Achse freigeben.
- 4. ➤** Sie haben jetzt die Möglichkeit über die entsprechenden Parameter Ihre Achse zu steuern und deren Status abzufragen. ➔ *Kapitel 5.7.1 "FB 885 - VMC_AxisControlV1000_PWM - Achskontrolle über PWM" auf Seite 246*

5.7 Antriebsspezifischer Baustein

5.7.1 FB 885 - VMC_AxisControlV1000_PWM - Achskontrolle über PWM

Beschreibung Mit dem FB *VMC_AxisControlV1000_PWM* können Sie einen über PWM angebandenen Frequenzumrichter steuern und dessen Status abrufen.

Parameter

Parameter	Deklaration	Datentyp	Beschreibung
I_Channel- NumberPWM	INPUT	INT	Kanalnummer des PWM-Ausgangs, der für die Ansteuerung des PWM-Eingangs des Frequenzumrichters verwendet wird.
I_MA_Alarm	INPUT	BOOL	<ul style="list-style-type: none"> Digitaleingang zur Anbindung des I_MA_Alarm-Signals (MA) <ul style="list-style-type: none"> TRUE: Der Frequenzumrichter hat einen Fehler erkannt.
I_P1_Ready	INPUT	BOOL	<ul style="list-style-type: none"> Digitaleingang zur Anbindung des I_P1_Ready-Signals <ul style="list-style-type: none"> FALSE: Der Frequenzumrichter ist bereit.
MaxVelocity- Drive	INPUT	REAL	<ul style="list-style-type: none"> Maximale Geschwindigkeit des Frequenzumrichters [Anwender-einheiten]. ↪ <i>Kapitel 5.7.1.1 "Berechnung" auf Seite 248</i>
AxisEnable	INPUT	BOOL	<ul style="list-style-type: none"> Achsenfreigabe <ul style="list-style-type: none"> Dieser Parameter dient zur bausteininternen Freigabe und hat keinen Einfluss auf den Frequenzumrichter. TRUE: Die Achse wird freigegeben. FALSE: Die Achse wird gesperrt.
AxisReset	INPUT	BOOL	<ul style="list-style-type: none"> Reset Achse <ul style="list-style-type: none"> Flanke 0-1: Reset der Achse wird durchgeführt. Der Zustand eines mit <i>AxisReset</i> gestarteten Reset wird nicht an den Ausgängen <i>CmdActive</i>, <i>CmdDone</i>, <i>CmdBusy</i>, <i>CmdAborted</i>, <i>CmdError</i> und <i>CmdErrorID</i> ausgegeben.
StopExecute	INPUT	BOOL	<ul style="list-style-type: none"> Achse stoppen <ul style="list-style-type: none"> Flanke 0-1: Stoppen der Achse wird gestartet. <p>Hinweis: StopExecute = 1: Kein anderes Kommando kann gestartet werden!</p>
MvVelocityExe- cute	INPUT	BOOL	<ul style="list-style-type: none"> Verfahren der Achse starten <ul style="list-style-type: none"> Flanke 0-1: Die Achse wird auf die angegebene Geschwindigkeit beschleunigt/abgebremst.
JogPositive	INPUT	BOOL	<p>Tipp-Betrieb positiv</p> <ul style="list-style-type: none"> Achse mit konstanter Geschwindigkeit in positive Richtung verfahren <ul style="list-style-type: none"> Flanke 0-1: Das Verfahren der Achse mit konstanter Geschwindigkeit wird gestartet. Flanke 1-0: Die Achse wird gestoppt.
JogNegative	INPUT	BOOL	<p>Tipp-Betrieb negativ</p> <ul style="list-style-type: none"> Achse mit konstanter Geschwindigkeit in negative Richtung verfahren <ul style="list-style-type: none"> Flanke 0-1: Das Verfahren der Achse mit konstanter Geschwindigkeit wird gestartet. Flanke 1-0: Die Achse wird gestoppt.

Parameter	Deklaration	Datentyp	Beschreibung
Velocity	INPUT	REAL	Geschwindigkeitsvorgabe (vorzeichenbehafteter Wert) in [Anwendereinheiten/s]. Hinweis: Bei JogPositive, JogNegative wird der absolute Wert der Geschwindigkeit verwendet.
I_S1_ForwardRun	OUTPUT	BOOL	<ul style="list-style-type: none"> ■ Digitalausgang zur Ansteuerung des Frequenzumrichter-Signals S1 <ul style="list-style-type: none"> – TRUE: Schaltet Frequenzumrichter in positive Richtung frei.
I_S2_ReverseRun	OUTPUT	BOOL	<ul style="list-style-type: none"> ■ Digitalausgang zur Ansteuerung des Frequenzumrichter-Signals S2 <ul style="list-style-type: none"> – TRUE: Schaltet Frequenzumrichter in negative Richtung frei.
I_S4_Alarm-Reset	OUTPUT	BOOL	<ul style="list-style-type: none"> ■ Digitalausgang zur Ansteuerung des Frequenzumrichter-Signals S4 <ul style="list-style-type: none"> – TRUE: Alarme werden im Frequenzumrichter zurückgesetzt. – FALSE: Alarme im Frequenzumrichter bleiben bestehen.
MinUserVelocity	OUTPUT	REAL	Minimale Geschwindigkeit (Periodendauer = 65535µs = maximale Periodendauer des PWM-Ausgangs) des Frequenzumrichters [Anwendereinheiten].
MaxUserVelocity	OUTPUT	REAL	Maximale Geschwindigkeit bei einer maximalen Frequenz von 20kHz des Frequenzumrichters [Anwendereinheiten].
AxisReady	OUTPUT	BOOL	<ul style="list-style-type: none"> ■ AxisReady <ul style="list-style-type: none"> – TRUE: Die Achse ist einschaltbereit. – FALSE: Die Achse ist nicht einschaltbereit. → Prüfe und behebe <i>AxisError</i> (siehe <i>AxisErrorID</i>). → Prüfe und behebe <i>DriveError</i>.
AxisEnabled	OUTPUT	BOOL	<ul style="list-style-type: none"> ■ Status Achse <ul style="list-style-type: none"> – TRUE: Achse ist eingeschaltet und nimmt Bewegungsaufträge an. – FALSE: Achse ist nicht eingeschaltet und nimmt keine Bewegungsaufträge an.
AxisError	OUTPUT	BOOL	<ul style="list-style-type: none"> ■ Fehler an Achse <ul style="list-style-type: none"> – TRUE: Ein Fehler ist aufgetreten. Zusätzliche Fehlerinformationen können dem Parameter <i>AxisErrorID</i> entnommen werden. → Die Achse wird gesperrt (<i>S_On</i> = FALSE und <i>AxisEnabled</i> = FALSE) Kommando wird nicht ausgeführt.
AxisErrorID	OUTPUT	WORD	Zusätzliche Fehlerinformationen 🔗 Kapitel 9 "ErrorID - Zusätzliche Fehlerinformationen" auf Seite 340
DriveError	OUTPUT	BOOL	<ul style="list-style-type: none"> ■ Fehler direkt am Frequenzumrichter <ul style="list-style-type: none"> – TRUE: Ein Fehler ist aufgetreten. → Die Achse wird gesperrt.

Parameter	Deklaration	Datentyp	Beschreibung
CmdActive	OUTPUT	BYTE	<ul style="list-style-type: none"> ■ Kommando <ul style="list-style-type: none"> – 0: kein Cmd aktiv – 1: STOP – 2: MvVelocity – 4: JogPos – 5: JogNeg
CmdDone	OUTPUT	BOOL	<ul style="list-style-type: none"> ■ Status Done <ul style="list-style-type: none"> – TRUE: Auftrag wurde ohne Fehler beendet.
CmdBusy	OUTPUT	BOOL	<ul style="list-style-type: none"> ■ Status Busy <ul style="list-style-type: none"> – TRUE: Auftrag ist in Bearbeitung.
CmdAborted	OUTPUT	BOOL	<ul style="list-style-type: none"> ■ Status Aborted <ul style="list-style-type: none"> – TRUE: Der Auftrag wurde während der Bearbeitung von einem anderen Auftrag abgebrochen. <p>Hinweis: <i>CmdAborted</i> wird beim Start eines Cmd zurückgesetzt</p>
CmdError	OUTPUT	BOOL	<ul style="list-style-type: none"> ■ Status Error <ul style="list-style-type: none"> – TRUE: Ein Fehler ist aufgetreten. Die Achse wird gesperrt <p>Zusätzliche Fehlerinformationen können dem Parameter <i>CmdErrorID</i> entnommen werden.</p>
CmdErrorID	OUTPUT	WORD	<p>Zusätzliche Fehlerinformationen</p> <p>🔗 Kapitel 9 "ErrorID - Zusätzliche Fehlerinformationen" auf Seite 340</p>

5.7.1.1 Berechnung

MaxVelocityDrive

$$n = 2 \cdot 60 \cdot \frac{f_{\max, \text{out}}}{\text{poles}} \frac{1}{\text{min}}$$

Dieser Wert dient zur Normierung des Eingangswerts *Velocity*.

$f_{\max, \text{out}}$ - Maximale Frequenz (Parameter E1-04)

poles - Anzahl der Motor-Pole (Parameter E5-04)

n - Maximale Geschwindigkeit des Frequenzumrichters [Anwendereinheiten] wie z.B 100.0 % oder 3000.0 Umdrehungen/min.

5.7.1.2 Funktionalität

Achse ein- bzw. ausschalten

- Der Eingang *AxisEnable* dient zum Ein- und Ausschalten einer Achse.
- Das Einschalten ist nur möglich, wenn *AxisReady* = TRUE meldet, d.h. die Achse einschaltbereit ist.
- Sobald die Achse eingeschaltet ist, wird dies durch die Statusinformation *AxisEnabled* angezeigt.
- Hat die Achse einen Fehler, wird dies durch die Statusinformation *AxisError* angezeigt. Weitere Informationen liefert *AxisErrorID*.

Achsfehler quittieren

- Mit *AxisReset* können Sie Achsfehler quittieren.
- Anliegende Fehler werden über *DriveError* zurück gemeldet.

Achse stoppen

- Eine Achse in Bewegung können Sie durch Setzen von *StopExecute* stoppen.
- Solange *StopExecute* gesetzt ist, werden keine weiteren Pulse generiert und alle Kommandos sind gesperrt.

Geschwindigkeitsmodus

- Voraussetzung: Die Achse ist eingeschaltet und *AxisReady* = TRUE.
- Mit *MvVelocityExecute* können Sie die Achse zum Drehen mit konstanter Geschwindigkeit bringen.
- Die Geschwindigkeit geben Sie über *Velocity* vor.
- Durch Vorgabe von 0 stoppt die Achse ebenso wie mit *StopExecute*.
- Die Drehrichtung bestimmen Sie über das Vorzeichen von *Velocity*.
- Der Wert für *Velocity* darf 0 sein oder $MinUserVelocity \leq Velocity \leq MaxUserVelocity$.

Tipp-Betrieb - Jogging

- Voraussetzung: Die Achse ist eingeschaltet und *AxisReady* = TRUE.
- Mit einer Flanke 0-1 an *JogPositive* oder *JogNegative* können Sie Ihre Achse im Tipp-Betrieb steuern. Hierbei wird ein Jogging Kommando in die entsprechende Drehrichtung ausgeführt.
- Die Geschwindigkeit geben Sie über *Velocity* vor. Das Vorzeichen ist nicht relevant.
- Mit einer Flanke 1-0 an *JogPositive* oder *JogNegative* bzw. durch Setzen von *StopExecute* wird die Achse gestoppt.

6 Einsatz Frequenzumrichter über Modbus RTU

6.1 Übersicht

Voraussetzung

- SPEED7 Studio ab V1.7.1
oder
- Siemens SIMATIC Manager ab V 5.5 SP2 & *Simple Motion Control Library*
oder
- Siemens TIA Portal V 14 & *Simple Motion Control Library*
- System MICRO bzw. System SLIO CPU mit serieller Schnittstelle wie z.B. CPU M13-CCF0000 bzw. CPU 013-CCF0R00.
- V1000-Frequenzumrichter mit serieller Schnittstelle und zugehörigem Motor

Schritte der Projektierung

1. ➔ Parameter am Frequenzumrichter einstellen
 - Die Einstellung der Parameter hat mit dem Softwaretool *Drive Wizard+* zu erfolgen.
2. ➔ Hardwarekonfiguration im VIPA *SPEED7 Studio*, Siemens SIMATIC Manager oder Siemens TIA Portal.
 - Projektierung der CPU.
3. ➔ Programmierung im VIPA *SPEED7 Studio*, Siemens SIMATIC Manager oder Siemens TIA Portal.
 - Baustein zur seriellen Kommunikation beschalten.
 - Baustein für jeden Modbus-Slave beschalten.
 - Baustein für die Anzahl der Modbus-Slaves beschalten.
 - Baustein für die Kommunikationsdaten aller Modbus-Slaves beschalten.
 - Baustein für den Kommunikations-Manager beschalten.
 - Baustein zur Initialisierung des Frequenzumrichters beschalten.
 - Bausteine für die Bewegungsabläufe beschalten.

6.2 Parameter am Frequenzumrichter einstellen



VORSICHT!

Vor der Erstinbetriebnahme müssen Sie Ihren Frequenzumrichter mit dem Softwaretool *Drive Wizard+* an Ihre Applikation anpassen! Näheres hierzu finden Sie im Handbuch zu Ihrem Frequenzumrichter.

Die nachfolgende Tabelle zeigt alle Parameter auf, die nicht den Standardwerten entsprechen. Zur Abstimmung auf die *Simple Motion Control Library* sind diese über *Drive Wizard+* einzustellen:

Nr.	Bezeichnung	Wertebereich	Einstellung für <i>Simple Motion Control Library</i>
H5-01	Slave-Adresse Frequenzumrichter	00h ... 20h	Standardmäßig ist die Slave-Adresse auf 1Fh eingestellt. Bitte beachten Sie, dass Adressen im Netzwerk nicht mehrfach vergeben sein dürfen!
H5-02	Kommunikationsgeschwindigkeit MEMOBUS/Modbus	0, 1, 2, ..., 8	■ 3: 9600Bit/s

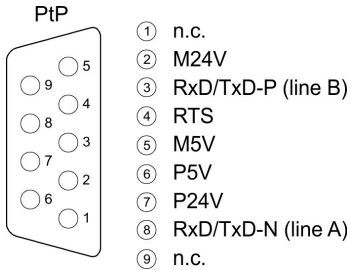
Nr.	Bezeichnung	Wertebereich	Einstellung für <i>Simple Motion Control Library</i>
H5-03	Übertragungsparität MEMOBUS/Modbus	0, 1, 2	■ 0: keine Parität
H5-04	Stoppmethode nach Kommunikations-Fehler (CE-Fehler)	0, 1, 2, 3	■ 3: Betrieb wird fortgesetzt mit Alarm
H5-05	Erkennung Kommunikations-Fehler (CE-Fehler)	0, 1	■ 1: Aktiviert - Bei einem Abbruch der Verbindung für länger als 2s (einstellbar über H2-09) wird ein CE-Fehler ausgelöst.
H5-06	Wartezeit zwischen Empfang und Senden von Daten des Frequenzumrichters	5 ... 65ms	■ 5ms
H5-07	Request to send (RTS)-Steuerung	0, 1	■ 1: Aktiviert - RTS wird nur beim Senden aktiviert (bei RS485 oder RS422 und <i>multi-drop</i>)
H5-09	Zeit nach der ein Kommunikations-Fehler (CE-Fehler) erkannt wird.	0,0 ... 10,0s	■ 2s
H5-10	Schrittgröße (Auflösung) für das MEMOBUS/Modbus-Register 0025h	0, 1	Standardmäßig ist die Auflösung auf 0,1V Inkremente (0) eingestellt. ■ 0: 0,1V Inkremente ■ 1: 1V Inkremente
H5-11	ENTER-Funktion für Verbindungen	0, 1	■ 1: Enter-Befehl nicht erforderlich
H5-12	Auswahl Startbefehlsmethode	0, 1	■ 1: Run/Stop
B1-01	Eingangsquelle Frequenzsollwert 1	0, 1, 2, 3, 4	■ 2: MEMOBUS/Modbus-Kommunikation
B1-02	Eingangsquelle Startbefehl 1	0, 1, 2, 3	■ 2: MEMOBUS/Modbus-Kommunikation
B1-15	Eingangsquelle Frequenzsollwert 2	0, 1, 2, 3, 4	■ 2: MEMOBUS/Modbus-Kommunikation
B1-16	Eingangsquelle Startbefehl 2	0, 1, 2, 3	■ 2: MEMOBUS/Modbus-Kommunikation



Damit alle Einstellungen übernommen werden, müssen Sie den Frequenzumrichter nach der Parametrierung neu starten!

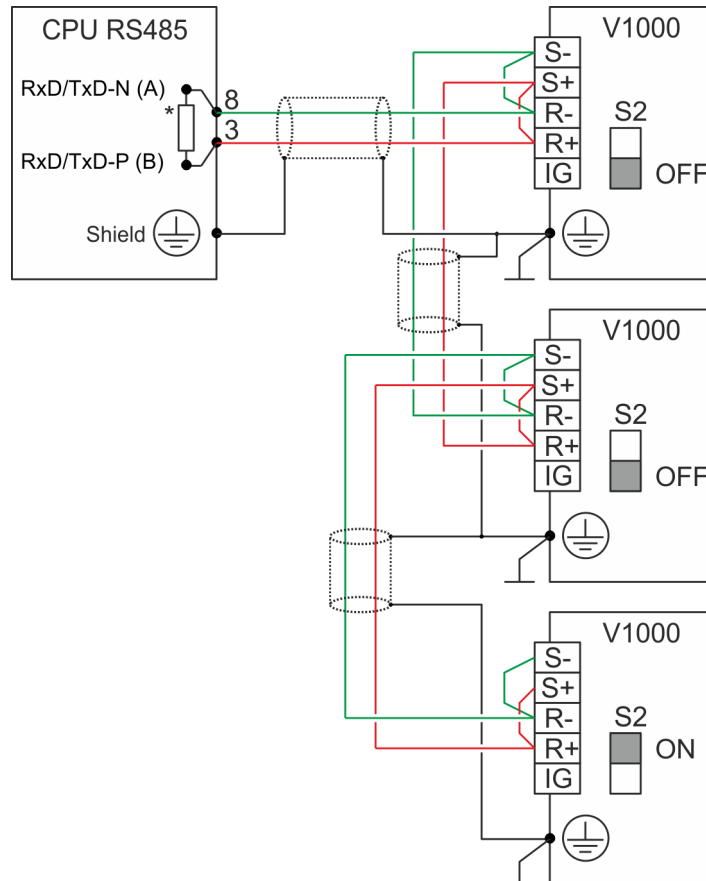
6.3 Beschaltung

RS485-Verkabelung



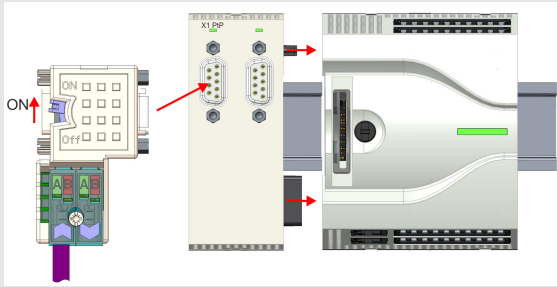
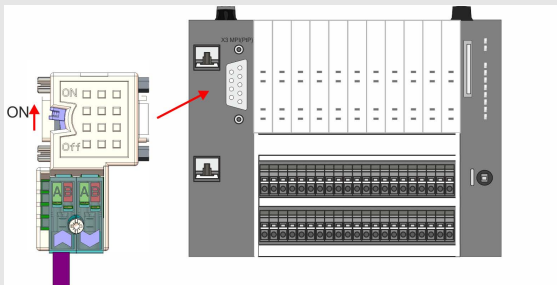
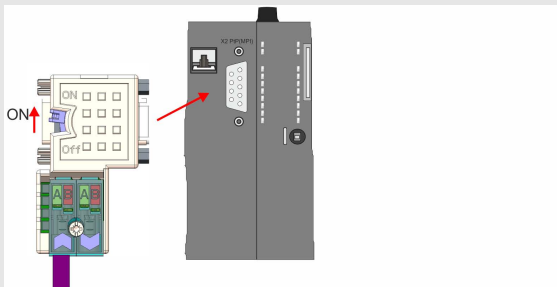
Die nachfolgende Abbildung zeigt den Anschluss von V1000-Frequenzumrichter über RS485. Hierbei werden die einzelnen Frequenzumrichter über PROFIBUS-Kabel verbunden und über einen PROFIBUS-Stecker an die PtP-Schnittstelle (Point-to-Point) der CPU angebunden.

- Bei allen angebundenen Frequenzumrichter ist der Parameter H5-07 auf 1 zu setzen.
- Die serielle Leitung ist an Ihrem Ende mit einem Abschlusswiderstand abzuschließen. Zur Aktivierung müssen Sie hierzu am entsprechenden Frequenzumrichter den Schalter S2 auf "ON" stellen.



- *) Verwenden Sie für einen störungsfreien Datenverkehr einen Abschlusswiderstand von ca. 120Ω an der CPU, wie z.B. beim PROFIBUS-Stecker von VIPA.
- Verbinden Sie niemals Kabelschirm und M5V (Pin 5) miteinander, da aufgrund von Ausgleichsströmen die Schnittstellen zerstört werden könnten!

Anschluss der CPU

CPU	Anschluss	Kommentar
MICRO CPU M13C		<ul style="list-style-type: none"> Für die PtP-Kommunikation ist der Einsatz des optional erhältlichen Erweiterungsmoduls EM M09 erforderlich. Das Erweiterungsmodul stellt die Schnittstelle X1: PtP (RS422/485) mit fixer Pinbelegung zur Verfügung. Verwenden Sie zur Anbindung an die CPU einen PROFIBUS-Stecker von VIPA. Aktivieren Sie am PROFIBUS-Stecker den Abschlusswiderstand. Nach Einschalten der Spannungsversorgung und kurzer Anlaufzeit ist die CPU bereit für die PtP-Kommunikation.
System SLIO CPU 013C		<ul style="list-style-type: none"> Die CPU besitzt die Schnittstelle X3 MPI(PtP) mit fixer Pinbelegung. Verwenden Sie zur Anbindung an die CPU einen PROFIBUS-Stecker von VIPA. Aktivieren Sie am PROFIBUS-Stecker den Abschlusswiderstand. Nach Einschalten der Spannungsversorgung und kurzer Anlaufzeit bzw. nach dem Urlöschen hat die Schnittstelle MPI-Funktionalität. Über die Hardware-Konfiguration können Sie die PtP-Funktionalität aktivieren. <p>↪ Kapitel 6.4 "Einsatz im VIPA SPEED7 Studio" auf Seite 254</p> <p>↪ Kapitel 6.5 "Einsatz im Siemens SIMATIC Manager" auf Seite 269</p> <p>↪ Kapitel 6.6 "Einsatz im Siemens TIA Portal" auf Seite 284</p>
System SLIO CPU 014 ... 017		<ul style="list-style-type: none"> Die CPU besitzt die Schnittstelle X2 PtP(MPI) die standardmäßig auf PtP-Kommunikation (point to point) eingestellt ist. Verwenden Sie zur Anbindung an die CPU einen PROFIBUS-Stecker von VIPA. Aktivieren Sie am PROFIBUS-Stecker den Abschlusswiderstand. Nach Einschalten der Spannungsversorgung und kurzer Anlaufzeit ist die CPU bereit für die PtP-Kommunikation.

6.4 Einsatz im VIPA SPEED7 Studio

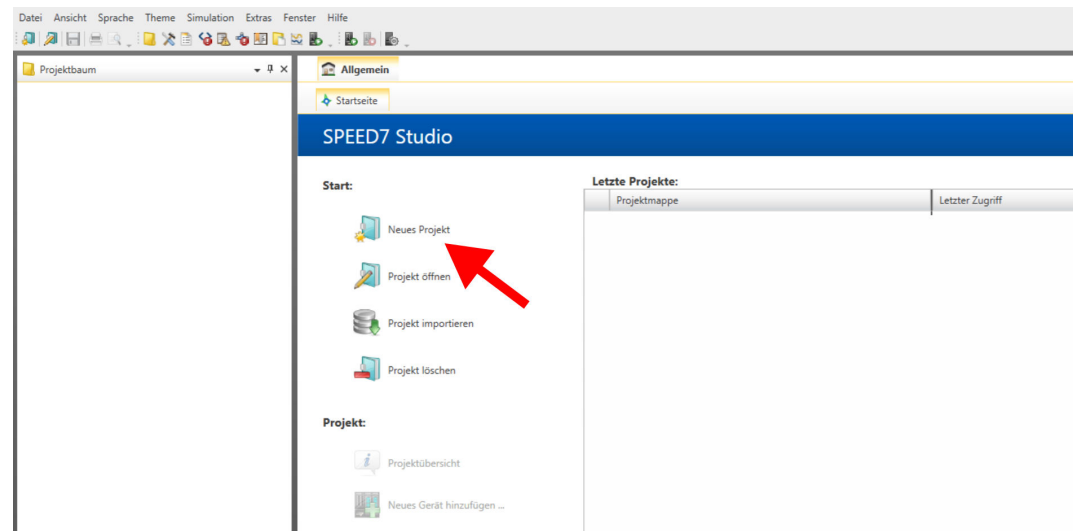
6.4.1 Hardware-Konfiguration

6.4.1.1 Hardware-Konfiguration System MICRO

CPU im Projekt anlegen

Bitte verwenden Sie für die Projektierung das *SPEED7 Studio* ab V1.7.1

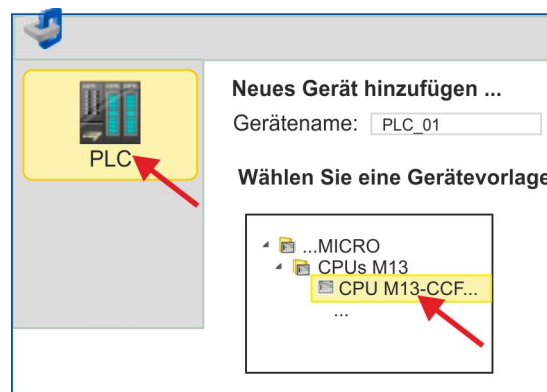
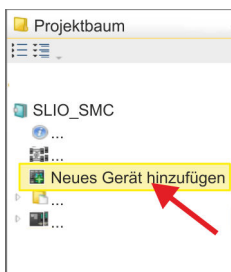
1. Starten Sie das *SPEED7 Studio*.



2. Erstellen sie auf der Startseite mit *"Neues Projekt"* ein neues Projekt und vergeben Sie einen *"Projektnamen"*.

⇒ Ein neues Projekt wird angelegt und in die Sicht *"Geräte und Netze"* gewechselt.

3. Klicken Sie im *Projektbaum* auf *"Neues Gerät hinzufügen ..."*.



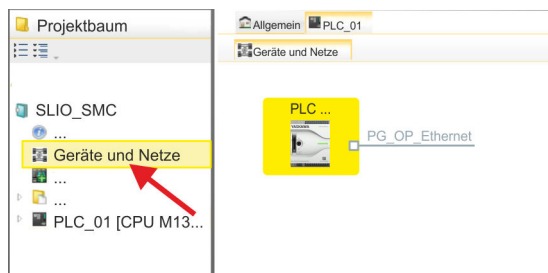
⇒ Es öffnet sich ein Dialog für die Geräteauswahl.

4. Wählen Sie unter den *"Gerätevorlagen"* die System MICRO CPU M13-CCF0000 und klicken Sie auf [OK].

⇒ Die CPU wird in *"Geräte und Netze"* eingefügt und die *"Gerätekonfiguration"* geöffnet.

Ethernet-PG/OP-Kanal parametrieren

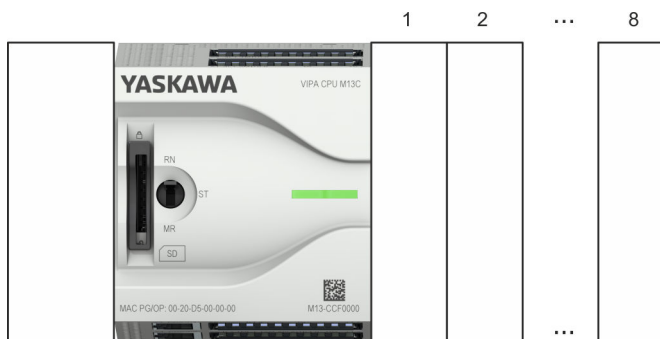
1. ➤ Klicken Sie im *Projektbaum* auf "*Geräte und Netze*".
⇒ Sie erhalten eine grafische Objekt-Ansicht Ihrer CPU.



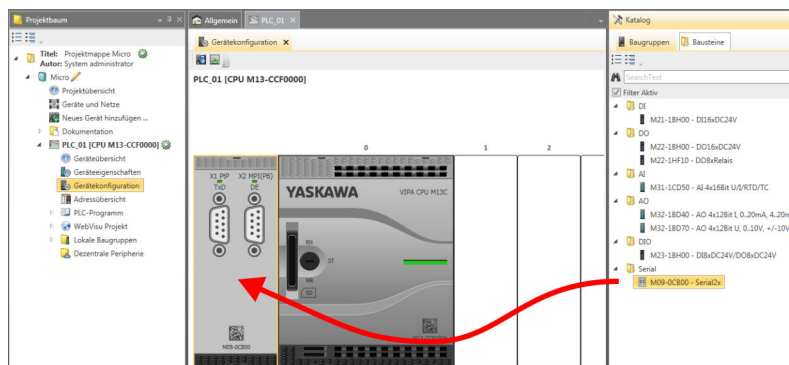
2. ➤ Klicken Sie auf das Netzwerk "*PG_OP_Ethernet*".
3. ➤ Wählen Sie "*Kontextmenü* ➔ *Eigenschaften der Schnittstelle*".
⇒ Es öffnet sich ein Dialogfenster. Hier können Sie IP-Adressdaten für Ihren Ethernet-PG/OP-Kanal angeben. Gültige IP-Adress-Parameter erhalten Sie von Ihrem Systemadministrator.
4. ➤ Bestätigen Sie Ihre Eingabe mit [OK].
⇒ Die IP-Adressdaten werden in Ihr Projekt übernommen und in "*Geräte und Netze*" unter "*Lokale Baugruppen*" aufgelistet.
Nach der Übertragung Ihres Projekts ist Ihre CPU über die angegebenen IP-Adressdaten via Ethernet-PG/OP-Kanal erreichbar.

PtP-Funktionalität aktivieren

1. ➤ Klicken Sie im *Projektbaum* auf "*PLC..CPU M13....* ➔ *Gerätekonfiguration*".
⇒ Die "*Gerätekonfiguration*" wird geöffnet.



2. ➤ Öffnen Sie im "*Katalog*" unter "*Baugruppen*" die Sammlung "*Seriell*" und ziehen Sie per Drag&Drop das serielle Modul "*M09-0CB00 - Serial2x*" auf den linken Steckplatz der CPU. Per Default ist die Schnittstelle X1 auf PtP-Funktionalität eingestellt.

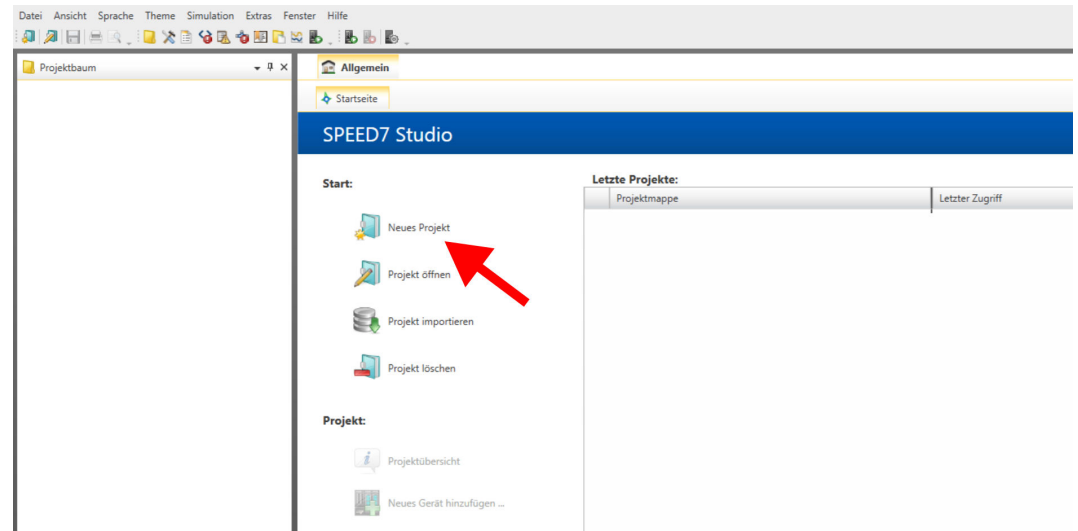


6.4.1.2 Hardware-Konfiguration System SLIO CPU 013C

CPU im Projekt anlegen

Bitte verwenden Sie für die Projektierung das *SPEED7 Studio* ab V1.7.1

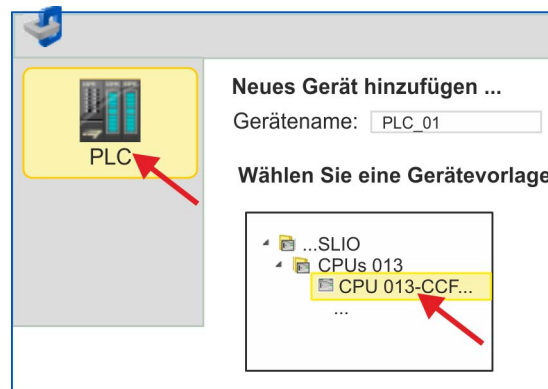
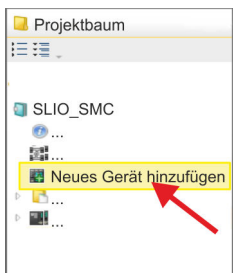
1. Starten Sie das *SPEED7 Studio*.



2. Erstellen sie auf der Startseite mit "*Neues Projekt*" ein neues Projekt und vergeben Sie einen "*Projektnamen*".

⇒ Ein neues Projekt wird angelegt und in die Sicht "*Geräte und Netze*" gewechselt.

3. Klicken Sie im *Projektbaum* auf "*Neues Gerät hinzufügen ...*".



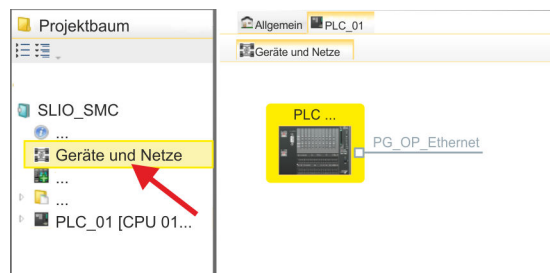
⇒ Es öffnet sich ein Dialog für die Geräteauswahl.

4. Wählen Sie unter den "*Gerätevorlagen*" die System SLIO CPU 013-CCF0R00 und klicken Sie auf [OK].

⇒ Die CPU wird in "*Geräte und Netze*" eingefügt und die "*Gerätekonfiguration*" geöffnet.

Ethernet-PG/OP-Kanal parametrieren

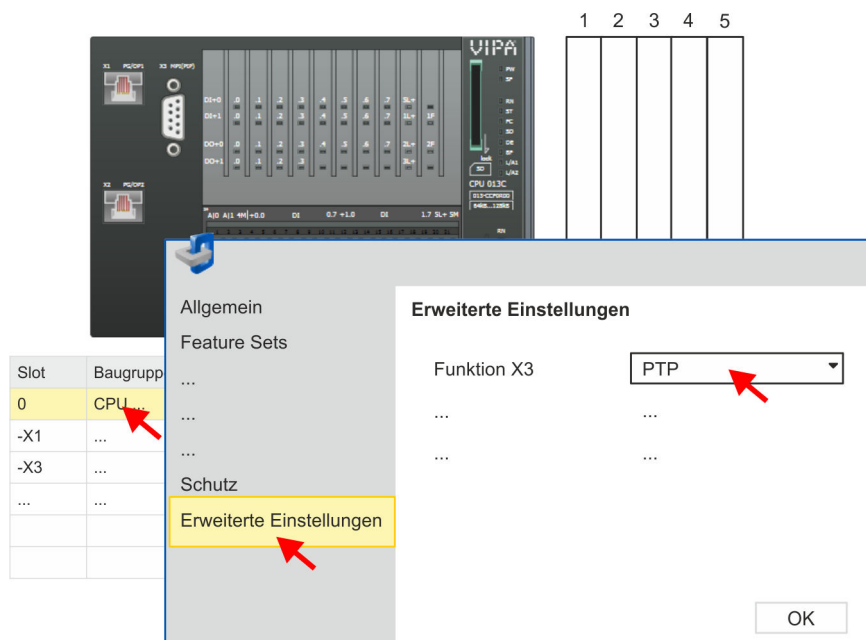
1. Klicken Sie im *Projektbaum* auf "Geräte und Netze".
⇒ Sie erhalten eine grafische Objekt-Ansicht Ihrer CPU.



2. Klicken Sie auf das Netzwerk "PG_OP_Ethernet".
3. Wählen Sie "Kontextmenü → Eigenschaften der Schnittstelle".
⇒ Es öffnet sich ein Dialogfenster. Hier können Sie IP-Adressdaten für Ihren Ethernet-PG/OP-Kanal angeben. Gültige IP-Adress-Parameter erhalten Sie von Ihrem Systemadministrator.
4. Bestätigen Sie Ihre Eingabe mit [OK].
⇒ Die IP-Adressdaten werden in Ihr Projekt übernommen und in "Geräte und Netze" unter "Lokale Baugruppen" aufgelistet.
Nach der Übertragung Ihres Projekts ist Ihre CPU über die angegebenen IP-Adressdaten via Ethernet-PG/OP-Kanal erreichbar.

PtP-Funktionalität aktivieren

1. Klicken Sie im *Projektbaum* auf "PLC... > Gerätekonfiguration".
2. Klicken Sie in der "Gerätekonfiguration" auf "0 CPU 013..." und wählen Sie "Kontextmenü → Eigenschaften der Baugruppe".
⇒ Es öffnet sich der Eigenschaften-Dialog.



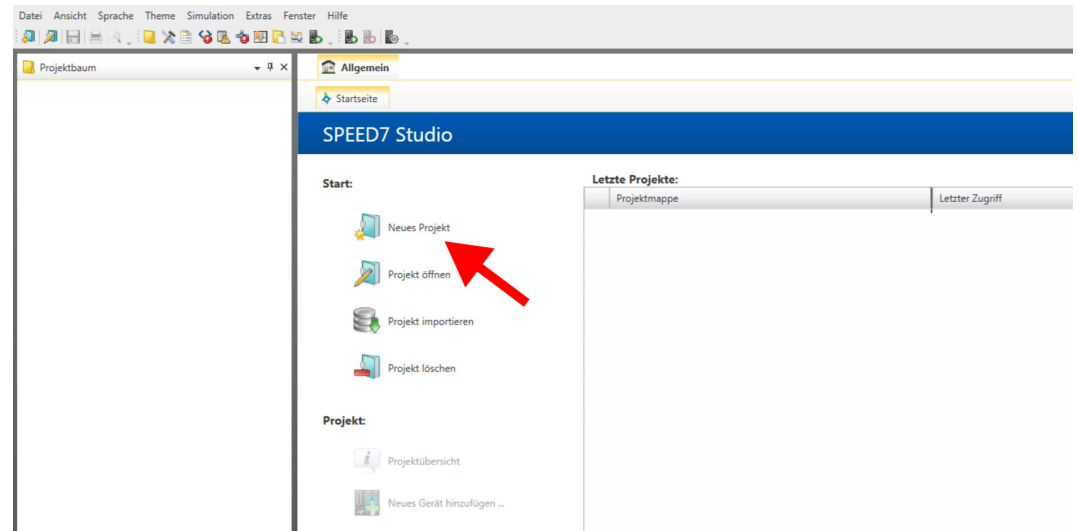
3. Klicken Sie auf "Erweiterte Einstellungen" und stellen Sie unter "Funktion X3" die Funktion "PTP" ein.

6.4.1.3 Hardware-Konfiguration System SLIO CPU 014 ... 017

CPU im Projekt anlegen

Bitte verwenden Sie für die Projektierung das *SPEED7 Studio* ab V1.7.1

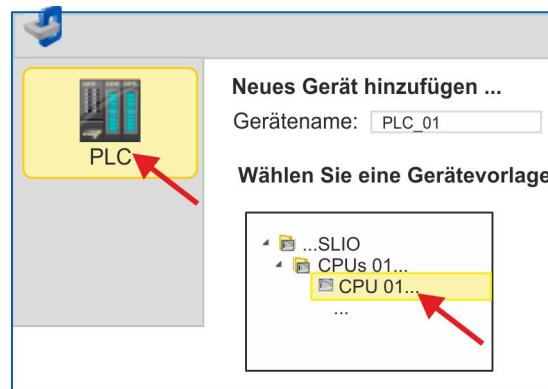
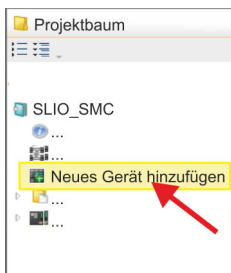
1. Starten Sie das *SPEED7 Studio*.



2. Erstellen sie auf der Startseite mit "Neues Projekt" ein neues Projekt und vergeben Sie einen "Projektnamen".

⇒ Ein neues Projekt wird angelegt und in die Sicht "Geräte und Netze" gewechselt.

3. Klicken Sie im *Projektbaum* auf "Neues Gerät hinzufügen ...".



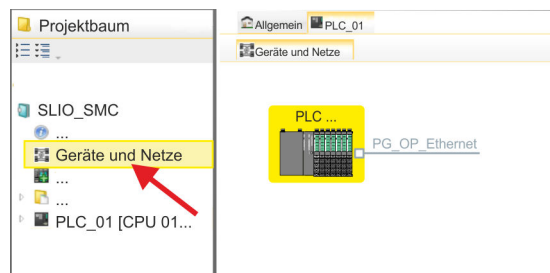
⇒ Es öffnet sich ein Dialog für die Geräteauswahl.

4. Wählen Sie unter den "Gerätevorlagen" die entsprechende System SLIO CPU aus und klicken Sie auf [OK].

⇒ Die CPU wird in "Geräte und Netze" eingefügt und die "Gerätekonfiguration" geöffnet.

Ethernet-PG/OP-Kanal parametrieren

1. ➔ Klicken Sie im *Projektbaum* auf "Geräte und Netze".
⇒ Sie erhalten eine grafische Objekt-Ansicht Ihrer CPU.



2. ➔ Klicken Sie auf das Netzwerk "PG_OP_Ethernet".
3. ➔ Wählen Sie "Kontextmenü ➔ Eigenschaften der Schnittstelle".
⇒ Es öffnet sich ein Dialogfenster. Hier können Sie IP-Adressdaten für Ihren Ethernet-PG/OP-Kanal angeben. Gültige IP-Adress-Parameter erhalten Sie von Ihrem Systemadministrator.
4. ➔ Bestätigen Sie Ihre Eingabe mit [OK].
⇒ Die IP-Adressdaten werden in Ihr Projekt übernommen und in "Geräte und Netze" unter "Lokale Baugruppen" aufgelistet.
Nach der Übertragung Ihres Projekts ist Ihre CPU über die angegebenen IP-Adressdaten via Ethernet-PG/OP-Kanal erreichbar.

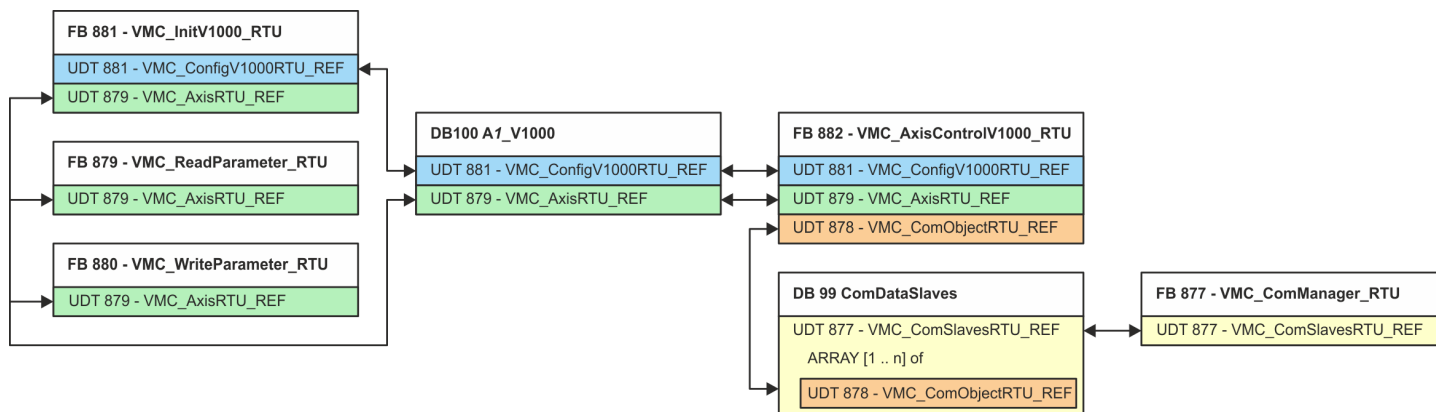
PtP-Funktionalität aktivieren

Bei den System SLIO CPUs 014 ... 017 ist die RS485-Schnittstelle standardmäßig auf PtP-Kommunikation eingestellt. Eine Hardware-Konfiguration zur Einstellung der PtP-Funktionalität ist nicht erforderlich.

6.4.2 Anwender-Programm**6.4.2.1 Programmstruktur****OB 100**

FB 876 - VMC_ConfigMaster_RTU
SFC 216 - SER_CFG

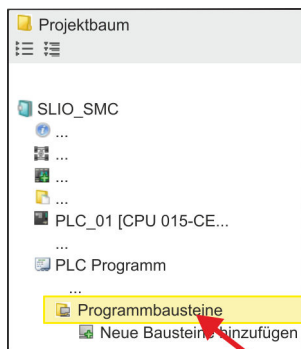
- FB 876 - VMC_ConfigMaster_RTU ⚡ 303
 - Dieser Baustein dient zur Parametrierung der seriellen Schnittstelle der CPU für Modbus RTU Kommunikation.
 - Intern wird der Baustein SFC 216 - SER_CFG aufgerufen.

OB 1

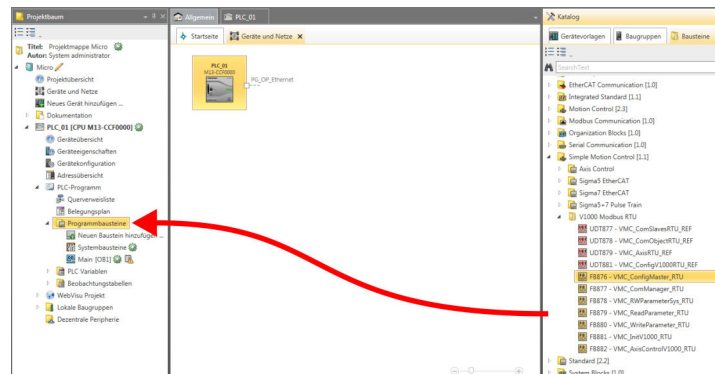
Mit Ausnahme der Bausteine DB 99 und FB 877 müssen Sie für jeden angebundenen Frequenzumrichter die nachfolgend aufgeführten Bausteine anlegen:

- FB 881 - VMC_InitV1000_RTU ↗ 307
 - Der FB 881 - VMC_InitV1000_RTU initialisiert den entsprechenden Frequenzumrichter mit den Nutzerdaten.
 - Bevor ein Frequenzumrichter angesteuert werden kann, muss dieser initialisiert werden.
 - UDT 881 - VMC_ConfigV1000RTU_REF ↗ 303
 - UDT 879 - VMC_AxisRTU_REF ↗ 303
- FB 879 - VMC_ReadParameter_RTU ↗ 305
 - Mit diesem FB haben Sie lesenden Zugriff auf die Parameter eines seriell über Modbus RTU angebundenen Frequenzumrichters.
 - Die gelesenen Daten werden in einem Datenbaustein erfasst.
 - UDT 879 - VMC_AxisRTU_REF ↗ 303
- FB 880 - VMC_WriteParameter_RTU ↗ 306
 - Mit diesem FB haben Sie schreibenden Zugriff auf die Parameter eines seriell über Modbus RTU angebundenen Frequenzumrichters.
 - Die zu schreibenden Daten sind in einem Datenbaustein abzulegen.
 - UDT 879 - VMC_AxisRTU_REF ↗ 303
- DB 100 - A1_V1000
 - Für jeden Frequenzumrichter, welcher seriell über Modbus RTU angekoppelt ist, ist jeweils ein Datenbaustein anzulegen.
 - UDT 879 - VMC_AxisRTU_REF ↗ 303
 - UDT 881 - VMC_ConfigV1000RTU_REF ↗ 303
- FB 882 - VMC_AxisControlV1000_RTU ↗ 309
 - Mit diesem Baustein können Sie einen seriell über Modbus RTU angebundenen Frequenzumrichter steuern und dessen Status abrufen.
 - UDT 881 - VMC_ConfigV1000RTU_REF ↗ 303
 - UDT 879 - VMC_AxisRTU_REF ↗ 303
 - UDT 878 - VMC_ComObjectRTU_REF ↗ 303
- DB 99 - ComDataSlaves
 - Für die Erfassung der Kommunikationsdaten aller Frequenzumrichter, welche seriell über Modbus RTU angekoppelt sind, ist ein gemeinsamer Datenbaustein anzulegen.
 - UDT 877 - VMC_ComSlavesRTU_REF ↗ 303
 - UDT 878 - VMC_ComObjectRTU_REF ↗ 303
- FB 877 - VMC_ComManager_RTU ↗ 305
 - Der Baustein sorgt dafür, dass immer nur 1 Frequenzumrichter (Modbus-Slave) die serielle Schnittstelle benutzen kann. Bei Einsatz mehrerer Frequenzumrichter sendet dieser Baustein als Kommunikations-Manager die Aufträge an die jeweiligen Modbus-Slaves und wertet deren Antworten aus.
 - UDT 877 - VMC_ComSlavesRTU_REF ↗ 303

6.4.2.2 Bausteine in Projekt kopieren



1. Klicken Sie auf "Projektbaum → ...CPU... → PLC-Programm → Programmbausteine".



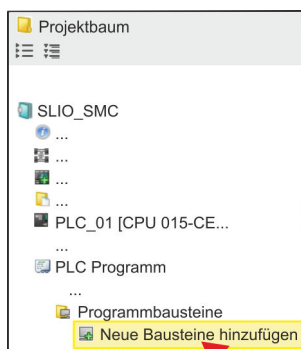
2. Öffnen Sie im "Katalog" unter "Bausteine → Simple Motion Control" die Sammlung "V1000 Modbus RTU" und ziehen Sie per Drag&Drop folgende Bausteine in "Programmbausteine" des Projektbaums:

- FB 876 - VMC_ConfigMaster_RTU
- FB 877 - VMC_ComManager_RTU
- FB 878 - VMC_RWParameterSys_RTU
- FB 879 - VMC_ReadParameter_RTU
- FB 880 - VMC_WriteParameter_RTU
- FB 881 - VMC_InitV1000_RTU
- FB 882 - VMC_AxisControlV1000_RTU

Hierbei werden folgende Bausteine automatisch dem Projekt hinzugefügt:

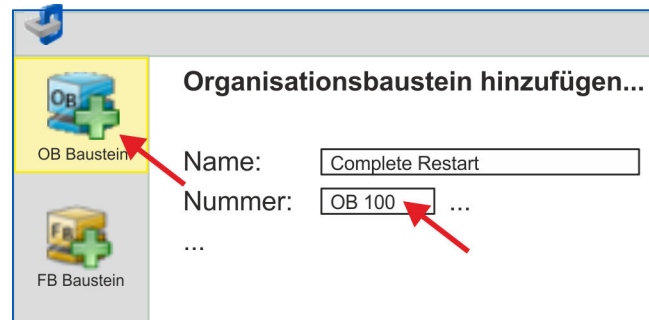
- SEND (FB 60)
- RECEIVE (FB 61)
- RTU_MB_MASTER (FB 72)
- SER_CFG (FC 216)
- SER_SND (FC 217)
- SER_RCV (FC 218)
- VMC_ComSlavesRTU_REF (UDT 877)
- VMC_ComObjectRTU_REF (UDT 878)
- VMC_AxisRTU_REF (UDT 879)
- VMC_ConfigV1000RTU_REF (UDT 881)

6.4.2.3 OB 100 für serielle Kommunikation anlegen



1. Klicken Sie auf "Projektbaum → ...CPU... → PLC-Programm → Programmbausteine → Neuen Baustein hinzufügen".

⇒ Das Dialogfenster "Baustein hinzufügen" öffnet sich.



2. ➤ Geben Sie OB 100 an und bestätigen Sie mit [OK].
⇒ Der OB 100 wird angelegt und geöffnet.
3. ➤ Fügen Sie dem OB 100 einen Call FB876, DB876 zu.
⇒ Der Bausteinaufruf wird angelegt und es öffnet sich ein Dialog zur Angabe des Instanz-Datenbausteins "VMC_ConfigMaster_RTU_876".
4. ➤ Bestätigen Sie die Abfrage des Instanz-Datenbausteins mit [OK].
5. ➤ Geben Sie folgende Parameter vor:

Call FB876, DB876 ➤ Kapitel 6.7.5 "FB 876 - VMC_ConfigMaster_RTU - Modbus RTU CPU-Schnittstelle" auf Seite 303

Baudrate	:= B#16#09	// Baudrate: 09h (9600Bit/s)	IN: BYTE
CharLen	:= B#16#03	// Anzahl Datenbits: 03h (8Bit)	IN: BYTE
Parity	:= B#16#00	// Parität: 0 (none = keine)	IN: BYTE
StopBits	:= B#16#01	// Stopbits: 1 (1Bit)	IN: BYTE
TimeOut	:= W#16#1FFF	// Fehler-Wartezeit: 1FFFh (hoch gewählt)	IN: WORD
Valid	:= "ModbusConfigValid"	// Konfiguration	OUT: BOOL
Error	:= "ModbusConfigError"	// Fehlerrückmeldung	OUT: BOOL
ErrorID	:= "ModbusConfigErrorID"	// Zusätzliche Fehlerinformationen	OUT: WORD

Symbolische Variable

Die symbolischen Variablen erstellen Sie über "Kontextmenü ➔ Symbol erstellen/bearbeiten". Hier können Sie über einen Dialog den entsprechenden Operanden zuordnen.

6.4.2.4 Datenbaustein für Modbus-Slave anlegen

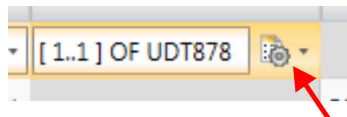
Für jeden Frequenzumrichter, welcher seriell über Modbus RTU angekoppelt ist, ist jeweils ein Datenbaustein anzulegen.

1. ➤ Klicken Sie hierzu auf "Projektbaum ➔ ...CPU... ➔ PLC-Programm ➔ Programmbausteine ➔ Neuen Baustein hinzufügen".
⇒ Das Dialogfenster "Baustein hinzufügen" öffnet sich.
2. ➤ Wählen Sie den Bausteintyp "DB Baustein" und vergeben Sie diesem den Namen "A1_V1000". Die DB-Nr. können Sie frei wählen wie z.B. DB 100. Geben Sie DB 100 an und legen Sie diesen als globalen DB mit [OK] an.
⇒ Der Baustein wird angelegt und geöffnet.

3. ➤ Legen Sie in "A1_V1000" folgende Variablen an:
 - "AxisData" vom Typ UDT 879 - VMC_AxisRTU_REF
 - "V1000Data" vom Typ UDT 881 - VMC_ConfigV1000RTU_REF

6.4.2.5 Anzahl der Modbus-Slaves definieren

Die Anzahl der Frequenzumrichter, welche seriell über Modbus RTU angekoppelt sind, können Sie über die UDT 877 - VMC_ComManager_RTU vorgeben.



1. ➤ Öffnen Sie die UDT 877 - VMC_ComManager_RTU
2. ➤ Stellen Sie in der Variable "Slave" den "Array [1..1] OF" Datentyp auf die Anzahl der Frequenzumrichter ein, welche seriell über Modbus RTU angebunden sind.
Beispielsweise ist bei 3 Frequenzumrichter der Datentyp auf "Array [1..3] OF" zu ändern. Klicken Sie hierzu auf "Datentypeinstellung".
Bitte beachten Sie, dass "OF UDT 878" unverändert bleibt.

6.4.2.6 Datenbaustein für alle Modbus-Slaves anlegen

Für die Erfassung der Kommunikationsdaten aller Frequenzumrichter, welche seriell über Modbus RTU angekoppelt sind, ist ein gemeinsamer Datenbaustein anzulegen.

1. ➤ Klicken Sie hierzu auf "Projektbaum ➔ ...CPU... ➔ PLC-Programm ➔ Programmbausteine ➔ Neuen Baustein hinzufügen".
⇒ Das Dialogfenster "Baustein hinzufügen" öffnet sich.
2. ➤ Wählen Sie den Bausteintyp "DB Baustein" und vergeben Sie diesem den Namen "ComDataSlaves". Die DB-Nr. können Sie frei wählen wie z.B. DB 99. Geben Sie DB 99 an und legen Sie diesen als globalen DB mit [OK] an.
⇒ Der Baustein wird angelegt und geöffnet.
3. ➤ Legen Sie in "ComDataSlaves" folgende Variable an:
 - "Slaves" vom Typ UDT 877 - VMC_ComSlavesRTU_REF

6.4.2.7 OB 1 - Instanz des Kommunikations-Managers anlegen

Der FB 877 - VMC_ComManager_RTU sorgt dafür, dass immer nur 1 Frequenzumrichter (Modbus-Slave) die serielle Schnittstelle benutzen kann. Als Kommunikations-Manager sendet der Baustein die Aufträge an die jeweiligen Modbus-Slaves und wertet deren Antworten aus.

1. ➤ Doppelklicken Sie auf "Projektbaum ➔ ...CPU... ➔ PLC-Programm ➔ Programmbausteine ➔ Main [OB1]".
⇒ Das Programmierfenster für den OB 1 wird geöffnet.
2. ➤ Fügen Sie dem OB 1 einen Call FB877, DB877 zu.
⇒ Der Bausteinaufruf wird angelegt und es öffnet sich ein Dialog zur Angabe des Instanz-Datenbausteins "VMC_ComManager_RTU_877".
3. ➤ Bestätigen Sie die Abfrage des Instanz-Datenbausteins mit [OK].
4. ➤ Geben Sie folgende Parameter vor:

Call FB877, DB877 ➔ Kapitel 6.7.6 "FB 877 - VMC_ComManager_RTU - Modbus RTU Kommunikations-Manager" auf Seite 305

NumberOfSlaves := 1

// Anzahl angebundener Frequenzumrichter: 1

IN: INT

WaitCycles	:= "ComWaitCycles"	// Mindestanzahl Wartezyklen	IN: DINT
SlavesComData	:= "ComDataSlaves.Slave"	// Referenz zu allen Kommunikationobjekten	IN-OUT: UDT 877

6.4.2.8 OB 1 - Instanz der V1000-Initialisierung anlegen

Der FB 881 - VMC_InitV1000_RTU initialisiert den entsprechenden Frequenzumrichter mit den Nutzerdaten. Bevor ein Frequenzumrichter angesteuert werden kann, muss dieser initialisiert werden.

1. ➤ Fügen Sie dem OB 1 einen Call FB881, DB881 hinzu.
 ➔ Der Bausteinanruf wird angelegt und es öffnet sich ein Dialog zur Angabe des Instanz-Datenbausteins "VMC_InitV1000_RTU_881".
2. ➤ Bestätigen Sie die Abfrage des Instanz-Datenbausteins mit [OK].
3. ➤ Geben Sie folgende Parameter vor:

Call FB881, DB881 ➔ Kapitel 6.7.10 "FB 881 - VMC_InitV1000_RTU - Modbus RTU Initialisierung" auf Seite 307

Execute	:= "A1_InitExecute"	// Mit Flanke 0-1 wird der Auftrag ausgeführt.	IN: BOOL
Hardware	:= "A1_InitHardware"	// Angabe der eingesetzten Hardware // 1: System SLIO CP040, 2: SPEED7 CPU	IN: BYTE
Laddr	:= "A1_InitLaddr"	// Logische Adresse bei Einsatz CP040	IN: INT
UnitId	:= "A1_InitUnitId"	// Modbus-Adresse des V1000	IN: BYTE
UserUnitsVelocity	:= "A1_InitUserUnitsVel"	// Benutzereinheit für Geschwindigkeiten // 0: Hz, 1: %, 2: U/min	IN: INT
UserUnitsAcceleration	:= "A1_InitUserUnitsAcc"	// Benutzereinheit Beschleunigung/Verzögerung // 0: 0,01s, 1: 0,1s	IN: INT
MaxVelocityApp	:= "A1_InitMaxVelocityApp"	// Max. Geschwindigkeit in Benutzereinheiten	IN: REAL
Done	:= "A1_InitDone"	// Status Auftrag fertig	OUT: BOOL
Busy	:= "A1_InitBusy"	// Status Auftrag in Bearbeitung	OUT: BOOL
Error	:= "A1_InitError"	// Fehlerrückmeldung	OUT: BOOL
ErrorID	:= "A1_InitErrorID"	// Zusätzliche Fehlerinformationen	OUT: WORD
Axis	:= "A1_V1000".AxisData	// Referenz zu den allgemeinen Achsdaten	IN-OUT: UDT 879
V1000	:= "A1_V1000".V1000Data	// Referenz zu den antriebsspezifischen Daten	IN-OUT: UDT 881

Eingabewerte

Alle Parameter sind mit den entsprechenden Variablen bzw. Operanden zu verschalten. Folgende Eingabe-Parameter sind entsprechend vorzubelegen:

- Hardware
Geben Sie hier die Hardware an, über welche Sie Ihre Frequenzumrichter ansteuern:
 - 1: System SLIO CP040 dessen logische Adresse über *Laddr* vorzugeben ist.
 - 2: SPEED7 CPU
- Laddr
 - Logische Adresse für System SLIO CP040 (*Hardware* = 1). Ansonsten wird dieser Parameter ignoriert.
- UnitId
 - Modbus-Adresse des V1000.

- **UserUnitsVelocity**
Benutzereinheit für Geschwindigkeiten:
 - 0: Hz
Angabe in Hertz
 - 1: %
Angabe als prozentualer Bezug auf die maximale Geschwindigkeit

$$= 2 \cdot f_{\max} / p$$
 mit f_{\max} : max. Ausgabefrequenz (Parameter E1-04)
 p: Anzahl der Motorpole (motorabhängiger Parameter E2-04, E4-04 oder E5-04)
 - 2: U/min
Angabe in Umdrehungen pro Minute
- **UserUnitsAcceleration**
Benutzereinheiten für die Beschleunigung und Verzögerung
 - 0: 0,01s (Wertebereich: 0,00s - 600,00s)
 - 1: 0,1s (Wertebereich: 0,0 - 6000,0s)
- **MaxVelocityApp**
Max. Geschwindigkeit für die Applikation. Die Angabe hat in Benutzereinheiten zu erfolgen und wird bei Bewegungskommandos für den Abgleich verwendet.

6.4.2.9 OB 1 - Instanz Achskontrolle V1000 anlegen

Mit dem FB 882 - VMC_AxisControlV1000_RTU können Sie einen seriell über Modbus RTU angebundene Frequenzumrichter steuern und dessen Status abrufen.

1. ➞ Fügen Sie dem OB 1 einen Call FB882, DB882 hinzu.
 ➞ Der Bausteinanruf wird angelegt und es öffnet sich ein Dialog zur Angabe des Instanz-Datenbausteins "VMC_AxisControlV1000_RTU_882".
2. ➞ Bestätigen Sie die Abfrage des Instanz-Datenbausteins mit [OK].
3. ➞ Geben Sie folgende Parameter vor:

Call FB882, DB882 ➞ *Kapitel 6.7.11 "FB 882 - VMC_AxisControlV1000_RTU - Modbus RTU Achskontrolle" auf Seite 309*

AxisEnable	:= "A1_AxisEnable"	// Aktivierung der Achse	IN: BOOL
AxisReset	:= "A1_AxisReset"	// Kommando: Fehler des V1000 zurücksetzen.	IN: BOOL
StopExecute	:= "A1_StopExecute"	// Kommando: Stop - Achse stoppen	IN: BOOL
MvVelocityExecute	:= "A1_MvVelocityExecute"	// Kommando: MoveVelocity (Geschwindigkeitsregelung)	IN: BOOL
Velocity	:= "A1_Velocity"	// Parameter: Geschwindigkeitsvorgabe für MoveVelocity	IN: REAL
AccelerationTime	:= "A1_AccelerationTime"	// Parameter: Beschleunigungszeit	IN: REAL
DecelerationTime	:= "A1_DecelerationTime"	// Parameter: Verzögerungszeit	IN: REAL
JogPositive	:= "A1_JogPositive"	// Kommando: JogPos	IN: BOOL
JogNegative	:= "A1_JogNegative"	// Kommando: JogNeg	IN: BOOL
JogVelocity	:= "A1_JogVelocity"	// Parameter: Geschwindigkeitsvorgabe für Jogging	IN: REAL
JogAccelerationTime	:= "A1_JogAccelerationTime"	// Parameter: Beschleunigungszeit für Jogging	IN: REAL
JogDecelerationTime	:= "A1_JogDecelerationTime"	// Parameter: Verzögerungszeit für Jogging	IN: REAL
AxisReady	:= "A1_AxisReady"	// Status: Bereitschaft der Achse	OUT: BOOL
AxisEnabled	:= "A1_AxisEnabled"	// Status: Aktivierung der Achse	OUT: BOOL
AxisError	:= "A1_AxisError"	// Status: Achsfehler	OUT: BOOL

AxisErrorID	:= "A1_AxisErrorID"	// Status: Zusätzliche Fehlerinformationen für <i>AxisError</i>	OUT: WORD
DriveError	:= "A1_DriveError"	// Status: Fehler Frequenzumrichter	OUT: BOOL
ActualVelocity	:= "A1_ActualVelocity"	// Status: Aktuelle Geschwindigkeit	OUT: REAL
InVelocity	:= "A1_InVelocity"	// Status Zielgeschwindigkeit	OUT: BOOL
CmdDone	:= "A1_CmdDone"	// Status: Kommando fertig	OUT: BOOL
CmdBusy	:= "A1_CmdBusy"	// Status: Kommando in Bearbeitung	OUT: BOOL
CmdAborted	:= "A1_CmdAborted"	// Status: Kommando abgebrochen	OUT: BOOL
CmdError	:= "A1_CmdError"	// Status: Kommando Fehler	OUT: BOOL
CmdErrorID	:= "A1_CmdErrorID"	// Status: Zusätzliche Fehlerinformationen für <i>CmdError</i>	OUT: WORD
CmdActive	:= "A1_CmdActive"	// Status: Aktives Kommando	OUT: INT
DirectionPositive	:= "A1_DirectionPositive"	// Status: Drehrichtung positiv	OUT: BOOL
DirectionNegative	:= "A1_DirectionNegative"	// Status: Drehrichtung negativ	OUT: BOOL
Axis	:= "A1_V1000".AxisData	// Referenz zu den allgemeinen Achsdaten	IN-OUT: UDT 879
V1000	:= "A1_V1000".V1000Data	// Referenz zu den allgemeinen Daten des // Frequenzumrichters	IN-OUT: UDT 881
AxisComData	:= "ComDataSlaves".Slaves.Slave(1)	// Referenz zu den Kommunikationsdaten	IN-OUT: UDT 878

6.4.2.10 OB 1 - Instanz Parameter lesen anlegen

Mit dem FB 879 - VMC_ReadParameter_RTU haben Sie lesenden Zugriff auf die Parameter eines seriell über Modbus RTU angeordneten Frequenzumrichters. Für die Erfassung der Parameterdaten ist ein DB anzulegen.

1. ➞ Klicken Sie hierzu auf *"Projektbaum ➞ ...CPU... ➞ PLC-Programm ➞ Programmbausteine ➞ Neuen Baustein hinzufügen"*.
⇒ Das Dialogfenster *"Baustein hinzufügen"* öffnet sich.
2. ➞ Wählen Sie den Bausteintyp *"DB Baustein"* und vergeben Sie diesem den Namen *"A1_TransferData"*. Die DB-Nr. können Sie frei wählen wie z.B. DB 98. Geben Sie DB 98 an und legen Sie diesen als globalen DB mit [OK] an.
⇒ Der Baustein wird angelegt und geöffnet.
3. ➞ Legen Sie in *"A1_TransferData"* folgende Variablen an:
 - *"Data_0"* vom Typ WORD
 - *"Data_1"* vom Typ WORD
 - *"Data_2"* vom Typ WORD
 - *"Data_3"* vom Typ WORD
4. ➞ Fügen Sie dem OB 1 einen Call FB879, DB879 hinzu.
⇒ Der Bausteinaufruf wird angelegt und es öffnet sich ein Dialog zur Angabe des Instanz-Datenbausteins *"VMC_ReadParameter_RTU"*.
5. ➞ Bestätigen Sie die Abfrage des Instanz-Datenbausteins mit [OK].
6. ➞ Geben Sie folgende Parameter vor:

Call FB879, DB879 ➞ *Kapitel 6.7.8 "FB 879 - VMC_ReadParameter_RTU - Modbus RTU Parameter lesen" auf Seite 305*

Execute	:= "A1_RdParExecute"	// Mit Flanke 0-1 wird der Auftrag ausgeführt.	IN: BOOL
StartAddress	:= "A1_RdParStartAddress"	// Start-Adresse des 1. Registers	IN: INT

Quantity	:= "A1_RdParQuantity"	// Anzahl der zu lesenden Register	IN: INT
Done	:= "A1_RdParDone"	// Status Auftrag fertig	IN: REAL
Busy	:= "A1_RdParBusy"	// Status Auftrag in Bearbeitung	OUT: BOOL
Error	:= "A1_RdParError"	// Fehlerrückmeldung	OUT: BOOL
ErrorID	:= "A1_RdParErrorID"	// Zusätzliche Fehlerinformationen	OUT: BOOL
Data	:= P#DB98.DBX0.0 BYTES 8	// Ablageort der Parameterdaten	OUT: WORD
Axis	:= "A1_V1000".AxisData	// Referenz zu den allgemeinen Achsdaten	IN-OUT: UDT 879



Bitte beachten Sie, dass immer nur ganze Register als WORD gelesen werden können. Zur Auswertung einzelner Bits müssen Sie High- und Low-Byte vertauschen!

6.4.2.11 OB 1 - Instanz Parameter schreiben anlegen

Mit dem FB 880 - VMC_WriteParameter_RTU haben Sie schreibenden Zugriff auf die Parameter eines seriell über Modbus RTU angebundenen Frequenzumrichters. Für die Erfassung können Sie den für Lesezugriff angelegten DB verwenden - hier DB 98.

1. ➞ Fügen Sie dem OB 1 einen Call FB880, DB880 hinzu.
 ➞ Der Bausteinanruf wird angelegt und es öffnet sich ein Dialog zur Angabe des Instanz-Datenbausteins "VMC_WriteParameter_RTU".
2. ➞ Bestätigen Sie die Abfrage des Instanz-Datenbausteins mit [OK].
3. ➞ Geben Sie folgende Parameter vor:

Call FB880, DB880 ➞ Kapitel 6.7.9 "FB 880 - VMC_WriteParameter_RTU - Modbus RTU Parameter schreiben" auf Seite 306

Execute	:= "A1_WrParExecute"	// Mit Flanke 0-1 wird der Auftrag ausgeführt.	IN: BOOL
StartAddress	:= "A1_WrParStartAddress"	// Start-Adresse des 1. Registers	IN: INT
Quantity	:= "A1_WrParQuantity"	// Anzahl der zu schreibenden Register	IN: INT
Done	:= "A1_WrParDone"	// Status Auftrag fertig	IN: REAL
Busy	:= "A1_WrParBusy"	// Status Auftrag in Bearbeitung	OUT: BOOL
Error	:= "A1_WrParError"	// Fehlerrückmeldung	OUT: BOOL
ErrorID	:= "A1_WrParErrorID"	// Zusätzliche Fehlerinformationen	OUT: BOOL
Data	:= P#DB98.DBX0.0 BYTES 8	// Ablageort der Parameterdaten	OUT: WORD
Axis	:= "A1_V1000".AxisData	// Referenz zu den allgemeinen Achsdaten	IN-OUT: UDT 879

6.4.2.12 Zeitlicher Ablauf

1. ➤ Wählen Sie *"Projekt → Alles übersetzen"* und übertragen Sie das Projekt in Ihre CPU. Näheres zur Übertragung Ihres Projekt finden Sie in der Onlinehilfe zum *SPEED7 Studio*.
 - ⇒ Sie können jetzt Ihre Applikation über die bestehende Kommunikationsverbindung in Betrieb nehmen.

**VORSICHT!**

Bitte beachten Sie immer die Sicherheitshinweise zu ihrem Frequenzumrichter, insbesondere bei der Inbetriebnahme!

2. ➤ Mittels einer Beobachtungstabelle können Sie den Frequenzumrichter manuell steuern. Doppelklicken Sie auf *"Projektbaum → ...CPU... → PLC-Programm → Beobachtungstabellen → Neue Beobachtungstabelle hinzufügen"*.
3. ➤ Geben Sie einen Namen für die Beobachtungstabelle an wie z.B. *"V1000"* und bestätigen Sie mit [OK]
 - ⇒ Die Beobachtungstabelle wird angelegt und für die Bearbeitung geöffnet.
4. ➤ Passen Sie zuerst die Wartezeit zwischen 2 Aufträgen an. Diese beträgt für einen V1000-Frequenzumrichter mindestens 200ms. Stellen Sie hierzu in der Beobachtungstabelle unter *"Namen"* den Bezeichner *"ComWaitCycles"* als *"Dezimal"* ein und geben Sie unter *"Steuerwert"* einen Wert zwischen 200 und 400 vor.



Zur Performance-Steigerung können Sie diesen Wert später nach unten korrigieren, solange Sie keinen Timeout-Fehler (80C8h) erhalten. Bitte beachten Sie hierbei, dass manche Befehle, wie z.B. MoveVelocity aus mehreren Aufträgen bestehen können.

5. ➤ Bevor Sie einen Frequenzumrichter ansteuern können, muss dieser mit dem FB 881 - VMC_InitV1000_RTU initialisiert werden. ➤ Kapitel 6.7.10 *"FB 881 - VMC_InitV1000_RTU - Modbus RTU Initialisierung"* auf Seite 307
 Stellen Sie hierzu in der Beobachtungstabelle unter *"Namen"* den Bezeichner *"A1_InitExecute"* als *"Boolean"* ein und geben Sie unter *"Steuerwert"* den Wert *"True"* vor. Aktivieren Sie *"Steuern"* und starten Sie die Übertragung der Steuerwerte.
 - ⇒ Der Frequenzumrichter wird initialisiert. Nach Abarbeitung liefert der Ausgang *Done TRUE* zurück. Im Fehlerfall können Sie durch Auswertung der *ErrorID* den Fehler ermitteln.



Fahren Sie erst fort, wenn der Init-Baustein keinen Fehler meldet!

6. ➤ Bei erfolgreicher Initialisierung erfolgt zyklisch die Abarbeitung der Register der angebundenen Frequenzumrichter d.h. diese erhalten zyklisch Aufträge. Zur manuellen Steuerung können Sie mit dem FB 882 - VMC_AxisControlV1000_RTU Steuerbefehle an den entsprechenden Frequenzumrichter senden. ➤ Kapitel 6.7.11 *"FB 882 - VMC_AxisControlV1000_RTU - Modbus RTU Achskontrolle"* auf Seite 309
7. ➤ Legen Sie die Parameter des FB 882 - VMC_AxisControlV1000_RTU zur Steuerung und Abfrage in der Beobachtungstabelle an.
8. ➤ Aktivieren Sie durch Setzen von *AxisEnable* die entsprechende Achse. Sobald diese *AxisReady = TRUE* zurückmeldet, können Sie diese mit den entsprechenden Fahrbefehlen ansteuern.

6.5 Einsatz im Siemens SIMATIC Manager

6.5.1 Voraussetzung

Übersicht

- Bitte verwenden Sie für die Projektierung den Siemens SIMATIC Manager ab V 5.5 SP2.
- Bei einer System MICRO CPU wird durch Stecken des Erweiterungsmoduls die PtP-Funktionalität aktiviert. Die Projektierung erfolgt im Siemens SIMATIC Manager in Form eines virtuellen PROFINET IO Devices. Das PROFINET IO Device ist mittels GSDML im Hardware-Katalog zu installieren.
- Bei einer System SLIO CPU 013C erfolgt die Projektierung der PtP-Funktionalität im Siemens SIMATIC Manager in Form eines virtuellen PROFINET IO Devices. Das PROFINET IO Device ist mittels GSDML im Hardware-Katalog zu installieren.
- Bei den System SLIO CPUs 014 ... 017 ist die RS485-Schnittstelle standardmäßig auf PtP-Kommunikation eingestellt. Die Projektierung erfolgt im Siemens SIMATIC Manager in Form eines virtuellen PROFINET IO Devices. Das PROFINET IO Device ist mittels GSDML im Hardware-Katalog zu installieren.

VIPA IO Device installieren

Die Installation des PROFINET VIPA IO Device im Hardware-Katalog erfolgt nach folgender Vorgehensweise:

1. ➤ Gehen Sie in den Service-Bereich von www.vipa.com.
 2. ➤ Laden Sie aus dem Downloadbereich unter "*Config Dateien* ➔ *PROFINET*" die Konfigurationsdatei für Ihre CPU.
 3. ➤ Extrahieren Sie die Datei in Ihr Arbeitsverzeichnis.
 4. ➤ Starten Sie den Hardware-Konfigurator von Siemens.
 5. ➤ Schließen Sie alle Projekte.
 6. ➤ Gehen Sie auf "*Extras* ➔ *GSD-Dateien installieren*".
 7. ➤ Navigieren Sie in Ihr Arbeitsverzeichnis und installieren Sie die entsprechende GSDML-Datei.
- ⇒ Nach der Installation finden Sie das entsprechende PROFINET IO Device unter "*PROFINET IO* ➔ *Weitere Feldgeräte* ➔ *I/O* ➔ *VIPA* ...".

6.5.2 Hardware-Konfiguration

6.5.2.1 Hardware-Konfiguration System MICRO

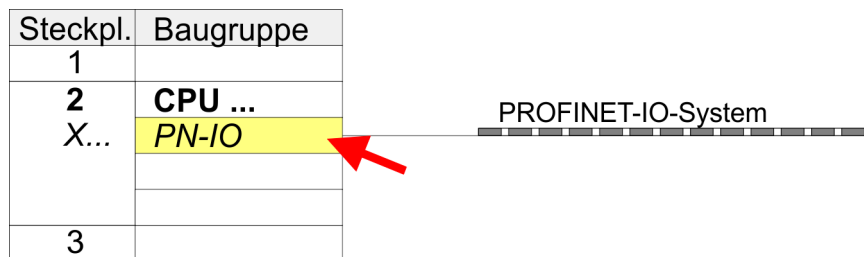
CPU im Projekt anlegen

Steckp..	Baugruppe
1	
2	CPU 314C-2PN/DP
X1	MPI/DP
X2	PN-IO
X2...	Port 1
X2...	Port 2
...	...
3	

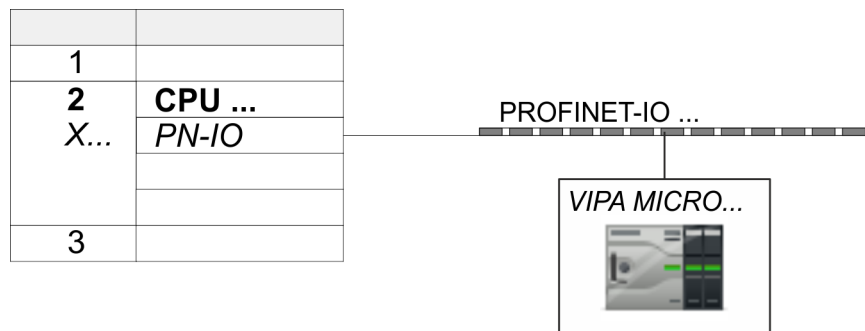
Um kompatibel mit dem Siemens SIMATIC Manager zu sein, sind folgende Schritte durchzuführen:

1. ➤ Starten Sie den Hardware-Konfigurator von Siemens mit einem neuen Projekt.
2. ➤ Fügen Sie aus dem Hardware-Katalog eine Profilschiene ein.
3. ➤ Platzieren Sie auf "Slot"-Nummer 2 die CPU 314C-2 PN/DP (314-6EH04-0AB0 V3.3).
4. ➤ Klicken Sie auf das Submodul "*PN-IO*" der CPU.

5. ➤ Wählen Sie "Kontextmenü → PROFINET IO-System einfügen".



6. ➤ Legen Sie mit [Neu] ein neues Subnetz an und vergeben Sie gültige IP-Adress-Daten.
7. ➤ Klicken Sie auf das Submodul "PN-IO" der CPU und öffnen Sie mit "Kontextmenü → Objekteigenschaften" den Eigenschafts-Dialog.
8. ➤ Geben Sie unter "Allgemein" einen "Gerätenamen" an. Der Geräte name muss eindeutig am Ethernet-Subnetz sein.



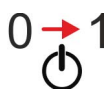
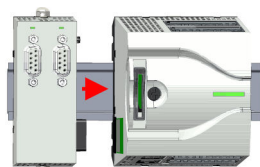
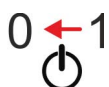
0	VIPA MICRO ...	M13-CCF0000	
X2	M13-CCF0000		
1			
2			
3			
...			

9. ➤ Navigieren Sie im Hardware-Katalog in das Verzeichnis "PROFINET IO → Weitere Feldgeräte → I/O → VIPA ..." und binden Sie z.B. für das System MICRO das IO-Device "M13-CCF0000" an Ihr PROFINET-System an.
- ⇒ In der *Steckplatzübersicht* des PROFINET-IO-Device "VIPA MICRO PLC" ist auf Steckplatz 0 die CPU bereits vorplatziert.

Ethernet-PG/OP-Kanal parametrieren

Steckpl.	Modul
1	
2	CPU ...
X...	PN-IO
3	
4	343-1EX30
5	
...	

1. Platzieren Sie für den Ethernet-PG/OP-Kanal auf Steckplatz 4 den Siemens CP 343-1 (SIMATIC 300 \ CP 300 \ Industrial Ethernet \ CP 343-1 \ 6GK7 343-1EX30 0XE0 V3.0).
2. Öffnen Sie durch Doppelklick auf den CP 343-1EX30 den Eigenschaften-Dialog und geben Sie für den CP unter "*Eigenschaften*" IP-Adress-Daten an. Gültige IP-Adress-Parameter erhalten Sie von Ihrem Systemadministrator.
3. Ordnen Sie den CP einem "*Subnetz*" zu. Ohne Zuordnung werden die IP-Adress-Daten nicht übernommen!

PtP-Funktionalität aktivieren

Eine Hardware-Konfiguration zur Einstellung der PtP-Funktionalität ist nicht erforderlich.

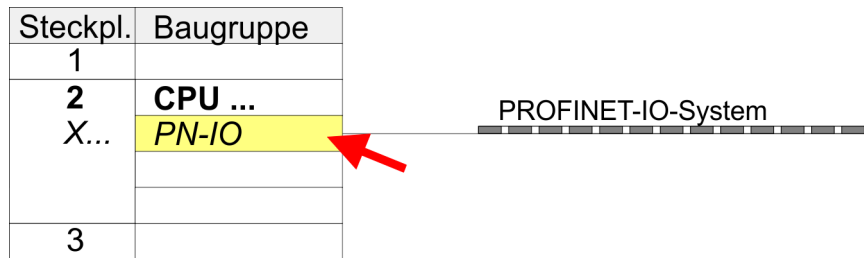
1. Schalten Sie die Spannungsversorgung aus.
2. Montieren Sie das Erweiterungsmodul.
3. Stellen Sie eine Kabelverbindung zum Kommunikationspartner her.
4. Schalten Sie die Spannungsversorgung ein.
⇒ Nach kurzer Hochlaufzeit ist die Schnittstelle X1 PtP bereit für die PtP-Kommunikation.

6.5.2.2 Hardware-Konfiguration System SLIO CPU 013C**CPU im Projekt anlegen**

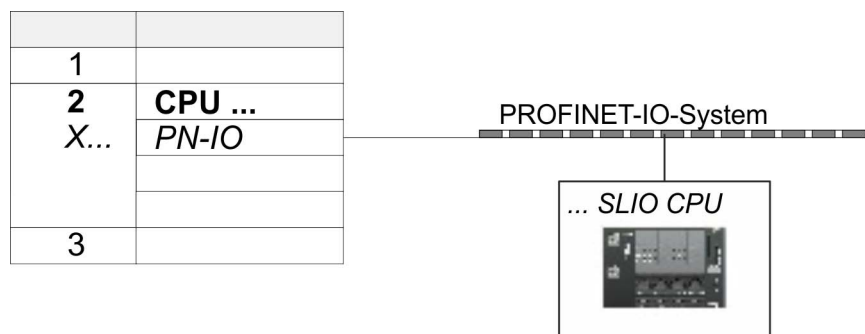
Steckp..	Baugruppe
1	
2	CPU 314C-2PN/DP
X1	MPI/DP
X2	PN-IO
X2...	Port 1
X2...	Port 2
...	...
3	

Um kompatibel mit dem Siemens SIMATIC Manager zu sein, sind folgende Schritte durchzuführen:

1. Starten Sie den Hardware-Konfigurator von Siemens mit einem neuen Projekt.
2. Fügen Sie aus dem Hardware-Katalog eine Profilschiene ein.
3. Platzieren Sie auf "*Slot*"-Nummer 2 die CPU 314C-2 PN/DP (314-6EH04-0AB0 V3.3).
4. Klicken Sie auf das Submodul "*PN-IO*" der CPU.
5. Wählen Sie "*Kontextmenü* → *PROFINET IO-System einfügen*".



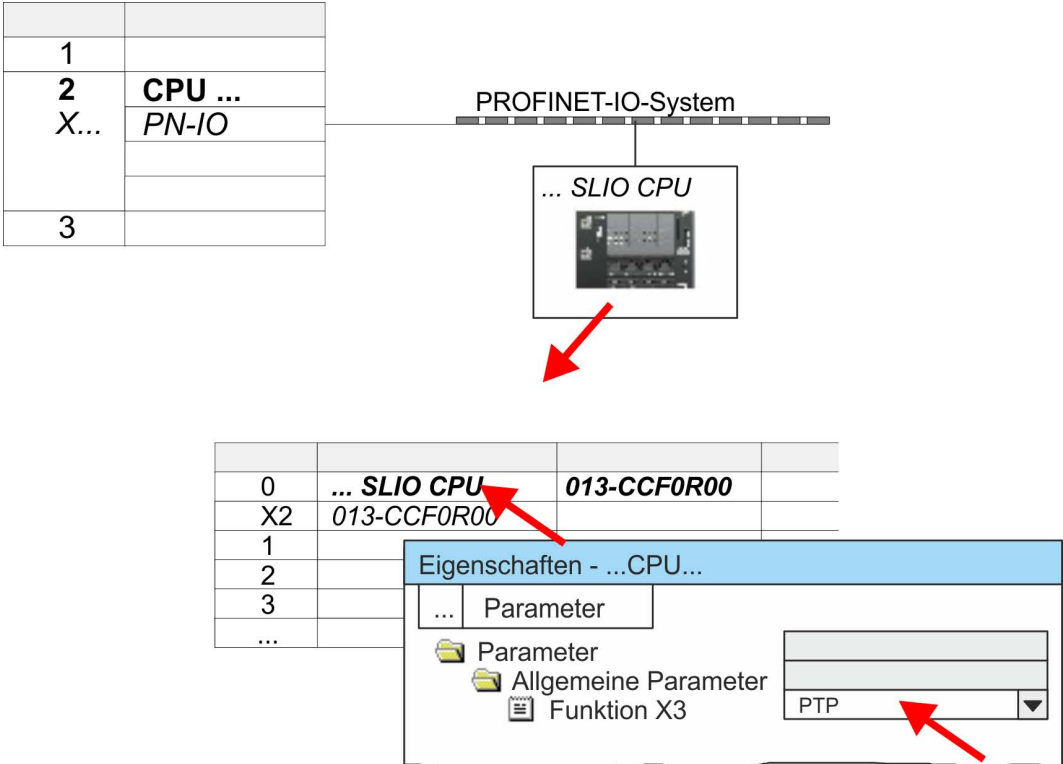
6. ➤ Legen Sie mit [Neu] ein neues Subnetz an und vergeben Sie gültige IP-Adress-Daten für Ihr PROFINET-System.
7. ➤ Klicken Sie auf das Submodul "PN-IO" der CPU und öffnen Sie mit "Kontextmenü ➔ Objekteigenschaften" den Eigenschafts-Dialog.
8. ➤ Geben Sie unter "Allgemein" einen "Gerätenamen" an. Der Geräte name muss eindeutig am Ethernet-Subnetz sein.



0	... SLIO CPU ...	013-CCF0R00	
X2	013-CCF0R00		
1			
2			
3			
...			

9. ➤ Navigieren Sie im Hardware-Katalog in das Verzeichnis "PROFINET IO ➔ Weitere Feldgeräte ➔ I/O ➔ VIPA ..." und binden das IO-Device "013-CCF0R00" CPU an Ihr PROFINET-System an.
 - ⇒ In der Steckplatzübersicht des PROFINET IO Device "VIPA SLIO CPU" ist auf Steckplatz 0 die CPU bereits vorplatziert. Ab Steckplatz 1 können Sie Ihre System SLIO Module platzieren.

PtP-Funktionalität aktivieren



1. Öffnen Sie den Eigenschaften-Dialog, indem Sie auf die "VIPA SLIO CPU" doppelklicken.
⇒ Im Eigenschaft-Dialog haben Sie Zugriff auf die VIPA-spezifischen Parameter.
2. Stellen Sie unter "Funktion X3" den Wert "PTP" ein.

Ethernet-PG/OP-Kanal parametrieren

Steckpl.	Modul
1	
2	CPU ...
X...	PN-IO
3	
4	343-1EX30
5	
...	

1. Platzieren Sie für den Ethernet-PG/OP-Kanal auf Steckplatz 4 den Siemens CP 343-1 (SIMATIC 300 \ CP 300 \ Industrial Ethernet \ CP 343-1 \ 6GK7 343-1EX30 0XE0 V3.0).
2. Öffnen Sie durch Doppelklick auf den CP 343-1EX30 den Eigenschaften-Dialog und geben Sie für den CP unter "Eigenschaften" IP-Adress-Daten an. Gültige IP-Adress-Parameter erhalten Sie von Ihrem Systemadministrator.
3. Ordnen Sie den CP einem "Subnetz" zu. Ohne Zuordnung werden die IP-Adress-Daten nicht übernommen!

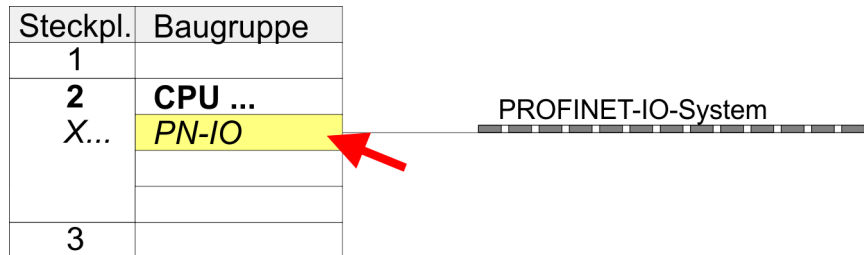
6.5.2.3 Hardware-Konfiguration System SLIO CPU 014 ... 017

CPU im Projekt anlegen

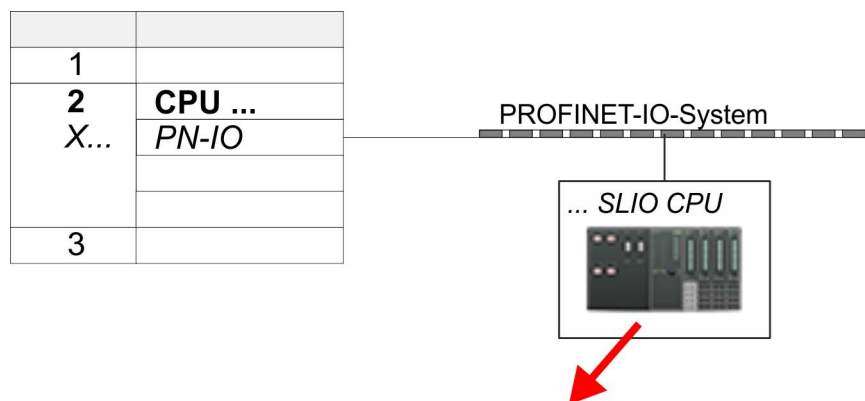
Steckp..	Baugruppe
1	
2	CPU 315-2 PN/DP
X1	MPI/DP
X2	PN-IO
X2...	Port 1
X2...	Port 2
...	...
3	

Um kompatibel mit dem Siemens SIMATIC Manager zu sein, sind folgende Schritte durchzuführen:

1. Starten Sie den Hardware-Konfigurator von Siemens mit einem neuen Projekt.
2. Fügen Sie aus dem Hardware-Katalog eine Profilschiene ein.
3. Platzieren Sie auf "Slot"-Nummer 2 die CPU 315-2 PN/DP (315-2EH14-0AB0 V3.2).
4. Klicken Sie auf das Submodul "PN-IO" der CPU.



5. Legen Sie mit [Neu] ein neues Subnetz an und vergeben Sie gültige IP-Adress-Daten für Ihr PROFINET-System.
6. Klicken Sie auf das Submodul "PN-IO" der CPU und öffnen Sie mit "Kontextmenü → Objekteigenschaften" den Eigenschafts-Dialog.
7. Geben Sie unter "Allgemein" einen "Gerätenamen" an. Der Geräte name muss eindeutig am Ethernet-Subnetz sein.



0	... SLIO CPU	
X2	...		
1			
2			
3			
...			

8. Navigieren Sie im Hardware-Katalog in das Verzeichnis "PROFINET IO → Weitere Feldgeräte → I/O → VIPA ..." und binden das Ihrer CPU entsprechende IO-Device an Ihr PROFINET-System an.
 - ⇒ In der Steckplatzübersicht des PROFINET IO Device "VIPA SLIO CPU" ist auf Steckplatz 0 die CPU bereits vorplatziert. Ab Steckplatz 1 können Sie Ihre System SLIO Module platzieren.

Ethernet-PG/OP-Kanal parametrieren

Steckpl.	Modul
1	
2	CPU ...
X...	PN-IO
3	
4	343-1EX30
5	
...	

1. ➔ Platzieren Sie für den Ethernet-PG/OP-Kanal auf Steckplatz 4 den Siemens CP 343-1 (SIMATIC 300 \ CP 300 \ Industrial Ethernet \ CP 343-1 \ 6GK7 343-1EX30 0XE0 V3.0).
2. ➔ Öffnen Sie durch Doppelklick auf den CP 343-1EX30 den Eigenschaften-Dialog und geben Sie für den CP unter "*Eigenschaften*" IP-Adress-Daten an. Gültige IP-Adress-Parameter erhalten Sie von Ihrem Systemadministrator.
3. ➔ Ordnen Sie den CP einem "*Subnetz*" zu. Ohne Zuordnung werden die IP-Adress-Daten nicht übernommen!

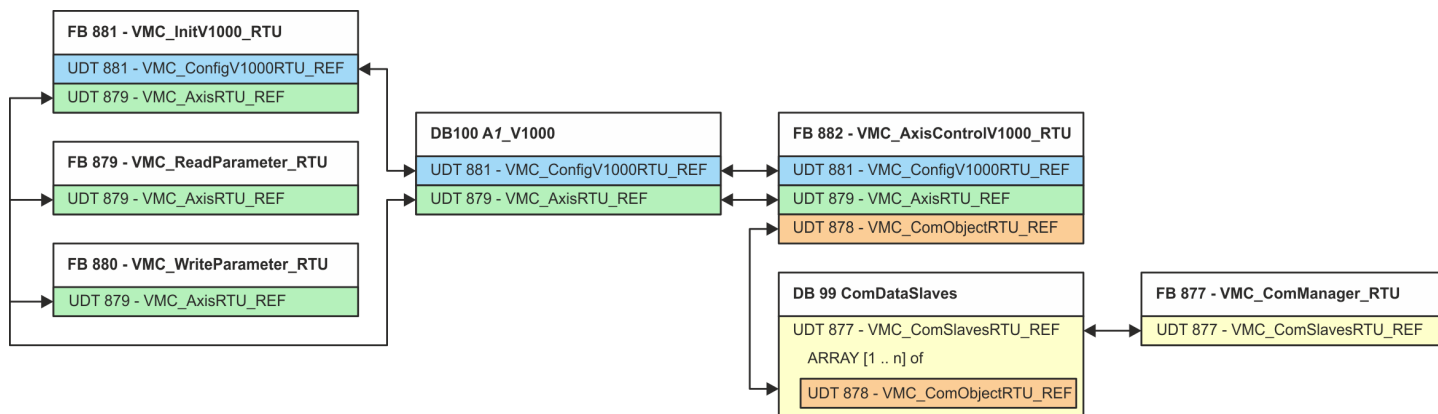
PtP-Funktionalität aktivieren

Bei den System SLIO CPUs 014 ... 017 ist die RS485-Schnittstelle standardmäßig auf PtP-Kommunikation eingestellt. Eine Hardware-Konfiguration zur Einstellung der PtP-Funktionalität ist nicht erforderlich.

6.5.3 Anwender-Programm**6.5.3.1 Programmstruktur****OB 100**

FB 876 - VMC_ConfigMaster_RTU
SFC 216 - SER_CFG

- FB 876 - VMC_ConfigMaster_RTU ⚡ 303
 - Dieser Baustein dient zur Parametrierung der seriellen Schnittstelle der CPU für Modbus RTU Kommunikation.
 - Intern wird der Baustein SFC 216 - SER_CFG aufgerufen.

OB 1

Mit Ausnahme der Bausteine DB 99 und FB 877 müssen Sie für jeden angebundenen Frequenzumrichter die nachfolgend aufgeführten Bausteine anlegen:

- FB 881 - VMC_InitV1000_RTU ↗ 307
 - Der FB 881 - VMC_InitV1000_RTU initialisiert den entsprechenden Frequenzumrichter mit den Nutzerdaten.
 - Bevor ein Frequenzumrichter angesteuert werden kann, muss dieser initialisiert werden.
 - UDT 881 - VMC_ConfigV1000RTU_REF ↗ 303
 - UDT 879 - VMC_AxisRTU_REF ↗ 303
- FB 879 - VMC_ReadParameter_RTU ↗ 305
 - Mit diesem FB haben Sie lesenden Zugriff auf die Parameter eines seriell über Modbus RTU angebundenen Frequenzumrichters.
 - Die gelesenen Daten werden in einem Datenbaustein erfasst.
 - UDT 879 - VMC_AxisRTU_REF ↗ 303
- FB 880 - VMC_WriteParameter_RTU ↗ 306
 - Mit diesem FB haben Sie schreibenden Zugriff auf die Parameter eines seriell über Modbus RTU angebundenen Frequenzumrichters.
 - Die zu schreibenden Daten sind in einem Datenbaustein abzulegen.
 - UDT 879 - VMC_AxisRTU_REF ↗ 303
- DB 100 - A1_V1000
 - Für jeden Frequenzumrichter, welcher seriell über Modbus RTU angekoppelt ist, ist jeweils ein Datenbaustein anzulegen.
 - UDT 879 - VMC_AxisRTU_REF ↗ 303
 - UDT 881 - VMC_ConfigV1000RTU_REF ↗ 303
- FB 882 - VMC_AxisControlV1000_RTU ↗ 309
 - Mit diesem Baustein können Sie einen seriell über Modbus RTU angebundenen Frequenzumrichter steuern und dessen Status abrufen.
 - UDT 881 - VMC_ConfigV1000RTU_REF ↗ 303
 - UDT 879 - VMC_AxisRTU_REF ↗ 303
 - UDT 878 - VMC_ComObjectRTU_REF ↗ 303
- DB 99 - ComDataSlaves
 - Für die Erfassung der Kommunikationsdaten aller Frequenzumrichter, welche seriell über Modbus RTU angekoppelt sind, ist ein gemeinsamer Datenbaustein anzulegen.
 - UDT 877 - VMC_ComSlavesRTU_REF ↗ 303
 - UDT 878 - VMC_ComObjectRTU_REF ↗ 303
- FB 877 - VMC_ComManager_RTU ↗ 305
 - Der Baustein sorgt dafür, dass immer nur 1 Frequenzumrichter (Modbus-Slave) die serielle Schnittstelle benutzen kann. Bei Einsatz mehrerer Frequenzumrichter sendet dieser Baustein als Kommunikations-Manager die Aufträge an die jeweiligen Modbus-Slaves und wertet deren Antworten aus.
 - UDT 877 - VMC_ComSlavesRTU_REF ↗ 303

6.5.3.2 Bausteine in Projekt kopieren

Bibliothek einbinden

1. ➞ Gehen Sie in den Service-Bereich von www.vipa.com.
2. ➞ Laden Sie aus dem Downloadbereich unter "*VIPA Lib*" die *Simple Motion Control Library*.
3. ➞ Öffnen Sie mit "*Datei ➞ Dearchivieren*" das Dialogfenster zur Auswahl der ZIP-Datei.
4. ➞ Wählen Sie die entsprechende ZIP-Datei an und klicken Sie auf [Öffnen].
5. ➞ Geben Sie ein Zielverzeichnis an, in dem die Bausteine abzulegen sind und starten Sie den Entpackvorgang mit [OK].

Bausteine in Projekt kopieren

➔ Öffnen Sie die Bibliothek nach dem Entpackvorgang und ziehen Sie per Drag&Drop alle Bausteine aus *"V1000 Modbus RTU"* in *"Bausteine"* Ihres Projekts:

- FB 876 - VMC_ConfigMaster_RTU
- FB 877 - VMC_ComManager_RTU
- FB 878 - VMC_RWParameterSys_RTU
- FB 879 - VMC_ReadParameter_RTU
- FB 880 - VMC_WriteParameter_RTU
- FB 881 - VMC_InitV1000_RTU
- FB 882 - VMC_AxisControlV1000_RTU
- FB 60 - SEND
- FB 61 - RECEIVE
- FB 72 - RTU MB_MASTER
- FC 216 - SER_CFG
- FC 217 - SER_SND
- FC 218 - SER_RCV
- UDT 877 - VMC_ComSlavesRTU_REF
- UDT 878 - VMC_ComObjectRTU_REF
- UDT 879 - VMC_AxisRTU_REF
- UDT 881 - VMC_ConfigV1000RTU_REF
- SFB 4 - TON

6.5.3.3 OB 100 für serielle Kommunikation anlegen**Alarm-OBs anlegen**

1. ➔ Klicken Sie in Ihrem Projekt auf *"Bausteine"* und wählen Sie *"Kontextmenü ➔ Neues Objekt einfügen ➔ Organisationsbaustein"*.
⇒ Das Dialogfenster *"Eigenschaften Organisationsbaustein"* öffnet sich.
2. ➔ Fügen Sie den OB 100 Ihrem Projekt hinzu.
3. ➔ Öffnen Sie den OB 100.
4. ➔ Fügen Sie dem OB 100 einen `Call FB876, DB876` hinzu.
⇒ Der Bausteinaufruf wird angelegt und es öffnet sich ein Dialog zur Angabe des Instanz-Datenbausteins *"VMC_ConfigMaster_RTU_876"*.
5. ➔ Geben Sie folgende Parameter vor:

`Call FB876, DB876` ➔ Kapitel 6.7.5 *"FB 876 - VMC_ConfigMaster_RTU - Modbus RTU CPU-Schnittstelle"* auf Seite 303

Baudrate	:= B#16#09	// Baudrate: 09h (9600Bit/s)	IN: BYTE
CharLen	:= B#16#03	// Anzahl Datenbits: 03h (8Bit)	IN: BYTE
Parity	:= B#16#00	// Parität: 0 (none = keine)	IN: BYTE
StopBits	:= B#16#01	// Stopbits: 1 (1Bit)	IN: BYTE
TimeOut	:= W#16#1FFF	// Fehler-Wartezeit: 1FFFh (hoch gewählt)	IN: WORD
Valid	:= "ModbusConfigValid"	// Konfiguration	OUT: BOOL
Error	:= "ModbusConfigError"	// Fehlerrückmeldung	OUT: BOOL
ErrorID	:= "ModbusConfigErrorID"	// Zusätzliche Fehlerinformationen	OUT: WORD

Symbolische Variable

Die symbolischen Variablen erstellen Sie über *"Kontextmenü ➔ Symbole bearbeiten"*. Hier können Sie über einen Dialog den entsprechenden Operanden zuordnen.

6.5.3.4 Datenbaustein für Modbus-Slave anlegen

Für jeden Frequenzumrichter, welcher seriell über Modbus RTU angekoppelt ist, ist jeweils ein Datenbaustein anzulegen.

1. ➤ Klicken Sie in Ihrem Projekt auf *"Bausteine"* und wählen Sie *"Kontextmenü ➔ Neues Objekt einfügen ➔ Datenbaustein"*.
⇒ Das Dialogfenster *"Baustein hinzufügen"* öffnet sich.
2. ➤ Geben Sie folgende Parameter an:
 - Name und Typ
 - Die DB-Nr. als *"Name"* können Sie frei wählen wie z.B. DB 100. Geben Sie DB 100 an.
 - Stellen Sie *"Global-DB"* als *"Typ"* ein.
 - Symbolischer Name
 - Geben Sie *"A1_V1000"* an.

Bestätigen Sie Ihre Eingaben mit [OK].

⇒ Der Baustein wird angelegt.
3. ➤ Öffnen Sie DB 100 *"A1_V1000"* durch Doppelklick.
4. ➤ Legen Sie in *"A1_V1000"* folgende Variablen an:
 - *"AxisData"* vom Typ UDT 879 - VMC_AxisRTU_REF
 - *"V1000Data"* vom Typ UDT 881 - VMC_ConfigV1000RTU_REF

6.5.3.5 Anzahl der Modbus-Slaves definieren

Die Anzahl der Frequenzumrichter, welche seriell über Modbus RTU angekoppelt sind, können Sie über die UDT 877 - VMC_ComManager_RTU vorgeben.

UDT 877

Adresse	Name	Typ	...
		Struct	
...	Slave	Array[1..1]	
...		"VMC_ComObjectRTU_REF"	
...		END_STRUCT	

1. ➤ Öffnen Sie die UDT 877 - VMC_ComManager_RTU unter *"Bausteine"*.
2. ➤ Stellen Sie in der Variable *"Slave"* den *"Array [1..1]"* Datentyp auf die Anzahl der Frequenzumrichter ein, welche seriell über Modbus RTU angebunden sind.
Beispielsweise ist bei 3 Frequenzumrichter der Datentyp auf *"Array [1..3]"* zu ändern.
Bitte beachten Sie, dass der Rest unverändert bleibt.

6.5.3.6 Datenbaustein für alle Modbus-Slaves anlegen

Für die Erfassung der Kommunikationsdaten aller Frequenzumrichter, welche seriell über Modbus RTU angekoppelt sind, ist ein gemeinsamer Datenbaustein anzulegen.

1. ➤ Klicken Sie in Ihrem Projekt auf *"Bausteine"* und wählen Sie *"Kontextmenü ➔ Neues Objekt einfügen ➔ Datenbaustein"*.
⇒ Das Dialogfenster *"Baustein hinzufügen"* öffnet sich.
2. ➤ Geben Sie folgende Parameter an:
 - Name und Typ
 - Die DB-Nr. als *"Name"* können Sie frei wählen wie z.B. DB 99. Geben Sie DB 99 an.
 - Stellen Sie *"Global-DB"* als *"Typ"* ein.
 - Symbolischer Name
 - Geben Sie *"ComDataSlaves"* an.

Bestätigen Sie Ihre Eingaben mit [OK].

⇒ Der Baustein wird angelegt.

3. ➤ Öffnen Sie DB 99 "ComDataSlaves" durch Doppelklick.
4. ➤ Legen Sie in "ComDataSlaves" folgende Variable an:
 - "Slaves" vom Typ UDT 877 - VMC_ComSlavesRTU_REF

6.5.3.7 OB 1 - Instanz des Kommunikations-Managers anlegen

Der FB 877 - VMC_ComManager_RTU sorgt dafür, dass immer nur 1 Frequenzumrichter (Modbus-Slave) die serielle Schnittstelle benutzen kann. Als Kommunikations-Manager sendet der Baustein die Aufträge an die jeweiligen Modbus-Slaves und wertet deren Antworten aus.

1. ➤ Öffnen Sie den OB 1.
2. ➤ Fügen Sie dem OB 1 einen Call FB877, DB877 hinzu.
 - ⇒ Der Bausteinaufruf wird angelegt und es öffnet sich ein Dialog zur Angabe des Instanz-Datenbausteins "VMC_ComManager_RTU_877".
3. ➤ Bestätigen Sie die Abfrage des Instanz-Datenbausteins mit [OK].
4. ➤ Geben Sie folgende Parameter vor:

Call FB877, DB877 ➤ Kapitel 6.7.6 "FB 877 - VMC_ComManager_RTU - Modbus RTU Kommunikations-Manager" auf Seite 305

NumberOfSlaves	:= 1	// Anzahl angebundener Frequenzumrichter: 1	IN: INT
WaitCycles	:= "ComWaitCycles"	// Mindestanzahl Wartezyklen	IN: DINT
SlavesComData	:= "ComDataSlaves.Slave"	// Referenz zu allen Kommunikationobjekten	IN-OUT: UDT 877

6.5.3.8 OB 1 - Instanz der V1000-Initialisierung anlegen

Der FB 881 - VMC_InitV1000_RTU initialisiert den entsprechenden Frequenzumrichter mit den Nutzerdaten. Bevor ein Frequenzumrichter angesteuert werden kann, muss dieser initialisiert werden.

1. ➤ Fügen Sie dem OB 1 einen Call FB881, DB881 hinzu.
 - ⇒ Der Bausteinaufruf wird angelegt und es öffnet sich ein Dialog zur Angabe des Instanz-Datenbausteins "VMC_InitV1000_RTU_881".
2. ➤ Bestätigen Sie die Abfrage des Instanz-Datenbausteins mit [OK].
3. ➤ Geben Sie folgende Parameter vor:

Call FB881, DB881 ➤ Kapitel 6.7.10 "FB 881 - VMC_InitV1000_RTU - Modbus RTU Initialisierung" auf Seite 307

Execute	:= "A1_InitExecute"	// Mit Flanke 0-1 wird der Auftrag ausgeführt.	IN: BOOL
Hardware	:= "A1_InitHardware"	// Angabe der eingesetzten Hardware	IN: BYTE
		// 1: System SLIO CP040, 2: SPEED7 CPU	
Laddr	:= "A1_InitLaddr"	// Logische Adresse bei Einsatz CP040	IN: INT
UnitId	:= "A1_InitUnitId"	// Modbus-Adresse des V1000	IN: BYTE
UserUnitsVelocity	:= "A1_InitUserUnitsVel"	// Benutzereinheit für Geschwindigkeiten	IN: INT
		// 0: Hz, 1: %, 2: U/min	
UserUnitsAcceleration	:= "A1_InitUserUnitsAcc"	// Benutzereinheit Beschleunigung/Verzögerung	IN: INT
		// 0: 0,01s, 1: 0,1s	

Einsatz im Siemens SIMATIC Manager > Anwender-Programm

MaxVelocityApp	:= "A1_InitMaxVelocityApp"	// Max. Geschwindigkeit in Benutzereinheiten	IN: REAL
Done	:= "A1_InitDone"	// Status Auftrag fertig	OUT: BOOL
Busy	:= "A1_InitBusy"	// Status Auftrag in Bearbeitung	OUT: BOOL
Error	:= "A1_InitError"	// Fehlerrückmeldung	OUT: BOOL
ErrorID	:= "A1_InitErrorID"	// Zusätzliche Fehlerinformationen	OUT: WORD
Axis	:= "A1_V1000".AxisData	// Referenz zu den allgemeinen Achsdaten	IN-OUT: UDT 879
V1000	:= "A1_V1000".V1000Data	// Referenz zu den antriebsspezifischen Daten	IN-OUT: UDT 881

Eingabewerte

Alle Parameter sind mit den entsprechenden Variablen bzw. Operanden zu verschalten. Folgende Eingabe-Parameter sind entsprechend vorzubelegen:

- **Hardware**
Geben Sie hier die Hardware an, über welche Sie Ihre Frequenzumrichter ansteuern:
 - 1: System SLIO CP040 dessen logische Adresse über *Laddr* vorzugeben ist.
 - 2: SPEED7 CPU
- **Laddr**
 - Logische Adresse für System SLIO CP040 (*Hardware* = 1). Ansonsten wird dieser Parameter ignoriert.
- **UnitId**
 - Modbus-Adresse des *V1000*.
- **UserUnitsVelocity**
Benutzereinheit für Geschwindigkeiten:
 - 0: Hz
Angabe in Hertz
 - 1: %
Angabe als prozentualer Bezug auf die maximale Geschwindigkeit

$$= 2 \cdot f_{\max} / p$$
 mit f_{\max} : max. Ausgabefrequenz (Parameter E1-04)
 p: Anzahl der Motorpole (motorabhängiger Parameter E2-04, E4-04 oder E5-04)
 - 2: U/min
Angabe in Umdrehungen pro Minute
- **UserUnitsAcceleration**
Benutzereinheiten für die Beschleunigung und Verzögerung
 - 0: 0,01s (Wertebereich: 0,00s - 600,00s)
 - 1: 0,1s (Wertebereich: 0,0 - 6000,0s)
- **MaxVelocityApp**
Max. Geschwindigkeit für die Applikation. Die Angabe hat in Benutzereinheiten zu erfolgen und wird bei Bewegungskommandos für den Abgleich verwendet.

6.5.3.9 OB 1 - Instanz Achskontrolle V1000 anlegen

Mit dem FB 882 - VMC_AxisControlV1000_RTU können Sie einen seriell über Modbus RTU angebundenen Frequenzumrichter steuern und dessen Status abrufen.

1. ➤ Fügen Sie dem OB 1 einen `Call FB882, DB882` hinzu.
 - ⇒ Der Bausteinanruf wird angelegt und es öffnet sich ein Dialog zur Angabe des Instanz-Datenbausteins "*VMC_AxisControlV1000_RTU_882*".
2. ➤ Bestätigen Sie die Abfrage des Instanz-Datenbausteins mit [OK].
3. ➤ Geben Sie folgende Parameter vor:

Call FB882, DB882 ↗ **Kapitel 6.7.11 "FB 882 - VMC_AxisControlV1000_RTU - Modbus RTU Achskontrolle" auf Seite 309**

AxisEnable	:= "A1_AxisEnable"	// Aktivierung der Achse	IN: BOOL
AxisReset	:= "A1_AxisReset"	// Kommando: Fehler des V1000 zurücksetzen.	IN: BOOL
StopExecute	:= "A1_StopExecute"	// Kommando: <i>Stop</i> - Achse stoppen	IN: BOOL
MvVelocityExecute	:= "A1_MvVelocityExecute"	// Kommando: <i>MoveVelocity</i> (Geschwindigkeitsregelung)	IN: BOOL
Velocity	:= "A1_Velocity"	// Parameter: Geschwindigkeitsvorgabe für MoveVelocity	IN: REAL
AccelerationTime	:= "A1_AccelerationTime"	// Parameter: Beschleunigungszeit	IN: REAL
DecelerationTime	:= "A1_DecelerationTime"	// Parameter: Verzögerungszeit	IN: REAL
JogPositive	:= "A1_JogPositive"	// Kommando: <i>JogPos</i>	IN: BOOL
JogNegative	:= "A1_JogNegative"	// Kommando: <i>JogNeg</i>	IN: BOOL
JogVelocity	:= "A1_JogVelocity"	// Parameter: Geschwindigkeitsvorgabe für Jogging	IN: REAL
JogAccelerationTime	:= "A1_JogAccelerationTime"	// Parameter: Beschleunigungszeit für Jogging	IN: REAL
JogDecelerationTime	:= "A1_JogDecelerationTime"	// Parameter: Verzögerungszeit für Jogging	IN: REAL
AxisReady	:= "A1_AxisReady"	// Status: Bereitschaft der Achse	OUT: BOOL
AxisEnabled	:= "A1_AxisEnabled"	// Status: Aktivierung der Achse	OUT: BOOL
AxisError	:= "A1_AxisError"	// Status: Achsfehler	OUT: BOOL
AxisErrorID	:= "A1_AxisErrorID"	// Status: Zusätzliche Fehlerinformationen für <i>AxisError</i>	OUT: WORD
DriveError	:= "A1_DriveError"	// Status: Fehler Frequenzumrichter	OUT: BOOL
ActualVelocity	:= "A1_ActualVelocity"	// Status: Aktuelle Geschwindigkeit	OUT: REAL
InVelocity	:= "A1_InVelocity"	// Status: Zielgeschwindigkeit	OUT: BOOL
CmdDone	:= "A1_CmdDone"	// Status: Kommando fertig	OUT: BOOL
CmdBusy	:= "A1_CmdBusy"	// Status: Kommando in Bearbeitung	OUT: BOOL
CmdAborted	:= "A1_CmdAborted"	// Status: Kommando abgebrochen	OUT: BOOL
CmdError	:= "A1_CmdError"	// Status: Kommando Fehler	OUT: BOOL
CmdErrorID	:= "A1_CmdErrorID"	// Status: Zusätzliche Fehlerinformationen für <i>CmdError</i>	OUT: WORD
CmdActive	:= "A1_CmdActive"	// Status: Aktives Kommando	OUT: INT
DirectionPositive	:= "A1_DirectionPositive"	// Status: Drehrichtung positiv	OUT: BOOL
DirectionNegative	:= "A1_DirectionNegative"	// Status: Drehrichtung negativ	OUT: BOOL
Axis	:= "A1_V1000".AxisData	// Referenz zu den allgemeinen Achsdaten	IN-OUT: UDT 879
V1000	:= "A1_V1000".V1000Data	// Referenz zu den allgemeinen Daten des // Frequenzumrichters	IN-OUT: UDT 881
AxisComData	:= "ComDataSlaves".Slaves.Slave(1)	// Referenz zu den Kommunikationsdaten	IN-OUT: UDT 878

6.5.3.10 OB 1 - Instanz Parameter lesen anlegen

Mit dem FB 879 - VMC_ReadParameter_RTU haben Sie lesenden Zugriff auf die Parameter eines seriell über Modbus RTU angebotenen Frequenzumrichters. Für die Erfassung der Parameterdaten ist ein DB anzulegen.

1. ➤ Klicken Sie in Ihrem Projekt auf **"Bausteine"** und wählen Sie **"Kontextmenü ➔ Neues Objekt einfügen ➔ Datenbaustein"**.

⇒ Das Dialogfenster **"Baustein hinzufügen"** öffnet sich.

2. ➔ Geben Sie folgende Parameter an:

- Name und Typ
 - Die DB-Nr. als "*Name*" können Sie frei wählen wie z.B. DB 98. Geben Sie DB 98 an.
 - Stellen Sie "*Global-DB*" als "*Typ*" ein.
- Symbolischer Name
 - Geben Sie "A1_TransferData" an.

Bestätigen Sie Ihre Eingaben mit [OK].

⇒ Der Baustein wird angelegt.

3. ➔ Öffnen Sie DB 98 "A1_TransferData" durch Doppelklick.**4.** ➔ Legen Sie in "A1_TransferData" folgende Variablen an:

- "*Data_0*" vom Typ WORD
- "*Data_1*" vom Typ WORD
- "*Data_2*" vom Typ WORD
- "*Data_3*" vom Typ WORD

5. ➔ Fügen Sie dem OB 1 einen Call FB879, DB879 hinzu.

⇒ Der Bausteinaufruf wird angelegt und es öffnet sich ein Dialog zur Angabe des Instanz-Datenbausteins "*VMC_ReadParameter_RTU*".

6. ➔ Bestätigen Sie die Abfrage des Instanz-Datenbausteins mit [OK].**7.** ➔ Geben Sie folgende Parameter vor:

Call FB879, DB879 ➔ *Kapitel 6.7.8 "FB 879 - VMC_ReadParameter_RTU - Modbus RTU Parameter lesen" auf Seite 305*

Execute	:= "A1_RdParExecute"	// Mit Flanke 0-1 wird der Auftrag ausgeführt.	IN: BOOL
StartAddress	:= "A1_RdParStartAddress"	// Start-Adresse des 1. Registers	IN: INT
Quantity	:= "A1_RdParQuantity"	// Anzahl der zu lesenden Register	IN: INT
Done	:= "A1_RdParDone"	// Status Auftrag fertig	IN: REAL
Busy	:= "A1_RdParBusy"	// Status Auftrag in Bearbeitung	OUT: BOOL
Error	:= "A1_RdParError"	// Fehlerrückmeldung	OUT: BOOL
ErrorID	:= "A1_RdParErrorID"	// Zusätzliche Fehlerinformationen	OUT: BOOL
Data	:= P#DB98.DBX0.0 BYTES 8	// Ablageort der Parameterdaten	OUT: WORD
Axis	:= "A1_V1000".AxisData	// Referenz zu den allgemeinen Achsdaten	IN-OUT: UDT 879



Bitte beachten Sie, dass immer nur ganze Register als WORD gelesen werden können. Zur Auswertung einzelner Bits müssen Sie High- und Low-Byte vertauschen!

6.5.3.11 OB 1 - Instanz Parameter schreiben anlegen

Mit dem FB 880 - VMC_WriteParameter_RTU haben Sie schreibenden Zugriff auf die Parameter eines seriell über Modbus RTU angebotenen Frequenzumrichters. Für die Erfassung können Sie den für Lesezugriff angelegten DB verwenden - hier DB 98.

1. ➔ Fügen Sie dem OB 1 einen Call FB880, DB880 hinzu.

⇒ Der Bausteinaufruf wird angelegt und es öffnet sich ein Dialog zur Angabe des Instanz-Datenbausteins "*VMC_WriteParameter_RTU*".

2. ➤ Bestätigen Sie die Abfrage des Instanz-Datenbausteins mit [OK].
3. ➤ Geben Sie folgende Parameter vor:

Call FB880, DB880 ↗ *Kapitel 6.7.9 "FB 880 - VMC_WriteParameter_RTU - Modbus RTU Parameter schreiben" auf Seite 306*

Execute	:= "A1_WrParExecute"	// Mit Flanke 0-1 wird der Auftrag ausgeführt.	IN: BOOL
StartAddress	:= "A1_WrParStartAddress"	// Start-Adresse des 1. Registers	IN: INT
Quantity	:= "A1_WrParQuantity"	// Anzahl der zu schreibenden Register	IN: INT
Done	:= "A1_WrParDone"	// Status Auftrag fertig	IN: REAL
Busy	:= "A1_WrParBusy"	// Status Auftrag in Bearbeitung	OUT: BOOL
Error	:= "A1_WrParError"	// Fehlerrückmeldung	OUT: BOOL
ErrorID	:= "A1_WrParErrorID"	// Zusätzliche Fehlerinformationen	OUT: BOOL
Data	:= P#DB98.DBX0.0 BYTES 8	// Ablageort der Parameterdaten	OUT: WORD
Axis	:= "A1_V1000".AxisData	// Referenz zu den allgemeinen Achsdaten	IN-OUT: UDT 879

6.5.3.12 Zeitlicher Ablauf

1. ➤ Speichern Sie Ihr Projekt mit *"Station ➔ Speichern und übersetzen"*.
2. ➤ Transferieren Sie Ihr Projekt in Ihre CPU.
⇒ Sie können jetzt Ihre Applikation in Betrieb nehmen.



VORSICHT!

Bitte beachten Sie immer die Sicherheitshinweise zu ihrem Frequenzumrichter, insbesondere bei der Inbetriebnahme!

3. ➤ Mittels einer Beobachtungstabelle können Sie den Frequenzumrichter manuell steuern. Zum Anlegen einer Beobachtungstabelle wählen Sie *"Zielsystem ➔ Variable beobachten/steuern"*.
⇒ Die Beobachtungstabelle wird angelegt und für die Bearbeitung geöffnet.
4. ➤ Passen Sie zuerst die Wartezeit zwischen 2 Aufträgen an. Diese beträgt für einen V1000-Frequenzumrichter mindestens 200ms. Stellen Sie hierzu in der Beobachtungstabelle unter *"Symbol"* den Bezeichner *"ComWaitCycles"* als *"DEZ"* ein und geben Sie unter *"Steuerwert"* einen Wert zwischen 200 und 400 vor.



Zur Performance-Steigerung können Sie diesen Wert später nach unten korrigieren, solange Sie keinen Timeout-Fehler (80C8h) erhalten. Bitte beachten Sie hierbei, dass manche Befehle, wie z.B. MoveVelocity aus mehreren Aufträgen bestehen können.

5. ➔ Bevor Sie einen Frequenzumrichter ansteuern können, muss dieser mit dem FB 881 - VMC_InitV1000_RTU initialisiert werden. ➔ *Kapitel 6.7.10 "FB 881 - VMC_InitV1000_RTU - Modbus RTU Initialisierung" auf Seite 307*

Stellen Sie hierzu in der Beobachtungstabelle unter "Symbol" den Bezeichner "A1_InitExecute" als "BOOL" ein und geben Sie unter "Steuerwert" den Wert "True" vor. Aktivieren Sie "Steuern" und starten Sie die Übertragung der Steuerwerte.

- ⇒ Der Frequenzumrichter wird initialisiert. Nach Abarbeitung liefert der Ausgang Done TRUE zurück. Im Fehlerfall können Sie durch Auswertung der ErrorID den Fehler ermitteln.



Fahren Sie erst fort, wenn der Init-Baustein keinen Fehler meldet!

6. ➔ Bei erfolgreicher Initialisierung erfolgt zyklisch die Abarbeitung der Register der angebundenen Frequenzumrichter d.h. diese erhalten zyklisch Aufträge. Zur manuellen Steuerung können Sie mit dem FB 882 - VMC_AxisControlV1000_RTU Steuerbefehle an den entsprechenden Frequenzumrichter senden. ➔ *Kapitel 6.7.11 "FB 882 - VMC_AxisControlV1000_RTU - Modbus RTU Achskontrolle" auf Seite 309*
7. ➔ Legen Sie die Parameter des FB 882 - VMC_AxisControlV1000_RTU zur Steuerung und Abfrage in der Beobachtungstabelle an.
8. ➔ Speichern Sie die Beobachtungstabelle unter einem Namen wie z.B. "V1000".
9. ➔ Aktivieren Sie durch Setzen von *AxisEnable* die entsprechende Achse. Sobald diese *AxisReady* = TRUE zurückmeldet, können Sie diese mit den entsprechenden Fahrbefehlen ansteuern.

6.6 Einsatz im Siemens TIA Portal

6.6.1 Voraussetzung

Übersicht

- Bitte verwenden Sie für die Projektierung das Siemens TIA Portal ab V 14.
- Bei einer System MICRO CPU wird durch Stecken des Erweiterungsmoduls die PtP-Funktionalität aktiviert. Die Projektierung erfolgt im Siemens TIA Portal in Form eines virtuellen PROFINET IO Devices. Das PROFINET IO Device ist mittels GSDML im Hardware-Katalog zu installieren.
- Bei einer System SLIO CPU 013C erfolgt die Projektierung der PtP-Funktionalität im Siemens TIA Portal in Form eines virtuellen PROFINET IO Devices. Das PROFINET IO Device ist mittels GSDML im Hardware-Katalog zu installieren.
- Bei den System SLIO CPUs 014 ... 017 ist die RS485-Schnittstelle standardmäßig auf PtP-Kommunikation eingestellt. Die Projektierung erfolgt im Siemens TIA Portal in Form eines virtuellen PROFINET IO Devices. Das PROFINET IO Device ist mittels GSDML im Hardware-Katalog zu installieren.

VIPA IO Device installieren

Die Installation des PROFINET VIPA IO Device im Hardware-Katalog erfolgt nach folgender Vorgehensweise:

1. ➔ Gehen Sie in den Service-Bereich von www.vipa.com.
2. ➔ Laden Sie aus dem Downloadbereich unter "Config Dateien ➔ PROFINET" die Konfigurationsdatei für Ihre CPU.
3. ➔ Extrahieren Sie die Datei in Ihr Arbeitsverzeichnis.
4. ➔ Starten das Siemens TIA Portal.
5. ➔ Schließen Sie alle Projekte.

6. ➔ Wechseln Sie in die *Projektansicht*.
7. ➔ Gehen Sie auf *"Extras ➔ Gerätebeschreibungsdatei (GSD) installieren"*.
8. ➔ Navigieren Sie in Ihr Arbeitsverzeichnis und installieren Sie die entsprechende GSDML-Datei.
 - ⇒ Nach der Installation wird der Hardware-Katalog aktualisiert und das Siemens TIA Portal beendet.

Nach einem Neustart des Siemens TIA Portals finden Sie das entsprechende PROFINET-IO-Device unter *Weitere Feldgeräte > PROFINET > IO > VIPA GmbH >*



Damit die VIPA-Komponenten angezeigt werden können, müssen Sie im Hardware-Katalog bei "Filter" den Haken entfernen.

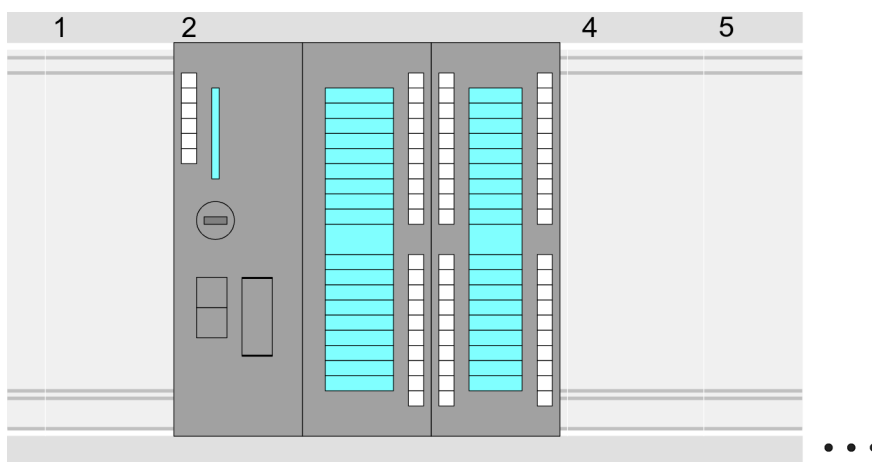
6.6.2 Hardware-Konfiguration

6.6.2.1 Hardware-Konfiguration System MICRO

CPU im Projekt anlegen

Um kompatibel mit dem Siemens TIA Portal zu sein, sind folgende Schritte durchzuführen:

1. ➔ Starten Sie das Siemens TIA Portal mit einem neuen Projekt.
 2. ➔ Wechseln Sie in die *Projektansicht*.
 3. ➔ Klicken Sie in der *Projektnavigation* auf *"Neues Gerät hinzufügen"*.
 4. ➔ Wählen Sie im Eingabedialog folgende CPU aus:
SIMATIC S7-300 > CPU 314C-2 PN/DP (314-6EH04-0AB0 V3.3)
- ⇒ Die CPU wird mit einer Profilschiene eingefügt.



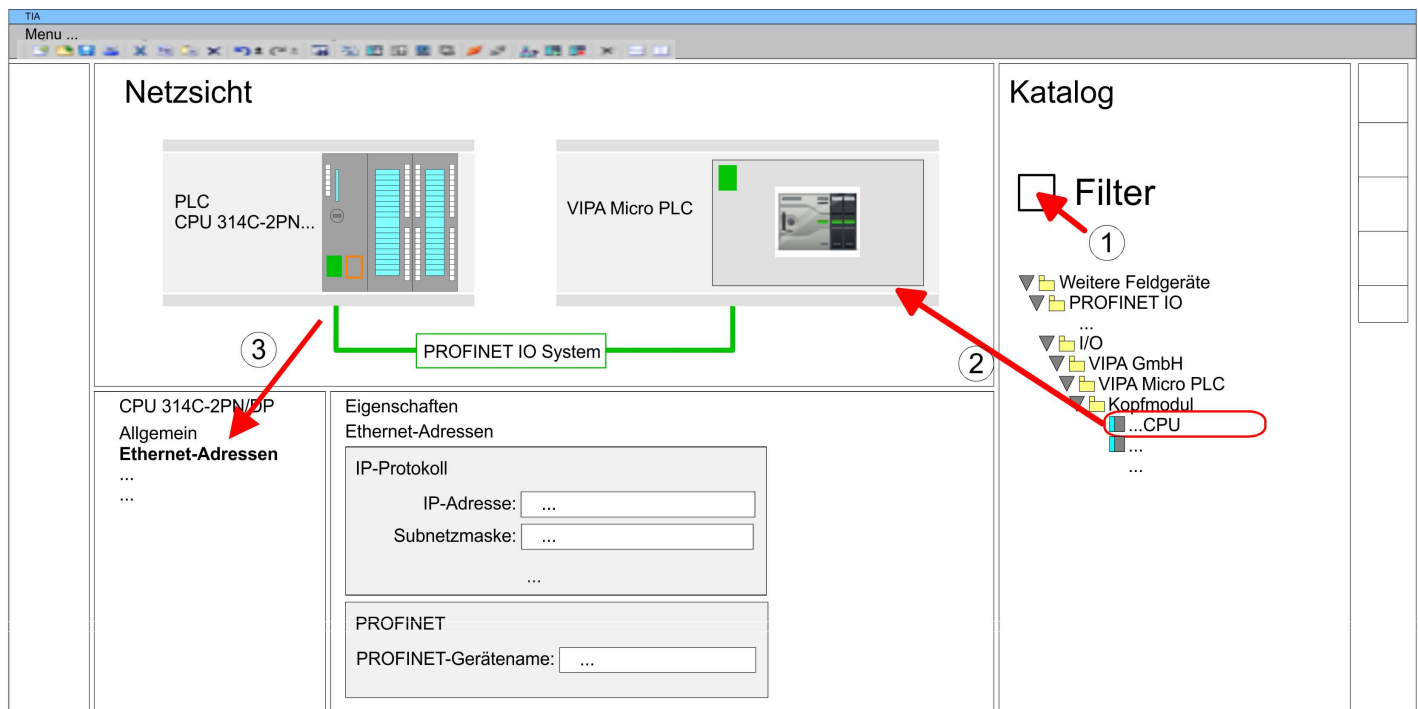
Geräteübersicht:

Baugruppe	...	Steckplatz	...	Typ	...
PLC...		2		CPU 314C-2PN/DP	
MPI-Schnittstelle...		2 X1		MPI/DP-Schnittstelle	

PROFINET-Schnitt...	2 X2	PROFINET-Schnittstelle
DI24/DO16...	2 5	DI24/DO16
AI5/AO2...	2 6	AI5/AO2
Zählen...	2 7	Zählen
...		

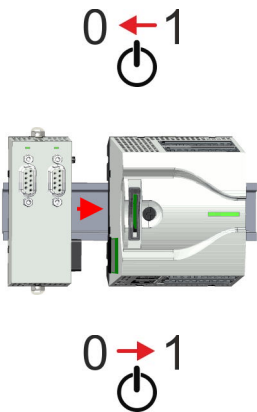
Anbindung CPU als PROFINET-IO-Device

1. ➤ Wechseln Sie im *Projektbereich* in die *"Netzsicht"*.
2. ➤ Nach der Installation der GSDML finden Sie das IO-Device für die CPU im Hardware-Katalog unter *Weitere Feldgeräte > PROFINET > IO > VIPA GmbH > VIPA MICRO PLC*. Binden Sie das Slave-System an die CPU an, indem Sie dies aus dem Hardware-Katalog in die *Netzsicht* ziehen und dieses über PROFINET an die CPU anbinden.
3. ➤ Klicken Sie in der *Netzsicht* auf den PROFINET-Teil der Siemens CPU und geben Sie in *"Eigenschaften"* unter *"Ethernet-Adressen"* im Bereich *"IP-Protokoll"* gültige IP-Adressdaten an.
4. ➤ Geben Sie unter *"PROFINET"* einen *"PROFINET Gerätenamen"* an. Der Geräte-name muss eindeutig am Ethernet-Subnetz sein.



5. ➤ Wählen Sie in der *Netzsicht* das IO-Device *"VIPA MICRO PLC"* an und wechseln Sie in die *Geräteübersicht*.
⇒ In der *Geräteübersicht* des PROFINET-IO-Device *"VIPA MICRO PLC"* ist auf Steckplatz 0 die CPU bereits vorplatziert.

PtP-Funktionalität aktivieren

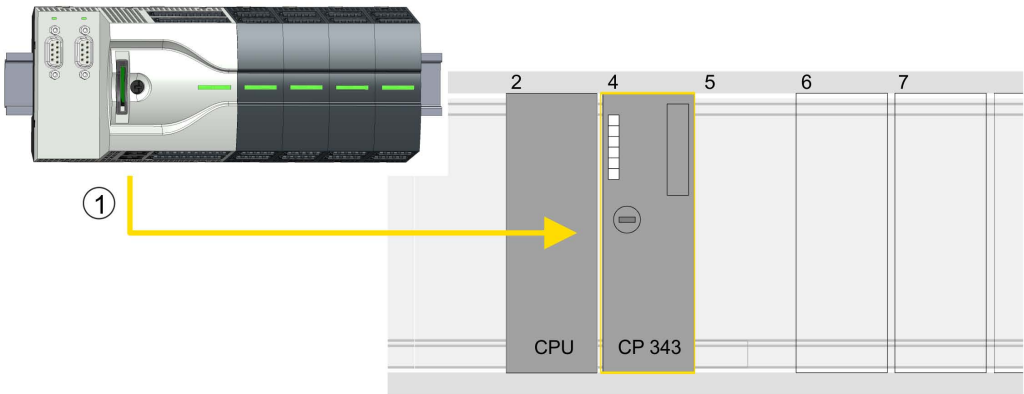


Eine Hardware-Konfiguration zur Einstellung der PtP-Funktionalität ist nicht erforderlich.

- 1. Schalten Sie die Spannungsversorgung aus.
- 2. Montieren Sie das Erweiterungsmodul.
- 3. Stellen Sie eine Kabelverbindung zum Kommunikationspartner her.
- 4. Schalten Sie die Spannungsversorgung ein.
⇒ Nach kurzer Hochlaufzeit ist die Schnittstelle X1 PtP bereit für die PtP-Kommunikation.

Ethernet-PG/OP-Kanal parametrieren

- 1. Platzieren Sie für den Ethernet-PG/OP-Kanal auf Steckplatz 4 den Siemens CP 343-1 (6GK7 343-1EX30 0XE0 V3.0).
- 2. Öffnen Sie durch Klick auf den CP 343-1EX30 den "Eigenschaften"-Dialog und geben Sie für den CP in den "Eigenschaften" unter "Ethernet-Adresse" IP-Adress-Daten an. Gültige IP-Adress-Parameter erhalten Sie von Ihrem Systemadministrator.



1 Ethernet-PG/OP-Kanal

Geräteübersicht

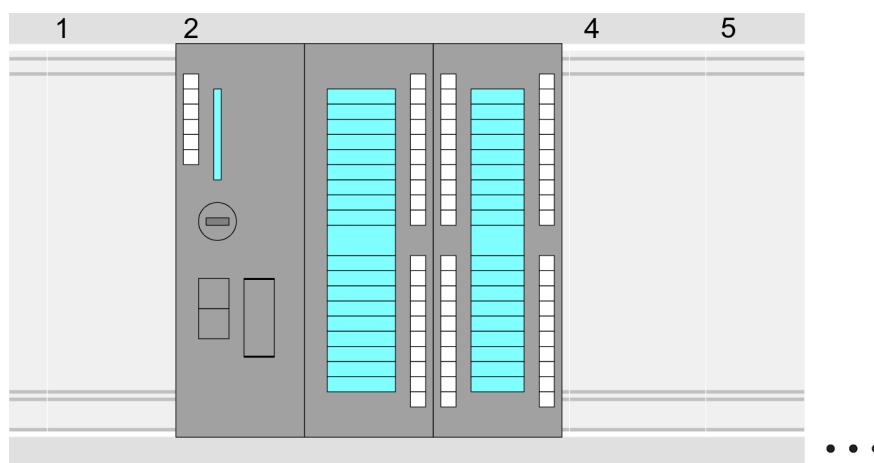
Baugruppe	...	Steckplatz	...	Typ	...
PLC ...		2		CPU 314C-2PN/DP	
MPI/DP-Schnittstelle		2 X1		MPI/DP-Schnittstelle	
PROFINET-Schnittstelle		2 X2		PROFINET-Schnittstelle	
...		
CP 343-1		4		CP 343-1	
...		

6.6.2.2 Hardware-Konfiguration System SLIO CPU 013C

CPU im Projekt anlegen

Um kompatibel mit dem Siemens TIA Portal zu sein, sind folgende Schritte durchzuführen:

1. ➔ Starten Sie das Siemens TIA Portal mit einem neuen Projekt.
2. ➔ Wechseln Sie in die *Projektansicht*.
3. ➔ Klicken Sie in der *Projektnavigation* auf "*Neues Gerät hinzufügen*".
4. ➔ Wählen Sie im Eingabedialog folgende CPU aus:
SIMATIC S7-300 > CPU 314C-2 PN/DP (314-6EH04-0AB0 V3.3)
⇒ Die CPU wird mit einer Profilschiene eingefügt.

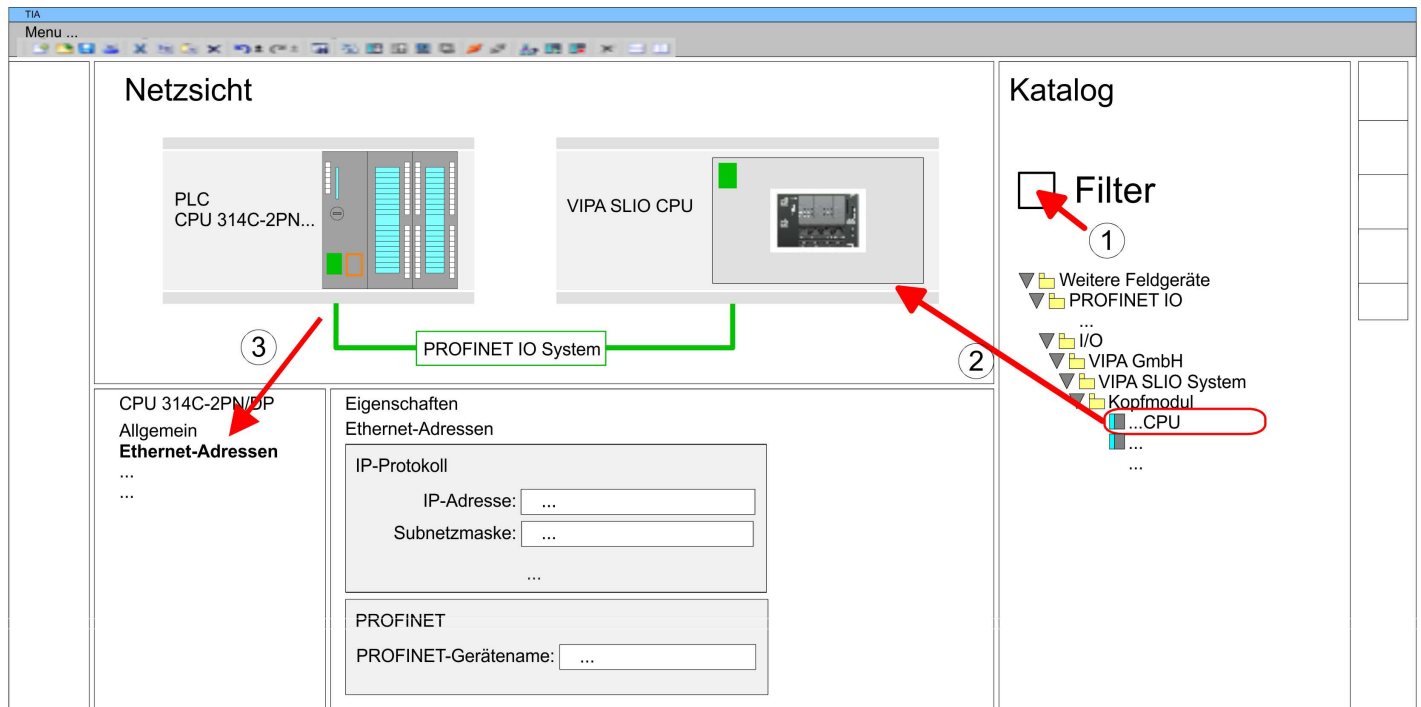


Geräteübersicht:

Baugruppe	...	Steckplatz	...	Typ	...
PLC...		2		CPU 314C-2PN/DP	
MPI-Schnittstelle...		2 X1		MPI/DP-Schnittstelle	
PROFINET-Schnitt...		2 X2		PROFINET-Schnittstelle	
DI24/DO16...		2 5		DI24/DO16	
AI5/AO2...		2 6		AI5/AO2	
Zählen...		2 7		Zählen	
...					

Anbindung CPU als PROFINET-IO-Device

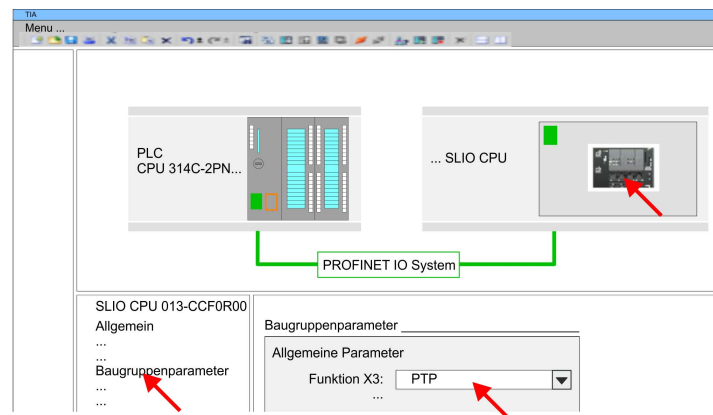
1. ➔ Wechseln Sie im *Projektbereich* in die "*Netzansicht*".
2. ➔ Nach der Installation der GSDML finden Sie das IO-Device für die SLIO CPU im Hardware-Katalog unter *Weitere Feldgeräte* > *PROFINET* > *IO* > *VIPA GmbH* > *VIPA SLIO System*. Binden Sie das Slave-System an die CPU an, indem Sie dies aus dem Hardware-Katalog in die *Netzansicht* ziehen und dieses über PROFINET an die CPU anbinden.
3. ➔ Klicken Sie in der *Netzansicht* auf den PROFINET-Teil der Siemens CPU und geben Sie in "*Eigenschaften*" unter "*Ethernet-Adressen*" im Bereich "*IP-Protokoll*" gültige IP-Adressdaten an.
4. ➔ Geben Sie unter "*PROFINET*" einen "*PROFINET Gerätenamen*" an. Der Geräte-name muss eindeutig am Ethernet-Subnetz sein.



5. ➤ Wählen Sie in der *Netzwerkansicht* das IO-Device "*VIPA SLIO CPU*" an und wechseln Sie in die *Geräteübersicht*.

⇒ In der *Geräteübersicht* des PROFINET-IO-Device "*VIPA SLIO CPU*" ist auf Steckplatz 0 die CPU bereits vorplatziert.

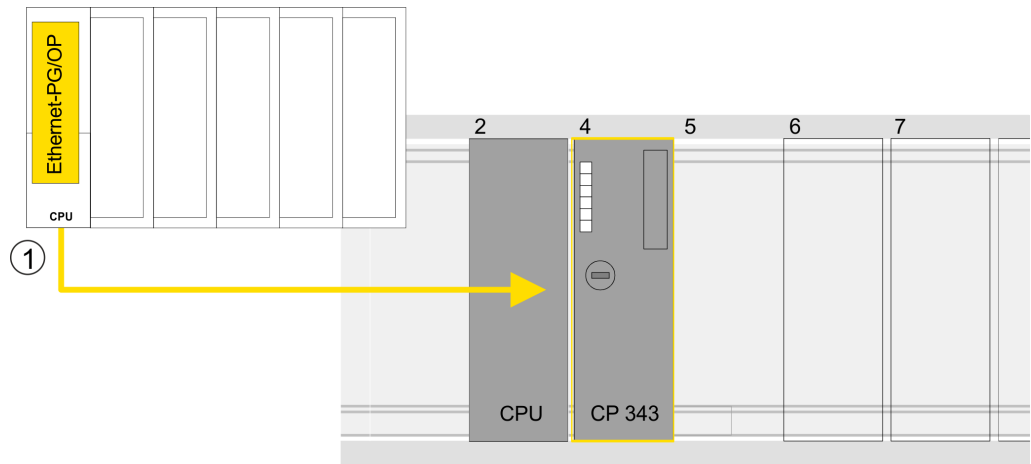
PtP-Funktionalität aktivieren



1. ➤ Öffnen Sie den Eigenschaften-Dialog, indem Sie auf die "*VIPA SLIO CPU*" doppelklicken.
2. ➤ Stellen Sie unter "*Funktion X3*" den Wert "*PTP*" ein.

Ethernet-PG/OP-Kanal parametrieren

1. ➤ Platzieren Sie für den Ethernet-PG/OP-Kanal auf Steckplatz 4 den Siemens CP 343-1 (6GK7 343-1EX30 0XE0 V3.0).
2. ➤ Öffnen Sie durch Klick auf den CP 343-1EX30 den "Eigenschaften"-Dialog und geben Sie für den CP in den "Eigenschaften" unter "Ethernet-Adresse" IP-Adress-Daten an. Gültige IP-Adress-Parameter erhalten Sie von Ihrem Systemadministrator.



1 Ethernet-PG/OP-Kanal

Geräteübersicht

Baugruppe	...	Steckplatz	...	Typ	...
PLC ...		2		CPU 315-2 PN/DP	
MPI/DP-Schnittstelle		2 X1		MPI/DP-Schnittstelle	
PROFINET-Schnittstelle		2 X2		PROFINET-Schnittstelle	
...		
CP 343-1		4		CP 343-1	
...		

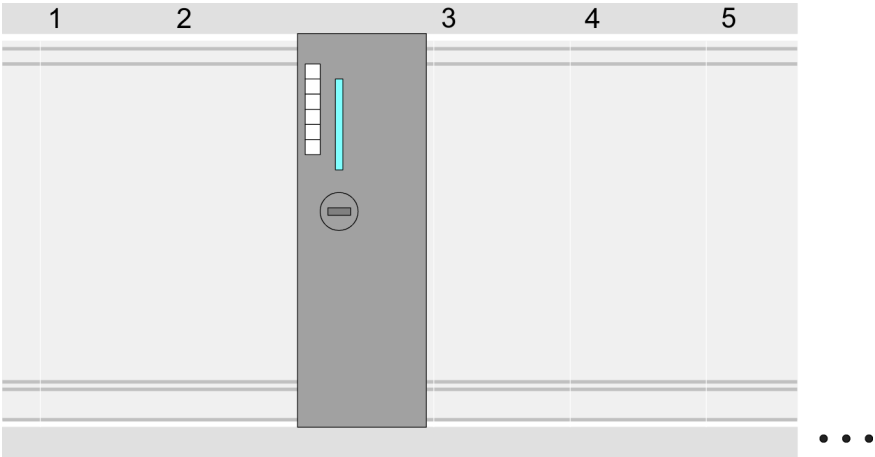
6.6.2.3 Hardware-Konfiguration System SLIO CPU 014 ... 017

CPU im Projekt anlegen

Um kompatibel mit dem Siemens TIA Portal zu sein, sind folgende Schritte durchzuführen:

1. Starten Sie das Siemens TIA Portal mit einem neuen Projekt.
2. Wechseln Sie in die *Projektansicht*.
3. Klicken Sie in der *Projektnavigation* auf "*Neues Gerät hinzufügen*".

4. ➔ Wählen Sie im Eingabedialog folgende CPU aus:
SIMATIC S7-300 > CPU 315-2 PN/DP (315-2EH14-0AB0 V3.2)
⇒ Die CPU wird mit einer Profilschiene eingefügt.

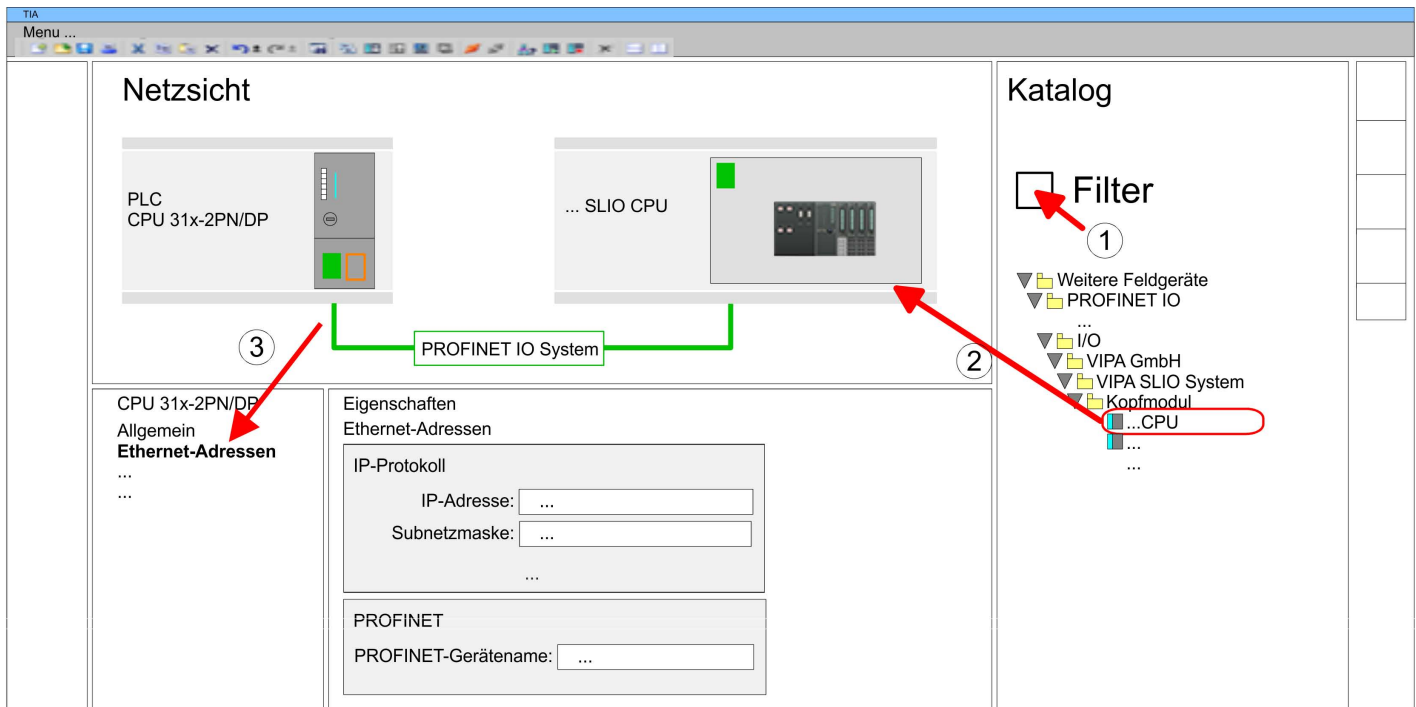


Geräteübersicht

Baugruppe	...	Steckplatz	...	Typ	...
PLC ...		2		CPU 315-2 PN/DP	
MPI/DP-Schnittstelle		2 X1		MPI/DP-Schnittstelle	
PROFINET-Schnittstelle		2 X2		PROFINET-Schnittstelle	
...		

Anbindung CPU als PROFINET-IO-Device

1. ➔ Wechseln Sie im *Projektbereich* in die *"Netzsicht"*.
2. ➔ Nach der Installation der GSDML finden Sie das IO-Device für die SLIO CPU im Hardware-Katalog unter *Weitere Feldgeräte > PROFINET > IO > VIPA GmbH > VIPA SLIO System*. Binden Sie das Slave-System an die CPU an, indem Sie dies aus dem Hardware-Katalog in die *Netzsicht* ziehen und dieses über PROFINET an die CPU anbinden.
3. ➔ Klicken Sie in der *Netzsicht* auf den PROFINET-Teil der Siemens CPU und geben Sie in *"Eigenschaften"* unter *"Ethernet-Adressen"* im Bereich *"IP-Protokoll"* gültige IP-Adressdaten an.
4. ➔ Geben Sie unter *"PROFINET"* einen *"PROFINET Gerätenamen"* an. Der Geräte-name muss eindeutig am Ethernet-Subnetz sein.



5. ➤ Wählen Sie in der *Netzansicht* das IO-Device "VIPA SLIO CPU" an und wechseln Sie in die *Geräteübersicht*.

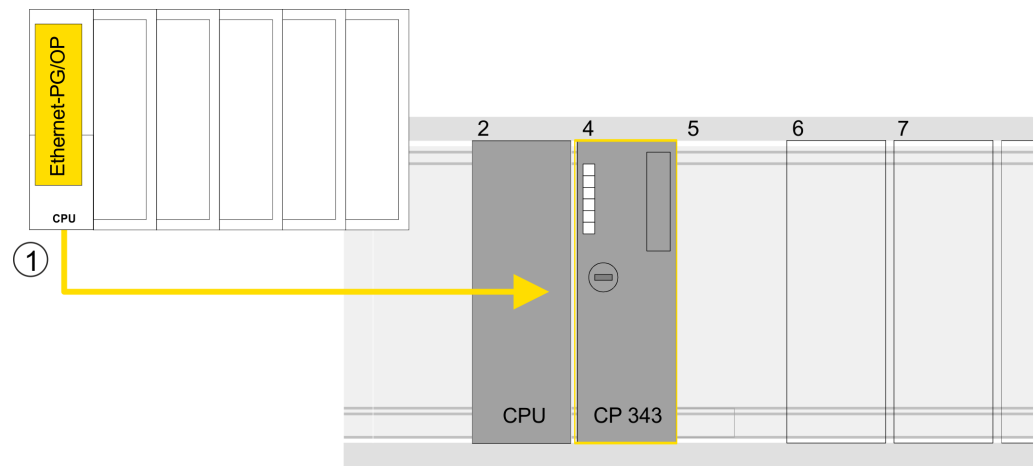
⇒ In der *Geräteübersicht* des PROFINET-IO-Device "VIPA SLIO CPU" ist auf Steckplatz 0 die CPU bereits vorplatziert.

PtP-Funktionalität aktivieren

Bei den System SLIO CPUs 014 ... 017 ist die RS485-Schnittstelle standardmäßig auf PtP-Kommunikation eingestellt. Eine Hardware-Konfiguration zur Einstellung der PtP-Funktionalität ist nicht erforderlich.

Ethernet-PG/OP-Kanal parametrieren

1. ➤ Platzieren Sie für den Ethernet-PG/OP-Kanal auf Steckplatz 4 den Siemens CP 343-1 (6GK7 343-1EX30 0XE0 V3.0).
2. ➤ Öffnen Sie durch Klick auf den CP 343-1EX30 den "Eigenschaften"-Dialog und geben Sie für den CP in den "Eigenschaften" unter "Ethernet-Adresse" IP-Adress-Daten an. Gültige IP-Adress-Parameter erhalten Sie von Ihrem Systemadministrator.



1 Ethernet-PG/OP-Kanal

Geräteübersicht

Baugruppe	...	Steckplatz	...	Typ	...
PLC ...		2		CPU 315-2 PN/DP	
MPI/DP-Schnittstelle		2 X1		MPI/DP-Schnittstelle	
PROFINET-Schnittstelle		2 X2		PROFINET-Schnittstelle	
...		
CP 343-1		4		CP 343-1	
...		

6.6.3 Anwender-Programm

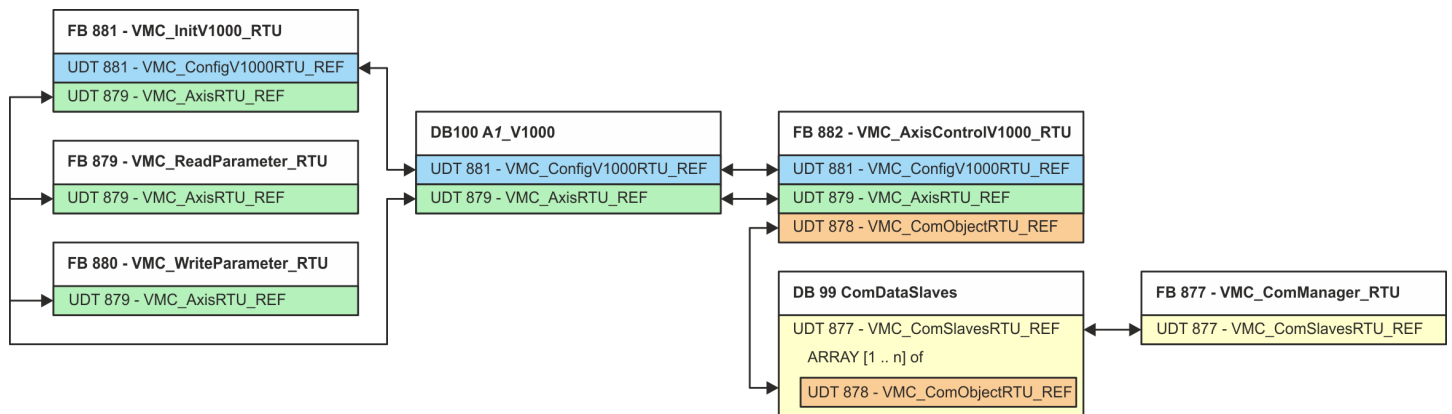
6.6.3.1 Programmstruktur

OB 100

FB 876 - VMC_ConfigMaster_RTU
SFC 216 - SER_CFG

- FB 876 - VMC_ConfigMaster_RTU 303
 - Dieser Baustein dient zur Parametrierung der seriellen Schnittstelle der CPU für Modbus RTU Kommunikation.
 - Intern wird der Baustein SFC 216 - SER_CFG aufgerufen.

OB 1



Mit Ausnahme der Bausteine DB 99 und FB 877 müssen Sie für jeden angebundenen Frequenzumrichter die nachfolgend aufgeführten Bausteine anlegen:

- FB 881 - VMC_InitV1000_RTU ↗ 307
 - Der FB 881 - VMC_InitV1000_RTU initialisiert den entsprechenden Frequenzumrichter mit den Nutzerdaten.
 - Bevor ein Frequenzumrichter angesteuert werden kann, muss dieser initialisiert werden.
 - UDT 881 - VMC_ConfigV1000RTU_REF ↗ 303
 - UDT 879 - VMC_AxisRTU_REF ↗ 303
- FB 879 - VMC_ReadParameter_RTU ↗ 305
 - Mit diesem FB haben Sie lesenden Zugriff auf die Parameter eines seriell über Modbus RTU angebundenen Frequenzumrichters.
 - Die gelesenen Daten werden in einem Datenbaustein erfasst.
 - UDT 879 - VMC_AxisRTU_REF ↗ 303
- FB 880 - VMC_WriteParameter_RTU ↗ 306
 - Mit diesem FB haben Sie schreibenden Zugriff auf die Parameter eines seriell über Modbus RTU angebundenen Frequenzumrichters.
 - Die zu schreibenden Daten sind in einem Datenbaustein abzulegen.
 - UDT 879 - VMC_AxisRTU_REF ↗ 303
- DB 100 - A1_V1000
 - Für jeden Frequenzumrichter, welcher seriell über Modbus RTU angekoppelt ist, ist jeweils ein Datenbaustein anzulegen.
 - UDT 879 - VMC_AxisRTU_REF ↗ 303
 - UDT 881 - VMC_ConfigV1000RTU_REF ↗ 303
- FB 882 - VMC_AxisControlV1000_RTU ↗ 309
 - Mit diesem Baustein können Sie einen seriell über Modbus RTU angebundenen Frequenzumrichter steuern und dessen Status abrufen.
 - UDT 881 - VMC_ConfigV1000RTU_REF ↗ 303
 - UDT 879 - VMC_AxisRTU_REF ↗ 303
 - UDT 878 - VMC_ComObjectRTU_REF ↗ 303

- DB 99 - ComDataSlaves
 - Für die Erfassung der Kommunikationsdaten aller Frequenzumrichter, welche seriell über Modbus RTU angekoppelt sind, ist ein gemeinsamer Datenbaustein anzulegen.
 - UDT 877 - VMC_ComSlavesRTU_REF ↗ 303
 - UDT 878 - VMC_ComObjectRTU_REF ↗ 303
- FB 877 - VMC_ComManager_RTU ↗ 305
 - Der Baustein sorgt dafür, dass immer nur 1 Frequenzumrichter (Modbus-Slave) die serielle Schnittstelle benutzen kann. Bei Einsatz mehrerer Frequenzumrichter sendet dieser Baustein als Kommunikations-Manager die Aufträge an die jeweiligen Modbus-Slaves und wertet deren Antworten aus.
 - UDT 877 - VMC_ComSlavesRTU_REF ↗ 303

6.6.3.2 Bausteine in Projekt kopieren

Bibliothek einbinden

1. ➞ Gehen Sie in den Service-Bereich von www.vipa.com.
2. ➞ Laden Sie aus dem Downloadbereich unter "VIPA Lib" die *Simple Motion Control Library*.
Die Bibliothek liegt für die entsprechende TIA Portal Version als gepackte zip-Datei vor.
3. ➞ Starten Sie mit einem Doppelklick auf die Datei ...TIA_Vxx.zip ihr Unzip-Programm, entpacken Sie Dateien und Ordner in ein Arbeitsverzeichnis für das Siemens TIA Portal.
4. ➞ Wechseln sie im Siemens TIA Portal in die *Projektansicht*.
5. ➞ Wählen Sie auf der rechten Seite die Task-Card "Bibliotheken".
6. ➞ Klicken Sie auf "Globale Bibliothek".
7. ➞ Klicken Sie innerhalb der "*Globalen Bibliothek*" auf die freie Fläche und wählen Sie "*Kontextmenü ➞ Bibliothek dearchivieren*".
8. ➞ Navigieren Sie zu ihrem Arbeitsverzeichnis und laden Sie die Datei ...Simple Motion.zalxx.

Bausteine in Projekt kopieren

- ➞ Kopieren Sie alle Bausteine aus der Bibliothek in das Verzeichnis "*Programmbausteine*" in der *Projektnavigation* Ihres Projekts:
 - FB 876 - VMC_ConfigMaster_RTU
 - FB 877 - VMC_ComManager_RTU
 - FB 878 - VMC_RWParameterSys_RTU
 - FB 879 - VMC_ReadParameter_RTU
 - FB 880 - VMC_WriteParameter_RTU
 - FB 881 - VMC_InitV1000_RTU
 - FB 882 - VMC_AxisControlV1000_RTU
 - FB 60 - SEND
 - FB 61 - RECEIVE
 - FB 72 - RTU MB_MASTER
 - FC 216 - SER_CFG
 - FC 217 - SER_SND
 - FC 218 - SER_RCV
 - UDT 877 - VMC_ComSlavesRTU_REF
 - UDT 878 - VMC_ComObjectRTU_REF
 - UDT 879 - VMC_AxisRTU_REF
 - UDT 881 - VMC_ConfigV1000RTU_REF
 - SFB 4 - TON

6.6.3.3 OB 100 für serielle Kommunikation anlegen

1. ➤ Klicken Sie auf *"Projektnavigation → ...CPU... → Programmbausteine → Neuen Baustein hinzufügen"*.
⇒ Das Dialogfenster *"Neuen Baustein hinzufügen"* öffnet sich.
2. ➤ Geben Sie OB 100 an und bestätigen Sie mit [OK].
⇒ Der OB 100 wird angelegt und geöffnet.
3. ➤ Fügen Sie dem OB 100 einen `Call FB876, DB876` zu.
⇒ Der Bausteinanruf wird angelegt und es öffnet sich ein Dialog zur Angabe des Instanz-Datenbausteins *"VMC_ConfigMaster_RTU_876"*.
4. ➤ Bestätigen Sie die Abfrage des Instanz-Datenbausteins mit [OK].
5. ➤ Geben Sie folgende Parameter vor:

`Call FB876, DB876` ➤ *Kapitel 6.7.5 "FB 876 - VMC_ConfigMaster_RTU - Modbus RTU CPU-Schnittstelle" auf Seite 303*

Baudrate	:= B#16#09	// Baudrate: 09h (9600Bit/s)	IN: BYTE
CharLen	:= B#16#03	// Anzahl Datenbits: 03h (8Bit)	IN: BYTE
Parity	:= B#16#00	// Parität: 0 (none = keine)	IN: BYTE
StopBits	:= B#16#01	// Stopbits: 1 (1Bit)	IN: BYTE
TimeOut	:= W#16#1FFF	// Fehler-Wartezeit: 1FFFh (hoch gewählt)	IN: WORD
Valid	:= "ModbusConfigValid"	// Konfiguration	OUT: BOOL
Error	:= "ModbusConfigError"	// Fehlerrückmeldung	OUT: BOOL
ErrorID	:= "ModbusConfigErrorID"	// Zusätzliche Fehlerinformationen	OUT: WORD

6.6.3.4 Datenbaustein für Modbus-Slave anlegen

Für jeden Frequenzumrichter, welcher seriell über Modbus RTU angekoppelt ist, ist jeweils ein Datenbaustein anzulegen.

1. ➤ Klicken Sie auf *"Projektnavigation → ...CPU... → Programmbausteine → Neuen Baustein hinzufügen"*.
⇒ Das Dialogfenster *"Baustein hinzufügen"* öffnet sich.
2. ➤ Wählen Sie den Bausteintyp *"DB Baustein"* und vergeben Sie diesem den Namen *"A1_V1000"*. Die DB-Nr. können Sie frei wählen wie z.B. DB 100. Geben Sie DB 100 an und legen Sie diesen als globalen DB mit [OK] an.
⇒ Der Baustein wird angelegt und geöffnet.
3. ➤ Legen Sie in *"A1_V1000"* folgende Variablen an:
 - *"AxisData"* vom Typ UDT 879 - `VMC_AxisRTU_REF`
 - *"V1000Data"* vom Typ UDT 881 - `VMC_ConfigV1000RTU_REF`

6.6.3.5 Anzahl der Modbus-Slaves definieren

Die Anzahl der Frequenzumrichter, welche seriell über Modbus RTU angekoppelt sind, können Sie über die UDT 877 - VMC_ComManager_RTU vorgeben.

UDT 877

...	Name	Datentyp	...
...		Struct	
...	Slave	Array[1..1] of ...	
...		END_STRUCT	

1. Öffnen Sie die UDT 877 - VMC_ComManager_RTU unter "Bausteine".
2. Stellen Sie in der Variable "Slave" den "Array [1..1]" Datentyp auf die Anzahl der Frequenzumrichter ein, welche seriell über Modbus RTU angebunden sind.

Beispielsweise ist bei 3 Frequenzumrichter der Datentyp auf "Array [1..3]" zu ändern.

Bitte beachten Sie, dass der Rest unverändert bleibt.

6.6.3.6 Datenbaustein für alle Modbus-Slaves anlegen

Für die Erfassung der Kommunikationsdaten aller Frequenzumrichter, welche seriell über Modbus RTU angekoppelt sind, ist ein gemeinsamer Datenbaustein anzulegen.

1. Klicken Sie auf "Projektnavigation → ...CPU... → Programmbausteine → Neuen Baustein hinzufügen".
⇒ Das Dialogfenster "Baustein hinzufügen" öffnet sich.
2. Wählen Sie den Bausteintyp "DB Baustein" und vergeben Sie diesem den Namen "ComDataSlaves". Die DB-Nr. können Sie frei wählen wie z.B. DB 99. Geben Sie DB 99 an und legen Sie diesen als globalen DB mit [OK] an.
⇒ Der Baustein wird angelegt und geöffnet.
3. Legen Sie in "ComDataSlaves" folgende Variable an:
 - "Slaves" vom Typ UDT 877 - VMC_ComSlavesRTU_REF

6.6.3.7 OB 1 - Instanz des Kommunikations-Managers anlegen

Der FB 877 - VMC_ComManager_RTU sorgt dafür, dass immer nur 1 Frequenzumrichter (Modbus-Slave) die serielle Schnittstelle benutzen kann. Als Kommunikations-Manager sendet der Baustein die Aufträge an die jeweiligen Modbus-Slaves und wertet deren Antworten aus.

1. Öffnen Sie den OB 1.
2. Fügen Sie dem OB 1 einen Call FB877, DB877 hinzu.
⇒ Der Bausteinaufruf wird angelegt und es öffnet sich ein Dialog zur Angabe des Instanz-Datenbausteins "VMC_ComManager_RTU_877".
3. Bestätigen Sie die Abfrage des Instanz-Datenbausteins mit [OK].
4. Geben Sie folgende Parameter vor:

Call FB877, DB877 ↪ Kapitel 6.7.6 "FB 877 - VMC_ComManager_RTU - Modbus RTU Kommunikations-Manager" auf Seite 305

NumberOfSlaves	:= 1	// Anzahl angebundener Frequenzumrichter: 1	IN: INT
WaitCycles	:= "ComWaitCycles"	// Mindestanzahl Wartezyklen	IN: DINT
SlavesComData	:= "ComDataSlaves.Slave"	// Referenz zu allen Kommunikationobjekten	IN-OUT: UDT 877

6.6.3.8 OB 1 - Instanz der V1000-Initialisierung anlegen

Der FB 881 - VMC_InitV1000_RTU initialisiert den entsprechenden Frequenzumrichter mit den Nutzerdaten. Bevor ein Frequenzumrichter angesteuert werden kann, muss dieser initialisiert werden.

- 1.** ➔ Fügen Sie dem OB 1 einen Call FB881, DB881 hinzu.
 ➔ Der Bausteinanruf wird angelegt und es öffnet sich ein Dialog zur Angabe des Instanz-Datenbausteins "VMC_InitV1000_RTU_881".
- 2.** ➔ Bestätigen Sie die Abfrage des Instanz-Datenbausteins mit [OK].
- 3.** ➔ Geben Sie folgende Parameter vor:

Call FB881, DB881 ➔ *Kapitel 6.7.10 "FB 881 - VMC_InitV1000_RTU - Modbus RTU Initialisierung" auf Seite 307*

Execute	:= "A1_InitExecute"	// Mit Flanke 0-1 wird der Auftrag ausgeführt.	IN: BOOL
Hardware	:= "A1_InitHardware"	// Angabe der eingesetzten Hardware // 1: System SLIO CP040, 2: SPEED7 CPU	IN: BYTE
Laddr	:= "A1_InitLaddr"	// Logische Adresse bei Einsatz CP040	IN: INT
UnitId	:= "A1_InitUnitId"	// Modbus-Adresse des V1000	IN: BYTE
UserUnitsVelocity	:= "A1_InitUserUnitsVel"	// Benutzereinheit für Geschwindigkeiten // 0: Hz, 1: %, 2: U/min	IN: INT
UserUnitsAcceleration	:= "A1_InitUserUnitsAcc"	// Benutzereinheit Beschleunigung/Verzögerung // 0: 0,01s, 1: 0,1s	IN: INT
MaxVelocityApp	:= "A1_InitMaxVelocityApp"	// Max. Geschwindigkeit in Benutzereinheiten	IN: REAL
Done	:= "A1_InitDone"	// Status Auftrag fertig	OUT: BOOL
Busy	:= "A1_InitBusy"	// Status Auftrag in Bearbeitung	OUT: BOOL
Error	:= "A1_InitError"	// Fehlerrückmeldung	OUT: BOOL
ErrorID	:= "A1_InitErrorID"	// Zusätzliche Fehlerinformationen	OUT: WORD
Axis	:= "A1_V1000".AxisData	// Referenz zu den allgemeinen Achsdaten	IN-OUT: UDT 879
V1000	:= "A1_V1000".V1000Data	// Referenz zu den antriebsspezifischen Daten	IN-OUT: UDT 881

Eingabewerte

Alle Parameter sind mit den entsprechenden Variablen bzw. Operanden zu verschalten. Folgende Eingabe-Parameter sind entsprechend vorzubelegen:

- **Hardware**
Geben Sie hier die Hardware an, über welche Sie Ihre Frequenzumrichter ansteuern:
 - 1: System SLIO CP040 dessen logische Adresse über *Laddr* vorzugeben ist.
 - 2: SPEED7 CPU
- **Laddr**
 - Logische Adresse für System SLIO CP040 (*Hardware* = 1). Ansonsten wird dieser Parameter ignoriert.
- **UnitId**
 - Modbus-Adresse des V1000.

- **UserUnitsVelocity**
Benutzereinheit für Geschwindigkeiten:
 - 0: Hz
Angabe in Hertz
 - 1: %
Angabe als prozentualer Bezug auf die maximale Geschwindigkeit

$$= 2 \cdot f_{\max} / p$$
 mit f_{\max} : max. Ausgabefrequenz (Parameter E1-04)
 p: Anzahl der Motorpole (motorabhängiger Parameter E2-04, E4-04 oder E5-04)
 - 2: U/min
Angabe in Umdrehungen pro Minute
- **UserUnitsAcceleration**
Benutzereinheiten für die Beschleunigung und Verzögerung
 - 0: 0,01s (Wertebereich: 0,00s - 600,00s)
 - 1: 0,1s (Wertebereich: 0,0 - 6000,0s)
- **MaxVelocityApp**
Max. Geschwindigkeit für die Applikation. Die Angabe hat in Benutzereinheiten zu erfolgen und wird bei Bewegungskommandos für den Abgleich verwendet.

6.6.3.9 OB 1 - Instanz Achskontrolle V1000 anlegen

Mit dem FB 882 - VMC_AxisControlV1000_RTU können Sie einen seriell über Modbus RTU angebundenen Frequenzumrichter steuern und dessen Status abrufen.

1. ➔ Fügen Sie dem OB 1 einen Call FB882, DB882 hinzu.
 ⇒ Der Bausteinanruf wird angelegt und es öffnet sich ein Dialog zur Angabe des Instanz-Datenbausteins "VMC_AxisControlV1000_RTU_882".
2. ➔ Bestätigen Sie die Abfrage des Instanz-Datenbausteins mit [OK].
3. ➔ Geben Sie folgende Parameter vor:







Call FB882, DB882 ↗ *Kapitel 6.7.11 "FB 882 - VMC_AxisControlV1000_RTU - Modbus RTU Achskontrolle" auf Seite 309*

AxisEnable	:= "A1_AxisEnable"	// Aktivierung der Achse	IN: BOOL
AxisReset	:= "A1_AxisReset"	// Kommando: Fehler des V1000 zurücksetzen.	IN: BOOL
StopExecute	:= "A1_StopExecute"	// Kommando: Stop - Achse stoppen	IN: BOOL
MvVelocityExecute	:= "A1_MvVelocityExecute"	// Kommando: MoveVelocity (Geschwindigkeitsregelung)	IN: BOOL
Velocity	:= "A1_Velocity"	// Parameter: Geschwindigkeitsvorgabe für MoveVelocity	IN: REAL
AccelerationTime	:= "A1_AccelerationTime"	// Parameter: Beschleunigungszeit	IN: REAL
DecelerationTime	:= "A1_DecelerationTime"	// Parameter: Verzögerungszeit	IN: REAL
JogPositive	:= "A1_JogPositive"	// Kommando: JogPos	IN: BOOL
JogNegative	:= "A1_JogNegative"	// Kommando: JogNeg	IN: BOOL
JogVelocity	:= "A1_JogVelocity"	// Parameter: Geschwindigkeitsvorgabe für Jogging	IN: REAL
JogAccelerationTime	:= "A1_JogAccelerationTime"	// Parameter: Beschleunigungszeit für Jogging	IN: REAL
JogDecelerationTime	:= "A1_JogDecelerationTime"	// Parameter: Verzögerungszeit für Jogging	IN: REAL
AxisReady	:= "A1_AxisReady"	// Status: Bereitschaft der Achse	OUT: BOOL
AxisEnabled	:= "A1_AxisEnabled"	// Status: Aktivierung der Achse	OUT: BOOL
AxisError	:= "A1_AxisError"	// Status: Achsfehler	OUT: BOOL

AxisErrorID	:= "A1_AxisErrorID"	// Status: Zusätzliche Fehlerinformationen für <i>AxisError</i>	OUT: WORD
DriveError	:= "A1_DriveError"	// Status: Fehler Frequenzumrichter	OUT: BOOL
ActualVelocity	:= "A1_ActualVelocity"	// Status: Aktuelle Geschwindigkeit	OUT: REAL
InVelocity	:= "A1_InVelocity"	// Status: Zielgeschwindigkeit	OUT: BOOL
CmdDone	:= "A1_CmdDone"	// Status: Kommando fertig	OUT: BOOL
CmdBusy	:= "A1_CmdBusy"	// Status: Kommando in Bearbeitung	OUT: BOOL
CmdAborted	:= "A1_CmdAborted"	// Status: Kommando abgebrochen	OUT: BOOL
CmdError	:= "A1_CmdError"	// Status: Kommando Fehler	OUT: BOOL
CmdErrorID	:= "A1_CmdErrorID"	// Status: Zusätzliche Fehlerinformationen für <i>CmdError</i>	OUT: WORD
CmdActive	:= "A1_CmdActive"	// Status: Aktives Kommando	OUT: INT
DirectionPositive	:= "A1_DirectionPositive"	// Status: Drehrichtung positiv	OUT: BOOL
DirectionNegative	:= "A1_DirectionNegative"	// Status: Drehrichtung negativ	OUT: BOOL
Axis	:= "A1_V1000".AxisData	// Referenz zu den allgemeinen Achsdaten	IN-OUT: UDT 879
V1000	:= "A1_V1000".V1000Data	// Referenz zu den allgemeinen Daten des // Frequenzumrichters	IN-OUT: UDT 881
AxisComData	:= "ComDataSlaves".Slaves.Slave(1)	// Referenz zu den Kommunikationsdaten	IN-OUT: UDT 878

6.6.3.10 OB 1 - Instanz Parameter lesen anlegen

Mit dem FB 879 - VMC_ReadParameter_RTU haben Sie lesenden Zugriff auf die Parameter eines seriell über Modbus RTU angeordneten Frequenzumrichters. Für die Erfassung der Parameterdaten ist ein DB anzulegen.

1.  Klicken Sie auf "*Projektnavigation* → ...CPU... → *Programmbausteine* → *Neuen Baustein hinzufügen*".
⇒ Das Dialogfenster "*Baustein hinzufügen*" öffnet sich.
2.  Wählen Sie den Bausteintyp "*DB Baustein*" und vergeben Sie diesem den Namen "A1_TransferData". Die DB-Nr. können Sie frei wählen. Geben Sie DB 98 an und legen Sie diesen als globalen DB mit [OK] an.
⇒ Der Baustein wird angelegt und geöffnet.
3.  Legen Sie in "A1_TransferData" folgende Variablen an:
 - "*Data_0*" vom Typ WORD
 - "*Data_1*" vom Typ WORD
 - "*Data_2*" vom Typ WORD
 - "*Data_3*" vom Typ WORD
4.  Fügen Sie dem OB 1 einen Call FB879, DB879 hinzu.
⇒ Der Bausteinaufruf wird angelegt und es öffnet sich ein Dialog zur Angabe des Instanz-Datenbausteins "*VMC_ReadParameter_RTU*".
5.  Bestätigen Sie die Abfrage des Instanz-Datenbausteins mit [OK].
6.  Geben Sie folgende Parameter vor:

Call FB879, DB879  Kapitel 6.7.8 "*FB 879 - VMC_ReadParameter_RTU - Modbus RTU Parameter lesen*" auf Seite 305

Execute	:= "A1_RdParExecute"	// Mit Flanke 0-1 wird der Auftrag ausgeführt.	IN: BOOL
StartAddress	:= "A1_RdParStartAddress"	// Start-Adresse des 1. Registers	IN: INT

Quantity	:= "A1_RdParQuantity"	// Anzahl der zu lesenden Register	IN: INT
Done	:= "A1_RdParDone"	// Status Auftrag fertig	IN: REAL
Busy	:= "A1_RdParBusy"	// Status Auftrag in Bearbeitung	OUT: BOOL
Error	:= "A1_RdParError"	// Fehlerrückmeldung	OUT: BOOL
ErrorID	:= "A1_RdParErrorID"	// Zusätzliche Fehlerinformationen	OUT: BOOL
Data	:= P#DB98.DBX0.0 BYTES 8	// Ablageort der Parameterdaten	OUT: WORD
Axis	:= "A1_V1000".AxisData	// Referenz zu den allgemeinen Achsdaten	IN-OUT: UDT 879



Bitte beachten Sie, dass immer nur ganze Register als WORD gelesen werden können. Zur Auswertung einzelner Bits müssen Sie High- und Low-Byte vertauschen!

6.6.3.11 OB 1 - Instanz Parameter schreiben anlegen

Mit dem FB 880 - VMC_WriteParameter_RTU haben Sie schreibenden Zugriff auf die Parameter eines seriell über Modbus RTU angebundenen Frequenzumrichters. Für die Erfassung können Sie den für Lesezugriff angelegten DB verwenden - hier DB 98.

- 1.** ➞ Fügen Sie dem OB 1 einen Call FB880, DB880 hinzu.
 ➞ Der Bausteinanruf wird angelegt und es öffnet sich ein Dialog zur Angabe des Instanz-Datenbausteins "VMC_WriteParameter_RTU".
- 2.** ➞ Bestätigen Sie die Abfrage des Instanz-Datenbausteins mit [OK].
- 3.** ➞ Geben Sie folgende Parameter vor:

Call FB880, DB880 ➞ *Kapitel 6.7.9 "FB 880 - VMC_WriteParameter_RTU - Modbus RTU Parameter schreiben" auf Seite 306*

Execute	:= "A1_WrParExecute"	// Mit Flanke 0-1 wird der Auftrag ausgeführt.	IN: BOOL
StartAddress	:= "A1_WrParStartAddress"	// Start-Adresse des 1. Registers	IN: INT
Quantity	:= "A1_WrParQuantity"	// Anzahl der zu schreibenden Register	IN: INT
Done	:= "A1_WrParDone"	// Status Auftrag fertig	IN: REAL
Busy	:= "A1_WrParBusy"	// Status Auftrag in Bearbeitung	OUT: BOOL
Error	:= "A1_WrParError"	// Fehlerrückmeldung	OUT: BOOL
ErrorID	:= "A1_WrParErrorID"	// Zusätzliche Fehlerinformationen	OUT: BOOL
Data	:= P#DB98.DBX0.0 BYTES 8	// Ablageort der Parameterdaten	OUT: WORD
Axis	:= "A1_V1000".AxisData	// Referenz zu den allgemeinen Achsdaten	IN-OUT: UDT 879

6.6.3.12 Zeitlicher Ablauf

1. ➤ Speichern und übersetzen Sie Ihr Projekt.
2. ➤ Transferieren Sie Ihr Projekt in Ihre CPU.
 - ⇒ Sie können jetzt Ihre Applikation in Betrieb nehmen.

**VORSICHT!**

Bitte beachten Sie immer die Sicherheitshinweise zu ihrem Frequenzumrichter, insbesondere bei der Inbetriebnahme!

3. ➤ Mittels einer Beobachtungstabelle können Sie den Frequenzumrichter manuell steuern. Zum Anlegen einer Beobachtungstabelle doppelklicken Sie auf "Projektnavigation ➔ ...CPU... ➔ Beobachtungs- und Forcetabellen ➔ Neue Beobachtungstabelle hinzufügen".
 - ⇒ Die Beobachtungstabelle wird angelegt und für die Bearbeitung geöffnet.
4. ➤ Passen Sie zuerst die Wartezeit zwischen 2 Aufträgen an. Diese beträgt für einen V1000-Frequenzumrichter mindestens 200ms. Stellen Sie hierzu in der Beobachtungstabelle unter "Name" den Bezeichner "ComWaitCycles" als "DEZ" ein und geben Sie unter "Steuerwert" einen Wert zwischen 200 und 400 vor.



Zur Performance-Steigerung können Sie diesen Wert später nach unten korrigieren, solange Sie keinen Timeout-Fehler (80C8h) erhalten. Bitte beachten Sie hierbei, dass manche Befehle, wie z.B. MoveVelocity aus mehreren Aufträgen bestehen können.

5. ➤ Bevor Sie einen Frequenzumrichter ansteuern können, muss dieser mit dem FB 881 - VMC_InitV1000_RTU initialisiert werden. ➔ Kapitel 6.7.10 "FB 881 - VMC_InitV1000_RTU - Modbus RTU Initialisierung" auf Seite 307
 Stellen Sie hierzu in der Beobachtungstabelle unter "Name" den Bezeichner "A1_InitExecute" als "BOOL" ein und geben Sie unter "Steuerwert" den Wert "True" vor. Aktivieren Sie das Steuern der Variable und starten Sie die Übertragung der Steuerwerte.
 - ⇒ Der Frequenzumrichter wird initialisiert. Nach Abarbeitung liefert der Ausgang Done TRUE zurück. Im Fehlerfall können Sie durch Auswertung der ErrorID den Fehler ermitteln.



Fahren Sie erst fort, wenn der Init-Baustein keinen Fehler meldet!

6. ➤ Bei erfolgreicher Initialisierung erfolgt zyklisch die Abarbeitung der Register der angebundenen Frequenzumrichter d.h. diese erhalten zyklisch Aufträge. Zur manuellen Steuerung können Sie mit dem FB 882 - VMC_AxisControlV1000_RTU Steuerbefehle an den entsprechenden Frequenzumrichter senden. ➔ Kapitel 6.7.11 "FB 882 - VMC_AxisControlV1000_RTU - Modbus RTU Achskontrolle" auf Seite 309
7. ➤ Legen Sie die Parameter des FB 882 - VMC_AxisControlV1000_RTU zur Steuerung und Abfrage in der Beobachtungstabelle an.
8. ➤ Speichern Sie die Beobachtungstabelle unter einem Namen wie z.B. "V1000".
9. ➤ Aktivieren Sie durch Setzen von AxisEnable die entsprechende Achse. Sobald diese AxisReady = TRUE zurückmeldet, können Sie diese mit den entsprechenden Fahrbefehlen ansteuern.

6.7 Antriebsspezifische Bausteine

6.7.1 UDT 877 - VMC_ComSlavesRTU_REF - Modbus RTU Datenstruktur Kommunikationsdaten aller Slaves

Dies ist eine benutzerdefinierte Datenstruktur, für die Kommunikationsdaten der angebundenen Modbus RTU Slaves. Die UDT ist speziell angepasst an die Verwendung von Frequenzumrichter, welche über Modbus RTU angebunden sind.

6.7.2 UDT 878 - VMC_ComObjectRTU_REF - Modbus RTU Datenstruktur Kommunikationsdaten Slave

Dies ist eine benutzerdefinierte Datenstruktur, für die Kommunikationsdaten eines angebundenen Modbus RTU Slaves. Die UDT ist speziell angepasst an die Verwendung von Frequenzumrichter, welche über Modbus RTU angebunden sind.

6.7.3 UDT 879 - VMC_AxisRTU_REF - Modbus RTU Datenstruktur Achsdaten

Dies ist eine benutzerdefinierte Datenstruktur, die Statusinformationen des Frequenzumrichters beinhaltet. Diese Struktur dient als Referenz zu den allgemeinen Achsdaten des Frequenzumrichters.

6.7.4 UDT 881 - VMC_ConfigV1000RTU_REF - Modbus RTU Datenstruktur Konfiguration

Dies ist eine benutzerdefinierte Datenstruktur, die Informationen zu den Konfigurationsdaten eines Frequenzumrichters beinhaltet, welcher über Modbus RTU angebunden ist.

6.7.5 FB 876 - VMC_ConfigMaster_RTU - Modbus RTU CPU-Schnittstelle

Beschreibung

Dieser Baustein dient zur Parametrierung der seriellen Schnittstelle der CPU für Modbus RTU Kommunikation.



Bitte beachten Sie, dass dieser Baustein intern den SFC 216 aufruft.

Im SPEED7 Studio wird dieser Baustein automatisch in Ihr Projekt eingefügt.

Im Siemens SIMATIC Manager müssen Sie den SFC 216 aus der Motion Control Library in Ihr Projekt kopieren.

Parameter

Parameter	Deklaration	Datentyp	Beschreibung
Baudrate	IN	BYTE	Geschwindigkeit der Datenübertragung in Bit/s (Baud).
			<div> <ul style="list-style-type: none"> ■ 04h: 1200Baud ■ 05h: 1800Baud ■ 06h: 2400Baud ■ 07h: 4800Baud ■ 08h: 7200Baud ■ 09h: 9600Baud </div> <div> <ul style="list-style-type: none"> ■ 0Ah: 14400Baud ■ 0Bh: 19200Baud ■ 0Ch: 38400Baud ■ 0Dh: 57600Baud ■ 0Eh: 115200Baud </div>

Parameter	Deklaration	Datentyp	Beschreibung
CharLen	IN	BYTE	Anzahl der Datenbits, auf die ein Zeichen abgebildet wird <ul style="list-style-type: none"> ■ 0: 5Bit ■ 1: 6Bit ■ 2: 7Bit ■ 3: 8Bit
Parity	IN	BYTE	Die Parität ist je nach Wert gerade oder ungerade. Zur Paritätskontrolle werden die Informationsbits um das Paritätsbit erweitert, das durch seinen Wert ("0" oder "1") den Wert aller Bits auf einen vereinbarten Zustand ergänzt. Ist keine Parität vereinbart, wird das Paritätsbit auf "1" gesetzt, aber nicht ausgewertet. <ul style="list-style-type: none"> ■ 0: None (keine) ■ 1: Odd (ungerade) ■ 2: Even (gerade)
StopBits	IN	BYTE	Die Stopbits werden jedem zu übertragenden Zeichen nachgesetzt und kennzeichnen das Ende eines Zeichens <ul style="list-style-type: none"> ■ 1: 1Bit ■ 2: 1.5Bit ■ 3: 2Bit
TimeOut	IN	WORD	Wartezeit, bis ein Fehler generiert wird, wenn ein Slave nicht antwortet. Die Zeitangabe für <i>TimeOut</i> ist als hexadezimaler Wert anzugeben. Den hexadezimalen Wert erhalten Sie, indem Sie die gewünschte Zeit in Sekunden mit der Baudrate multiplizieren. Beispiel: Gewünschte Zeit 8ms bei einer Baudrate von 19200Bit/s Berechnung: $19200\text{Bit/s} \times 0,008\text{s} \approx 154\text{Bit} >>>> (9Ah)$ Als Hex-Wert ist 9Ah vorzugeben.
Valid	OUT	BOOL	Konfiguration <ul style="list-style-type: none"> ■ TRUE: Die Konfiguration ist gültig. ■ FALSE: Die Konfiguration ist nicht gültig.
Error	OUT	BOOL	Fehlerrückmeldung <ul style="list-style-type: none"> ■ TRUE: Es ist ein Fehler aufgetreten - siehe <i>ErrorID</i>. ■ FALSE: Es liegt kein Fehler vor.
ErrorID	OUTPUT	WORD	Zusätzliche Fehlerinformationen 🔗 <i>Kapitel 9 "ErrorID - Zusätzliche Fehlerinformationen" auf Seite 340</i>

6.7.6 FB 877 - VMC_ComManager_RTU - Modbus RTU Kommunikations-Manager

Beschreibung

Dieser Baustein regelt, dass nacheinander immer nur ein Slave über die serielle Schnittstelle kommunizieren kann. Über die UDT 877 hat dieser Baustein Zugriff auf die Kommunikationsdaten aller Slaves.



Pro serielle Schnittstelle dürfen Sie immer nur einen FB 877 in Ihrem Projekt verwenden!

Parameter

Parameter	Deklaration	Datentyp	Beschreibung
NumberOfSlaves	IN	INT	Anzahl der aktuell verwendeten Modbus Slaves
WaitCycles	IN	DINT	Mindestanzahl an Zyklen, die zwischen zwei Anfragen eines Slaves gewartet werden soll. Hiermit lassen sich Überläufe am Slave und hieraus resultierende Timeouts verhindern.
SlavesComData	IN-OUT	UDT 877	Referenz zum Datenbaustein mit allen Kommunikationobjekten

6.7.7 FB 878 - VMC_RWParameterSys_RTU - Modbus RTU Parameter System lesen/schreiben

Beschreibung

Dieser Baustein wird intern vom System für die Parameterübertragung verwendet.



Diesen Baustein dürfen Sie nicht aufrufen, da dies zu einem Fehlverhalten Ihres Systems führen kann!

6.7.8 FB 879 - VMC_ReadParameter_RTU - Modbus RTU Parameter lesen

Beschreibung

Mit diesem Baustein können Sie Parameter vom entsprechenden Slave lesen.



Bitte beachten Sie, dass immer nur ganze Register als WORD gelesen werden können. Zur Auswertung einzelner Bits müssen Sie High- und Low-Byte vertauschen!

Parameter

Parameter	Deklaration	Datentyp	Beschreibung
Execute	IN	BOOL	Mit Flanke 0-1 wird der Auftrag ausgeführt.
StartAddress	IN	WORD	Start-Adresse des Registers, ab dem gelesen werden soll.
Quantity	IN	BYTE	Anzahl der Register, die gelesen werden sollen.
Done	OUT	BOOL	Status ■ TRUE: Auftrag erfolgreich durchgeführt

Parameter	Deklaration	Datentyp	Beschreibung
Busy	OUT	BOOL	Status ■ TRUE: Auftrag ist in Bearbeitung
Error	OUT	BOOL	Status ■ TRUE: Ein Fehler ist aufgetreten. Zusätzliche Fehlerinformationen können dem Parameter <i>ErrorID</i> entnommen werden.
ErrorID	OUT	WORD	Zusätzliche Fehlerinformationen 🔗 Kapitel 9 "ErrorID - Zusätzliche Fehlerinformationen" auf Seite 340
Data	IN-OUT	ANY	Referenz wohin die gelesenen Daten gespeichert werden sollen
Axis	IN-OUT	UDT 879	Referenz zu den allgemeinen Achsdaten des Frequenzumrichters

6.7.9 FB 880 - VMC_WriteParameter_RTU - Modbus RTU Parameter schreiben

Beschreibung

Mit diesem Baustein können Sie Parameter in die Register des entsprechenden Slave schreiben.



Bitte beachten Sie, dass immer nur ganze Register als WORD geschrieben werden können. Zum Setzen bzw. Rücksetzen einzelner Bits müssen Sie High- und Low-Byte vertauschen!

Parameter

Parameter	Deklaration	Datentyp	Beschreibung
Execute	IN	BOOL	Mit Flanke 0-1 wird der Auftrag ausgeführt.
StartAddress	IN	WORD	Start-Adresse des Registers ab dem geschrieben werden soll.
Quantity	IN	BYTE	Anzahl der Register, die geschrieben werden sollen.
Done	OUT	BOOL	Status ■ TRUE: Auftrag erfolgreich durchgeführt
Busy	OUT	BOOL	Status ■ TRUE: Auftrag ist in Bearbeitung
Error	OUT	BOOL	Status ■ TRUE: Ein Fehler ist aufgetreten. Zusätzliche Fehlerinformationen können dem Parameter <i>ErrorID</i> entnommen werden.
ErrorID	OUT	WORD	Zusätzliche Fehlerinformationen 🔗 Kapitel 9 "ErrorID - Zusätzliche Fehlerinformationen" auf Seite 340

Parameter	Deklaration	Datentyp	Beschreibung
Data	IN-OUT	ANY	Referenz zu den Daten, die geschrieben werden sollen.
Axis	IN-OUT	UDT 879	Referenz zu den allgemeinen Achsdaten des Frequenzumrichters

6.7.10 FB 881 - VMC_InitV1000_RTU - Modbus RTU Initialisierung

Beschreibung

Dieser Baustein dient zur Initialisierung eines Frequenzumrichters mit den entsprechenden Nutzerdaten und muss vor der Übergabe von Kommandos abgearbeitet sein. Der Baustein ist speziell angepasst an die Verwendung eines Frequenzumrichters, welcher über Modbus RTU angebunden ist.

Parameter

Parameter	Deklaration	Datentyp	Beschreibung
Execute	IN	BOOL	Mit Flanke 0-1 wird der Auftrag ausgeführt.
Hardware	IN	BYTE	Angabe der eingesetzten Hardware <ul style="list-style-type: none"> ■ 1: System SLIO CP040 dessen logische Adresse über <i>Laddr</i> vorzugeben ist. ■ 2: SPEED7 CPU
Laddr	IN	INT	Logische Adresse für System SLIO CP040 (<i>Hardware</i> = 1). Ansonsten wird dieser Parameter ignoriert.
UnitId	IN	BYTE	Modbus-Adresse des <i>V1000</i> .
UserUnitsVelocity	IN	INT	Benutzereinheit für Geschwindigkeiten <ul style="list-style-type: none"> ■ 0: Hz <ul style="list-style-type: none"> – Angabe in Hertz ■ 1: % <ul style="list-style-type: none"> – Angabe als prozentualer Bezug auf die maximale Geschwindigkeit – $= 2 \cdot f_{\max} / p$ mit f_{\max}: max. Ausgabefrequenz (Parameter E1-04) p: Anzahl der Motorpole (motorabhängiger Parameter E2-04, E4-04 oder E5-04) ■ 2: U/min <ul style="list-style-type: none"> – Angabe in Umdrehungen pro Minute
UserUnitsAcceleration	IN	INT	Benutzereinheiten für die Beschleunigung und Verzögerung <ul style="list-style-type: none"> ■ 0: 0,01s (Wertebereich: 0,00s - 600,00s) ■ 1: 0,1s (Wertebereich: 0,0 - 6000,0s)
MaxVelocityApp	IN	REAL	Max. Geschwindigkeit für die Applikation. Die Angabe hat in Benutzereinheiten zu erfolgen und wird bei Bewegungskommandos für den Abgleich verwendet.
Done	OUT	BOOL	Status <ul style="list-style-type: none"> ■ TRUE: Auftrag erfolgreich durchgeführt
Busy	OUT	BOOL	Status <ul style="list-style-type: none"> ■ TRUE: Auftrag ist in Bearbeitung

Parameter	Deklaration	Datentyp	Beschreibung
Error	OUT	BOOL	Status <ul style="list-style-type: none">■ TRUE: Ein Fehler ist aufgetreten. Zusätzliche Fehlerinformationen können dem Parameter <i>ErrorID</i> entnommen werden.
ErrorID	OUT	WORD	Zusätzliche Fehlerinformationen ↳ Kapitel 9 " <i>ErrorID - Zusätzliche Fehlerinformationen</i> " auf Seite 340
Axis	IN-OUT	UDT 879	Referenz zu den allgemeinen Achsdaten des Frequenzumrichters
V1000	IN-OUT	UDT 881	Referenz zu den Nutzerdaten des Frequenzumrichters

6.7.11 FB 882 - VMC_AxisControlV1000_RTU - Modbus RTU Achskontrolle

Beschreibung

Mit dem FB 882 *VMC_AxisControlV1000_RTU* können Sie einen über Modbus RTU angebundenen Frequenzumrichter steuern und dessen Status abrufen.



Die Ansteuerung eines V1000-Frequenzumrichters, welcher über Modbus RTU angebunden ist, erfolgt ausschließlich mit dem FB 882 VMC_AxisControlV1000_RTU. PLCopen-Bausteine werden nicht unterstützt!

Parameter

Parameter	Deklaration	Datentyp	Beschreibung
AxisEnable	IN	BOOL	Aktivierung der Achse <ul style="list-style-type: none"> ■ TRUE: Achse einschalten → <i>AxisEnabled</i> = 1, Kommandos können ausgeführt werden ■ FALSE: Achse ausschalten → <i>AxisEnabled</i> = 0, es können keine Kommandos ausgeführt werden.
AxisReset	IN	BOOL	Kommando: Fehler des Frequenzumrichters zurücksetzen. → <i>CmdActive</i> = 1
StopExecute	IN	BOOL	Kommando: <i>Stop</i> - Achse stoppen → <i>CmdActive</i> = 1
MvVelocityExecute	IN	BOOL	Kommando: <i>MoveVelocity</i> (Geschwindigkeitsregelung) ausführen → <i>CmdActive</i> = 2
Velocity	IN	REAL	Parameter: Geschwindigkeitsvorgabe für <i>MoveVelocity</i> in Benutzereinheiten. Siehe Beispiel nach Tabelle
AccelerationTime	IN	REAL	Parameter: Beschleunigungszeit in Sekunden (Genauigkeit je nach <i>UserUnitsAcceleration</i> am Init Baustein). Ist immer bezogen auf die Zeit, vom Stillstand auf die maximal eingestellte Geschwindigkeit. Siehe Beispiel nach Tabelle
DecelerationTime	IN	REAL	Parameter: Verzögerungszeit in Sekunden (Genauigkeit je nach <i>UserUnitsAcceleration</i> am Init Baustein). Ist immer bezogen auf die Zeit, vom Stillstand auf die maximal eingestellte Geschwindigkeit. Siehe Beispiel unten.
JogPositive	IN	BOOL	Kommando: <i>JogPos</i> <ul style="list-style-type: none"> ■ Flanke 0-1: Achse starten in positive Richtung (Jogging positiv) ■ Flanke 1-0: Achse stoppen
JogNegative	IN	BOOL	Kommando: <i>JogNeg</i> <ul style="list-style-type: none"> ■ Flanke 0-1: Achse starten in negative Richtung (Jogging negativ) ■ Flanke 1-0: Achse stoppen
JogVelocity	IN	REAL	Parameter: Geschwindigkeitsvorgabe für Jogging in Benutzereinheiten. Siehe Beispiel unten.
JogAccelerationTime	IN	REAL	Parameter: Beschleunigungszeit für Jogging in Sekunden (Genauigkeit je nach <i>UserUnitsAcceleration</i> am Init-Baustein). Ist immer bezogen auf die Zeit, vom Stillstand auf die maximal eingestellte Geschwindigkeit. Siehe Beispiel nach Tabelle

Parameter	Deklaration	Datentyp	Beschreibung
JogDeceleration-Time	IN	REAL	Parameter: Verzögerungszeit für Jogging in Sekunden (Genauigkeit je nach <i>UserUnitsAcceleration</i> von FB 881). Parameter bezieht sich immer auf die Zeit vom Stillstand auf die maximal eingestellte Geschwindigkeit. Siehe Beispiel nach Tabelle
AxisReady	OUT	BOOL	Status: Bereitschaft der Achse <ul style="list-style-type: none"> ■ TRUE: Achse ist einschaltbereit ■ FALSE: Achse ist nicht einschaltbereit
AxisEnabled	OUT	BOOL	Status: Aktivierung der Achse <ul style="list-style-type: none"> ■ TRUE: Achse ist eingeschaltet ■ FALSE: Achse ist ausgeschaltet
AxisError	OUT	BOOL	Status: Achsfehler <ul style="list-style-type: none"> ■ TRUE: Achse meldet einen Fehler und wird gesperrt. Nähere Fehlerinformationen befinden sich in <i>AxisErrorID</i>. ■ FALSE: Achse meldet keine Fehler.
AxisErrorID	OUT	WORD	Status: Zusätzliche Fehlerinformationen für <i>AxisError</i> 🔗 Kapitel 9 "ErrorID - Zusätzliche Fehlerinformationen" auf Seite 340
DriveError	OUT	BOOL	Status: Fehler Frequenzumrichter <ul style="list-style-type: none"> ■ TRUE: Frequenzumrichter meldet einen Fehler und wird gesperrt. ■ FALSE: Frequenzumrichter meldet keine Fehler.
ActualVelocity	OUT	REAL	Status: Aktuelle Geschwindigkeit in Benutzereinheiten
InVelocity	OUT	BOOL	Status Zielgeschwindigkeit <ul style="list-style-type: none"> ■ TRUE: Die Zielgeschwindigkeit <i>Velocity</i> wurde erreicht. ■ FALSE: Die Zielgeschwindigkeit <i>Velocity</i> wurde noch nicht erreicht.
CmdDone	OUT	BOOL	Status: Kommando fertig <ul style="list-style-type: none"> ■ TRUE: Kommando wurde erfolgreich ausgeführt. ■ FALSE: Kommando wurde noch nicht ausgeführt bzw. befindet sich noch in der Bearbeitung.
CmdBusy	OUT	BOOL	Status: Kommando in Bearbeitung <ul style="list-style-type: none"> ■ TRUE: Kommando befindet sich in der Bearbeitung ■ FALSE: Aktuell wird kein Kommando ausgeführt.
CmdAborted	OUT	BOOL	Status: Kommando abgebrochen <ul style="list-style-type: none"> ■ TRUE: Kommando wurde abgebrochen. ■ FALSE: Kommando wurde nicht abgebrochen
CmdError	OUT	BOOL	Status: Kommando Fehler <ul style="list-style-type: none"> ■ TRUE: Bei der Ausführung eines Kommandos ist ein Fehler aufgetreten ■ FALSE: Die Ausführung eines Kommandos verlief fehlerfrei.
CmdErrorID	OUT	WORD	Status: Zusätzliche Fehlerinformationen für <i>CmdError</i> 🔗 Kapitel 9 "ErrorID - Zusätzliche Fehlerinformationen" auf Seite 340

Parameter	Deklaration	Datentyp	Beschreibung
CmdActive	OUT	INT	Status: Aktives Kommando <ul style="list-style-type: none"> ■ 0: NoCmd - kein Kommando aktiv ■ 1: Stop ■ 2: MvVelocity ■ 3: MvRelative ■ 4: JogPos ■ 5: JogNeg
DirectionPositive	OUT	BOOL	Status: Drehrichtung positiv <ul style="list-style-type: none"> ■ TRUE: Aktuelle Drehrichtung ist positiv ■ FALSE: Aktuelle Drehrichtung ist nicht positiv
DirectionNegative	OUT	BOOL	Status: Drehrichtung negativ <ul style="list-style-type: none"> ■ TRUE: Aktuelle Drehrichtung ist negative ■ FALSE: Aktuelle Drehrichtung ist nicht negative
Axis	IN-OUT	UDT 879	Referenz zu den allgemeinen Achsdaten des Frequenzumrichters
V1000	IN-OUT	UDT 881	Referenz zu den Nutzerdaten des Frequenzumrichters
AxisComData	IN-OUT	UDT 878	Referenz zu den Kommunikationsdaten des aktuellen Slave

Beispiel AccelerationTime

Die Werte für *Velocity*, *AccelerationTime* und *DecelerationTime* sind in den unter FB 881 - VMC_InitV1000_RTU eingestellten Benutzereinheiten vorzugeben. *AccelerationTime* bzw. *DecelerationTime* beziehen sich immer auf die Zeit vom Stillstand bis zur maximal eingestellten Geschwindigkeit bzw. von der maximalen Geschwindigkeit bis zum Stillstand.

Die maximal Geschwindigkeit ergibt sich über die Formel

$$v_{max} = \frac{2 \cdot f}{p}$$

v_{max} max. Geschwindigkeit in 1/s

f max. Ausgabefrequenz (Parameter E1-04)

p Anzahl der Motorpole (motorabhängiger Parameter E2-04, E4-04 oder E5-04)

Zeitlicher Ablauf

1. ➤ Wählen Sie "Projekt ➔ Alles übersetzen" und übertragen Sie das Projekt in Ihre CPU. Näheres zur Übertragung Ihres Projekt finden Sie in der Onlinehilfe zum *SPEED7 Studio*.
 ➔ Sie können jetzt Ihre Applikation in Betrieb nehmen.

**VORSICHT!**

Bitte beachten Sie immer die Sicherheitshinweise zu ihrem Frequenzumrichter, insbesondere bei der Inbetriebnahme!

2. ➤ Bringen Sie Ihre CPU in RUN und schalten Sie Ihren Frequenzumrichter ein.
 ➔ Der FB 882 - VMC_AxisControlV1000_RTU wird zyklisch abgearbeitet.
3. ➤ Sobald *AxisReady* = TRUE meldet, können Sie mit *AxisEnable* die Achse freigeben.

4. ➡ Sie haben jetzt die Möglichkeit über die entsprechenden Parameter Ihre Achse zu steuern und deren Status abzufragen.

7 Antrieb über HMI steuern

Übersicht

Die Antriebssteuerung über ein HMI ist bei folgende Bibliotheks-Gruppen möglich:

- *Sigma-5 EtherCAT* ↗ 9
- *Sigma-7S EtherCAT* ↗ 47
- *Sigma-7W EtherCAT* ↗ 86
- *Sigma-5/7 Pulse Train* ↗ 201

Zur Steuerung des entsprechenden Antriebs über ein HMI wie Touch Panel bzw. Panel-PC gibt es für Movicon eine Symbolbibliothek. Hiermit können Sie mit Hilfe von Templates den entsprechenden VMC_AxisControl-Funktionsbaustein ansteuern. Die Symbolbibliothek bietet Ihnen folgende Templates:

- Numeric Touchpad
 - Dies ist ein an die VMC_AxisControl-Templates angepasstes Eingabefeld für verschiedene Displayauflösungen.
 - Das Touchpad können Sie anstelle des Default-Eingabefeldes verwenden.
- VMC_AxisControl
 - Template zur Ansteuerung des FB 860 - VMC_AxisControl-Funktionsbausteins in der CPU.
 - Das Template gibt es für verschiedene Displayauflösungen.
- VMC_AxisControl ... Trend
 - Template zur Ansteuerung des FB 860 - VMC_AxisControl-Funktionsbausteins in der CPU mit zusätzlicher Anzeige der Fahrkurve.
 - Der Einsatz dieses Templates kann sich auf die Performance des Panels auswirken.
 - Das Template gibt es für verschiedene Displayauflösungen.
- VMC_AxisControl_PT
 - Template zur Ansteuerung des FB 875 - VMC_AxisControl_PT-Funktionsbausteins in der CPU, dessen Antrieb über Pulse Train angebunden ist.
 - Das Template gibt es für verschiedene Displayauflösungen.



Bitte beachten Sie, dass aktuell keine ECO-Panels unterstützt werden!

Installation in Movicon

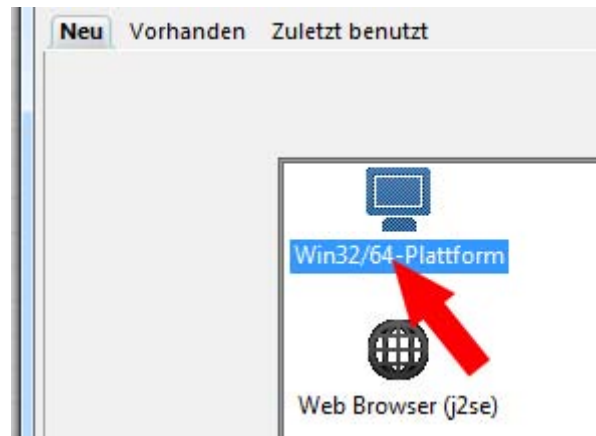
1. ➔ Gehen Sie in den Service-Bereich von www.vipa.com.
2. ➔ Laden Sie aus dem Downloadbereich unter "*VIPA Lib*" die "*Symbolbibliothek für Movicon*".
3. ➔ Geben Sie ein Zielverzeichnis an, in dem die Bausteine abzulegen sind und starten Sie den Entpackvorgang mit [OK].
4. ➔ Öffnen Sie die Bibliothek nach dem Entpackvorgang und ziehen Sie per Drag&Drop die *Symbolbibliothek* "*vipa simple motion control VX.X.X.msxz*" und die *Sprachta-*
belle "*vipa simple motion control VX.X.X.CSV*" in das Movicon-Benutzer-Verzeichnis ...\\Public\\Documents\\Progea\\Movicon\\Symbols.
 - ⇒ Nach einem Neustart von Movicon steht Ihnen in Movicon die Symbolbibliothek über die "*Vorlagenbibliothek*" zur Verfügung.

Damit die Texte der Templates richtig angezeigt werden, müssen Sie die Sprachtabelle in Ihr Projekt importieren. ↗ "*Sprachtabelle importieren*" auf Seite 319

7.1 Neues Projekt erstellen

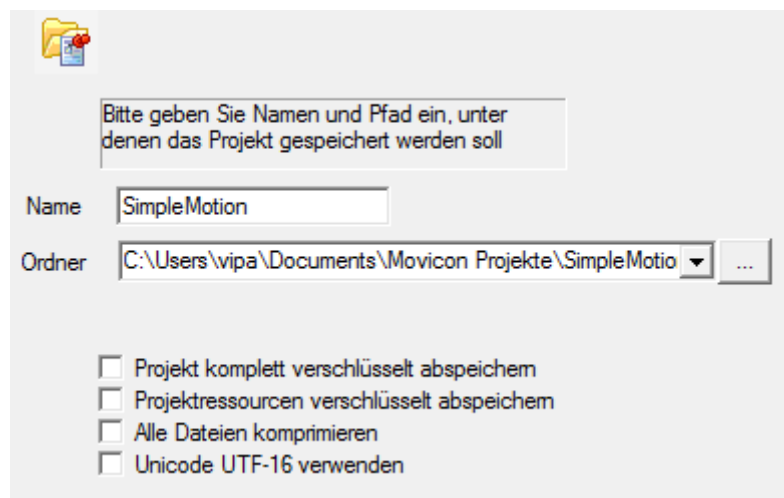
Projekt erstellen

1. ➤ Starten Sie Movicon und öffnen Sie den Projekt-Wizard über "*Datei* ➔ *Neu*".
2. ➤ Wählen Sie als Zielplattform "*Win32/64-Plattform* " und klicken Sie auf [Öffnen].



⇒ Es öffnet sich das Dialogfenster "*Projektname*".

3. ➤ Geben Sie unter "*Name*" einen Namen für Ihr Projekt an.
Geben Sie unter "*Ordner*" einen Speicherort an.
Lassen Sie alle Einstellungen deaktiviert und klicken Sie auf [Weiter].



⇒ Es öffnet sich das Dialogfenster "*Benutzer*".

4. ➔ Treffen Sie die passenden Benutzereinstellungen, sofern gewünscht, bzw. aktivieren Sie nur "CRF-21-Part..." und klicken Sie auf [Weiter].

Benutzereinstellungen

☐ Projekt mit Passwort geschützt

Name Entwickler: <Projektentwicklername hier eingeben>

Passwort Entwickler: []

Passwort wiederholen: []

☐ Passwort-Manager aktiv

☐ Standard-Benutzergruppen erstellen

☐ Windows-Benutzergruppen übernehmen: <Servername hier eingeben>

☐ Ändern von Benutzern zur Laufzeit

☐ Login lokaler Windows-Benutzer einschalten

☒ CRF-21-Part 11-Einstellungen einschalten

⇒ Es öffnet sich das Dialogfenster "Kommunikationstreiber hinzufügen".

5. ➔ Da die Anbindung zur CPU über TCP/IP erfolgt, aktivieren Sie in der "Liste der verfügbaren Kommunikationstreiber" den Treiber "VIPA" > "Ethernet S7 TCP" und klicken Sie auf [Weiter].

Kommunikationstreiber hinzufügen

Liste der verfügbaren Kommunikationstreiber...

- Shared Memory
- Siemens
- SNMP
- Systeme Helmholz
- WITS
- Vipa
 - ☒ Ethernet S7 TCP
 - ☐ S7-MPI PC Adapter
 - ☐ Vipa Embedded MPI
 - ☐ VIPA PROFIBUS DP Slave

Eigenschaften der Kommunikationstreiber

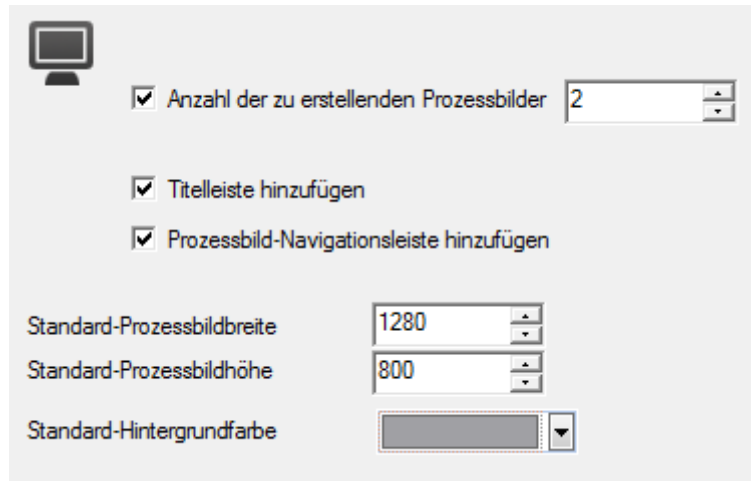
Property	Value
<input checked="" type="checkbox"/> Allgemein	
Name	S7 TCP
Dateiname	S7TCP.dll
Version	S7 TCP ver. 11....
Letzter Fehler	

Supported protocol: TCP protocol
 Activation Code: No, Require License Option: No
 Supported devices: Siemens SIMATIC PLCs S7-300/400 series, VIPA System 200V, 300V, 300V

< Zurück Weiter > Abbrechen Hilfe

⇒ Es öffnet sich das Dialogfenster "Prozessbilder".

6. ➤ Geben Sie 2 Prozessbilder und deren Größe, welche zu ihrem Panel passt an und klicken Sie auf [Weiter].



Monitor Icon

☒ Anzahl der zu erstellenden Prozessbilder 2

☒ Titelleiste hinzufügen

☒ Prozessbild-Navigationsleiste hinzufügen

Standard-Prozessbildbreite 1280

Standard-Prozessbildhöhe 800

Standard-Hintergrundfarbe

⇒ Es öffnet sich das Dialogfenster *"Datenbankeinstellungen (ODBC)"*.

7. ➤ Sofern Sie eine Datenbankbindung wünschen, können Sie hier die entsprechenden Einstellungen durchführen. Ansonsten klicken sie auf [Weiter].

⇒ Es öffnet sich das Dialogfenster *"Datenlogger und Rezept-Einstellungen (ODBC)"*.

8. ➤ Sofern Vorlagen generiert werden sollen, können Sie hier die entsprechenden Einstellungen durchführen. Ansonsten klicken sie auf [Weiter].

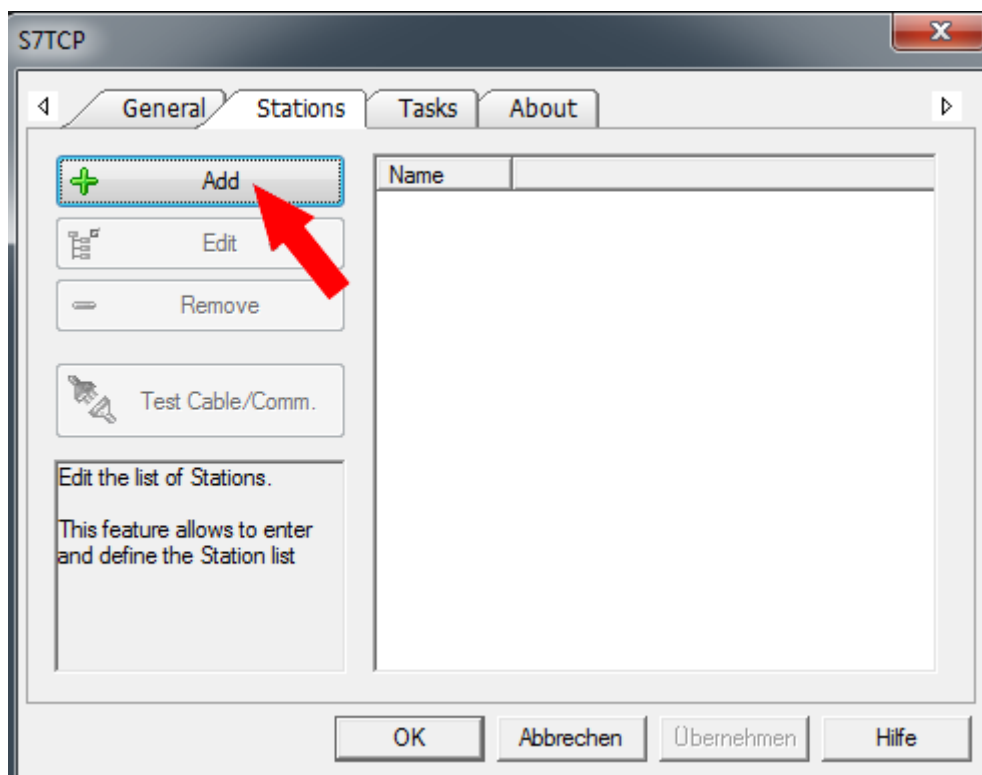
⇒ Es öffnet sich das Dialogfenster *"Alarm-Einstellungen"*.

9. ➤ Sofern Alarme generiert werden sollen, können Sie hier die entsprechenden Einstellungen durchführen. Ansonsten klicken sie auf [Fertig stellen].

⇒ Ihr Projekt wird mit den getroffenen Einstellungen erstellt und es öffnet sich automatisch der Einstelldialog für den Kommunikationstreiber *"S7TCP"*.

10. ➤ Wählen Sie den Reiter *"Stations"*.

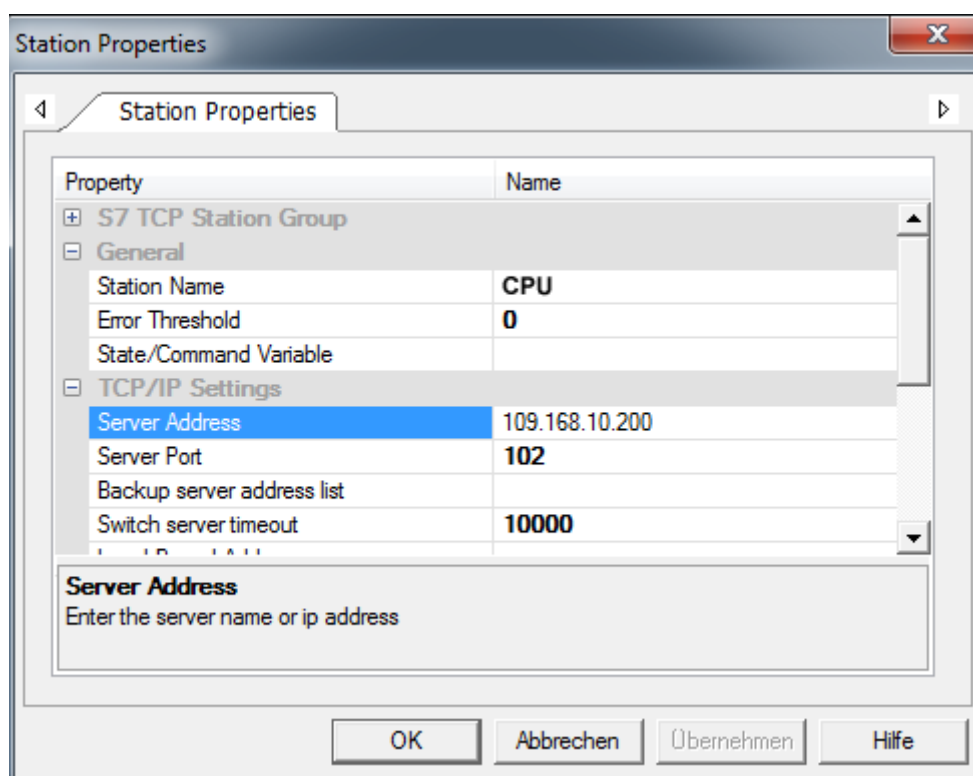
11. ➤ Zum Hinzufügen einer neuen Station klicken Sie auf [+ Add].



⇒ Es öffnet sich das Dialogfenster "Station Properties".

12. ➤ Geben Sie unter "Station Name" einen Stationsnamen an. Dieser Name ist weiter unten für das Prozessbild im Initialisierungs-Dialog zu übernehmen. Erlaubte Zeichen: A-Z, a-z, 0-9, Leerzeichen und die Trennzeichen _ und -

Geben Sie unter "Server Address" die IP-Adresse der CPU an und klicken Sie auf [OK].



13. ➤ Verneinen Sie die Abfrage zum Import von Variablen aus der SPS-Datenbank und schließen Sie den "S7TCP"-Dialog mit [OK].

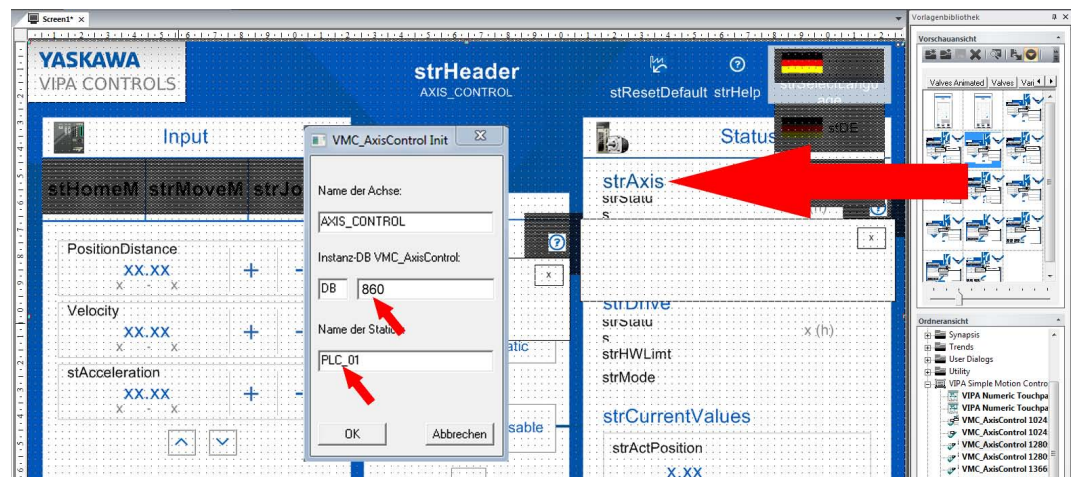
⇒ Das Projekt und der Arbeitsbereich sind nun für die Benutzung freigeben. Im Projektbaum wurde unter "*Ressourcen > SimpleMotion*" zu den Standardelementen folgendes angelegt:

- Kommunikationsobjekte
 - Kommunikationstreiber S7 TCP
- Prozessbilder
 - Prozessbild1
 - Prozessbild2
 - Schaltflächenleiste

7.2 Projekt in Movicon anpassen

Prozessbild konfigurieren

1. ➤ Öffnen Sie über "*Ressourcen > SimpleMotion > Prozessbilder*" das "Prozessbild1".
2. ➤ Navigieren Sie in der "Ordneransicht" zu "*vipa simple motion control ...*" und ziehen Sie aus der "Vorschauansicht" das Template in das "Prozessbild1", welches der Auflösung Ihres Panels entspricht.



⇒ Es öffnet sich der Initialisierungs-Dialog

3. ➤ Geben Sie einem Namen für die Achse an. Erlaubte Zeichen: A-Z, a-z, 0-9, Leerzeichen und die Trennzeichen _ und -

Geben Sie die Nummer des Instanz-DB an, welchen Sie in ihrem SPS-Programm verwenden.

Geben Sie den Stationsnamen an. Dieser muss mit dem "*Station Name*" aus den "*Station Properties*" der "*S7 TCP*"-Kommunikationseinstellungen übereinstimmen. Erlaubte Zeichen: A-Z, a-z, 0-9, Leerzeichen und die Trennzeichen _ und -

⇒ Mit [OK] werden alle Variablen, sowie deren Strukturen erzeugt und die Adressen werden auf die vorgegebene Zieladresse eingestellt.

4. ➔ Platzieren Sie das Template und passen Sie ggf. dessen Größe an.



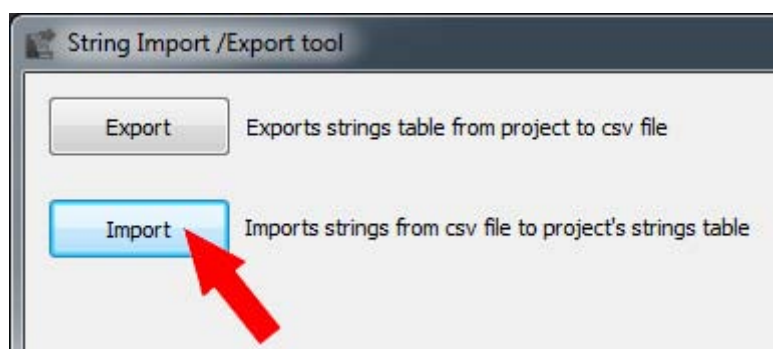
Für jedes Template werden Variablen unter dem entsprechenden Namen angelegt. Beim Löschen des Templates sind die entsprechenden Variablen wieder zu löschen. Diese können Sie sich unter "Ressourcen > SimpleMotion > Kommunikationsobjekte > Variablen" markieren lassen. Beim Löschen sind diese zusammen mit dem übergeordneten Verzeichnis zu löschen. Sofern keine weiteren Templates auf die "Strukturdefinitionen" für den Axis-Control zugreifen, sind diese ebenfalls zu löschen.

Sprachtabelle importieren

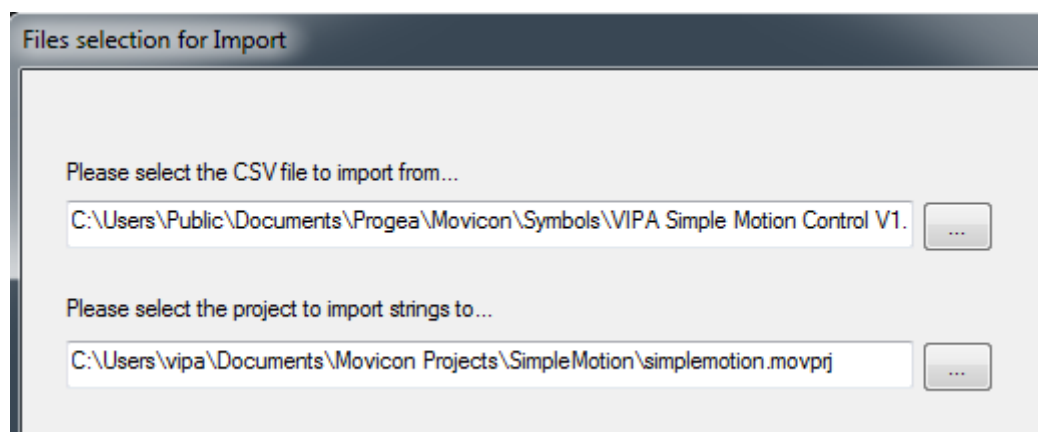
Die Templates beziehen die angezeigten Texte aus einer Sprachtabelle, welche aus dem Arbeitsverzeichnis in Ihr Projekt zu importieren ist.

1. ➔ Wählen Sie "Extras ➔ Zeichenfolgen Import/Export".

⇒ Es öffnet sich das "String Import/Export tool".



2. ➔ Klicken Sie auf [Import].
3. ➔ Für die CSV-Datei navigieren Sie mit [...] in Ihr Movicon-Benutzer-Verzeichnis ... \Public\Documents\Progea\Movicon\Symbols und wählen Sie die Datei "vipa simple motion control VX.X.X.CSV" aus.
4. ➔ Als Projektverzeichnis geben Sie die Projekt-Datei "simplemotion.movprj" an, welche sich im Benutzer-Verzeichnis wie z.B. ... \vipa\Documents\Movicon Projects \SimpleMotion befindet.



5. ➔ Klicken Sie auf [Weiter].
- ⇒ Es öffnet sich "Language selection".

6. ➤ Wählen Sie [Select all languages] und klicken Sie auf [Fertig stellen].



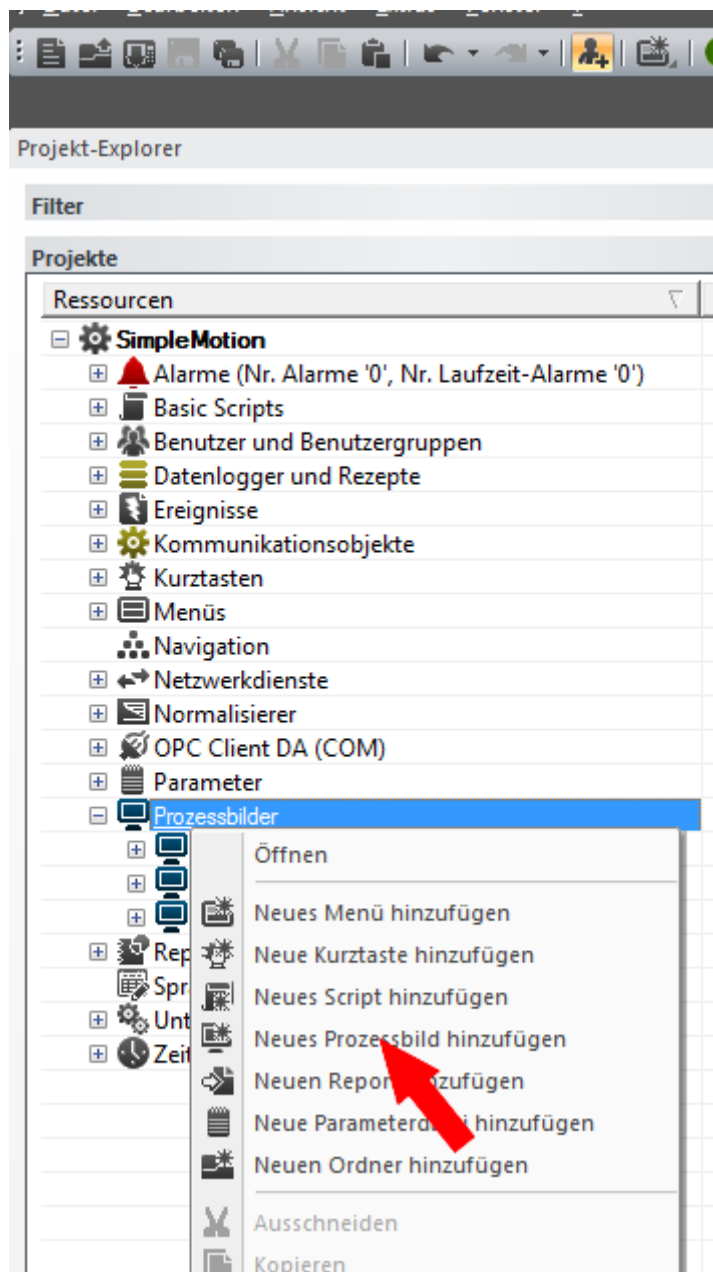
⇒ Die Sprachtabelle wird in Ihr Projekt importiert.

7. ➤ Schließen Sie nach erfolgreichem Import das *"String Import/Export tool"*.


Numerisches Eingabefeld anpassen

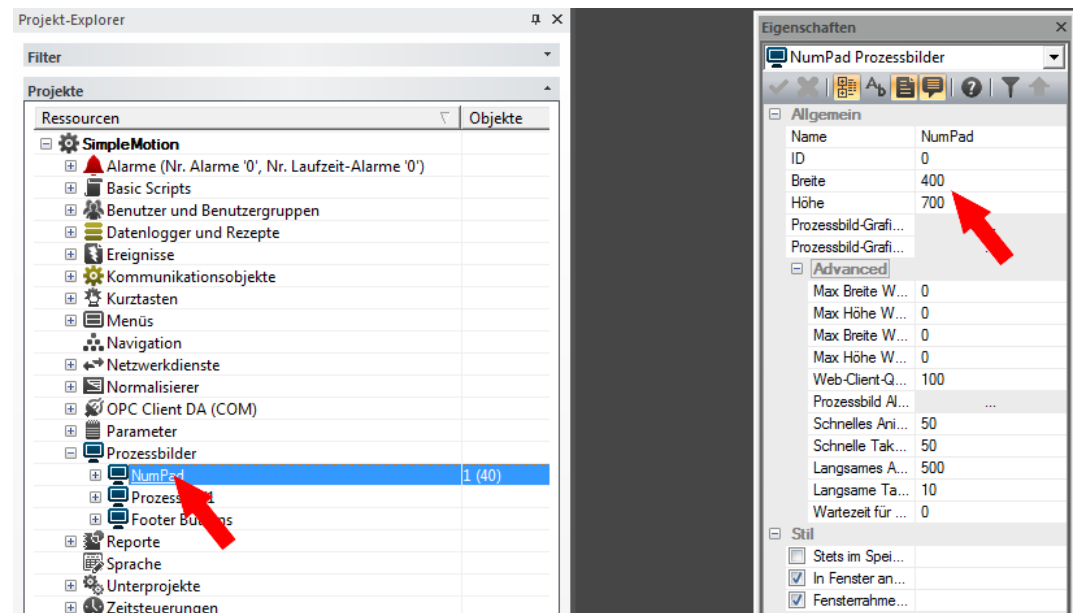
Unter den Templates befindet sich ein *"Numeric Touchpad"* in verschiedenen Auflösungen. Dies ist ein an die VMC_AxisControl-Templates angepasstes Eingabefeld für verschiedene Displayauflösungen. Mit folgender Vorgehensweise können Sie dieses Touchpad anstelle des Default-Eingabefeldes verwenden.

1. ➤ Klicken Sie auf *"Ressourcen > SimpleMotion > Prozessbilder"* und wählen Sie *"Kontextmenü → Neues Prozessbild hinzufügen"*.

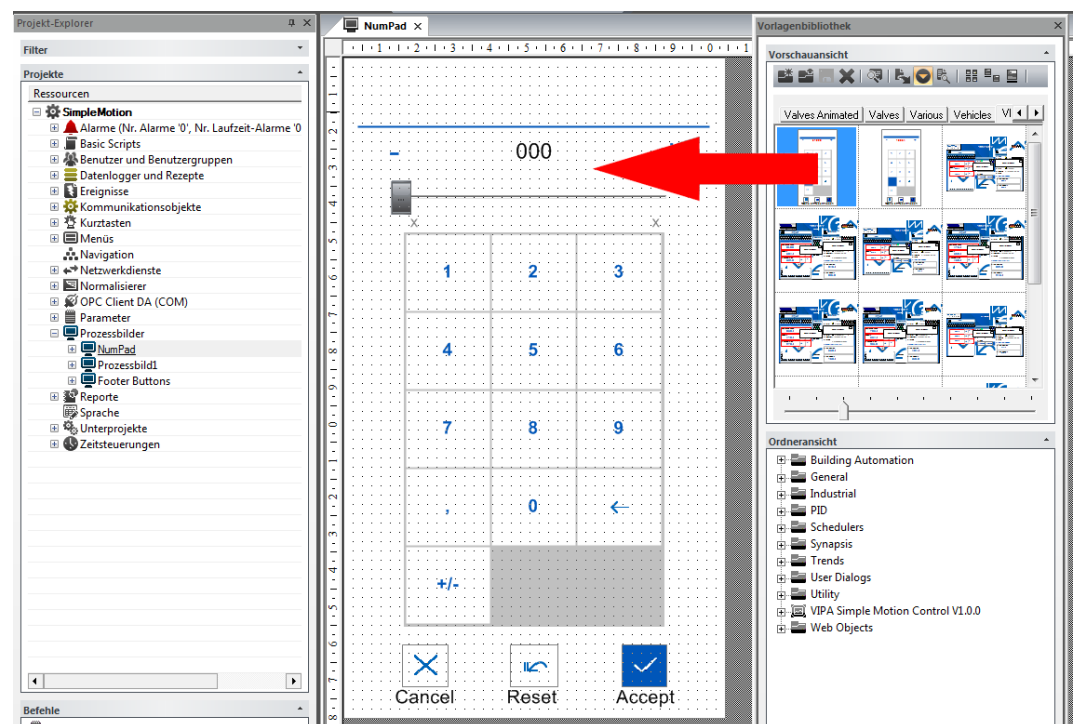


2. ➤ Vergeben Sie einen Namen wie z.B. *"NumPad"* und bestätigen Sie mit [OK].

3. Klicken Sie auf das Prozessbild "NumPad" und passen Sie über "Kontextmenü → Eigenschaften" Breite und Höhe an wie z.B. "Breite" = 400 und "Höhe" = 700. Übernehmen Sie mit  Ihre Einstellungen.

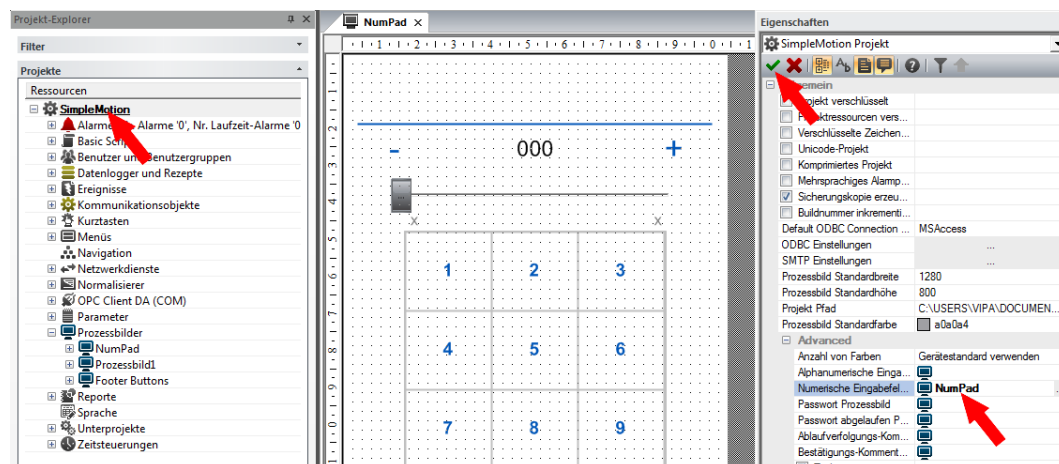


4. Wählen Sie "Ansicht → Vorlagenbibliothek". Navigieren Sie in der "Ordneransicht" zu "vipa simple motion control ..." und ziehen Sie aus der "Vorschauansicht" das "Numeric Touchpad"-Template in das Prozessbild "NumPad", welches der Auflösung Ihres Panels entspricht.

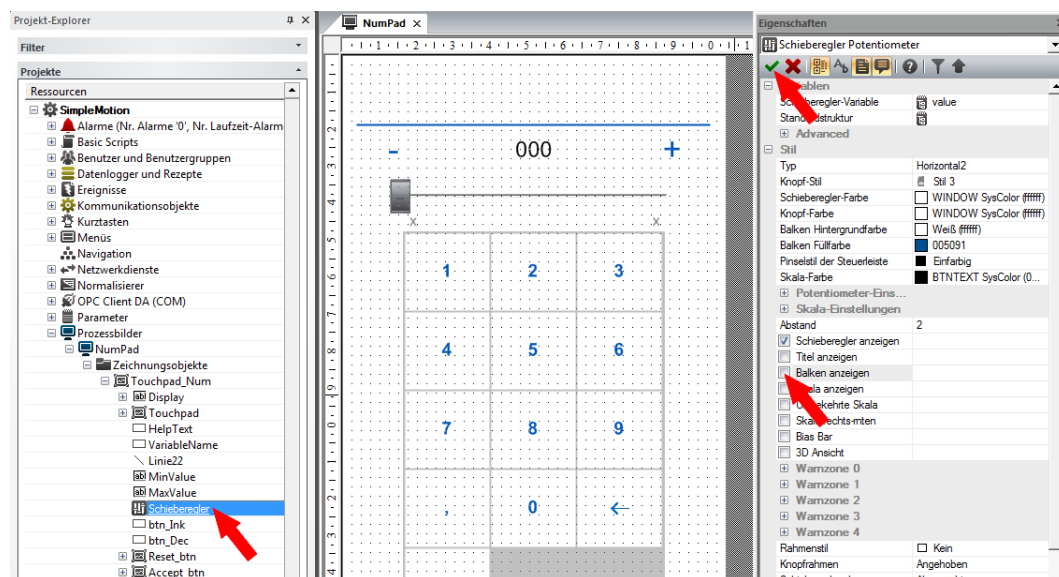


5. Passen Sie ggf. dessen Größe an.
6. Klicken Sie auf "Ressourcen > SimpleMotion" und wählen Sie "Kontextmenü → Eigenschaften".

7. Stellen Sie unter **"Allgemein > Advanced"** das numerische Eingabefeld **"NumPad"** ein. Übernehmen Sie mit ☒ Ihre Einstellungen.



8. Zur optischen Anpassung klicken Sie unter **"Ressourcen > SimpleMotion > Prozessbilder > NumPad > Zeichnungsobjekte > Touchpad_Num"** auf **"Schieberegler"** und wählen Sie **"Kontextmenü → Eigenschaften"**. Erweitern Sie den Bereich **"Stil"** und deaktivieren Sie **"Balken anzeigen"**.




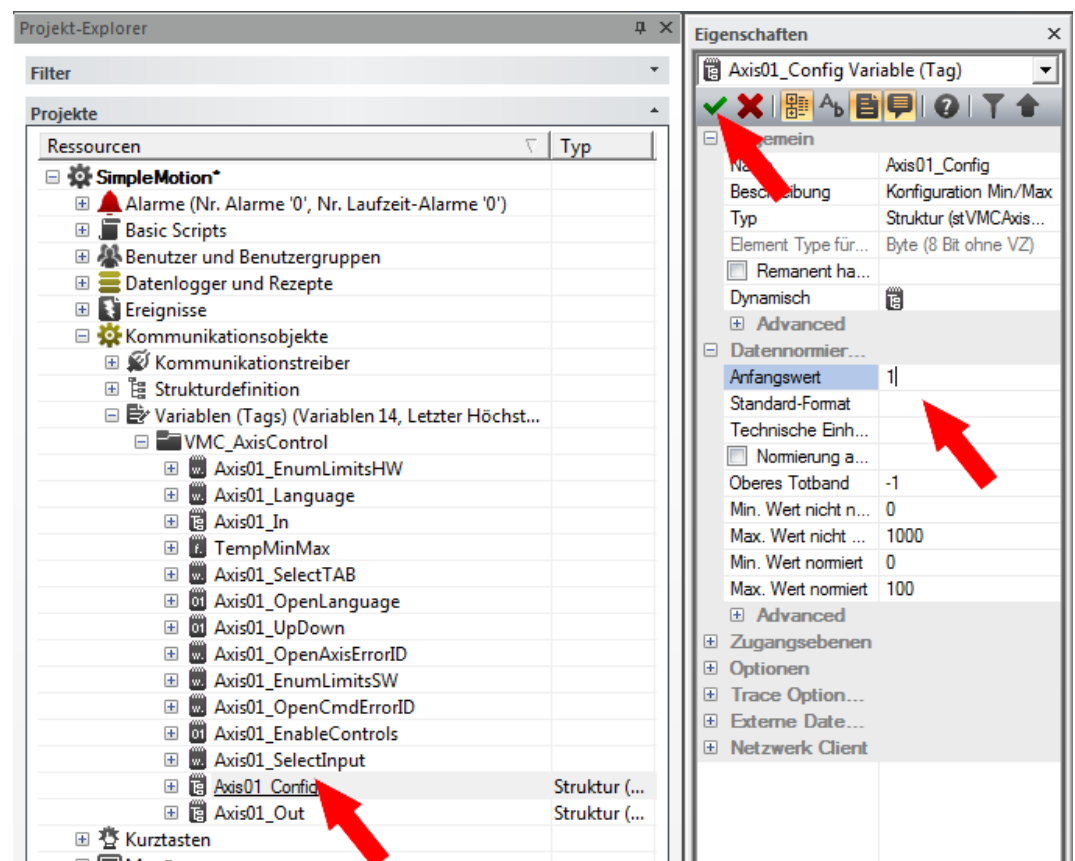
Grenzwerte und Standardwerte anpassen

Bei der Platzierung eines Templates in einem Prozessbild werden die zugehörigen Variablen und Strukturdefinitionen unter *"Ressourcen > SimpleMotion > Kommunikationsobjekte > Variablen > VMC_AxisControl > ..._Config"* automatisch erzeugt. Hierbei werden folgende Variablen angelegt und mit Initialwerten versehen:

AccelerationMaxValue	- Maximaler Beschleunigungswert
AccelerationMinValue	- Minimaler Beschleunigungswert
DecelerationMaxValue	- Maximaler Verzögerungswert
DecelerationMinValue	- Minimaler Verzögerungswert
HomePosMaxValue	- Maximale Home-Position
HomePosMinValue	- Minimale Home-Position
JogAccelerationMaxValue	- Maximale Beschleunigungswert Jog-Modus
JogAccelerationMinValue	- Minimaler Beschleunigungswert Jog-Modus
JogDecelerationMaxValue	- Maximaler Verzögerungswert Jog-Modus
JogDecelerationMinValue	- Minimaler Verzögerungswert Jog-Modus
PositionMaxValue	- Maximaler Positionswert
PositionMinValue	- Minimaler Positionswert
VelocityMaxValue	- Maximaler Geschwindigkeitswert
VelocityMinValue	- Minimaler Geschwindigkeitswert


➔ Zur Anpassung der Grenz- und Standardwerte klicken Sie auf *"Ressourcen > SimpleMotion > Kommunikationsobjekte > Variablen > VMC_AxisControl > ..._Config"* und wählen Sie *"Kontextmenü → Eigenschaften"*.

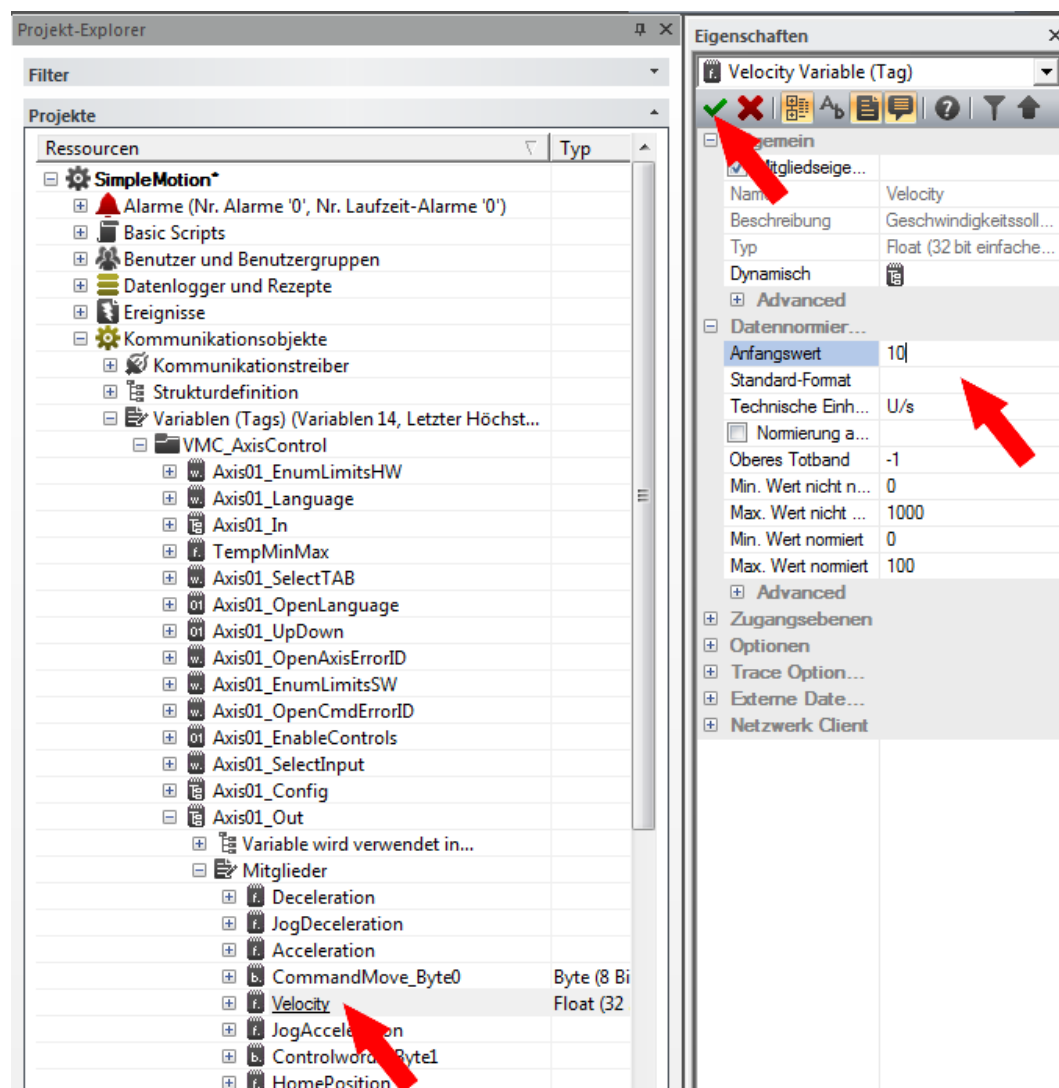
⇒ Unter *"Datennormierung"* können Sie die entsprechenden Werte anpassen. Übernehmen Sie mit  Ihre Einstellungen.



Technische Einheiten anpassen

Bei der Platzierung eines Templates in einem Prozessbild werden die zugehörigen Variablen mit ihren technischen Einheiten automatisch erzeugt. Diese können Sie über die Eigenschaften anpassen.

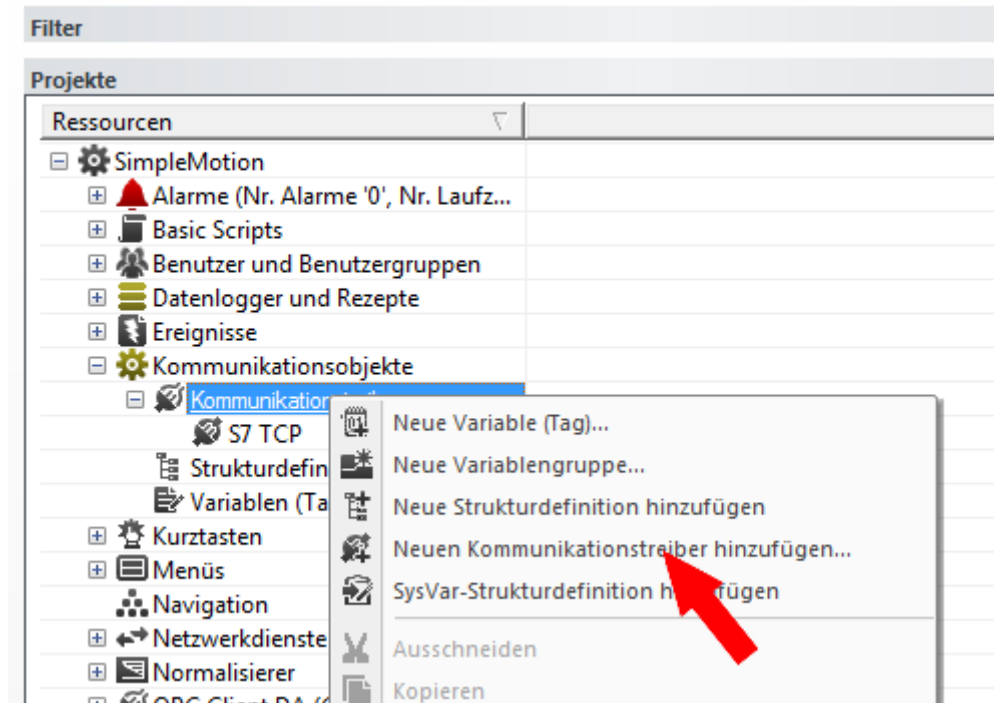
- ➔ Zur Anpassung der Technischen Einheiten z.B. für die Geschwindigkeit klicken Sie auf "Ressourcen > SimpleMotion > Kommunikationsobjekte > Variablen > VMC_AxisControl > ..._Out > Mitglieder > Velocity" und wählen Sie "Kontextmenü → Eigenschaften".
- ⇒ Unter "Datennormierung" können Sie die entsprechenden Werte anpassen. Übernehmen Sie mit  Ihre Einstellungen.



**Kommunikationstreiber
manuell hinzufügen**

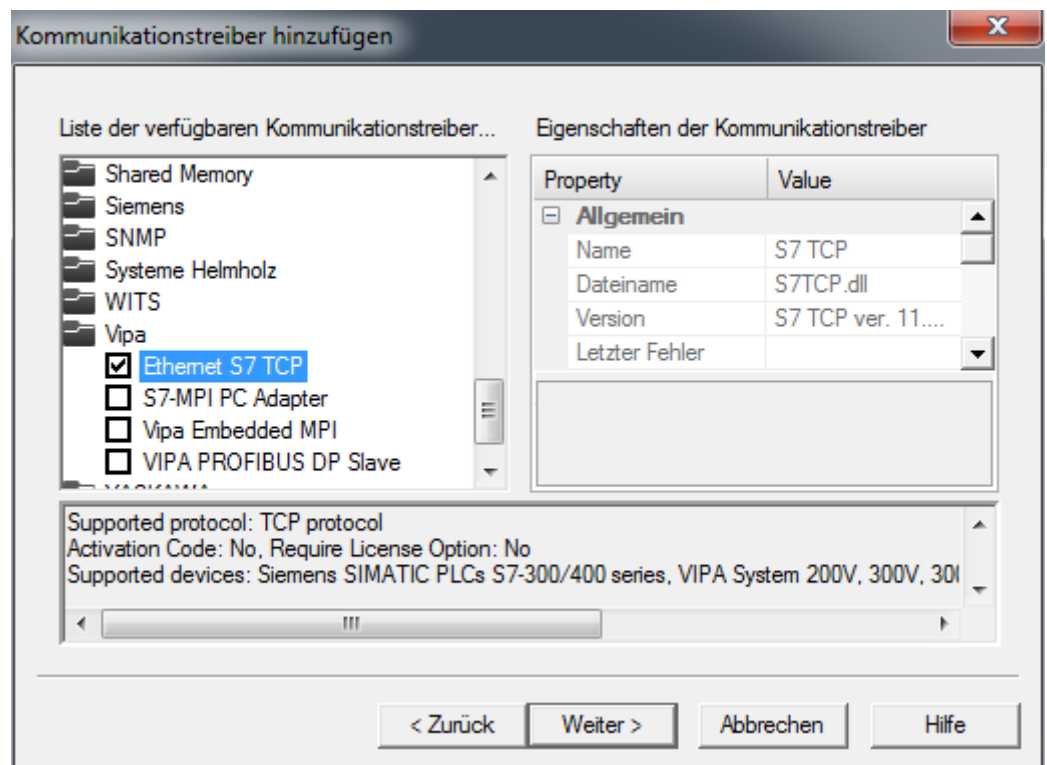
Anstelle über Wizard können Sie den Kommunikationstreiber auch manuell hinzufügen:

1. Klicken Sie unter *"Ressourcen > SimpleMotion > Kommunikationsobjekte"* auf *"Kommunikationstreiber"* und wählen Sie *"Kontextmenü
→ Neuen Kommunikationstreiber hinzufügen"*.



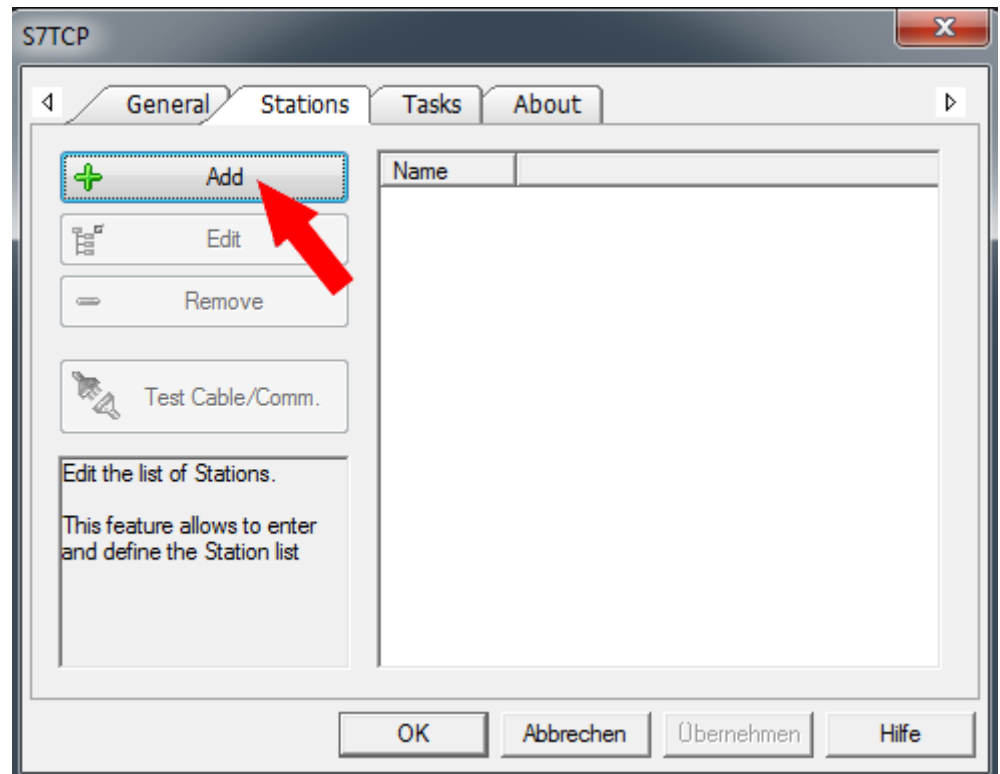
⇒ Es öffnet sich das Dialogfenster *"Neuer Komm. E/A-Treiber"*.

2. ➔ Da die Anbindung zur CPU über TCP/IP erfolgt, aktivieren Sie in der *"Liste der verfügbaren Kommunikationstreiber"* den Treiber *"VIPA" > "Ethernet S7 TCP"* und klicken Sie auf [OK].



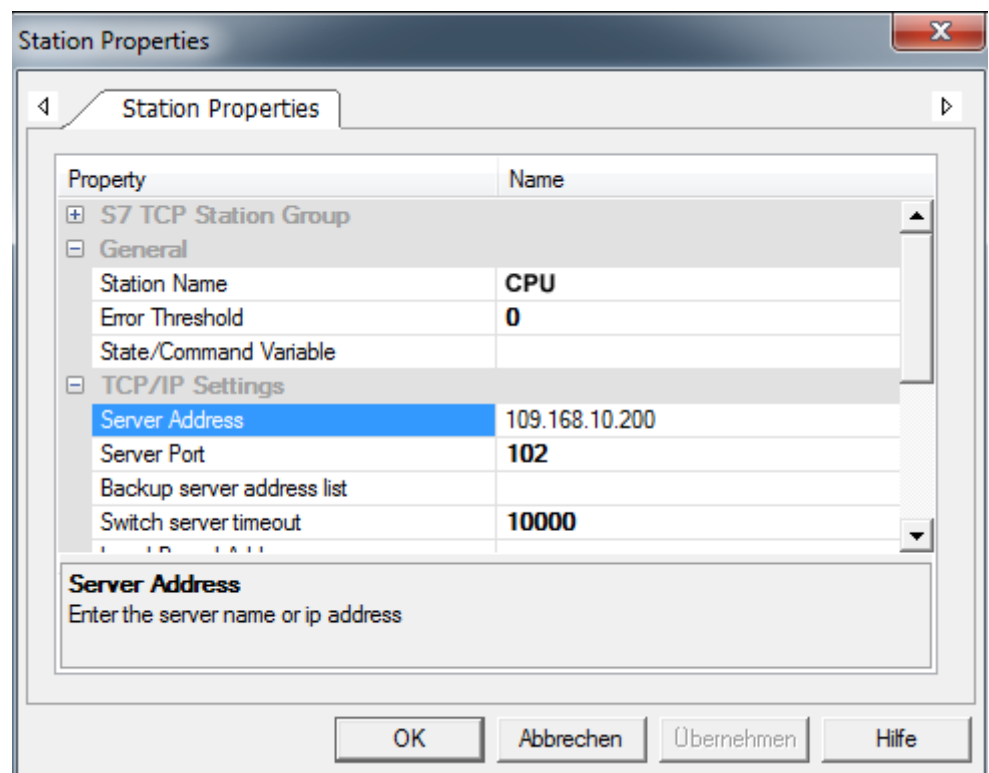
- ⇒ Der Kommunikationstreiber *"S7 TCP"* wird unter *"Ressourcen > SimpleMotion > Kommunikationsobjekte > Kommunikationstreiber"* aufgeführt.
3. ➔ Klicken Sie auf *"S7 TCP"* und wählen Sie *"Kontextmenü > Kommunikationstreiber-Einstellungen"*.
- ⇒ Es öffnet sich das Dialogfenster *"S7 TCP"*.
4. ➔ Wählen Sie den Reiter *"Stations"*.

5. ➔ Zum Hinzufügen einer neuen Station klicken Sie auf [+ Add].



⇒ Es öffnet sich das Dialogfenster "Station Properties".

6. ➔ Geben Sie unter "Station Name" einen Stationsnamen an. Erlaubte Zeichen: A-Z, a-z, 0-9, Leerzeichen und die Trennzeichen _ und -
Geben Sie unter "Server Address" die IP-Adresse der CPU an und klicken Sie auf [OK].



7. ➤ Verneinen Sie die Abfrage zum Import von Variablen aus der SPS-Datenbank und schließen Sie den "S7 TCP"-Dialog mit [OK].

7.3 Inbetriebnahme

Projekt auf Zielgerät übertragen

Ihr Projekt können Sie über Ethernet in Ihr Panel übertragen. Über die in Ihrem Panel vorinstallierte Movicon Runtime-Version wird Ihr Projekt ablauffähig.

1. ➤ Verbinden Sie Ihren PC und Ihr Panel über Ethernet.
2. ➤ Starten Sie Ihr Panel und ermitteln Sie im "Startup-Manager" die IP-Adresse Ihres Panels.
3. ➤ Rufen Sie im "Startup-Manager" den Menüpunkt "Autostart" auf.
4. ➤ Damit Sie mit Movicon über Ethernet ein Projekt in Ihr Panel übertragen können, müssen Sie unter "Autostart" die Option "Movicon TCP Upload Server" aktivieren.

⇒ Bestätigen Sie die Abfrage zur Aktivierung.

5. ➤ Sie können jetzt aus Movicon Ihr Projekt in Ihr Panel übertragen. Klicken Sie hierzu in Movicon in "Ressourcen" auf "SimpleMotion" und wählen Sie "Kontextmenü ➔ Projekt zum Zielgerät transferieren".

⇒ Es öffnet sich der Transfer-Dialog.

6. ➤ Wählen Sie unter "PlugIn-Typ" "TCP".

Geben Sie unter "Server" die IP-Adresse des Panels an.

Tragen Sie unter "Benutzername" und "Passwort" die Zugangsdaten zu Ihrem Panel ein.

Standardmäßig werden folgende Zugangsdaten verwendet:

- Benutzername: wince
- Passwort: vipatp

Geben Sie unter "Zielpfad" Ihre Speicherkarte an und legen Sie ein Projektverzeichnis an.

7. ➤ Starten Sie die Übertragung mit [Projekt transferieren].

8. ➔ Nach erfolgreichem Transfer können Sie Ihr Projekt auf dem Panel im Autostart-Verzeichnis eintragen und in Betrieb nehmen.

**VORSICHT!**

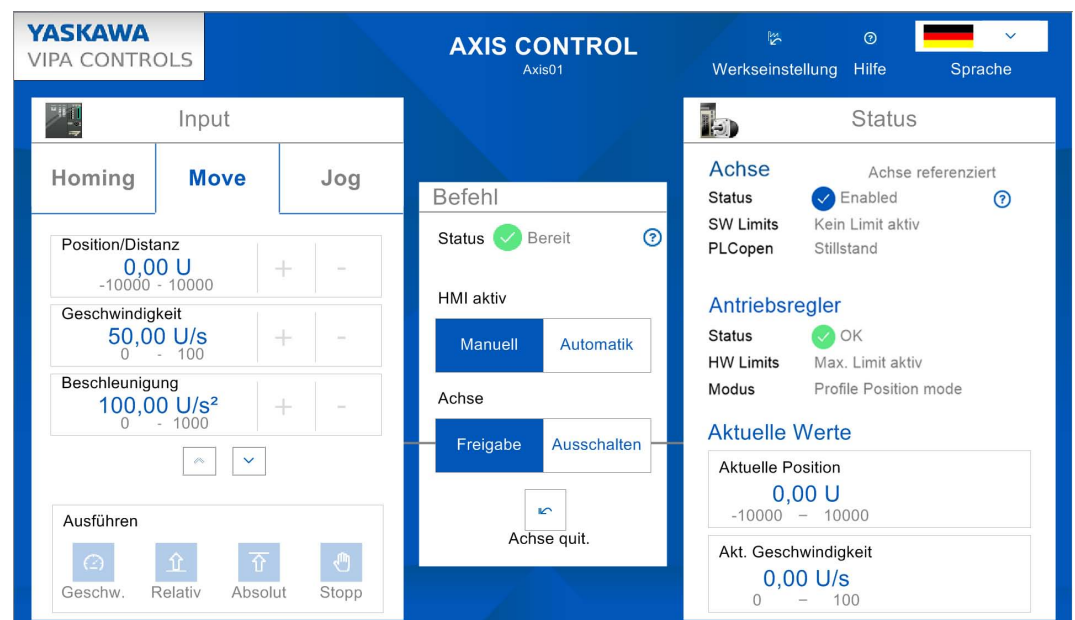
Bitte beachten Sie immer die Sicherheitshinweise zu Ihrem Antrieb, insbesondere bei der Inbetriebnahme!

7.3.1 VMC_AxisControl über das Panel ansteuern

Inbetriebnahme

Es wird vorausgesetzt, dass Sie Ihre Applikation aufgebaut haben und Sie Ihren Antrieb mit dem VMC_AxisControl-Funktionsbaustein ansteuern können.

- ➔ Binden Sie Ihre CPU an Ihr Panel an und schalten Sie Ihre Applikation ein.
 ➔ Das Panel startet mit der Oberfläche zur Steuerung Ihres Antriebs.



Damit Sie Ihren Antrieb über das Panel steuern können, schalten Sie "HMI aktiv" auf [Manuell]. Sofern die Statusanzeige keine Fehler zurückmeldet, können Sie den Antrieb mit [Freigabe] für die Ansteuerung freigeben. Sie können jetzt Ihren Antrieb über die entsprechenden Schaltflächen steuern.

7.3.1.1 Bedienung

Oberfläche



"Werkseinstellung"

- Über *"Werkseinstellung"* werden folgende Werte auf die Defaultwerte der Applikation zurückgesetzt, welche Sie, wie weiter oben beschrieben, entsprechend anpassen können:
 - Geschwindigkeit: 50U/s
 - Beschleunigung/Verzögerung: 100U/s²
 - Position/Home Position: 0U

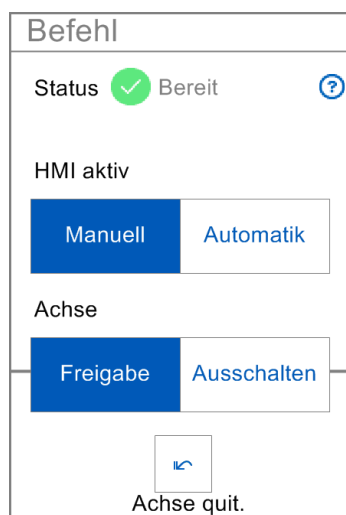
"Hilfe"

- Über *"Hilfe"* können Sie Ihre eigene Hilfedatei abrufen. Diese ist in Movicon entsprechend einzubinden.

"Sprache"

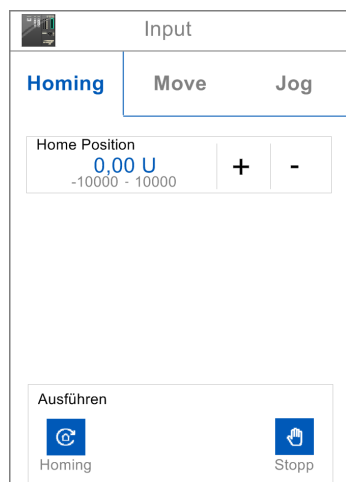
- Über *"Sprache"* können Sie die entsprechende Sprache für die Bedienoberfläche vorgeben.

"Befehl"



- *"Status"*
 - Hier bekommen Sie den aktuellen Status Ihres Fahrbefehls angezeigt.
- *"HMI aktiv"*
 - *"Manuell"*: Im aktivierten Zustand kann der Antrieb über das Panel gesteuert werden.
 - *"Automatik"*: Im aktivierten Zustand erfolgt die Steuerung des Antriebs über das SPS-Programm Ihrer CPU und kann vom Panel nicht beeinflusst werden.
- *"Achse"*
 - *"Freigabe"*: Im aktivierten Zustand und bei aktiviertem *"Manuell"* von *"HMI aktiv"* ist der Antrieb freigegeben und Sie können diesen über den Bereich *"Input"* steuern.
 - *"Ausschalten"*: Im aktivierten Zustand ist der Antrieb gesperrt und keine Steuerung möglich.
- *"Achse quit"*
 - Im Fehlerfall werden die Schaltflächen zur Steuerung inaktiv. Mit *"Achse quit."* können Sie Fehler quittieren und Schaltflächen wieder aktivieren.

"Input"



"Homing"

- Über das Eingabefeld bzw. [+] und [-] können Sie eine Homing-Position vorgeben und diese über *"Ausführen > Homing"* als Referenzpunkt anfahren
- Mit *"Ausführen > Stopp"* können Sie die Referenzfahrt stoppen.

Input

Homing Move Jog

Position/Distanz
0,00 U
-10000 - 10000 + -

Geschwindigkeit
50,00 U/s
0 - 100 + -

Beschleunigung
100,00 U/s²
0 - 1000 + -

Ausführen

Geschw. Relativ Absolut Stopp

Input

Homing Move Jog

Verzögerung
100,00 U/s²
0 - 1000 + -

Aktuelle Richtung Kürzeste Strecke Negative Richtung Positive Richtung

Ausführen

Geschw. Relativ Absolut Stopp

Input

Homing Move Jog

Geschwindigkeit
100,00 U/s
0 - 100 + -

Beschleunigung
100,00 U/s²
0 - 1000 + -

Verzögerung
100,00 U/s²
0 - 1000 + -

Ausführen

Negativ Positiv

"Move"

- Über das entsprechende Eingabefeld bzw. [+] und [-] können Sie *"Position/Distanz"*, *"Geschwindigkeit"*, *"Beschleunigung"* und *"Verzögerung"* vorgeben und über den entsprechenden Fahrbefehl unter *"Ausführen"* ausführen lassen. Mit [v] navigieren Sie nach unten.
 - *"Geschw"*: Bei Betätigung führt der Antrieb den Fahrbefehl mit konstanter Geschwindigkeit aus.
 - *"Relativ"*: Bei Betätigung fährt der Antrieb an die relative Position, welche Sie unter *"Position/Distanz"* vorgeben können.
 - *"Absolut"*: Bei Betätigung fährt der Antrieb an die absolute Position, welche Sie unter *"Position/Distanz"* vorgeben können.
 - *"Stopp"*: Bei Betätigung wird der Antrieb gestoppt.
 - *"Aktuelle Richtung"*: Im aktivierten Zustand wird die Fahrtrichtung beibehalten.
 - *"Kürzeste Strecke"*: Im aktivierten Zustand wird die kürzeste Entfernung zur vorgegebenen Position verwendet.
 - *"Negative Richtung"*: Im aktivierten Zustand wird die negative Fahrtrichtung verwendet.
 - *"Positive Richtung"*: Im aktivierten Zustand wird die positive Fahrtrichtung verwendet.

"Jog"

- Über das entsprechende Eingabefeld bzw. [+] und [-] können Sie *"Geschwindigkeit"*, *"Beschleunigung"* und *"Verzögerung"* vorgeben und über die Richtungstaste unter *"Ausführen"* den entsprechenden Fahrbefehl in negative bzw. positive Richtung ausführen lassen.
- Solange Sie eine der Richtungstasten betätigen, wird der Antrieb mit der eingestellten Beschleunigung auf die gewünschte Geschwindigkeit beschleunigt.
- Beim Loslassen der Richtungstaste wird der Antrieb mit der eingestellten Verzögerung gestoppt.

"Status"

The screenshot shows the 'Status' screen of the HMI. It is divided into three main sections: 'Achse' (Axis), 'Antriebsregler' (Drive Controller), and 'Aktuelle Werte' (Current Values).

Achse (Axis):

- Status:** Enabled (indicated by a blue checkmark icon).
- SW Limits:** Kein Limit aktiv (No limit active).
- PLCopen:** Stillstand (Standstill).

Antriebsregler (Drive Controller):

- Status:** OK (indicated by a green checkmark icon).
- HW Limits:** Max. Limit aktiv (Max. limit active).
- Modus:** Profile Position mode.

Aktuelle Werte (Current Values):

- Aktuelle Position:** 0,00 U (Current position: 0,00 U). The range is -10000 to 10000.
- Akt. Geschwindigkeit:** 0,00 U/s (Current speed: 0,00 U/s). The range is 0 to 100.

"Achse"

- **"Status"**: Hier wird Ihnen der Status Ihrer Achse angezeigt.
 - **"Enabled"**: Die Achse ist eingeschaltet.
 - **"Ready"**: Die Achse ist einschaltbereit.
 - **"Disabled"**: Die Achse ist deaktiviert.
 - **"Achsfehler"**: Es liegt ein Achsfehler vor mit Angabe der Fehlernummer.
 ↪ Kapitel 9 "ErrorID - Zusätzliche Fehlerinformationen" auf Seite 340
- **"SW Limits"**: Sobald SW-Grenzen bestehen, wird dies hier angezeigt.
- **"PLCopen"**: Hier wird Ihnen der PLCopen-Status angezeigt.

"Antriebsregler"

- **"Status"**: Hier wird Ihnen der Status des Antriebsreglers angezeigt.
- **"HW-Limits"**: Hier wird Ihnen eine eventuell eingestellte Begrenzung in Ihrem Antriebsregler angezeigt.
- **"Modus"**: Hier erhalten sie Informationen über das aktuell eingestellt Antriebsprofil.

"Aktuelle Werte"

- Hier werden die aktuellen Werte von **"Position"** und **"Geschwindigkeit"** angezeigt.
- Werte, welche außerhalb der definierten Grenzwerte liegen, werden rot eingerahmt.

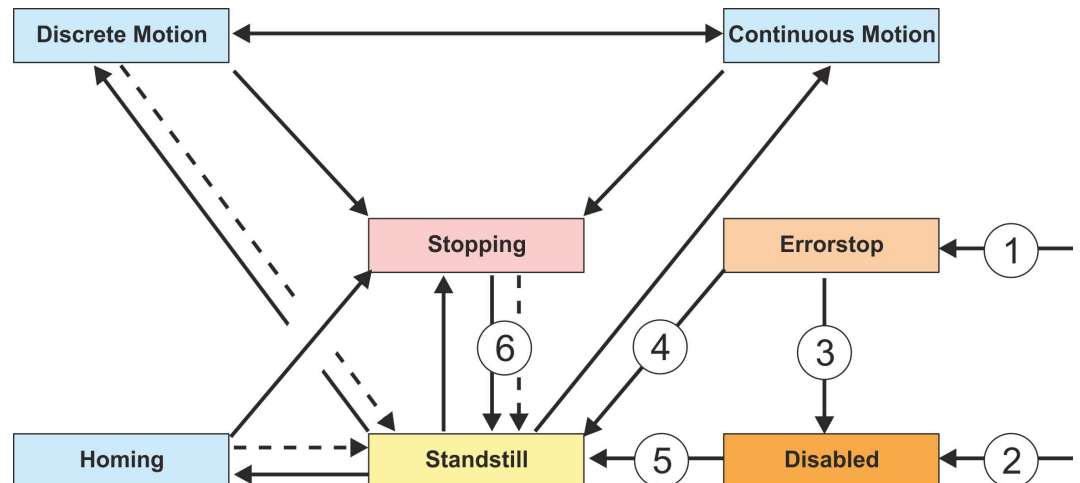
8 Zustände und Verhalten der Ausgänge

8.1 Zustände

Zustandsdiagramm

Im *Zustandsdiagramm* sind alle Zustände aufgeführt, die eine Achse annehmen kann. Eine Achse befindet sich immer in einem dieser Zustände. Je nach Ausgangszustand kann ein Zustandswechsel automatisch oder über die Bausteine der Achskontrolle erfolgen. Grundsätzlich werden Bewegungsaufgaben sequenziell abgearbeitet. Mit folgenden Funktionsbausteinen können Sie den Status abfragen:

- ↗ Kapitel 3.4.3.11 "FB 812 - MC_ReadStatus - PLCopen Status" auf Seite 153
- Parameter `PLCopenState` von ↗ Kapitel 3.4.2.2 "FB 860 - VMC_AxisControl - Control-Baustein Achskontrolle" auf Seite 132



-- ➔ Rücksprung wenn fertig

- (1) Aus jedem Status: An der Achse ist ein Fehler aufgetreten
- (2) Aus jedem Status: `MC_Power.Enable = FALSE` und es gibt keinen Fehler an der Achse
- (3) `MC_Reset` und `MC_Power.Status = FALSE`
- (4) `MC_Reset` und `MC_Power.Status = TRUE` und `MC_Power.Enable = TRUE`
- (5) `MC_Power.Enable = TRUE` und `MC_Power.Status = TRUE`
- (6) `MC_Stop.Done = TRUE` und `MC_Stop.Execute = FALSE`

Es gibt folgende Zustände

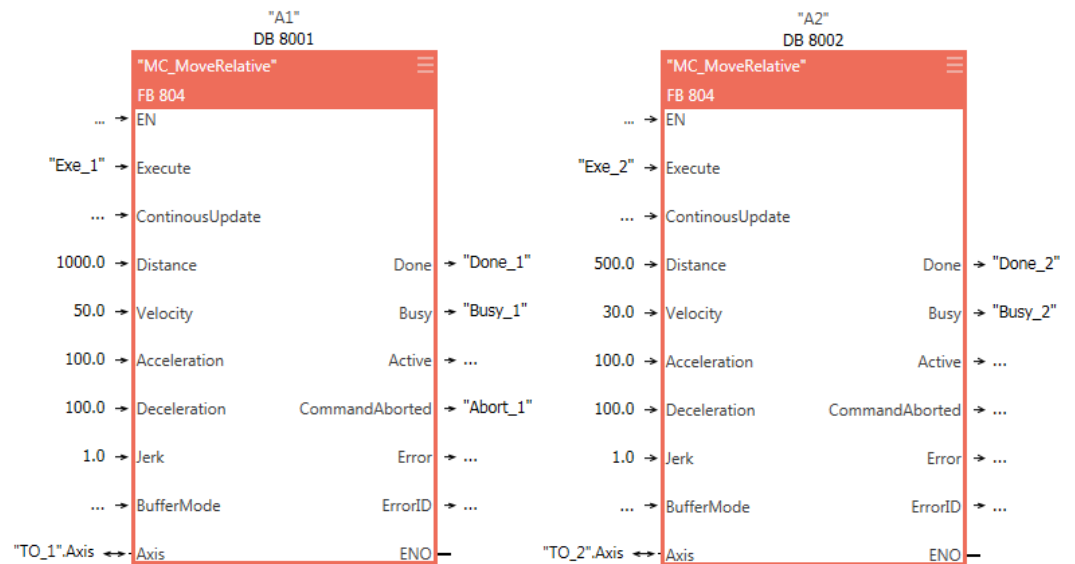
- Disabled
 - Grundzustand einer Achse.
 - Achse kann durch keinen Funktionsbaustein bewegt werden.
- Error Stop
 - Es ist ein Fehler an der Achse aufgetreten.
 - Achse wird gestoppt und ist für weitere Bewegungsaufgaben gesperrt.
 - Achse bleibt in diesem Zustand bis der Fehler behoben ist und ein RESET ausgelöst wird.
 - Fehler an einer Achse werden auch über den entsprechenden Funktionsbaustein zurück gemeldet.
 - Fehler an einem Funktionsbaustein führen nicht in diesen Zustand
- Stand Still
 - Bereit für Bewegungsaufgaben
 - Es liegt kein Fehler an der Achse vor
 - Es sind keine Bewegungsaufgaben an der Achse aktiv
 - Achse wird mit Spannung versorgt

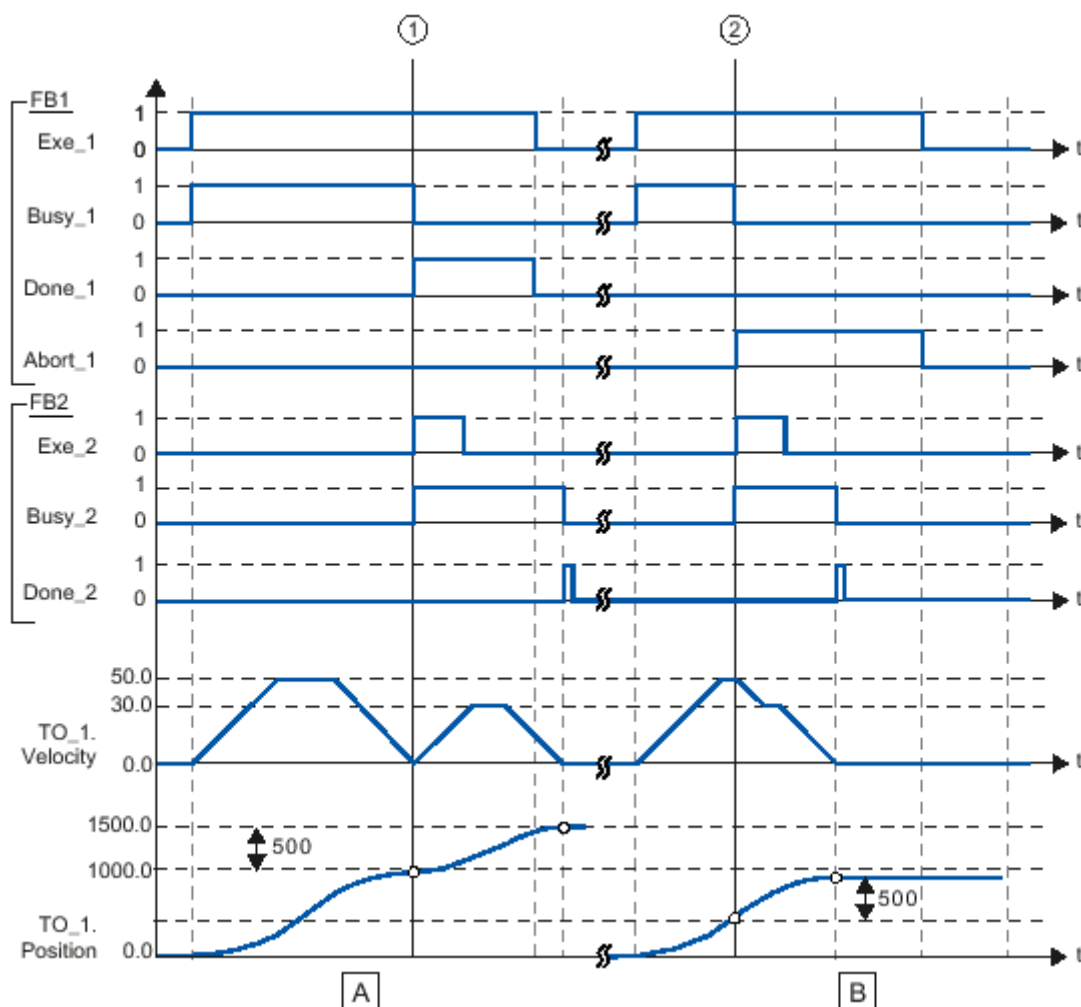
- Stopping
 - Achse wird aktuell gestoppt:
 - ↳ Kapitel 3.4.3.5 "FB 802 - MC_Stop - Achse stoppen" auf Seite 141
 - ↳ Kapitel 3.4.2.2 "FB 860 - VMC_AxisControl - Control-Baustein Achskontrolle" auf Seite 132
 - Der Zustand *Stopping* ist aktiv solange ein Stop Kommando vom aktiv ist (*Execute* = 1). Auch wenn die Achse schon gestoppt ist. Danach wechselt der Zustand automatisch nach *Standstill*.
- Homing
 - Die Achse führt aktuell eine Referenzfahrt durch:
 - ↳ Kapitel 3.4.3.4 "FB 801 - MC_Home - Achse referenzieren" auf Seite 139
 - ↳ Kapitel 3.4.2.2 "FB 860 - VMC_AxisControl - Control-Baustein Achskontrolle" auf Seite 132
 - Sobald die Achse referenziert ist, wechselt der Zustand automatisch nach *Standstill*.
- Discrete Motion
 - Die Achse führt aktuell eine Bewegungsaufgabe durch:
 - ↳ Kapitel 3.4.3.9 "FB 808 - MC_MoveAbsolute - Achse auf absolute Position verfahren" auf Seite 149
 - ↳ Kapitel 3.4.3.7 "FB 804 - MC_MoveRelative - Achse relativ verfahren" auf Seite 145
 - ↳ Kapitel 3.4.3.6 "FB 803 - MC_Halt - Achse anhalten" auf Seite 143
 - ↳ Kapitel 3.4.2.2 "FB 860 - VMC_AxisControl - Control-Baustein Achskontrolle" auf Seite 132
 - Sobald das Ziel der Bewegungsaufgabe erreicht ist, wechselt der Zustand automatisch nach *Standstill*.
- Continuous Motion
 - Die Achse führt eine dauerhafte Bewegungsaufgabe durch:
 - ↳ Kapitel 3.4.3.8 "FB 805 - MC_MoveVelocity - Achse verfahren mit konstanter Geschwindigkeit" auf Seite 147
 - ↳ Kapitel 3.4.2.2 "FB 860 - VMC_AxisControl - Control-Baustein Achskontrolle" auf Seite 132

8.2 Ablöseverhalten von Bewegungsaufträgen

Beispiel

Das Ablöseverhalten von Bewegungsaufträgen wird nachfolgend am Beispiel von MC_MoveRelative erklärt. ↗ Kapitel 3.4.3.7 "FB 804 - MC_MoveRelative - Achse relativ verfahren" auf Seite 145





- (A) Die Achse wird durch einen "MC_MoveRelative"-Auftrag (A1) um die Wegstrecke *Distance* 1000.0 verfahren (Startposition ist hier die Position 0.0).
- (1) Das Erreichen der Zielposition wird zum Zeitpunkt (1) über *Done_1* gemeldet. Zu diesem Zeitpunkt (1) wird ein weiterer MC_MoveRelative-Auftrag (A2) mit der Wegstrecke 500.0 gestartet. Das erfolgreiche Erreichen der neuen Zielposition wird über *Done_2* gemeldet. Da *Exe_2* vorher zurückgesetzt wurde, steht *Done_2* nur für einen Zyklus an.
- (B) Ein laufender MC_MoveRelative-Auftrag (A1) wird durch einen weiteren MC_MoveRelative-Auftrag (A2) abgelöst.
- (2) Der Abbruch wird zum Zeitpunkt (2) über *Abort_1* gemeldet. Die Achse wird anschließend mit der neuen Geschwindigkeit um die Wegstrecke *Distance* 500.0 verfahren. Das Erreichen der neuen Zielposition wird über *Done_2* gemeldet.

8.3 Verhalten der Ein- und Ausgänge

Ausschließlichkeit der Ausgänge

- Die Ausgänge *Busy*, *Done*, *Error* und *CommandAborted* schließen sich gegenseitig aus, es kann also an einem Funktionsbaustein nur einer dieser Ausgänge zu einer Zeit TRUE sein.
- Sobald der Eingang *Execute* TRUE wird, muss einer der Ausgänge TRUE werden. Ebenfalls kann nur einer der Ausgänge *Active*, *Error*, *Done* und *CommandAborted* zu einer Zeit TRUE sein.

Ausgangs-Zustand

- Die Ausgänge *Done*, *InVelocity*, *Error*, *ErrorID* und *CommandAborted* werden mit einer Flanke 1-0 am Eingang *Execute* zurückgesetzt, wenn der Funktionsbaustein nicht aktiv ist (*Busy* = FALSE).
- Die Kommandoausführung wird durch eine Flanke 1-0 an *Execute* nicht beeinflusst.
- Falls *Execute* bereits während der Kommandoausführung zurückgesetzt wird, so ist sichergestellt, dass einer der Ausgänge am Ende des Kommandos für einen SPS-Zyklus gesetzt wird. Erst danach werden die Ausgänge zurückgesetzt.

Eingangs-Parameter

- Die Eingangs-Parameter werden mit Flanke 0-1 an *Execute* übernommen.
- Zur Änderung der Parameter muss das Kommando neu getriggert werden.
- Falls ein Eingangs-Parameter nicht an den Funktionsbaustein übergeben wird, so bleibt der zuletzt an diesen Baustein übergebene Wert gültig.
- Beim ersten Aufruf muss ein sinnvoller Default-Wert übergeben werden.

Position und Distanz

- Der Eingang *Position* bezeichnet einen absoluten Positionswert.
- *Distance* bezeichnet ein relatives Maß als Abstand zweier Positionen.
- Sowohl *Position*, als auch *Distance* werden in technischen Einheiten, z.B. [mm] oder [°], entsprechend der Skalierung der Achse angegeben.

Parameter für das dynamische Verhalten

- Die Dynamikparameter für *Move*-Funktionen werden in technischen Einheiten mit der Zeitbasis Sekunde angegeben.
Ist eine Achse beispielsweise in Millimetern skaliert, so sind die Einheiten für *Velocity* [mm/s], *Acceleration* [mm/s²], und *Deceleration* [mm/s²].

Fehlerbehandlung

- Alle Funktionsbausteine haben zwei Fehlerausgänge um Fehler während der Kommandoausführung anzuzeigen.
- *Error* zeigt den Fehler an und *ErrorID* gibt eine ergänzende Fehlernummer aus.
- Die Ausgänge *Done* und *InVelocity*, bezeichnen eine erfolgreiche Kommandoausführung und werden nicht gesetzt, wenn *Error* TRUE wird.

Fehlertypen

- Funktionsbausteinfehler
 - Funktionsbausteinfehler sind Fehler, die ausschließlich den Funktionsbaustein und nicht die Achse betreffen wie z.B. fehlerhafte Parametrierung.
 - Funktionsbausteinfehler müssen nicht explizit zurückgesetzt werden, sondern werden selbständig zurückgesetzt, wenn der Eingang *Execute* zurückgesetzt wird.
- Kommunikationsfehler
 - Kommunikationsfehler wie z.B. der Funktionsbaustein kann die Achse nicht adressieren.
 - Kommunikationsfehler deuten oft auf eine fehlerhafte Konfiguration oder Parametrierung hin.
 - Ein Reset ist nicht möglich, sondern der Funktionsbaustein kann neu getriggert werden, nachdem die Konfiguration korrigiert wurde.
- Achsfehler
 - Achsfehler treten üblicherweise während der Fahrt auf wie z.B. Schleppabstandsfehler.
 - Ein Achsfehler muss durch MC_Reset zurückgesetzt werden.

Verhalten des *Done*-Ausgangs

- Der *Done*-Ausgang wird gesetzt, wenn ein Kommando erfolgreich ausgeführt wurde.
- Wenn mit mehreren Funktionsbausteinen an einer Achse gearbeitet wird und das laufende Kommando durch einen weiteren Baustein unterbrochen wird, so wird der *Done*-Ausgang des ersten Bausteins nicht gesetzt.

Verhalten des *CommandAborted*-Ausgangs

- *CommandAborted* wird gesetzt, wenn ein Kommando durch einen anderen Baustein unterbrochen wird.

Verhalten des *Busy*-Ausgangs

- Der *Busy*-Ausgang zeigt an, dass der Funktionsbaustein aktiv ist.
- *Busy* wird sofort mit Flanke 0-1 an *Execute* gesetzt und wird erst zurückgesetzt, wenn das Kommando erfolgreich oder auch nicht erfolgreich beendet wurde.
- Solange *Busy* TRUE ist, muss der Funktionsbaustein zyklisch aufgerufen werden, um das Kommando ausführen zu können.

Verhalten des *Active*-Ausgangs

- Wenn die Bewegung einer Achse durch mehrere Funktionsbausteine gesteuert wird, so zeigt der *Active*-Ausgang jedes Bausteins an, dass das Kommando durch die Achse ausgeführt wird.

***Enable*-Eingang und *Valid*-Ausgang**

- Im Gegensatz zu *Execute* führt der *Enable*-Eingang dazu, dass eine Aktion permanent und wiederholt ausgeführt wird, solange *Enable* TRUE ist. MC_ReadStatus aktualisiert beispielsweise zyklisch den Zustand einer Achse solange *Enable* TRUE ist.
- Ein Funktionsbaustein mit einem *Enable*-Eingang zeigt durch den *Valid*-Ausgang an, dass die an den Ausgängen angezeigten Daten gültig sind. Die Daten können jedoch ständig aktualisiert werden während *Valid* TRUE ist.

BufferMode

- *BufferMode* wird nicht unterstützt.

9 ErrorID - Zusätzliche Fehlerinformationen

ErrorID	Beschreibung	Bemerkung
0x0000	Kein Fehler	
0x8y24	Fehler in Baustein-Parameter y, mit y: <ul style="list-style-type: none"> ■ 1: Fehler in PROTOKOLL ■ 2: Fehler in PARAMETER ■ 3: Fehler in BAUDRATE ■ 4: Fehler in CHARLENGTH ■ 5: Fehler in PARITY ■ 6: Fehler in STOPBITS ■ 7: Fehler in FLOWCONTROL (Parameter fehlt) 	VMC_ConfigMaster_RTU
0x8001	Unzulässiger Wert beim Parameter <i>Position</i> .	
0x8002	Unzulässiger Wert beim Parameter <i>Distance</i> .	
0x8003	Unzulässiger Wert beim Parameter <i>Velocity</i> .	
0x8004	Unzulässiger Wert beim Parameter <i>Acceleration</i> .	
0x8005	Unzulässiger Wert beim Parameter <i>Deceleration</i> .	
0x8007	Unzulässiger Wert beim Parameter <i>ContinuousUpdate</i> .	
0x8008	Unzulässiger Wert beim Parameter <i>BufferMode</i> .	
0x8009	Unzulässiger Wert beim Parameter <i>EnablePositive</i> .	
0x800A	Unzulässiger Wert beim Parameter <i>EnableNegative</i> .	
0x800B	Unzulässiger Wert beim Parameter <i>MasterOffset</i> .	
0x800C	Unzulässiger Wert beim Parameter <i>SlaveOffset</i> .	
0x800D	Unzulässiger Wert beim Parameter <i>MasterScaling</i> .	
0x800E	Unzulässiger Wert beim Parameter <i>SlaveScaling</i> .	
0x800F	Unzulässiger Wert beim Parameter <i>StartMode</i> .	
0x8010	Unzulässiger Wert beim Parameter <i>ActivationMode</i> .	
0x8011	Unzulässiger Wert beim Parameter <i>Source</i> .	
0x8012	Unzulässiger Wert beim Parameter <i>Direction</i> .	
0x8014	Unzulässiger Parameter der physikalischen Achse.	MC_ReadParameter
0x8015	Unzulässiger Index oder Subindex.	MC_ReadParameter
0x8016	Unzulässige Parameterlänge.	MC_ReadParameter
0x8017	Unzulässige LADDR.	MC_ReadParameter
0x8018	Unzulässiger Wert beim Parameter <i>RatioDenominator</i> .	MC_GearIn
0x8019	Unzulässiger Wert beim Parameter <i>RatioNumerator</i> .	MC_GearIn
0x801A	Parameternummer nicht bekannt.	MC_ReadParameter, MC_Write-Parameter
0x801B	Parameter kann nicht geschrieben werden, Parameter ist schreibgeschützt.	MC_WriteParameter
0x801C	Parameter Kommunikation mit unbekanntem Mode.	MC_Home, MC_WriteParameter

ErrorID	Beschreibung	Bemerkung
0x801D	Parameterkommunikation mit allgemeinem Fehler. Die Fehlerursache ist nicht näher beschrieben.	
0x801E	SDO-Parameterwert außerhalb des zulässigen Bereichs.	MC_Home, MC_WriteParameter
0x801F	Der Typ in ANY ist nicht BYTE.	Parameter lesen/schreiben
0x8020	Unterschiedliche Konfiguration der Anwendereinheiten in Cam und Master-Achse.	
0x8021	Unterschiedliche Konfiguration der Anwendereinheiten in Cam und Slave-Achse.	
0x8022	Auf der über LADDR angegebenen logischen Adresse gibt es kein PROFIBUS/PROFINET-Device, von dem konsistente Daten gelesen werden können.	Parameter lesen/schreiben
0x8023	Es wurde ein Zugriffsfehler beim Zugriff auf ein I/O-Geräte erkannt.	Parameter lesen/schreiben
0x8024	Slave-Fehler an externem DP-Slave.	Parameter lesen/schreiben
0x8025	Systemfehler an externem DP-Slave.	Parameter lesen/schreiben
0x8026	Systemfehler an externem DP-Slave.	Parameter lesen/schreiben
0x8027	Die Daten wurden noch nicht vom Modul gelesen.	Parameter lesen/schreiben
0x8028	Systemfehler an externem DP-Slave.	Parameter lesen/schreiben
0x8029	Schreibversuch auf eine Objekt, welches nur gelesen werden kann.	Parameter lesen/schreiben
0x802A	Leseversuch auf ein Objekt, welches nur geschrieben werden kann.	Parameter lesen/schreiben
0x802B	Nicht unterstützte Zugriff auf ein Objekt.	Parameter lesen/schreiben
0x802C	Falscher Datentyp.	Parameter lesen/schreiben
0x802D	Fehler im Geräteprofil.	Parameter lesen/schreiben
0x802E	Fehler Kommando-Typ.	Parameter lesen/schreiben
0x802F	Keine Systemressourcen verfügbar.	Parameter lesen/schreiben
0x8030	Unzulässiger Wert beim Parameter <i>Hardware</i> (1 = SLIO CP; 2 = VIPA CPU).	Modbus; Init
0x8031	Unzulässiger Wert beim Parameter <i>UnitId</i> .	Modbus; Init
0x8032	Unzulässiger Wert beim Parameter <i>UserUnitsVelocity</i> (0 = Hz, 1 = %, 2 = U/min).	Modbus; Init
0x8033	Unzulässiger Wert beim Parameter <i>UserUnitsAcceleration</i> (0 = 0.00s, 1 = 0.0s).	Modbus; Init
0x8034	Unzulässiger Wert beim Parameter <i>MaxVelocityApp</i> (muss > 0 sein).	Modbus; Init
0x8035	Fehler beim Lesezugriff auf <i>MonitorData</i> .	Modbus; Init
0x8036	Fehler beim Lesezugriff auf <i>NumberOfPoles</i> .	Modbus; Init
0x8037	Fehler beim Schreibzugriff auf <i>UserUnitsVelocity</i> .	Modbus; Init
0x8038	Fehler beim Lesezugriff auf <i>MinOutputFrequency</i> .	Modbus; Init
0x8039	Fehler beim Lesezugriff auf <i>MaxOutputFrequency</i> .	Modbus; Init
0x803A	Fehler beim Schreibzugriff auf <i>StoppingMethodSelection</i> .	Modbus; Init
0x803B	Fehler beim Schreibzugriff auf <i>UserUnitsAcceleration</i> .	Modbus; Init

ErrorID	Beschreibung	Bemerkung
0x8041	Unzulässiger Wert beim Parameter <i>AccelerationTime</i> .	Modbus V1000
0x8042	Unzulässiger Wert beim Parameter <i>DecelerationTime</i> .	Modbus V1000
0x8043	Unzulässiger Wert beim Parameter <i>JogAccelerationTime</i> .	Modbus V1000
0x8044	Unzulässiger Wert beim Parameter <i>JogDecelerationTime</i> .	Modbus V1000
0x8045	Unzulässiger Wert beim Parameter <i>JogVelocity</i> ($\leq \text{MaxVelocityApp}$).	Modbus V1000
0x80C8	Modbus Übertragungsfehler: Keine Antwort des Servers im definierten Zeitraum (Timeout über Schnittstelle parametrierbar).	Modbus V1000
0x809y	Fehler in Wert des Baustein-Parameter y, mit y: <ul style="list-style-type: none"> ■ 1: Fehler in PROTOKOLL ■ 3: Fehler in BAUDRATE ■ 4: Fehler in CHARLENGTH ■ 5: Fehler in PARITY ■ 6: Fehler in STOPBITS 	VMC_ConfigMaster_RTU
0x8092	Zugriffsfehler auf Parameter-DB (DB zu kurz).	VMC_ConfigMaster_RTU
0x809A	Schnittstelle nicht vorhanden bzw. wird unter PROFIBUS betrieben.	VMC_ConfigMaster_RTU
0x8101	Keine zyklische Kommunikation mit der Achse möglich.	
0x8102	Befehl ist im aktuellen PLCopen-State nicht zulässig.	
0x8103	Befehl wird von der Achse nicht unterstützt.	
0x8104	Achse ist nicht einschaltbereit, mögliche Gründe: <ul style="list-style-type: none"> ■ Kommunikation zur Achse nicht bereit. ■ Antrieb ist nicht im Zustand <i>"eingeschaltet"</i> → Antriebsfehler evtl. mit MC_Reset zurücksetzen. ■ Kommunikation wurde unterbrochen, z.B. durch Aus- Einschalten der CPU. Fehler mit MC_Reset zurücksetzen. 	<i>PreOperational</i> muss auch im Status <i>Operational</i> gesetzt werden.
0x8105	Kommando wird von virtuellen Achsen nicht unterstützt.	
0x8106	PLCopen-State ist nicht definiert.	
0x8107	Befehl ist bei deaktiviertem Antrieb nicht zulässig.	VMC_AxisControl_PT, ModbusV1000
0x8188	Modbus Übertragungsfehler: Interner Fehler MB_FUNCTION ungültig.	Modbus V1000
0x8189	Modbus Übertragungsfehler: Interner Fehler MB_DATA_ADDR ungültig.	Modbus V1000
0x818A	Modbus Übertragungsfehler: Interner Fehler MB_DATA_LEN ungültig.	Modbus V1000
0x818B	Modbus Übertragungsfehler: Interner Fehler MB_DATA_PTR ungültig.	Modbus V1000
0x8201	Wegen Mangels an internen Ressourcen kann der Befehl aktuell nicht ausgeführt werden (kein freier Slot im CommandBuffer).	
0x8202	Fehler beim Schreiben des Offsets für Referenzfahrt (kein freier Slot im CommandBuffer).	DriveManager → Referenzfahrt (aktives Kommando)

ErrorID	Beschreibung	Bemerkung
0x8210	Modbus Übertragungsfehler: Die Hardware ist inkompatibel zur Baustein-Bibliothek Modbus RTU/TCP.	Modbus V1000
0x828y	Fehler in Parameter y von DB-Parameter, mit y: <ul style="list-style-type: none"> ■ 1: Fehler im 1. Parameter ■ 2: Fehler im 2. Parameter ■ ... 	VMC_ConfigMaster_RTU
0x8301	Keine zyklische Kommunikation mit der Master-Achse möglich.	
0x8302	Befehl ist im aktuellen PLCopen-State der Master-Achse nicht zulässig.	
0x8303	Befehl wird von der Master-Achse nicht unterstützt.	
0x8304	Master-Achse befindet sich nicht im Status <i>Pre-Operational</i> .	
0x8305	Die Nummer des Datenbausteins der Master-Achse hat sich geändert.	
0x8306	Kommunikationsfehler an der Master Achse. Slave Achse wird mit Schnellhalt gestoppt.	
0x8311	Keine zyklische Kommunikation mit der Slave-Achse möglich.	
0x8312	Befehl ist im aktuellen PLCopen-State der Slave-Achse nicht zulässig.	
0x8313	Befehl wird von der Slave-Achse nicht unterstützt.	
0x8314	Slave-Achse befindet sich nicht im Status <i>Pre-Operational</i> .	
0x8315	Die Nummer des Datenbausteins der Slave-Achse hat sich geändert.	
0x8321	Koppeln mit <i>StartMode</i> = relative und <i>ActivationMode</i> = nextcycle ist nicht zulässig.	
0x8322	Koppeln oder schalten mit <i>StartMode</i> = absolute und <i>ActivationMode</i> = nextcycle ist nicht zulässig.	
0x8323	Schalten mit einem unterschiedlichen <i>StartMode</i> (<i>StartMode</i> der Kopplung ist zu verwenden).	
0x8331	MC_CamIn ist nicht aktiv.	
0x8332	MC_GearIn ist nicht aktiv.	
0x8340	Ungültiger Wert in TriggerInput.Probe.	MC_TouchProbe und MC_AbortTrigger
0x8341	Ungültiger Wert in TriggerInput.Source.	MC_TouchProbe und MC_AbortTrigger
0x8342	Ungültiger Wert in TriggerInput.TriggerMode.	MC_TouchProbe und MC_AbortTrigger
0x8350	Ungültiger Wert in VelocitySearchSwitch.	Referenzfahrt, Initialisierung
0x8351	Ungültiger Wert in VelocitySearchZero.	Referenzfahrt, Initialisierung
0x8352	Ungültige Kombination von Eingängen.	Referenzfahrt, Initialisierung
0x8360	Die CPU unterstützt kein Pulse Train.	VMC_AxisControl_PT

ErrorID	Beschreibung	Bemerkung
0x8361	Falscher Wert in <i>S_ChannelNumberPWM</i> .	VMC_AxisControl_PT
0x8362	Allgemeiner Fehler bei der Pulse Train Ausgabe.	VMC_AxisControl_PT
0x8363	Fahr-Kommando erhalten bei gesetztem <i>StopExecute</i> .	VMC_AxisControl_PT, ModbusV1000
0x8381	Modbus Übertragungsfehler: Server liefert Exception-Code 01h.	Modbus V1000
0x8382	Modbus Übertragungsfehler: Server liefert Exception-Code 03h oder falsche Startadresse.	Modbus V1000
0x8383	Modbus Übertragungsfehler: Server liefert Exception-Code 02h.	Modbus V1000
0x8384	Modbus Übertragungsfehler: Server liefert Exception-Code 04h.	Modbus V1000
0x8386	Modbus Übertragungsfehler: Server liefert falschen Funktions-Code.	Modbus V1000
0x8388	Modbus Übertragungsfehler: Server liefert falschen Wert oder falsche Anzahl.	Modbus V1000
0x8400	MC_Power: Unerwarteter Drive-State Drive-State <> Operation enabled	MC_Power
0x8401	MC_Power: Unerwarteter Drive-State Drive-State = Quick stop active	MC_Power
0x8402	MC_Power: Unerwarteter Drive-State Drive-State = Fault reaction active	MC_Power
0x8403	MC_Power: Unerwarteter Drive-State Drive-State = Fault	MC_Power
0x8410	Zeitüberschreitung beim Versuch den Antrieb zurückzusetzen.	Kernel FB --> MC_Reset
0x8500	Falscher Wert in <i>EncoderType</i> (1 oder 2).	Init-Baustein
0x8501	Falscher Wert in <i>EncoderResolutionBits</i> (>0 und ≤32).	Init-Baustein
0x8502	Falscher Wert in <i>LogicalAddress</i> (≥0).	Init-Baustein
0x8503	Falscher Wert in <i>StartInputAddress</i> (≥0).	Init-Baustein
0x8504	Falscher Wert in <i>StartOutputAddress</i> (≥0).	Init-Baustein
0x8505	Falscher Wert in <i>FactorPosition</i> (>0.0).	Init-Baustein
0x8506	Falscher Wert in <i>FactorVelocity</i> (>0.0).	Init-Baustein
0x8507	Falscher Wert in <i>FactorAcceleration</i> (>0.0).	Init-Baustein
0x8508	Falscher Wert in <i>MaxVelocityApp</i> (>0.0).	Init-Baustein
0x8509	Falscher Wert in <i>MaxAccelerationApp</i> (>0.0).	Init-Baustein
0x850A	Falscher Wert in <i>MaxDecelerationApp</i> (>0.0).	Init-Baustein
0x850B	Falscher Wert in <i>MaxVelocityDrive</i> (>0.0).	Init-Baustein
0x850C	Falscher Wert in <i>MaxAccelerationDrive</i> (>0.0).	Init-Baustein
0x850D	Falscher Wert in <i>MaxDecelerationDrive</i> (>0.0).	Init-Baustein
0x850E	Falscher Wert in <i>MinPosition</i> (≥MinUserPos).	Init-Baustein
0x850F	Falscher Wert in <i>MaxPosition</i> (≥MaxUserPos).	Init-Baustein

ErrorID	Beschreibung	Bemerkung
0x8510	Falscher Wert in <i>M2_EncoderType</i> .	VMC_InitSigma7W_EC
0x8511	Falscher Wert in <i>M2_EncoderResolutionBits</i> .	VMC_InitSigma7W_EC
0x8513	Falscher Wert in <i>M2_PdoInputs</i> .	VMC_InitSigma7W_EC
0x8514	Falscher Wert in <i>M2_PdoOutputs</i> .	VMC_InitSigma7W_EC
0x8515	Falscher Wert in <i>M2_FactorPosition</i> .	VMC_InitSigma7W_EC
0x8516	Falscher Wert in <i>M2_FactorVelocity</i> .	VMC_InitSigma7W_EC
0x8517	Falscher Wert in <i>M2_FactorAcceleration</i> .	VMC_InitSigma7W_EC
0x8518	Falscher Wert in <i>M2_MaxVelocityApp</i> .	VMC_InitSigma7W_EC
0x8519	Falscher Wert in <i>M2_MaxAccelerationApp</i> .	VMC_InitSigma7W_EC
0x851A	Falscher Wert in <i>M2_MaxDecelerationApp</i> .	VMC_InitSigma7W_EC
0x8603	Fehler Referenzfahrt am Antrieb, Geschwindigkeit $\neq 0$.	MC_Home
0x8604	Fehler Referenzfahrt am Antrieb, Geschwindigkeit = 0.	MC_Home
0x8700	Fehler: Unzulässige Größe.	
0x8710	SDO-Fehler: Toggle-Bit hat nicht gewechselt.	
0x8711	SDO-Fehler: SDO-Protokoll Timeout.	
0x8712	SDO-Fehler: Client/Server-Kommando nicht gültig oder unbekannt.	
0x8713	SDO-Fehler: Unzulässige Blockgröße (nur im Blockmodus).	
0x8714	SDO-Fehler: Unzulässige Sequenznummer (nur im Blockmodus).	
0x8715	SDO-Fehler: CRC-Fehler (nur im Block-Modus).	
0x8716	SDO-Fehler: Nicht genügend Arbeitsspeicher.	
0x8717	SDO-Fehler: Nicht unterstützter Zugriff auf ein Objekt.	
0x8718	SDO-Fehler: Leseversuch auf ein Objekt, welches nur geschrieben werden kann.	
0x8719	SDO-Fehler: Schreibversuch auf ein Objekt, welches nur gelesen werden kann.	
0x871A	SDO-Fehler: Objekt existiert nicht im Objektverzeichnis.	
0x871B	SDO-Fehler: Objekt kann nicht auf ein PDO gemappt werden.	
0x871C	SDO-Fehler: Anzahl und Länge der zu mappenden Objekte übersteigt die PDO-Länge.	
0x871D	SDO-Fehler: Allgemeine Parameter-Inkompatibilität.	
0x871E	SDO-Fehler: Allgemeine interne Inkompatibilität im Gerät.	
0x871F	SDO-Fehler: Zugriff fehlgeschlagen aufgrund eines Hardwarefehlers.	
0x8720	SDO-Fehler: Datentyp passt nicht, Länge Service- Parameter passt nicht.	
0x8721	SDO-Fehler: Datentyp passt nicht, Service-Parameter ist zu lang.	
0x8722	SDO-Fehler: Datentyp passt nicht, Service-Parameter ist zu kurz.	
0x8723	SDO-Fehler: Es existiert kein Subindex.	

ErrorID	Beschreibung	Bemerkung
0x8724	SDO-Fehler: Schreibzugriff - Parameterwert liegt außerhalb des zulässigen Bereichs.	
0x8725	SDO-Fehler: Schreibzugriff - Parameterwert überschreitet den zulässigen Bereich.	
0x8726	SDO-Fehler: Schreibzugriff - Parameterwert unterschreitet den zulässigen Bereich.	
0x8727	SDO-Fehler: Maximaler Wert < Minimaler Wert.	
0x8728	SDO-Fehler: Allgemeiner Fehler.	
0x8729	SDO-Fehler: Daten können nicht in die Anwendung übertragen oder dort gespeichert werden.	
0x872A	SDO-Fehler: Daten können nicht in die Anwendung übertragen oder dort gespeichert werden, da die lokale Steuerung aktiviert ist.	
0x872B	SDO-Fehler: Aufgrund des aktuellen Gerätezustands können keine Daten in die Anwendung übertragen oder dort gespeichert werden.	
0x872C	SDO-Fehler: Die dynamische Generierung des Objektverzeichnisses konnte nicht durchgeführt werden bzw. das Objektverzeichnis ist nicht vorhanden.	
0x872D	SDO-Fehler: Unbekannter Code.	
0x8750	Falscher Wert in <i>LADDR</i> .	
0x8751	Typ in ANY-Pointer weicht von BYTE ab.	
0x8752	Auf der über <i>LADDR</i> spezifizierten Adresse gibt es kein PRO-FIBUS-DP-Modul bzw. PROFINET-IO-Device, von dem konsistente Daten gelesen werden können.	
0x8753	Zugriffsfehler beim Zugriff auf ein PROFINET-IO-Device.	
0x8754	Slave-Fehler am externen PROFIBUS-DP-Slave.	
0x8755	Länge der SFB-Daten passt nicht zur Länge der Benutzerdaten.	
0x8756	Fehler auf externem PROFIBUS-DP-Slave.	
0x8757	Systemfehler auf externem PROFIBUS-DP-Slave.	
0x8758	Die Daten wurden vom Gerät noch nicht gelesen.	
0x8759	Systemfehler auf externem PROFIBUS-DP-Slave.	
0x875A	Es sind keine Systemressourcen verfügbar.	
0x8799	SDO-Fehler: Es ist ein anderer Fehler aufgetreten, nähere Informationen finden Sie in den Daten von <i>Info1</i> und <i>Info2</i> .	
0x8888	Intern: BufferIndex-Fehler	VMC_AxisControl
0xC000	Interner Fehler: Status Init ist undefiniert.	Modbus; Init
0xC001	Interner Fehler: Unzulässiger Wert beim Parameter <i>Cmd.Active-Type</i> .	Modbus V1000
0xC002	Internal Error: Unzulässiger Wert beim Parameter <i>Cmd.State</i> .	Modbus V1000