

VIPA Library

OPL-LIB | SW90MS0MA | Handbuch

HB00 | OPL-LIB | SW90MS0MA | de | 17-21

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 |
| 2.1 | Simple Motion Control..... | 8 |
| 2.2 | Funktionsprinzip..... | 9 |
| 3 | Einsatz <i>Sigma-5</i> EtherCAT..... | 10 |
| 3.1 | Übersicht..... | 10 |
| 3.2 | Parameter am Antrieb einstellen..... | 10 |
| 3.3 | Einsatz im VIPA <i>SPEED7 Studio</i> | 11 |
| 3.3.1 | Hardware-Konfiguration..... | 11 |
| 3.3.2 | Anwender-Programm..... | 23 |
| 3.4 | Einsatz im Siemens SIMATIC Manager..... | 27 |
| 3.4.1 | Voraussetzung..... | 27 |
| 3.4.2 | Hardware-Konfiguration..... | 29 |
| 3.4.3 | Anwender-Programm..... | 38 |
| 3.4.4 | Projekt kopieren..... | 43 |
| 3.5 | Antriebsspezifische Bausteine..... | 45 |
| 3.5.1 | UDT 870 - VMC_ConfigSigma5EC_REF - <i>Sigma-5</i> EtherCAT Datenstruktur Achskonfiguration..... | 45 |
| 3.5.2 | FB 870 - VMC_KernelSigma5_EC - <i>Sigma-5</i> EtherCAT Kernel..... | 45 |
| 3.5.3 | FB 871 - VMC_InitSigma5_EC - <i>Sigma-5</i> EtherCAT Initialisierung..... | 45 |
| 4 | Einsatz <i>Sigma-7S</i> EtherCAT..... | 48 |
| 4.1 | Übersicht..... | 48 |
| 4.2 | Parameter am Antrieb einstellen..... | 48 |
| 4.3 | Einsatz im VIPA <i>SPEED7 Studio</i> | 49 |
| 4.3.1 | Hardware-Konfiguration..... | 49 |
| 4.3.2 | Anwender-Programm..... | 60 |
| 4.4 | Einsatz im Siemens SIMATIC Manager..... | 65 |
| 4.4.1 | Voraussetzung..... | 65 |
| 4.4.2 | Hardware-Konfiguration..... | 67 |
| 4.4.3 | Anwender-Programm..... | 77 |
| 4.4.4 | Projekt kopieren..... | 82 |
| 4.5 | Antriebsspezifische Bausteine..... | 84 |
| 4.5.1 | UDT 872 - VMC_ConfigSigma7EC_REF - <i>Sigma-7</i> EtherCAT Datenstruktur Achskonfiguration..... | 84 |
| 4.5.2 | FB 872 - VMC_KernelSigma7_EC - <i>Sigma-7</i> EtherCAT Kernel..... | 84 |
| 4.5.3 | FB 873 - VMC_InitSigma7S_EC - <i>Sigma-7S</i> EtherCAT Initialisierung..... | 84 |
| 5 | Einsatz <i>Sigma-7W</i> EtherCAT..... | 87 |
| 5.1 | Übersicht..... | 87 |
| 5.2 | Parameter am Antrieb einstellen..... | 87 |
| 5.3 | Einsatz im VIPA <i>SPEED7 Studio</i> | 88 |
| 5.3.1 | Hardware-Konfiguration..... | 88 |
| 5.3.2 | Anwender-Programm..... | 99 |
| 5.4 | Einsatz im Siemens SIMATIC Manager..... | 106 |
| 5.4.1 | Voraussetzung..... | 106 |
| 5.4.2 | Hardware-Konfiguration..... | 108 |

| | | |
|----------|---|------------|
| 5.4.3 | Anwender-Programm..... | 118 |
| 5.4.4 | Projekt kopieren..... | 124 |
| 5.5 | Antriebsspezifische Bausteine..... | 126 |
| 5.5.1 | UDT 872 - VMC_ConfigSigma7EC_REF - <i>Sigma-7</i> EtherCAT Datenstruktur Achskonfiguration..... | 126 |
| 5.5.2 | FB 872 - VMC_KernelSigma7_EC - <i>Sigma-7</i> EtherCAT Kernel..... | 126 |
| 5.5.3 | FB 874 - VMC_InitSigma7W_EC - <i>Sigma-7W</i> EtherCAT Initialisierung.... | 126 |
| 6 | Bausteine zur Achskontrolle..... | 130 |
| 6.1 | Übersicht..... | 130 |
| 6.2 | Einfache Bewegungsaufgaben..... | 132 |
| 6.2.1 | UDT 860 - MC_AXIS_REF - Datenstruktur Achsdaten..... | 132 |
| 6.2.2 | FB 860 VMC_AxisControl - Control-Baustein Achskontrolle..... | 132 |
| 6.3 | Komplexe Bewegungsaufgaben - PLCopen-Bausteine..... | 136 |
| 6.3.1 | UDT 860 - MC_AXIS_REF - Datenstruktur Achsdaten..... | 136 |
| 6.3.2 | UDT 861 - MC_TRIGGER_REF - Datenstruktur Triggersignal..... | 136 |
| 6.3.3 | FB 800 - MC_Power - Achsenfreigabe..... | 137 |
| 6.3.4 | FB 801 - MC_Home - Achse referenzieren..... | 139 |
| 6.3.5 | FB 802 - MC_Stop - Achse stoppen..... | 141 |
| 6.3.6 | FB 803 - MC_Halt - Achse anhalten..... | 143 |
| 6.3.7 | FB 804 - MC_MoveRelative - Achse relativ verfahren..... | 145 |
| 6.3.8 | FB 805 - MC_MoveVelocity - Achse verfahren mit konstanter Geschwindigkeit..... | 147 |
| 6.3.9 | FB 808 - MC_MoveAbsolute - Achse auf absolute Position verfahren..... | 149 |
| 6.3.10 | FB 811 - MC_Reset - Reset Achse..... | 151 |
| 6.3.11 | FB 812 - MC_ReadStatus - PLCopen Status..... | 153 |
| 6.3.12 | FB 813 - MC_ReadAxisError - Fehler von Achse lesen..... | 155 |
| 6.3.13 | FB 814 - MC_ReadParameter - Parameter der Achse lesen..... | 157 |
| 6.3.14 | FB 815 - MC_WriteParameter - Parameter an Achse schreiben..... | 159 |
| 6.3.15 | FB 816 - MC_ReadActualPosition - Aktuelle Position der Achse lesen.. | 161 |
| 6.3.16 | FB 817 - MC_ReadActualVelocity - Aktuelle Geschwindigkeit der Achse lesen..... | 162 |
| 6.3.17 | FB 818 - MC_ReadAxisInfo - Zusatzinformationen der Achse lesen..... | 163 |
| 6.3.18 | FB 819 - MC_ReadMotionState - Zustand Bewegungsauftrag lesen..... | 165 |
| 6.3.19 | FB 823 - MC_TouchProbe - Achsposition erfassen..... | 167 |
| 6.3.20 | FB 824 - MC_AbortTrigger - Achsposition erfassen abbrechen..... | 169 |
| 6.3.21 | FB 825 - MC_ReadBoolParameter - Boolean-Parameter von Achse lesen..... | 170 |
| 6.3.22 | FB 826 - MC_WriteBoolParameter - Boolean-Parameter an Achse schreiben..... | 172 |
| 6.3.23 | FB 827 - VMC_ReadDWordParameter - Doppelwort-Parameter von Achse lesen..... | 174 |
| 6.3.24 | FB 828 - VMC_WriteDWordParameter - Doppelwort-Parameter an Achse schreiben..... | 176 |
| 6.3.25 | FB 829 - VMC_ReadWordParameter - Wort-Parameter von Achse lesen..... | 178 |
| 6.3.26 | FB 830 - VMC_WriteWordParameter - Wort-Parameter an Achse schreiben..... | 180 |
| 6.3.27 | FB 831 - VMC_ReadByteParameter - Byte-Parameter von Achse lesen..... | 182 |
| 6.3.28 | FB 832 - VMC_WriteByteParameter - Byte-Parameter an Achse schreiben..... | 184 |

| | | |
|----------|---|------------|
| 6.3.29 | FB 835 - VMC_HomeInit_LimitSwitch - Initialisierung Referenzfahrt auf Endschalter..... | 186 |
| 6.3.30 | FB 836 - VMC_HomeInit_HomeSwitch - Initialisierung Referenzfahrt auf Referenzschalter..... | 188 |
| 6.3.31 | FB 837 - VMC_HomeInit_ZeroPulse - Initialisierung Referenzfahrt auf Null Impuls..... | 191 |
| 6.3.32 | FB 838 - VMC_HomeInit_SetPosition - Initialisierung Referenzfahrt auf aktuelle Position..... | 193 |
| 6.3.33 | PLCopen Parameter..... | 193 |
| 6.3.34 | VIPA-spezifische Parameter..... | 195 |
| 7 | Zustände und Verhalten der Ausgänge..... | 197 |
| 7.1 | Zustände..... | 197 |
| 7.2 | Ablöseverhalten von Bewegungsaufträgen..... | 199 |
| 7.3 | Verhalten der Ein- und Ausgänge..... | 201 |
| 8 | ErrorID - Zusätzliche Fehlerinformationen..... | 203 |

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 *Simple Motion Control Library* von VIPA:

- Beschrieben wird Aufbau, Projektierung und Anwendung in verschiedenen Programmersystemen.
- 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

2.1 Simple Motion Control

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:

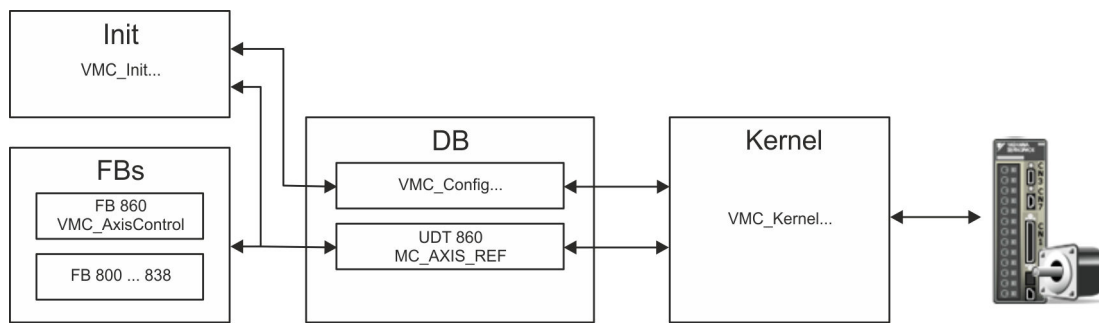
- Einsetzbar im VIPA *SPEED7 Studio* und Siemens SIMATIC Manager
- Umsetzung von einfachen Antriebsfunktionen
 - Einschalten bzw. Ausschalten
 - Drehzahlvorgabe
 - Relative bzw. absolute Positionierung
 - Referenzfahrt (Homing)
 - Lesen und Schreiben von Parametern
 - Abfrage von Achsposition und Status
- 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.
- *Sigma-5* EtherCAT
 - Spezifische Bausteine für den Einsatz von *Sigma-5*-Antrieben, welche über EtherCAT angebunden sind.
- *Sigma-7S* EtherCAT
 - Spezifische Bausteine für den Einsatz von *Sigma-7*-Antrieben, welche über EtherCAT angebunden sind.
- *Sigma-7W* EtherCAT
 - Spezifische Bausteine für den Einsatz von *Sigma-7W*-Antrieben, welche über EtherCAT angebunden sind.

2.2 Funktionsprinzip



- **DB**
 - Für jede Achse eines Antriebs ist ein Datenbaustein (Achs-DB) für Konfiguration und Statusdaten anzulegen.
 - Zur Typ-Deklaration innerhalb des DB gibt es für jeden Antrieb eine gesonderte Datenstruktur (VMC_Config...) in der *Simple Motion Control Library*.
- **Init**
 - Der *Init*-Baustein dient zur Konfiguration einer Achse.
 - Die Konfigurationsdaten für die Initialisierung sind im *Achs-DB* abzulegen.
 - Für jeden Antrieb gibt es in der *Simple Motion Control Library* einen gesonderten *Init*-Baustein (VMC_Init...).
- **Kernel**
 - Der *Kernel*-Baustein kommuniziert mit dem Antrieb über das entsprechende Bussystem, verarbeitet die Benutzeraufträge und liefert Statusmeldungen zurück.
 - Der Austausch der Daten erfolgt mittels des *Achs-DB*.
 - Für jeden Antrieb gibt es in der *Simple Motion Control Library* einen gesonderten *Kernel*-Baustein (VMC_Kernel...).
- **FBs**
 - Allgemeine Bausteine für alle Antriebe und Bussysteme.
 - Der Austausch der Daten erfolgt mittels des *Achs-DB*.
 - Der FB 860 - VMC_AxisControl ist ein Universalbaustein für einfache Bewegungskommandos und Statusabfragen.
 - Die Bausteine FB 800 ... FB 838 sind PLCopen-Bausteine zur Programmierung von komplexen Bewegungsabläufen und Statusabfragen.

3 Einsatz *Sigma-5* EtherCAT

3.1 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-CEFNR00
- *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.

3.2 Parameter am Antrieb einstellen



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-5 (20Bit Encoder)

| Servopack Parameter | Adresse | Name | Wert |
|---------------------|------------|--------------------------------------|-------|
| Pn205 | (2205h) | Multiturn Limit Setting | 65535 |
| Pn20E | (220Eh) | ElectronicGear 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 |
| PnB0A | (2703h:01) | Acceleration User Unit (Numerator) | 1 |
| PnB0C | (2703h:02) | Acceleration User Unit (Denominator) | 1 |

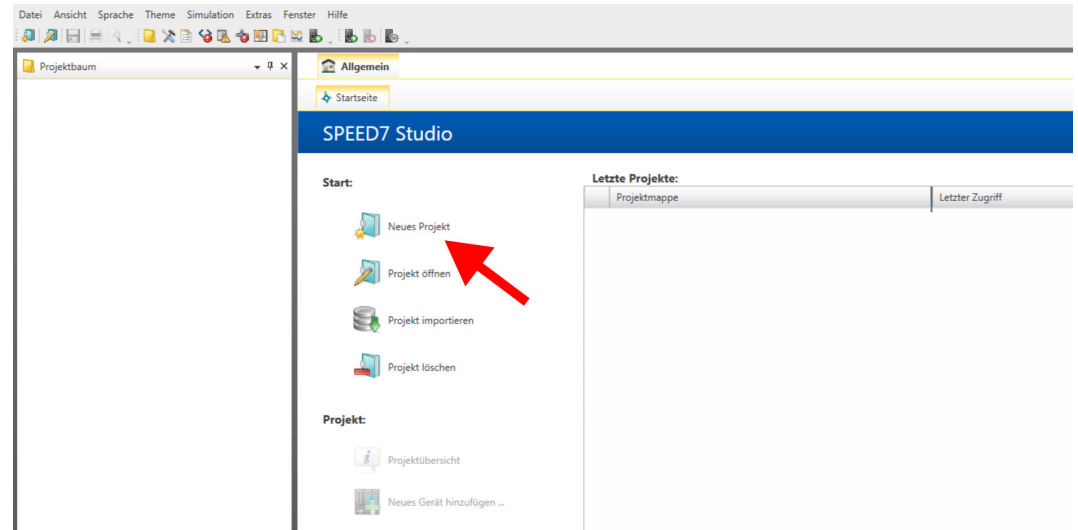
3.3 Einsatz im VIPA SPEED7 Studio

3.3.1 Hardware-Konfiguration

CPU im Projekt anlegen

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

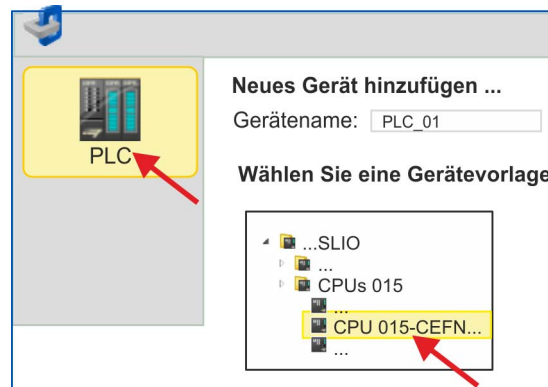
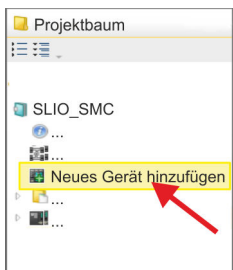
1. Starten Sie das *SPEED7 Studio*.



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 ..."*.



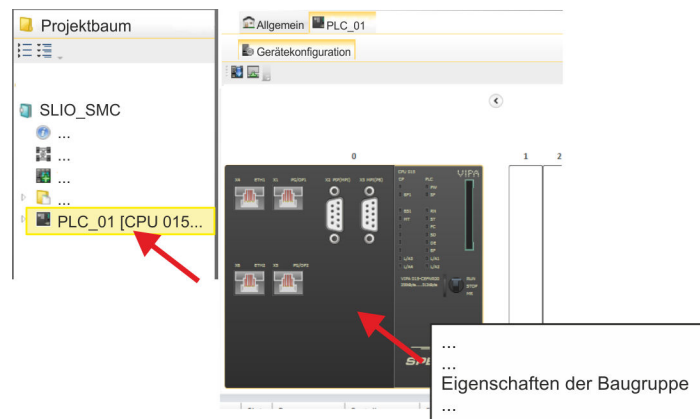
⇒ Es öffnet sich ein Dialog für die Geräteauswahl.

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.

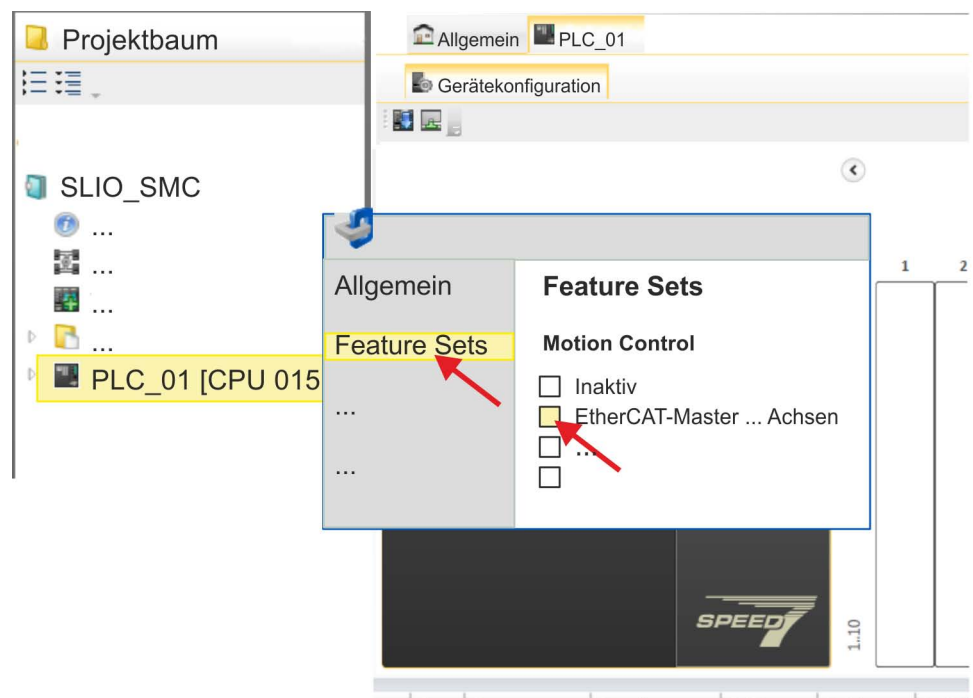
Motion-Control-Funktionen aktivieren

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].

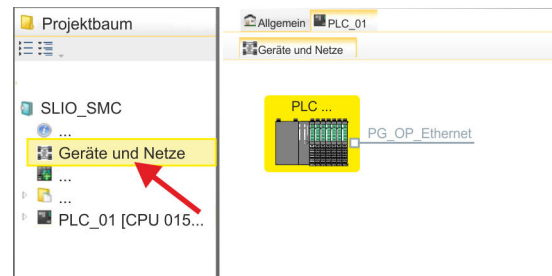
⇒ Die Motion-Control-Funktionen stehen Ihnen nun in Ihrem Projekt zur Verfügung.

**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.

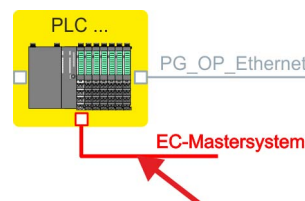
ESI-Datei installieren

Damit der Sigma-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*".

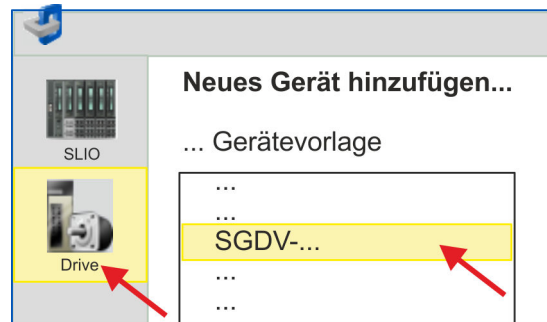
1. ➤ Laden Sie die zu Ihrem Antrieb passende ESI-Datei herunter. Entpacken Sie diese falls erforderlich.
2. ➤ Gehen Sie in Ihr *SPEED7 Studio*.
3. ➤ Öffnen Sie mit "*Extras → Gerätebeschreibungsdatei installieren (EtherCAT - ESI)*" das zugehörige Dialogfenster.
4. ➤ Geben Sie unter "*Quellpfad*" die 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*".



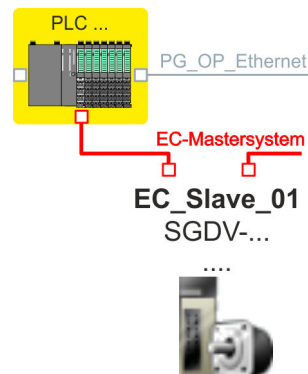
- ⇒ Es öffnet sich die Gerätevorlage zur Auswahl eines EtherCAT-Devices.



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

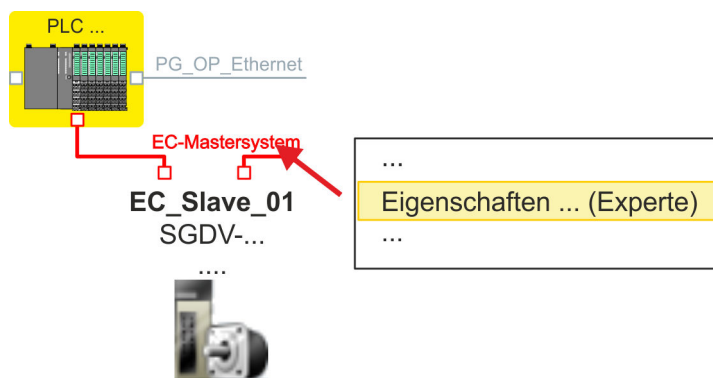
- SGDV-xxxxE5...
- SGDV-xxxxE1...

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



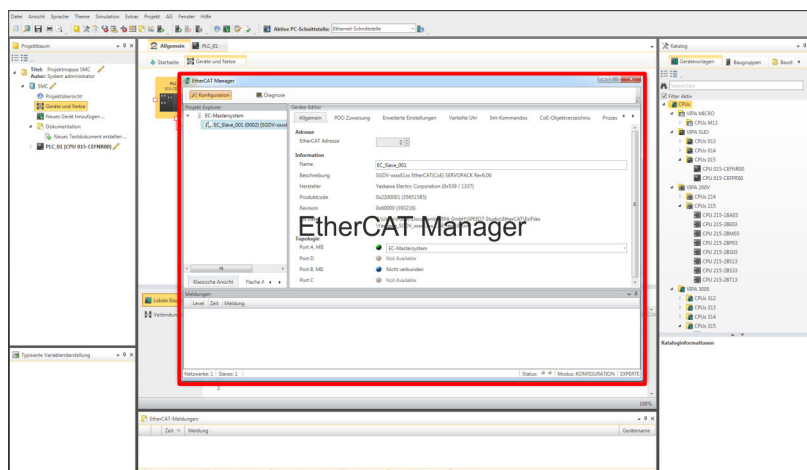
1. Klicken Sie auf "EC-Mastersystem" und wählen sie "Kontextmenü → Eigenschaft des Busystems (Experte)".



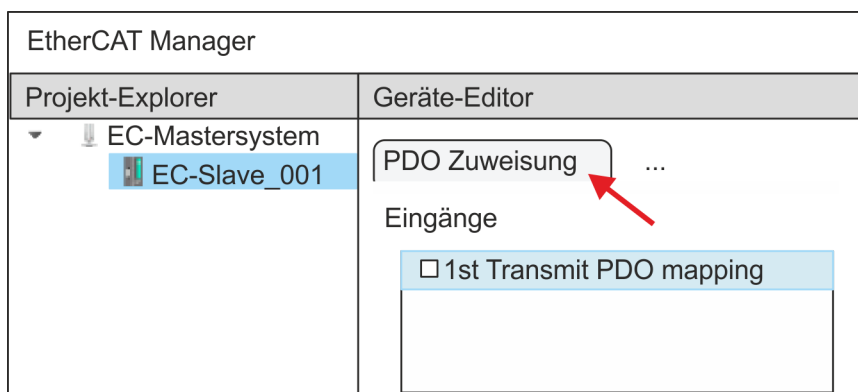
PDOs können Sie nur im "Experten-Modus" bearbeiten! Ansonsten werden die Schaltflächen ausgeblendet.

- ⇒ Der SPEED7 EtherCAT Manager wird gestartet. Hier können Sie die EtherCAT-Kommunikation zu Ihrem Sigma-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' 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

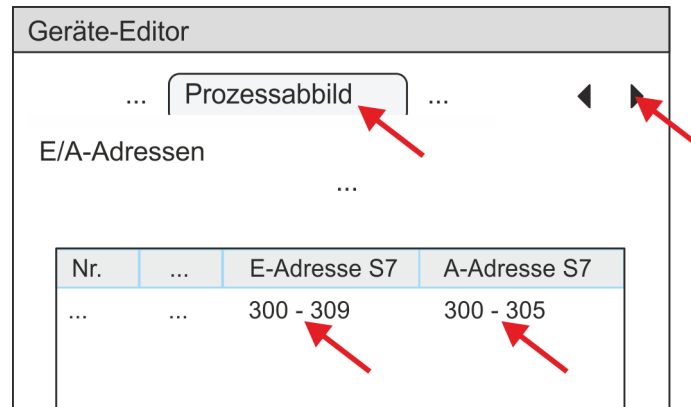
Verteilte Uhren ...

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 871 - VMC_InitSigma5_EC folgende PDO-Anfangsadressen:

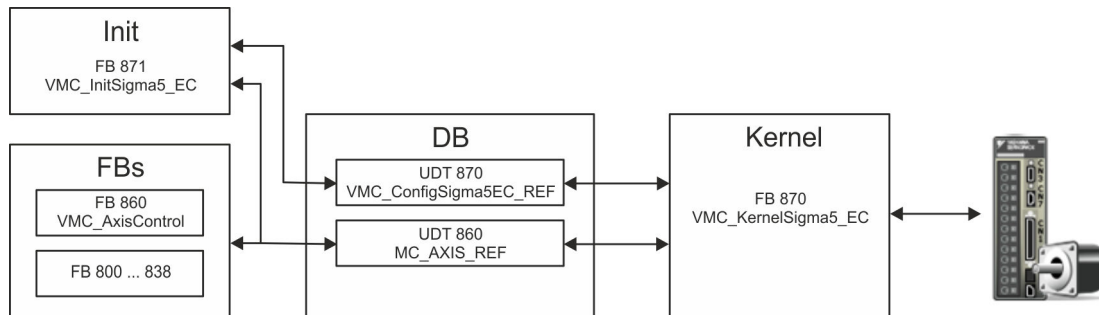
- "E-Adresse S7" → "InputsStartAddressPDO"
- "A-Adresse S7" → "OutputsStartAddressPDO"



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

3.3.2 Anwender-Programm

3.3.2.1 Programmstruktur



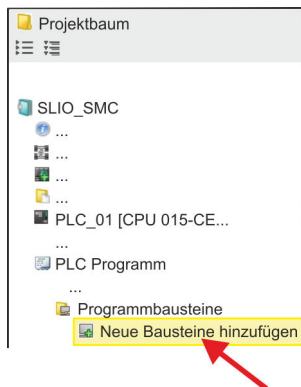
- **DB**

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

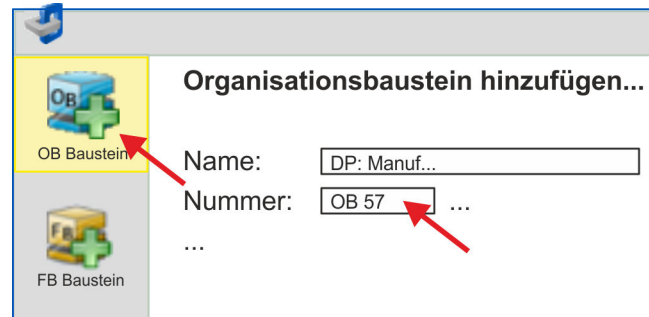
 - **UDT 870 - VMC_ConfigSigma5EC_REF**
Die Datenstruktur beschreibt den Aufbau der Konfiguration des Antriebs. Spezifische Datenstruktur für *Sigma-5* EtherCAT.
 - **UDT 860 - MC_AXIS_REF**
Die Datenstruktur beschreibt den Aufbau der Parameter und Statusinformationen von Antrieben. Allgemeine Datenstruktur 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.

3.3.2.2 Programmierung

Bausteine in Projekt kopieren

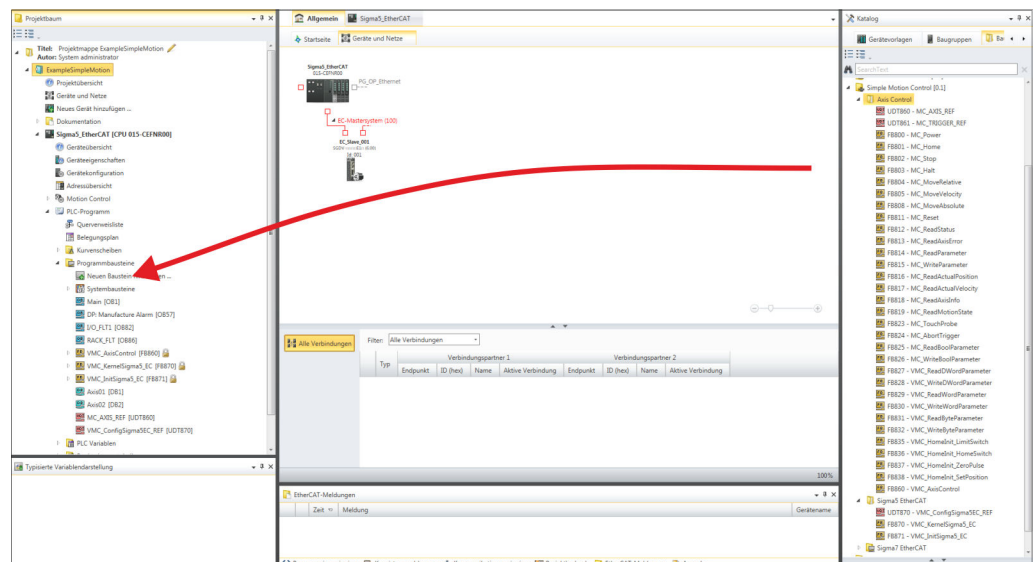


1. Klicken Sie im *Projektbaum* innerhalb der CPU unter "*PLC-Programm*", "*Programmbausteine*" auf "*Neuen Baustein hinzufügen*".



⇒ 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 3.5.3 "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. ↪ 22

```
⇒ 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 3.5.2 "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 6.2.2 "FB 860 VMC_AxisControl - Control-Baustein Achskontrolle" auf Seite 132*

```
⇒ CALL "VMC_AxisControl" , "DI_AxisControl01"
    SourceInputs      := "AxCtrl1_SourceInputs"
    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-Bausteinen 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.

3.4 Einsatz im Siemens SIMATIC Manager

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.

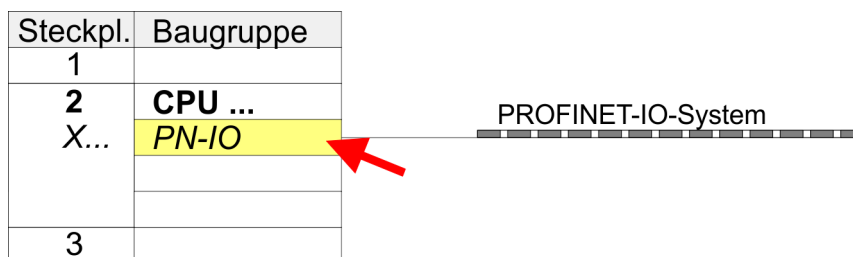
3.4.2 Hardware-Konfiguration

CPU im Projekt anlegen

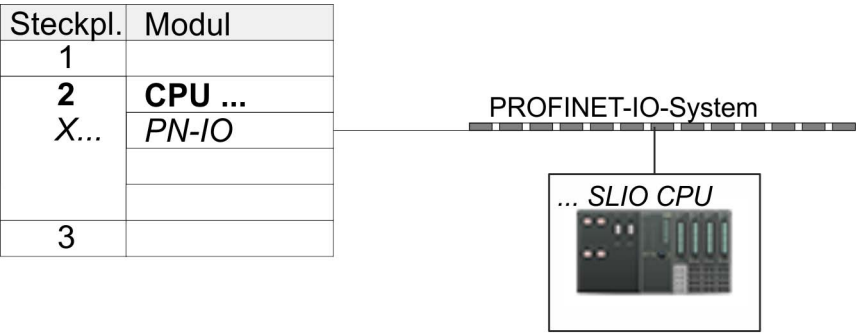
| Steckp.. | Baugruppe |
|----------|------------------------|
| 1 | |
| 2 | CPU 315-2 PN/DP |
| X1 | MPI/DP |
| X2 | PN-IO |
| X2... | Port 1 |
| X2... | Port 2 |
| 3 | |

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 | | | |
| ... | | | |

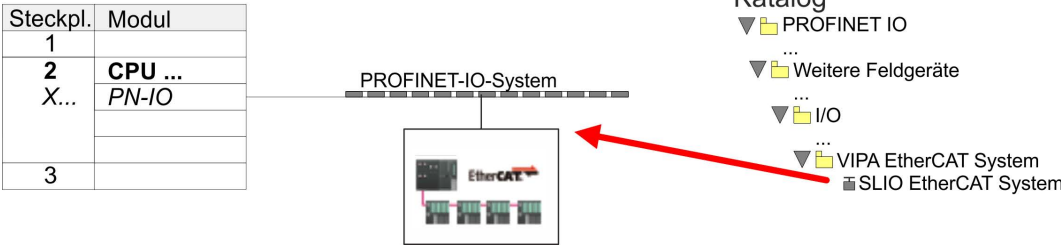
11. Navigieren Sie im Hardware-Katalog in das Verzeichnis *"PROFINET IO → Weitere Feldgeräte → I/O → VIPA SLIO System"* und binden Sie das IO-Device *"015-CEFNR00 CPU"* an Ihr PROFINET-System an.
- ⇒ In der Steckplatzübersicht des PROFINET-IO-Device *"VIPA SLIO CPU"* ist auf Steckplatz 0 die CPU bereits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

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

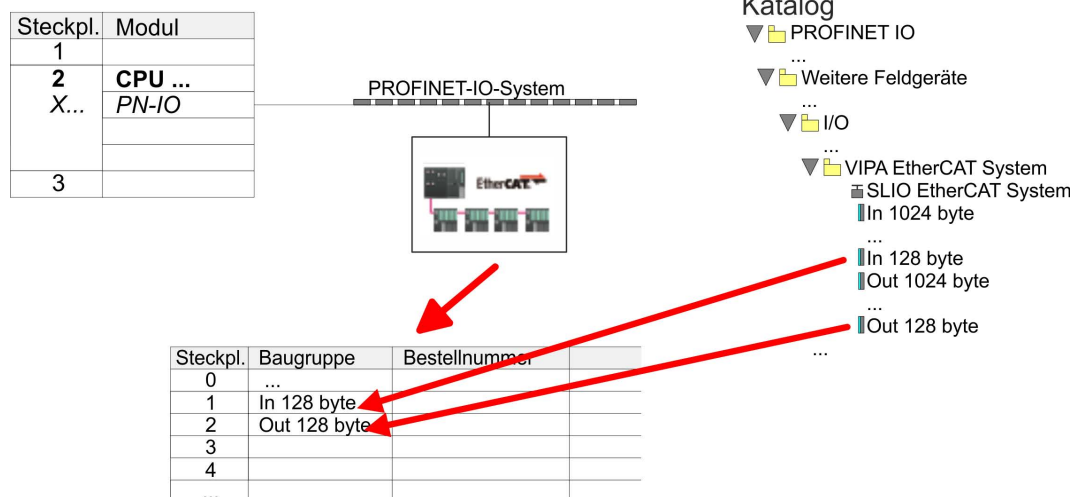


1. Navigieren Sie im Hardware-Katalog in das Verzeichnis *"PROFINET IO → Weitere Feldgeräte → I/O → VIPA EtherCAT System"* und binden Sie das IO Device *"SLIO EtherCAT System"* an Ihr PROFINET-System an.

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**.



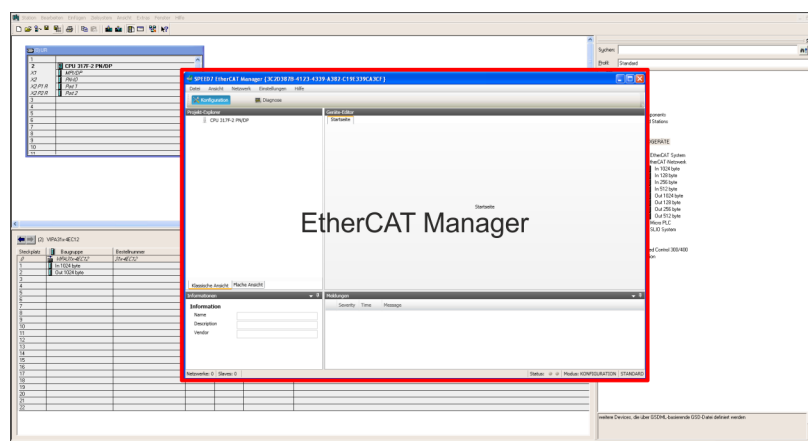
Vor dem Aufruf des **SPEED7 EtherCAT Manager** müssen Sie immer Ihr Projekt mit **"Station → Speichern und übersetzen"** speichern.



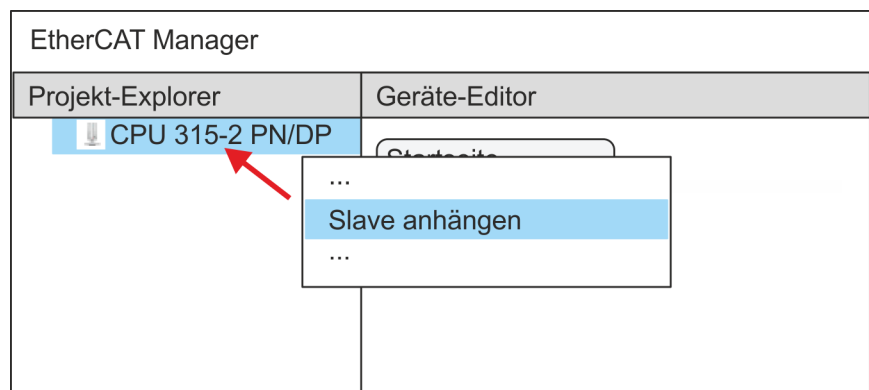
1. ➔ Klicken Sie auf das eingefügte IO Device **"EtherCAT-Netzwerk"** und wählen Sie **"Kontextmenü → Device Tool starten → SPEED7 EtherCAT Manager"**.

➔ 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.

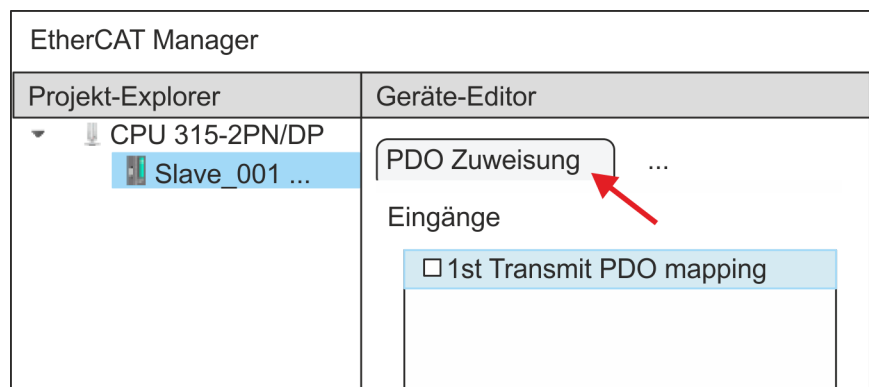


7. ➔ Klicken Sie im EtherCAT Manager auf ihre CPU und öffnen Sie über "Kontextmenü ➔ Slave anhängen" das Dialogfenster zum Hinzufügen eines EtherCAT-Slave.
- ⇒ Das Dialogfenster zur Auswahl eines EtherCAT-Slave wird geöffnet.
8. ➔ Wählen Sie Ihren *Sigma-5* EtherCAT Antrieb und bestätigen Sie Ihre Auswahl 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.

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 |

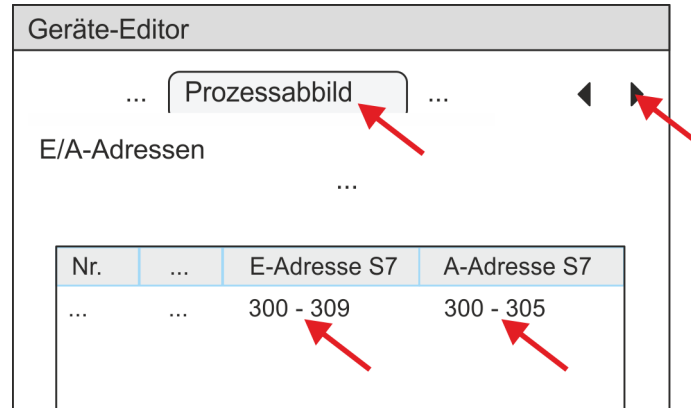
Schließen Sie den Dialog "PDO bearbeiten" mit [OK].

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 871 - VMC_InitSigma5_EC folgende PDO-Anfangsadressen:

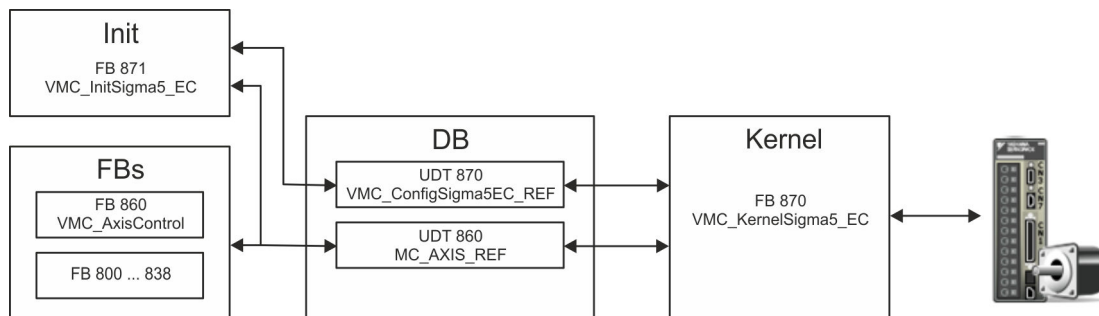
- "E-Adresse S7" → "InputsStartAddressPDO"
- "A-Adresse S7" → "OutputsStartAddressPDO"



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

3.4.3 Anwender-Programm

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 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.

3.4.3.2 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 3.5.3 "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 3.5.2 "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 6.2.2 "FB 860 VMC_AxisControl - Control-Baustein Achskontrolle" auf Seite 132*

```
⇒ CALL "VMC_AxisControl" , "DI_AxisControl01"
    SourceInputs      := "AxCtrl1_SourceInputs"
    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-Bausteinen 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.

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.

3.5 Antriebsspezifische Bausteine

3.5.1 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.

3.5.2 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. |

3.5.3 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 |

| Parameter | Deklaration | Datentyp | Beschreibung |
|-----------------------|-------------|----------|--|
| EncoderType | INPUT | INT | Encoder-Typ <ul style="list-style-type: none"> 1: Absolut-Encoder 2: Inkremental-Encoder |
| 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 <ul style="list-style-type: none"> TRUE: Aktivierung der Überwachung der maximalen Position. |
| EnableMinPosition | INPUT | BOOL | Überwachung minimale Position <ul style="list-style-type: none"> TRUE: Aktivierung der Überwachung der minimalen Position. |

| Parameter | Deklaration | Datentyp | Beschreibung |
|-----------------|-------------|----------|--|
| MinUserPosition | OUTPUT | REAL | Minimale Benutzerposition basierend auf dem minimalen Encoder Wert von 0x80000000 und dem <i>FactorPosition</i> [u]. |
| MaxUserPosition | OUTPUT | REAL | Maximale Benutzerposition 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 8 " <i>ErrorID - Zusätzliche Fehlerinformationen</i> " auf Seite 203 |

4 Einsatz *Sigma-7S* EtherCAT

4.1 Übersicht

Einsatz des Doppelachs-Antriebs ↗ *Kapitel 5 "Einsatz Sigma-7W EtherCAT" auf Seite 87*

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.

4.2 Parameter am Antrieb einstellen



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-7S (24Bit Encoder)

| Servopack Parameter | Adresse | 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 |

| Servopack Parameter | Adresse | Name | Wert |
|---------------------|------------|--------------------------------------|------|
| PnB08 | (2702h:02) | Velocity User Unit (Denominator) | 1 |
| PnB0A | (2703h:01) | Acceleration User Unit (Numerator) | 1 |
| PnB0C | (2703h:02) | Acceleration User Unit (Denominator) | 1 |

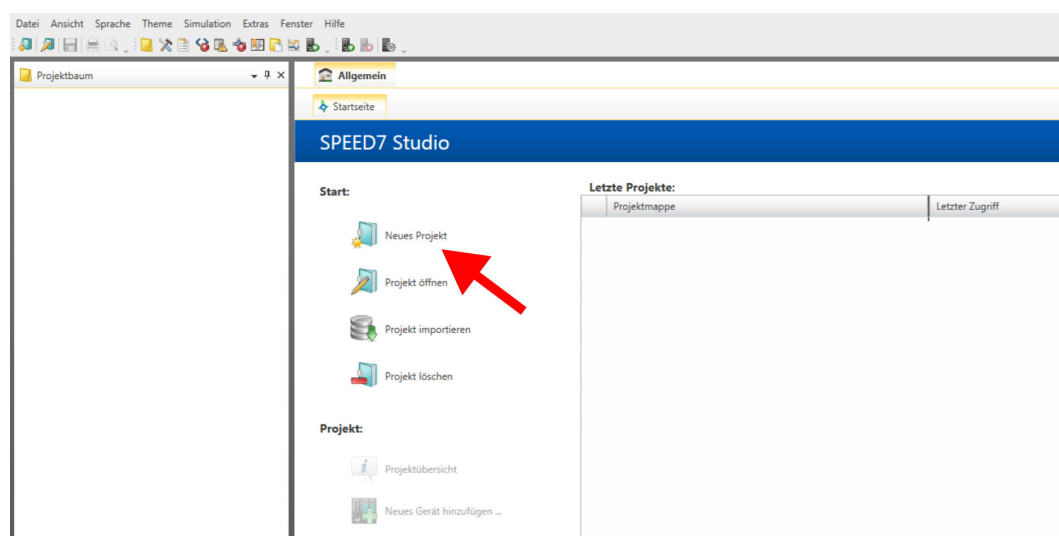
4.3 Einsatz im VIPA SPEED7 Studio

4.3.1 Hardware-Konfiguration

CPU im Projekt anlegen

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

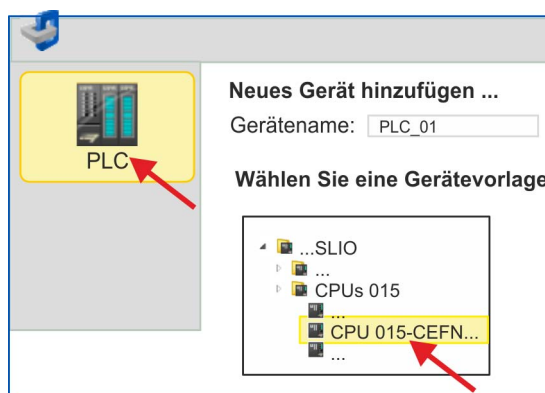
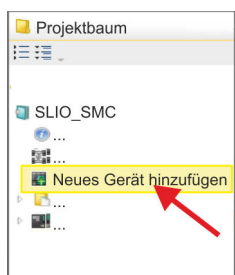
1. Starten Sie das *SPEED7 Studio*.



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

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

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



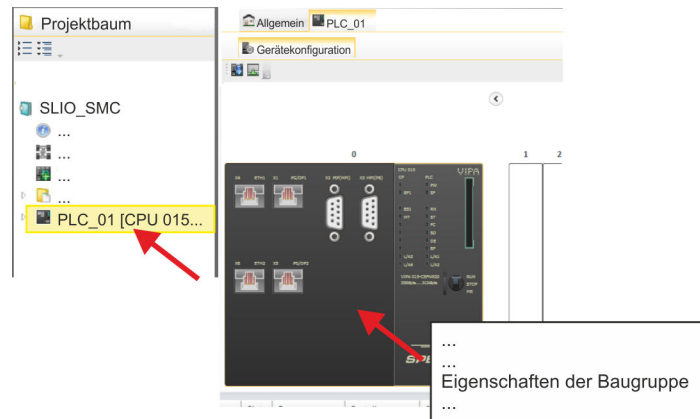
⇒ Es öffnet sich ein Dialog für die Geräteauswahl.

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.

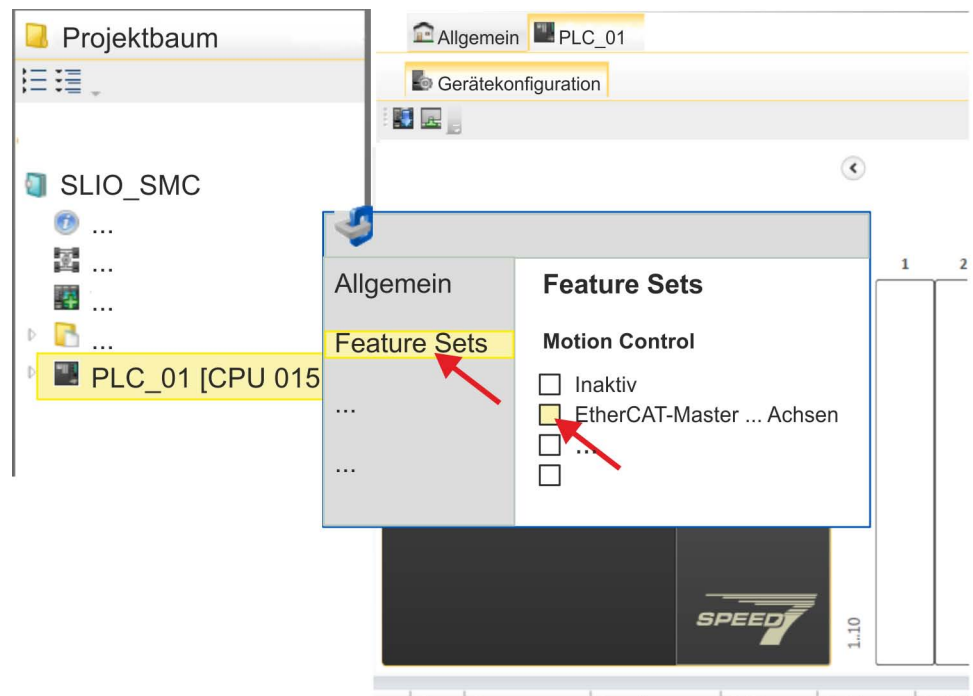
Motion-Control-Funktionen aktivieren

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].

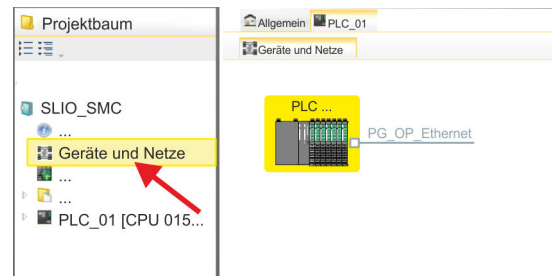
⇒ Die Motion-Control-Funktionen stehen Ihnen nun in Ihrem Projekt zur Verfügung.

**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.

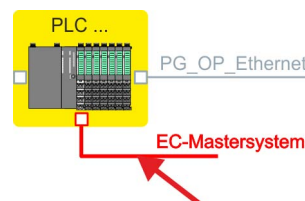
ESI-Datei installieren

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"*.

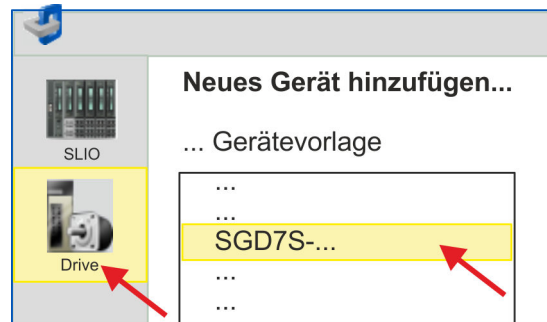
1. ➤ Laden Sie die zu Ihrem Antrieb passende ESI-Datei herunter. Entpacken Sie diese falls erforderlich.
2. ➤ Gehen Sie in Ihr *SPEED7 Studio*.
3. ➤ Öffnen Sie mit *"Extras → Gerätebeschreibungsdatei installieren (EtherCAT - ESI)"* das zugehörige Dialogfenster.
4. ➤ Geben Sie unter *"Quellpfad"* die 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"*.



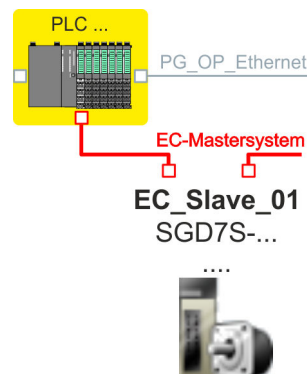
- ⇒ Es öffnet sich die Gerätevorlage zur Auswahl eines EtherCAT-Devices.



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

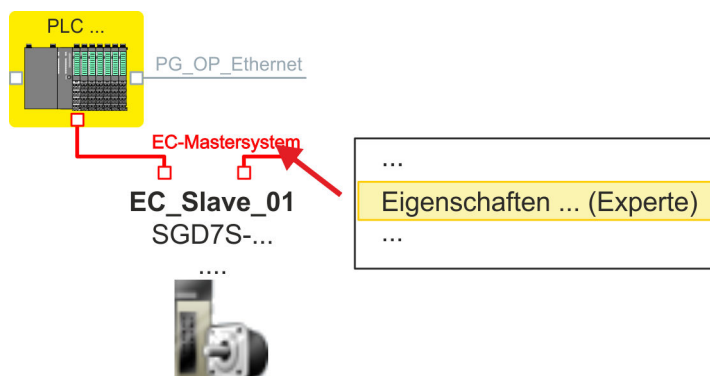
- SGD7S-xxxxAA0...
- SGD7S-xxxxDA0...
- SGD7S-xxxxA0...

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



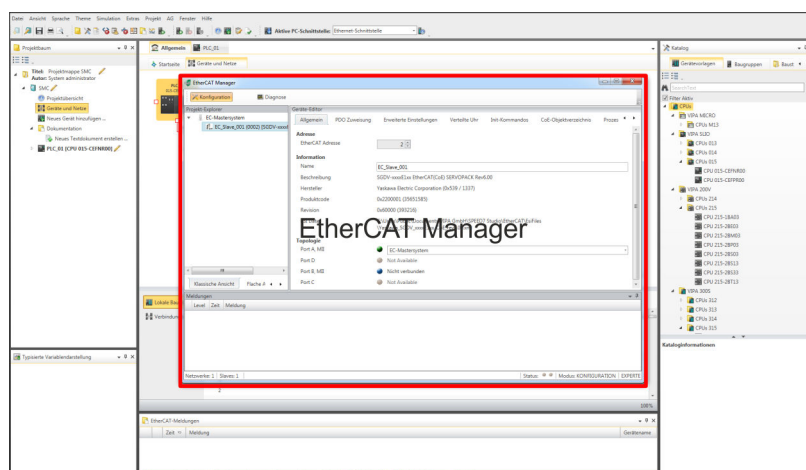
1. Klicken Sie auf "EC-Mastersystem" und wählen sie "Kontextmenü → Eigenschaft des Busystems (Experte)".



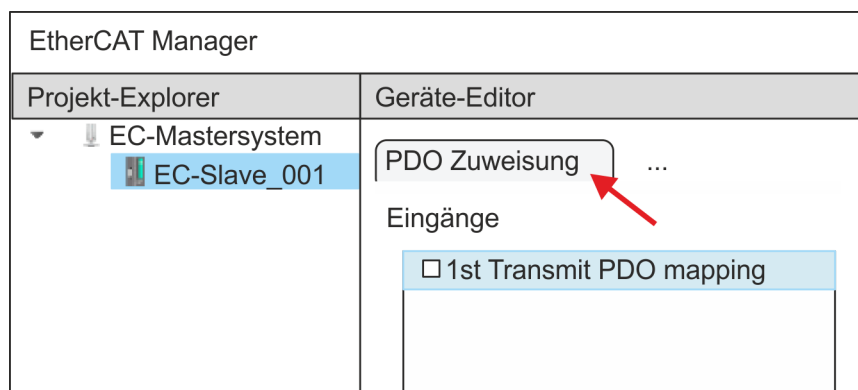
PDOs können Sie nur im "Experten-Modus" bearbeiten! Ansonsten werden die Schaltflächen ausgeblendet.

- ⇒ 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 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 sections:

- 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"*.

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 873 - VMC_InitSigma7S_EC folgende PDO-Anfangsadressen:

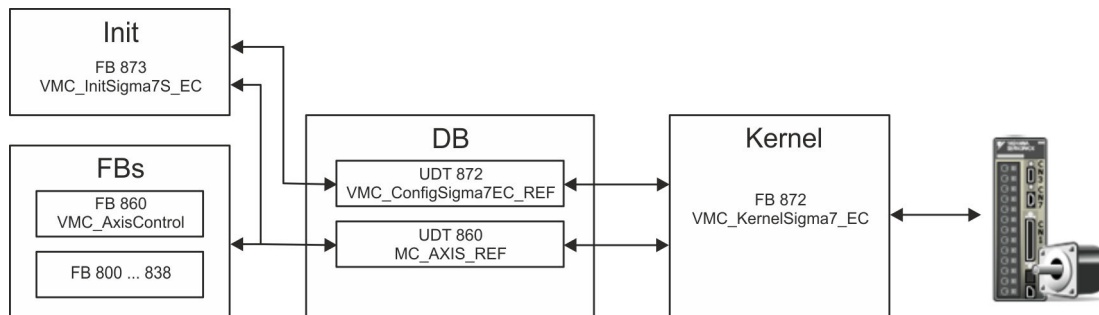
- "E-Adresse S7" → "InputsStartAddressPDO"
- "A-Adresse S7" → "OutputsStartAddressPDO"

| Geräte-Editor | | | |
|------------------------------|-----|--------------|--------------|
| ... Prozessabbild ... | | | |
| E/A-Adressen | | | |
| ... | | | |
| Nr. | ... | E-Adresse S7 | A-Adresse S7 |
| ... | ... | 300 - 309 | 300 - 305 |

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

4.3.2 Anwender-Programm

4.3.2.1 Programmstruktur



- **DB**

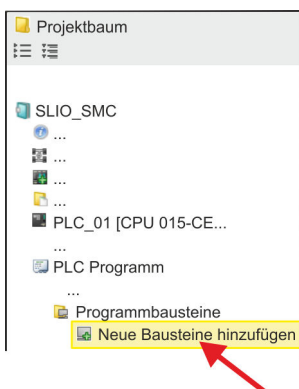
Für jede Achse ist ein Datenbaustein (Achs-DB) für Konfiguration und Statusdaten anzulegen. Der Datenbaustein besteht aus folgenden Datenstrukturen:

 - 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.

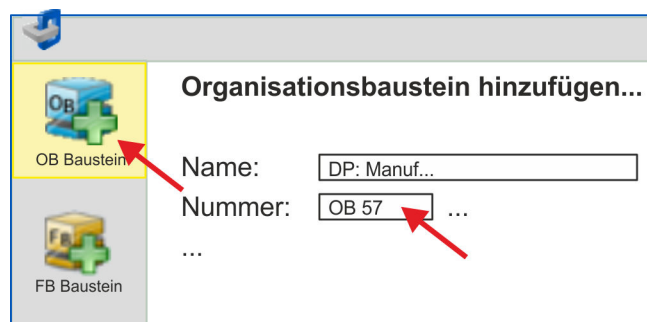
- FB 872 - *VMC_KernelSigma7_EC*
 - Der *Kernel*-Baustein kommuniziert mit dem Antrieb über das entsprechende Bussystem, verarbeitet die Benutzeraufträge und liefert Statusmeldungen zurück.
 - Spezifischer Baustein für *Sigma-7* EtherCAT.
 - Der Austausch der Daten erfolgt mittels des *Achs-DB*.
- FB 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.

4.3.2.2 Programmierung

Bausteine in Projekt kopieren

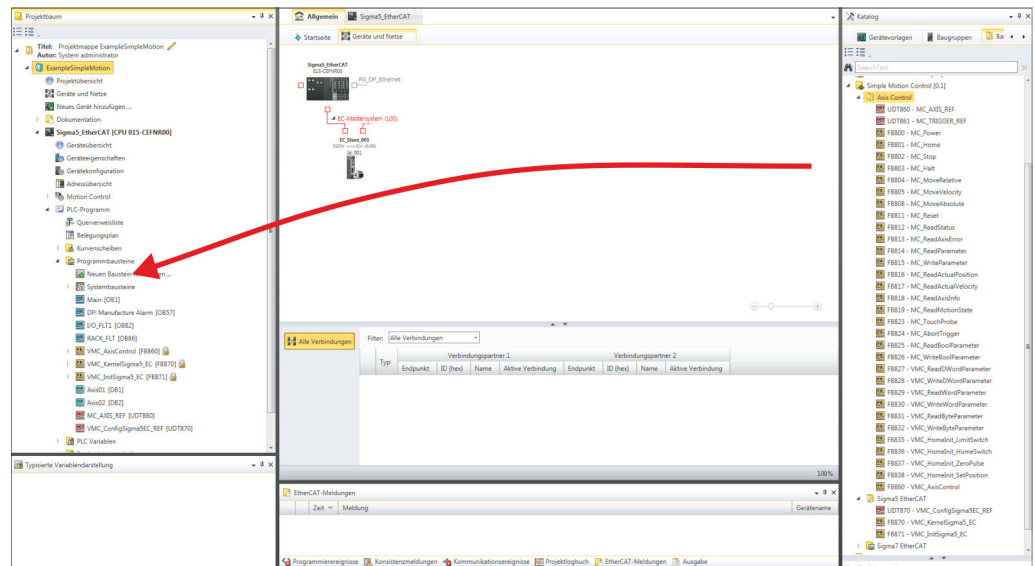


1. ➤ Klicken Sie im *Projektbaum* innerhalb der CPU unter "*PLC-Programm*", "*Programmbausteine*" auf "*Neuen Baustein hinzufügen*".



⇒ 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 4.5.3 "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 4.5.2 "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 6.2.2 "FB 860 VMC_AxisControl - Control-Baustein Achskontrolle" auf Seite 132*

```
⇒ CALL "VMC_AxisControl" , "DI_AxisControl01"
    SourceInputs      := "AxCtrl1_SourceInputs"
    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-Bausteinen 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.

4.4 Einsatz im Siemens SIMATIC Manager

4.4.1 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.

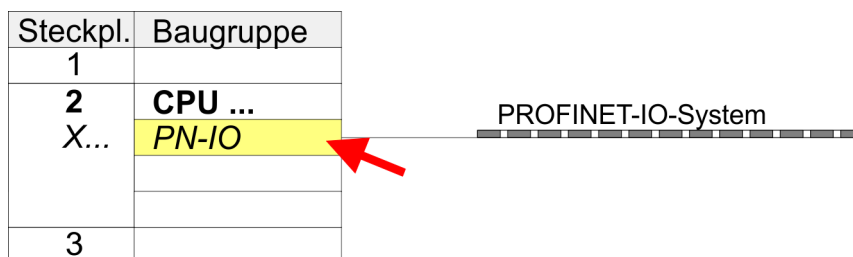
4.4.2 Hardware-Konfiguration

CPU im Projekt anlegen

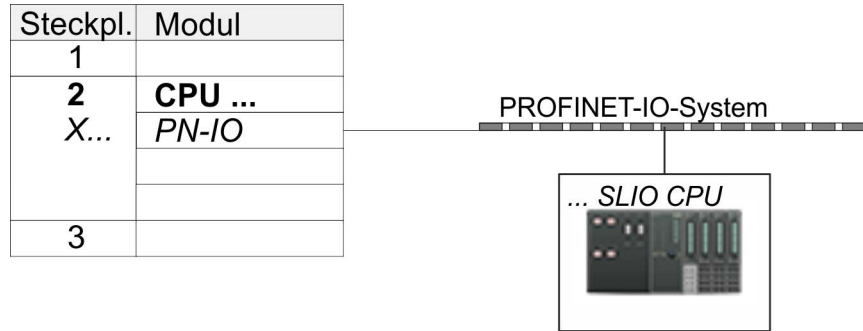
| Steckp.. | Baugruppe |
|----------|------------------------|
| 1 | |
| 2 | CPU 315-2 PN/DP |
| X1 | MPI/DP |
| X2 | PN-IO |
| X2... | Port 1 |
| X2... | Port 2 |
| 3 | |

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.

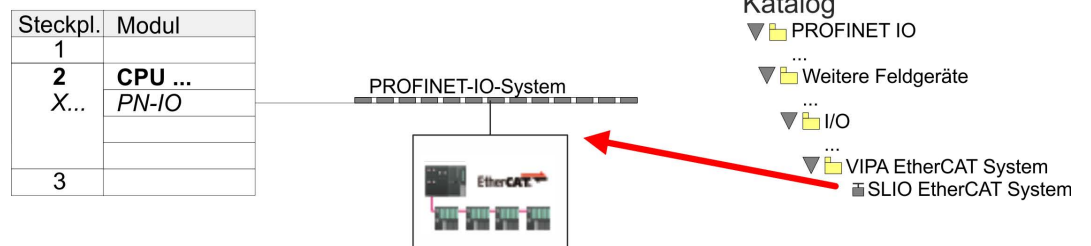
⇒ In der Steckplatzübersicht des PROFINET-IO-Device "*VIPA SLIO CPU*" ist auf Steckplatz 0 die CPU bereits vorplatziert. Ab Steckplatz 1 können Sie Ihre System SLIO Module platzieren.

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

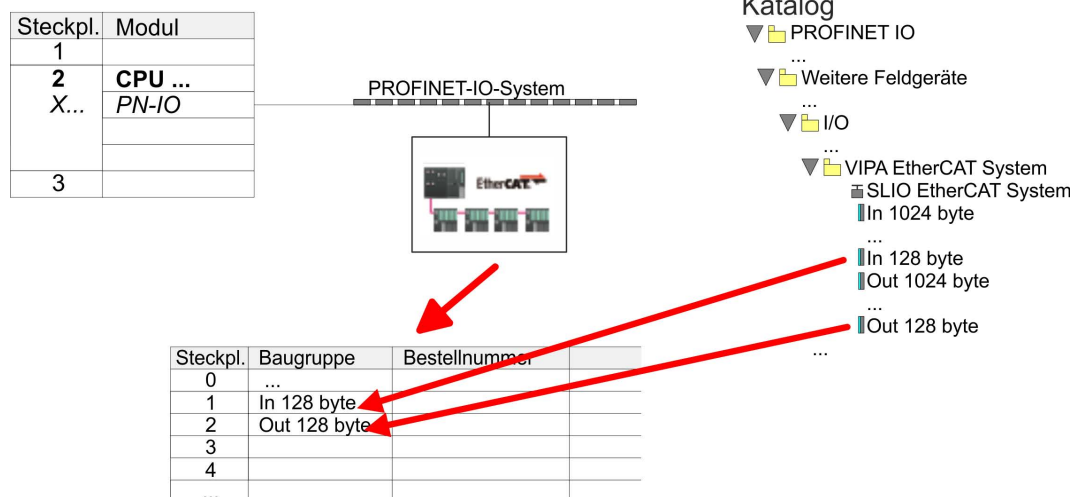


1. Navigieren Sie im Hardware-Katalog in das Verzeichnis "*PROFINET IO* → *Weitere Feldgeräte* → *I/O* → *VIPA EtherCAT System*" und binden Sie das IO Device "*SLIO EtherCAT System*" an Ihr PROFINET-System an.

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-7S EtherCAT Antrieb konfigurieren

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



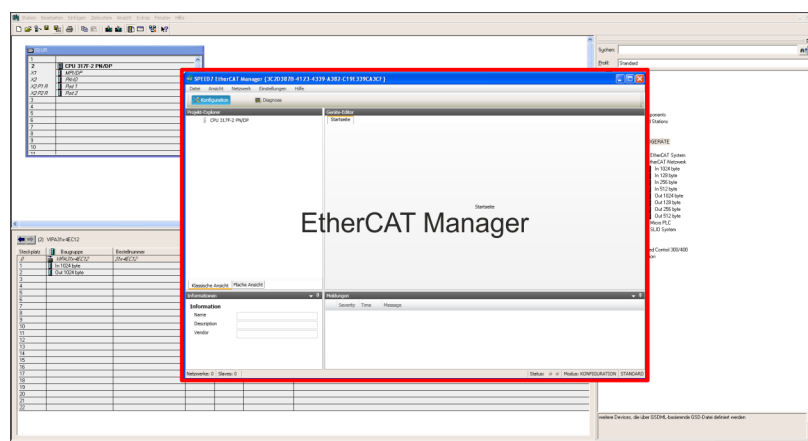
Vor dem Aufruf des *SPEED7 EtherCAT Manager* müssen Sie immer Ihr Projekt mit "Station → Speichern und übersetzen" speichern.



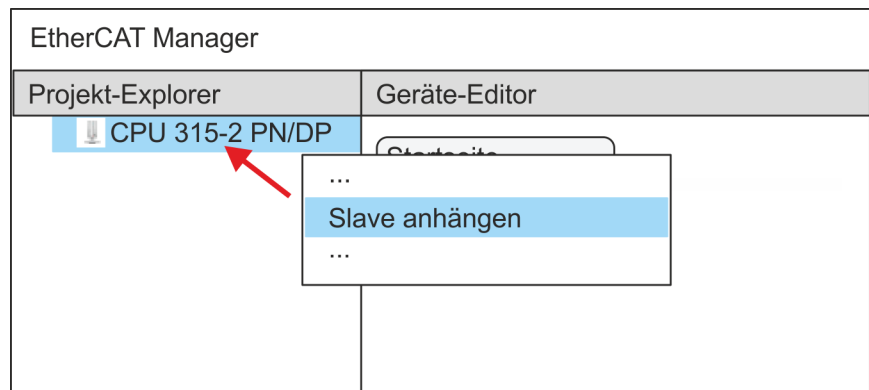
1. ➔ Klicken Sie auf das eingefügte IO Device "EtherCAT-Netzwerk" und wählen Sie "Kontextmenü → Device Tool starten → *SPEED7 EtherCAT Manager*".

➔ 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.

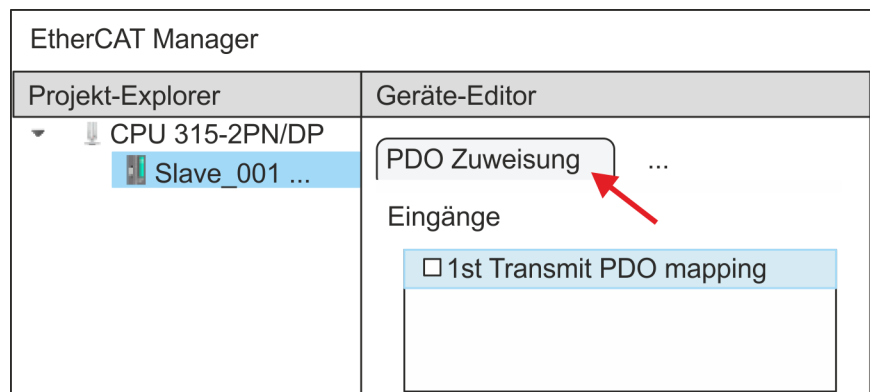


7. ➤ Klicken Sie im EtherCAT Manager auf ihre CPU und öffnen Sie über "Kontextmenü ➔ Slave anhängen" das Dialogfenster zum Hinzufügen eines EtherCAT-Slave.
 - ⇒ Das Dialogfenster zur Auswahl eines EtherCAT-Slave wird geöffnet.
8. ➤ Wählen Sie Ihren *Sigma-7S* EtherCAT Antrieb und bestätigen Sie Ihre 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 |

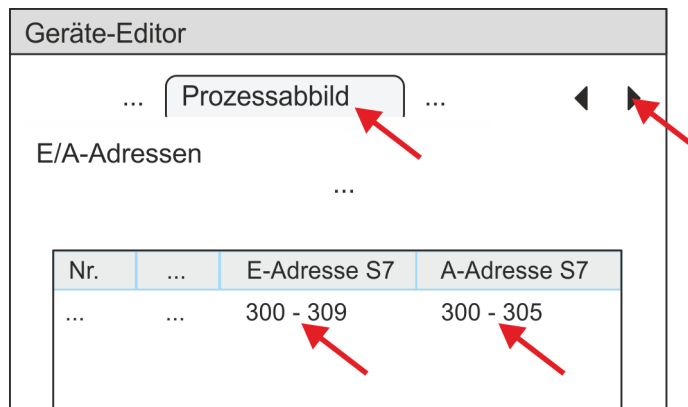
Schließen Sie den Dialog *"PDO bearbeiten"* mit [OK].

- 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:

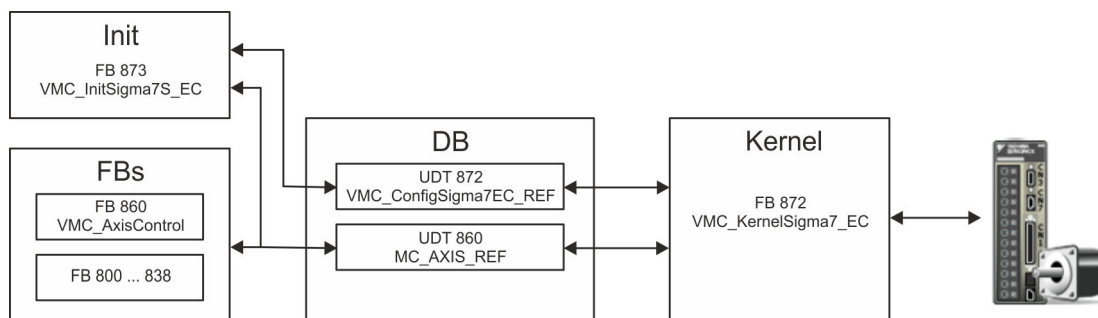
- "E-Adresse S7" → "InputsStartAddressPDO"
- "A-Adresse S7" → "OutputsStartAddressPDO"



- 19.** 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.

4.4.3 Anwender-Programm

4.4.3.1 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.

4.4.3.2 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 4.5.3 "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. ↪ 77

```
⇒ 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 4.5.2 "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 6.2.2 "FB 860 VMC_AxisControl - Control-Baustein Achskontrolle" auf Seite 132*

```

⇒      CALL  "VMC_AxisControl" , "DI_AxisControl01"
        SourceInputs      := "AxCtrl1_SourceInputs"
        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 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 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-Bausteinen 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.

4.4.4 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.

4.5 Antriebsspezifische Bausteine

4.5.1 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.

4.5.2 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. |

4.5.3 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 |

| Parameter | Deklaration | Datentyp | Beschreibung |
|-----------------------|-------------|----------|--|
| EncoderType | INPUT | INT | Encoder-Typ <ul style="list-style-type: none"> 1: Absolut-Encoder 2: Inkremental-Encoder |
| 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 <ul style="list-style-type: none"> TRUE: Aktivierung der Überwachung der maximalen Position. |
| EnableMinPosition | INPUT | BOOL | Überwachung minimale Position <ul style="list-style-type: none"> TRUE: Aktivierung der Überwachung der minimalen Position. |

| Parameter | Deklaration | Datentyp | Beschreibung |
|-----------------|-------------|----------|--|
| MinUserPosition | OUTPUT | REAL | Minimale Benutzerposition basierend auf dem minimalen Encoder Wert von 0x80000000 und dem <i>FactorPosition</i> [u]. |
| MaxUserPosition | OUTPUT | REAL | Maximale Benutzerposition 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 8 " <i>ErrorID - Zusätzliche Fehlerinformationen</i> " auf Seite 203 |

5 Einsatz *Sigma-7W* EtherCAT

5.1 Übersicht

Einsatz des Singelachs-Antriebs ➔ Kapitel 4 "Einsatz *Sigma-7S* EtherCAT" auf Seite 48

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.

5.2 Parameter am Antrieb einstellen



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 | 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 |

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

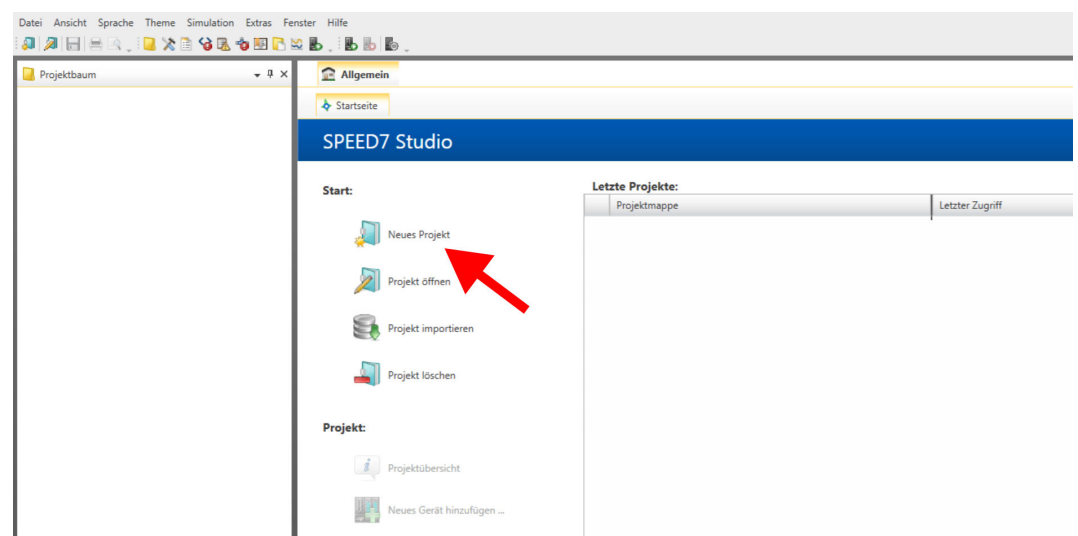
Achse 2 - Module 2 (24Bit Encoder)

| Servopack Parameter | Adresse | 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 |

5.3 Einsatz im VIPA *SPEED7 Studio***5.3.1 Hardware-Konfiguration****CPU im Projekt anlegen**

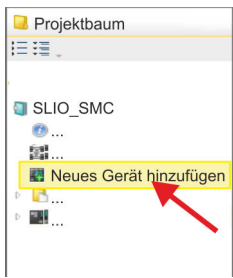
Bitte verwenden Sie für die Projektierung das *SPEED7 Studio* ab V1.6.1.

1. Starten Sie das *SPEED7 Studio*.

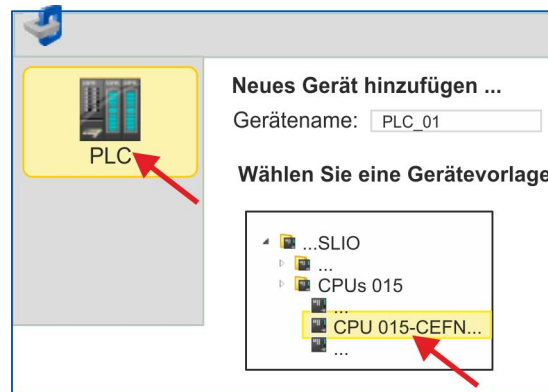


2. Erstellen sie auf der Startseite mit "*Neues Projekt*" ein neues Projekt und vergeben Sie einen "*Projektnamen*".

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



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

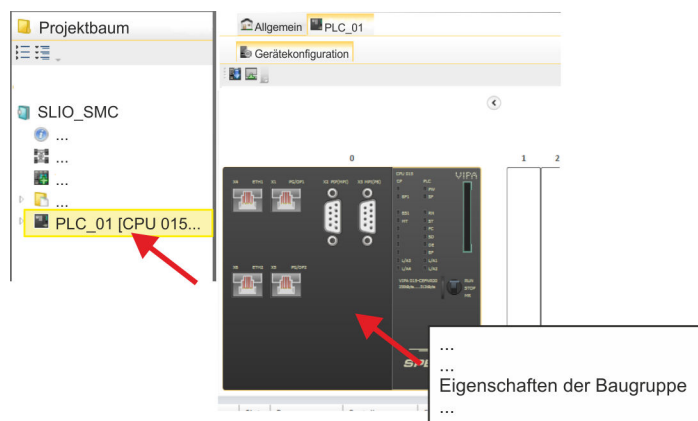


⇒ Es öffnet sich ein Dialog für die Geräteauswahl.

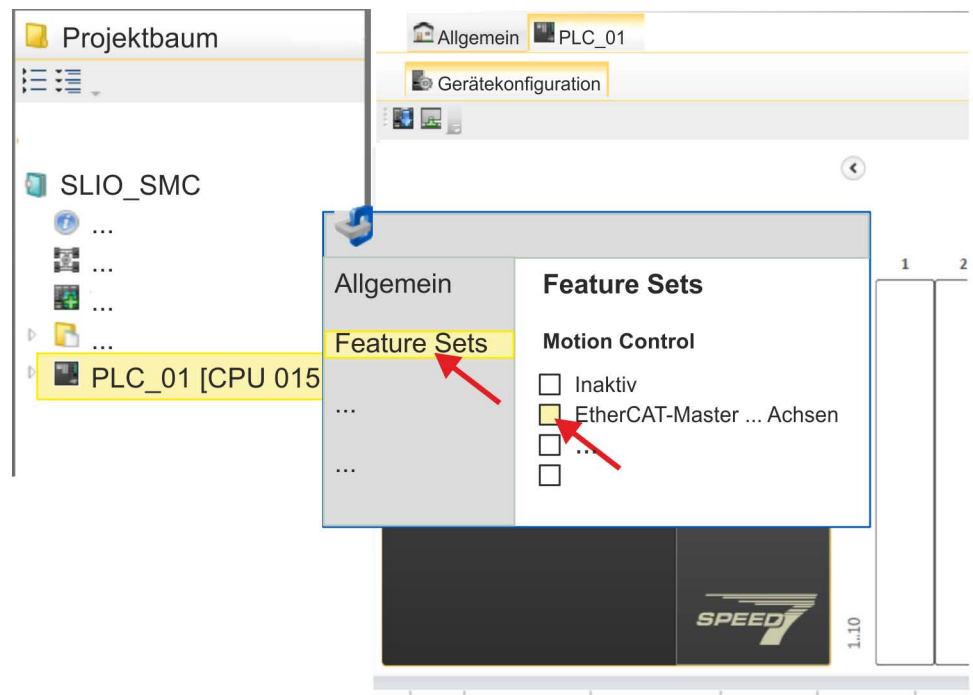
4. ➤ Wählen Sie unter den "Gerätevorlagen" eine CPU mit EtherCAT-Master-Funktionalität wie z.B. die CPU 015-CEFN00 und klicken Sie auf [OK].
- ⇒ Die CPU wird in "Geräte und Netze" eingefügt und die "Gerätekonfiguration" geöffnet.

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.

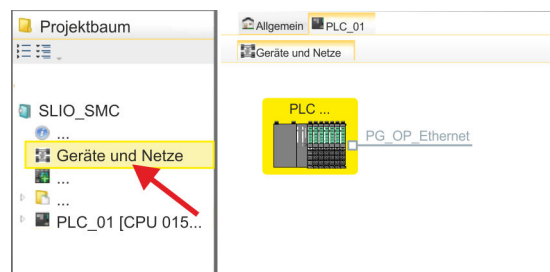


VORSICHT!

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.

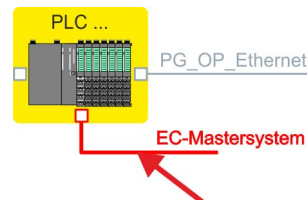
ESI-Datei installieren

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"*.

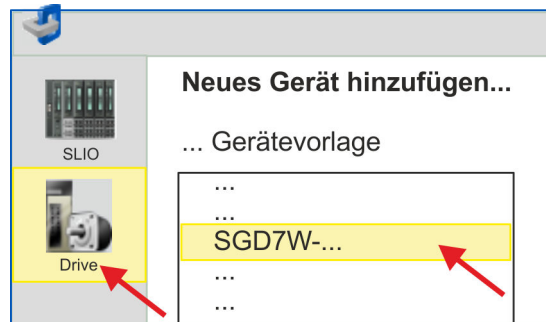
1. ➤ Laden Sie die zu Ihrem Antrieb passende ESI-Datei herunter. Entpacken Sie diese falls erforderlich.
2. ➤ Gehen Sie in Ihr *SPEED7 Studio*.
3. ➤ Öffnen Sie mit *"Extras → Gerätebeschreibungsdatei installieren (EtherCAT - ESI)"* das zugehörige Dialogfenster.
4. ➤ Geben Sie unter *"Quellpfad"* die 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"*.



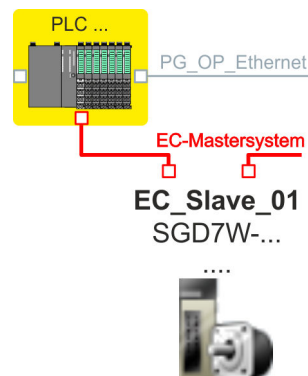
- ⇒ Es öffnet sich die Gerätevorlage zur Auswahl eines EtherCAT-Devices.



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

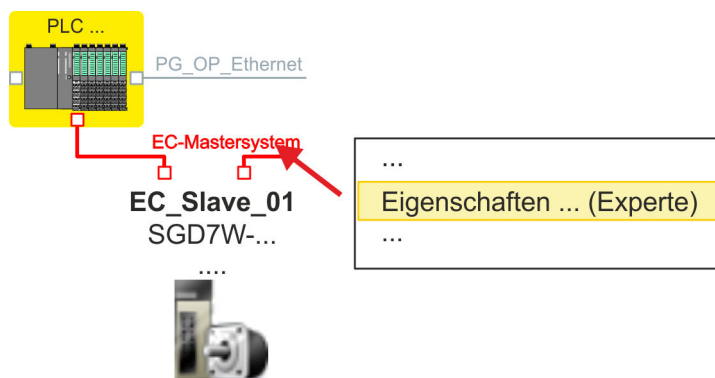
- SGD7W-xxxxA0...

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



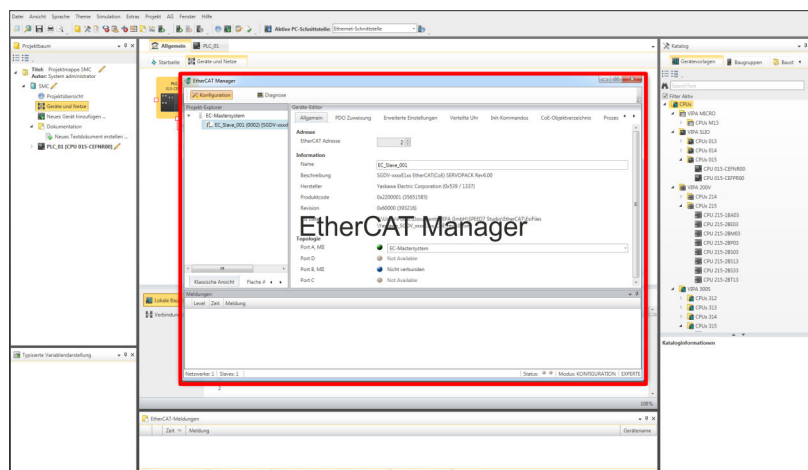
1. Klicken Sie auf "EC-Mastersystem" und wählen sie "Kontextmenü → Eigenschaft des Busystems (Experte)".



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.

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 of the dialog 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.

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! | |

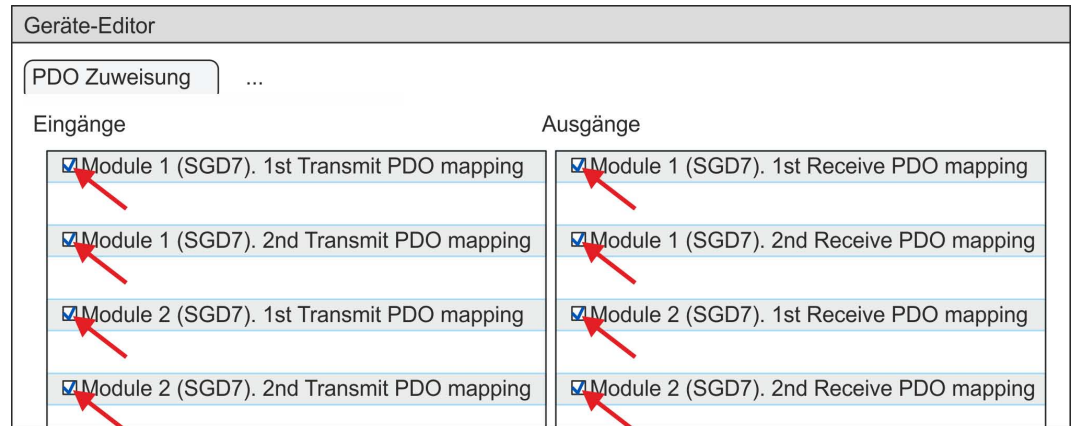
| Einträge | Modul 1 (Achse 1) | Modul 2 (Achse 2) | Bitlänge |
|----------------------|-------------------|-------------------|----------|
| Name | Index | Index | |
| Control word | 0x6040:00 | 0x6840:00 | 16Bit |
| Target position | 0x607A:00 | 0x687A:00 | 32Bit |
| Target velocity | 0x60FF:00 | 0x68FF:00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

Ausgänge: 2nd Receive PDO

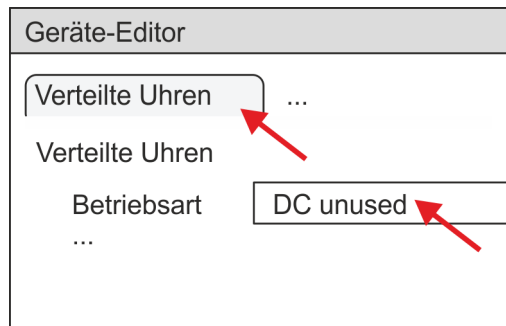
| Module 1 (SGD7). 2nd Receive PDO | Module 2 (SGD7). 2nd Receive PDO |
|---|--|
| Name: Module 1 (SGD7). 2nd Receive PDO mapping | Name: Module 2 (SGD7). 2nd Receive PDO mapping |
| Index: 0x1601 | Index: 0x1611 |
| Flags: Alles deaktiviert | |
| Richtung: RxPdo (Ausgang): aktiviert | |
| Ausschließen: 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".

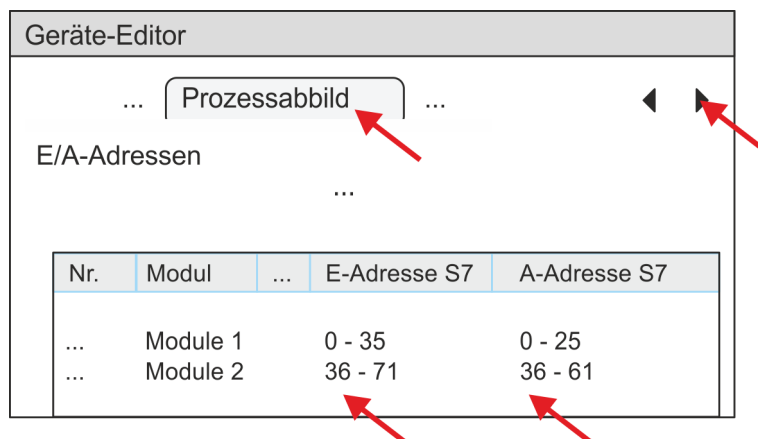


7. ➔ Wählen Sie im "Geräte-Editor" des SPEED7 EtherCAT Manager den Reiter "Verteilte Uhren" an und stellen Sie "DC unused" als "Betriebsart" ein.



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:

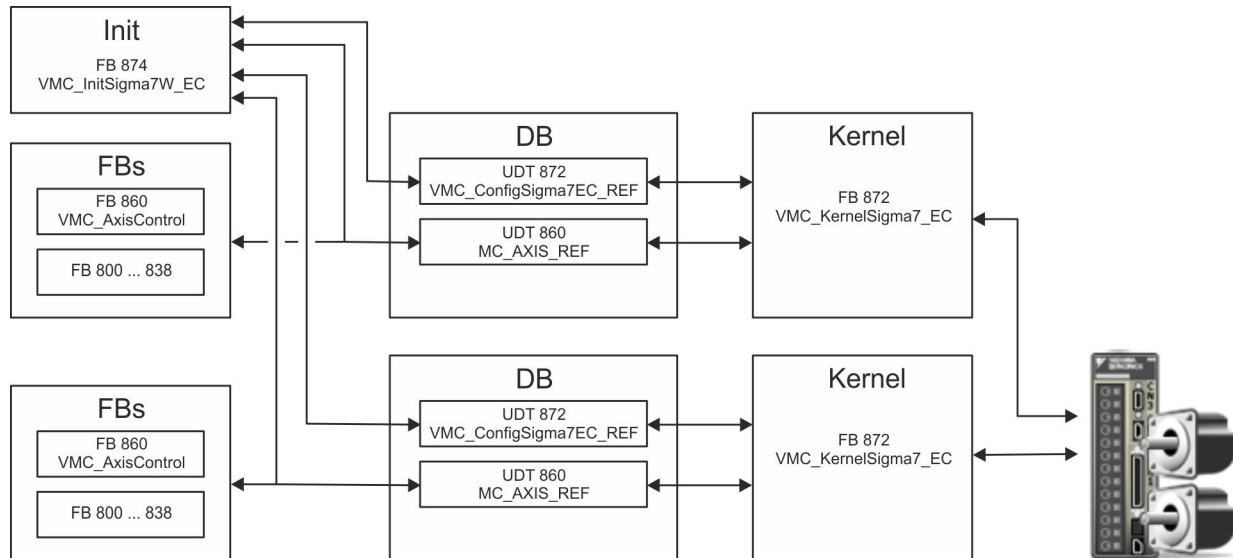
- Module 1: "E-Adresse S7" → "M1_PdoInputs" (hier 0)
- Module 2: "E-Adresse S7" → "M2_PdoInputs" (hier 36)
- Module 1: "A-Adresse S7" → "M1_PdoOutputs" (hier 0)
- Module 2: "A-Adresse S7" → "M2_PdoOutputs" (hier 36)



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

5.3.2 Anwender-Programm

5.3.2.1 Übersicht



■ 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.

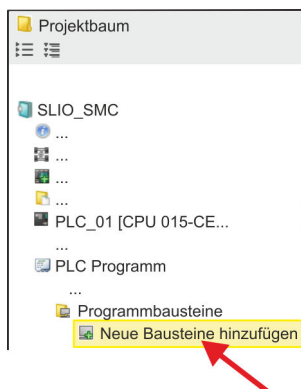
■ FB 872 - *VMC_KernelSigma7_EC*

- 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*.

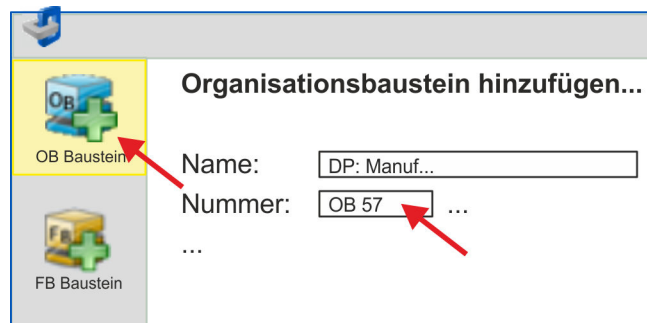
- **FB 860 - *VMC_AxisControl***
 - Universal-Baustein für alle Antriebe und Bussysteme.
 - Je Achse ist der FB 860 - *VMC_AxisControl* aufzurufen.
 - Unterstützt einfache Bewegungskommandos und liefert alle relevanten Statusmeldungen.
 - Der Austausch der Daten erfolgt mittels des *Achs-DB*.
 - Über 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.

5.3.2.2 Programmierung

Bausteine in Projekt kopieren

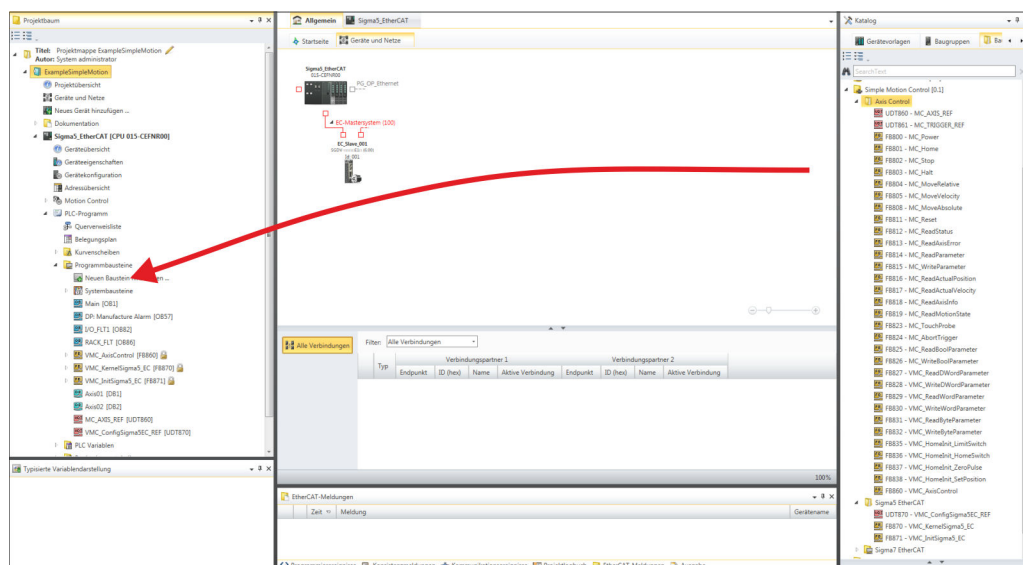


1. ➔ Klicken Sie im *Projektbaum* innerhalb der CPU unter "*PLC-Programm*", "*Programmbausteine*" auf "*Neuen Baustein hinzufügen*".



⇒ 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 5.5.3 "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. ↪ 99

```

⇒ 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 5.5.2 "FB 872 - VMC_KernelSigma7_EC - Sigma-7 EtherCAT Kernel" auf Seite 126*

```

⇒ 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 6.2.2 "FB 860 VMC_AxisControl - Control-Baustein Achskontrolle" auf Seite 132*

```
⇒ CALL "VMC_AxisControl" , "DI_AxisControl01"
    SourceInputs      := "AxCtrl1_SourceInputs"
    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-Bausteinen 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.

5.4 Einsatz im Siemens SIMATIC Manager

5.4.1 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.

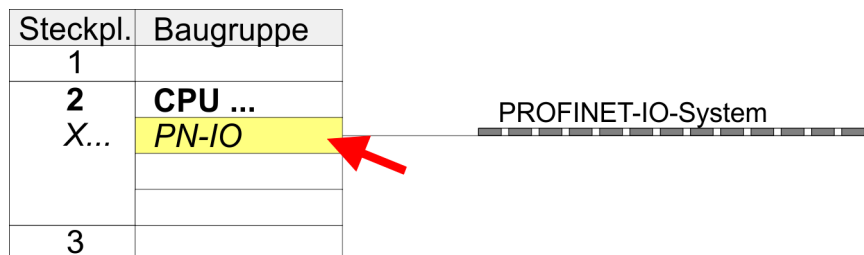
5.4.2 Hardware-Konfiguration

CPU im Projekt anlegen

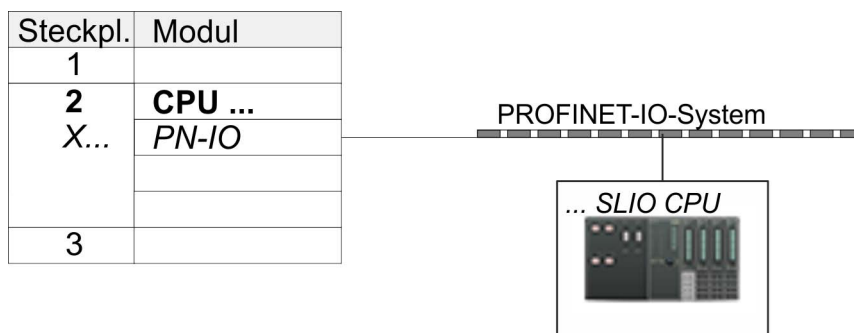
| Steckpl.. | Baugruppe |
|-----------|------------------------|
| 1 | |
| 2 | CPU 315-2 PN/DP |
| X1 | MPI/DP |
| X2 | PN-IO |
| X2... | Port 1 |
| X2... | Port 2 |
| 3 | |

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"*.



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.

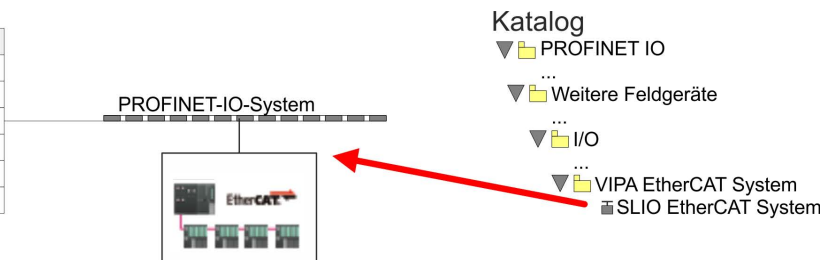
Ethernet-PG/OP-Kanal parametrieren

| 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

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

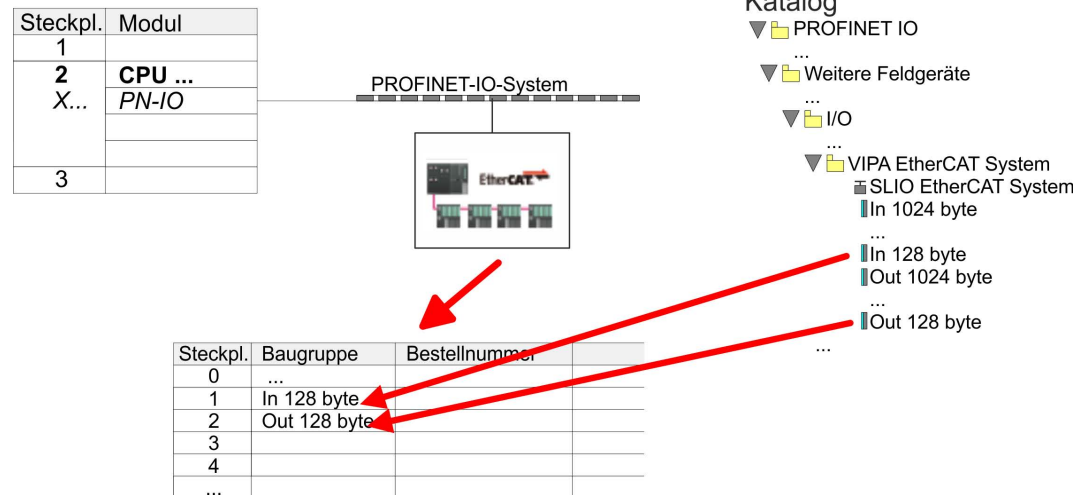


- Navigieren Sie im Hardware-Katalog in das Verzeichnis **"PROFINET IO"** → **Weitere Feldgeräte** → **I/O** → **VIPA EtherCAT System** und binden Sie das IO Device **"SLIO EtherCAT System"** an Ihr PROFINET-System an.

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



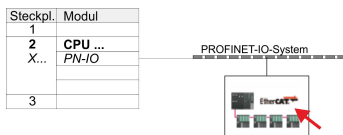
3. ➔ Wählen Sie "Station → Speichern und übersetzen"

Sigma-7W EtherCAT Doppelachs-Antrieb konfigurieren

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



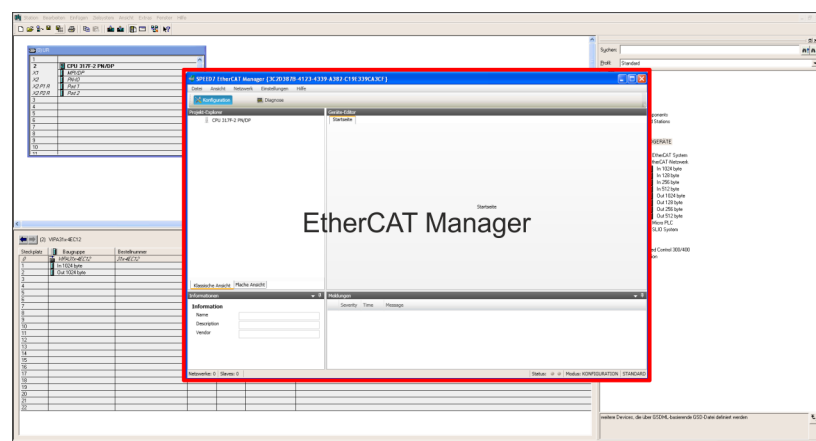
Vor dem Aufruf des *SPEED7 EtherCAT Manager* müssen Sie immer Ihr Projekt mit "Station → Speichern und übersetzen" speichern.



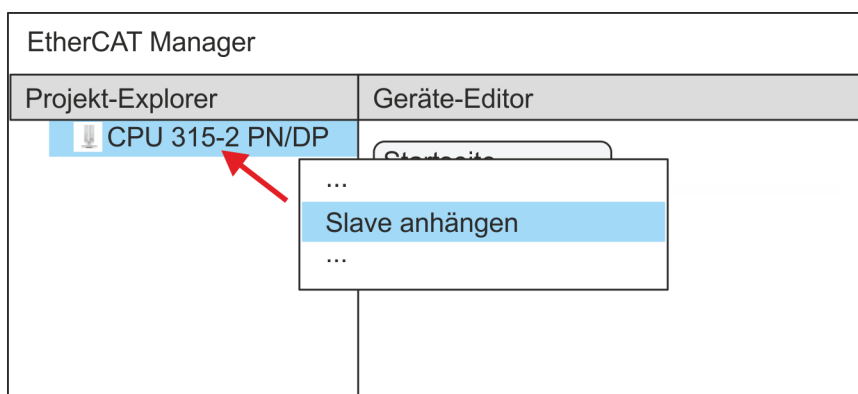
1. ➔ Klicken Sie auf das eingefügte IO Device "EtherCAT-Netzwerk" und wählen Sie "Kontextmenü → Device Tool starten → *SPEED7 EtherCAT Manager*".

➔ 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.

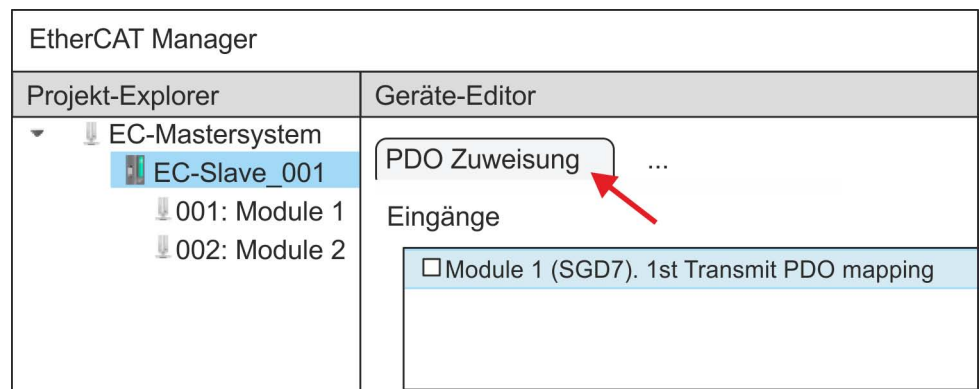


7. ➔ Klicken Sie im EtherCAT Manager auf ihre CPU und öffnen Sie über "Kontextmenü ➔ Slave anhängen" das Dialogfenster zum Hinzufügen eines EtherCAT-Slave.
 - ⇒ Das Dialogfenster zur Auswahl eines EtherCAT-Slave wird geöffnet.
8. ➔ Wählen Sie Ihren *Sigma-7W* EtherCAT Doppelachs-Antrieb und bestätigen 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.

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.

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! | |

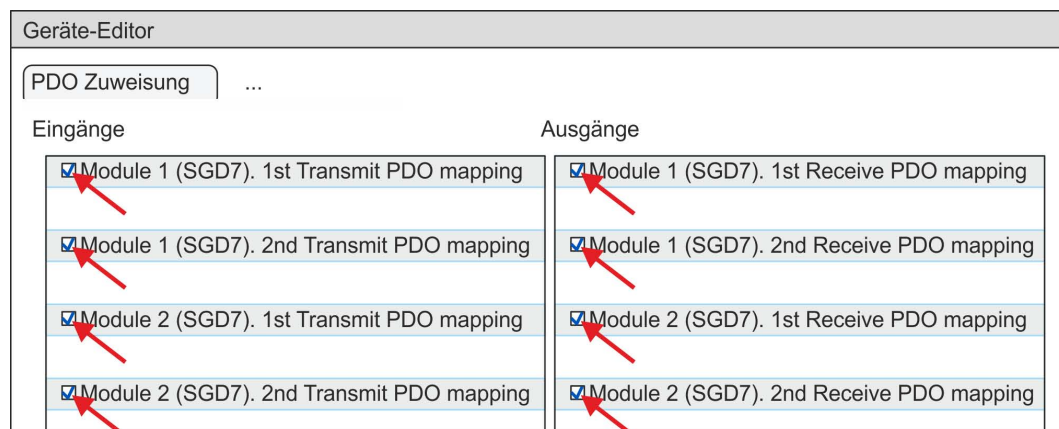
| Einträge | Modul 1 (Achse 1) | Modul 2 (Achse 2) | Bitlänge |
|----------------------|-------------------|-------------------|----------|
| Name | Index | Index | |
| Control word | 0x6040:00 | 0x6840:00 | 16Bit |
| Target position | 0x607A:00 | 0x687A:00 | 32Bit |
| Target velocity | 0x60FF:00 | 0x68FF:00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

Ausgänge: 2nd Receive PDO

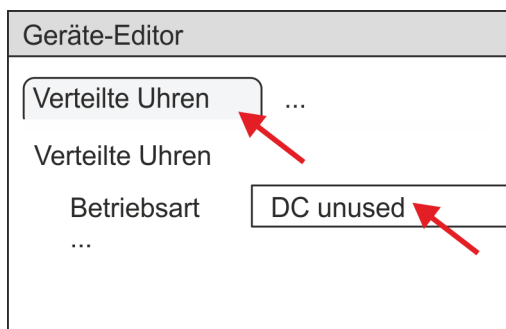
| Module 1 (SGD7). 2nd Receive PDO | Module 2 (SGD7). 2nd Receive PDO |
|---|--|
| Name: Module 1 (SGD7). 2nd Receive PDO mapping | Name: Module 2 (SGD7). 2nd Receive PDO mapping |
| Index: 0x1601 | Index: 0x1611 |
| Flags: Alles deaktiviert | |
| Richtung: RxPdo (Ausgang): aktiviert | |
| Ausschließen: 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".

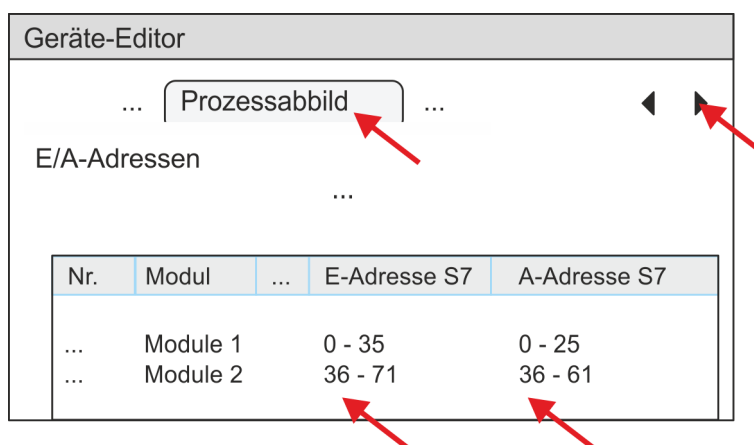


15. Wählen Sie im "Geräte-Editor" des *SPEED7 EtherCAT Manager* den Reiter "Verteilte Uhren" an und stellen Sie "DC unused" als "Betriebsart" ein.

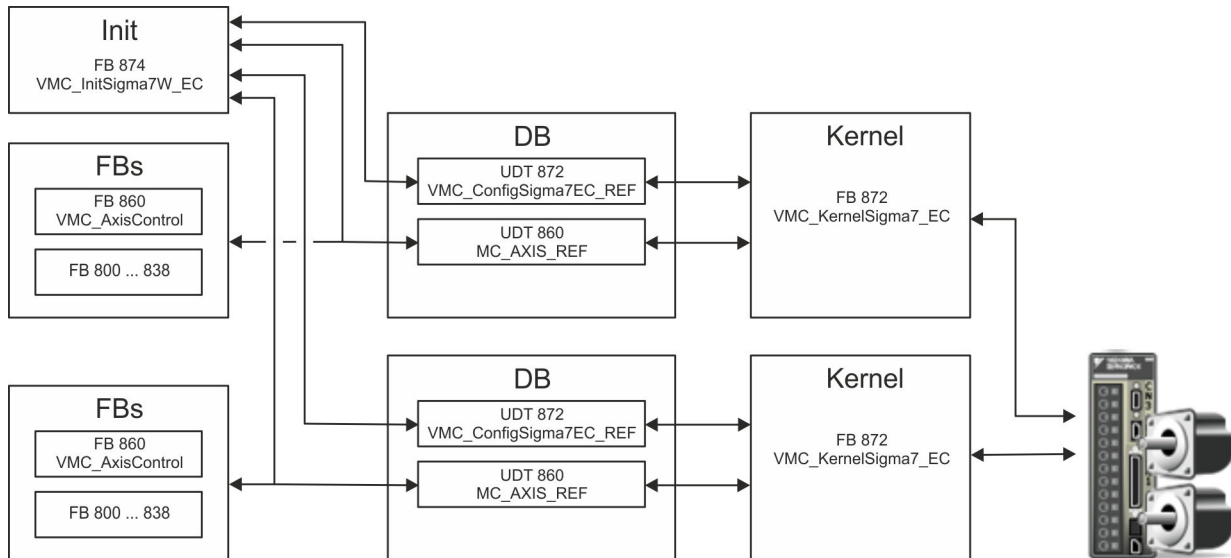


16. Wählen Sie im "Geräte-Editor" über die Pfeiltaste den Reiter "Prozessabbild" an und notieren Sie sich für die Parameter des Bausteins FB 874 - VMC_InitSigma7W_EC folgende PDO-Anfangsadressen:

- 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. 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.

18. Speichern und übersetzen Sie Ihre Konfiguration.**5.4.3 Anwender-Programm****5.4.3.1 Übersicht**■ **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.

5.4.3.2 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 Doppelachse

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

→ FB 874 - VMC_InitSigma7W_EC, DB 874 ↪ *Kapitel 5.5.3 "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. ↪ 118

```
⇒ 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 5.5.2 "FB 872 - VMC_KernelSigma7_EC - Sigma-7 EtherCAT Kernel" auf Seite 126*

```

⇒ 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 6.2.2 "FB 860 VMC_AxisControl - Control-Baustein Achskontrolle" auf Seite 132

```
⇒ CALL "VMC_AxisControl" , "DI_AxisControl01"
   SourceInputs      := "AxCtrl1_SourceInputs"
   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-Bausteinen 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.

5.4.4 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.

5.5 Antriebsspezifische Bausteine

5.5.1 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.

5.5.2 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. |

5.5.3 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 |

| Parameter | Deklaration | Datentyp | Beschreibung |
|--------------------------|-------------|----------|---|
| LogicalAddress | INPUT | INT | Startadresse der PDO-Eingangsdaten |
| 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. 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. |
| M1_FactorVelocity | INPUT | REAL | Faktor zur Umrechnung der Geschwindigkeit von Benutzereinheiten [u/s] in Antriebseinheiten [Inkrement/s] und zurück von Achse 1. 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. |
| 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. 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. |
| 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]. Die Kommandoeingaben werden vor Ausführung auf den Maximalwert überprüft. |
| M1_MaxAccelerationApp | INPUT | REAL | Maximale Beschleunigung der Applikation von Achse 1 [u/s^2]. Die Kommandoeingaben werden vor Ausführung auf den Maximalwert überprüft. |
| M1_MaxDecelerationApp | INPUT | REAL | Maximale Verzögerung der Applikation von Achse 1 [u/s^2]. Die Kommandoeingaben werden vor Ausführung auf den Maximalwert überprüft. |
| M1_MaxPosition | INPUT | REAL | Maximale Position für die Überwachung der Softwarelimits von Achse 1 [u]. |

| Parameter | Deklaration | Datentyp | Beschreibung |
|--------------------------|-------------|----------|--|
| M1_MinPosition | INPUT | REAL | Minimale Position für die Überwachung der Softwarelimits von Achse 1 [u]. |
| 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 [Inkmente] und zurück von Achse 2. 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. |
| M2_FactorVelocity | INPUT | REAL | Faktor zur Umrechnung der Geschwindigkeit von Benutzereinheiten [u/s] in Antriebseinheiten [Inkmente/s] und zurück von Achse 2. 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. |
| M2_FactorAcceleration | INPUT | REAL | Faktor zur Umrechnung der Beschleunigung von Benutzereinheiten [u/s ²] in Antriebseinheiten [10 ⁻⁴ x Inkmente/s ²] und zurück von Achse 2. 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. |
| 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 8 "ErrorID - Zusätzliche Fehlerinformationen" auf Seite 203 |

6 Bausteine zur Achskontrolle






6.1 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 |

| Baustein | Siehe Seite |
|--|---|
| FB 816 - MC_ReadActualPosition Aktuelle Position der Achse lesen |  161 |
| FB 817 - MC_ReadActualVelocity Aktuelle Geschwindigkeit der Achse lesen |  162 |
| FB 818 - MC_ReadAxisInfo Zusatzinformationen der Achse lesen |  163 |
| FB 819 - MC_ReadMotionState Zustand Bewegungsauftrag lesen |  165 |
| FB 823 - MC_TouchProbe Achspoint erfassen |  167 |
| FB 824 - MC_AbortTrigger Achspoint 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 835 - VMC_HomeInit_LimitSwitch Initialisierung Referenzfahrt auf Endschalter |  186 |
| FB 836 - VMC_HomeInit_HomeSwitch Initialisierung Referenzfahrt auf Referenzschalter |  188 |
| FB 837 - VMC_HomeInit_ZeroPulse Initialisierung Referenzfahrt auf Nullimpuls |  191 |
| FB 838 - VMC_HomeInit_SetPosition Initialisierung Referenzfahrt setze Position |  193 |

6.2 Einfache Bewegungsaufgaben

6.2.1 UDT 860 - MC_AXIS_REF - Datenstruktur Achsdaten

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

6.2.2 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. Hierzu müssen Sie den Parameter *SourceInputs* auf TRUE setzen.



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



VORSICHT!

Parameter *SourceInputs*

Bitte beachten Sie, dass eine Umschaltung über *SourceInputs* nur dann erfolgen darf, wenn sich die Achse im Zustand *Disabled* befindet. Ansonsten kann dies zu Schäden an Mensch und Maschine führen!

| Parameter | Deklaration | Datentyp | Beschreibung |
|--------------|-------------|----------|--|
| SourceInputs | INPUT | BOOL | <ul style="list-style-type: none"> ■ Eingang für den Baustein <ul style="list-style-type: none"> – FALSE: Baustein arbeitet mit den Bausteineingängen. – TRUE: Baustein arbeitet mit den optionalen Variablen, die sich im statischen Bereich des Instanz-Datenbaustein (Input-HMI) befinden, die zur Steuerung z.B. aus einem HMI heraus dienen. <p>Es ist somit eine Umschaltung zwischen "Automatik Modus" und "Manueller Modus" möglich.</p> |
| 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. |

| 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. |
| 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 | <p>Modus für absolute Positionierung:</p> <ul style="list-style-type: none"> ■ 0: kürzester Weg ■ 1: positive Richtung ■ 2: negative Richtung ■ 3: aktuelle Richtung |
| 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 einschaldbereit. – FALSE: Die Achse ist nicht einschaldbereit. <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. |

| Parameter | Deklaration | Datentyp | Beschreibung |
|-----------------|-------------|----------|--|
| 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>🔗 <i>Kapitel 8 "ErrorID - Zusätzliche Fehlerinformationen" auf Seite 203</i></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> |
| 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>Antriebsspezifischer Modus. Weitere Infos siehe Antriebsmanual.</p> <p>Beispiel <i>Sigma-5</i>:</p> <p>0: No mode changed/no mode assigned</p> <p>1: Profile Position mode</p> <p>2: Reserved (keep last mode)</p> <p>3: Profile Velocity mode</p> <p>4: Torque Profile mode</p> <p>6: Homing mode</p> <p>7: Interpolated Position mode</p> <p>8: Cyclic Sync Position mode</p> <p>9: Cyclic Sync Velocity mode</p> <p>10: Cyclic Sync Torque mode</p> <p>Other Reserved (keep last mode)</p> |

| Parameter | Deklaration | Datentyp | Beschreibung |
|-------------------|-------------|-------------|---|
| PLCopenState | OUTPUT | INT | Aktueller PLCopenState: 1: Disabled 2: Standstill 3: Homing 4: Discrete Motion 5: Continuous Motion 6: Synchronised Motion 7: Stopping 8: Errorstop |
| ActualPosition | OUTPUT | REAL | Position der Achse in [Anwendereinheit]. |
| ActualVelocity | OUTPUT | REAL | Geschwindigkeit der Achse in [Anwendereinheit/s]. |
| CmdDone | OUTPUT | BOOL | ■ Status – TRUE: Auftrag wurde ohne Fehler beendet. |
| CmdBusy | OUTPUT | BOOL | ■ Status – TRUE: Auftrag ist in Bearbeitung. |
| CmdAborted | OUTPUT | BOOL | ■ Status – TRUE: Der Auftrag wurde während der Bearbeitung von einem anderen Auftrag abgebrochen. |
| CmdError | OUTPUT | BOOL | ■ Status – TRUE: Ein Fehler ist aufgetreten. Zusätzliche Fehlerinformationen können dem Parameter <i>CmdErrorID</i> entnommen werden. |
| CmdErrorID | OUTPUT | WORD | Zusätzliche Fehlerinformationen ↳ Kapitel 8 "ErrorID - Zusätzliche Fehlerinformationen" auf Seite 203 |
| DirectionPositive | OUTPUT | BOOL | ■ Zustand Bewegungsauftrag: Position zunehmend – TRUE: Die Position der Achse nimmt zu. |
| DirectionNegative | OUTPUT | BOOL | ■ Zustand Bewegungsauftrag: Position abnehmend – TRUE: Die Position der Achse nimmt ab. |
| SWLimitMinActive | OUTPUT | BOOL | ■ Software Endschalter – TRUE: Software Endschalter Minimum aktiv (Minimale Position in negative Richtung überschritten). |
| SWLimitMaxActive | OUTPUT | BOOL | ■ Software Endschalter – TRUE: Software Endschalter Maximum aktiv (Maximale Position in positive Richtung überschritten). |
| HWLimitMinActive | OUTPUT | BOOL | ■ Hardware Endschalter – TRUE: Negativer Hardware Endschalter am Antrieb aktiv (NOT- Negative Overtravel). |
| HWLimitMaxActive | OUTPUT | BOOL | ■ Hardware Endschalter – 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.

6.3 Komplexe Bewegungsaufgaben - PLCopen-Bausteine

6.3.1 UDT 860 - MC_AXIS_REF - Datenstruktur Achsdaten

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

6.3.2 UDT 861 - MC_TRIGGER_REF - Datenstruktur Triggersignal

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

6.3.3 FB 800 - MC_Power - Achsenfreigabe

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

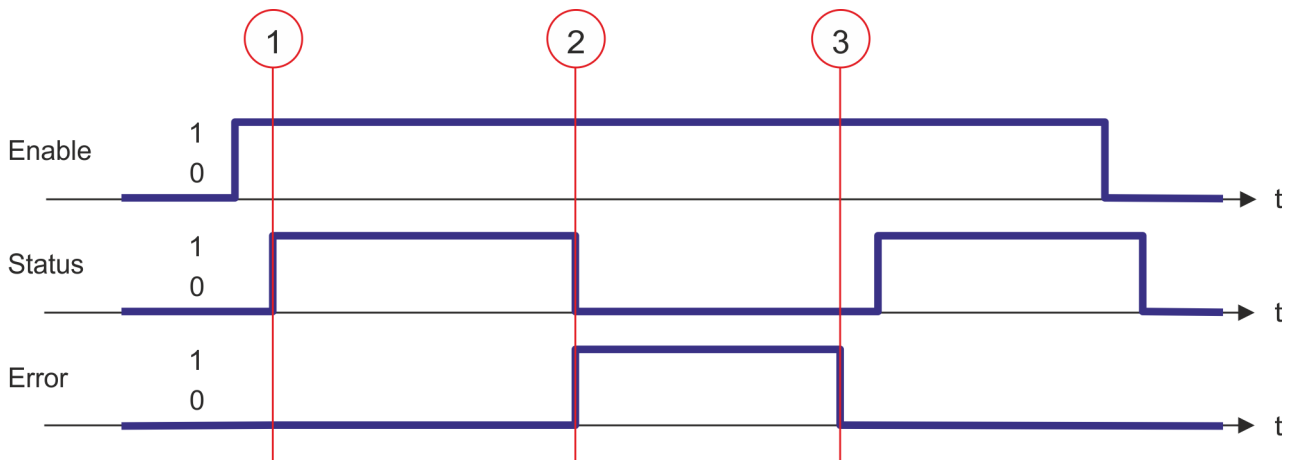
Parameter

| 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 8 "ErrorID - Zusätzliche Fehlerinformationen" auf Seite 203</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.

6.3.4 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. Bei einer virtuellen Achse ist keine Konfiguration möglich. Hier wird die Istposition der Achse gleich dem Eingangsparameter *Position* gesetzt.

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 8 "ErrorID - Zusätzliche Fehlerinformationen" auf Seite 203</p> |

PLCopen-State

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

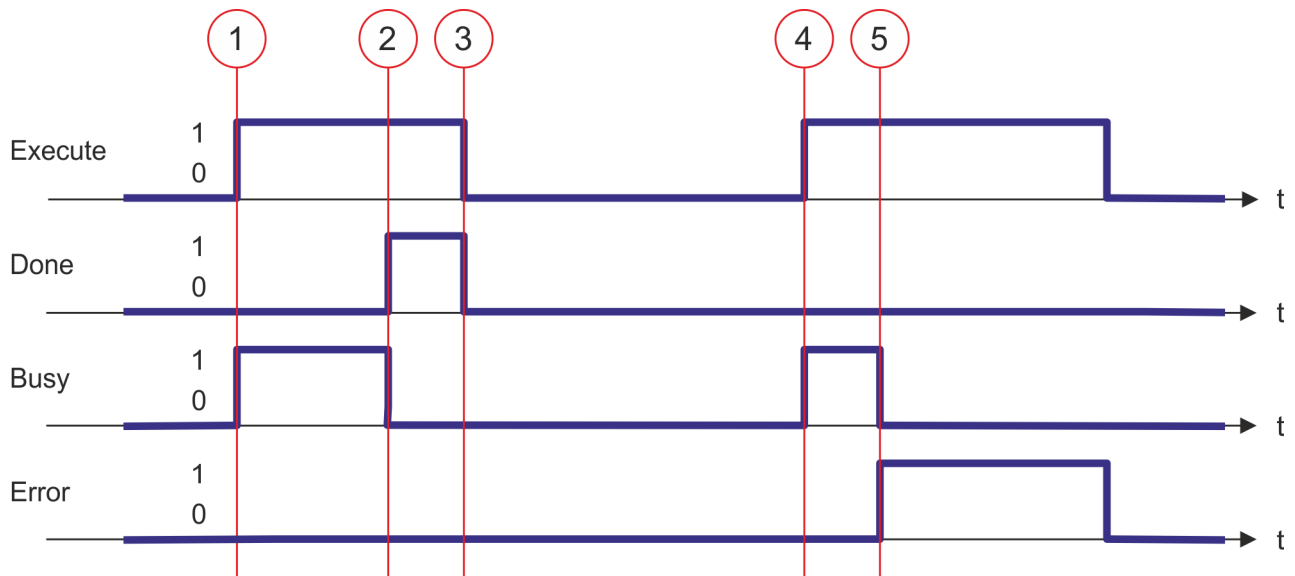
Achse referenzieren

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.

6.3.5 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 ↗ Kapitel 8 "ErrorID - Zusätzliche Fehlerinformationen" auf Seite 203 |

PLCopen-State

- Start des Auftrags in den PLCopen-States *Standstill*, *Homing*, *Discrete Motion*, *Synchronized 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.

6.3.6 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 ↪ Kapitel 8 "ErrorID - Zusätzliche Fehlerinformationen" auf Seite 203 |

PLCopen-State

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

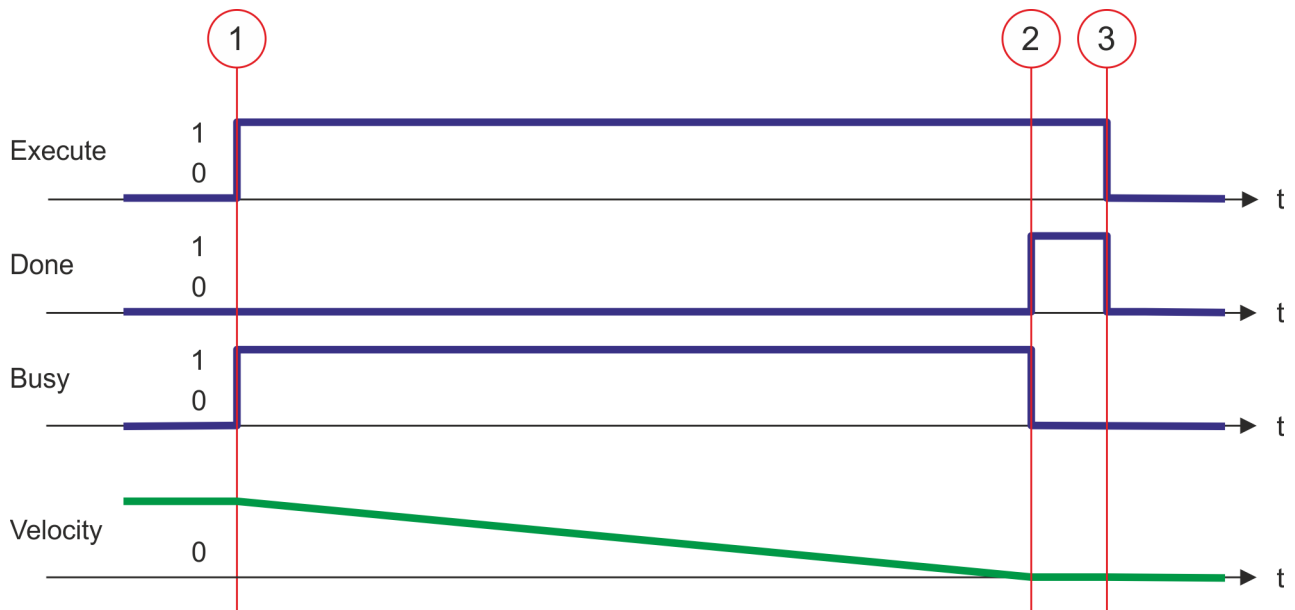
Achse anhalten

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.

6.3.7 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 <i>↪ Kapitel 8 "ErrorID - Zusätzliche Fehlerinformationen" auf Seite 203</i> |

PLCopen-State

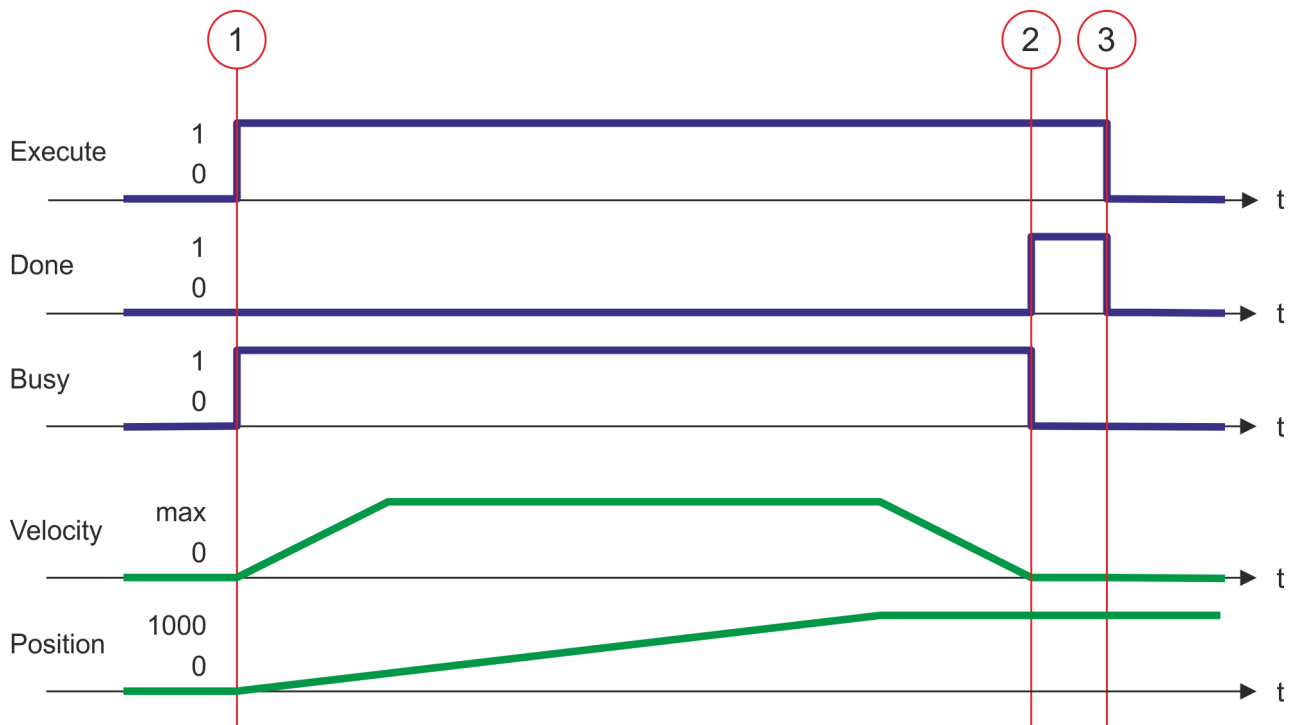
- Start des Auftrags in den PLCopen-States *Standstill*, *Discrete Motion*, *Synchronized Motion* und *Continuous Motion* möglich.
- MC_MoveRelative führt die Achse in den PLCopen-State *Discrete Motion* über.

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.

6.3.8 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 8 "ErrorID - Zusätzliche Fehlerinformationen" auf Seite 203</i> |

PLCopen-State

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

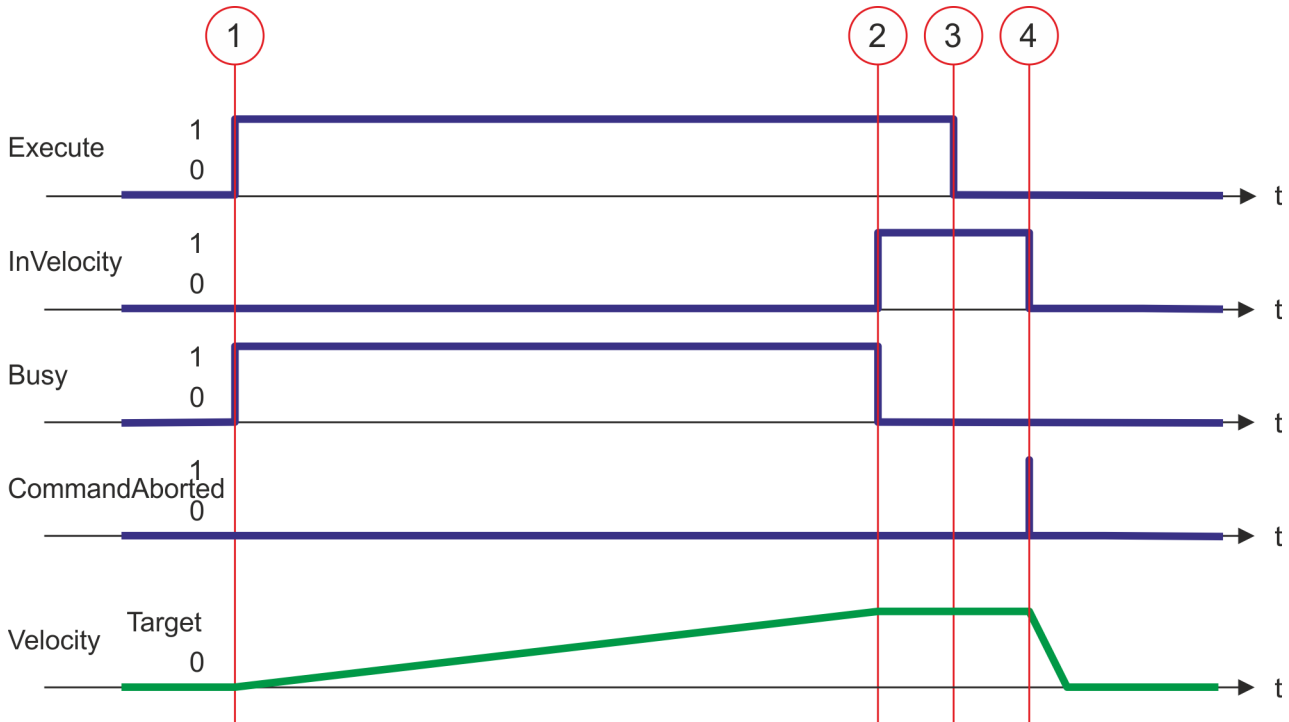
Achse mit Geschwindigkeitsvorgabe verfahren

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.

6.3.9 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 8 "ErrorID - Zusätzliche Fehlerinformationen" auf Seite 203</i> |

PLCopen-State

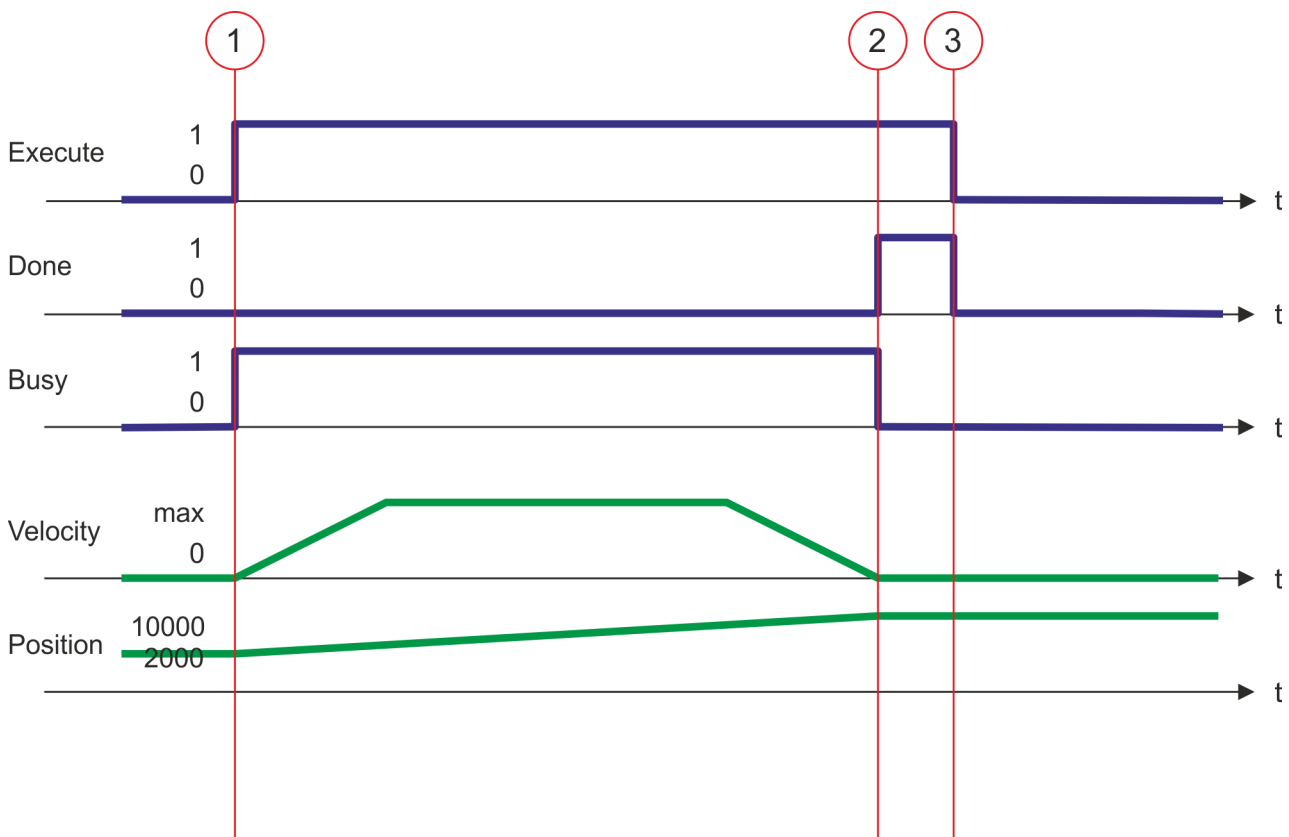
- Start des Auftrags in den PLCopen-States *Standstill*, *Discrete Motion*, *Synchronized Motion* und *Continuous Motion* möglich.
- MC_MoveVelocity führt die Achse in den PLCopen-State *Discrete Motion* über.

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.

6.3.10 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 <i>🔗 Kapitel 8 "ErrorID - Zusätzliche Fehlerinformationen" auf Seite 203</i> |

PLCopen-State

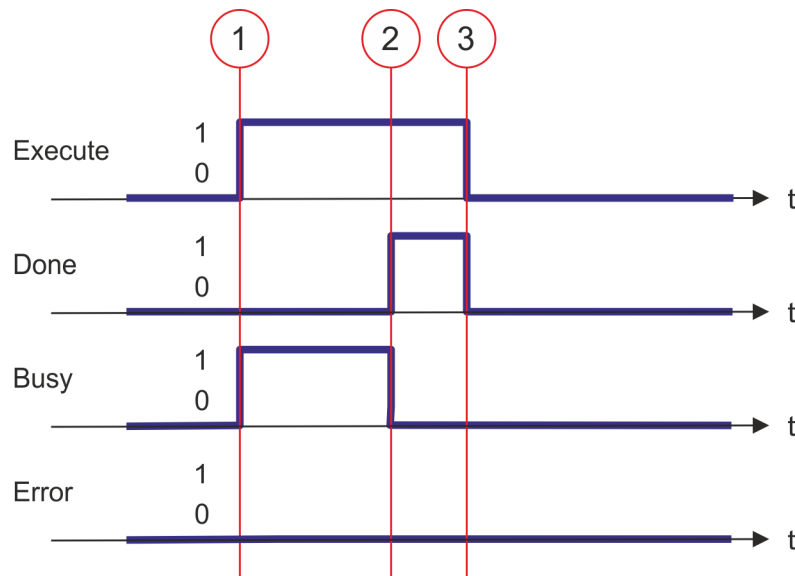
- Start des Auftrags im PLCopen-State *ErrorStop* möglich.
- MC_Reset führt die Achse in Abhängigkeit von MC_Power entweder in den PLCopen-State *Standstill* (Aufruf von MC_Power mit *Enable* = TRUE) oder *Disabled* (Aufruf von MC_Power mit *Enable* = FALSE) über.

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.

6.3.11 FB 812 - MC_ReadStatus - PLCopen Status

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

Parameter

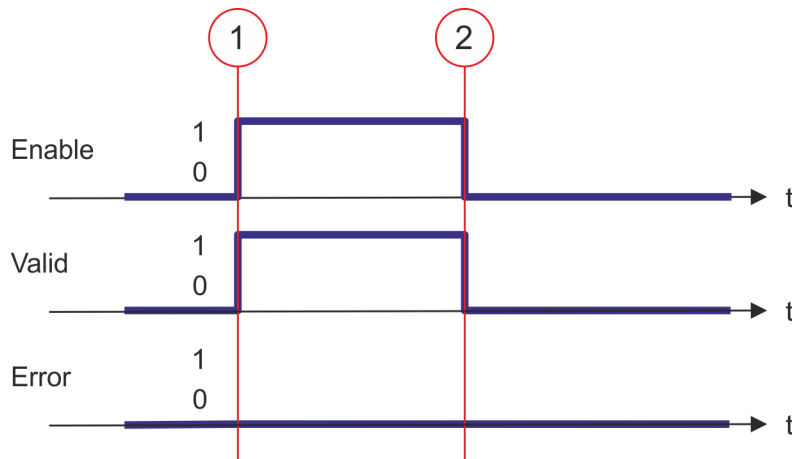
| Parameter | Deklaration | Datentyp | Beschreibung |
|--------------------|-------------|-------------|---|
| Axis | IN_OUT | MC_AXIS_REF | Referenz zur Slave-Achse |
| Enable | INPUT | BOOL | <ul style="list-style-type: none"> ■ Statusanzeige <ul style="list-style-type: none"> – TRUE: Der Status wird 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 8 "ErrorID - Zusätzliche Fehlerinformationen" auf Seite 203</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) |
| SynchronizedMotion | OUTPUT | BOOL | <ul style="list-style-type: none"> ■ Status Achse: Slave-Achse <ul style="list-style-type: none"> – TRUE: Achse ist eine Slave-Achse (MC_CamIn oder MC_GearIn 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.

6.3.12 FB 813 - MC_ReadAxisError - Fehler von Achse lesen

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

Parameter

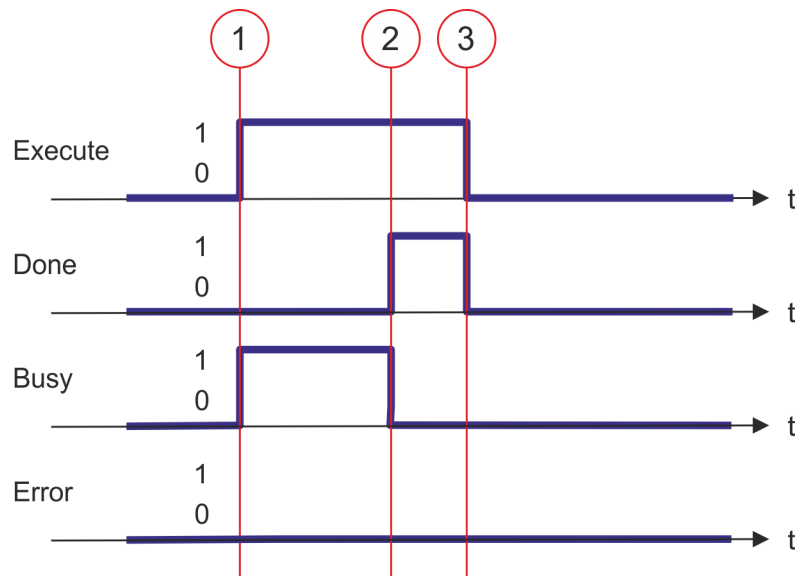
| Parameter | Deklaration | Datentyp | Beschreibung |
|-------------|-------------|-------------|---|
| Axis | IN_OUT | MC_AXIS_REF | Referenz zur Achse |
| Execute | INPUT | BOOL | <ul style="list-style-type: none"> ■ Reset Achse <ul style="list-style-type: none"> – Flanke 0-1: Achsfehler wird gelesen. |
| 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 8 "ErrorID - Zusätzliche Fehlerinformationen" auf Seite 203</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.

6.3.13 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 6.3.33 "PLCopen Parameter" auf Seite 193*

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 6.3.33 "PLCopen Parameter" auf Seite 193</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 8 "ErrorID - Zusätzliche Fehlerinformationen" auf Seite 203</i> |
| Value | OUTPUT | REAL | Wert des gelesenen Parameters |

PLCopen-State

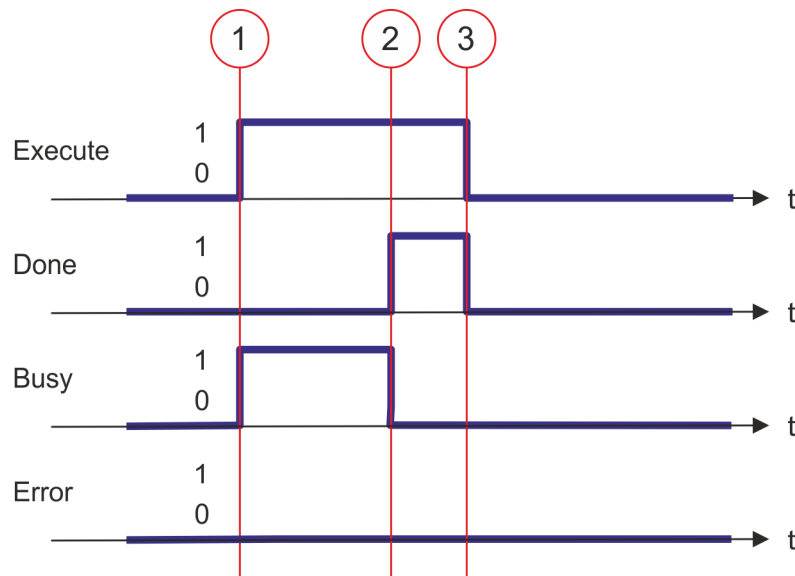
- Start des Auftrags in jedem PLCopen-State möglich.

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.

6.3.14 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 6.3.33 "PLCopen Parameter" auf Seite 193*

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 6.3.33 "PLCopen Parameter" auf Seite 193</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 8 "ErrorID - Zusätzliche Fehlerinformationen" auf Seite 203</i> |

PLCopen-State

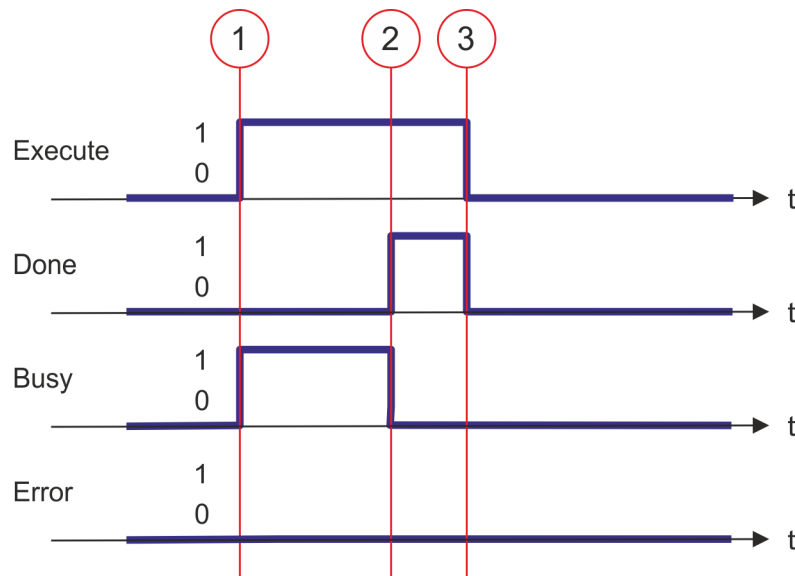
- Start des Auftrags in jedem PLCopen-State möglich.

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.

6.3.15 FB 816 - MC_ReadActualPosition - Aktuelle Position der Achse lesen

Beschreibung Mit MC_ReadActualPosition wird die aktuelle Position der Achse gelesen.

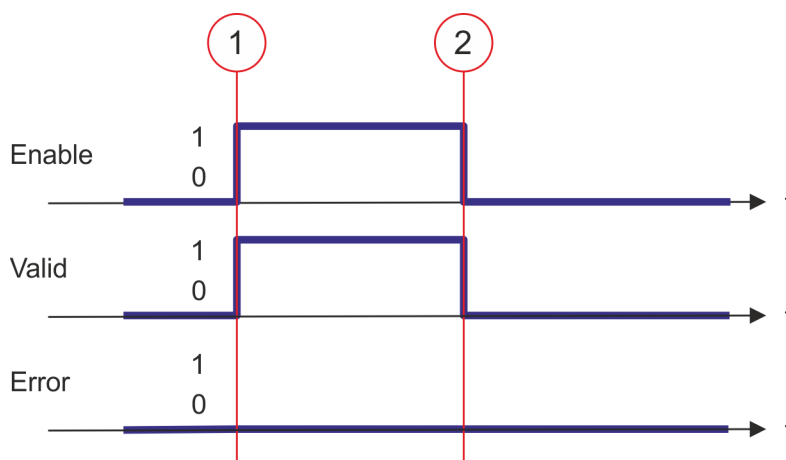
Parameter

| 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 ↗ Kapitel 8 "ErrorID - Zusätzliche Fehlerinformationen" auf Seite 203 |
| 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.

Komplexe Bewegungsaufgaben - PLCopen-Bausteine > FB 817 - MC_ReadActualVelocity - Aktuelle Geschwindigkeit der Achse lesen

6.3.16 FB 817 - MC_ReadActualVelocity - Aktuelle Geschwindigkeit der Achse lesen

Beschreibung Mit MC_ReadActualVelocity wird die aktuelle Geschwindigkeit der Achse gelesen.

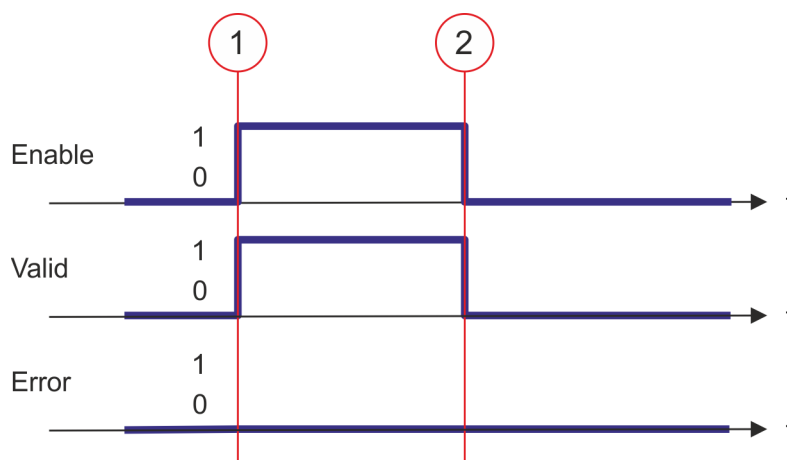
Parameter

| 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 8 "ErrorID - Zusätzliche Fehlerinformationen" auf Seite 203</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.

6.3.17 FB 818 - MC_ReadAxisInfo - Zusatzinformationen der Achse lesen

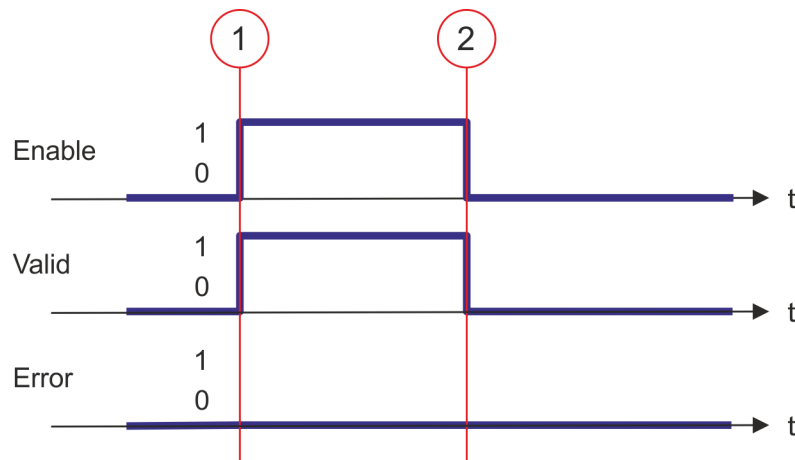
Beschreibung Mit MC_ReadAxisInfo werden einige Zusatzinformationen der Achse angezeigt.

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 8 "ErrorID - Zusätzliche Fehlerinformationen" auf Seite 203</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.

6.3.18 FB 819 - MC_ReadMotionState - Zustand Bewegungsauftrag lesen

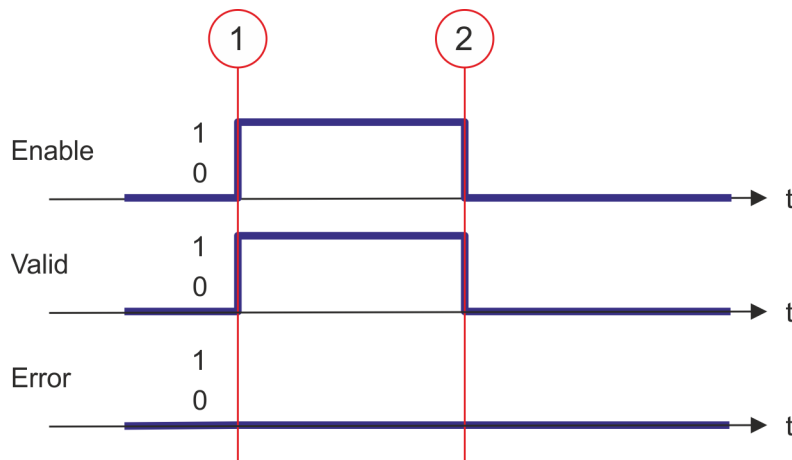
Beschreibung Mit MC_ReadMotionState wird der aktuelle Zustand des Bewegungsauftrags angezeigt.

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 8 "ErrorID - Zusätzliche Fehlerinformationen" auf Seite 203</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.

6.3.19 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 <i>☞ Kapitel 8 "ErrorID - Zusätzliche Fehlerinformationen" auf Seite 203</i> |
| 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 7.3 "Verhalten der Ein- und Ausgänge" auf Seite 201
- 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.

6.3.20 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 | <ul style="list-style-type: none"> ■ Status <ul style="list-style-type: none"> – TRUE: Auftrag erfolgreich durchgeführt. Die Erfassung der Achsposition wurde abgebrochen. |
| Busy | OUT | BOOL | <ul style="list-style-type: none"> ■ Status <ul style="list-style-type: none"> – TRUE: Auftrag ist in Bearbeitung. |
| 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 ↗ Kapitel 8 "ErrorID - Zusätzliche Fehlerinformationen" auf Seite 203 |



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.

6.3.21 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 6.3.33 "PLCopen Parameter" auf Seite 193*

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 6.3.33 "PLCopen Parameter" auf Seite 193</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 8 "ErrorID - Zusätzliche Fehlerinformationen" auf Seite 203</i> |
| Value | OUTPUT | BOOL | Wert des gelesenen Parameters |

PLCopen-State

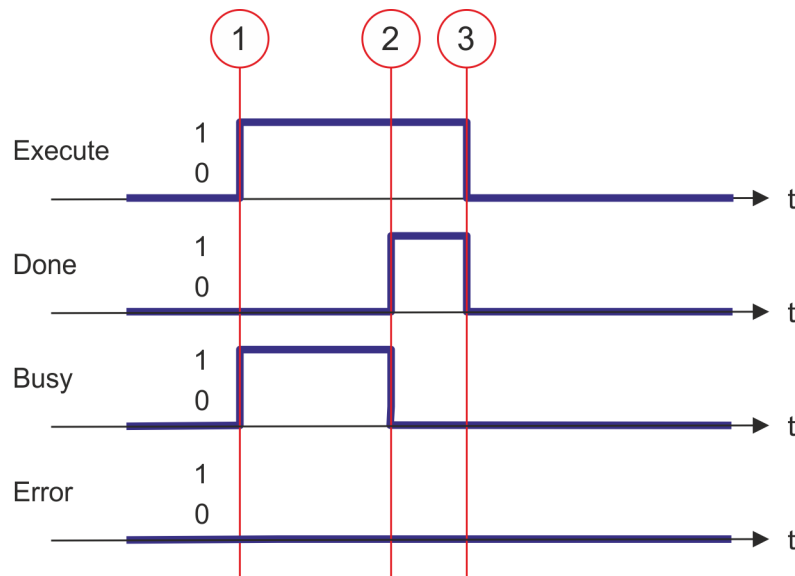
- Start des Auftrags in jedem PLCopen-State möglich.

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.

6.3.22 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 6.3.33 "PLCopen Parameter" auf Seite 193](#)

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 6.3.33 "PLCopen Parameter" auf Seite 193 |
| 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 8 "ErrorID - Zusätzliche Fehlerinformationen" auf Seite 203 |

PLCopen-State

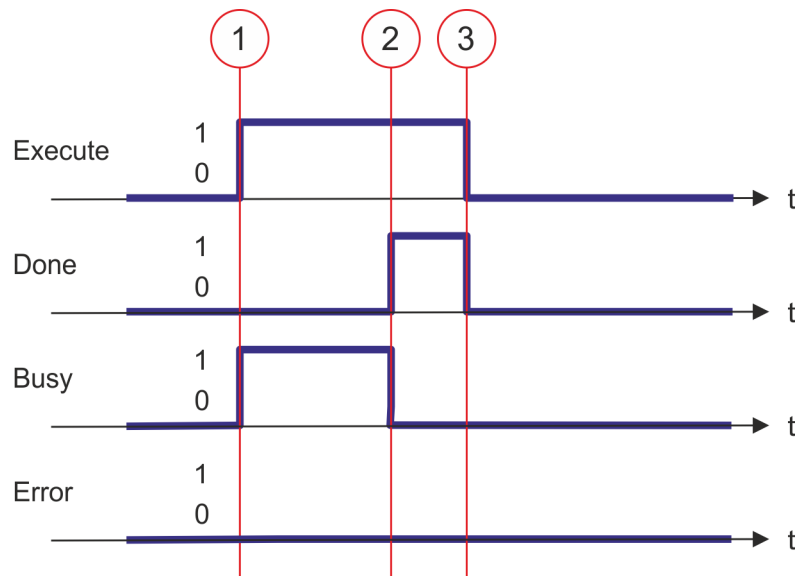
- Start des Auftrags in jedem PLCopen-State möglich.

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.

Komplexe Bewegungsaufgaben - PLCopen-Bausteine > FB 827 - VMC_ReadDWordParameter - Doppelwort-Parameter von Achse lesen

6.3.23 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 6.3.33 "PLCopen Parameter" auf Seite 193](#)

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. ↗ Kapitel 6.3.33 "PLCopen Parameter" auf Seite 193 |
| 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 8 "ErrorID - Zusätzliche Fehlerinformationen" auf Seite 203 |
| Value | OUTPUT | DWORD | Wert des gelesenen Parameters |

PLCopen-State

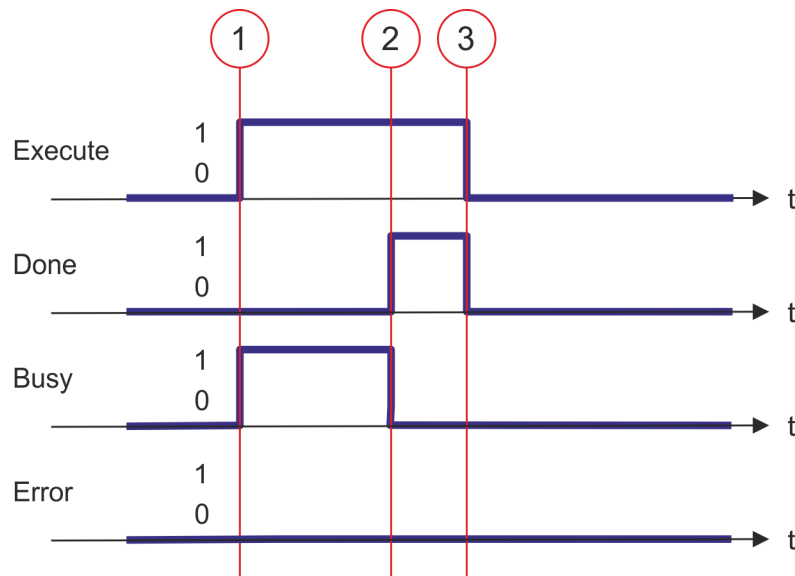
- Start des Auftrags in jedem PLCopen-State möglich.

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.

Komplexe Bewegungsaufgaben - PLCopen-Bausteine > FB 828 - VMC_WriteDWordParameter - Doppelwort-Parameter an Achse schreiben

6.3.24 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 6.3.33 "PLCopen Parameter" auf Seite 193*

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 6.3.33 "PLCopen Parameter" auf Seite 193</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 8 "ErrorID - Zusätzliche Fehlerinformationen" auf Seite 203</i> |

PLCopen-State

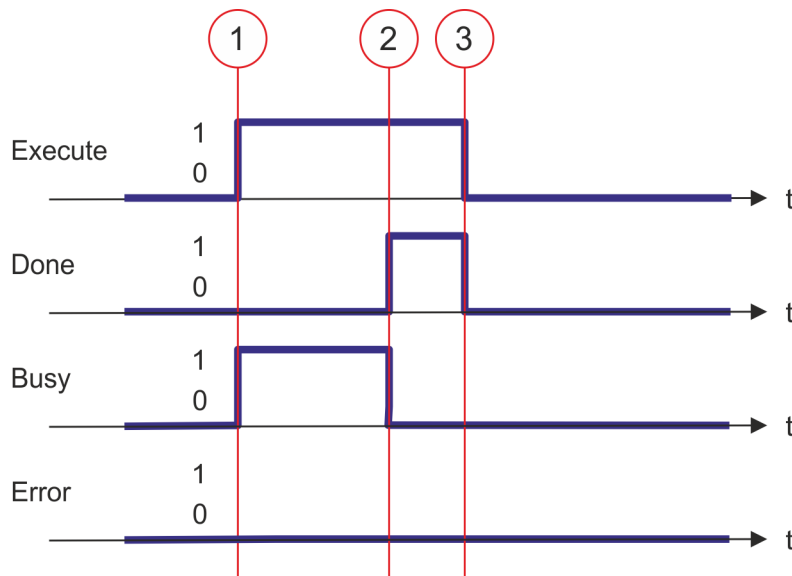
- Start des Auftrags in jedem PLCopen-State möglich.

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.

6.3.25 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 6.3.33 "PLCopen Parameter" auf Seite 193*

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 6.3.33 "PLCopen Parameter" auf Seite 193</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 8 "ErrorID - Zusätzliche Fehlerinformationen" auf Seite 203</i> |
| Value | OUTPUT | WORD | Wert des gelesenen Parameters |

PLCopen-State

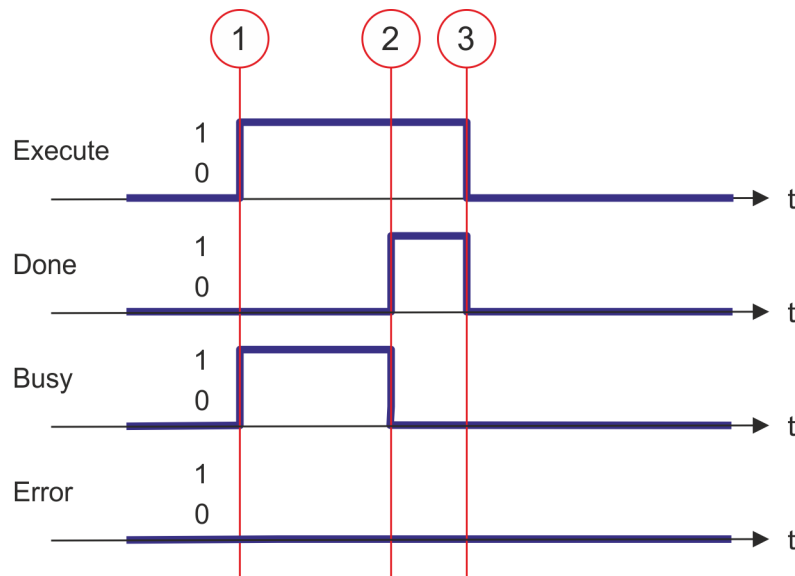
- Start des Auftrags in jedem PLCopen-State möglich.

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.

6.3.26 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 6.3.33 "PLCopen Parameter" auf Seite 193*

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 6.3.33 "PLCopen Parameter" auf Seite 193</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 8 "ErrorID - Zusätzliche Fehlerinformationen" auf Seite 203</i> |

PLCopen-State

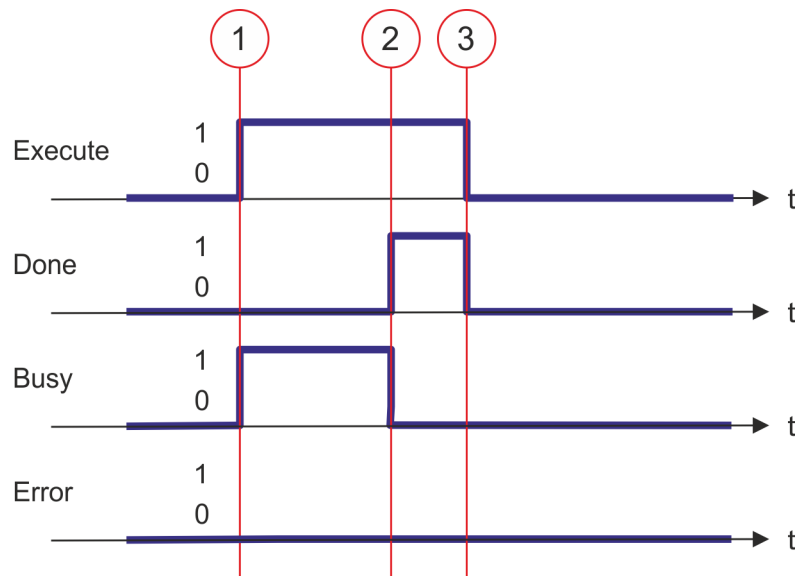
- Start des Auftrags in jedem PLCopen-State möglich.

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.

6.3.27 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 6.3.33 "PLCopen Parameter" auf Seite 193*

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 6.3.33 "PLCopen Parameter" auf Seite 193</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 8 "ErrorID - Zusätzliche Fehlerinformationen" auf Seite 203</i> |
| Value | OUTPUT | BYTE | Wert des gelesenen Parameters |

PLCopen-State

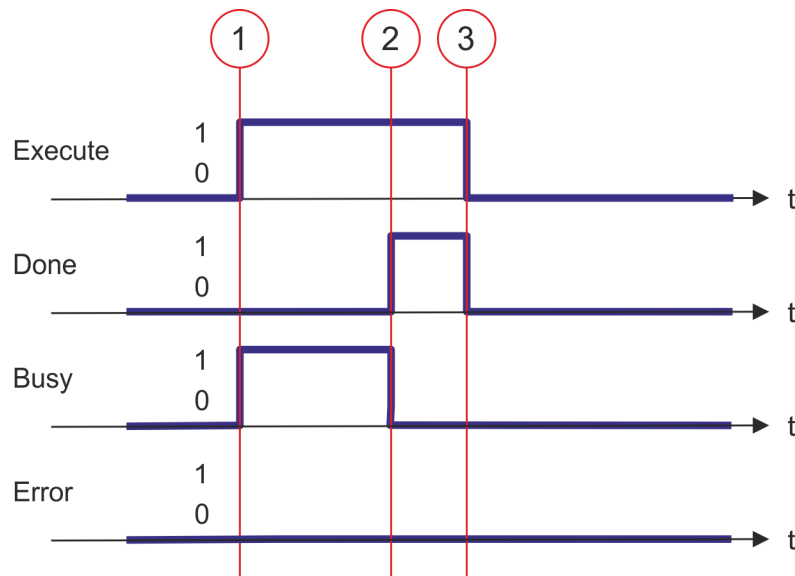
- Start des Auftrags in jedem PLCopen-State möglich.

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.

6.3.28 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 6.3.33 "PLCopen Parameter" auf Seite 193 |
| 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 8 "ErrorID - Zusätzliche Fehlerinformationen" auf Seite 203 |

PLCopen-State

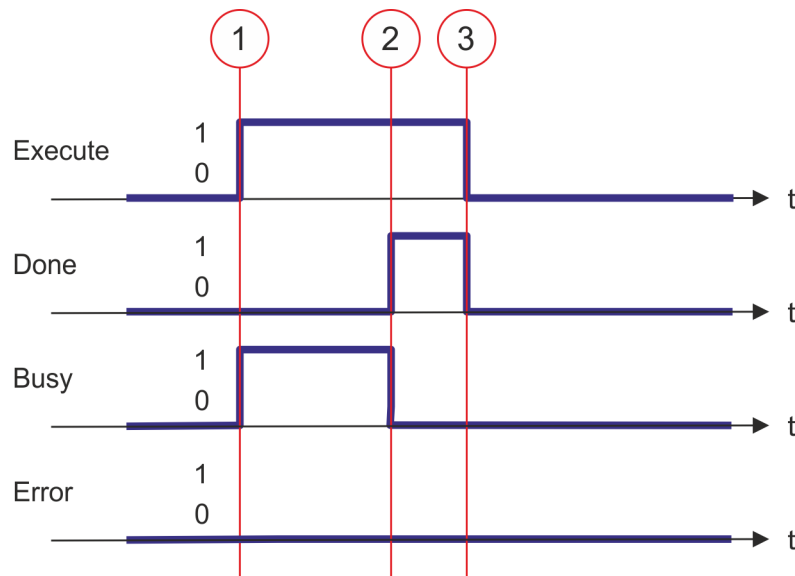
- Start des Auftrags in jedem PLCopen-State möglich.

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.

Komplexe Bewegungsaufgaben - PLCopen-Bausteine > FB 835 - VMC_HomeInit_LimitSwitch - Initialisierung Referenzfahrt auf Endschalter

6.3.29 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 8 "ErrorID - Zusätzliche Fehlerinformationen" auf Seite 203</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 |

Komplexe Bewegungsaufgaben - PLCopen-Bausteine > FB 836 - VMC_HomeInit_HomeSwitch - Initialisierung Referenzfahrt auf Referenzschalter

6.3.30 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 |
| SameDirIndex-Pulse | 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. |
| Velocity-SearchSwitch | IN | REAL | Geschwindigkeit für die Schaltersuche in [Anwendereinheiten/s] |
| VelocitySearch-Zero | 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 8 "ErrorID - Zusätzliche Fehlerinformationen" auf Seite 203</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 |

Komplexe Bewegungsaufgaben - PLCopen-Bausteine > FB 836 - VMC_HomeInit_HomeSwitch - Initialisierung Referenzfahrt auf Referenzschalter

| Homing Method | InitialDirection | WithIndexPulse | OnRisingEdge | SameDirIndexPulse |
|---------------|------------------|----------------|--------------|-------------------|
| 28 | negative | false | true | true |
| 28 | negative | false | false | false |
| 28 | negative | false | false | true |

6.3.31 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 8 "ErrorID - Zusätzliche Fehlerinformationen" auf Seite 203</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

Komplexe Bewegungsaufgaben - PLCopen-Bausteine > FB 837 - VMC_HomeInit_ZeroPulse - Initialisierung Referenzfahrt auf Null Impuls

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 |

6.3.32 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 8 "ErrorID - Zusätzliche Fehlerinformationen" auf Seite 203</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

6.3.33 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 |

6.3.34 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 6.3.21 "FB 825 - MC_ReadBoolParameter - Boolean-Parameter von Achse lesen"](#) auf Seite 170

2) Zugriff über [§ Kapitel 6.3.22 "FB 826 - MC_WriteBoolParameter - Boolean-Parameter an Achse schreiben"](#) auf Seite 172

3) Zugriff über [§ Kapitel 6.3.25 "FB 829 - VMC_ReadWordParameter - Wort-Parameter von Achse lesen"](#) auf Seite 178

4) Zugriff über [§ Kapitel 6.3.26 "FB 830 - VMC_WriteWordParameter - Wort-Parameter an Achse schreiben"](#) auf Seite 180

5) Zugriff über [§ Kapitel 6.3.23 "FB 827 - VMC_ReadDWordParameter - Doppelwort-Parameter von Achse lesen"](#) auf Seite 174

6) Zugriff über [§ Kapitel 6.3.24 "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 6.3.21 "FB 825 - MC_ReadBoolParameter - Boolean-Parameter von Achse lesen" auf Seite 170](#)

2) Zugriff über [§ Kapitel 6.3.22 "FB 826 - MC_WriteBoolParameter - Boolean-Parameter an Achse schreiben" auf Seite 172](#)

3) Zugriff über [§ Kapitel 6.3.25 "FB 829 - VMC_ReadWordParameter - Wort-Parameter von Achse lesen" auf Seite 178](#)

4) Zugriff über [§ Kapitel 6.3.26 "FB 830 - VMC_WriteWordParameter - Wort-Parameter an Achse schreiben" auf Seite 180](#)

5) Zugriff über [§ Kapitel 6.3.23 "FB 827 - VMC_ReadDWordParameter - Doppelwort-Parameter von Achse lesen" auf Seite 174](#)

6) Zugriff über [§ Kapitel 6.3.24 "FB 828 - VMC_WriteDWordParameter - Doppelwort-Parameter an Achse schreiben" auf Seite 176](#)

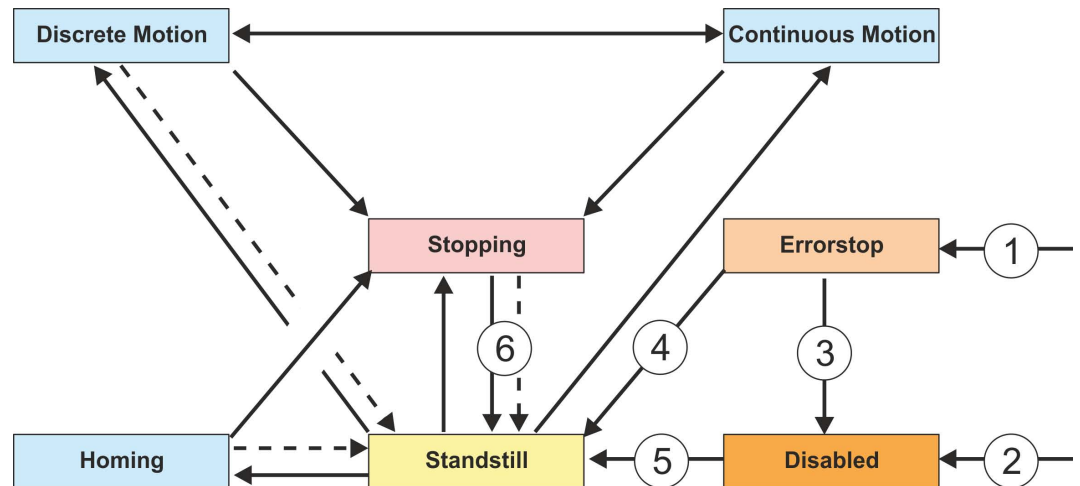
7 Zustände und Verhalten der Ausgänge

7.1 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 6.3.11 "FB 812 - MC_ReadStatus - PLCopen Status" auf Seite 153
- Parameter `PLCopenState` von ↗ Kapitel 6.2.2 "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

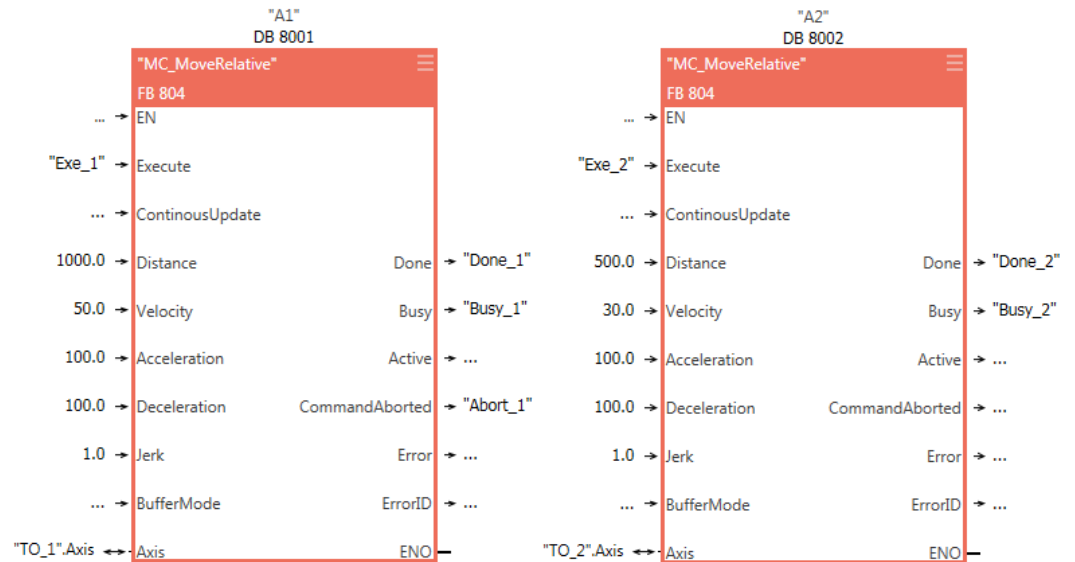
- Disabled
 - Grundzustand einer Achse.
 - Achse kann durch keinen Funktionsbaustein bewegt werden.
- Error Stop
 - Es ist ein Fehler an der Achse aufgetreten.
 - Achse wird gestoppt und ist für weitere Bewegungsaufgaben gesperrt.
 - Achse bleibt in diesem Zustand bis der Fehler behoben ist und ein RESET 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

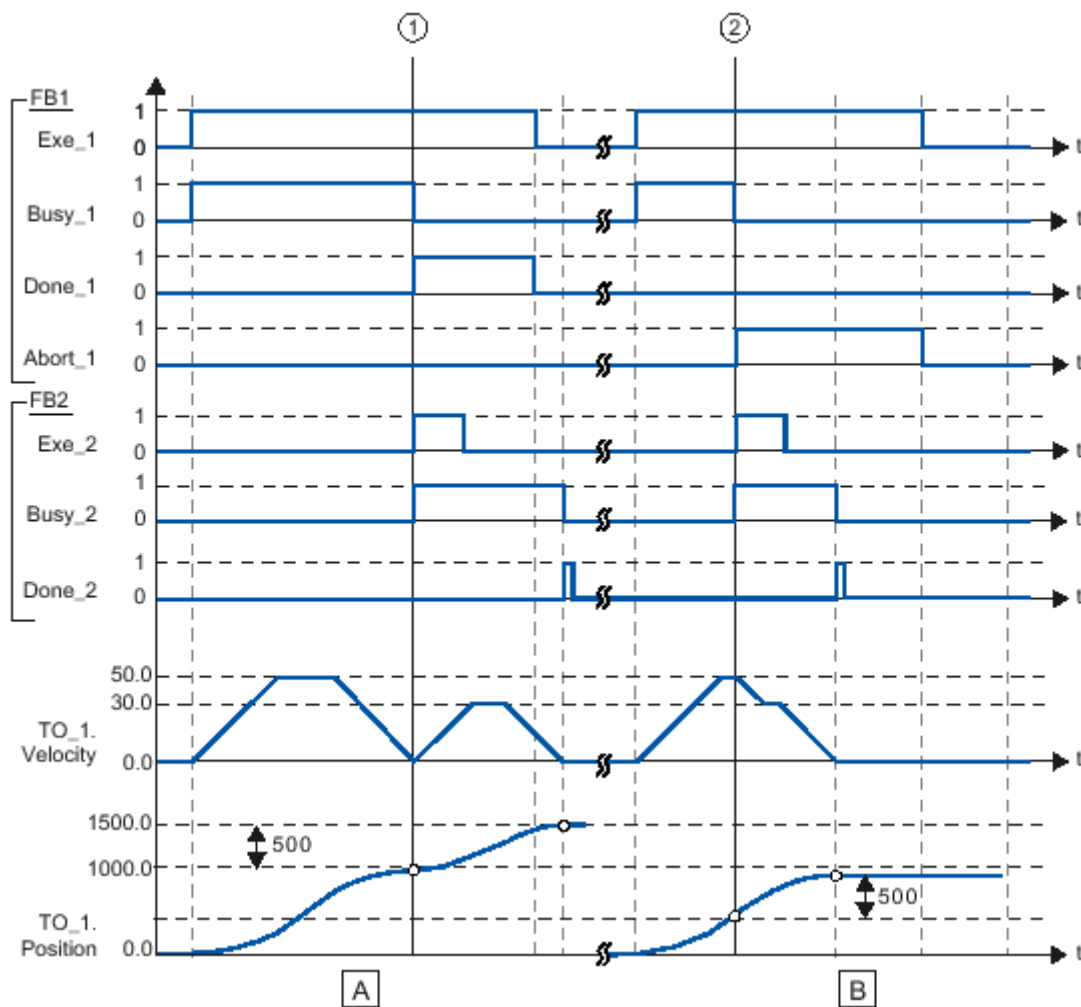
- Stopping
 - Achse wird aktuell gestoppt:
 - ↳ Kapitel 6.3.5 "FB 802 - MC_Stop - Achse stoppen" auf Seite 141
 - ↳ Kapitel 6.2.2 "FB 860 VMC_AxisControl - Control-Baustein Achskontrolle" auf Seite 132
 - Der Zustand *Stopping* ist aktiv solange ein Stop Kommando vom aktiv ist (*Execute* = 1). Auch wenn die Achse schon gestoppt ist. Danach wechselt der Zustand automatisch nach *Standstill*.
- Homing
 - Die Achse führt aktuell eine Referenzfahrt durch:
 - ↳ Kapitel 6.3.4 "FB 801 - MC_Home - Achse referenzieren" auf Seite 139
 - ↳ Kapitel 6.2.2 "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 6.3.9 "FB 808 - MC_MoveAbsolute - Achse auf absolute Position verfahren" auf Seite 149
 - ↳ Kapitel 6.3.7 "FB 804 - MC_MoveRelative - Achse relativ verfahren" auf Seite 145
 - ↳ Kapitel 6.3.6 "FB 803 - MC_Halt - Achse anhalten" auf Seite 143
 - ↳ Kapitel 6.2.2 "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 6.3.8 "FB 805 - MC_MoveVelocity - Achse verfahren mit konstanter Geschwindigkeit" auf Seite 147
 - ↳ Kapitel 6.2.2 "FB 860 VMC_AxisControl - Control-Baustein Achskontrolle" auf Seite 132

7.2 Ablöseverhalten von Bewegungsaufträgen

Beispiel

Das Ablöseverhalten von Bewegungsaufträgen wird nachfolgend am Beispiel von MC_MoveRelative erklärt. ↗ *Kapitel 6.3.7 "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.

7.3 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.

8 ErrorID - Zusätzliche Fehlerinformationen

| ErrorID | Beschreibung | Bemerkung |
|---------|---|-------------------------------------|
| 0x0000 | Kein Fehler | |
| 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> . | |
| 0x8013 | Unzulässiger Parameter der virtuellen Achse. | MC_ReadParameter |
| 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_WriteParameter |
| 0x801B | Parameter kann nicht geschrieben werden, Parameter ist schreibgeschützt. | MC_WriteParameter |
| 0x801C | Parameter Kommunikation mit unbekanntem Mode. | MC_Home, MC_WriteParameter |
| 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. | |

| ErrorID | Beschreibung | Bemerkung |
|---------|---|---|
| 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 |
| 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 "eingeschaltet" → 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 | Befehl wird von virtueller Achse nicht unterstützt. | |
| 0x8106 | PLCopen-State ist nicht definiert. | |
| 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) |
| 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. | |

| ErrorID | Beschreibung | Bemerkung |
|---------|--|--------------------------------|
| 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. | |
| 0x8341 | Ungültiger Wert in TriggerInput.Source. | |
| 0x8342 | Ungültiger Wert in TriggerInput.TriggerMode. | |
| 0x8350 | Ungültiger Wert in VelocitySearchSwitch. | Referenzfahrt, Initialisierung |
| 0x8351 | Ungültiger Wert in VelocitySearchZero. | Referenzfahrt, Initialisierung |
| 0x8352 | Ungültige Kombination von Eingängen. | Referenzfahrt, Initialisierung |
| 0x8400 | MC_Power: Unerwarteter Drive-State Drive-State <> Operation enabled | |
| 0x8401 | MC_Power: Unerwarteter Drive-State Drive-State = Quick stop active | |
| 0x8402 | MC_Power: Unerwarteter Drive-State Drive-State = Fault reaction active | |
| 0x8403 | MC_Power: Unerwarteter Drive-State Drive-State = Fault | |
| 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 |

| ErrorID | Beschreibung | Bemerkung |
|---------|--|---------------|
| 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> (\geq MinUserPos) | Init-Baustein |
| 0x850F | Falscher Wert in <i>MaxPosition</i> (\geq MaxUserPos) | Init-Baustein |
| 0x8603 | Fehler Referenzfahrt am Antrieb, Geschwindigkeit \neq 0. | MC_Home |
| 0x8604 | Fehler Referenzfahrt am Antrieb, Geschwindigkeit = 0. | MC_Home |