

# VIPA System 300S CPU 315SN/EC

**CPU | 315-4EC12 | Handbuch**

HB140 | CPU | 315-4EC12 | DE | 15-23

VIPA GmbH  
Ohmstr. 4  
91074 Herzogenaurach  
Telefon: +49 9132 744-0  
Telefax: +49 9132 744-1864  
E-Mail: [info@vipa.com](mailto:info@vipa.com)  
Internet: [www.vipa.com](http://www.vipa.com)

## Inhaltsverzeichnis

<b>1</b>	<b>Allgemeines</b> .....	<b>6</b>
1.1	Copyright © VIPA GmbH .....	6
1.2	Über dieses Handbuch.....	7
1.3	Sicherheitshinweise.....	8
<b>2</b>	<b>Grundlagen</b> .....	<b>10</b>
2.1	Sicherheitshinweis für den Benutzer.....	10
2.2	Arbeitsweise einer CPU.....	11
2.2.1	Allgemein.....	11
2.2.2	Programme .....	11
2.2.3	Operanden.....	12
2.3	CPU 315-4EC12.....	13
2.4	Allgemeine Daten.....	15
<b>3</b>	<b>Montage und Aufbaurichtlinien</b> .....	<b>17</b>
3.1	Einbaumaße.....	17
3.2	Montage Standard-Bus.....	18
3.3	Verdrahtung.....	19
3.4	Aufbaurichtlinien.....	22
<b>4</b>	<b>Hardwarebeschreibung</b> .....	<b>26</b>
4.1	Leistungsmerkmale.....	26
4.2	Aufbau.....	27
4.2.1	Allgemein.....	27
4.2.2	Schnittstellen.....	27
4.2.3	Speichermanagement.....	29
4.2.4	Steckplatz für Speichermedien.....	29
4.2.5	Batteriepufferung für Uhr und RAM.....	30
4.2.6	Betriebsartenschalter.....	30
4.2.7	LEDs.....	30
4.3	Technische Daten.....	33
<b>5</b>	<b>Einsatz CPU 315-4EC12</b> .....	<b>41</b>
5.1	Montage.....	41
5.2	Anlaufverhalten.....	41
5.3	Adressierung.....	42
5.3.1	Übersicht.....	42
5.3.2	Adressierung Rückwandbus Peripherie.....	42
5.4	Hardware-Konfiguration - CPU.....	43
5.5	Hardware-Konfiguration - I/O-Module.....	44
5.6	Hardware-Konfiguration - Ethernet-PG/OP-Kanal.....	45
5.7	Hardware-Konfiguration - Kommunikation.....	47
5.8	Einstellung Standard CPU-Parameter.....	47
5.8.1	Parametrierung über Siemens CPU.....	47
5.8.2	Parameter CPU.....	48
5.8.3	Parameter für DP.....	52
5.8.4	Parameter für MPI/DP .....	52
5.9	Einstellung VIPA-spezifische CPU-Parameter.....	53
5.9.1	Vorgehensweise.....	53
5.9.2	VIPA-spezifische Parameter.....	54
5.10	Projekt transferieren.....	57

5.10.1	Transfer über MPI.....	57
5.10.2	Transfer über Ethernet.....	58
5.10.3	Transfer über MMC.....	59
5.11	Zugriff auf integrierte Web-Seite.....	60
5.12	Betriebszustände.....	62
5.12.1	Übersicht.....	62
5.12.2	Funktionssicherheit.....	64
5.13	Urlöschen.....	65
5.14	Firmwareupdate.....	66
5.15	Rücksetzen auf Werkseinstellung.....	69
5.16	Steckplatz für Speichermedien.....	70
5.17	Speichererweiterung mit MCC.....	70
5.18	Erweiterter Know-how-Schutz.....	71
5.19	MMC-Cmd - Autobefehle.....	73
5.20	VIPA-spezifische Diagnose-Einträge.....	75
5.21	Mit Testfunktionen Variablen steuern und beobachten..	91
<b>6</b>	<b>Einsatz PtP-Kommunikation.....</b>	<b>93</b>
6.1	Schnelleinstieg.....	93
6.2	Prinzip der Datenübertragung.....	94
6.3	Einsatz der RS485-Schnittstelle für PtP.....	94
6.4	Parametrierung.....	97
6.4.1	FC/SFC 216 - SER_CFG.....	97
6.5	Kommunikation.....	101
6.5.1	Übersicht.....	101
6.5.2	FC/SFC 217 - SER_SND.....	101
6.5.3	FC/SFC 218 - SER_RCV.....	106
6.6	Protokolle und Prozeduren .....	108
6.7	Modbus - Funktionscodes .....	112
6.8	Modbus - Beispiel zur Kommunikation.....	117
<b>7</b>	<b>Einsatz PROFIBUS-Kommunikation.....</b>	<b>120</b>
7.1	Übersicht.....	120
7.2	Schnelleinstieg.....	120
7.3	Hardware-Konfiguration - CPU.....	121
7.4	Einsatz als PROFIBUS-DP-Master.....	122
7.5	Einsatz als PROFIBUS-DP-Slave.....	123
7.6	PROFIBUS-Aufbau Richtlinien.....	125
7.7	Inbetriebnahme und Anlaufverhalten.....	129
<b>8</b>	<b>Einsatz Ethernet-Kommunikation - Produktiv.....</b>	<b>130</b>
8.1	Grundlagen - Industrial Ethernet in der Automatisierung.....	130
8.2	Grundlagen - ISO/OSI-Schichtenmodell.....	131
8.3	Grundlagen - Begriffe.....	132
8.4	Grundlagen - Protokolle.....	133
8.5	Grundlagen - IP-Adresse und Subnetz.....	136
8.6	Schnelleinstieg.....	137
8.7	Montage und Inbetriebnahme.....	138
8.8	Hardware-Konfiguration - CPU.....	139
8.9	Siemens S7-Verbindungen projektieren.....	139
8.10	Offene Kommunikation projektieren.....	146

8.11	NCM-Diagnose - Hilfe zur Fehlersuche.....	149
<b>9</b>	<b>Einsatz Ethernet-Kommunikation - EtherCAT .....</b>	<b>152</b>
9.1	Grundlagen EtherCAT .....	152
9.1.1	Allgemeines.....	152
9.1.2	EtherCAT Zustandsmaschine.....	153
9.1.3	CoE - CANopen over Ethernet.....	155
9.1.4	ESI-Dateien.....	156
9.2	Inbetriebnahme und Anlaufverhalten.....	156
9.2.1	Montage und Inbetriebnahme.....	156
9.2.2	Anlaufverhalten.....	156
9.3	Hardware-Konfiguration - CPU.....	157
9.4	EtherCAT Diagnose.....	161
9.4.1	Diagnose über den <i>SPEED7 EtherCAT Manager</i> .....	161
9.4.2	Diagnose zur Laufzeit im Anwenderprogramm (OB 1, SFB 52).....	161
9.4.3	Diagnose über Systemzustandslisten - SZL.....	165
9.4.4	Diagnose über OB-Startinformationen.....	166
9.4.5	Diagnose über NCM-Diagnose.....	166
9.4.6	Diagnose über Diagnosepuffer CPU bzw. CP.....	166
9.4.7	Diagnose über Status-LEDs.....	166
9.5	Alarmverhalten.....	167
9.5.1	Übersicht.....	167
9.5.2	Alarmtypen.....	168
9.6	Systemeigenschaften.....	178
9.7	Firmwareupdate.....	179
9.8	Zugriff auf das Objektverzeichnis.....	179
9.8.1	Übersicht.....	179
9.8.2	FB 52 - Read SDO - Lesezugriff auf Objektverzeichnis.....	180
9.8.3	FB 53 - Write SDO - Schreibzugriff auf Objektverzeichnis.....	183
9.9	Objekt-Verzeichnis.....	186
9.9.1	Objektübersicht.....	186
9.9.2	CoE Communication Area Objects: 0x1000-0x1FFF..	187
9.9.3	Generic Master Objects: 0x2000-0x20FF.....	190
9.9.4	Distributed Clocks Objects: 0x2100-0x21FF.....	194
9.9.5	Slave specific objects.....	194
9.9.6	CoE Device Area Objects: 0xF000-0xFFFF.....	199

# 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](mailto: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](mailto: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](mailto:support@vipa.de)

**1.2 Über dieses Handbuch**

**Zielsetzung und Inhalt**

Das Handbuch beschreibt die SPEED7 CPU 315-4EC12 aus dem System 300S von VIPA. Beschrieben wird Aufbau, Projektierung und Anwendung.

Produkt	Best.-Nr.	ab Stand:			
		CPU-HW	CPU-FW	DPM-FW	CP-FW
CPU 315SN/EC	315-4EC12	01	V3.6.2	V3.3.0	V1.0.0

**Zielgruppe**

Das Handbuch ist geschrieben für Anwender mit Grundkenntnissen in der Automatisierungstechnik.

**Aufbau des Handbuchs**

Das Handbuch ist in Kapitel gegliedert. Jedes Kapitel beschreibt eine abgeschlossene Thematik.

**Orientierung im Dokument**

Als Orientierungshilfe stehen im Handbuch zur Verfügung:

- Gesamt-Inhaltsverzeichnis am Anfang des Handbuchs
- Verweise mit Seitenangabe

**Verfügbarkeit**

Das Handbuch ist verfügbar in:

- gedruckter Form auf Papier
- in elektronischer Form als PDF-Datei (Adobe Acrobat Reader)

**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*

### 1.3 Sicherheitshinweise

**Bestimmungsgemäße Verwendung**

Das System ist konstruiert und gefertigt für:

- Kommunikation und Prozesskontrolle
- Allgemeine Steuerungs- und Automatisierungsaufgaben
- den industriellen Einsatz
- den Betrieb innerhalb der in den technischen Daten spezifizierten Umgebungsbedingungen
- den Einbau in einen Schaltschrank

**GEFAHR!**

Das Gerät ist nicht zugelassen für den Einsatz  
– in explosionsgefährdeten Umgebungen (EX-Zone)

**Dokumentation**

Handbuch zugänglich machen für alle Mitarbeiter in

- Projektierung
- Installation
- Inbetriebnahme
- Betrieb

**VORSICHT!**

**Vor Inbetriebnahme und Betrieb der in diesem Handbuch beschriebenen Komponenten unbedingt beachten:**

- Änderungen am Automatisierungssystem nur im spannungslosen Zustand vornehmen!
- Anschluss und Änderung nur durch ausgebildetes Elektro-Fachpersonal
- Nationale Vorschriften und Richtlinien im jeweiligen Verwenderland beachten und einhalten (Installation, Schutzmaßnahmen, EMV ...)

**Entsorgung**

**Zur Entsorgung des Geräts nationale Vorschriften beachten!**

## 2 Grundlagen

### 2.1 Sicherheitshinweis für den Benutzer

#### Handhabung elektrostatisch gefährdeter Baugruppen

VIPA-Baugruppen sind mit hochintegrierten Bauelementen in MOS-Technik bestückt. Diese Bauelemente sind hoch empfindlich gegenüber Überspannungen, die z.B. bei elektrostatischer Entladung entstehen. Zur Kennzeichnung dieser gefährdeten Baugruppen wird nachfolgendes Symbol verwendet:



Das Symbol befindet sich auf Baugruppen, Baugruppenträgern oder auf Verpackungen und weist so auf elektrostatisch gefährdete Baugruppen hin. Elektrostatisch gefährdete Baugruppen können durch Energien und Spannungen zerstört werden, die weit unterhalb der Wahrnehmungsgrenze des Menschen liegen. Hantiert eine Person, die nicht elektrisch entladen ist, mit elektrostatisch gefährdeten Baugruppen, können Spannungen auftreten und zur Beschädigung von Bauelementen führen und so die Funktionsweise der Baugruppen beeinträchtigen oder die Baugruppe unbrauchbar machen. Auf diese Weise beschädigte Baugruppen werden in den wenigsten Fällen sofort als fehlerhaft erkannt. Der Fehler kann sich erst nach längerem Betrieb einstellen. Durch statische Entladung beschädigte Bauelemente können bei Temperaturänderungen, Erschütterungen oder Lastwechseln zeitweilige Fehler zeigen. Nur durch konsequente Anwendung von Schutzeinrichtungen und verantwortungsbewusste Beachtung der Handhabungsregeln lassen sich Funktionsstörungen und Ausfälle an elektrostatisch gefährdeten Baugruppen wirksam vermeiden.

#### Versenden von Baugruppen

Verwenden Sie für den Versand immer die Originalverpackung.

#### Messen und Ändern von elektrostatisch gefährdeten Baugruppen

Bei Messungen an elektrostatisch gefährdeten Baugruppen sind folgende Dinge zu beachten:

- Potenzialfreie Messgeräte sind kurzzeitig zu entladen.
- Verwendete Messgeräte sind zu erden.

Bei Änderungen an elektrostatisch gefährdeten Baugruppen ist darauf zu achten, dass ein geerdeter LötKolben verwendet wird.



#### **VORSICHT!**

Bei Arbeiten mit und an elektrostatisch gefährdeten Baugruppen ist auf ausreichende Erdung des Menschen und der Arbeitsmittel zu achten.

## 2.2 Arbeitsweise einer CPU

### 2.2.1 Allgemein

Die CPU enthält einen Standardprozessor mit internem Programmspeicher. In Verbindung mit der integrierten SPEED7-Technologie erhalten Sie ein leistungsfähiges Gerät zur Prozessautomatisierung innerhalb der System 300S Familie. In einer CPU gibt es folgende Arbeitsweisen:

- zyklische Bearbeitung
- zeitgesteuerte Bearbeitung
- alarmgesteuerte Bearbeitung
- Bearbeitung nach Priorität

#### zyklische Bearbeitung

Die **zyklische** Bearbeitung stellt den Hauptanteil aller Vorgänge in der CPU. In einem endlosen Zyklus werden die gleichen Bearbeitungsfolgen wiederholt.

#### zeitgesteuerte Bearbeitung

Erfordern Prozesse in konstanten Zeitabschnitten Steuersignale, so können Sie neben dem zyklischen Ablauf **zeitgesteuert** bestimmte Aufgaben durchführen z.B. zeitunkritische Überwachungsfunktionen im Sekundenraster.

#### alarmgesteuerte Bearbeitung

Soll auf ein Prozesssignal besonders schnell reagiert werden, so ordnen Sie diesem einen **alarmgesteuerten** Bearbeitungsabschnitt zu. Ein Alarm kann in Ihrem Programm eine Bearbeitungsfolge aktivieren.

#### Bearbeitung nach Priorität

Die oben genannten Bearbeitungsarten werden von der CPU nach Wichtigkeitsgrad behandelt (**Priorität**). Da auf ein Zeit- oder Alarmergebnis schnell reagiert werden muss, unterbricht die CPU zur Bearbeitung dieser hochpriorären Ereignisse die zyklische Bearbeitung, reagiert auf diese Ereignisse und setzt danach die zyklische Bearbeitung wieder fort. Die zyklische Bearbeitung hat daher die niedrigste Priorität.

### 2.2.2 Programme

Das in jeder CPU vorhandene Programm unterteilt sich in:

- Systemprogramm
- Anwenderprogramm

#### Systemprogramm

Das Systemprogramm organisiert alle Funktionen und Abläufe der CPU, die nicht mit einer spezifischen Steuerungsaufgabe verbunden sind.

#### Anwenderprogramm

Hier finden Sie alle Funktionen, die zur Bearbeitung einer spezifischen Steuerungsaufgabe erforderlich sind. Schnittstellen zum Systemprogramm stellen die Operationsbausteine zur Verfügung.

### 2.2.3 Operanden

Die CPU stellt Ihnen für das Programmieren folgende Operandenbereiche zur Verfügung:

- Prozessabbild und Peripherie
- Merker
- Zeiten und Zähler
- Datenbausteine

#### Prozessabbild und Peripherie

Auf das Prozessabbild der Aus- und Eingänge PAA/PAE kann Ihr Anwenderprogramm sehr schnell zugreifen. Sie haben Zugriff auf folgende Datentypen:

- Einzelbits
- Bytes
- Wörter
- Doppelwörter

Sie können mit Ihrem Anwenderprogramm über den Bus direkt auf Peripheriebaugruppen zugreifen. Folgende Datentypen sind möglich:

- Bytes
- Wörter
- Blöcke

#### Merker

Der Merkerbereich ist ein Speicherbereich, auf den Sie über Ihr Anwenderprogramm mit entsprechenden Operationen zugreifen können. Verwenden Sie den Merkerbereich für oft benötigte Arbeitsdaten.

Sie können auf folgende Datentypen zugreifen:

- Einzelbits
- Bytes
- Wörter
- Doppelwörter

#### Zeiten und Zähler

Sie können mit Ihrem Anwendungsprogramm eine Zeitzelle mit einem Wert zwischen 10ms und 9990s laden. Sobald Ihr Anwenderprogramm eine Startoperation ausführt, wird dieser Zeitwert um ein durch Sie vorgegebenes Zeitraster dekrementiert, bis Null erreicht wird.

Für den Einsatz von Zählern können Sie Zählerzellen mit einem Anfangswert laden (max. 999) und diesen hinauf- bzw. herunterzählen.

#### Datenbausteine

Ein Datenbaustein enthält Konstanten bzw. Variablen im Byte-, Wort- oder Doppelwortformat. Mit Operanden können Sie immer auf den aktuellen Datenbaustein zugreifen.

Sie haben Zugriff auf folgende Datentypen:

- Einzelbits
- Bytes
- Wörter
- Doppelwörter

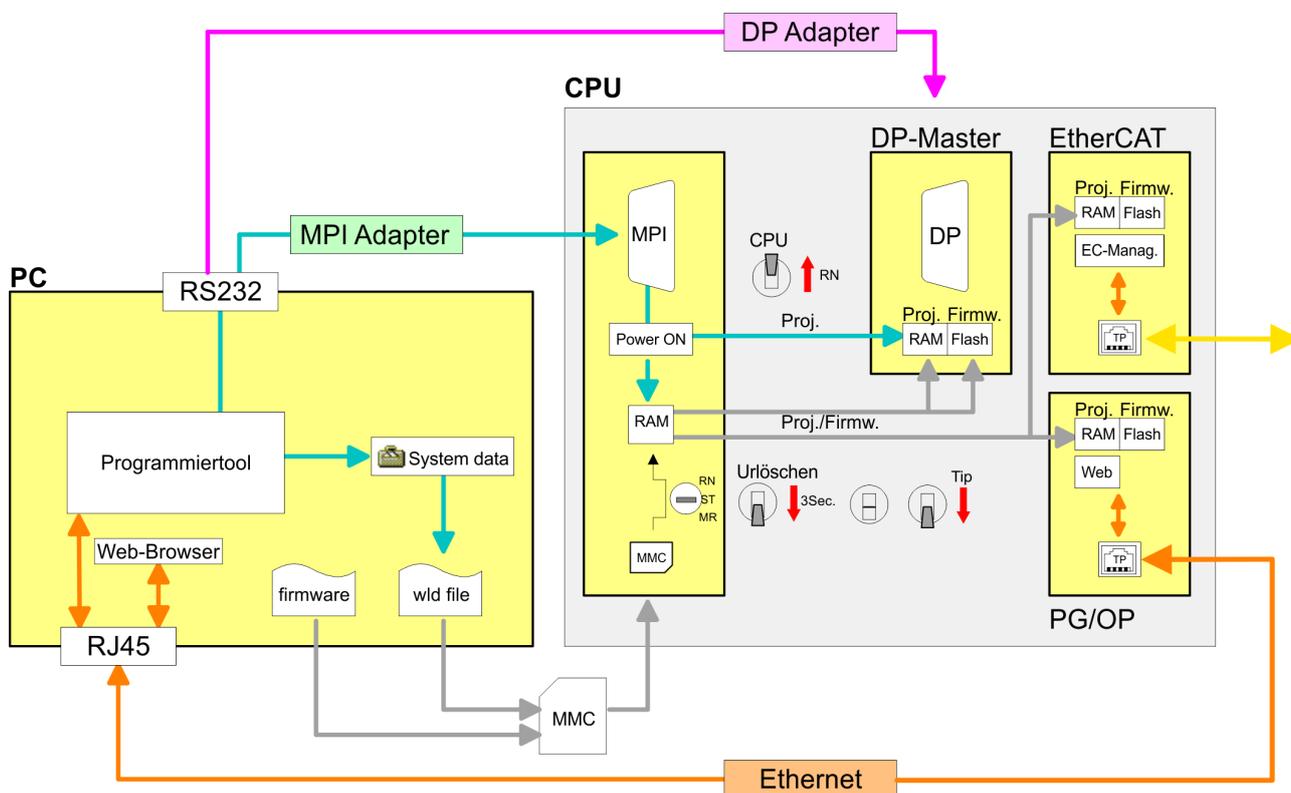
## 2.3 CPU 315-4EC12

### Übersicht

Die CPU 315-4EC12 basiert auf der SPEED7-Technologie. Hierbei wird die CPU durch Coprozessoren im Bereich Programmierung und Kommunikation unterstützt und erhält somit eine Leistungssteigerung, so dass diese höchsten Anforderungen genügt.

- Programmiert wird die CPU in STEP®7 von Siemens. Hierzu können Sie den SIMATIC Manager von Siemens verwenden. Hierbei kommt der Befehlssatz der S7-400 von Siemens zum Einsatz.
- Module und CPUs aus dem System 300S von VIPA und Siemens können als Mischkonfiguration am Bus eingesetzt werden.
- Das Anwenderprogramm wird im batteriegepufferten RAM oder auf einem zusätzlich steckbaren MMC-Speichermodul gespeichert.
- Projektiert wird die CPU als CPU 315-2PN/DP (6ES7 315-2EH14-0AB0 V3.2) von Siemens.

### Zugriffsmöglichkeiten



Bitte verwenden Sie zur Projektierung dieser CPU von VIPA immer die **CPU 315-2PN/DP (6ES7 315-2EH14-0AB0 V3.2)** von Siemens aus dem Hardware-Katalog. Zur Projektierung werden fundierte Kenntnisse im Umgang mit dem entsprechenden Siemens Projektierwerkzeug vorausgesetzt!

<b>Speicher</b>	<p>Die CPU hat einen Speicher integriert. Angaben über die Speicherkapazität finden Sie auf der Frontseite Ihrer CPU. Der Speicher gliedert sich in folgende Teile:</p> <ul style="list-style-type: none"><li>■ Ladespeicher 2MByte</li><li>■ Codespeicher (50% des Arbeitsspeichers)</li><li>■ Datenspeicher (50% des Arbeitsspeichers)</li><li>■ Arbeitsspeicher 1MByte<ul style="list-style-type: none"><li>– Sie haben die Möglichkeit den Arbeitsspeicher mittels einer MCC Speichererweiterungskarte bis zur maximal aufgedruckten Kapazität 2MByte zu erweitern.</li></ul></li></ul>
<b>Integrierter PROFIBUS-DP-Master/Slave bzw. PtP-Funktionalität</b>	<p>Die CPU besitzt eine PROFIBUS/PtP-Schnittstelle mit fixer Pinbelegung. Nach dem Umräumen ist diese Schnittstelle deaktiviert. Durch entsprechende Projektierung können Sie folgende Funktionalitäten für diese Schnittstelle aktivieren:</p> <ul style="list-style-type: none"><li>■ PROFIBUS-DP-Master-Betrieb: Projektierung erfolgt über das PROFIBUS-Submodul mit <i>"Betriebsart"</i> Master in der Hardware-Konfiguration.</li><li>■ PROFIBUS-DP-Slave-Betrieb: Projektierung erfolgt über das PROFIBUS-Submodul mit <i>"Betriebsart"</i> Slave in der Hardware-Konfiguration.</li><li>■ PtP-Funktionalität: Projektierung erfolgt in Form eines virtuellen PROFIBUS Master-Systems unter Einbindung der VIPA SPEEDBUS.GSD.</li></ul>
<b>Integrierter EtherCAT-Master</b>	<p>Die CPU hat einen EtherCAT-Master integriert. Die Projektierung des EtherCAT-Masters erfolgt im Siemens SIMATIC Manager in Form des virtuellen PROFINET IO Devices <i>"EtherCAT-Netzwerk"</i>. Das <i>"EtherCAT-Netzwerk"</i> ist mittels GSDML im Hardware-Katalog zu installieren und kann mit dem VIPA-Tool <i>SPEED7 EtherCAT Manager</i> konfiguriert werden.</p>
<b>Integrierter Ethernet-PG/OP-Kanal</b>	<p>Auf der CPU befindet sich eine Ethernet-Schnittstelle für PG/OP-Kommunikation. Nach der Zuweisung von IP-Adress-Parametern über Ihr Projektier-Tool können Sie über die <i>"Zielsystem"</i>-Funktionen den Ethernet-PG/OP-Kanal direkt ansprechen und Ihre CPU programmieren bzw. fernwarten. Sie haben auch die Möglichkeit über diese Verbindungen mit einer Visualisierungs-Software auf die CPU zuzugreifen.</p>
<b>Betriebssicherheit</b>	<ul style="list-style-type: none"><li>■ Anschluss über Federzugklemmen an Frontstecker</li><li>■ Aderquerschnitt 0,08...2,5mm<sup>2</sup></li><li>■ Vollisolierung der Verdrahtung bei Modulwechsel</li><li>■ Potenzialtrennung aller Peripherie-Module zum Rückwandbus</li></ul>
<b>Aufbau/Maße</b>	<p>Maße Grundgehäuse:</p> <ul style="list-style-type: none"><li>■ 2fach breit: (BxHxT) in mm: 80x125x120</li></ul>
<b>Integriertes Netzteil</b>	<p>Die CPU hat ein Netzteil integriert. Das Netzteil ist mit DC 24V zu versorgen. Über die Versorgungsspannung werden neben der internen Elektronik auch die angeschlossenen Module über den Rückwandbus versorgt. Das Netzteil ist gegen Verpolung und Überstrom geschützt.</p>

## 2.4 Allgemeine Daten

### Konformität und Approbation

Konformität		
CE	2006/95/EG	Niederspannungsrichtlinie
	2004/108/EG	EMV-Richtlinie
Approbation		
UL	UL 508	Zulassung für USA und Kanada
Sonstiges		
RoHS	2011/65/EU	Produkte bleifrei; Richtlinie zur Beschränkung der Verwendung bestimmter gefährlicher Stoffe in Elektro- und Elektronikgeräten

### Personenschutz und Geräteschutz

Schutzart	-	IP20
Potenzialtrennung		
Zum Feldbus	-	Galvanisch entkoppelt
Zur Prozessebene	-	Galvanisch entkoppelt
Isolationsfestigkeit		-
Isolationsspannung gegen Bezugserde		
Eingänge / Ausgänge	-	AC / DC 50V, bei Prüfspannung AC 500V
Schutzmaßnahmen	-	gegen Kurzschluss

### Umgebungsbedingungen gemäß EN 61131-2

Klimatisch		
Lagerung /Transport	EN 60068-2-14	-25...+70°C
Betrieb		
Horizontaler Einbau	EN 61131-2	0...+60°C
Vertikaler Einbau	EN 61131-2	0...+60°C
Luftfeuchtigkeit	EN 60068-2-30	RH1 (ohne Betauung, relative Feuchte 10 ... 95%)
Verschmutzung	EN 61131-2	Verschmutzungsgrad 2
Mechanisch		
Schwingung	EN 60068-2-6	1g, 9Hz ... 150Hz
Schock	EN 60068-2-27	15g, 11ms

Montagebedingungen		
Einbauort	-	Im Schaltschrank
Einbaulage	-	Horizontal und vertikal

EMV	Norm	Bemerkungen	
Störaussendung	EN 61000-6-4	Class A (Industriebereich)	
Störfestigkeit Zone B	EN 61000-6-2	Industriebereich	
		EN 61000-4-2	ESD 8kV bei Luftentladung (Schärfegrad 3), 4kV bei Kontaktentladung (Schärfegrad 2)
		EN 61000-4-3	HF-Einstrahlung (Gehäuse) 80MHz ... 1000MHz, 10V/m, 80% AM (1kHz) 1,4GHz ... 2,0GHz, 3V/m, 80% AM (1kHz) 2GHz ... 2,7GHz, 1V/m, 80% AM (1kHz)
		EN 61000-4-6	HF-Leitungsgeführt 150kHz ... 80MHz, 10V, 80% AM (1kHz)
		EN 61000-4-4	Burst, Schärfegrad 3
		EN 61000-4-5	Surge, Installationsklasse 3 *

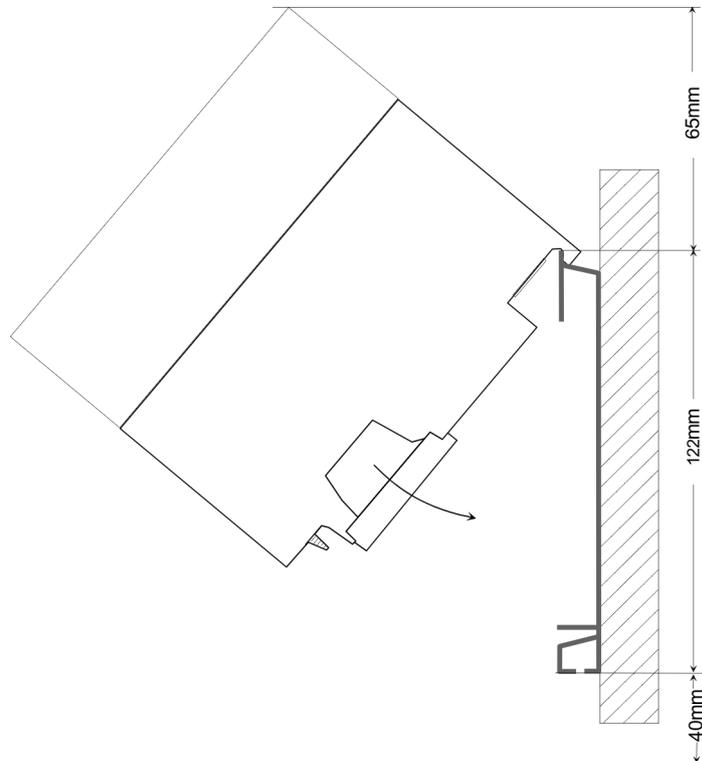
\*) Aufgrund der energiereichen Einzelimpulse ist bei Surge eine angemessene externe Beschaltung mit Blitzschutzelementen wie z.B. Blitzstromableitern und Überspannungsableitern erforderlich.

### 3 Montage und Aufbaurichtlinien

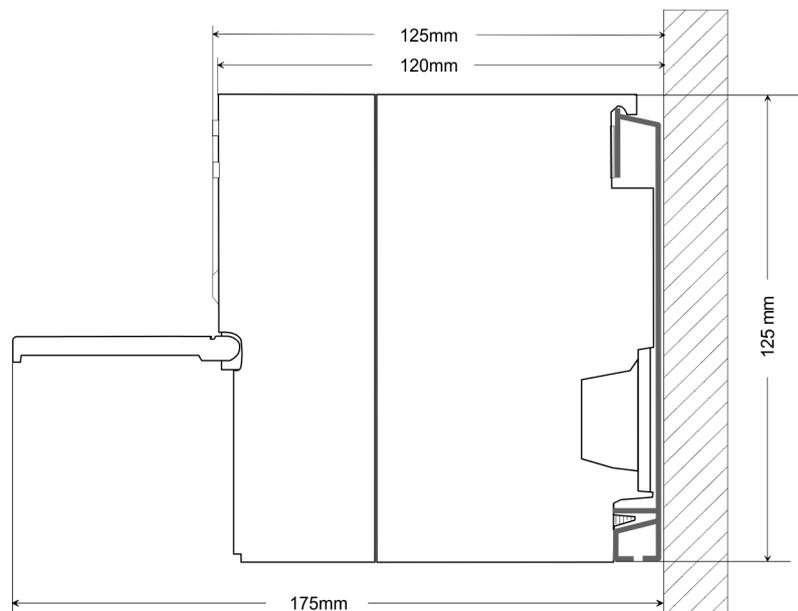
#### 3.1 Einbaumaße

**Maße Grundgehäuse** 2fach breit (BxHxT) in mm: 80 x 125 x 120

#### Montagemaße



#### Maße montiert



### 3.2 Montage Standard-Bus

#### Allgemein

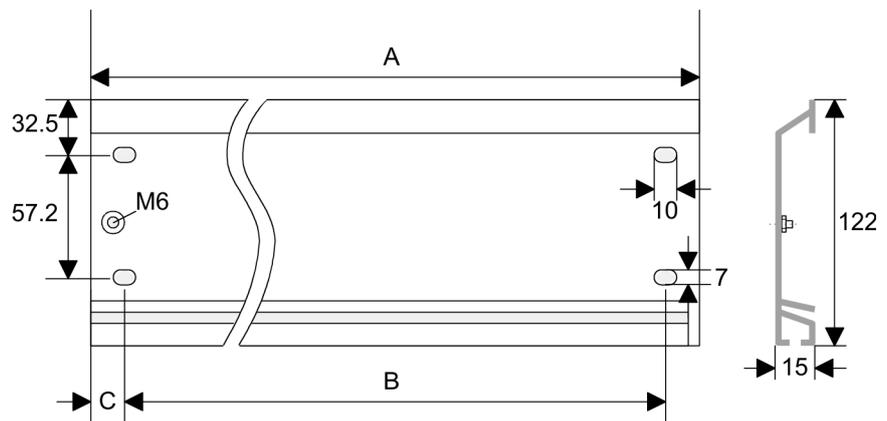
Die einzelnen Module werden direkt auf eine Profilschiene montiert und über den Rückwandbus-Verbinder verbunden. Vor der Montage ist der Rückwandbus-Verbinder von hinten an das Modul zu stecken. Die Rückwandbus-Verbinder sind im Lieferumfang der Peripherie-Module enthalten.

#### Profilschiene

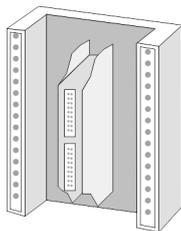
Bestellnummer	A	B	C
390-1AB60	160	140	10
390-1AE80	482	466	8,3
390-1AF30	530	500	15
390-1AJ30	830	800	15
390-9BC00*	2000	Bohrungen nur links	15

\*) Verpackungseinheit 10 Stück

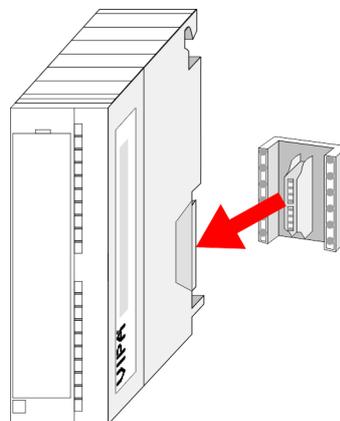
Maße in mm



#### Busverbinder

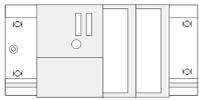


Für die Kommunikation der Module untereinander wird beim System 300S ein Rückwandbus-Verbinder eingesetzt. Die Rückwandbus-Verbinder sind im Lieferumfang der Peripherie-Module enthalten und werden vor der Montage von hinten an das Modul gesteckt.



## Montagemöglichkeiten

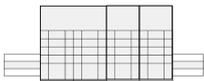
waagrechter Aufbau



senkrechter Aufbau



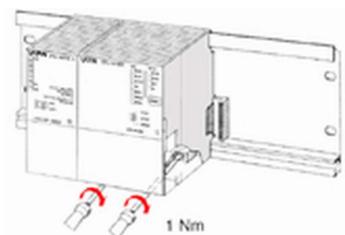
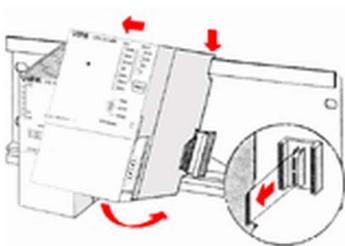
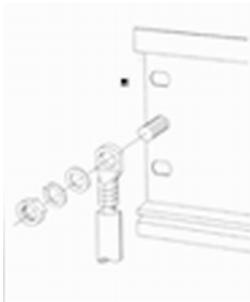
liegender Aufbau



Beachten Sie bitte die hierbei zulässigen Umgebungstemperaturen:

- waagrechter Aufbau: von 0 bis 60°C
- senkrechter Aufbau: von 0 bis 40°C
- liegender Aufbau: von 0 bis 40°C

## Vorgehensweise



1. ▶ Verschrauben Sie die Profilschiene mit dem Untergrund (Schraubengröße: M6) so, dass mindestens 65mm Raum oberhalb und 40mm unterhalb der Profilschiene bleibt.
2. ▶ Achten Sie bei geerdetem Untergrund auf eine niederohmige Verbindung zwischen Profilschiene und Untergrund.
3. ▶ Verbinden Sie die Profilschiene mit dem Schutzleiter. Für diesen Zweck befindet sich auf der Profilschiene ein Stehbolzen mit M6-Gewinde.
4. ▶ Der Mindestquerschnitt der Leitung zum Schutzleiter muss 10mm<sup>2</sup> betragen.
5. ▶ Hängen Sie die Spannungsversorgung ein und schieben Sie diese nach links bis an den Erdungsbolzen der Profilschiene.
6. ▶ Schrauben Sie die Spannungsversorgung fest.
7. ▶ Nehmen Sie einen Rückwandbus-Verbinder und stecken Sie ihn wie gezeigt von hinten an die CPU.
8. ▶ Hängen Sie die CPU rechts von der Spannungsversorgung ein und schieben Sie diese bis an die Spannungsversorgung.
9. ▶ Klappen Sie die CPU nach unten und schrauben Sie die CPU wie gezeigt fest.
10. ▶ Verfahren Sie auf die gleiche Weise mit Ihren Peripheriemodulen, indem Sie jeweils einen Rückwandbus-Verbinder stecken, Ihr Modul rechts neben dem Vorgänger-Modul einhängen, dieses nach unten klappen, in den Rückwandbus-Verbinder des Vorgängermoduls einrasten lassen und das Modul festschrauben.

## 3.3 Verdrahtung

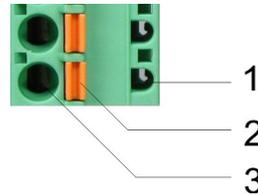


### VORSICHT!

- Die Spannungsversorgungen sind vor dem Beginn von Installations- und Instandhaltungsarbeiten unbedingt freizuschalten, d.h. vor Arbeiten an einer Spannungsversorgung oder an der Zuleitung, ist die Spannungszuführung stromlos zu schalten (Stecker ziehen, bei Festanschluss ist die zugehörige Sicherung abzuschalten)!
- Anschluss und Änderungen dürfen nur durch ausgebildetes Elektro-Fachpersonal ausgeführt werden.

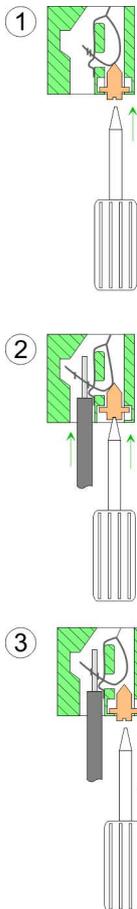
**Federklemmtechnik  
(grün)**

Zur Verdrahtung der Spannungsversorgung der CPU kommt eine grüne Anschlussklemmen mit Federzugklemmtechnik zum Einsatz. Die Anschlussklemme ist als Stecker ausgeführt, der im verdrahteten Zustand vorsichtig abgezogen werden kann. Hier können Sie Drähte mit einem Querschnitt von  $0,08\text{mm}^2$  bis  $2,5\text{mm}^2$  anschließen. Hierbei dürfen sowohl flexible Litzen ohne Aderendhülse, als auch starre Leiter verwendet werden.



- 1 Prüfabgriff für 2mm Messspitze
- 2 Verriegelung (orange) für Schraubendreher
- 3 Runde Öffnung für Drähte

Die nebenstehende Abfolge stellt die Schritte der Verdrahtung in der Draufsicht dar.



1. ► Zum Verdrahten drücken Sie mit einem geeigneten Schraubendreher, wie in der Abbildung gezeigt, die Verriegelung senkrecht nach innen und halten Sie den Schraubendreher in dieser Position.
2. ► Führen Sie durch die runde Öffnung Ihren abisolierten Draht ein. Sie können Drähte mit einem Querschnitt von  $0,08\text{mm}^2$  bis  $2,5\text{mm}^2$  anschließen.
3. ► Durch Entfernen des Schraubendrehers wird der Draht über einen Federkontakt sicher mit dem Steckverbinder verbunden.

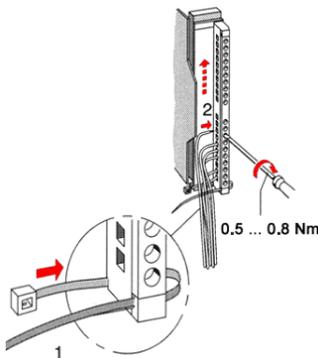
**Frontstecker der Ein-/  
Ausgabe-Module**

Nachfolgend ist die Verdrahtung der 2 Frontstecker-Varianten aufgezeigt.

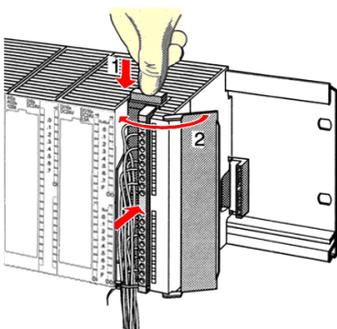
## 20-fach Schraubtechnik 392-1AJ00



1. ▶ Öffnen Sie die Frontklappe Ihres Ein-/Ausgabe-Moduls.
2. ▶ Bringen Sie den Frontstecker in Verdrahtungsstellung.  
Hierzu stecken Sie den Frontstecker auf das Modul, bis er einrastet. In dieser Stellung ragt der Frontstecker aus dem Modul heraus und hat noch keinen Kontakt.
3. ▶ Isolieren Sie Ihre Leitungen ab. Verwenden Sie ggf. Aderendhülsen.
4. ▶ Fädeln Sie den beiliegenden Kabelbinder in den Frontstecker ein.
5. ▶ Beginnen Sie mit der Verdrahtung von unten nach oben, wenn Sie die Leitungen nach unten aus dem Modul herausführen möchten, bzw. von oben nach unten, wenn die Leitungen nach oben herausgeführt werden sollen.
6. ▶ Schrauben Sie die Anschlusschrauben der nicht verdrahteten Schraubklemmen ebenfalls fest.

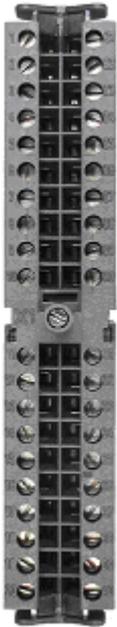


7. ▶ Ziehen Sie den Kabelbinder für den Leitungsstrang fest.

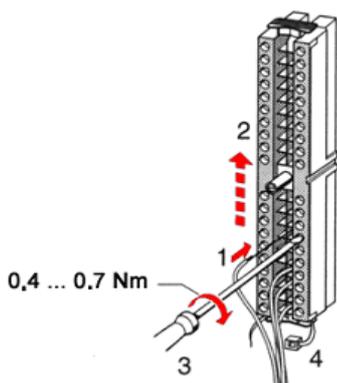


8. ▶ Drücken Sie die Entriegelungstaste am Frontstecker an der Moduloberseite und drücken Sie gleichzeitig den Frontstecker in das Modul, bis er einrastet.
9. ▶ Der Frontstecker ist nun elektrisch mit Ihrem Modul verbunden.
10. ▶ Schließen Sie die Frontklappe.
11. ▶ Füllen Sie den Beschriftungsstreifen zur Kennzeichnung der einzelnen Kanäle aus und schieben Sie den Streifen in die Frontklappe.

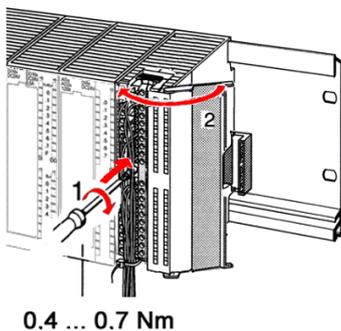
### 40-fach Schraubtechnik 392-1AM00



1. ▶ Öffnen Sie die Frontklappe Ihres Ein-/Ausgabe-Moduls.
2. ▶ Bringen Sie den Frontstecker in Verdrahtungsstellung.  
Hierzu stecken Sie den Frontstecker auf das Modul, bis er einrastet. In dieser Stellung ragt der Frontstecker aus dem Modul heraus und hat noch keinen Kontakt.
3. ▶ Isolieren Sie Ihre Leitungen ab. Verwenden Sie ggf. Aderendhülsen.
4. ▶ Beginnen Sie mit der Verdrahtung von unten nach oben, wenn Sie die Leitungen nach unten aus dem Modul herausführen möchten, bzw. von oben nach unten, wenn die Leitungen nach oben herausgeführt werden sollen.
5. ▶ Schrauben Sie die Anschlusschrauben der nicht verdrahteten Schraubklemmen ebenfalls fest.



6. ▶ Legen Sie den beigegefügteten Kabelbinder um den Leitungsstrang und den Frontstecker herum.
7. ▶ Ziehen Sie den Kabelbinder für den Leitungsstrang fest.



8. ▶ Schrauben Sie die Befestigungsschraube für den Frontstecker fest.
9. ▶ Der Frontstecker ist nun elektrisch mit Ihrem Modul verbunden.
10. ▶ Schließen Sie die Frontklappe.
11. ▶ Füllen Sie den Beschriftungsstreifen zur Kennzeichnung der einzelnen Kanäle aus und schieben Sie den Streifen in die Frontklappe.

## 3.4 Aufbaurichtlinien

### Allgemeines

Die Aufbaurichtlinien enthalten Informationen über den störsicheren Aufbau eines SPS-Systems. Es werden die Wege beschrieben, wie Störungen in Ihre Steuerung gelangen können, wie die elektromagnetische Verträglichkeit (EMV) sicher gestellt werden kann und wie bei der Schirmung vorzugehen ist.

- Was bedeutet EMV?**
- Unter Elektromagnetischer Verträglichkeit (EMV) versteht man die Fähigkeit eines elektrischen Gerätes, in einer vorgegebenen elektromagnetischen Umgebung fehlerfrei zu funktionieren, ohne vom Umfeld beeinflusst zu werden bzw. das Umfeld in unzulässiger Weise zu beeinflussen.
- Die Komponenten von VIPA sind für den Einsatz in Industrieumgebungen entwickelt und erfüllen hohe Anforderungen an die EMV. Trotzdem sollten Sie vor der Installation der Komponenten eine EMV-Planung durchführen und mögliche Störquellen in die Betrachtung einbeziehen.
- Mögliche Störeinträge**
- Elektromagnetische Störungen können sich auf unterschiedlichen Pfaden in Ihre Steuerung einkoppeln:
- Elektromagnetische Felder (HF-Einkopplung)
  - Magnetische Felder mit energietechnischer Frequenz
  - Bus-System
  - Stromversorgung
  - Schutzleiter
- Je nach Ausbreitungsmedium (leitungsgebunden oder -ungebunden) und Entfernung zur Störquelle gelangen Störungen über unterschiedliche Kopplungsmechanismen in Ihre Steuerung.
- Man unterscheidet:
- galvanische Kopplung
  - kapazitive Kopplung
  - induktive Kopplung
  - Strahlungskopplung
- Grundregeln zur Sicherstellung der EMV**
- Häufig genügt zur Sicherstellung der EMV das Einhalten einiger elementarer Regeln. Beachten Sie beim Aufbau der Steuerung deshalb die folgenden Grundregeln.
- Achten Sie bei der Montage Ihrer Komponenten auf eine gut ausgeführte flächenhafte Massung der inaktiven Metallteile.
    - Stellen Sie eine zentrale Verbindung zwischen der Masse und dem Erde/Schutzleitersystem her.
    - Verbinden Sie alle inaktiven Metallteile großflächig und impedanzarm.
    - Verwenden Sie nach Möglichkeit keine Aluminiumteile. Aluminium oxidiert leicht und ist für die Massung deshalb weniger gut geeignet.
  - Achten Sie bei der Verdrahtung auf eine ordnungsgemäße Leitungsführung.
    - Teilen Sie die Verkabelung in Leitungsgruppen ein. (Starkstrom, Stromversorgungs-, Signal- und Datenleitungen).
    - Verlegen Sie Starkstromleitungen und Signal- bzw. Datenleitungen immer in getrennten Kanälen oder Bündeln.
    - Führen Sie Signal- und Datenleitungen möglichst eng an Masseflächen (z.B. Tragholme, Metallschienen, Schrankbleche).

- Achten sie auf die einwandfreie Befestigung der Leitungsschirme.
  - Datenleitungen sind geschirmt zu verlegen.
  - Analogleitungen sind geschirmt zu verlegen. Bei der Übertragung von Signalen mit kleinen Amplituden kann das einseitige Auflegen des Schirms vorteilhaft sein.
  - Legen Sie die Leitungsschirme direkt nach dem Schrankeintritt großflächig auf eine Schirm-/Schutzleiterschiene auf, und befestigen Sie die Schirme mit Kabelschellen.
  - Achten Sie darauf, dass die Schirm-/Schutzleiterschiene impedanzarm mit dem Schrank verbunden ist.
  - Verwenden Sie für geschirmte Datenleitungen metallische oder metallisierte Steckergehäuse.
- Setzen Sie in besonderen Anwendungsfällen spezielle EMV-Maßnahmen ein.
  - Erwägen Sie bei Induktivitäten den Einsatz von Löschgliedern.
  - Beachten Sie, dass bei Einsatz von Leuchtstofflampen sich diese negativ auf Signalleitungen auswirken können.
- Schaffen Sie ein einheitliches Bezugspotential und erden Sie nach Möglichkeit alle elektrischen Betriebsmittel.
  - Achten Sie auf den gezielten Einsatz der Erdungsmaßnahmen. Das Erden der Steuerung dient als Schutz- und Funktionsmaßnahme.
  - Verbinden Sie Anlagenteile und Schränke mit Ihrer SPS sternförmig mit dem Erde/Schutzleitersystem. Sie vermeiden so die Bildung von Erdschleifen.
  - Verlegen Sie bei Potentialdifferenzen zwischen Anlagenteilen und Schränken ausreichend dimensionierte Potentialausgleichsleitungen.

## Schirmung von Leitungen

Elektrische, magnetische oder elektromagnetische Störfelder werden durch eine Schirmung geschwächt; man spricht hier von einer Dämpfung. Über die mit dem Gehäuse leitend verbundene Schirmschiene werden Störströme auf Kabelschirme zur Erde hin abgeleitet. Hierbei ist darauf zu achten, dass die Verbindung zum Schutzleiter impedanzarm ist, da sonst die Störströme selbst zur Störquelle werden.

Bei der Schirmung von Leitungen ist folgendes zu beachten:

- Verwenden Sie möglichst nur Leitungen mit Schirmgeflecht.
- Die Deckungsdichte des Schirmes sollte mehr als 80% betragen.
- In der Regel sollten Sie die Schirme von Leitungen immer beidseitig auflegen. Nur durch den beidseitigen Anschluss der Schirme erreichen Sie eine gute Störunterdrückung im höheren Frequenzbereich. Nur im Ausnahmefall kann der Schirm auch einseitig aufgelegt werden. Dann erreichen Sie jedoch nur eine Dämpfung der niedrigen Frequenzen. Eine einseitige Schirmanbindung kann günstiger sein, wenn:
  - die Verlegung einer Potentialausgleichsleitung nicht durchgeführt werden kann.
  - Analogsignale (einige mV bzw.  $\mu\text{A}$ ) übertragen werden.
  - Folienschirme (statische Schirme) verwendet werden.
- Benutzen Sie bei Datenleitungen für serielle Kopplungen immer metallische oder metallisierte Stecker. Befestigen Sie den Schirm der Datenleitung am Steckergehäuse. Schirm nicht auf den PIN 1 der Steckerleiste auflegen!
- Bei stationärem Betrieb ist es empfehlenswert, das geschirmte Kabel unterbrechungsfrei abzuisolieren und auf die Schirm-/Schutzleiterschiene aufzulegen.

- Benutzen Sie zur Befestigung der Schirmgeflechte Kabelschellen aus Metall. Die Schellen müssen den Schirm großflächig umschließen und guten Kontakt ausüben.
- Legen Sie den Schirm direkt nach Eintritt der Leitung in den Schrank auf eine Schirmschiene auf. Führen Sie den Schirm bis zu Ihrer SPS weiter, legen Sie ihn dort jedoch nicht erneut auf!

**VORSICHT!****Bitte bei der Montage beachten!**

Bei Potenzialdifferenzen zwischen den Erdungspunkten kann über den beidseitig angeschlossenen Schirm ein Ausgleichsstrom fließen.

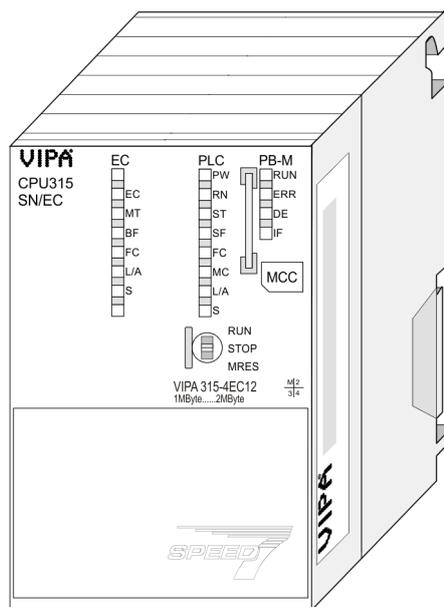
Abhilfe: Potenzialausgleichsleitung.

## 4 Hardwarebeschreibung

### 4.1 Leistungsmerkmale

#### CPU 315-4EC12

- SPEED7-Technologie integriert
- 1MByte Arbeitsspeicher integriert (512kByte Code, 512kByte Daten)
- Speicher erweiterbar bis max. 2MByte (1MByte Code, 1MByte Daten)
- 2MByte Ladespeicher
- EtherCAT-Master-Funktionalität
- PROFIBUS-DP-Master integriert (DP-V0, DP-V1)
- MPI-Schnittstelle
- MCC-Slot für externe Speichermedien und Speichererweiterung
- Status-LEDs für Betriebszustand und Diagnose
- Echtzeituhr akkugepuffert
- Ethernet-PG/OP-Schnittstelle integriert
- RS485-Schnittstelle konfigurierbar für PROFIBUS-DP-Master- bzw. PtP-Kommunikation
- E/A-Adressbereich digital/analog 8191Byte
- 512 Zeiten
- 512 Zähler
- 8192 Merker-Byte



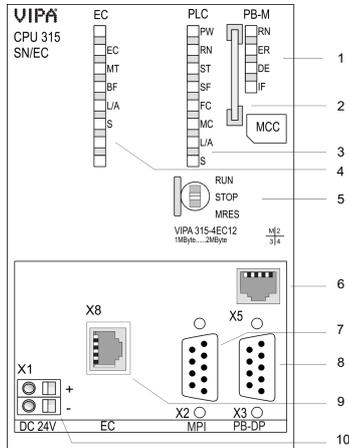
#### Bestelldaten

Typ	Bestellnummer	Beschreibung
315SN/EC	315-4EC12	MPI-Interface, Karten-Slot, Echtzeituhr, Ethernet-Interface für PG/OP, PROFIBUS-DP-Master, EtherCAT-Master-Funktionalität

## 4.2 Aufbau

### 4.2.1 Allgemein

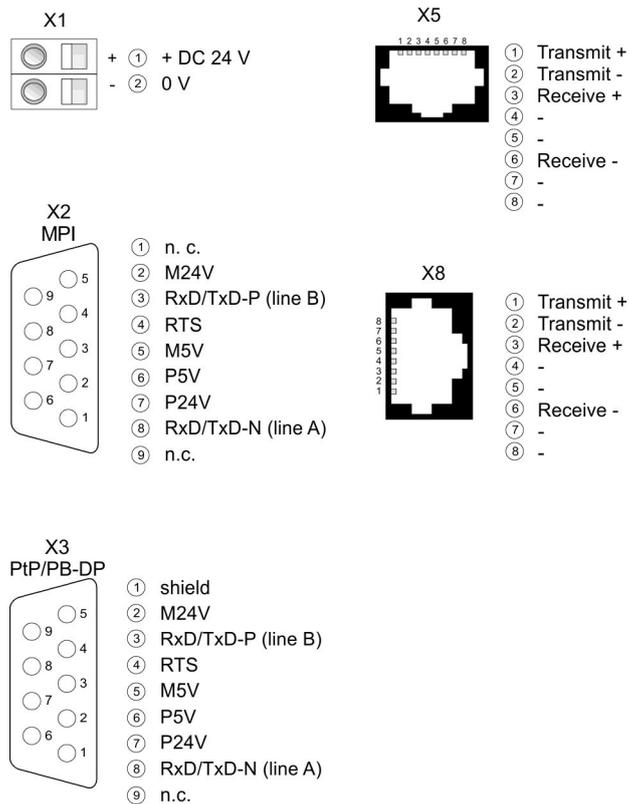
#### CPU 315-4EC12



- 1 LEDs des integrierten PROFIBUS-DP-Masters
- 2 Steckplatz für Speichermedien (verriegelbar)
- 3 LEDs des CPU-Teils
- 4 LEDs des integrierten EtherCAT-Masters
- 5 Betriebsarten-Schalter CPU
- 6 Twisted Pair Schnittstelle für Ethernet-PG/OP-Kanal
- 7 MPI-Schnittstelle
- 8 PROFIBUS-DP/PtP-Schnittstelle
- 9 Twisted Pair Schnittstelle für EtherCAT-Kommunikation
- 10 Anschluss für DC 24V Spannungsversorgung

**Komponenten 6 - 10 befinden sich unter der Frontklappe!**

### 4.2.2 Schnittstellen



#### X1: Spannungsversorgung

Die CPU besitzt ein eingebautes Netzteil:

- Das Netzteil ist mit DC 24V zu versorgen. Hierzu dient der DC 24V Anschluss, der sich unter der Frontklappe befindet.
- Mit der Versorgungsspannung werden neben der CPU-Elektronik auch die angeschlossenen Module über den Rückwandbus versorgt.
- Das Netzteil ist gegen Verpolung und Überstrom geschützt.
- Die interne Elektronik ist galvanisch an die Versorgungsspannung gebunden.

**X2: MPI-Schnittstelle***9polige SubD-Buchse:*

- Die MPI-Schnittstelle dient zur Verbindung zwischen Programmiergerät und CPU.
- Hierüber erfolgt beispielsweise die Projektierung und Programmierung.
- MPI dient zur Kommunikation zwischen mehreren CPUs oder zwischen HMIs und CPU.
- Standardmäßig ist die MPI-Adresse 2 eingestellt.

**X5: Ethernet-PG/OP-Kanal***8polige RJ45-Buchse:*

- Die RJ45-Buchse dient als Schnittstelle zum Ethernet-PG/OP-Kanal.
- Mittels dieser Schnittstelle können Sie Ihre CPU programmieren bzw. fernwarten und auf die integrierte Web-Seite zugreifen.
- Projektierbare Verbindungen sind nicht möglich.
- Damit Sie online auf den Ethernet-PG/OP-Kanal zugreifen können, müssen Sie diesem IP-Adress-Parameter zuweisen.

**X3: PROFIBUS/PtP-Schnittstelle mit projektierbarer Funktionalität***9polige SubD-Buchse:*

Die CPU besitzt eine PROFIBUS/PtP-Schnittstelle mit fixer Pinbelegung. Nach dem Umräumen ist diese Schnittstelle deaktiviert. Durch entsprechende Projektierung können Sie folgende Funktionalitäten für diese Schnittstelle aktivieren:

- PROFIBUS-DP-Master-Betrieb
  - Projektierung erfolgt über das PROFIBUS-Submodul X1 (MPI/DP) der CPU mit *"Betriebsart"* Master in der Hardware-Konfiguration.
- PROFIBUS-DP-Slave-Betrieb
  - Projektierung erfolgt über das PROFIBUS-Submodul X1 (MPI/DP) der CPU mit *"Betriebsart"* Slave in der Hardware-Konfiguration.
- PtP-Funktionalität
  - Mit der Funktionalität PtP ermöglicht die RS485-Schnittstelle eine serielle Punkt-zu-Punkt-Prozessankopplung zu verschiedenen Ziel- oder Quell-Systemen.
  - Unterstützt werden die Protokolle ASCII, STX/ETX, 3964R, USS und Modbus-Master (ASCII, RTU).
  - Die Aktivierung der PtP-Funktionalität erfolgt durch Einbindung der SPEEDBUS.GSD von VIPA im Hardwarekatalog. Nach der Installation können Sie die CPU in einem PROFIBUS-Master-System projektieren und hier auch die Schnittstelle auf PtP-Kommunikation umschalten.

**Schnittstelle für EtherCAT-Kommunikation X8***8polige RJ45-Buchse:*

- Verbinden Sie diese Schnittstellen mit der RJ45-Buchse "IN" Ihrer Slave-Station.
- EtherCAT verwendet als Übertragungsmedium Ethernet. Es kommen Standard-CAT5-Kabel zum Einsatz. Hierbei sind Leitungslängen von bis zu 100m zwischen 2 Teilnehmern möglich.
- In einem EtherCAT-Netzwerk dürfen nur EtherCAT-Komponenten verwendet werden. Für die Realisierung von Topologien abweichend von der Linienstruktur sind entsprechende EtherCAT-Komponenten erforderlich, welche dies unterstützen. Der Einsatz von Hubs ist nicht möglich.

- Ein EtherCAT-Netz besteht immer aus einem Master und einer beliebigen Anzahl an EtherCAT-Slaves (Koppler).
- Jeder EtherCAT-Slave besitzt eine RJ45-Buchse "IN" und "OUT". Das ankommende EtherCAT-Kabel aus Richtung des Masters ist in die mit "IN" bezeichnete Buchse zu stecken. Die mit "OUT" bezeichnete Buchse ist mit dem nachfolgenden Teilnehmer zu verbinden. Beim jeweiligen letzten Teilnehmer bleibt die "OUT"-Buchse frei.

### 4.2.3 Speichermanagement

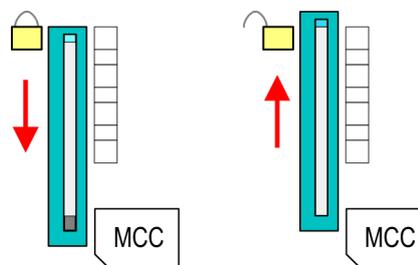
#### Speicher

Die CPU hat einen Speicher integriert. Angaben über die Speicherkapazität finden Sie auf der Frontseite Ihrer CPU. Der Speicher gliedert sich in folgende Teile:

- Ladespeicher 2MByte
- Codespeicher (50% des Arbeitsspeichers)
- Datenspeicher (50% des Arbeitsspeichers)
- Arbeitsspeicher 1MByte
  - Sie haben die Möglichkeit den Arbeitsspeicher mittels einer MCC Speichererweiterungskarte bis zur maximal aufgedruckten Kapazität 2MByte zu erweitern.

### 4.2.4 Steckplatz für Speichermedien

- Über diesen Steckplatz können Sie eine MMC (**M**ultimedia **C**ard) als externes Speichermedium für Programme und Firmware stecken.
- Die VIPA-Speicherkarten sind mit dem PC-Format FAT vorformatiert und können mit einem Kartenlesegerät beschrieben werden.
- Nach PowerON bzw. nach Urlöschen überprüft die CPU, ob eine Speicherkarte gesteckt ist und sich hier für die CPU gültige Daten befinden.
- Schieben Sie ihr Speichermedium in den Steckplatz, bis dieses geführt durch eine Federmechanik einrastet. Dies gewährleistet eine sichere Kontaktierung.
- Mit der Schiebemechanik können Sie durch Schieben nach unten ein gestecktes Speichermedium gegen Herausfallen sichern.
- Zum Entnehmen schieben Sie die Schiebemechanik wieder nach oben und drücken Sie das Speichermedium gegen den Federdruck nach innen, bis dieses mit einem Klick entriegelt wird.



#### VORSICHT!

Sofern das Speichermedium schon durch die Federmechanik entriegelt wurde, kann dieses bei Betätigung der Schiebemechanik herauspringen!

### 4.2.5 Batteriepufferung für Uhr und RAM

Jede CPU 31xS besitzt einen internen Akku, der zur Sicherung des RAMs bei Stromausfall dient. Zusätzlich wird die interne Uhr über den Akku gepuffert. Der Akku wird direkt über die eingebaute Spannungsversorgung über eine Ladeelektronik geladen und gewährleistet eine Pufferung für max. 30 Tage.



**VORSICHT!**

Bitte schließen Sie die CPU mindestens für 24 Stunden an die Spannungsversorgung an, damit der interne Akku entsprechend geladen wird.

Bei leerem Akku läuft die CPU nach einem Spannungsreset mit einem BAT-Fehler an und führt ein automatisches Urlöschen der CPU durch. Der BAT-Fehler hat keinen Einfluss auf den Ladevorgang.

Den BAT-Fehler können Sie wieder löschen, wenn einmalig beim Power-Cycle zwischen dem Aus- und Einschalten der Versorgungsspannung mindestens 30sec. liegen und der Akku der CPU voll geladen ist. Ansonsten bleibt bei einem kurzen Power-Cycle der BAT-Fehler bestehen und die CPU wird urgelöscht.

### 4.2.6 Betriebsartenschalter



- Mit dem Betriebsartenschalter können Sie bei der CPU zwischen den Betriebsarten STOP und RUN wählen.
- Beim Übergang vom Betriebszustand STOP nach RUN durchläuft die CPU den Betriebszustand ANLAUF.
- Mit der Tasterstellung MRES (Memory Reset) fordern Sie das Urlöschen an mit anschließendem Laden von Speicherkarte, sofern dort ein Projekt hinterlegt ist.

### 4.2.7 LEDs

#### LEDs CPU

Sobald die CPU intern mit 5V versorgt wird, leuchtet die grüne PW-LED (Power).

RN (RUN)	ST (STOP)	SF (SFAIL)	FC (FRCE)	MC (MMC)	Bedeutung
grün 	gelb 	rot 	gelb 	gelb 	
Bootvorgang nach NetzEIN					
•	BB*	•	•	•	* Blinken mit 10Hz: Firmware wird geladen.
•	•	•	•	•	Initialisierung: Phase 1
•	•	•	•	○	Initialisierung: Phase 2
•	•	•	○	○	Initialisierung: Phase 3

RN (RUN)	ST (STOP)	SF (SFAIL)	FC (FRCE)	MC (MMC)	Bedeutung
○	●	●	○	○	Initialisierung: Phase 4
<b>Betrieb</b>					
○	●	X	X	X	CPU befindet sich im Zustand STOP.
BB	○	X	X	X	CPU befindet sich im Zustand Anlauf. Solange der OB100 durchlaufen wird, blinkt die RUN-LED, mindestens für 3s.
●	○	○	X	X	CPU befindet sich ohne Fehler im Zustand RUN.
X	X	●	X	X	Es liegt ein Systemfehler vor. Nähere Informationen hierzu finden Sie im Diagnosepuffer der CPU.
X	X	X	●	X	Variablen sind geforced (fixiert).
X	X	X	X	●	Zugriff auf Speicherkarte.
X	BB*	○	○	○	* Blinken mit 10Hz: Konfiguration wird geladen.
<b>Urlöschen</b>					
○	BB	X	X	X	Urlöschen wird angefordert.
○	BB*	X	X	X	* Blinken mit 5Hz: Urlöschen wird durchgeführt.
<b>Rücksetzen auf Werkseinstellung</b>					
●	●	○	○	○	Rücksetzen auf Werkseinstellung wird durchgeführt.
○	●	●	●	●	Rücksetzen auf Werkseinstellung war erfolgreich.
<b>Firmwareupdate</b>					
○	●	BB	BB	●	Das abwechselnde Blinken zeigt an, dass neue Firmware auf der Speicherkarte vorhanden ist.
○	○	BB	BB	●	Das abwechselnde Blinken zeigt an, dass ein Firmwareupdate durchgeführt wird.
○	●	●	●	●	Firmwareupdate wurde fehlerfrei durchgeführt.
○	BB*	BB*	BB*	BB*	* Blinken mit 10Hz: Fehler bei Firmwareupdate.

an: ● | aus: ○ | blinkend (2Hz): BB | nicht relevant: X

#### LEDs Ethernet-PG/OP-Kanal L/A, S

Die grüne L/A-LED (Link/Activity) zeigt an, dass der Ethernet-PG/OP-Kanal physikalisch mit Ethernet verbunden ist. Unregelmäßiges Blinken der L/A-LED zeigt Kommunikation des Ethernet-PG/OP-Kanals über Ethernet an.

Leuchtet die grüne S-LED (Speed), so hat der Ethernet-PG/OP-Kanal eine Übertragungsrate von 100MBit/s ansonsten 10MBit/s.

**LEDs PROFIBUS/PtP-Schnittstelle X3**

Abhängig von der Betriebsart geben die LEDs nach folgendem Schema Auskunft über den Betriebszustand des PROFIBUS-Teils:

**Master-Betrieb**

RN (RUN)	ER (ERR)	DE	IF	Bedeutung
grün 	rot 	grün 	rot 	
○	○	○	○	Master hat keine Projektierung, d.h. die Schnittstelle ist deaktiviert bzw. PtP ist aktiv.
●	○	○	○	Master hat Busparameter und befindet sich im RUN ohne Slaves.
●	○	BB	○	Master befindet sich im "clear"-Zustand (sicherer Zustand). Die Eingänge der Slaves können gelesen werden. Die Ausgänge sind gesperrt.
●	○	●	○	Master befindet sich im "operate"-Zustand, d.h. er tauscht Daten mit den Slaves aus. Ausgänge können angesprochen werden.
●	●	●	○	CPU ist im Zustand RUN, es fehlt mindestens 1 Slave.
●	●	BB	○	CPU ist im Zustand STOP, es fehlt mindestens 1 Slave.
○	○	○	●	Initialisierungsfehler bei fehlerhafter Parametrierung.
○	●	○	●	Wartezustand auf Start-Kommando von der CPU.
an: ●   aus: ○   blinkend (2Hz): BB				

**Slave-Betrieb**

RN (RUN)	ER (ERR)	DE	IF	Bedeutung
grün 	rot 	grün 	rot 	
○	○	○	○	Slave hat keine Projektierung bzw. PtP ist aktiv.
BB	○	○	○	Slave ist ohne Master.
BB*	○	BB*	○	* Abwechselndes Blinken bei Projektierungsfehler (configuration fault).
●	○	●	○	Slave tauscht Daten mit dem Master aus.
an: ●   aus: ○   blinkend (2Hz): BB				

**LEDs EtherCAT-Schnittstelle X8**

EC	MT	BF	Bedeutung
grün 	gelb 	rot 	
○	○	○	Master ist im Zustand INIT
BB	○	○	Master ist im Zustand Pre-Op
P	○	○	Master ist im Zustand Safe-Op
●	○	○	Master ist im Zustand OP
X	○	X	Es liegt kein Maintenance-Ereignis an
X	●	X	Ein Maintenance-Ereignis liegt an. Näheres hierzu finden Sie in der Diagnose
X	X	○	Es liegt kein Fehler am EtherCAT-Bus vor
X	X	●	<ul style="list-style-type: none"> <li>■ EtherCAT-Busfehler, keine Verbindung zu Subnetz</li> <li>■ falsche Übertragungsgeschwindigkeit</li> <li>■ Voll duplexübertragung ist nicht aktiviert</li> </ul>
X	X	B	<ul style="list-style-type: none"> <li>■ Ausfall eines angeschlossenen IO-Device</li> <li>■ Mindestens ein IO-Device ist nicht ansprechbar (Topologie-Fehler)</li> <li>■ Fehlerhafte Projektierung</li> </ul>
○	B4	B4	Fehlerhafte Projektierung: Im Diagnosepuffer wurde 0xEA64 eingetragen. Zusätzlich leuchtet die SF-LED der CPU.
○	BB*	BB*	* Das abwechselnde Blinken mit 4Hz zeigt an, dass ein Firmwareupdate des EtherCAT-Masters durchgeführt wird.
●	●	●	Firmwareupdate des EtherCAT-Masters wurde fehlerfrei durchgeführt.

an: ● | aus: ○ | blinkend (1Hz): B | blinkend (2Hz): BB | B4: blinkend (4s an, 1s aus) | pulsierend: P | flackernd: F | nicht relevant: X

**LEDs L/A, S**

Die grüne L/A-LED (Link/Activity) zeigt an, dass der EtherCAT-Master physikalisch mit Ethernet verbunden ist. Unregelmäßiges Blinken der L/A-LED zeigt Kommunikation des EtherCAT-Masters über Ethernet an.

Leuchtet die grüne S-LED (Speed), so hat der EtherCAT-Master eine Übertragungsrate von 100MBit/s ansonsten mit 10MBit/s.

**4.3 Technische Daten**

<b>Artikelnr.</b>	<b>315-4EC12</b>
Bezeichnung	CPU 315SN/EC
SPEED-Bus	-
<b>Technische Daten Stromversorgung</b>	
Versorgungsspannung (Nennwert)	DC 24 V
Versorgungsspannung (zulässiger Bereich)	DC 20,4...28,8 V
Verpolschutz	✓

## Technische Daten

Artikelnr.	315-4EC12
Stromaufnahme (im Leerlauf)	270 mA
Stromaufnahme (Nennwert)	1,1 A
Einschaltstrom	6 A
$I^2t$	0,28 A <sup>2</sup> s
max. Stromabgabe am Rückwandbus	2,5 A
Verlustleistung	8,5 W
<b>Lade- und Arbeitsspeicher</b>	
Ladespeicher integriert	2 MB
Ladespeicher maximal	2 MB
Arbeitsspeicher integriert	1 MB
Arbeitsspeicher maximal	2 MB
Speicher geteilt 50% Code / 50% Daten	✓
Memory Card Slot	MMC-Card mit max. 1 GB
<b>Ausbau</b>	
Baugruppenträger max.	4
Baugruppen je Baugruppenträger	8 bei mehrzeiligem, 32 bei einzeiligem Aufbau
Anzahl DP-Master integriert	1
Anzahl DP-Master über CP	4
Betreibbare Funktionsbaugruppen	8
Betreibbare Kommunikationsbaugruppen PtP	8
Betreibbare Kommunikationsbaugruppen LAN	8
<b>Befehlsbearbeitungszeiten</b>	
Bitoperation, min.	0,01 µs
Wortoperation, min.	0,01 µs
Festpunktarithmetik, min.	0,01 µs
Gleitpunktarithmetik, min.	0,06 µs
<b>Zeiten/Zähler und deren Remanenz</b>	
Anzahl S7-Zähler	512
S7-Zähler Remanenz	einstellbar von 0 bis 512
S7-Zähler Remanenz voreingestellt	Z0 .. Z7
Anzahl S7-Zeiten	512
S7-Zeiten Remanenz	einstellbar von 0 bis 512
S7-Zeiten Remanenz voreingestellt	keine Remanenz
<b>Datenbereiche und Remanenz</b>	
Anzahl Merker	8192 Byte
Merker Remanenz einstellbar	einstellbar von 0 bis 8192

Artikelnr.	315-4EC12
Merker Remanenz voreingestellt	MB0 .. MB15
Anzahl Datenbausteine	4095
max. Datenbausteingröße	64 KB
Nummernband DBs	1 ... 4095
max. Lokaldatengröße je Ablaufebene	3072 Byte
max. Lokaldatengröße je Baustein	3072 Byte
<b>Bausteine</b>	
Anzahl OBs	24
maximale OB-Größe	64 KB
Gesamtanzahl DBs, FBs, FCs	-
Anzahl FBs	2048
maximale FB-Größe	64 KB
Nummernband FBs	0 ... 2047
Anzahl FCs	2048
maximale FC-Größe	64 KB
Nummernband FCs	0 ... 2047
maximale Schachtelungstiefe je Prioklasse	8
maximale Schachtelungstiefe zusätzlich innerhalb Fehler OB	4
<b>Uhrzeit</b>	
Uhr gepuffert	✓
Uhr Pufferungsdauer (min.)	6 w
Art der Pufferung	Vanadium Rechargeable Lithium Batterie
Ladezeit für 50% Pufferungsdauer	20 h
Ladezeit für 100% Pufferungsdauer	48 h
Genauigkeit (max. Abweichung je Tag)	10 s
Anzahl Betriebsstundenzähler	8
Uhrzeit Synchronisation	✓
Synchronisation über MPI	Master/Slave
Synchronisation über Ethernet (NTP)	Slave
<b>Adressbereiche (Ein-/Ausgänge)</b>	
Peripherieadressbereich Eingänge	2048 Byte
Peripherieadressbereich Ausgänge	2048 Byte
Prozessabbild einstellbar	✓
Prozessabbild Eingänge voreingestellt	128 Byte
Prozessabbild Ausgänge voreingestellt	128 Byte

Artikelnr.	315-4EC12
Prozessabbild Eingänge maximal	2048 Byte
Prozessabbild Ausgänge maximal	2048 Byte
Digitale Eingänge	16384
Digitale Ausgänge	16384
Digitale Eingänge zentral	1024
Digitale Ausgänge zentral	1024
Integrierte digitale Eingänge	-
Integrierte digitale Ausgänge	-
Analoge Eingänge	1024
Analoge Ausgänge	1024
Analoge Eingänge zentral	256
Analoge Ausgänge zentral	256
Integrierte analoge Eingänge	-
Integrierte analoge Ausgänge	-
<b>Kommunikationsfunktionen</b>	
PG/OP Kommunikation	✓
Globale Datenkommunikation	✓
Anzahl GD-Kreise max.	8
Größe GD-Pakete, max.	22 Byte
S7-Basis-Kommunikation	✓
S7-Basis-Kommunikation Nutzdaten je Auftrag	76 Byte
S7-Kommunikation	✓
S7-Kommunikation als Server	✓
S7-Kommunikation als Client	-
S7-Kommunikation Nutzdaten je Auftrag	160 Byte
Anzahl Verbindungen gesamt	32
<b>Funktionalität Sub-D Schnittstellen</b>	
Bezeichnung	X2
Physik	RS485
Anschluss	9polige SubD Buchse
Potenzialgetrennt	✓
MPI	✓
MP <sup>2</sup> I (MPI/RS232)	-
DP-Master	-
DP-Slave	-
Punkt-zu-Punkt-Kopplung	-

Artikelnr.	315-4EC12
Bezeichnung	X3
Physik	RS485
Anschluss	9polige SubD Buchse
Potenzialgetrennt	✓
MPI	-
MP <sup>2</sup> I (MPI/RS232)	-
DP-Master	ja
DP-Slave	ja
Punkt-zu-Punkt-Kopplung	✓
<b>Funktionalität MPI</b>	
Anzahl Verbindungen, max.	32
PG/OP Kommunikation	✓
Routing	✓
Globale Datenkommunikation	✓
S7-Basis-Kommunikation	✓
S7-Kommunikation	✓
S7-Kommunikation als Server	✓
S7-Kommunikation als Client	-
Übertragungsgeschwindigkeit, min.	19,2 kbit/s
Übertragungsgeschwindigkeit, max.	12 Mbit/s
<b>Funktionalität PROFIBUS Master</b>	
PG/OP Kommunikation	✓
Routing	✓
S7-Basis-Kommunikation	✓
S7-Kommunikation	✓
S7-Kommunikation als Server	✓
S7-Kommunikation als Client	-
Aktivieren/Deaktivieren von DP-Slaves	✓
Direkter Datenaustausch (Querverkehr)	-
DPV1	✓
Übertragungsgeschwindigkeit, min.	9,6 kbit/s
Übertragungsgeschwindigkeit, max.	12 Mbit/s
Anzahl DP-Slaves, max.	124
Adressbereich Eingänge, max.	8 KB
Adressbereich Ausgänge, max.	8 KB

## Technische Daten

Artikelnr.	315-4EC12
Nutzdaten Eingänge je Slave, max.	244 Byte
Nutzdaten Ausgänge je Slave, max.	244 Byte
<b>Funktionalität PROFIBUS Slave</b>	
PG/OP Kommunikation	✓
Routing	✓
S7-Kommunikation	✓
S7-Kommunikation als Server	✓
S7-Kommunikation als Client	-
Direkter Datenaustausch (Querverkehr)	-
DPV1	✓
Übertragungsgeschwindigkeit, min.	9,6 kbit/s
Übertragungsgeschwindigkeit, max.	12 Mbit/s
Automatische Baudratesuche	-
Übergabespeicher Eingänge, max.	244 Byte
Übergabespeicher Ausgänge, max.	244 Byte
Adressbereiche, max.	32
Nutzdaten je Adressbereich, max.	32 Byte
<b>Point-to-Point Kommunikation</b>	
PtP-Kommunikation	✓
Schnittstelle potentialgetrennt	✓
Schnittstelle RS232	-
Schnittstelle RS422	-
Schnittstelle RS485	✓
Anschluss	9polige SubD Buchse
Übertragungsgeschwindigkeit, min.	1200 bit/s
Übertragungsgeschwindigkeit, max.	115,5 kbit/s
Leitungslänge, max.	500 m
<b>Point-to-Point Protokolle</b>	
Protokoll ASCII	✓
Protokoll STX/ETX	✓
Protokoll 3964(R)	✓
Protokoll RK512	-
Protokoll USS Master	✓
Protokoll Modbus Master	✓
Protokoll Modbus Slave	-
Spezielle Protokolle	-

Artikelnr.	315-4EC12
<b>Funktionalität RJ45 Schnittstellen</b>	
Bezeichnung	X5
Physik	Ethernet 10/100 MBit
Anschluss	RJ45
Potenzialgetrennt	✓
PG/OP Kommunikation	✓
max. Anzahl Verbindungen	4
Produktiv Verbindungen	-
<b>Funktionalität RJ45 Schnittstellen</b>	
Bezeichnung	X8
Physik	Ethernet 10/100 MBit
Anschluss	RJ45
Potenzialgetrennt	✓
PG/OP Kommunikation	✓
max. Anzahl Verbindungen	8
Produktiv Verbindungen	✓
<b>Ethernet Kommunikations CP</b>	
Anzahl projektierbarer Verbindungen, max.	8
Anzahl via NetPro projektierbarer Verbindungen, max.	8
S7-Verbindungen	BSEND, BRCV, GET, PUT, Verbindungsaufbau aktiv und passiv
Nutzdaten je S7-Verbindung, max.	32 KB
TCP-Verbindungen	FETCH PASSIV, WRITE PASSIV, Verbindungsaufbau passiv über Hantierungsbaustein
Nutzdaten je TCP-Verbindung, max.	64 KB
ISO-Verbindungen	-
Nutzdaten je ISO-Verbindung, max.	-
ISO on TCP Verbindungen (RFC 1006)	FETCH PASSIV, WRITE PASSIV, Verbindungsaufbau passiv über Hantierungsbaustein
Nutzdaten je ISO on TCP-Verbindung, max.	32 KB
UDP-Verbindungen	-
Nutzdaten je UDP-Verbindung, max.	-
UDP-Multicast-Verbindungen	-
UDP-Broadcast-Verbindungen	-
<b>Ethernet Offene Kommunikation</b>	
Anzahl Verbindungen, max.	8
Nutzdaten je ISO on TCP-Verbindung, max.	8 KB

## Technische Daten

Artikelnr.	315-4EC12
Nutzdaten je native TCP-Verbindung, max.	8 KB
Nutzdaten je ad-hoc TCP-Verbindung, max.	1460 Byte
Nutzdaten je UDP-Verbindung, max.	1472 Byte
<b>EtherCAT Master</b>	
Anzahl der EtherCAT-Slaves	128
Aktualisierungszeit	500 µs .. 512 ms
Adressbereich Eingänge, max.	2 KB
Adressbereich Ausgänge, max.	2 KB
EoE Unterstützung	✓
FoE Unterstützung	✓
Distributed Clock Unterstützung	✓
Hotconnect Slaves	✓
<b>Management &amp; Diagnose</b>	
Protokolle	ICMP LLC
Web based Diagnose	-
NCM Diagnose	✓
<b>Gehäuse</b>	
Material	PPE
Befestigung	Profilschiene System 300
<b>Mechanische Daten</b>	
Abmessungen (BxHxT)	80 mm x 125 mm x 120 mm
Gewicht	430 g
<b>Umgebungsbedingungen</b>	
Betriebstemperatur	0 °C bis 60 °C
Lagertemperatur	-25 °C bis 70 °C
<b>Zertifizierungen</b>	
Zertifizierung nach UL508	in Vorbereitung

## 5 Einsatz CPU 315-4EC12

### 5.1 Montage



*Informationen zur Montage und zur Verdrahtung:  
↳ Kapitel 3 "Montage und Aufbaurichtlinien" auf Seite 17*

### 5.2 Anlaufverhalten

#### Stromversorgung einschalten

Nach dem Einschalten der Stromversorgung geht die CPU in den Betriebszustand über, der am Betriebsartenschalter eingestellt ist.

#### Auslieferungszustand

Im Auslieferungszustand ist die CPU urgelöscht. Nach einem STOP→RUN Übergang geht die CPU ohne Programm in RUN.

#### Anlauf mit gültiger Projektierung in der CPU

Die CPU geht mit dem Programm, das sich im batteriegepufferten RAM befindet, in RUN.

#### Anlauf bei leerem Akku

- Der Akku wird direkt über die eingebaute Spannungsversorgung über eine Ladeelektronik geladen und gewährleistet eine Pufferung für min. 30 Tage. Wird dieser Zeitraum überschritten, kann es zur vollkommenen Entladung des Akkus kommen. Hierbei wird das batteriegepufferte RAM gelöscht.
- In diesem Zustand führt die CPU ein Urlöschen durch. Ist eine MMC gesteckt, werden Programmcode und Datenbausteine von der MMC in den Arbeitsspeicher der CPU übertragen. Ist keine MMC gesteckt, transferiert die CPU permanent abgelegte "protected" Bausteine, falls diese vorhanden sind, in den Arbeitsspeicher.
- Abhängig von der Stellung des Betriebsartenschalters geht die CPU in RUN, sofern der OB81 vorhanden ist, bzw. bleibt im STOP. Dieser Vorgang wird im Diagnosepuffer unter folgendem Eintrag festgehalten: "Start Urlöschen automatisch (ungepuffert NetzEIN)".



#### **VORSICHT!**

Bei leerem Akku läuft die CPU nach einem Spannungsreset mit einem BAT-Fehler an und führt ein automatisches Urlöschen der CPU durch. Den BAT-Fehler können Sie wieder löschen, wenn einmalig beim Power-Cycle zwischen dem Aus- und Einschalten der Versorgungsspannung mindestens 30sec. liegen und der Akku der CPU voll geladen ist. Ansonsten bleibt bei einem kurzen Power-Cycle der BAT-Fehler bestehen und die CPU wird urgelöscht.

## 5.3 Adressierung

### 5.3.1 Übersicht

Damit die gesteckten Peripheriemodule gezielt angesprochen werden können, müssen ihnen bestimmte Adressen in der CPU zugeordnet werden. Beim Hochlauf der CPU vergibt diese steckplatzabhängig automatisch von 0 an aufsteigend Peripherieadressen für die gesteckten digitalen Ein- /Ausgabe-Module.

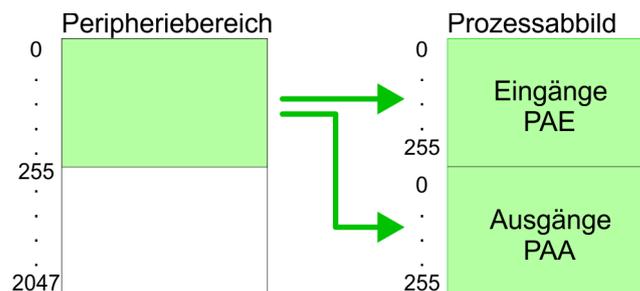
Sofern keine Hardwareprojektierung vorliegt, legt die CPU gesteckte Analog- Module bei der automatischen Adressierung auf gerade Adressen ab 256 ab.

### 5.3.2 Adressierung Rückwandbus Peripherie

Bei der CPU 315-4EC12 gibt es einen Peripheriebereich (Adresse 0 ... 2047) und ein Prozessabbild der Ein- und Ausgänge (je Adresse 0 ... 255). Beim Prozessabbild werden die Signalzustände der unteren Adresse (0 ... 255) zusätzlich in einem besonderen Speicherbereich gespeichert.

Das Prozessabbild ist in zwei Teile gegliedert:

- Prozessabbild der Eingänge (PAE)
- Prozessabbild der Ausgänge (PAA)



Nach jedem Zyklusdurchlauf wird das Prozessabbild aktualisiert.

#### Maximale Anzahl steckbarer Module

Für die CPU 315-4EC12 können Sie bis zu 8 Peripherie-Module pro Zeile projektieren.

Für die Projektierung von Modulen, die über die Anzahl von 8 hinausgehen, können Zeilenanschlungen verwendet werden. Hierbei setzen Sie im Siemens Hardware-Konfigurator auf Ihre 1. Profilschiene auf Steckplatz 3 die Anschaltung IM 360 aus dem Hardware-Katalog. Nun können Sie Ihr System um bis zu 3 Profilschienen ergänzen, indem Sie jede auf Steckplatz 3 mit einer IM 361 von Siemens beginnen. Unter Berücksichtigung des max. Summenstroms können bei der CPU 315-4EC12 von VIPA bis zu 32 Module in einer Zeile angeordnet werden. Hierbei ist die Montage der IM 360/361 Anschaltungen von Siemens nicht erforderlich.

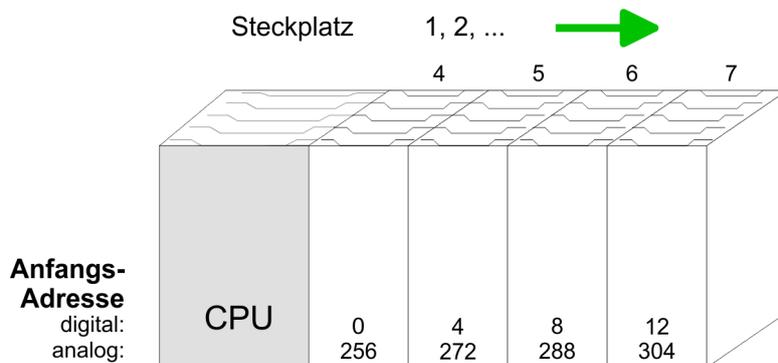
#### Über Hardware-Konfiguration Adressen definieren

Über Lese- bzw. Schreibzugriffe auf die Peripheriebytes oder auf das Prozessabbild können Sie die Module ansprechen. Mit einer Hardware-Konfiguration können Sie Adressen definieren. Klicken Sie hierzu auf die Eigenschaften des entsprechenden Moduls und stellen Sie die gewünschte Adresse ein.

### Automatische Adressierung

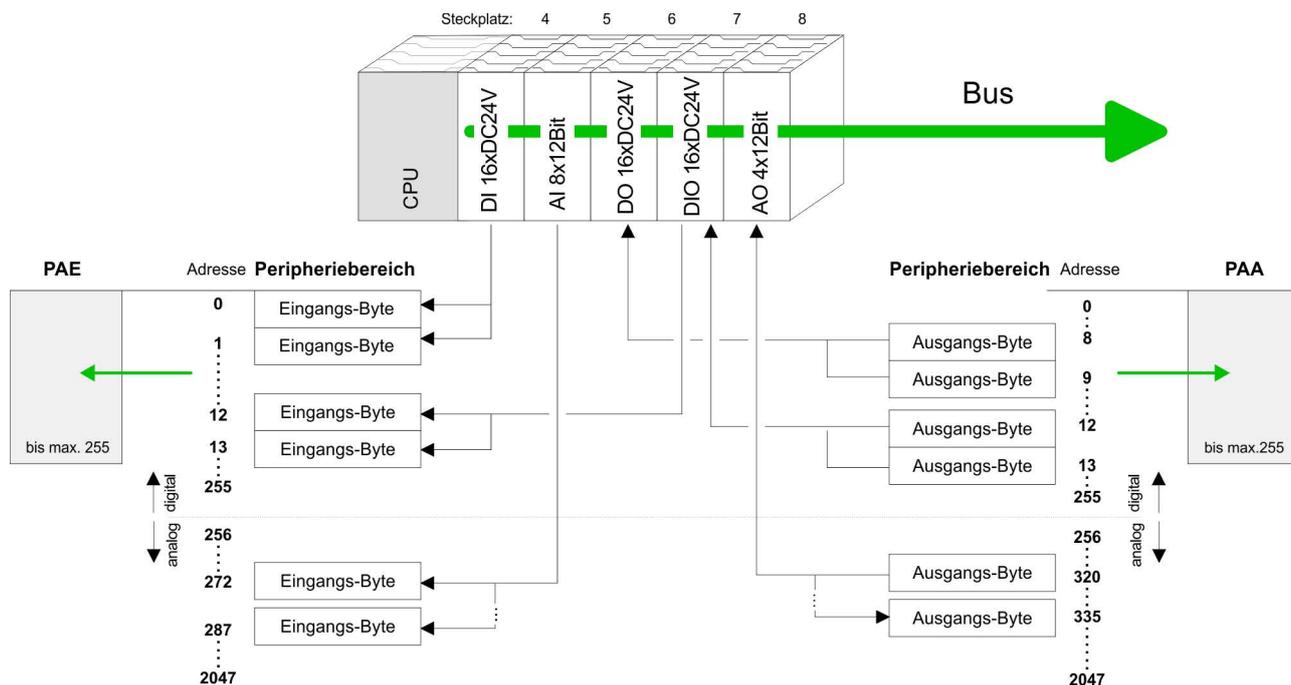
Falls Sie keine Hardware-Konfiguration verwenden möchten, tritt eine automatische Adressierung in Kraft. Bei der automatischen Adressierung belegen steckplatzabhängig DIOs immer 4Byte und AIOs, FMs, CPs immer 16Byte am Bus. Nach folgenden Formeln wird steckplatzabhängig die Anfangsadresse ermittelt, ab der das entsprechende Modul im Adressbereich abgelegt wird:

- DIOs: Anfangsadresse =  $4 \times (\text{Steckplatz} - 1)$
- AIOs, FMs, CPs: Anfangsadresse =  $16 \times (\text{Steckplatz} - 1) + 256$



### Beispiel Automatische Adressierung

In dem nachfolgenden Beispiel ist die Funktionsweise der automatischen Adressierung nochmals aufgeführt:



## 5.4 Hardware-Konfiguration - CPU

### Voraussetzung

Die Konfiguration der CPU erfolgt im "Hardware-Konfigurator" von Siemens. Der Hardware-Konfigurator ist Bestandteil des Siemens SIMATIC Managers. Er dient der Projektierung. Bitte verwenden Sie für die Projektierung den Siemens SIMATIC Manager ab V 5.5 SP2. Die Module, die hier projiziert werden können, entnehmen Sie dem Hardware-Katalog, ggf. müssen Sie mit "Extras → Katalog aktualisieren" den Hardware-Katalog aktualisieren.

Für die Projektierung werden fundierte Kenntnisse im Umgang Siemens SIMATIC Manager und dem Hardware-Konfigurator vorausgesetzt!



Bitte beachten Sie, dass diese SPEED7-CPU 4 AKKUs besitzt. Nach einer arithmetischen Operation (+I, -I, \*I, /I, +D, -D, \*D, /D, MOD, +R, -R, \*R, /R) wird der Inhalt des AKKUs 3 und 4 in die AKKUs 2 und 3 geladen. Dies kann bei Programmen, die einen unveränderten AKKU 2 voraussetzen, zu Konflikten führen.

Nähere Informationen hierzu finden Sie im Handbuch "VIPA Operationsliste SPEED7" unter "Unterschiede zwischen SPEED7 und 300V Programmierung".

**Vorgehensweise**

Steckpl.	Modul
1	
2	<b>CPU 315-2PN/DP</b>
X1	MPI/DP
X2	PN-IO
X2...	Port 1
X2...	Port 2
3	

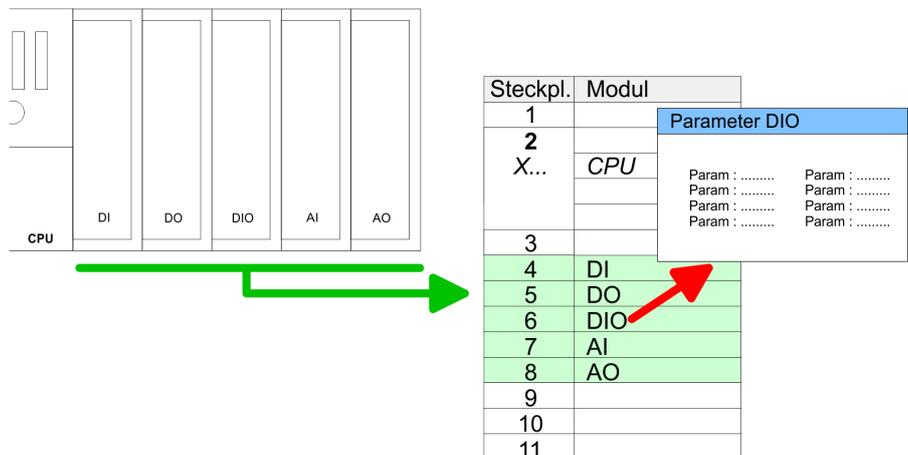
Um kompatibel mit dem Siemens SIMATIC Manager zu sein, sind folgende Schritte durchzuführen:

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 315-2EH14-0AB0 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.

**5.5 Hardware-Konfiguration - I/O-Module**

**Hardware-Konfiguration der Module**

Binden Sie nach der Hardware-Konfiguration der CPU beginnend mit Steckplatz 4 Ihre System 300 Module auf dem Bus in der gesteckten Reihenfolge ein.



**Parametrierung** Zur Parametrierung doppelklicken Sie in Ihrer Steckplatzübersicht auf das zu parametrierende Modul. Daraufhin öffnet sich ein Dialogfenster. Hier können Sie Ihre Parametereinstellungen vornehmen. Unter Einsatz der SFCs 55, 56 und 57 können Sie zur Laufzeit Parameter ändern und an die entsprechenden Module übertragen. Hierbei sind die modulspezifischen Parameter in sogenannten "Datensätzen" abzulegen. Näheres zum Aufbau der Datensätze finden Sie in der Beschreibung zu den Modulen.

**Buserweiterung mit IM 360 und IM 361** Für die Projektierung von Modulen, die über die Anzahl von 8 hinausgehen, können Zeilenanschlungen verwendet werden. Hierbei setzen Sie im Siemens Hardware-Konfigurator auf Ihre 1. Profilschiene auf Steckplatz 3 die Anschaltung IM 360 aus dem Hardware-Katalog. Nun können Sie Ihr System um bis zu 3 Profilschienen ergänzen, indem Sie jede auf Steckplatz 3 mit einer IM 361 von Siemens beginnen. Unter Berücksichtigung des max. Summenstroms können bei VIPA-SPEED7-CPU's bis zu 32 Module in einer Zeile angeordnet werden. Hierbei ist die Montage der IM 360/361 Anschaltungen von Siemens nicht erforderlich.

## 5.6 Hardware-Konfiguration - Ethernet-PG/OP-Kanal

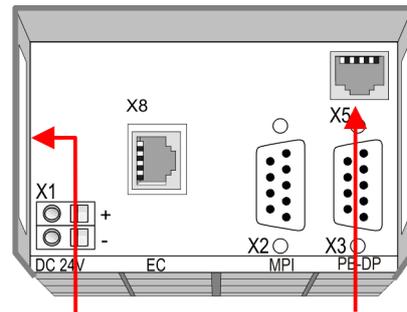
**Übersicht** Die CPU 315-4EC12 hat einen Ethernet-PG/OP-Kanal integriert. Über diesen Kanal können Sie Ihre CPU programmieren und fernwarten. Mit dem PG/OP-Kanal haben Sie auch Zugriff auf die interne Web-Seite, auf der Sie Informationen zu Firmwarestand, angebundene Peripherie, aktuelle Zyklus-Zeiten usw. finden. Bei Erstinbetriebnahme bzw. nach dem Rücksetzen auf Werkseinstellungen besitzt der Ethernet-PG/OP-Kanal keine IP-Adresse. Damit Sie online über den Ethernet-PG/OP-Kanal auf die CPU zugreifen können, müssen Sie diesem gültige IP-Adress-Parameter über den Siemens SIMATIC Manager zuordnen. Diesen Vorgang nennt man "Initialisierung" oder "Urtaufe".

**Montage und Inbetriebnahme**

1. ► Bauen Sie Ihr System 300S mit Ihrer CPU auf.
2. ► Verdrahten Sie das System, indem Sie die Leitungen für Spannungsversorgung und Signale anschließen.
3. ► Verbinden Sie die Ethernet-Buchse des Ethernet-PG/OP-Kanals mit Ethernet.
4. ► Schalten Sie die Spannungsversorgung ein  
⇒ Nach kurzer Hochlaufzeit ist der CP bereit für die Kommunikation. Er besitzt ggf. noch keine IP-Adressdaten und erfordert eine Urtaufe.

**"Urtaufe" über Zielsystemfunktionen**

Die Urtaufe über die Zielsystemfunktion erfolgt nach folgender Vorgehensweise:

**Ethernet address**

1. Ethernet PG/OP channel
2. EtherCAT master

**PG/OP channel**

- Ermitteln Sie die aktuelle Ethernet (MAC) Adresse Ihres Ethernet PG/OP-Kanals. Sie finden diese immer als 1. Adresse unter der Frontklappe der CPU auf einem Aufkleber auf der linken Seite.

**IP-Adress-Parameter zuweisen**

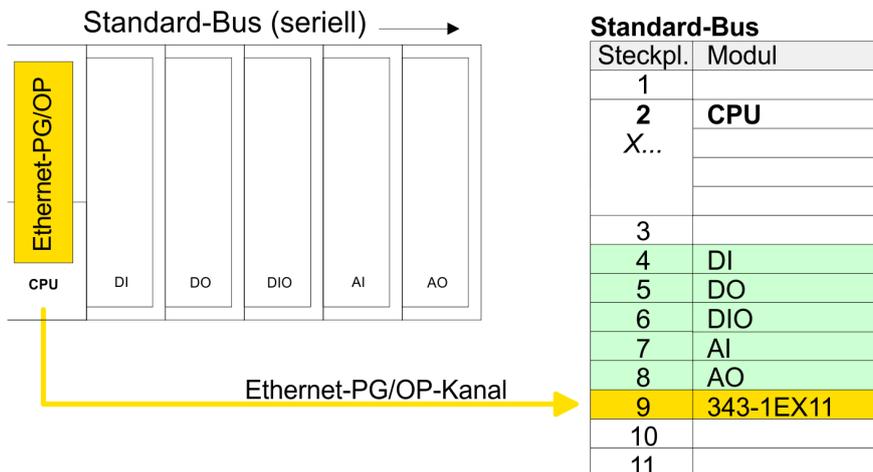
Gültige IP-Adress-Parameter erhalten Sie von Ihrem Systemadministrator. Die Zuweisung der IP-Adress-Daten erfolgt online im Siemens SIMATIC Manager ab Version V 5.3 & SP3 nach folgender Vorgehensweise:

1. ▶ Starten Sie den Siemens SIMATIC Manager und stellen Sie über *"Extras → PG/PC-Schnittstelle einstellen"* auf *"TCP/IP -> Netzwerkkarte ...."* ein.
2. ▶ Öffnen Sie mit *"Zielsystem → Ethernet-Teilnehmer bearbeiten"* das gleichnamige Dialogfenster.
3. ▶ Benutzen Sie die Schaltfläche [Durchsuchen], um die über MAC-Adresse erreichbaren Geräte zu ermitteln oder tragen Sie die MAC-Adresse ein. Die MAC-Adresse finden Sie auf dem 1. Aufkleber unter der Frontklappe der CPU.
4. ▶ Wählen Sie ggf. bei der Netzwerksuche aus der Liste die Baugruppe mit der Ihnen bekannten MAC-Adresse aus.
5. ▶ Stellen Sie nun die IP-Konfiguration ein, indem Sie IP-Adresse, Subnetz-Maske und den Netzübergang eintragen.
6. ▶ Bestätigen Sie mit [IP-Konfiguration zuweisen] Ihre Eingabe.
  - ⇒ Direkt nach der Zuweisung ist der Ethernet-PG/OP-Kanal über die angegebenen IP-Adress-Daten online erreichbar. Der Wert bleibt bestehen, solange dieser nicht neu zugewiesen, mit einer Hardware-Projektierung überschrieben oder Rücksetzen auf Werkseinstellung ausgeführt wird.

**IP-Adress-Parameter in Projekt übernehmen**

1. ▶ Öffnen Sie den Siemens Hardware-Konfigurator und projizieren Sie die Siemens CPU 315-2 PN/DP (6ES7 315-2EH14-0AB0 V3.2).
2. ▶ Projizieren Sie die Module am Standard-Bus.
3. ▶ Für den Ethernet-PG/OP-Kanal ist immer unterhalb der reell gesteckten Module ein Siemens CP 343-1 (SIMATIC 300 \ CP 300 \ Industrial Ethernet \ CP 343-1 \ 6GK7 343-1EX11 0XE0) zu platzieren.

4. Öffnen Sie durch Doppelklick auf den CP 343-1EX11 den Eigenschaften-Dialog und geben Sie für den CP unter "Eigenschaften" die zuvor zugewiesenen IP-Adress-Daten an.
5. Ordnen Sie den CP einem "Subnetz" zu. Ohne Zuordnung werden die IP-Adress-Daten nicht übernommen!
6. Übertragen Sie Ihr Projekt.



## 5.7 Hardware-Konfiguration - Kommunikation

Die Hardware-Konfiguration von PROFIBUS, PtP und EtherCAT ist auf folgenden Seiten beschrieben:

- PROFIBUS-DP
  - Master-Betrieb: ↪ Kapitel 7.4 "Einsatz als PROFIBUS-DP-Master" auf Seite 122
  - Slave-Betrieb: ↪ Kapitel 7.5 "Einsatz als PROFIBUS-DP-Slave" auf Seite 123
- PtP
  - PtP: ↪ Kapitel 6.3 "Einsatz der RS485-Schnittstelle für PtP" auf Seite 94
- EtherCAT
  - EtherCAT-Master: ↪ Kapitel 9.3 "Hardware-Konfiguration - CPU" auf Seite 157

## 5.8 Einstellung Standard CPU-Parameter

### 5.8.1 Parametrierung über Siemens CPU

#### Parametrierung über Siemens CPU 315-2EH14

Da die CPU im Hardware-Konfigurator als Siemens CPU 315-2PN/DP (6ES7 315-2EH14-0AB0 V3.2) zu projektieren ist, können Sie bei der Hardware-Konfiguration unter den "Eigenschaften" der CPU 315-2PN/DP die Standard-Parameter für die VIPA-CPU einstellen. Durch Doppelklick auf die CPU 315-2PN/DP gelangen Sie in das Parametrierfenster für die CPU. Über die Register haben Sie Zugriff auf alle Standard-Parameter Ihrer CPU.

Steckpl.	Modul
1	
2	<b>CPU ...</b>
X1	MPI/DP
X2	PN-IO
X2 P1	Port 1
3	



Parameter CPU	
Param : .....	Param : .....
Param : .....	Param : .....
Param : .....	Param : .....
Param : .....	Param : .....

## 5.8.2 Parameter CPU

### Parameter, die unterstützt werden

Die CPU wertet nicht alle Parameter aus, welche Sie bei der Hardware-Konfiguration einstellen können. Die Parameter folgender Register werden aktuell nicht unterstützt: Taktsynchronalarml, Kommunikation und Web. Folgende Parameter werden zur Zeit in der CPU ausgewertet:

### Allgemein

- Kurzbezeichnung
  - Die Kurzbezeichnung der Siemens CPU 315-2EH14 ist CPU 315-2 PN/DP.
- Bestell-Nr./ Firmware
  - Bestellnummer und Firmware sind identisch zu den Angaben im Fenster "Hardware Katalog".
- Name
  - Als Name steht hier die Kurzbezeichnung der CPU.
  - Wenn Sie den Namen ändern, erscheint dieser im Siemens SIMATIC Manager.
- Anlagenkennzeichen
  - Hier haben Sie die Möglichkeit für die CPU ein spezifisches Anlagenkennzeichen festzulegen.
  - Mit dem Anlagenkennzeichen werden Teile der Anlage eindeutig nach funktionalen Gesichtspunkten gekennzeichnet.
  - Es ist gemäß IEC 1346-1 hierarchisch aufgebaut.
- Ortskennzeichen
  - Das Ortskennzeichen ist Teil des Betriebsmittelkennzeichens.
  - Hier können Sie die genaue Lage Ihrer Baugruppe innerhalb Ihrer Anlage angeben.
- Kommentar
  - Hier können Sie den Einsatzzweck der Baugruppe eingeben.

## Anlauf

- Anlauf bei Sollausbau ungleich Istausbau
  - Wenn *"Anlauf bei Sollausbau ungleich Istausbau"* deaktiviert ist und mindestens eine Baugruppe nicht auf dem projektierten Steckplatz steckt, oder dort eine Baugruppe von einem anderen Typ steckt, geht die CPU nicht in RUN und verbleibt in STOP.
  - Wenn *"Anlauf bei Sollausbau ungleich Istausbau"* aktiviert ist, läuft die CPU an, auch wenn Baugruppen nicht auf den projektierten Steckplätzen stecken oder dort Baugruppen eines anderen Typs stecken (z.B. bei Inbetriebnahme).
- Überwachungszeit für Fertigmeldung durch Baugruppen [100ms]
  - Maximale Dauer für die Fertigmeldung aller konfigurierten Baugruppen nach NetZEIN.
  - Hierbei werden auch angebundene PROFIBUS-DP-Slaves berücksichtigt, bis diese parametrierbar sind.
  - Wenn nach Ablauf dieser Zeit die Baugruppen keine Fertigmeldung an die CPU senden, ist der Istausbau ungleich dem Sollausbau.
- Überwachungszeit für Übertragung der Parameter an Baugruppen [100ms]
  - Maximale Dauer für die Übertragung der Parameter an die parametrierbaren Baugruppen.
  - Hierbei werden auch angebundene PROFINET-IO-Devices berücksichtigt, bis diese parametrierbar sind.
  - Wenn nach Ablauf dieser Zeit nicht alle Baugruppen parametrierbar sind, ist der Istausbau ungleich dem Sollausbau.

## Zyklus / Taktmerker

- OB1-Prozessabbild zyklisch aktualisieren
  - Dieser Parameter ist nicht relevant.
- Zyklusüberwachungszeit
  - Hier geben Sie die Zyklusüberwachungszeit in ms ein.
  - Wenn die Zykluszeit die Zyklusüberwachungszeit überschreitet, geht die CPU in STOP.
  - Ursachen für eine Überschreitung:
    - Kommunikationsprozesse
    - Häufung von Alarmereignissen
    - Fehler im CPU-Programm
- Mindestzykluszeit
  - Dieser Parameter ist nicht relevant.
- Zyklusbelastung durch Kommunikation
  - Mit diesem Parameter können Sie die Dauer von Kommunikationsprozessen, welche immer auch die Zykluszeit verlängern, in bestimmten Grenzen steuern.
  - Bei Einstellung der Zyklusbelastung durch Kommunikation auf 50% kann sich eine Verdopplung der OB 1-Zykluszeit ergeben. Außerdem wird der OB 1-Zyklus zusätzlich durch asynchrone Ereignisse (z.B. Prozessalarme) verlängert.
- Größe Prozessabbild der Ein-/Ausgänge
  - Hier können Sie die Größe des Prozessabbilds max. 2048 für die Ein-/ Ausgabe-Peripherie festlegen (Default: 128).

- OB85-Aufruf bei Peripheriezugriffsfehler
  - Sie können die voreingestellte Reaktion der CPU bei Peripheriezugriffsfehlern während der systemseitigen Aktualisierung des Prozessabbildes ändern.
  - Die VIPA-CPU ist so voreingestellt, dass sie bei Peripheriezugriffsfehlern keinen OB 85 aufruft und auch keinen Eintrag im Diagnosepuffer erzeugt.
- Taktmerker
  - Aktivieren Sie dieses Kästchen, wenn Sie einen Taktmerker einsetzen und geben Sie die Nummer des Merkerbytes ein.



*Das gewählte Merkerbyte kann nicht für die Zwischenspeicherung von Daten genutzt werden.*

## Remanenz

- Anzahl Merkerbytes ab MB0
  - Die Anzahl der remanenten Merkerbytes ab Merkerbyte 0 können Sie hier angeben.
- Anzahl S7-Timer ab T0
  - Hier tragen Sie die Anzahl der remanenten S7-Timer ab T0 ein.
- Anzahl S7-Zähler ab Z0
  - Tragen Sie die Anzahl der remanenten S7-Zähler ab Z0 hier ein.
- Bereiche
  - Diese Parameter sind nicht relevant.

## Alarmer

- Priorität
  - Hier werden die Prioritäten angezeigt, nach denen der entsprechende Alarm-OB (Prozessalarm, Verzögerungsalarm, Asynchronfehleralarm) bearbeitet wird.

## Uhrzeitalarmer

- Priorität
  - Dieser Wert ist fix auf 2 eingestellt.
- Aktiv
  - Durch Anwahl von "Aktiv" wird die Funktionalität für Uhrzeitalarmer aktiviert.
- Ausführung
  - Hier wählen Sie aus, wie oft die Alarmer ausgeführt werden sollen.
  - Die Intervalle von minütlich bis jährlich beziehen sich auf die Einstellungen unter *Startdatum* und *Uhrzeit*.
- Startdatum/Uhrzeit
  - Hier geben Sie an, wann der Uhrzeitalarm zum ersten Mal ausgeführt werden soll.
- Teilprozessabbild
  - Dieser Parameter wird nicht unterstützt.

## Weckalarme

- **Priorität**
  - Hier können Sie die Prioritäten bestimmen, nach denen der entsprechende Weckalarm-OB bearbeitet werden soll.
  - Mit Priorität "0" wählen Sie den entsprechenden OB ab.
- **Ausführung**
  - Geben Sie die Zeitabstände in ms an, in denen die Weckalarm-OBs bearbeitet werden.
  - Startzeitpunkt ist der Betriebszustandwechsel von STOP nach RUN.
- **Phasenverschiebung**
  - Geben Sie hier eine Zeit in ms an, um welche der tatsächliche Ausführungszeitpunkt des Weckalarms verzögert werden soll. Dies ist sinnvoll, wenn mehrere Weckalarme aktiv sind.
  - Mit der *Phasenverschiebung* können diese über den Zyklus hinweg verteilt werden.
- **Teilprozessabbild**
  - Dieser Parameter wird nicht unterstützt.

## Diagnose/Uhr

- **STOP-Ursache melden**
  - Aktivieren Sie diesen Parameter, wenn die CPU bei Übergang nach STOP die STOP-Ursache an PG bzw. OP melden soll.
- **Anzahl Meldungen im Diagnosepuffer**
  - Dieser Parameter wird ignoriert. Die CPU besitzt einen Diagnosepuffer (Ringpuffer) für 100 Diagnosemeldungen.
- **Synchronisationsart**
  - Legen Sie hier fest, ob die Uhr andere Uhren synchronisiert oder nicht.
    - als Slave: Die Uhr wird von einer anderen Uhr synchronisiert.
    - als Master: Die Uhr synchronisiert andere Uhren als Master.
    - keine: Es findet keine Synchronisation statt.
- **Zeitintervall**
  - Zeitintervalle, innerhalb welcher die Synchronisation erfolgen soll.
- **Korrekturfaktor**
  - Durch Vorgabe eines Korrekturfaktors in ms können Sie die Abweichung der Uhr innerhalb 24 Stunden ausgleichen.
  - Geht Ihre Uhr innerhalb von 24 Stunden 1s nach, können Sie dies mit dem Korrekturfaktor "+1000" ms ausgleichen.

## Schutz

- **Schutzstufe**
  - Hier können Sie eine von 3 Schutzstufen einstellen, um die CPU vor unbefugtem Zugriff zu schützen.
  - *Schutzstufe 1 (voreingestellt):*  
kein Passwort parametrierbar; keine Einschränkungen
  - *Schutzstufe 2 mit Passwort:*  
Kenntnis des Passworts: lesender und schreibender Zugriff  
Unkenntnis des Passworts: nur lesender Zugriff.
  - *Schutzstufe 3:*  
Kenntnis des Passworts: lesender und schreibender Zugriff  
Unkenntnis des Passworts: weder lesender noch schreibender Zugriff

### 5.8.3 Parameter für DP

Über Doppelklick auf das Submodul DP gelangen Sie in den Eigenschaften-Dialog des PROFIBUS-Teils.

#### Allgemein

- Kurzbezeichnung: Hier wird als Kurzbezeichnung "DP" für PROFIBUS-DP aufgeführt.
- Bestell-Nr.: Eine Bestell-Nr. wird nicht angezeigt.
- Name: Hier steht die Bezeichnung "DP". Wenn Sie die Bezeichnung ändern, erscheint die neue Bezeichnung im Siemens SIMATIC Manager.
- Schnittstelle: Hier wird die PROFIBUS-Adresse eingeblendet.
- Eigenschaften: Über diese Schaltfläche können Sie die Eigenschaften der PROFIBUS-DP-Schnittstelle einstellen.
- Kommentar: Hier können Sie den Einsatzzweck der PROFIBUS-Schnittstelle eingeben.

#### Adresse

- Diagnose: Geben Sie hier eine Diagnoseadresse für PROFIBUS-DP an. Über diese Adresse bekommt die CPU eine Rückmeldung im Fehlerfall.
- Betriebsart: Hier können Sie die entsprechende Betriebsart des PROFIBUS-Teils einstellen. Näheres hierzu finden Sie im Teil "Einsatz unter PROFIBUS".
- Konfiguration: Hier können Sie in der Betriebsart "DP-Slave" Ihr Slave-System konfigurieren. Näheres hierzu finden im Teil "Einsatz unter PROFIBUS".
- Uhr: Diese Parameter werden nicht unterstützt.

### 5.8.4 Parameter für MPI/DP

Über Doppelklick auf das Submodul MPI/DP gelangen Sie in den Eigenschaften-Dialog der MPI-Schnittstelle.

#### Allgemein

- Kurzbezeichnung: Hier wird als Kurzbezeichnung "MPI/DP" für die MPI-Schnittstelle aufgeführt.
- Bestell-Nr.: Hier erfolgt keine Anzeige.
- Name: Unter *Name* finden Sie die Bezeichnung "MPI/DP". Wenn Sie den Namen ändern, erscheint der neue Name im Siemens SIMATIC Manager.
- Typ: Bitte beachten Sie, dass die VIPA CPU ausschließlich den Typ "MPI" unterstützt.
- Schnittstelle: Hier wird die MPI-Adresse eingeblendet.
- Eigenschaften: Über diese Schaltfläche können Sie die Eigenschaften der MPI-Schnittstelle einstellen.
- Kommentar: Geben Sie hier den Einsatzzweck der MPI-Schnittstelle an.

#### Adresse

- Diagnose: Geben Sie hier eine Diagnoseadresse für die MPI-Schnittstelle an. Über diese Adresse bekommt die CPU eine Rückmeldung im Fehlerfall.
- Betriebsart, Konfiguration, Uhr: Diese Parameter werden nicht unterstützt.

## 5.9 Einstellung VIPA-spezifische CPU-Parameter

### 5.9.1 Vorgehensweise

#### Übersicht

Mit Ausnahme der VIPA-spezifischen CPU-Parameter erfolgt die CPU-Parametrierung im Parameter-Dialog der Siemens CPU. Durch Einbindung der SPEEDBUS.GSD können Sie in der Hardware-Konfiguration VIPA-spezifische Parameter einstellen. Hierbei haben Sie Zugriff auf folgende Parameter:

- Funktion RS485 X3 (PtP, Synchronisation DP-Master und CPU)
- Token Watch
- Anzahl Remanenzmerker, Timer, Zähler
- Priorität OB 28, OB 29, OB 57
- MPI Adresse X2

#### Voraussetzung

Damit Sie die VIPA-spezifischen CPU-Parameter einstellen können, ist die Installation der SPEEDBUS.GSD von VIPA im Hardwarekatalog erforderlich. Nach der Installation können Sie die CPU in einem PROFIBUS-Master-System projektieren und entsprechend die Parameter anpassen.

#### SPEEDBUS.GSD installieren

Die GSD (Geräte-Stamm-Datei) ist in folgenden Sprachversionen online verfügbar. Weitere Sprachen erhalten Sie auf Anfrage:

Name	Sprache
SPEEDBUS.GSD	deutsch (default)
SPEEDBUS.GSG	deutsch
SPEEDBUS.GSE	englisch

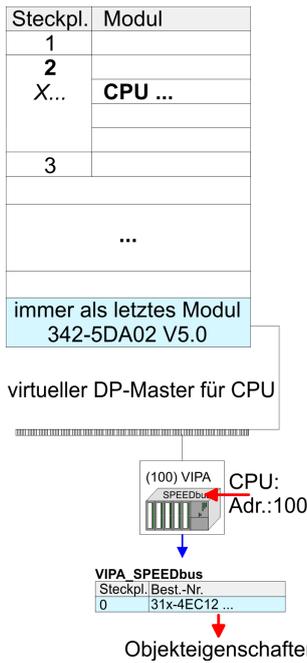
Die GSD-Dateien finden Sie auf [www.vipa.com](http://www.vipa.com) im "Service"-Bereich. Die Einbindung der SPEEDBUS.GSD erfolgt nach folgender Vorgehensweise:

1. Gehen Sie auf [www.vipa.com](http://www.vipa.com)
2. Klicken Sie auf "Service → Download → GSD- und EDS-Files → Profibus"
3. Laden Sie die Datei Cx000023\_Vxxx.
4. Extrahieren Sie die Datei in Ihr Arbeitsverzeichnis. Die SPEEDBUS.GSD befindet sich im Verzeichnis VIPA\_System\_300S.
5. Starten Sie den Hardware-Konfigurator von Siemens.
6. Schließen Sie alle Projekte.
7. Gehen Sie auf "Extras → Neue GSD-Datei installieren".
8. Navigieren Sie in das Verzeichnis VIPA\_System\_300S und geben Sie **SPEEDBUS.GSD** an.
  - ⇒ Alle SPEED7-CPU's und -Module des System 300S von VIPA sind jetzt im Hardwarekatalog unter Profibus-DP / Weitere Feldgeräte / I/O / VIPA\_SPEEDBUS enthalten.

#### Hardware-Konfiguration

Die Einbindung der CPU 315-4EC12 erfolgt in Form eines virtuellen PROFIBUS Master-Systems nach folgender Vorgehensweise:

Einstellung VIPA-spezifische CPU-Parameter > VIPA-spezifische Parameter



1. ➤ Führen Sie eine Hardware-Konfiguration für die CPU durch. *☞ Kapitel 5.4 "Hardware-Konfiguration - CPU" auf Seite 43*
2. ➤ Projektieren Sie immer als letztes Modul einen Siemens DP-Master CP 342-5 (342-5DA02 V5.0). Vernetzen und parametrieren Sie diesen in der Betriebsart "DP-Master".
3. ➤ Binden Sie das Slave-System "VIPA\_SPEEDbus" an. Nach der Installation der SPEEDBUS.GSD finden Sie dieses im Hardware-Katalog unter Profibus-DP / Weitere Feldgeräte / I/O / VIPA\_SPEEDBUS.
4. ➤ Stellen Sie für das Slave-System die PROFIBUS-Adresse 100 ein.
5. ➤ Platzieren Sie auf dem Steckplatz 0 die VIPA CPU 315-4EC12 aus dem Hardware-Katalog von VIPA\_SPEEDbus.
6. ➤ Durch Doppelklick auf die eingefügte CPU 315-4EC12 gelangen Sie in den Eigenschaften-Dialog der CPU.

### 5.9.2 VIPA-spezifische Parameter

Im Eigenschaften-Dialog derVIPA-CPU haben Sie Zugriff auf die nachfolgend aufgeführten Parameter.

#### 5.9.2.1 Funktion RS485 X3

Mit diesem Parameter können Sie die RS485-Schnittstelle auf PtP-Kommunikation (**point to point**) umschalten bzw. das Synchronisationsverhalten zwischen DP-Master-System und CPU vorgeben:

Deaktiviert	Deaktiviert die RS485-Schnittstelle.
PtP	In dieser Betriebsart wird der PROFIBUS-DP-Master deaktiviert und die RS485-Schnittstelle arbeitet als Schnittstelle für serielle Punkt-zu-Punkt-Kommunikation. Hier können Sie unter Einsatz von Protokollen seriell zwischen zwei Stationen Daten austauschen.
PROFIBUS-DP async	PROFIBUS-DP-Master-Betrieb asynchron zum CPU-Zyklus Die RS485-Schnittstelle ist defaultmäßig auf PROFIBUS-DP async eingestellt. Hier laufen CPU-Zyklus und die Zyklen aller VIPA PROFIBUS-DP-Master an der CPU unabhängig voneinander.
PROFIBUS-DP syncln	Die CPU wartet auf DP-Master-Eingangsdaten.

PROFIBUS-DP syncOut	Das DP-Master-System wartet auf CPU-Ausgangsdaten.
PROFIBUS-DP syncInOut	CPU und DP-Master-System warten aufeinander und bilden damit einen Zyklus.
Default: PROFIBUS-DP async	

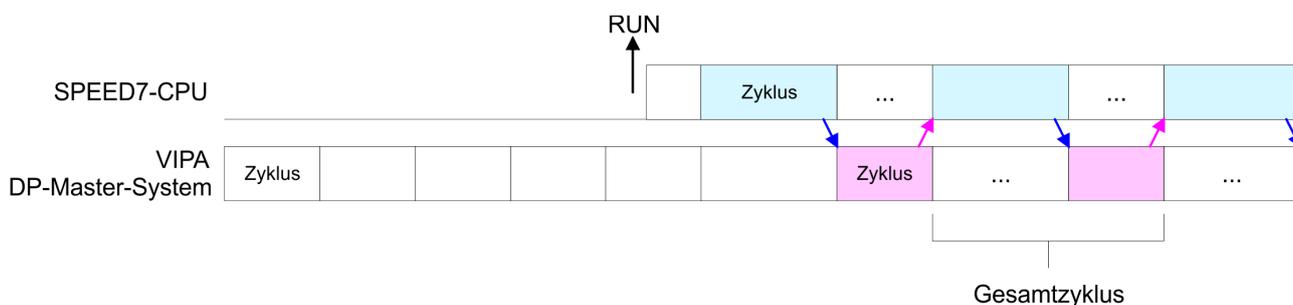
### 5.9.2.1.1 Synchronisation zwischen Master-System und CPU

#### Übersicht

Normalerweise laufen die Zyklen von CPU und DP-Master unabhängig voneinander. Die Zykluszeit der CPU ist die Zeit, welche die CPU für einen OB1-Durchlauf und für das Lesen bzw. Schreiben der Ein- bzw. Ausgänge benötigt. Da die Zykluszeit eines DP-Masters unter anderem abhängig ist von der Anzahl der angebotenen DP-Slaves und der Baud-Rate, entsteht bei jedem angebotenen DP-Master eine andere Zykluszeit. Aufgrund der Asynchronität von CPU und DP-Master ergeben sich für das Gesamtsystem relativ hohe Reaktionszeiten. Über eine Hardware-Konfiguration können Sie, wie oben gezeigt, das Synchronisations-Verhalten zwischen allen VIPA PROFIBUS-DP-Master an der CPU parametrieren. Die verschiedenen Modi für die Synchronisation sind nachfolgend beschrieben.

#### PROFIBUS-DP SyncInOut

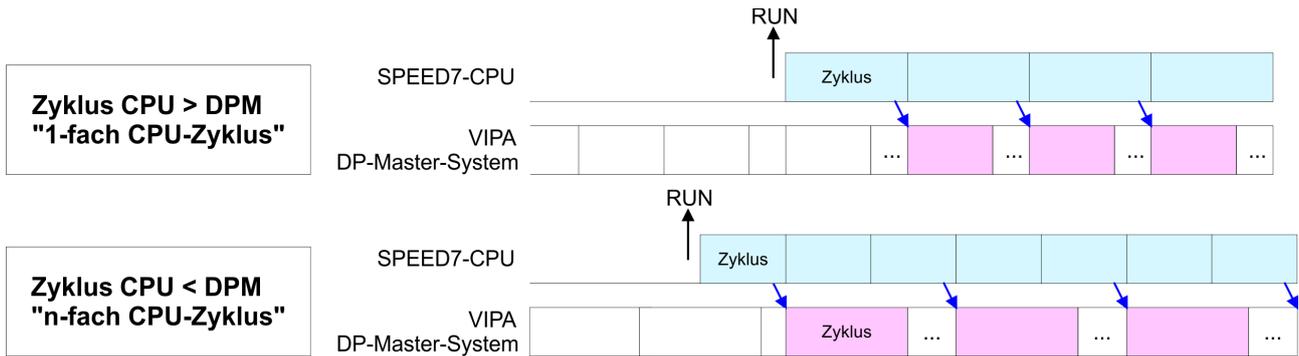
Im PROFIBUS-DP SyncInOut warten CPU und DP-Master-System jeweils aufeinander und bilden damit einen Zyklus. Hierbei ist der Gesamtzyklus die Summe aus dem längsten DP-Master-Zyklus und CPU-Zyklus. Durch diesen Synchronisations-Modus erhalten Sie global konsistente Ein-/ Ausgabedaten, da innerhalb des Gesamtzyklus CPU und das DP-Master-System nacheinander mit den gleichen Ein- bzw. Ausgabedaten arbeiten. Gegebenenfalls müssen Sie in diesem Modus die Ansprechüberwachungszeit in den Bus-Parametern erhöhen.



#### PROFIBUS-DP SyncOut

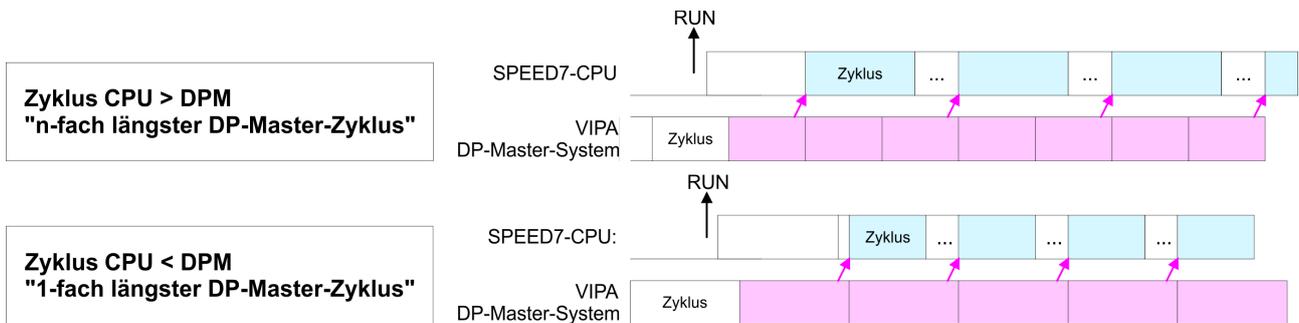
In dieser Betriebsart richtet sich der Zyklus des VIPA DP-Master-Systems nach dem CPU-Zyklus. Geht die CPU in RUN, werden die DP-Master synchronisiert. Sobald deren Zyklus durchlaufen ist, warten diese auf den nächsten Synchronisationsimpuls mit Ausgabedaten der CPU. Auf diese Weise können Sie die Reaktionszeit Ihres Systems verbessern, da Ausgangsdaten möglichst schnell an die DP-Master übergeben werden. Gegebenenfalls müssen Sie in diesem Modus die Ansprechüberwachungszeit in den Bus-Parametern erhöhen.

Einstellung VIPA-spezifische CPU-Parameter > VIPA-spezifische Parameter



**PROFIBUS-DP Syncln**

In der Betriebsart PROFIBUS-DP Syncln wird der CPU-Zyklus auf den Zyklus des VIPA PROFIBUS-DP-Master-Systems synchronisiert. Hierbei richtet sich der CPU-Zyklus nach dem VIPA DP-Master mit der längsten Zykluszeit. Geht die CPU in RUN, wird diese mit allen VIPA DP-Master synchronisiert. Sobald die CPU ihren Zyklus durchlaufen hat, wartet diese, bis das DP-Master-System mit dem Synchronimpuls neue Eingangsdaten liefert. Gegebenenfalls müssen Sie in diesem Modus die Zyklusüberwachungszeit der CPU erhöhen.



**5.9.2.2 Token Watch**

Über die Vorgaben der PROFIBUS-Bus-Parameter bei der Hardware-Konfiguration ergibt sich eine Token-Zeit für den PROFIBUS. Die Token-Zeit definiert die Zeitdauer, bis das Token wieder beim DP-Master ist. Per Default wird diese Zeit überwacht. Starke Störungen auf dem Bus können aber aufgrund dieser Überwachung zu einem Reboot des DP-Master führen. Hier können Sie mit dem Parameter Token Watch die Überwachung der Token-Zeit aus- bzw. einschalten. Default: Ein

**5.9.2.3 Anzahl Remanenz- Merker**

Geben Sie hier die Anzahl der Merker-Bytes an. Durch Eingabe von 0 wird der Wert übernommen, welchen Sie in den Parametern der Siemens CPU unter Remanenz > Anzahl Merker-Bytes ab MB0 angegeben haben. Ansonsten wird der hier angegebene Wert (1 ... 8192) übernommen. Default: 0

#### 5.9.2.4 Priorität von OB 28, OB 29 und OB 57

Die Priorität legt die Reihenfolge der Unterbrechung des entsprechenden Alarm-OBs fest. Hierbei werden folgende Prioritäten unterstützt: 0 (Alarm-OB ist deaktiviert), 2, 3, 4, 9, 12, 16, 17, 24. Default: 24

#### 5.9.2.5 MPI Adresse X2

Die MPI-Schnittstelle dient zur Verbindung zwischen Programmiergerät und CPU. Hierüber erfolgt beispielsweise die Projektierung und Programmierung. Außerdem dient MPI zur Kommunikation zwischen mehreren CPUs oder zwischen HMIs und CPU. Standardmäßig ist die MPI-Adresse 2 eingestellt. Diesen Wert können Sie hier ändern.

### 5.10 Projekt transferieren

#### Übersicht

Sie haben folgende Möglichkeiten für den Projekt-Transfer in die CPU:

- Transfer über MPI
- Transfer über Ethernet
- Transfer über MMC

#### 5.10.1 Transfer über MPI

##### Allgemein

Für den Transfer über MPI besitzt die CPU folgende Schnittstelle:

- X2: MPI-Schnittstelle

##### Netz-Struktur

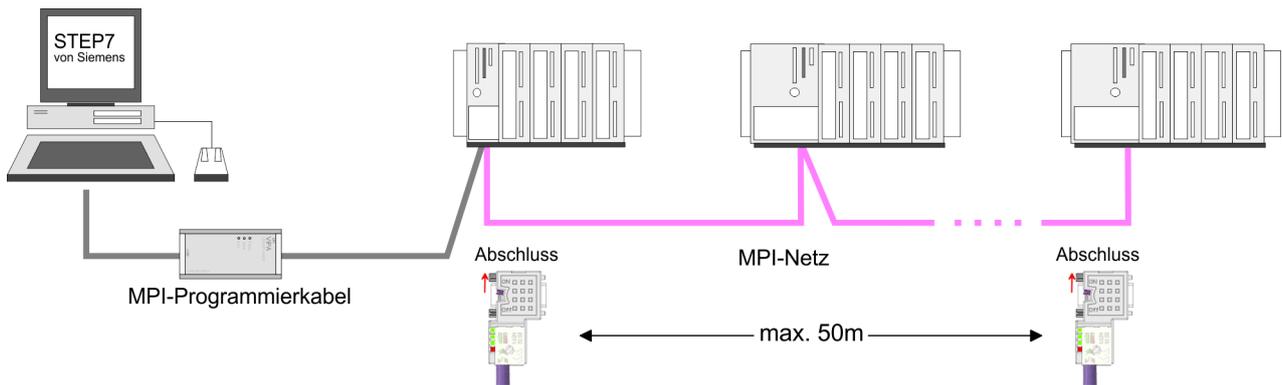
Der Aufbau eines MPI-Netzes gleicht elektrisch dem Aufbau eines PROFIBUS-Netzes. Das heißt, es gelten dieselben Regeln und Sie verwenden für beide Netze die gleichen Komponenten zum Aufbau. Die einzelnen Teilnehmer werden über Busanschlussstecker und PROFIBUS-Kabel verbunden. Bitte beachten Sie hierbei für die CPU 315-4EC12, dass die Gesamtausdehnung des MPI-Netzes 50m nicht überschreitet. Defaultmäßig wird das MPI-Netz mit 187,5kBaude betrieben. VIPA-CPU's werden mit der MPI-Adresse 2 ausgeliefert.

##### MPI-Programmierkabel

Die MPI-Programmierkabel erhalten Sie in verschiedenen Varianten von VIPA. Die Kabel bieten einen RS232- bzw. USB-Anschluss für den PC und einen busfähigen RS485-Anschluss für die CPU. Aufgrund des RS485-Anschlusses dürfen Sie die MPI-Programmierkabel direkt auf einen an der RS485-Buchse schon gesteckten Stecker aufstecken. Jeder Busteilnehmer identifiziert sich mit einer eindeutigen Adresse am Bus, wobei die Adresse 0 für Programmiergeräte reserviert ist.

##### Abschlusswiderstand

Eine Leitung muss mit ihrem Wellenwiderstand abgeschlossen werden. Hierzu schalten Sie den Abschlusswiderstand am ersten und am letzten Teilnehmer eines Netzes oder eines Segments zu. Achten Sie darauf, dass die Teilnehmer, an denen der Abschlusswiderstand zugeschaltet ist, immer mit Spannung versorgt sind. Ansonsten kann es zu Störungen auf dem Bus kommen.



### Vorgehensweise Transfer über MPI- Schnittstelle

1. ▶ Verbinden Sie Ihren PC über ein MPI-Programmierkabel mit der MPI-Buchse Ihrer CPU.
2. ▶ Laden Sie im Siemens SIMATIC Manager Ihr Projekt.
3. ▶ Wählen Sie im Menü "Extras → PG/PC-Schnittstelle einstellen".
4. ▶ Wählen Sie in der Auswahlliste "PC Adapter (MPI)" aus; ggf. müssen Sie diesen erst hinzufügen und klicken Sie auf [Eigenschaften].
5. ▶ Stellen Sie im Register MPI die Übertragungsparameter Ihres MPI-Netzes ein und geben Sie eine gültige Adresse an.
6. ▶ Wechseln Sie in das Register *Lokaler Anschluss*.
7. ▶ Geben Sie den COM-Port des PCs an und stellen Sie für Ihr MPI-Programmierkabel die Übertragungsrate 38400Baud ein.
8. ▶ Mit "Zielsystem → Laden in Baugruppe" können Sie Ihr Projekt über MPI in die CPU übertragen und mit "Zielsystem → RAM nach ROM kopieren" auf einer MMC sichern, falls diese gesteckt ist.

### 5.10.2 Transfer über Ethernet

Die CPU besitzt für den Transfer über Ethernet folgende Schnittstelle:

- X5: Ethernet-PG/OP-Kanal

#### Initialisierung

Damit Sie auf die Ethernet-Schnittstelle online zugreifen können, müssen Sie dieser durch die "Initialisierung" bzw. "Urtaufe" IP-Adress-Parameter zuweisen. ↪ Kapitel 5.6 "Hardware-Konfiguration - Ethernet-PG/OP-Kanal" auf Seite 45

#### Transfer

1. ▶ Für den Transfer verbinden Sie, wenn nicht schon geschehen, die entsprechende Ethernet-Buchse mit Ihrem Ethernet.
2. ▶ Öffnen Sie Ihr Projekt im Siemens SIMATIC Manager.
3. ▶ Stellen Sie über "Extras → PG/PC-Schnittstelle" den Zugriffsweg "TCP/IP → Netzwerkkarte ...." ein.

4. ▶ Gehen Sie auf *"Zielsystem → Laden in Baugruppe"* es öffnet sich das Dialogfenster *"Zielbaugruppe auswählen"*. Wählen Sie die Zielbaugruppe aus und geben Sie als Teilnehmeradresse die IP-Adress-Parameter des entsprechenden Ethernet-Schnittstelle an. Sofern keine neue Hardware-Konfiguration in die CPU übertragen wird, wird die hier angegebene Ethernet-Verbindung dauerhaft als Transferkanal im Projekt gespeichert.
5. ▶ Starten Sie mit [OK] den Transfer.



*Systembedingt kann es zu einer Meldung kommen, dass sich die projektierte von der Zielbaugruppe unterscheidet. Quittieren Sie diese Meldung mit [OK].*

*→ Ihr Projekt wird übertragen und kann nach der Übertragung in der CPU ausgeführt werden.*

### 5.10.3 Transfer über MMC

Die MMC (**Memory Card**) dient als externes Speichermedium. Es dürfen sich mehrere Projekte und Unterverzeichnisse auf einer MMC befinden. Bitte beachten Sie, dass sich Ihre aktuelle Projektierung im Root-Verzeichnis befindet und einen der folgenden Dateinamen hat:

- S7PROG.WLD
- AUTOLOAD.WLD

Mit *"Datei → Memory-Card-Datei → Neu"* können Sie im Siemens SIMATIC Manager eine WLD-Datei erzeugen. Danach kopieren Sie aus dem Baustein-Ordner Ihres Projekts alle Bausteine und die *Systemdaten* in die WLD-Datei.

#### Transfer MMC → CPU

Das Übertragen des Anwenderprogramms von der MMC in die CPU erfolgt je nach Dateiname nach Urlöschen oder nach PowerON.

- *S7PROG.WLD* wird nach Urlöschen von der MMC gelesen.
- *AUTOLOAD.WLD* wird nach NetzEIN von der MMC gelesen.

Das Blinken der MC-LED der CPU kennzeichnet den Übertragungsvorgang. Bitte beachten Sie, dass Ihr Anwenderspeicher ausreichend Speicherplatz für Ihr Anwenderprogramm bietet, ansonsten wird Ihr Anwenderprogramm unvollständig geladen und die SF-LED leuchtet.

#### Transfer CPU → MMC

Bei einer in der CPU gesteckten MMC wird durch einen Schreibbefehl der Inhalt des batteriegepufferten RAMs als *S7PROG.WLD* auf die MMC übertragen.

Den Schreibbefehl starten Sie aus dem Siemens SIMATIC Manager auf Bausteinebene über *"Zielsystem → RAM nach ROM kopieren"*. Während des Schreibvorgangs blinkt die MC-LED. Erlischt die LED, ist der Schreibvorgang beendet.

Soll dieses Projekt automatisch nach einem NetzEIN von der MMC geladen werden, so müssen Sie dieses auf der MMC in *AUTOLOAD.WLD* umbenennen.

Zugriff auf integrierte Web-Seite

**Kontrolle des Transfer-  
vorgangs**

Nach einem MMC-Zugriff erfolgt ein Diagnose-Eintrag der CPU. Zur Anzeige der Diagnoseeinträge gehen Sie im Siemens SIMATIC Manager auf "Zielsystem → Baugruppenzustand". Über das Register "Diagnosepuffer" gelangen Sie in das Diagnosefenster.

Nähere Informationen zu den Ereignis-IDs ↪ "Übersicht der Ereignis-IDs" auf Seite 76.

**5.11 Zugriff auf integrierte Web-Seite**

**Zugriff auf Web-Seite**

Über die IP-Adresse des Ethernet-PG/OP-Kanals steht Ihnen eine Web-Seite zur Verfügung, die Sie mit einem Internet-Browser aufrufen können. Auf der Web-Seite finden Sie Informationen zu Firmwarestand, aktuelle Zyklus-Zeiten usw. Mit dem MMC-Command WEBPAGE wird der aktuelle Inhalt der Web-Seite auf MMC gespeichert.  
↪ Kapitel 5.19 "MMC-Command - Autobefehle" auf Seite 73

**Voraussetzung**

Es wird vorausgesetzt, dass zwischen dem PC mit Internet-Browser und der CPU 315-4EC12 eine Verbindung über den PG/OP-Kanal besteht. Dies können Sie testen über Ping auf die IP-Adresse des Ethernet-PG/OP-Kanals.

**Web-Seite**

Der Zugriff auf die Web-Seite erfolgt über die IP-Adresse des Ethernet-PG/OP-Kanals. Die Web-Seite dient ausschließlich der Informationsausgabe. Die angezeigten Werte können nicht geändert werden.



**CPU mit Ethernet-PG/OP**

Slot 100	
VIPA 315-4EC12 V3.6.1.26 Px000176.pkg, SERIALNUMBER 17601	Best.-Nr., Firmware-Vers., Package, Serien-Nr.
SUPPORTDATA : PRODUCT V36126, HARDWARE V0110, 5679P-V11 , HX000066.110 , Bx000227 V66126, Ax000086 V1210, fx000007.wld V1180, FlashFileSystem : V102	Angaben für den Support
Memorysizes (Bytes): LoadMem : 2097152, Work- MemCode : 524288, WorkMemData : 524288	Angaben zum Speicherausbau Ladespeicher, Arbeitsspei- cher(Code/Daten)
OnBoardEthernet : MacAddress : 0020D50144C1, IP-Add- ress : 172.20.120.62, SubnetMask : 255.255.255.0, Gateway : 172.20.120.62	Ethernet-PG/OP: Adressan- gaben
Cpu state : Run	CPU-Statusangabe
FunctionRS485 X2/COM1: MPI FunctionRS485 X3/COM2: DPM-async	Betriebsart RS485 (MPI: MPI-Betrieb, PtP: Punkt zu Punkt-Betrieb)

Slot 100	
Cycletime [microseconds] : min=0 cur=770 ave=750 max=878	CPU-Zykluszeit: min= minimale cur= aktuelle max= maximale
ArmLoad [percent] : cur=67, max=70	Angaben für den Support
PowerCycleHxRetries : 29, 0, 0, 0, 0	
AutoCompress activated	

Slot 201	
VIPA 342-1DA70 V3.3.0 Px000062.pkg	<b>CPU-Komponente: DP-Master</b> Name, Firmware-Version, Package
SUPPORTDATA : PRODUCT V3300, BB000218 V5300, AB000068 V4170, ModuleType CB2C0010	Angaben für den Support
Cycletime [microseconds] : min=65535000 cur=0 ave=0 max=0 cnt=0	CPU-Zykluszeit: min = minimale cur = aktuelle max = maximale

Slot 206	
V0.0.1 Px000153.pkg, SUPPORTDATA : Bx000562 V00132, AB000125 V0103 PRODUCT V00132, Hx000075 V1100 ModuleType ACDB0100 Address Input 2046	<b>CPU-Komponente: EtherCAT IO-Controller</b> Angaben für den Support

### Standard Bus

Standard Bus	Module am Standard-Bus
BaudRate Read Mode1, BaudRate Write Mode1	Angaben für den Support
Line 1: ModuleType 94F9: IM36x	<i>IM-Anschaltung falls vor- handen</i>
Rack 0 /Slot 4	Rack-Nr. / Steckplatz
ModuleType: 9FC3: Digital Input 32 Baseaddress Input 0	Typkennung des Moduls Projektierte Basisadresse ggf. Firmware-Nr. und Package

Standard Bus	Module am Standard-Bus
Rack 0 /Slot 5 ...	Rack-Nr. / Steckplatz
...	
Line 2: ModuleType A4FE: IM36x	<i>IM-Anschaltung falls vorhanden</i>
Rack 1 /Slot 4	
ModuleType: 9FC3: Digital Input 32 Baseaddress Input 0	Typkennung des Moduls Projektierte Basisadresse ggf. Firmware-Nr. und Package
Rack 1 /Slot 5 ...	Rack-Nr. / Steckplatz

## 5.12 Betriebszustände

### 5.12.1 Übersicht

Die CPU kennt 4 Betriebszustände:

- Betriebszustand STOP
- Betriebszustand ANLAUF
- Betriebszustand RUN
- Betriebszustand HALT

In den Betriebszuständen ANLAUF und RUN können bestimmte Ereignisse auftreten, auf die das Systemprogramm reagieren muss. In vielen Fällen wird dabei ein für das Ereignis vorgesehener Organisationsbaustein als Anwenderschnittstelle aufgerufen.

#### Betriebszustand STOP

- Das Anwenderprogramm wird nicht bearbeitet.
- Hat zuvor eine Programmbearbeitung stattgefunden, bleiben die Werte von Zählern, Zeiten, Merkern und des Prozessabblids beim Übergang in den STOP-Zustand erhalten.
- Die Befehlsausgabe ist gesperrt, d.h. alle digitalen Ausgaben sind gesperrt.
- RUN-LED aus
- STOP-LED an

#### Betriebszustand ANLAUF

- Während des Übergangs von STOP nach RUN erfolgt ein Sprung in den Anlauf-Organisationsbaustein OB 100. Der Ablauf des OBs wird zeitlich nicht überwacht. Im Anlauf-OB können weitere Bausteine aufgerufen werden.
- Beim Anlauf sind alle digitalen Ausgaben gesperrt, d.h. die Befehlsausgabesperre ist aktiv.
- RUN-LED blinkt, solange der OB 100 bearbeitet wird und für mindestens 3s, auch wenn der Anlauf kürzer ist oder die CPU aufgrund eines Fehler in STOP geht. Dies zeigt den Anlauf an.
- STOP-LED aus

Wenn die CPU einen Anlauf fertig bearbeitet hat, geht Sie in den Betriebszustand RUN über.

<b>Betriebszustand RUN</b>	<ul style="list-style-type: none"> <li>■ Das Anwenderprogramm im OB 1 wird zyklisch bearbeitet, wobei zusätzlich alarmgesteuert weitere Programmteile eingeschachtelt werden können.</li> <li>■ Alle im Programm gestarteten Zeiten und Zähler laufen und das Prozessabbild wird zyklisch aktualisiert.</li> <li>■ Das BASP-Signal (Befehlsausgabesperre) wird deaktiviert, d.h. alle Ausgänge sind freigegeben.</li> <li>■ RUN-LED an</li> <li>■ STOP-LED aus</li> </ul>
<b>Betriebszustand HALT</b>	Die CPU bietet Ihnen die Möglichkeit bis zu 3 Haltepunkte zur Programmdiagnose einzusetzen. Das Setzen und Löschen von Haltepunkten erfolgt in Ihrer Programmierumgebung. Sobald ein Haltepunkt erreicht ist, können Sie schrittweise Ihre Befehlszeilen abarbeiten.
<b>Voraussetzung</b>	<p>Für die Verwendung von Haltepunkten müssen folgende Voraussetzungen erfüllt sein:</p> <ul style="list-style-type: none"> <li>■ Das Testen im Einzelschrittmodus ist in AWL möglich, ggf. über "Ansicht → AWL" Ansicht in AWL ändern</li> <li>■ Der Baustein muss online geöffnet und darf nicht geschützt sein.</li> </ul>
<b>Vorgehensweise zur Arbeit mit Haltepunkten</b>	<ol style="list-style-type: none"> <li>1. ► Blenden Sie über "Ansicht → Haltepunkte" diese ein.</li> <li>2. ► Setzen Sie Ihren Cursor auf die Anweisungszeile, in der ein Haltepunkt gesetzt werden soll.</li> <li>3. ► Setzen Sie den Haltepunkt mit "Test → Haltepunkt setzen". ⇒ Die Anweisungszeile wird mit einem Kreisring markiert.</li> <li>4. ► Zur Aktivierung des Haltepunkts gehen Sie auf "Test → Haltepunkt" aktiv. ⇒ Der Kreisring wird zu einer Kreisfläche.</li> <li>5. ► Bringen Sie Ihre CPU in RUN. ⇒ Wenn Ihr Programm auf den Haltepunkt trifft, geht Ihre CPU in den Zustand HALT über, der Haltepunkt wird mit einem Pfeil markiert und die Registerinhalte werden eingeblendet.</li> <li>6. ► Nun können Sie mit "Test → Nächste Anweisung ausführen" schrittweise Ihren Programmcode durchfahren oder über "Test → Fortsetzen" Ihre Programmausführung bis zum nächsten Haltepunkt fortsetzen.</li> <li>7. ► Mit "Test → (Alle) Haltepunkte löschen" können Sie (alle) Haltepunkte wieder löschen.</li> </ol>
<b>Verhalten im Betriebszustand HALT</b>	<ul style="list-style-type: none"> <li>■ RUN-LED blinkt und die STOP-LED leuchtet.</li> <li>■ Die Bearbeitung des Codes ist angehalten. Alle Ablafebeneen werden nicht weiterbearbeitet.</li> <li>■ Alle Zeiten werden eingefroren.</li> <li>■ Echtzeituhr läuft weiter.</li> <li>■ Ausgänge werden abgeschaltet (BASP ist aktiv).</li> <li>■ Projektierte CP-Verbindungen bleiben bestehen.</li> </ul>



*Der Einsatz von Haltepunkten ist immer möglich. Eine Umschaltung in die Betriebsart Testbetrieb ist nicht erforderlich.*

*Sobald Sie mehr als 2 Haltepunkte gesetzt haben, ist eine Einzelschrittbearbeitung nicht mehr möglich.*

## 5.12.2 Funktionssicherheit

Die CPUs besitzen Sicherheitsmechanismen, wie einen Watchdog (100ms) und eine parametrierbare Zykluszeitüberwachung (parametrierbar min. 1ms), die im Fehlerfall die CPU stoppen bzw. einen RESET auf der CPU durchführen und diese in einen definierten STOP-Zustand versetzen. Die CPUs von VIPA sind funktionssicher ausgelegt und besitzen folgende Systemeigenschaften:

Ereignis	betrifft	Effekt
RUN → STOP	allgemein	BASP (Befehls-Ausgabe-Sperre) wird gesetzt.
	zentrale digitale Ausgänge	Die Ausgänge werden abgeschaltet.
	zentrale analoge Ausgänge	Die Ausgänge werden abgeschaltet. <ul style="list-style-type: none"> <li>■ Spannungsausgänge geben 0V aus</li> <li>■ Stromausgänge 0...20mA geben 0mA aus</li> <li>■ Stromausgänge 4...20mA geben 4mA aus</li> </ul> Falls parametriert können auch Ersatzwerte ausgegeben werden.
	dezentrale Ausgänge	Verhalten wie bei zentralen digitalen/analogen Ausgängen
	dezentrale Eingänge	Die Eingänge werden von der dezentralen Station zyklisch gelesen und die aktuellen Werte zur Verfügung gestellt.
STOP → RUN bzw. NetzEin	allgemein	Zuerst wird das PAE gelöscht, danach erfolgt der Aufruf des OB 100. Nachdem dieser abgearbeitet ist, wird das BASP zurückgesetzt und der Zyklus gestartet mit: PAA löschen → PAE lesen → OB 1.
	dezentrale Eingänge	Die Eingänge werden von der dezentralen Station einmalig gelesen und die aktuellen Werte zur Verfügung gestellt.
RUN	allgemein	Der Programmablauf ist zyklisch und damit vorhersehbar: PAE lesen → OB 1 → PAA schreiben.

PAE: Prozessabbild der Eingänge, PAA: Prozessabbild der Ausgänge

## 5.13 Urlöschen

### Übersicht

Beim Urlöschen wird der komplette Anwenderspeicher gelöscht. Ihre Daten auf der Memory Card bleiben erhalten. Bei urlöschter CPU übernimmt der EtherCAT-Master eine Default-Konfiguration. Das EtherCAT-System befindet sich dann im Zustand PreOp.

Sie haben 2 Möglichkeiten zum Urlöschen:

- Urlöschen über Betriebsartenschalter
- Urlöschen über Konfigurations-Software wie z.B. Siemens SIMATIC Manager



*Vor dem Laden Ihres Anwenderprogramms in Ihre CPU sollten Sie die CPU immer urlöschen, um sicherzustellen, dass sich kein alter Baustein mehr in Ihrer CPU befindet.*

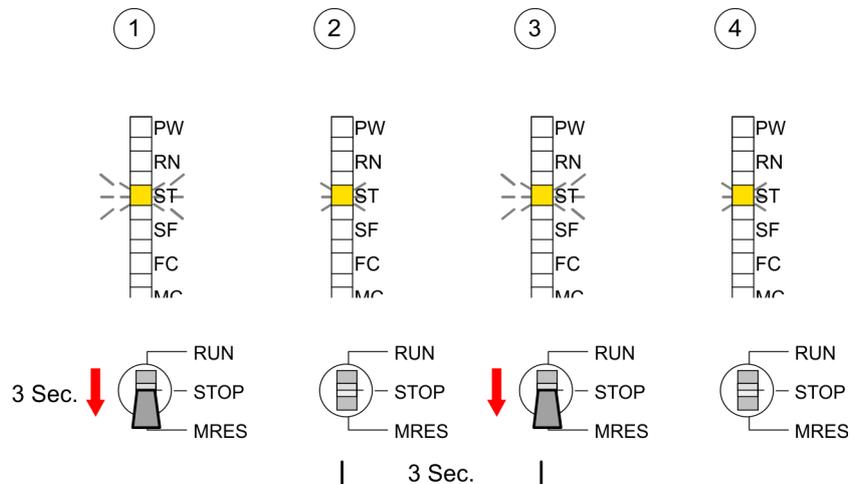
### Urlöschen über Betriebsartenschalter

#### Voraussetzung

- ➔ Ihre CPU muss sich im STOP-Zustand befinden. Stellen Sie hierzu den CPU-Betriebsartenschalter auf "STOP".
- ⇒ Die STOP-LED leuchtet.

#### Urlöschen

- 1.** ➔ Bringen Sie den Betriebsartenschalter in Stellung MRES und halten Sie ihn ca. 3 Sekunden.
  - ⇒ Die STOP-LED geht von Blinken über in Dauerlicht.
- 2.** ➔ Bringen Sie den Betriebsartenschalter in Stellung STOP und innerhalb von 3 Sekunden kurz in MRES dann wieder auf STOP.
  - ⇒ Die STOP-LED blinkt (Urlösch-Vorgang).
- 3.** ➔ Das Urlöschen ist abgeschlossen, wenn die STOP-LED in Dauerlicht übergeht.
  - ⇒ Die STOP-LED leuchtet. Die nachfolgende Abbildung zeigt nochmals die Vorgehensweise:



## Firmwareupdate

**Urlöschen über Siemens SIMATIC Manager**

- Voraussetzung: Ihre CPU muss sich im STOP-Zustand befinden. Mit dem Menübefehl "Zielsystem → Betriebszustand" bringen Sie Ihre CPU in STOP.
- Urlöschen: Über den Menübefehl "Zielsystem → Urlöschen" fordern Sie das Urlöschen an. In dem Dialogfenster können Sie, wenn noch nicht geschehen, Ihre CPU in STOP bringen und das Urlöschen starten. Während des Urlöschvorgangs blinkt die STOP-LED. Geht die STOP-LED in Dauerlicht über, ist der Urlöschvorgang abgeschlossen.

**Automatisch nachladen**

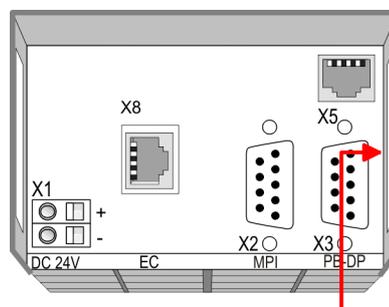
Falls nach dem Urlöschen auf der MMC ein Projekt S7PROG.WLD vorhanden ist, versucht die CPU dieses von der MMC neu zu laden. → Die MC-LED leuchtet. Nach dem Nachladen erlischt die LED. Abhängig von der Einstellung des Betriebsartenschalters bleibt die CPU in STOP bzw. geht in RUN.

**Rücksetzen auf Werkseinstellung**

Das *Rücksetzen auf Werkseinstellung* löscht das interne RAM der CPU vollständig und bringt diese zurück in den Auslieferungszustand. Bitte beachten Sie, dass hierbei auch die MPI-Adresse defaultmäßig auf 2 zurückgestellt wird! ↪ Kapitel 5.15 "Rücksetzen auf Werkseinstellung" auf Seite 69

**5.14 Firmwareupdate****Übersicht**

- Sie haben die Möglichkeit unter Einsatz einer MMC für die CPU und ihre Komponenten ein Firmwareupdate durchzuführen. Hierzu muss sich in der CPU beim Hochlauf eine entsprechend vorbereitete MMC befinden.
- Damit eine Firmwaredatei beim Hochlauf erkannt und zugeordnet werden kann, ist für jede update-fähige Komponente und jeden Hardware-Ausgabestand ein pkg-Dateiname reserviert, der mit "px" beginnt und sich in einer 6-stelligen Ziffer unterscheidet. Bei jedem updatefähigen Modul finden Sie den pkg-Dateinamen unter der Frontklappe auf einem Aufkleber auf der rechten Seite des Moduls.
- Nach NetzEIN und CPU-STOP prüft die CPU, ob eine \*.pkg-Datei auf der MMC vorhanden ist. Wenn sich diese Firmware-Version von der zu überschreibenden Firmware-Version unterscheidet, zeigt die CPU dies über LED-Blinken an und sie können die Firmware über eine Updateanforderung installieren.



**Firmware package and Version**

**Aktuelle Firmware auf  
www.vipa.com**

Die aktuellsten Firmwarestände finden Sie auf [www.vipa.com](http://www.vipa.com) im Service-Bereich. Beispielsweise sind für den Firmwareupdate der CPU 315-4EC12 und Ihrer Komponenten für den Ausgabestand 1 folgende Dateien erforderlich:

- 315-4EC12, Ausgabestand 1: Px000176.pkg
- PROFIBUS-DP Master: Px000062.pkg
- EtherCAT-Master: Px000153.pkg

**VORSICHT!**

Beim Aufspielen einer neuen Firmware ist äußerste Vorsicht geboten. Unter Umständen kann Ihre CPU unbrauchbar werden, wenn beispielsweise während der Übertragung die Spannungsversorgung unterbrochen wird oder die Firmware-Datei fehlerhaft ist. Setzen Sie sich in diesem Fall mit der VIPA-Hotline in Verbindung!

Bitte beachten Sie auch, dass sich die zu überschreibende Firmware-Version von der Update-Version unterscheidet, ansonsten erfolgt kein Update.

**Firmwarestand des  
SPEED7-Systems über  
Web-Seite ausgeben**

Die CPU hat eine Web-Seite integriert, die auch Informationen zum Firmwarestand der SPEED7-Komponenten bereitstellt. Über den Ethernet-PG/OP-Kanal haben Sie Zugriff auf diese Web-Seite. Zur Aktivierung des PG/OP-Kanals müssen Sie diesem IP-Parameter zuweisen. Dies kann im Siemens SIMATIC Manager entweder über eine Hardware-Konfiguration erfolgen, die Sie über MMC bzw. MPI einspielen oder über Ethernet durch Angabe der MAC-Adresse unter "*Zielsystem → Ethernet-Adresse vergeben*". Danach können Sie mit einem Web-Browser über die angegebene IP-Adresse auf den PG/OP-Kanal zugreifen.

↳ Kapitel 5.11 "Zugriff auf integrierte Web-Seite" auf Seite 60

**Firmware laden und auf  
MMC übertragen**

- Gehen Sie auf [www.vipa.com](http://www.vipa.com)
- Klicken Sie auf "*Service → Download → Firmware*".
- Navigieren Sie über "*System 300S → CPU*" zu Ihrer CPU und laden Sie die zip-Datei auf Ihren PC.
- Entpacken Sie die zip-Datei und kopieren Sie die extrahierten pkg-Dateien auf Ihre MMC.

**VORSICHT!**

Beim Firmwareupdate wird automatisch ein Urlöschen durchgeführt. Sollte sich Ihr Programm nur im Ladespeicher der CPU befinden, so wird es hierbei gelöscht! Sichern Sie Ihr Programm, bevor Sie ein Firmwareupdate durchführen! Auch sollten Sie nach dem Firmwareupdate ein "Rücksetzen auf Werkseinstellung" durchführen.

↳ Kapitel 5.15 "Rücksetzen auf Werkseinstellung" auf Seite 69

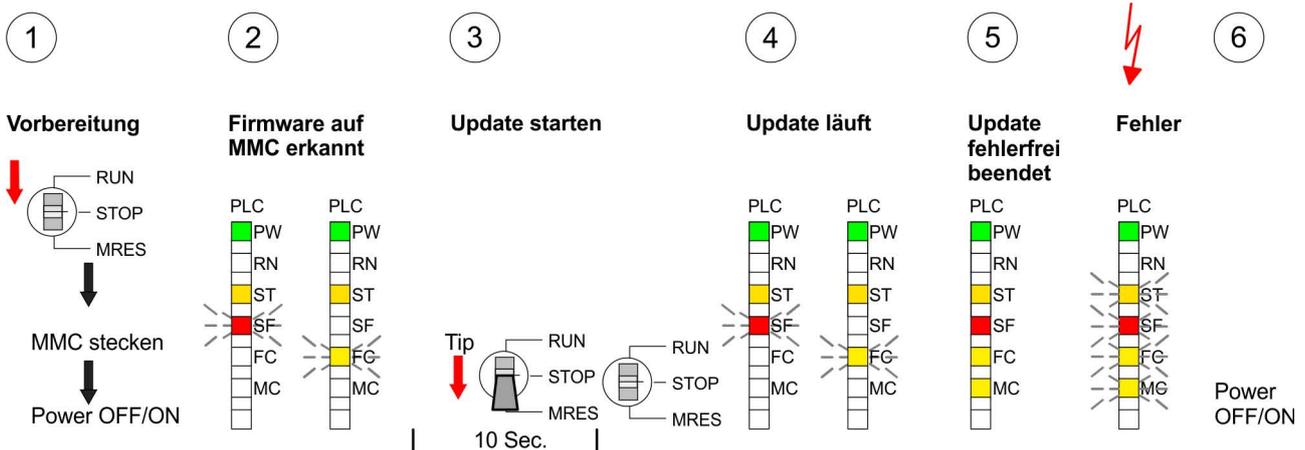
## Firmware von MMC in CPU übertragen

1. ▶ Bringen Sie den Betriebsartenschalter Ihrer CPU in Stellung STOP. Schalten Sie die Spannungsversorgung aus. Stecken Sie die MMC mit den Firmware-Dateien in die CPU. Achten Sie hierbei auf die Steckrichtung der MMC. Schalten Sie die Spannungsversorgung ein.
2. ▶ Nach einer kurzen Hochlaufzeit zeigt das abwechselnde Blinken der LEDs SF und FC an, dass auf der MMC mindestens eine aktuellere Firmware-Datei gefunden wurde.
3. ▶ Sie starten die Übertragung der Firmware, sobald Sie innerhalb von 10s den Betriebsartenschalter kurz nach MRES tippen und dann den Schalter in der STOP-Position belassen.
4. ▶ Während des Update-Vorgangs blinken die LEDs SF und FC abwechselnd und die MC-LED leuchtet. Dieser Vorgang kann mehrere Minuten dauern.
5. ▶ Das Update ist fehlerfrei beendet, wenn die LEDs PW, ST, SF, FC und MC leuchten. Blinken diese schnell, ist ein Fehler aufgetreten.



*Erst dann, wenn die LEDs PW, ST, SF, FC und MC an der CPU leuchten, dürfen Sie einen Power-Cycle an der CPU durchführen!*

6. ▶ Schalten Sie die Spannungsversorgung aus und wieder ein. Jetzt prüft die CPU, ob noch weitere Firmware-Updates durchzuführen sind. Ist dies der Fall, blinken, wiederum nach einer kurzen Hochlaufzeit, die LEDs SF und FC. Fahren Sie mit Punkt 3 fort.
  - ⇒ Blinken die LEDs nicht, ist das Firmware-Update abgeschlossen. Führen Sie jetzt wie nachfolgend beschrieben ein *Rücksetzen auf Werkseinstellungen* durch. Danach ist die CPU wieder einsatzbereit.



### 5.15 Rücksetzen auf Werkseinstellung

#### Vorgehensweise

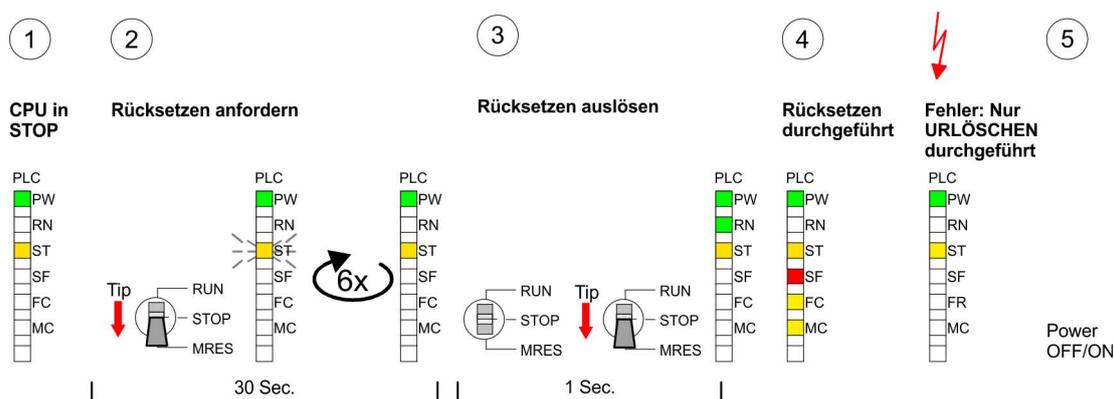
Die folgende Vorgehensweise löscht das interne RAM der CPU vollständig und bringt diese zurück in den Auslieferungszustand.

Bitte beachten Sie, dass hierbei auch die MPI-Adresse auf 2 und die IP-Adresse des Ethernet-PG/OP-Kanals auf 0.0.0.0 zurückgestellt wird!

Sie können auch das Rücksetzen auf Werkseinstellung mit dem MMC-Cmd FACTORY\_RESET ausführen. *↳ Kapitel 5.19 "MMC-Cmd - Autobefehle" auf Seite 73*

1. ▶ Bringen Sie die CPU in STOP.
2. ▶ Drücken Sie den Betriebsartenschalter für ca. 30 Sekunden nach unten in Stellung MRES. Hierbei blinkt die STOP-LED. Nach ein paar Sekunden leuchtet die STOP-LED. Die STOP-LED wechselt jetzt von Leuchten in Blinken. Zählen Sie, wie oft die STOP-LED leuchtet.
3. ▶ Nach dem 6. Mal Leuchten der STOP-LED lassen Sie den Reset-Schalter wieder los, um ihn nochmals kurzzeitig nach unten zu drücken. Jetzt leuchtet die grüne RUN-LED einmal auf. Das bedeutet, dass das RAM vollständig gelöscht ist.
4. ▶ Zur Bestätigung des Rücksetzvorgangs leuchten die LEDs PW, ST, SF, FC und MC. Leuchtet diese nicht, wurde nur URLöschen ausgeführt und das Rücksetzen auf Werkseinstellung ist fehlgeschlagen. In diesem Fall können Sie den Vorgang wiederholen. Das Rücksetzen auf Werkseinstellung wird nur dann ausgeführt, wenn die STOP-LED genau 6 Mal geleuchtet hat.
5. ▶ Am Ende des Rücksetzvorgangs leuchten die LEDs PW, ST, SF, FC und MC. Danach ist die Spannungsversorgung aus- und wieder einzuschalten.

Die nachfolgende Abbildung soll die Vorgehensweise verdeutlichen:



Bitte führen Sie nach einem Firmwareupdate der CPU immer ein Rücksetzen auf Werkseinstellung durch.

## 5.16 Steckplatz für Speichermedien

### Übersicht

Auf der Frontseite der CPU befindet sich ein Steckplatz für Speichermedien. Über diesen Steckplatz können Sie eine Multimedia Card (MMC) als externes Speichermedium für Programme und Firmware stecken. Mittels vorgegebener Dateinamen können Sie die CPU veranlassen automatisch ein Projekt zu laden bzw. eine Kommandodatei auszuführen.

### Zugriff auf das Speichermedium

Zu folgenden Zeitpunkten erfolgt ein Zugriff auf ein Speichermedium:

- Nach Urlöschen
  - Die CPU prüft, ob ein Projekt mit dem Namen S7PROG.WLD vorhanden ist. Wenn ja, wird dieses automatisch geladen.
  - Die CPU prüft, ob ein Projekt mit dem Namen PROTECT.WLD mit geschützten Bausteinen vorhanden ist. Wenn ja, wird dieses automatisch geladen. Diese Bausteine verbleiben in der CPU bis zum Rücksetzen der CPU auf Werkseinstellungen oder Laden einer "leeren" PROTECT.WLD.
  - Die CPU prüft, ob eine MCC-Speichererweiterung gesteckt ist. Falls ja, erfolgt die Freigabe der Speichererweiterung, ansonsten wird eine zuvor aktivierte Speichererweiterung deaktiviert.
- Nach NetzEIN
  - Die CPU prüft, ob ein Projekt mit dem Namen AUTO-LOAD.WLD vorhanden ist. Wenn ja, wird Urlöschen durchgeführt und das Projekt automatisch geladen.
  - Die CPU prüft, ob eine Kommandodatei mit dem Namen VIPA\_CMD.MMC vorhanden ist. Wenn ja, wird die Kommandodatei geladen und die enthaltenen Befehle werden ausgeführt.
  - Nach NetzEIN und CPU-STOP prüft die CPU, ob eine \*.pkg-Datei (Firmware-Datei) vorhanden ist. Wenn ja, zeigt die CPU dies über LED-Blinken an und sie können die Firmware über eine Updateanforderung installieren.
- Einmalig im Zustand STOP
  - Wird eine Speicherkarte mit einer Kommandodatei mit dem Namen VIPA\_CMD.MMC gesteckt, so wird die Kommandodatei geladen und die enthaltenen Befehle werden ausgeführt.

## 5.17 Speichererweiterung mit MCC

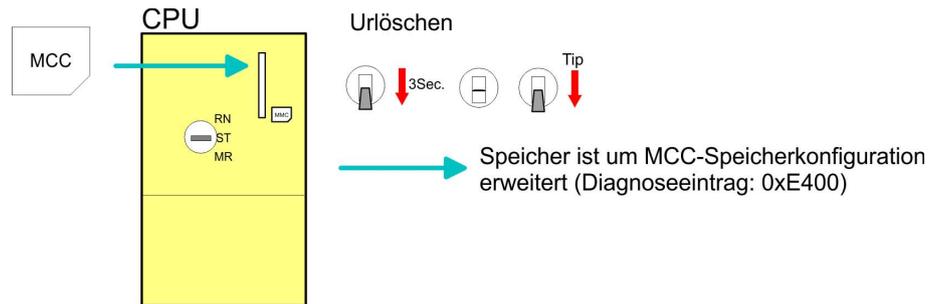
### Übersicht



Bei der CPU haben Sie die Möglichkeit den Arbeitsspeicher zu erweitern. Hierzu ist bei VIPA eine MCC Speichererweiterungskarte verfügbar. Bei der MCC handelt es sich um eine speziell vorbereitete MMC (Multimedia Card). Durch Stecken der MCC im MCC-Slot und anschließendem Urlöschen wird die entsprechende Speichererweiterung freigeschaltet. Es kann immer nur eine Speichererweiterung aktiviert sein. Auf der MCC befindet sich die Datei memory.key. Diese Datei darf weder bearbeitet noch gelöscht werden. Sie können die MCC auch als "normale" MMC zur Speicherung Ihrer Projekte verwenden.

### Vorgehensweise

Zur Erweiterung des Speichers stecken Sie die MCC in den mit "MCC" bezeichneten Kartenslot der CPU und führen Sie Urlöschen durch.



Sollte die Speichererweiterung auf der MCC den maximal erweiterbaren Speicherbereich der CPU überschreiten, wird automatisch der maximal mögliche Speicher der CPU verwendet. Den Speicherausbau können Sie über die integrierte Web-Seite oder mit dem Siemens SIMATIC Manager über den Baugruppenzustand unter "Speicher" ermitteln.



#### VORSICHT!

Bitte beachten Sie, dass, sobald Sie eine Speichererweiterung auf Ihrer CPU durchgeführt haben, die MCC gesteckt bleiben muss. Ansonsten geht die CPU nach 72 Stunden in STOP. Auch kann die MCC nicht gegen eine MCC mit gleicher Speicherkonfiguration getauscht werden.

#### Verhalten

Wurde die MCC-Speicherkonfiguration übernommen, finden Sie den Diagnoseeintrag 0xE400 im Diagnosepuffer der CPU.

Nach Ziehen der MCC erfolgt der Eintrag 0xE401 im Diagnosepuffer, die SF-LED leuchtet und nach 72 Stunden geht die CPU in STOP. Hier ist ein Anlauf erst wieder möglich nach Stecken der MCC oder nach Urlöschen.

Die verbleibende Zeit nach dem Ziehen der MCC können Sie jederzeit über den Parameter *MCC-Trial-Time* auf der Web-Seite ermitteln.

Nach erneutem Stecken der MCC erlischt die SF-LED und 0xE400 wird im Diagnosepuffer eingetragen. Sie können jederzeit die Speicherkonfiguration Ihrer CPU auf den ursprünglichen Zustand wieder zurücksetzen, indem Sie Urlöschen ohne MCC ausführen.

### 5.18 Erweiterter Know-how-Schutz

#### Übersicht

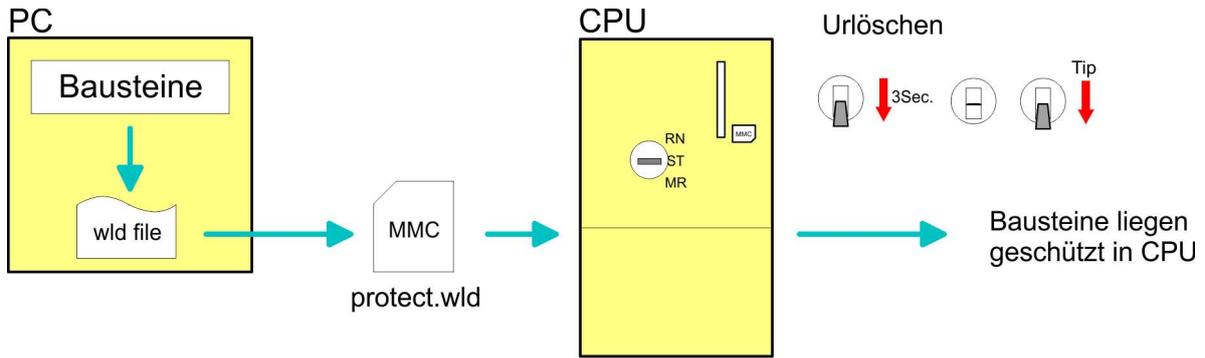
Neben dem "Standard" Know-how-Schutz besitzen die SPEED7-CPU von VIPA einen "erweiterten" Know-how-Schutz, der einen sicheren Baustein-Schutz vor Zugriff Dritter bietet.

#### Standard-Schutz

Beim Standard-Schutz von Siemens werden auch geschützte Bausteine in das PG übertragen, aber deren Inhalt nicht dargestellt. Durch entsprechende Manipulation ist der Know-how-Schutz aber nicht sichergestellt.

**Erweiterter Schutz**

Mit dem von VIPA entwickelten "erweiterten" Know-how-Schutz besteht aber die Möglichkeit Bausteine permanent in der CPU zu speichern. Beim "erweiterten" Schutz übertragen Sie die zu schützenden Bausteine in eine WLD-Datei mit Namen protect.wld. Durch Stecken der MMC und anschließendem Urlöschen werden die in protect.wld gespeicherten Bausteine permanent in der CPU abgelegt. Geschützt werden können OBs, FBs und FCs. Beim Zurücklesen von geschützten Bausteinen in Ihr PG werden ausschließlich die Baustein-Header geladen. Der schützenswerte Baustein-Code bleibt in der CPU und kann nicht ausgelesen werden.

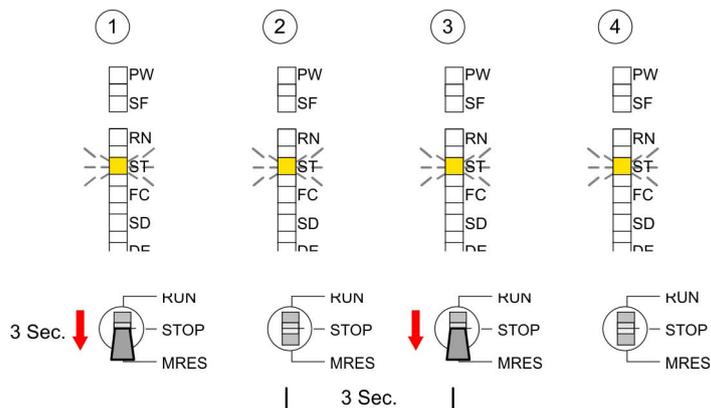


**Bausteine mit protect.wld schützen**

Erzeugen Sie in Ihrem Projekttool mit "Datei → Memory Card Datei → Neu" eine WLD-Datei und benennen Sie diese um in "protect.wld". Übertragen Sie die zu schützenden Bausteine in die Datei, indem Sie diese mit der Maus aus Ihrem Projekt in das Dateifenster von protect.wld ziehen.

**protect.wld mit Urlöschen in CPU übertragen**

Übertragen Sie die Datei protect.wld auf eine MMC-Speicherkarte, stecken Sie die MMC in Ihre CPU und führen Sie nach folgender Vorgehensweise Urlöschen durch:



Mit Urlöschen werden die in protect.wld enthaltenen Bausteine, permanent vor Zugriffen Dritter geschützt, in der CPU abgelegt.

**Schutzverhalten**

Geschützte Bausteine werden durch eine neue protect.wld überschrieben. Mit einem PG können Dritte auf geschützte Bausteine zugreifen, hierbei wird aber ausschließlich der Baustein-Header in das PG übertragen. Der schützenswerte Baustein-Code bleibt in der CPU und kann nicht ausgelesen werden.

**Geschützte Bausteine überschreiben bzw. löschen**

Sie haben jederzeit die Möglichkeit geschützte Bausteine durch gleichnamige Bausteine im RAM der CPU zu überschreiben. Diese Änderung bleibt bis zum nächsten URLöschen erhalten. Geschützte Bausteine können nur dann vom PG dauerhaft überschrieben werden, wenn diese zuvor aus der protect.wld gelöscht wurden. Durch Übertragen einer leeren protect.wld von der MMC können Sie in der CPU alle geschützten Bausteine löschen.

**Einsatz von geschützten Bausteinen**

Da beim Auslesen eines "protected" Bausteins aus der CPU die Symbol-Bezeichnungen fehlen, ist es ratsam dem Endanwender die "Bausteinhüllen" zur Verfügung zu stellen. Erstellen Sie hierzu aus allen geschützten Bausteinen ein Projekt. Löschen Sie aus diesen Bausteinen alle Netzwerke, so dass diese ausschließlich die Variablen-Definitionen in der entsprechenden Symbolik beinhalten.

**5.19 MMC-Cmd - Autobefehle**

**Übersicht**

Eine *Kommando*-Datei auf einer MMC wird unter folgenden Bedingungen automatisch ausgeführt:

- CPU befindet sich in STOP und MMC wird gesteckt
- Bei jedem Einschaltvorgang (NetzeIN)

**Kommando-Datei**

Bei der *Kommando*-Datei handelt es sich um eine Text-Datei mit einer Befehlsabfolge, die unter dem Namen **vipa\_cmd.mmc** im Root-Verzeichnis der MMC abzulegen ist. Die Datei muss mit dem 1. Befehl `CMD_START` beginnen, gefolgt von den gewünschten Befehlen (kein anderer Text) und ist immer mit dem letzten Befehl `CMD_END` abzuschließen.

Texte wie beispielsweise Kommentare nach dem letzten Befehl `CMD_END` sind zulässig, da diese ignoriert werden. Sobald eine Kommandodatei erkannt und ausgeführt wird, werden die Aktionen in der Datei `Logfile.txt` auf der MMC gespeichert. Zusätzlich finden Sie für jeden ausgeführten Befehl einen Diagnoseeintrag im Diagnosepuffer.

**Befehle**

Bitte beachten Sie, dass Sie immer Ihre Befehlsabfolge mit `CMD_START` beginnen und mit `CMD_END` beenden.

Kommando	Beschreibung	Diagnoseeintrag
CMD_START	In der ersten Zeile muss CMD_START stehen.	0xE801
	Fehlt CMD_START erfolgt ein Diagnoseeintrag	0xE8FE
WAIT1SECOND	Wartet ca. 1 Sekunde.	0xE803
WEBPAGE	Speichert die Web-Seite der CPU als Datei "webpage.htm" auf der MMC.	0xE804
LOAD_PROJECT	Ruft die Funktion "Urlöschen mit Nachladen von der MMC" auf. Durch Angabe einer wld-Datei nach dem Kommando, wird diese wld-Datei nachgeladen, ansonsten wird die Datei "s7prog.wld" geladen.	0xE805

Kommando	Beschreibung	Diagnoseeintrag
SAVE_PROJECT	Speichert das Anwenderprojekt (Bausteine und Hardware-Konfiguration) auf der MMC als "s7prog.wld". Falls bereits eine Datei mit dem Namen "s7prog.wld" existiert, wird diese in "s7prog.old" umbenannt. Sollte Ihre CPU durch ein Passwort geschützt sein, so müssen Sie dies als Parameter mitliefern. Ansonsten wird kein Projekt geschrieben. Beispiel: SAVE_PROJECT password	0xE806
FACTORY_RESET	Führt "Rücksetzen auf Werkseinstellung" durch.	0xE807
DIAGBUF	Speichert den Diagnosepuffer der CPU als Datei "diagbuff.txt" auf der MMC.	0xE80B
SET_NETWORK	Mit diesem Kommando können Sie die IP-Parameter für den Ethernet-PG/OP-Kanal einstellen. Die IP-Parameter sind in der Reihenfolge IP-Adresse, Subnetz-Maske und Gateway jeweils getrennt durch ein Komma im Format von x.x.x.x einzugeben. Wird kein Gateway verwendet, tragen Sie die IP-Adresse als Gateway ein.	0xE80E
CMD_END	In der letzten Zeile muss CMD_END stehen.	0xE802

## Beispiele

Nachfolgend ist der Aufbau einer Kommando-Datei an Beispielen gezeigt. Den jeweiligen Diagnoseeintrag finden Sie in Klammern gesetzt.

### Beispiel 1

CMD_START	Kennzeichnet den Start der Befehlsliste (0xE801)
LOAD_PROJECT proj.wld	Urlöschen und Nachladen von "proj.wld" (0xE805)
WAIT1SECOND	Wartet ca. 1 Sekunde (0xE803)
WEBPAGE	Web-Seite als "webpage.htm" speichern (0xE804)
DIAGBUF	Diagnosepuffer der CPU als "diagbuff.txt" speichern (0xE80B)
CMD_END	Kennzeichnet das Ende der Befehlsliste (0xE802)
... beliebiger Text ...	Texte nach dem CMD_END werden nicht mehr ausgewertet.

### Beispiel 2

CMD_START	Kennzeichnet den Start der Befehlsliste (0xE801)
LOAD_PROJECT proj2.wld	Urlöschen und Nachladen von "proj2.wld" (0xE805)
WAIT1SECOND	Wartet ca. 1 Sekunde (0xE803)
WAIT1SECOND	Wartet ca. 1 Sekunde (0xE803)
	IP-Parameter (0xE80E)
SET_NETWORK 172.16.129.210,255.255.224.0,172.16.129.210	
WAIT1SECOND	Wartet ca. 1 Sekunde (0xE803)
WAIT1SECOND	Wartet ca. 1 Sekunde (0xE803)

WEBPAGE	Web-Seite als "webpage.htm" speichern (0xE804)
DIAGBUF	Diagnosepuffer der CPU als "diagbuff.txt" speichern (0xE80B)
CMD_END	Kennzeichnet das Ende der Befehlsliste (0xE802)
... beliebiger Text ...	Texte nach dem CMD_END werden nicht mehr ausgewertet.



Die Parameter IP-Adresse, Subnetz-Maske und Gateway erhalten Sie von Ihrem Systemadministrator.

Wird kein Gateway verwendet, tragen Sie die IP-Adresse als Gateway ein.

## 5.20 VIPA-spezifische Diagnose-Einträge

### Einträge im Diagnosepuffer

Sie haben die Möglichkeit im Siemens SIMATIC Manager den Diagnosepuffer der CPU auszulesen. Neben den Standardeinträgen im Diagnosepuffer gibt es in den CPUs der VIPA noch zusätzliche Einträge, welche ausschließlich in Form einer Ereignis-ID angezeigt werden.

Mit dem CMD DIAGBUF wird der aktuelle Inhalt des Diagnosepuffers auf die Speicherkarte gespeichert.



Die CPUs von VIPA unterstützen alle Register des Baugruppenzustands. Eine nähere Beschreibung der einzelnen Register finden Sie in der Online-Hilfe Ihres Siemens SIMATIC Managers.

### Anzeige der Diagnose-einträge

Zur Anzeige der Diagnoseeinträge gehen Sie in Ihrem Siemens SIMATIC Manager auf "Zielsystem → Baugruppenzustand". Über das Register "Diagnosepuffer" gelangen Sie in das Diagnosefenster:

The screenshot shows the 'Baugruppenzustand' window in SIMATIC Manager. The 'Diagnosepuffer' register is selected, displaying a table of diagnostic events. The entry 'Ereignis-ID: 16# E0CC' is highlighted with a red arrow and labeled 'VIPA-ID'.

Nr.	Uhrzeit	Datum	Ereignis
8	...	...	...
9	...	...	...
10	13:18:11:370	19.12.2011	Ereignis-ID: 16# E0CC
11	...	...	...
12	...	...	...
13	...	...	...

Für die Diagnose ist der Betriebszustand der CPU irrelevant. Es können maximal 100 Diagnoseeinträge in der CPU gespeichert werden.

## Übersicht der Ereignis-IDs

Ereignis-ID	Bedeutung
0x115C	Herstellerspezifischer Alarm (OB 57) bei EtherCAT OB: OB-Nummer (57) ZInfo1: Logische Adresse des Slaves, der den Alarm ausgelöst hat ZInfo2: Alarmtyp ZInfo3: Reserviert
0xE003	Fehler beim Zugriff auf Peripherie Zinfo1: Peripherie-Adresse Zinfo2: Steckplatz
0xE004	Mehrfach-Parametrierung einer Peripherieadresse Zinfo1: Peripherie-Adresse Zinfo2: Steckplatz
0xE005	Interner Fehler - Kontaktieren Sie bitte die VIPA-Hotline!
0xE006	Interner Fehler - Kontaktieren Sie bitte die VIPA-Hotline!
0xE007	Konfigurierte Ein-/Ausgangsbytes passen nicht in Peripheriebereich
0xE008	Interner Fehler - Kontaktieren Sie bitte die VIPA-Hotline!
0xE009	Fehler beim Zugriff auf Standard-Rückwandbus
0xE010	Nicht definierte Baugruppe am Rückwandbus erkannt Zinfo2: Steckplatz Zinfo3: Typkennung
0xE011	Masterprojektierung auf Slave-CPU nicht möglich oder fehlerhafte Slave-Konfiguration
0xE012	Fehler bei Parametrierung
0xE013	Fehler bei Schieberegisterzugriff auf Standardbus-Digitalmodule
0xE014	Fehler bei Check_Sys
0xE015	Fehler beim Zugriff auf Master Zinfo2: Steckplatz des Masters (32=Kachelmaster)
0xE016	Maximale Blockgröße bei Mastertransfer überschritten Zinfo1: Peripherie-Adresse Zinfo2: Steckplatz
0xE017	Fehler beim Zugriff auf integrierten Slave
0xE018	Fehler beim Mappen der Master-Peripherie
0xE019	Fehler bei Erkennung des Standard Rückwandbus Systems
0xE01A	Fehler bei Erkennung der Betriebsart (8 / 9 Bit)

Ereignis-ID	Bedeutung
0xE01B	Fehler - Maximale Anzahl steckbarer Baugruppen überschritten
0xE020	Fehler - Alarminformationen undefiniert
0xE030	Fehler vom Standard-Bus
0xE033	Interner Fehler - Kontaktieren Sie bitte die VIPA-Hotline!
0xE0B0	SPEED7 kann nicht mehr gestoppt werden (Evtl. undefinierter BCD-Wert bei Timer)
0xE0C0	Nicht genug Speicherplatz im Arbeitsspeicher für Codebaustein (Baustein zu groß)
0xE0CB	Fehler bei SZL-Zugriff Zinfo1: 4=SZL falsch, 5=SubSZL falsch, 6=Index falsch Zinfo2: SZL-ID Zinfo3: Index
0xE0CC	Kommunikationsfehler MPI / Seriell Zinfo1: Code 1: Falsche Priorität 2: Pufferüberlauf 3: Telegrammformatfehler 4: Falsche SZL-Anforderung (SZL-ID ungültig) 5: Falsche SZL-Anforderung (SZL-SubID ungültig) 6: Falsche SZL-Anforderung (SZL-Index ungültig) 7: Falsche Wert 8: Falscher RetVal 9: Falscher SAP 10: Falscher Verbindungstyp 11: Falsche Sequenznummer 12: Fehlerhafte Bausteinnummer im Telegramm 13: Fehlerhafter Bausteintyp im Telegramm 14: Inaktive Funktion 15: Fehlerhafte Größe im Telegramm 20: Fehler beim Schreiben auf die Speicherkarte 90: Fehlerhafte Puffergröße 98: Unbekannter Fehler 99: Interner Fehler
0xE0CD	Fehler bei DP-V1 Auftragsverwaltung
0xE0CE	Fehler: Timeout beim Senden der i-Slave Diagnose
0xE0CF	Timeout beim Laden einer neuen HW-Konfiguration (Timeout-Zeit: 39 Sekunden)

## VIPA-spezifische Diagnose-Einträge

Ereignis-ID	Bedeutung
0xE100	Speicherkarten-Zugriffsfehler
0xE101	Speicherkarten-Fehler Filesystem
0xE102	Speicherkarten-Fehler FAT
0xE104	Speicherkarten-Fehler beim Speichern
0xE200	Speicherkarte schreiben beendet (Copy Ram2Rom)
0xE210	Speicherkarte Lesen beendet (Nachladen nach Urlöschen)
0xE21E	Speicherkarte Lesen: Fehler beim Nachladen (nach Urlöschen), Datei "Protect.wld" zu groß
0xE21F	Speicherkarte Lesen: Fehler beim Nachladen (nach Urlöschen), Lesefehler, Speicher voll
0xE300	Internes Flash Schreiben beendet (Copy Ram2Rom)
0xE310	Internes Flash Lesen beendet (Nachladen nach Batterieausfall)
0xE311	Internes Flash fx0000yy.wld Datei zu groß, Laden fehlerhaft
0xE400	Speicherkarte mit der Option Speichererweiterung wurde gesteckt.
0xE401	Speicherkarte mit der Option Speichererweiterung wurde gezogen.
0xE402	Die PROFIBUS-DP-Master-Funktionalität ist nicht aktiviert. Die Schnittstelle ist weiter als MPI-Schnittstelle aktiv.
0xE403	Die PROFIBUS-DP-Slave-Funktionalität ist nicht aktiviert. Die Schnittstelle ist weiter als MPI-Schnittstelle aktiv.
0xE500	Speicherverwaltung: Baustein ohne zugehörigen Eintrag in der BstListe gelöscht Zinfo2: BlockTyp Zinfo3: BlockNr
0xE604	Mehrfach-Parametrierung einer Peripherieadresse für Ethernet-PG/OP-Kanal Zinfo1: Peripherie-Adresse Zinfo3: 0: Peripherie-Adresse ist Eingang, 1: Peripherie-Adresse ist Ausgang
0xE701	Interner Fehler - Kontaktieren Sie bitte die VIPA-Hotline!
0xE703	Interner Fehler - Kontaktieren Sie bitte die VIPA-Hotline!
0xE720	Interner Fehler - Kontaktieren Sie bitte die VIPA-Hotline!
0xE721	Interner Fehler - Kontaktieren Sie bitte die VIPA-Hotline!
0xE801	CMD - Autobefehl: CMD_START erkannt und erfolgreich ausgeführt
0xE802	CMD - Autobefehl: CMD_END erkannt und erfolgreich ausgeführt

Ereignis-ID	Bedeutung
0xE803	CMD - Autobefehl: WAIT1SECOND erkannt und erfolgreich ausgeführt
0xE804	CMD - Autobefehl: WEBPAGE erkannt und erfolgreich ausgeführt
0xE805	CMD - Autobefehl: LOAD_PROJECT erkannt und erfolgreich ausgeführt
0xE806	CMD - Autobefehl: SAVE_PROJECT Zinfo3: 0x0000: SAVE_PROJECT erkannt und erfolgreich ausgeführt Zinfo3: 0x8000: Fehler beim Ausführen von SAVE_PROJECT z.B. falsches Passwort
0xE807	CMD - Autobefehl: FACTORY_RESET erkannt und erfolgreich ausgeführt
0xE80B	CMD - Autobefehl: DIAGBUF erkannt und erfolgreich ausgeführt
0xE80E	CMD - Autobefehl: SET_NETWORK erkannt und erfolgreich ausgeführt
0xE816	CMD - Autobefehl: SAVE_PROJECT: Fehler - CPU urgelöscht - es wurde keine wld-Datei erzeugt.
0xE8FB	CMD - Autobefehl: Fehler: Initialisierung des Ethernet-PG/OP-Kanals mittels SET_NETWORK fehlerhaft.
0xE8FC	CMD - Autobefehl: Fehler: In SET_NETWORK wurden nicht alle IP-Parameter angegeben.
0xE8FE	CMD - Autobefehl: Fehler: CMD_START nicht gefunden
0xE8FF	CMD - Autobefehl: Fehler: Fehler beim Lesen des CMD-Files (Speicherkarten-Fehler)
0xE901	Checksummen-Fehler
0xEA00	Interner Fehler - Kontaktieren Sie bitte die VIPA-Hotline!
0xEA01	Interner Fehler - Kontaktieren Sie bitte die VIPA-Hotline!
0xEA02	SBUS: Interner Fehler (intern gestecktes Submodul nicht erkannt) Zinfo1: Interner Steckplatz
0xEA03	SBUS: Kommunikationsfehler CPU - PROFINET-IO-Controller Zinfo1: Steckplatz Zinfo2: Status (0: OK, 1: ERROR, 2: BUSSY, 3: TIMEOUT, 4: LOCKED, 5: UNKNOWN)
0xEA04	SBUS: Mehrfach-Parametrierung einer Peripherieadresse Zinfo1: Peripherie-Adresse Zinfo2: Steckplatz Zinfo3: Datenbreite
0xEA05	Interner Fehler - Kontaktieren Sie bitte die VIPA-Hotline!
0xEA07	Interner Fehler - Kontaktieren Sie bitte die VIPA-Hotline!

Ereignis-ID	Bedeutung
0xEA08	SBUS: Parametrierte Eingangsdatenbreite ungleich der gesteckten Eingangsdatenbreite Zinfo1: Parametrierte Eingangsdatenbreite Zinfo2: Steckplatz Zinfo3: Eingangsdatenbreite der gesteckten Baugruppe
0xEA09	SBUS: Parametrierte Ausgangsdatenbreite ungleich der gesteckten Ausgangsdatenbreite Zinfo1: Parametrierte Ausgangsdatenbreite Zinfo2: Steckplatz Zinfo3: Ausgangsdatenbreite der gesteckten Baugruppe
0xEA10	SBUS: Eingangs-Peripherieadresse außerhalb des Peripheriebereiches Zinfo1: Peripherie-Adresse Zinfo2: Steckplatz Zinfo3: Datenbreite
0xEA11	SBUS: Ausgangs-Peripherieadresse außerhalb des Peripheriebereiches Zinfo1: Peripherie-Adresse Zinfo2: Steckplatz Zinfo3: Datenbreite
0xEA12	SBUS: Fehler beim Datensatz schreiben Zinfo1: Steckplatz Zinfo2: Datensatznummer Zinfo3: Datensatzlänge
0xEA14	SBUS: Mehrfach-Parametrierung einer Peripherieadresse (Diagnoseadresse) Zinfo1: Peripherie-Adresse Zinfo2: Steckplatz Zinfo3: Datenbreite
0xEA15	Interner Fehler - Kontaktieren Sie bitte die VIPA-Hotline!
0xEA18	SBUS: Fehler beim Mappen der Masterperipherie Zinfo2: Steckplatz des Masters
0xEA19	Interner Fehler - Kontaktieren Sie bitte die VIPA-Hotline!
0xEA20	Fehler - RS485-Schnittstelle ist nicht auf PROFIBUS-DP-Master eingestellt aber es ist ein PROFIBUS-DP-Master projektiert.
0xEA21	Fehler - Projektierung RS485-Schnittstelle X2/X3: PROFIBUS-DP-Master ist projektiert aber nicht vorhanden Zinfo2: Schnittstelle x
0xEA22	Fehler - RS485-Schnittstelle X2 - Wert ist außerhalb der Grenzen Zinfo: Projektierter Wert von X2
0xEA23	Fehler - RS485-Schnittstelle X3 - Wert ist außerhalb der Grenzen Zinfo: Projektierter Wert von X3

Ereignis-ID	Bedeutung
0xEA24	Fehler - Projektierung RS485-Schnittstelle X2/X3: Schnittstelle/Protokoll ist nicht vorhanden, die Defaulteinstellungen werden verwendet. Zinfo2: Projektierter Wert für X2 Zinfo3: Projektierter Wert für X3
0xEA30	Interner Fehler - Kontaktieren Sie bitte die VIPA-Hotline!
0xEA40	Interner Fehler - Kontaktieren Sie bitte die VIPA-Hotline!
0xEA41	Interner Fehler - Kontaktieren Sie bitte die VIPA-Hotline!
0xEA50	Fehler - PROFINET-Konfiguration Zinfo1: User-Slot des PROFINET-IO-Controllers Zinfo2: IO-Device-Nr. Zinfo3: IO-Device Slot
0xEA51	Fehler - Kein PROFINET-IO-Controller auf dem projektierten Slot erkannt Zinfo1: User-Slot des PROFINET-IO-Controllers Zinfo2: Erkannte Typkennung auf dem projektierten Slot
0xEA53	Fehler - PROFINET-Konfiguration - Es sind zu viele PROFINET-IO-Devices projektiert Zinfo1 : Anzahl der projektierten Devices Zinfo2 : Steckplatz Zinfo3 : Maximal mögliche Anzahl Devices
0xEA54	Fehler - PROFINET-IO-Controller meldet Mehrfachparametrierung einer Peripherieadresse Zinfo1: Peripherieadresse Zinfo2: User-Slot des PROFINET-IO-Controllers Zinfo3: Datenbreite
0xEA61 ... 0xEA63	Interner Fehler - Kontaktieren Sie bitte die VIPA-Hotline!

Ereignis-ID	Bedeutung
0xEA64	PROFINET/EtherCAT-CP Konfigurationsfehler Zinfo1: Bit 0: Zu viele Devices Bit 1: Zu viele Devices pro Millisekunde Bit 2: Zu viele Eingangsbytes pro Millisekunde Bit 3: Zu viele Ausgangsbytes pro Millisekunde Bit 4: Zu viele Eingangsbytes pro Device Bit 5: Zu viele Ausgangsbytes pro Device Bit 6: Zu viele Produktiv-Verbindungen Bit 7: Zu viele Eingangsbytes im Prozessabbild Bit 8: Zu viele Ausgangsbytes im Prozessabbild Bit 9: Konfiguration nicht verfügbar Bit 10: Konfiguration ungültig Bit 11: Zykluszeit zu klein Bit 12: Aktualisierungszeit zu groß Bit 13: Ungültige Devicenummer Bit 14: CPU ist als I-Device konfiguriert Bit 15: IP Adresse auf anderem Weg beziehen, wird für die IP-Adresse des Controllers nicht unterstützt
0xEA65	Interner Fehler - Kontaktieren Sie bitte die VIPA-Hotline!
0xEA66	PROFINET-IO-Controller Fehler im Kommunikationsstack PK: Rackslot OBNr: StackError.Service DatId: StackError.DeviceRef ZInfo1: StackError.Error.Code ZInfo2: StackError.Error.Detail ZInfo3: StackError.Error.AdditionalDetail << 8 + StackError.Error.AreaCode
0xEA67	Fehler - PROFINET-IO-Controller - Datensatz lesen PK: Fehlertyp 0: DATA_RECORD_ERROR_LOCAL 1: DATA_RECORD_ERROR_STACK 2: DATA_RECORD_ERROR_REMOTE OBNr: PROFINET-IO-Controller slot DatId: Device-Nr ZInfo1: Datensatznummer ZInfo2: Datensatzhandle ZInfo3: Interner Fehlercode für Service-Zwecke

Ereignis-ID	Bedeutung
0xEA68	Fehler - PROFINET-IO-Controller - Datensatz schreiben PK: Fehlertyp 0: DATA_RECORD_ERROR_LOCAL 1: DATA_RECORD_ERROR_STACK 2: DATA_RECORD_ERROR_REMOTE OBNr: PROFINET-IO-Controller slot DatId: Device-Nr ZInfo1: Datensatznummer ZInfo2: Datensatzhandle ZInfo3: Interner Fehlercode für Service-Zwecke
0xEA69	Interner Fehler - Kontaktieren Sie bitte die VIPA-Hotline!
0xEA6A	PROFINET-IO-Controller Service-Fehler im Kommunikationsstack PK: Rackslot OBNr: ServiceIdentifier DatId: 0 ZInfo1: ServiceError.Code ZInfo2: ServiceError.Detail ZInfo3: ServiceError.AdditionalDetail
0xEA6B	PROFINET-IO-Controller Vendor ID mismatch PK: Rackslot OBNr: PLC-Mode DatId: 0 ZInfo1: Device ID ZInfo2: - ZInfo3: -
0xEA6C	PROFINET-IO-Controller Device ID mismatch PK: Rackslot OBNr: PLC-Mode DatId: 0 ZInfo1: Device ID ZInfo2: - ZInfo3: -

## VIPA-spezifische Diagnose-Einträge

Ereignis-ID	Bedeutung
0xEA6D	PROFINET-IO-Controller No empty name PK: Rackslot OBNr: PLC-Mode DatId: 0 ZInfo1: Device ID ZInfo2: - ZInfo3: -
0xEA6E	PROFINET-IO-Controller RPC response missing PK: Rackslot OBNr: PLC-Mode DatId: 0 ZInfo1: Device ID ZInfo2: - ZInfo3: -
0xEA6F	PROFINET-IO-Controller PN module mismatch PK: Rackslot OBNr: PLC-Mode DatId: 0 ZInfo1: Device ID ZInfo2: - ZInfo3: -
0xEA97	Speicherfehler SBUS-Service Kanal ZInfo3 = Slot
0xEA98	Timeout beim Warten, dass ein SBUS-Modul (Server) rebootet hat
0xEA99	Fehler beim File-Lesen über SBUS

Ereignis-ID	Bedeutung
0xEAA0	Emac Error ist aufgetreten OBNr: Aktueller PLC-Mode ZInfo1: Diagnoseadresse des Masters / Controllers ZInfo2: 0: Kein Rx Queue ist voll 1: Kein Sendepuffer verfügbar 2: Sendestrom ist abgerissen; senden fehlgeschlagen 3: Wiederholungsversuche ausgeschöpft 4: Kein Empfangspuffer in Emac DMA verfügbar 5: Emac DMA Transfer abgebrochen 6: Queue Overflow 7: Nicht erwartetes Packet empfangen ZInfo3: Anzahl der aufgetretenen Fehler
0xEAB0	Ungültiger Link-Mode OBNr: Aktueller PLC-Mode ZInfo1: Diagnoseadresse des Masters/Controllers ZInfo2: Aktueller LinkMode 0x01: 10MBit Full-Duplex 0x02: 100MBit Half-Duplex 0x03: 100Mbit Full-Duplex 0x05: 10Mbit Half-Duplex 0xFF: Link Mode nicht definiert
0xEB03	SLIO Fehler beim IO-Mapping
0xEB10	SLIO Fehler: Busfehler ZInfo1: Fehlerart 0x82: ErrorAlarm
0xEB20	SLIO Fehler: Alarminformationen undefiniert
0xEB21	SLIO Fehler bei Zugriff auf Konfigurationsdaten

Ereignis-ID	Bedeutung
0xEC03	<p>EtherCAT: Konfigurationsfehler</p> <p>ZInfo1: Errorcode</p> <p>1: NUMBER_OF_SLAVES_NOT_SUPPORTED</p> <p>2: SYSTEM_IO_NR_INVALID</p> <p>3: INDEX_FROM_SLOT_ERROR</p> <p>4: MASTER_CONFIG_INVALID</p> <p>5: MASTER_TYPE_ERROR</p> <p>6: SLAVE_DIAG_ADDR_INVALID</p> <p>7: SLAVE_ADDR_INVALID</p> <p>8: SLAVE_MODULE_IO_CONFIG_INVALID</p> <p>9: LOG_ADDR_ALREADY_IN_USE</p> <p>10: NULL_PTR_CHECK_ERROR</p> <p>11: IO_MAPPING_ERROR</p> <p>12: ERROR</p>
0xEC04	<p>EtherCAT: Mehrfach-Parametrierung einer Peripherieadresse</p> <p>Zinfo1 : Peripherie-Adresse</p> <p>Zinfo2 : Steckplatz</p>
0xEC10	<p>EtherCAT: Wiederkehr Bus mit allen Slaves</p> <p>OB-StartInfo (Lokaldaten) StartEvent und Eventclass: 0xEC10</p> <p>DatID:</p> <p>0xXXYY:</p> <p>XX=0x54 bei Eingangsadresse in ZInfo1,</p> <p>XX=0x55 bei Ausgangsadresse.</p> <p>YY=0x00 Station nicht verfügbar,</p> <p>YY=0x01 Station verfügbar (Prozessdaten)</p> <p>ZInfo1: 0xXXYY (XX=OldState, YY=NewState)</p> <p>ZInfo2: Diagnoseadresse des Masters</p> <p>ZInfo3: Anzahl der Station, die nicht im selben State sind, wie der Master (&gt; 0)</p>

Ereignis-ID	Bedeutung
0xEC11	<p>EtherCAT: Wiederkehr Bus mit fehlenden Slaves</p> <p>OB-StartInfo (Lokaldaten) StartEvent und Eventclass: 0xEC11</p> <p>DatID:</p> <p>0xXXYY:</p> <p>XX=0x54 bei Eingangsadresse in ZInfo1, XX=0x55 bei Ausgangsadresse.</p> <p>YY=0x00 Station nicht verfügbar, YY=0x01 Station verfügbar (Prozessdaten)</p> <p>ZInfo1: 0xXXYY (XX=OldState, YY=NewState)</p> <p>ZInfo2: Diagnoseadresse des Masters</p> <p>ZInfo3: Anzahl der Station, die nicht im selben State sind, wie der Master (&gt; 0)</p>
0xEC12	<p>EtherCAT: Wiederkehr Slave</p> <p>OB-StartInfo (Lokaldaten) StartEvent und Eventclass: 0xEC12</p> <p>DatID:</p> <p>0xXXYY:</p> <p>XX=0x54 bei Eingangsadresse in ZInfo1, XX=0x55 bei Ausgangsadresse.</p> <p>YY=0x00 Station nicht verfügbar, YY=0x01 Station verfügbar (Prozessdaten)</p> <p>ZInfo1: 0xXXYY (XX=OldState, YY=NewState)</p> <p>ZInfo2: Diagnoseadresse der Station</p> <p>ZInfo3: AIStatusCode</p>
0xEC30	<p>EtherCAT: Topologie OK</p> <p>OB-StartInfo (Lokaldaten) StartEvent und Eventclass: 0xEC30</p> <p>ZInfo2: Diagnoseadresse des Masters</p>
0xEC50	<p>EtherCAT: DC nicht in Sync</p> <p>ZInfo1: Diagnoseadresse des Masters</p>
0xED10	<p>EtherCAT: Ausfall Bus</p> <p>OB-StartInfo (Lokaldaten) StartEvent und Eventclass: 0xED10</p> <p>DatID:</p> <p>0xXXYY:</p> <p>XX=0x54 bei Eingangsadresse in ZInfo1, XX=0x55 bei Ausgangsadresse.</p> <p>YY=0x00 Station nicht verfügbar, YY=0x01 Station verfügbar (Prozessdaten)</p> <p>ZInfo1: 0xXXYY (XX=OldState, YY=NewState)</p> <p>ZInfo2: Diagnoseadresse des Masters</p> <p>ZInfo3: Anzahl der Station, die nicht im selben State sind, wie der Master</p>

Ereignis-ID	Bedeutung
0xED12	<p>EtherCAT: Ausfall Slave</p> <p>OB-StartInfo (Lokaldaten) StartEvent und Eventclass: 0xED12</p> <p>DatID:</p> <p>0xXXYY:</p> <p>XX=0x54 bei Eingangsadresse in ZInfo1, XX=0x55 bei Ausgangsadresse.</p> <p>YY=0x00 Station nicht verfügbar, YY=0x01 Station verfügbar (Prozessdaten)</p> <p>ZInfo1: 0xXXYY (XX=OldState, YY=NewState)</p> <p>ZInfo2: Diagnoseadresse der Station</p> <p>ZInfo3: AIStatusCode</p>
0xED20	<p>EtherCAT: Bus-Statuswechsel, der keinen OB86 hervorruft</p> <p>OB-StartInfo (Lokaldaten) StartEvent und Eventclass: 0xED20</p> <p>DatID:</p> <p>0xXXYY:</p> <p>XX=0x54 bei Eingangsadresse in ZInfo1, XX=0x55 bei Ausgangsadresse.</p> <p>YY=0x00 Station nicht verfügbar, YY=0x01 Station verfügbar (Prozessdaten)</p> <p>ZInfo1: 0xXXYY (XX=OldState, YY=NewState)</p> <p>ZInfo2: Diagnoseadresse des Masters</p> <p>ZInfo3: Anzahl der Station, die nicht im selben State sind, wie der Master</p>
0xED21	<p>EtherCAT: fehlerhafter Bus-Statuswechsel</p> <p>OB: 0x00</p> <p>PK: 0x00</p> <p>DatID:</p> <p>0xXXYY:</p> <p>XX=0x54 bei Eingangsadresse in ZInfo1, XX=0x55 bei Ausgangsadresse.</p> <p>YY=0x00 Station nicht verfügbar, YY=0x01 Station verfügbar (Prozessdaten)</p> <p>ZInfo1: 0xXXYY (XX = current state, YY = expected state)</p> <p>ZInfo2: Diagnoseadresse des Masters</p> <p>ZInfo3: ErrorCode:</p> <p>0x0008: Busy</p> <p>0x000B: Unzulässige Parameter</p> <p>0x000E: Unzulässiger Status</p> <p>0x0010: Zeitüberschreitung</p>

Ereignis-ID	Bedeutung
0xED22	<p>EtherCAT: Slave-Statuswechsel, der keinen OB86 hervorruft            OB-StartInfo (Lokaldaten) StartEvent und Eventclass: 0xED22            DatID:            0xXXYY:            XX=0x54 bei Eingangsadresse in ZInfo1,            XX=0x55 bei Ausgangsadresse.            YY=0x00 Station nicht verfügbar,            YY=0x01 Station verfügbar (Prozessdaten)            ZInfo1: 0xXXYY (XX=OldState, YY=NewState)            ZInfo2: Diagnoseadresse der Station            ZInfo3: AIStatusCode</p>
0xED30	<p>EtherCAT: Topolgy Mismatch            OB-StartInfo (Lokaldaten) StartEvent und Eventclass: 0xED30            ZInfo2: Diagnoseadresse des Masters</p>
0xED31	<p>EtherCAT: Alarm Queue Overflow            OB-StartInfo (Lokaldaten) StartEvent und Eventclass: 0xED31            ZInfo2: Diagnoseadresse des Masters</p>
0xED40 ... 0xED4F	<p>Interner Fehler - Kontaktieren Sie bitte die VIPA-Hotline!</p>
0xED50	<p>EtherCAT: DC in Sync            ZInfo1: Diagnoseadresse des Masters</p>
0xED60	<p>EtherCAT: Diagnosepuffer CP:            Slave-Statuswechsel            PK: 0            OB: PLC-Mode            DatID 1/2: 0            ZInfo1: 0x00YY:            YY: Neuer EtherCAT-Status des Slaves            ZInfo2: EtherCAT-Stationsadresse            ZInfo3: AIStatusCode (EtherCAT-spezifischer Fehlercode)</p>

Ereignis-ID	Bedeutung
0xED61	EtherCAT: Diagnosepuffer CP: CoE-Emergency PK: EtherCAT-Stationsadresse (Low-Byte) OB: EtherCAT-Stationsadresse (High-Byte) DatID 1/2: Error-Code ZInfo1: 0xYYZZ: YY: Error-Register ZZ: MEF Byte 1 ZInfo 2: 0xYYZZ: YY: MEF Byte 2 ZZ: MEF Byte 3 Zinfo3: 0xYYZZ: YY: MEF Byte 4 ZZ: MEF Byte 5
0xED62	EtherCAT: Diagnosepuffer CP: Fehler bei SDO-Zugriff während State-Wechsel PK: EtherCAT-Stationsadresse (Low-Byte) OB: EtherCAT-Stationsadresse (High-Byte) DatID 1/2: Subindex ZInfo1: Index ZInfo2: SDO-Errorcode (High-Word) Zinfo3: SDO-Errorcode (Low-Word)
0xED70	EtherCAT: Diagnosepuffer CP: Doppelte Hot Connect Gruppe erkannt PK: 0 OB: PLC-Mode DatID 1/2: 0 ZInfo1: Diagnoseadresse des Masters ZInfo2: EtherCAT-Stationsadresse Zinfo3: 0
0xEE00	Zusatzinformation bei UNDEF_OPCODE
0xEE01	Interner Fehler - Kontaktieren Sie bitte die VIPA-Hotline!
0xEEEE	CPU wurde komplett urgelöscht, da der Hochlauf nach NetzeIN nicht beendet werden konnte.
0xEF11 ... 0xEF13	Interner Fehler - Kontaktieren Sie bitte die VIPA-Hotline!

Ereignis-ID	Bedeutung
0xEFFF	Interner Fehler - Kontaktieren Sie bitte die VIPA-Hotline!
PK: C-Sourcemodulnummer   DatID: Zeilennummer	

## 5.21 Mit Testfunktionen Variablen steuern und beobachten

### Übersicht

Zur Fehlersuche und zur Ausgabe von Variablenzuständen können Sie in Ihrem Siemens SIMATIC Manager unter dem Menüpunkt **Test** verschiedene Testfunktionen aufrufen.

- Mit der Testfunktion "**Test → Beobachten**" können die Signalzustände von Operanden und das VKE angezeigt werden.
- Mit der Testfunktion "**Zielsystem → Variablen beobachten/steuern**" können die Signalzustände von Variablen geändert und angezeigt werden.

### "Test → Beobachten"

Diese Testfunktion zeigt die aktuellen Signalzustände und das VKE der einzelnen Operanden während der Programmbearbeitung an. Es können außerdem Korrekturen am Programm durchgeführt werden.



*Die CPU muss bei der Testfunktion "Beobachten" in der Betriebsart RUN sein!*

Die Statusbearbeitung kann durch Sprungbefehle oder Zeit- und Prozessalarmlen unterbrochen werden. Die Unterbrechung der Statusbearbeitung hat keinen Einfluss auf die Programmbearbeitung, sondern macht nur deutlich, dass die angezeigten Daten ab der Unterbrechungsstelle nicht mehr gültig sind. Die CPU hört an der Unterbrechungsstelle auf, Daten für die Statusanzeige zu sammeln und übergibt dem PG anstelle der noch benötigten Daten nur Daten mit dem Wert 0. Deshalb kann es bei Verwendung von Sprungbefehlen oder von Zeit- und Prozessalarmen vorkommen, dass in der Statusanzeige eines Bausteins während dieser Programmbearbeitung nur der Wert 0 angezeigt wird für:

- das Verknüpfungsergebnis VKE
- Status / AKKU 1
- AKKU 2
- Zustandsbyte
- absolute Speicheradresse SAZ. Hinter SAZ erscheint dann ein "?".

### "Zielsystem → Variablen beobachten/steuern"

Diese Testfunktion gibt den Zustand eines beliebigen Operanden (Eingänge, Ausgänge, Merker, Datenwort, Zähler oder Zeiten) am Ende einer Programmbearbeitung an. Diese Informationen werden aus dem Prozessabbild der ausgesuchten Operanden entnommen. Während der "Bearbeitungskontrolle" oder in der Betriebsart STOP wird bei den Eingängen direkt die Peripherie eingelesen. Andernfalls wird nur das Prozessabbild der aufgerufenen Operanden angezeigt.

---

Mit Testfunktionen Variablen steuern und beobachten

- Steuern von Ausgängen
  - Dadurch kann die Verdrahtung und die Funktionstüchtigkeit von Ausgabebaugruppen kontrolliert werden.
  - Auch ohne Steuerungsprogramm können Ausgänge auf den gewünschten Signalzustand eingestellt werden. Das Prozessabbild wird dabei nicht verändert, die Sperre der Ausgänge jedoch aufgehoben.
- Steuern von Variablen
  - Folgende Variablen können geändert werden: E, A, M, T, Z und D.
  - Unabhängig von der Betriebsart der CPU wird das Prozessabbild binärer und digitaler Operanden verändert.
  - In der Betriebsart RUN wird die Programmbearbeitung mit den geänderten Prozessvariablen ausgeführt. Im weiteren Programmablauf können sie jedoch ohne Rückmeldung wieder verändert werden.
  - Die Prozessvariablen werden asynchron zum Programmablauf gesteuert.

## 6 Einsatz PtP-Kommunikation

### 6.1 Schnelleinstieg

#### Allgemein

Die CPU besitzt eine PROFIBUS/PtP-Schnittstelle mit fixer Pinbelegung. Nach dem Umräumen ist diese Schnittstelle deaktiviert. Durch entsprechende Projektierung können Sie die PtP-Funktionalität (point to point) aktivieren:

- PtP-Funktionalität
  - Mit der Funktionalität PtP ermöglicht die RS485-Schnittstelle eine serielle Punkt-zu-Punkt-Prozessanbindung zu verschiedenen Ziel- oder Quell-Systemen.
  - Die Aktivierung der PtP-Funktionalität erfolgt durch Einbindung der SPEEDBUS.GSD von VIPA im Hardwarekatalog. Nach der Installation können Sie die CPU in einem PROFIBUS-Master-System projektieren und hier auch die Schnittstelle auf PtP-Kommunikation umschalten.

#### Protokolle

Unterstützt werden die Protokolle bzw. Prozeduren ASCII, STX/ETX, 3964R, USS und Modbus.

#### Parametrierung

Die Parametrierung der seriellen Schnittstelle erfolgt zur Laufzeit unter Einsatz des FC/SFC 216 (SER\_CFG). Hierbei sind für alle Protokolle mit Ausnahme von ASCII die Parameter in einem DB abzufragen.

#### Kommunikation

Mit FCs/SFCs steuern Sie die Kommunikation. Das Senden erfolgt unter Einsatz des FC/SFC 217 (SER\_SND) und das Empfangen über FC/SFC 218 (SER\_RCV). Durch erneuten Aufruf des FC/SFC 217 SER\_SND bekommen Sie bei 3964R, USS und Modbus über RetVal einen Rückgabewert geliefert, der unter anderem auch aktuelle Informationen über die Quittierung der Gegenseite beinhaltet. Bei den Protokollen USS und Modbus können Sie durch Aufruf des FC/SFC 218 SER\_RCV nach einem SER\_SND das Quittungstelegramm auslesen. Die FCs/SFCs befinden sich im Lieferumfang der CPU.

#### Übersicht der FCs/SFCs für die serielle Kommunikation

Folgende FC/SFCs kommen für die serielle Kommunikation zum Einsatz:

FC/SFC		Beschreibung
FC/SFC 216	SER_CFG	RS485 Parametrieren
FC/SFC 217	SER_SND	RS485 Senden
FC/SFC 218	SER_RCV	RS485 Empfangen

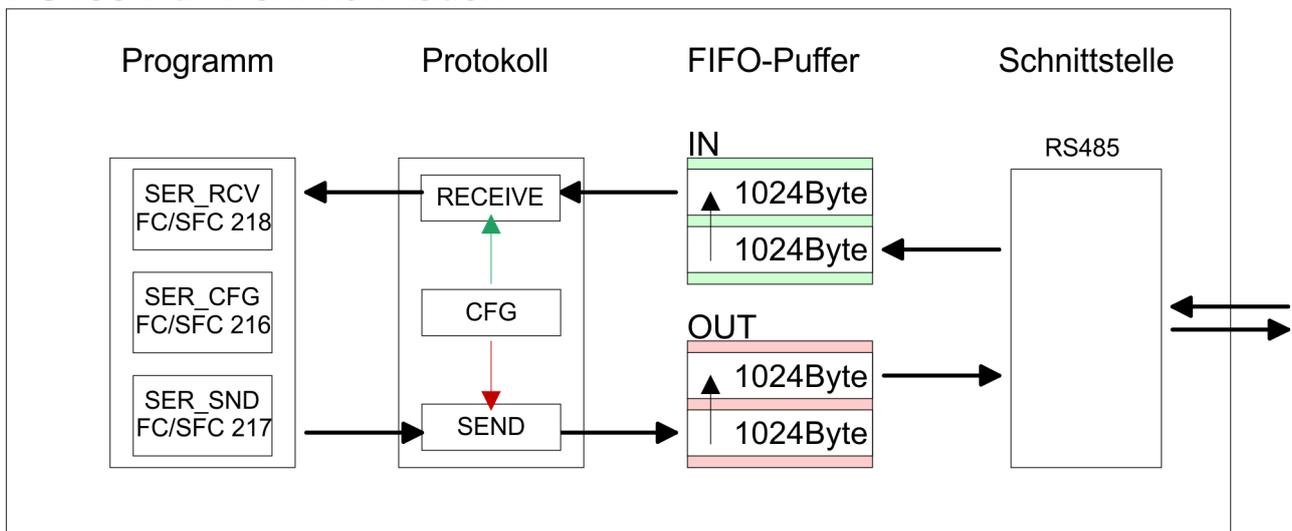
## 6.2 Prinzip der Datenübertragung

### Übersicht

Die Datenübertragung wird zur Laufzeit über FC/SFCs gehandhabt. Das Prinzip der Datenübertragung ist für alle Protokolle identisch und soll hier kurz gezeigt werden.

- Daten, die von der CPU in den entsprechenden Datenkanal geschrieben werden, werden in einen FIFO-Sendepuffer (first in first out) mit einer Größe von 2x1024Byte abgelegt und von dort über die Schnittstelle ausgegeben.
- Empfängt die Schnittstelle Daten, werden diese in einem FIFO-Empfangspuffer mit einer Größe von 2x1024Byte abgelegt und können dort von der CPU gelesen werden.
- Sofern Daten mittels eines Protokolls übertragen werden, erfolgt die Einbettung der Daten in das entsprechende Protokoll automatisch.
- Im Gegensatz zu ASCII- und STX/ETX erfolgt bei den Protokollen 3964R, USS und Modbus die Datenübertragung mit Quittierung der Gegenseite.
- Durch erneuten Aufruf des FC/SFC 217 SER\_SND bekommen Sie über RetVal einen Rückgabewert geliefert, der unter anderem auch aktuelle Informationen über die Quittierung der Gegenseite beinhaltet.
- Zusätzlich ist bei USS und Modbus nach einem SER\_SND das Quittungstelegramm durch Aufruf des FC/SFC 218 SER\_RCV auszulesen.

## RS485-PtP-Kommunikation



## 6.3 Einsatz der RS485-Schnittstelle für PtP

### Aktivierung der RS485 für PtP-Betrieb

Standardmäßig ist die RS485-Schnittstelle deaktiviert. Über eine Hardware-Konfiguration können Sie unter Objekteigenschaften über den Parameter "Funktion RS485" die RS485-Schnittstelle auf PtP-Betrieb (point to point) umschalten.

**Voraussetzung**

Damit Sie die VIPA-spezifischen CPU-Parameter einstellen können, ist die Installation der SPEEDBUS.GSD von VIPA im Hardwarekatalog erforderlich. Nach der Installation können Sie die CPU in einem PROFIBUS-Master-System projektieren und entsprechend die Parameter anpassen.

**SPEEDBUS.GSD installieren**

Die GSD (Geräte-Stamm-Datei) ist in folgenden Sprachversionen online verfügbar. Weitere Sprachen erhalten Sie auf Anfrage:

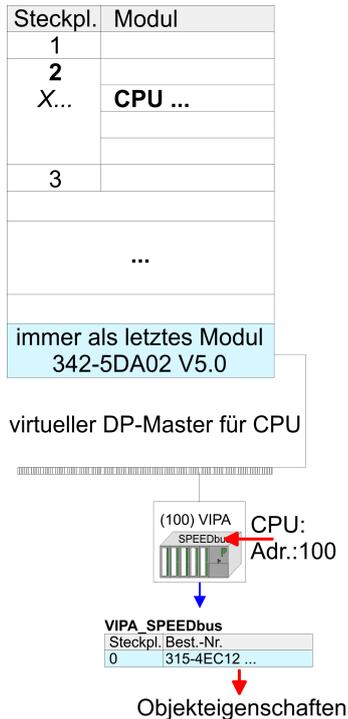
Name	Sprache
SPEEDBUS.GSD	deutsch (default)
SPEEDBUS.GSG	deutsch
SPEEDBUS.GSE	englisch

Die GSD-Dateien finden Sie auf [www.vipa.com](http://www.vipa.com) im "Service"-Bereich.

Die Einbindung der SPEEDBUS.GSD erfolgt nach folgender Vorgehensweise:

1. ▶ Gehen Sie auf [www.vipa.com](http://www.vipa.com)
2. ▶ Klicken Sie auf "Service → Download → GSD- und EDS-Files → Profibus"
3. ▶ Laden Sie die Datei Cx000023\_Vxxx.
4. ▶ Extrahieren Sie die Datei in Ihr Arbeitsverzeichnis. Die SPEEDBUS.GSD befindet sich im Verzeichnis VIPA\_System\_300S.
5. ▶ Starten Sie den Hardware-Konfigurator von Siemens.
6. ▶ Schließen Sie alle Projekte.
7. ▶ Gehen Sie auf "Extras → Neue GSD-Datei installieren".
8. ▶ Navigieren Sie in das Verzeichnis VIPA\_System\_300S und geben Sie **SPEEDBUS.GSD** an.
  - ⇒ Alle SPEED7-CPU's und -Module des System 300S von VIPA sind jetzt im Hardwarekatalog unter Profibus-DP / Weitere Feldgeräte / I/O / VIPA\_SPEEDBUS enthalten.

## Vorgehensweise



Die Einbindung der CPU 315-4EC12 erfolgt in Form eines virtuellen PROFIBUS Master-Systems nach folgender Vorgehensweise:

1. ► Führen Sie eine Hardware-Konfiguration für die CPU durch.  
↳ Kapitel 5.4 "Hardware-Konfiguration - CPU" auf Seite 43
2. ► Projektieren Sie immer als letztes Modul einen Siemens DP-Master CP 342-5 (342-5DA02 V5.0). Vernetzen und parametrieren Sie diesen in der Betriebsart "DP-Master".
3. ► Binden Sie das Slave-System "VIPA\_SPEEDbus" an. Nach der Installation der SPEEDBUS.GSD finden Sie dieses im Hardware-Katalog unter Profibus-DP / Weitere Feldgeräte / I/O / VIPA\_SPEEDBUS.
4. ► Stellen Sie für das Slave-System die PROFIBUS-Adresse 100 ein.
5. ► Platzieren Sie auf dem Steckplatz 0 die VIPA CPU 315-4EC12 aus dem Hardware-Katalog von VIPA\_SPEEDbus.
6. ► Durch Doppelklick auf die eingefügte CPU 315-4EC12 gelangen Sie in den Eigenschaften-Dialog der CPU.

Sobald Sie Ihr Projekt zusammen mit Ihrem SPS-Programm in die CPU übertragen, werden die Parameter nach dem Hochlauf übernommen.



Die hier gezeigte Hardware-Konfiguration ist nur erforderlich, wenn Sie die VIPA-spezifischen Parameter anpassen möchten.

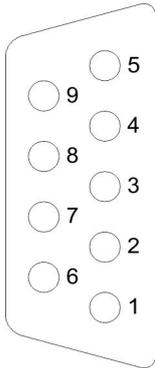
## Einstellung der PtP-Parameter

1. ► Durch Doppelklick auf die im Slave-System eingefügte CPU 315-4EC12 gelangen Sie in den Eigenschaften-Dialog der CPU.
2. ► Stellen Sie den Parameter "Funktion RS485 X3" auf "PtP".

## Eigenschaften RS485

- Logische Zustände als Spannungsdifferenz zwischen 2 verdrehten Adern
- Serielle Busverbindung in Zweidrahttechnik im Halbduplex-Verfahren
- Datenübertragung bis 500m Entfernung
- Datenübertragungsrate bis 115,2kBit/s

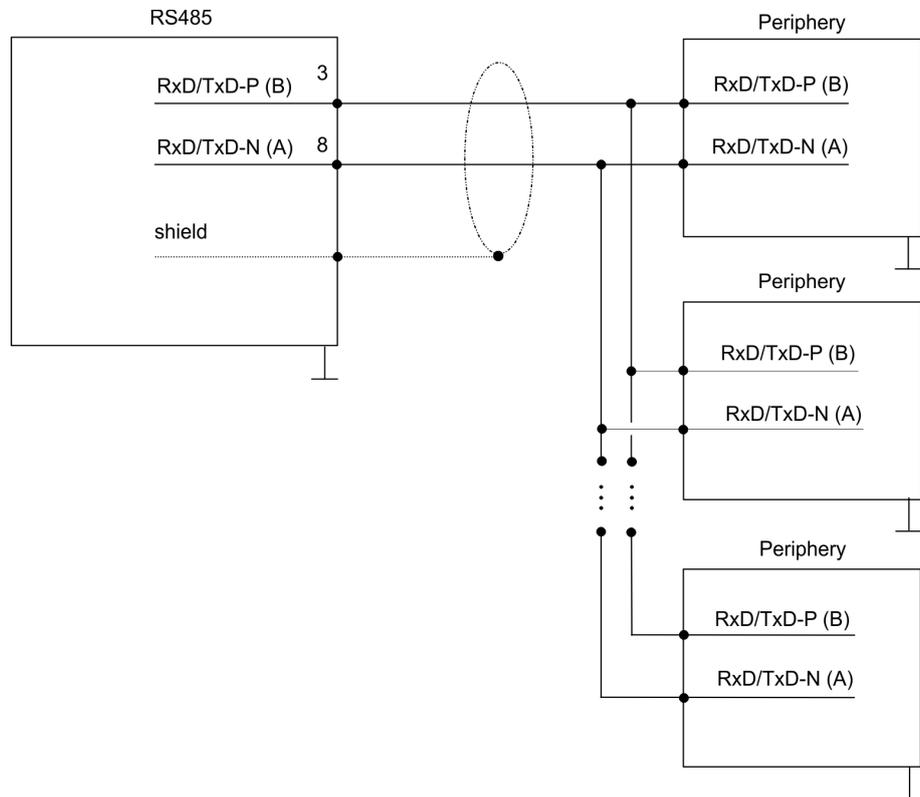
**RS485**



*9polige SubD-Buchse*

Pin	RS485
1	n.c.
2	M24V
3	RxD/TxD-P (Leitung B)
4	RTS
5	M5V
6	P5V
7	P24V
8	RxD/TxD-N (Leitung A)
9	n.c.

**Anschluss**



**6.4 Parametrierung**

**6.4.1 FC/SFC 216 - SER\_CFG**

**Beschreibung**

Die Parametrierung erfolgt zur Laufzeit unter Einsatz des FC/SFC 216 (SER\_CFG). Hierbei sind die Parameter für STX/ETX, 3964R, USS und Modbus in einem DB abzulegen.

**Parameter**

Parameter	Deklaration	Datentyp	Beschreibung
PROTOCOL	IN	BYTE	1=ASCII, 2=STX/ETX, 3=3964R
PARAMETER	IN	ANY	Zeiger zu den Protokoll-Parametern
BAUDRATE	IN	BYTE	Nr. der Baudrate
CHARLEN	IN	BYTE	0=5Bit, 1=6Bit, 2=7Bit, 3=8Bit
PARITY	IN	BYTE	0=Non, 1=Odd, 2=Even
STOPBITS	IN	BYTE	1=1Bit, 2=1,5Bit, 3=2Bit
FLOWCONTROL	IN	BYTE	1 (fix)
RETVAL	OUT	WORD	Rückgabewert (0 = OK)

Alle Zeitangaben für Timeouts sind als Hexadezimaler Wert anzugeben. Den Hex-Wert erhalten Sie, indem Sie die gewünschte Zeit in Sekunden mit der Baudrate multiplizieren.

Beispiel:

Gewünschte Zeit 8ms bei einer Baudrate von 19200Baud

Berechnung:  $19200\text{Bit/s} \times 0,008\text{s} \approx 154\text{Bit} \rightarrow (9Ah)$

Als Hex-Wert ist 9Ah vorzugeben.

**PROTOCOL**

Geben Sie hier das Protokoll an, das verwendet werden soll.

Zur Auswahl stehen:

- 1: ASCII
- 2: STX/ETX
- 3: 3964R
- 4: USS Master
- 5: Modbus RTU Master
- 6: Modbus ASCII Master

**PARAMETER (als DB)**

Bei eingestelltem ASCII-Protokoll wird dieser Parameter ignoriert.

Für die Protokolle geben Sie hier einen DB an, der die Kommunikationsparameter beinhaltet und für die jeweiligen Protokolle STX/ETX, 3964R, USS und Modbus folgenden Aufbau hat:

Datenbaustein bei STX/ETX			
DBB0:	STX1	BYTE	(1. Start-Zeichen in hexadezimaler Form)
DBB1:	STX2	BYTE	(2. Start-Zeichen in hexadezimaler Form)
DBB2:	ETX1	BYTE	(1. Ende-Zeichen in hexadezimaler Form)

DBB3:	ETX2	BYTE	(2. Ende-Zeichen in hexadezimaler Form)
DBW4:	TIMEOUT	WORD	(max. zeitlicher Abstand zwischen 2 Telegrammen)



*Das Zeichen für Start bzw. Ende sollte immer ein Wert kleiner 20 sein, ansonsten wird das Zeichen ignoriert!*

*Tragen Sie immer für nicht benutzte Zeichen FFh ein!*

Datenbaustein bei 3964R

DBB0:	Prio	BYTE	(Die Priorität beider Partner muss unterschiedlich sein)
DBB1:	ConnAttmptNr	BYTE	(Anzahl der Verbindungsaufbauversuche)
DBB2:	SendAttmptNr	BYTE	(Anzahl der Telegrammwiederholungen)
DBB4:	CharTimeout	WORD	(Zeichenverzugszeit)
DBW6:	ConfTimeout	WORD	(Quittungsverzugszeit)

Datenbaustein bei USS

DBW0:	Timeout	WORD	(Verzugszeit)
-------	---------	------	---------------

Datenbaustein bei Modbus-Master

DBW0:	Timeout	WORD	(Antwort-Verzugszeit)
-------	---------	------	-----------------------

**BAUDRATE**

Geschwindigkeit der Datenübertragung in Bit/s (Baud).

04h:	1200Baud	05h:	1800Baud	06h:	2400Baud	07h:	4800Baud
08h:	7200Baud	09h:	9600Baud	0Ah:	14400Baud	0Bh:	19200Baud
0Ch:	38400Baud	0Dh:	57600Baud	0Eh:	115200Baud		

**CHARLEN**

Anzahl der Datenbits, auf die ein Zeichen abgebildet wird.

0: 5Bit	1: 6Bit	2: 7Bit	3: 8Bit
---------	---------	---------	---------

**PARITY**

Die Parität ist je nach Wert gerade oder ungerade. Zur Paritätskontrolle werden die Informationsbits um das Paritätsbit erweitert, das durch seinen Wert ("0" oder "1") den Wert aller Bits auf einen vereinbarten Zustand ergänzt. Ist keine Parität vereinbart, wird das Paritätsbit auf "1" gesetzt, aber nicht ausgewertet.

0: NONE	1: ODD	2: EVEN
---------	--------	---------

**STOPBITS**

Die Stopbits werden jedem zu übertragenden Zeichen nachgesetzt und kennzeichnen das Ende eines Zeichens.

1: 1Bit	2: 1,5Bit	3: 2Bit
---------	-----------	---------

**FLOWCONTROL**

Der Parameter *FLOWCONTROL* wird ignoriert. Beim Senden ist RTS=1, beim Empfangen ist RTS=0.

**RETVAL FC/SFC 216 (Rückgabewert)**

Rückgabewerte, die der Baustein liefert:

Fehler-code	Beschreibung
0000h	kein Fehler
809Ah	Schnittstelle ist nicht vorhanden bzw. Schnittstelle wird für PROFIBUS verwendet.  Bei der VIPA System SLIO-CPU und FeatureSet PTP_NO ist nur das ASCII Protokoll konfigurierbar. Wird ein anderes Protokoll ausgewählt wird der FC/SFC 216 ebenfalls mit diesem Fehlercode verlassen.
8x24h	Fehler in FC/SFC-Parameter x, mit x: 1: Fehler in <i>PROTOKOLL</i> 2: Fehler in <i>PARAMETER</i> 3: Fehler in <i>BAUDRATE</i> 4: Fehler in <i>CHARLENGTH</i> 5: Fehler in <i>PARITY</i> 6: Fehler in <i>STOPBITS</i> 7: Fehler in <i>FLOWCONTROL</i> (Parameter fehlt)
809xh	Fehler in Wert des FC/SFC-Parameter x, mit x: 1: Fehler in <i>PROTOKOLL</i> 3: Fehler in <i>BAUDRATE</i> 4: Fehler in <i>CHARLENGTH</i> 5: Fehler in <i>PARITY</i> 6: Fehler in <i>STOPBITS</i>
8092h	Zugriffsfehler auf Parameter-DB (DB zu kurz)
828xh	Fehler in Parameter x von DB-Parameter mit x: 1: Fehler im 1. Parameter 2: Fehler im 2. Parameter ...

## 6.5 Kommunikation

### 6.5.1 Übersicht

Die Kommunikation erfolgt über die Sende- und Empfangsbausteine FC/SFC 217 (SER\_SND) und FC/SFC 218 (SER\_RCV). Die FCs/ SFCs befinden sich im Lieferumfang der CPU.

### 6.5.2 FC/SFC 217 - SER\_SND

#### Beschreibung

Mit diesem Baustein werden Daten über die serielle Schnittstelle gesendet. Durch erneuten Aufruf des FC/SFC 217 SER\_SND bekommen Sie bei 3964R, USS und Modbus über RETVAL einen Rückgabewert geliefert, der unter anderem auch aktuelle Informationen über die Quittierung der Gegenseite beinhaltet.

Zusätzlich ist bei USS und Modbus nach einem SER\_SND das Quittungstelegramm durch Aufruf des FC/SFC 218 SER\_RCV auszu- lesen.

#### Parameter

Parameter	Deklaration	Datentyp	Beschreibung
DATAPTR	IN	ANY	Zeiger auf Sendedaten
DATALEN	OUT	WORD	Länge der Sendedaten
RETVAL	OUT	WORD	Rückgabewert (0 = OK)

#### DATAPTR

Geben Sie hier einen Bereich vom Typ Pointer für den Sendepuffer an, in den die Daten, die gesendet werden sollen, abzulegen sind. Anzugeben sind Typ, Anfang und Länge.

Beispiel:

Daten liegen in DB5 ab 0.0 mit einer Länge von 124Byte

DataPtr:=P#DB5.DBX0.0 BYTE 124

#### DATALEN

Wort, in dem die Anzahl der gesendeten Bytes abgelegt wird.

Werden unter **ASCII** die Daten intern mittels FC/SFC 217 schneller an die serielle Schnittstelle übertragen als sie gesendet werden können, kann aufgrund eines Pufferüberlaufs die zu sendende Datenlänge von *DATALEN* abweichen. Dies sollte im Anwenderprogramm berücksichtigt werden!

Bei **STX/ETX**, **3964R**, **Modbus** und **USS** wird immer die unter *DATAPTR* angegebene Länge oder 0 eingetragen.

#### RETVAL FC/SFC 217 (Rückgabewerte)

Rückgabewerte, die der Baustein liefert:

Fehler- code	Beschreibung
0000h	Daten gesendet - fertig
1000h	Nichts gesendet (Datenlänge 0)

Fehler-code	Beschreibung
20xxh	Protokoll wurde fehlerfrei ausgeführt mit xx-Bitmuster für Diagnose
7001h	Daten liegen im internen Puffer - aktiv (busy)
7002h	Transfer - aktiv
80xxh	Protokoll wurde fehlerhaft ausgeführt mit xx-Bitmuster für Diagnose (keine Quittung der Gegenseite)
90xxh	Protokoll wurde nicht ausgeführt mit xx-Bitmuster für Diagnose (keine Quittung der Gegenseite)
8x24h	Fehler in FC/SFC-Parameter x, mit x: 1: Fehler in <i>DATAPTR</i> 2: Fehler in <i>DATALEN</i>
8122h	Fehler in Parameter <i>DATAPTR</i> (z.B. DB zu kurz)
807Fh	Interner Fehler
809Ah	Schnittstelle nicht vorhanden bzw. Schnittstelle wird für PROFIBUS verwendet
809Bh	Schnittstelle nicht konfiguriert

### Protokollspezifische RETVAL-Werte

#### ASCII

Wert	Beschreibung
9000h	Pufferüberlauf (keine Daten gesendet)
9002h	Daten sind zu kurz (0Byte)

#### STX/ETX

Wert	Beschreibung
9000h	Pufferüberlauf (keine Daten gesendet)
9001h	Daten sind zu lang (>1024Byte)
9002h	Daten sind zu kurz (0Byte)
9004h	Unzulässiges Zeichen

#### 3964R

Wert	Beschreibung
2000h	Senden fertig ohne Fehler
80FFh	NAK empfangen - Fehler in der Kommunikation
80FEh	Datenübertragung ohne Quittierung der Gegenseite oder mit fehlerhafter Quittierung
9000h	Pufferüberlauf (keine Daten gesendet)

Wert	Beschreibung
9001h	Daten sind zu lang (>1024Byte)
9002h	Daten sind zu kurz (0Byte)

### USS

Fehler-code	Beschreibung
2000h	Senden fertig ohne Fehler
8080h	Empfangspuffer voll (kein Platz für Quittung)
8090h	Quittungsverzugszeit überschritten
80F0h	Falsche Checksumme in Rückantwort
80FEh	Falsches Startzeichen in der Rückantwort
80FFh	Falsche Slave-Adresse in der Rückantwort
9000h	Pufferüberlauf (keine Daten gesendet)
9001h	Daten sind zu lang (>1024Byte)
9002h	Daten sind zu kurz (<2Byte)

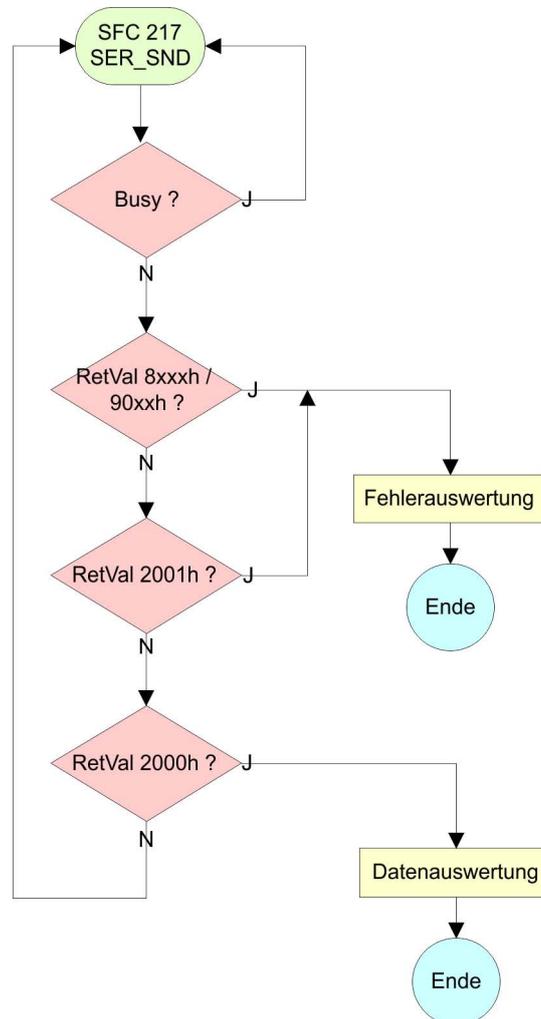
### Modbus RTU/ASCII Master

Fehler-code	Beschreibung
2000h	Senden fertig (positive Slave-Rückmeldung vorhanden)
2001h	Senden fertig (negative Slave-Rückmeldung vorhanden)
8080h	Empfangspuffer voll (kein Platz für Quittung)
8090h	Quittungsverzugszeit überschritten
80F0h	Falsche Checksumme in Rückantwort
80FDh	Länge der Rückantwort ist zu lang
80FEh	Falscher Funktionscode in der Rückantwort
80FFh	Falsche Slave-Adresse in der Rückantwort
9000h	Pufferüberlauf (keine Daten gesendet)
9001h	Daten sind zu lang (>1024Byte)
9002h	Daten sind zu kurz (<2Byte)

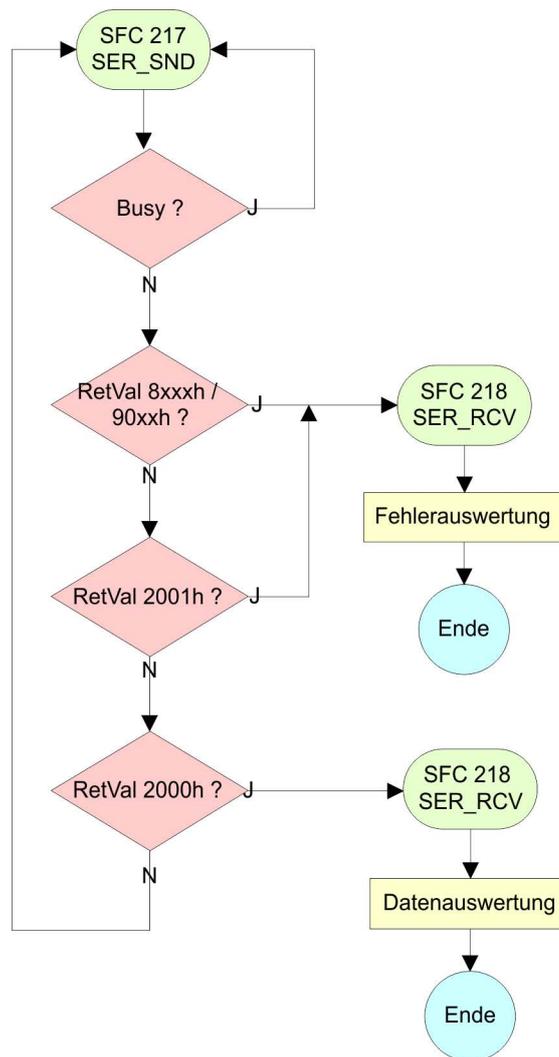
**Prinzip der Programmierung**

Nachfolgend soll kurz die Struktur zur Programmierung eines Sendeauftrags für die verschiedenen Protokolle gezeigt werden.

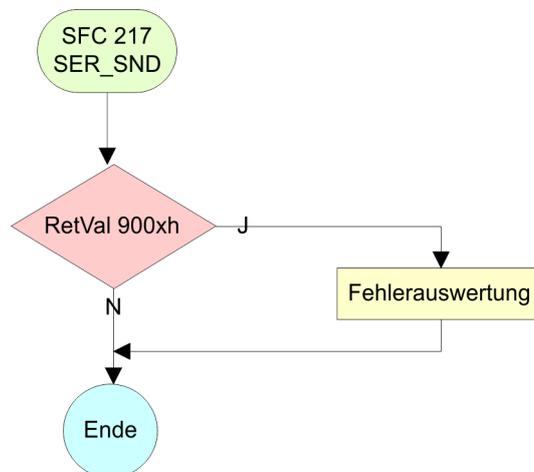
3964R



USS / Modbus



ASCII / STX/ETX



### 6.5.3 FC/SFC 218 - SER\_RCV

#### Beschreibung

Mit diesem Baustein werden Daten über die serielle Schnittstelle empfangen.

Bei den Protokollen USS und Modbus können Sie durch Aufruf des FC/SFC 218 SER\_RCV nach einem SER\_SND das Quittungstelegramm auslesen.

#### Parameter

Parameter	Deklaration	Datentyp	Beschreibung
DATAPTR	IN	ANY	Zeiger auf Empfangspuffer
DATALEN	OUT	WORD	Länge der empfangenen Daten
ERROR	OUT	WORD	Fehler-Nr.
RETVAL	OUT	WORD	Rückgabewert (0 = OK)

#### DATAPTR

Geben Sie hier einen Bereich vom Typ Pointer für den Empfangspuffer an, in den die Daten, die empfangen werden, abzulegen sind. Anzugeben sind Typ, Anfang und Länge.

Beispiel:

Daten sind in DB5 ab 0.0 mit einer Länge von 124Byte abzulegen  
DataPtr:=P#DB5.DBX0.0 BYTE 124

#### DATALEN

Wort, in dem die Anzahl der empfangenen Bytes abgelegt wird.

Bei **STX/ETX** und **3964R** wird immer die Länge der empfangenen Nutzdaten oder 0 eingetragen.

Unter **ASCII** wird hier die Anzahl der gelesenen Zeichen eingetragen. Dieser Wert kann von der Telegrammlänge abweichen.

#### ERROR

In diesem Wort erfolgt ein Eintrag im Fehlerfall.

Folgende Fehlermeldungen können protokollabhängig generiert werden:

#### ASCII

Bit	Fehler	Beschreibung
0	overrun	Überlauf, ein Zeichen konnte nicht schnell genug aus der Schnittstelle gelesen werden kann
1	framing error	Fehler, der anzeigt, dass ein definierter Bitrahmen nicht übereinstimmt, die zulässige Länge überschreitet oder eine zusätzliche Bitfolge enthält (Stopbitfehler)
2	parity	Paritätsfehler
3	overflow	Der Puffer ist voll.

**STX/ETX**

Bit	Fehler	Beschreibung
0	over-flow	Das empfangene Telegramm übersteigt die Größe des Empfangspuffers.
1	char	Es wurde ein Zeichen außerhalb des Bereichs 20h ... 7Fh empfangen.
3	over-flow	Der Puffer ist voll.

**3964R / Modbus RTU/ASCII Master**

Bit	Fehler	Beschreibung
0	over-flow	Das empfangene Telegramm übersteigt die Größe des Empfangspuffers.

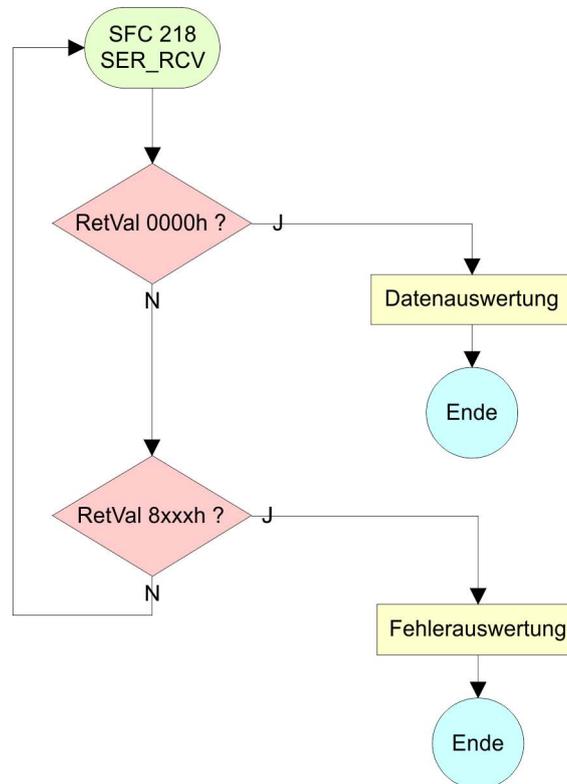
**RETVAL FC/SFC 218  
(Rückgabewert)**

Rückgabewerte, die der Baustein liefert:

Fehler-code	Beschreibung
0000h	kein Fehler
1000h	Empfangspuffer ist zu klein (Datenverlust)
8x24h	Fehler in FC/SFC-Parameter x, mit x: 1: Fehler in <i>DATAPTR</i> 2: Fehler in <i>DATALEN</i> 3: Fehler in <i>ERROR</i>
8122h	Fehler in Parameter <i>DATAPTR</i> (z.B. DB zu kurz)
809Ah	Schnittstelle nicht vorhanden bzw. Schnittstelle wird für PROFIBUS verwendet
809Bh	Schnittstelle ist nicht konfiguriert

**Prinzip der Programmierung**

Nachfolgend sehen Sie die Grundstruktur zur Programmierung eines Receive-Auftrags. Diese Struktur können Sie für alle Protokolle verwenden.

**6.6 Protokolle und Prozeduren****Übersicht**

Die CPU unterstützt folgende Protokolle und Prozeduren:

- ASCII-Übertragung
- STX/ETX
- 3964R
- USS
- Modbus

**ASCII**

Die Datenkommunikation via ASCII ist die einfachste Form der Kommunikation. Die Zeichen werden 1 zu 1 übergeben. Bei ASCII werden je Zyklus mit dem Lese-FC/SFC die zum Zeitpunkt des Aufrufs im Puffer enthaltenen Daten im parametrisierten Empfangsdatenbaustein abgelegt. Ist ein Telegramm über mehrere Zyklen verteilt, so werden die Daten überschrieben. Eine Empfangsbestätigung gibt es nicht. Der Kommunikationsablauf ist vom jeweiligen Anwenderprogramm zu steuern. Einen entsprechenden Receive\_ASCII-FB finden Sie im Service-Bereich unter [www.vipa.com](http://www.vipa.com).

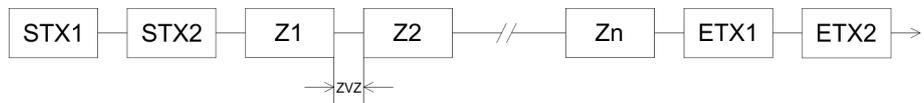
**STX/ETX**

STX/ETX ist ein einfaches Protokoll mit Start- und Ende-Kennung. Hierbei stehen STX für **S**tart of **T**ext und ETX für **E**nd of **T**ext. Die Prozedur STX/ETX wird zur Übertragung von ASCII-Zeichen eingesetzt. Sie arbeitet ohne Blockprüfung (BCC).

- Sollen Daten von der Peripherie eingelesen werden, muss das Start-Zeichen vorhanden sein, anschließend folgen die zu übertragenden Zeichen. Danach muss das Ende-Zeichen vorliegen. Abhängig von der Byte-Breite können folgende ASCII-Zeichen übertragen werden: 5Bit: nicht zulässig; 6Bit: 20...3Fh, 7Bit: 20...7Fh, 8Bit: 20...FFh.
- Die Nutzdaten, d.h. alle Zeichen zwischen Start- und Ende-Kennung, werden nach Empfang des Schlusszeichens an die CPU übergeben.
- Beim Senden der Daten von der CPU an ein Peripheriegerät werden die Nutzdaten an den FC/SFC 217 (SER\_SND) übergeben und von dort mit angefügten Start- und Endezeichen über die serielle Schnittstelle an den Kommunikationspartner übertragen.
- Es kann mit 1, 2 oder keiner Start- und mit 1, 2 oder keiner Ende-Kennung gearbeitet werden.
- Wird kein Ende-Zeichen definiert, so werden alle gelesenen Zeichen nach Ablauf einer parametrierbaren Zeichenverzugszeit (Timeout) an die CPU übergeben.

Als Start- bzw. Ende-Kennung sind alle Hex-Werte von 00h bis 1Fh zulässig. Zeichen größer 1Fh werden ignoriert und nicht berücksichtigt. In den Nutzdaten sind Zeichen kleiner 20h nicht erlaubt und können zu Fehlern führen. Die Anzahl der Start- und Endezeichen kann unterschiedlich sein (1 Start, 2 Ende bzw. 2 Start, 1 Ende oder andere Kombinationen). Für nicht verwendete Start- und Endezeichen muss in der Hardware-Konfiguration FFh eingetragen werden.

*Telegrammaufbau:*



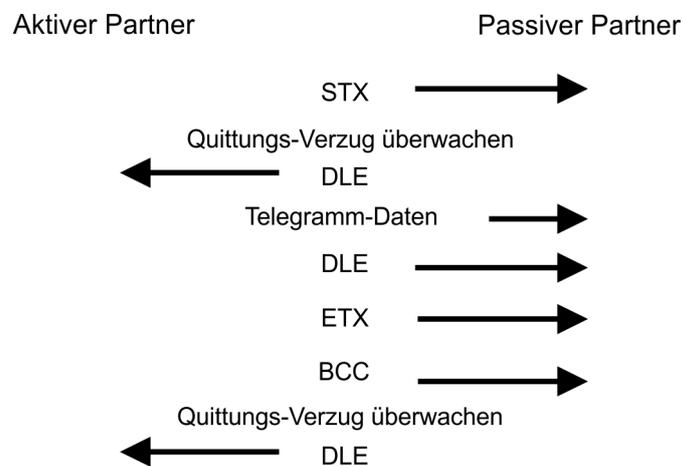
**3964**

Die Prozedur 3964R steuert die Datenübertragung bei einer Punkt-zu-Punkt-Kopplung zwischen der CPU und einem Kommunikationspartner. Die Prozedur fügt bei der Datenübertragung den Nutzdaten Steuerzeichen hinzu. Durch diese Steuerzeichen kann der Kommunikationspartner kontrollieren, ob die Daten vollständig und fehlerfrei bei ihm angekommen sind.

Die Prozedur wertet die folgenden Steuerzeichen aus:

- STX: **S**tart of **T**ext
- DLE: **D**ata **L**ink **E**scape
- ETX: **E**nd of **T**ext
- BCC: **B**lock **C**heck **C**haracter
- NAK: **N**egative **A**cknowledge

Sie können pro Telegramm maximal 255Byte übertragen.

*Prozedurablauf*

Wird ein "DLE" als Informationszeichen übertragen, so wird dieses zur Unterscheidung vom Steuerzeichen "DLE" beim Verbindungsauf- und -abbau auf der Sendeleitung doppelt gesendet (DLE-Verdoppelung). Der Empfänger macht die DLE-Verdoppelung wieder rückgängig.

Unter 3964R muss einem Kommunikationspartner eine niedrigere Priorität zugeordnet sein. Wenn beide Kommunikationspartner gleichzeitig einen Sendeauftrag erteilen, dann stellt der Partner mit niedriger Priorität seinen Sendeauftrag zurück.

**USS**

Das USS-Protokoll (**U**niverselle **s**erielle **S**chnittstelle) ist ein von Siemens definiertes serielles Übertragungsprotokoll für den Bereich der Antriebstechnik. Hiermit lässt sich eine serielle Buskopplung zwischen einem übergeordneten Master - und mehreren Slave-Systemen aufbauen. Das USS-Protokoll ermöglicht durch Vorgabe einer fixen Telegrammlänge einen zeitzyklischen Telegrammverkehr.

Folgende Merkmale zeichnen das USS-Protokoll aus:

- Mehrpunktfähige Kopplung
- Master-Slave Zugriffsverfahren

- Single-Master-System
- Maximal 32 Teilnehmer
- Einfacher, sicherer Telegrammrahmen

Es gilt:

- Am Bus können 1 Master und max. 31 Slaves angebunden sein.
- Die einzelnen Slaves werden vom Master über ein Adresszeichen im Telegramm angewählt.
- Die Kommunikation erfolgt ausschließlich über den Master im Halbduplex-Betrieb.
- Nach einem Senden-Auftrag ist das Quittungstelegramm durch Aufruf des FC/SFC 218 SER\_RCV auszulesen.

Die Telegramme für Senden und Empfangen haben folgenden Aufbau:

**Master-Slave-Telegramm**

STX	LGE	ADR	PKE		IND		PWE		STW		HSW		BCC
02h			H	L	H	L	H	L	H	L	H	L	

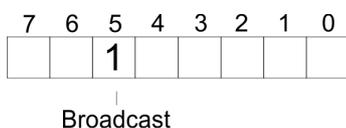
**Slave-Master-Telegramm**

STX	LGE	ADR	PKE		IND		PWE		ZSW		HIW		BCC
02h			H	L	H	L	H	L	H	L	H	L	

mit

- STX - Startzeichen
- STW - Steuerwort
- LGE - Telegrammlänge
- ZSW - Zustandswort
- ADR - Adresse
- HSW - Hauptsollwert
- PKE - Parameterkennung
- HIW - Hauptistwert
- IND - Index
- BCC - Block Check Character
- PWE - Parameterwert

**USS-Broadcast mit gesetztem Bit 5 in ADR-Byte**



Eine Anforderung kann an einen bestimmten Slave gerichtet sein oder als Broadcast-Nachricht an alle Slaves gehen. Zur Kennzeichnung einer Broadcast-Nachricht ist Bit 5 im ADR-Byte auf 1 zu setzen. Hierbei wird die Slave-Adr. (Bit 0 ... 4) ignoriert. Im Gegensatz zu einem "normalen" Senden-Auftrag ist beim Broadcast keine Telegrammauswertung über FC/SFC 218 SER\_RCV erforderlich. Nur Schreibaufträge dürfen als Broadcast gesendet werden.

**Modbus**

- Das Protokoll Modbus ist ein Kommunikationsprotokoll, das eine hierarchische Struktur mit einem Master und mehreren Slaves festlegt.
- Physikalisch arbeitet Modbus über eine serielle Halbduplex-Verbindung. Es treten keine Buskonflikte auf, da der Master immer nur mit einem Slave kommunizieren kann.
- Nach einer Anforderung vom Master wartet dieser solange auf die Antwort des Slaves, bis eine einstellbare Wartezeit abgelaufen ist. Während des Wartens ist eine Kommunikation mit einem anderen Slave nicht möglich.
- Nach einem Sende-Auftrag ist das Quittungstelegramm durch Aufruf des FC/SFC 218 SER\_RCV auszulesen.
- Die Anforderungs-Telegramme, die ein Master sendet und die Antwort-Telegramme eines Slaves haben den gleichen Aufbau:

**Telegrammaufbau**

Startzeichen	Slave-Adresse	Funktions-Code	Daten	Flusskontrolle	Endezeichen
--------------	---------------	----------------	-------	----------------	-------------

**Broadcast mit Slave-Adresse = 0**

- Eine Anforderung kann an einen bestimmten Slave gerichtet sein oder als Broadcast-Nachricht an alle Slaves gehen.
- Zur Kennzeichnung einer Broadcast-Nachricht wird die Slave-Adresse 0 eingetragen.
- Im Gegensatz zu einem "normalen" Send-Auftrag ist beim Broadcast keine Telegrammauswertung über FC/SFC 218 SER\_RCV erforderlich.
- Nur Schreibaufträge dürfen als Broadcast gesendet werden.

**ASCII-, RTU-Modus**

Bei Modbus gibt es zwei unterschiedliche Übertragungsmodi. Die Modus-Wahl erfolgt zur Laufzeit unter Einsatz des FC/SFC 216 SER\_CFG.

- ASCII-Modus: Jedes Byte wird im 2 Zeichen ASCII-Code übertragen. Die Daten werden durch Anfang- und Ende-Zeichen gekennzeichnet. Dies macht die Übertragung transparent aber auch langsam.
- RTU-Modus: Jedes Byte wird als ein Zeichen übertragen. Hierdurch haben Sie einen höheren Datendurchsatz als im ASCII-Modus. Anstelle von Anfang- und Ende-Zeichen wird eine Zeitüberwachung eingesetzt.

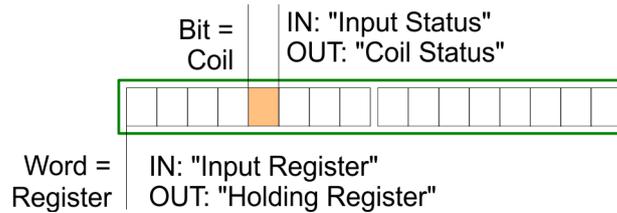
**Unterstützte Modbus-Protokolle**

Die RS485-Schnittstelle unterstützt folgende Modbus-Protokolle:

- Modbus RTU Master
- Modbus ASCII Master

**6.7 Modbus - Funktionscodes****Namenskonventionen**

Für Modbus gibt es Namenskonventionen, die hier kurz aufgeführt sind:



- Modbus unterscheidet zwischen Bit- und Wortzugriff; Bits = "Coils" und Worte = "Register".
- Bit-Eingänge werden als "Input-Status" bezeichnet und Bit-Ausgänge als "Coil-Status".
- Wort-Eingänge werden als "Input-Register" und Wort-Ausgänge als "Holding-Register" bezeichnet.

**Bereichsdefinitionen**

Üblicherweise erfolgt unter Modbus der Zugriff mittels der Bereiche 0x, 1x, 3x und 4x.

Mit 0x und 1x haben Sie Zugriff auf digitale Bit-Bereiche und mit 3x und 4x auf analoge Wort-Bereiche.

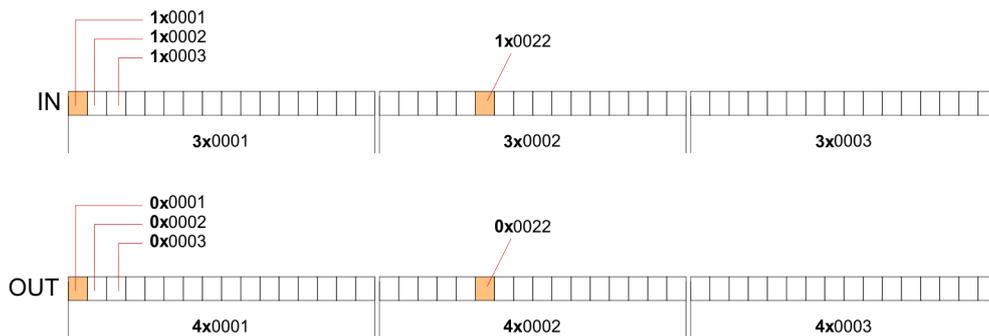
Da aber bei den CPs von VIPA keine Unterscheidung zwischen Digital- und Analogdaten stattfindet, gilt folgende Zuordnung:

0x - Bit-Bereich für Ausgabe-Daten des Masters  
Zugriff über Funktions-Code 01h, 05h, 0Fh

1x - Bit-Bereich für Eingabe-Daten des Masters  
Zugriff über Funktions-Code 02h

3x - Wort-Bereich für Eingabe-Daten des Masters  
Zugriff über Funktions-Code 04h

4x - Wort-Bereich für Ausgabe-Daten des Masters  
Zugriff über Funktions-Code 03h, 06h, 10h



Eine Beschreibung der Funktions-Codes finden Sie auf den Folge-seiten.

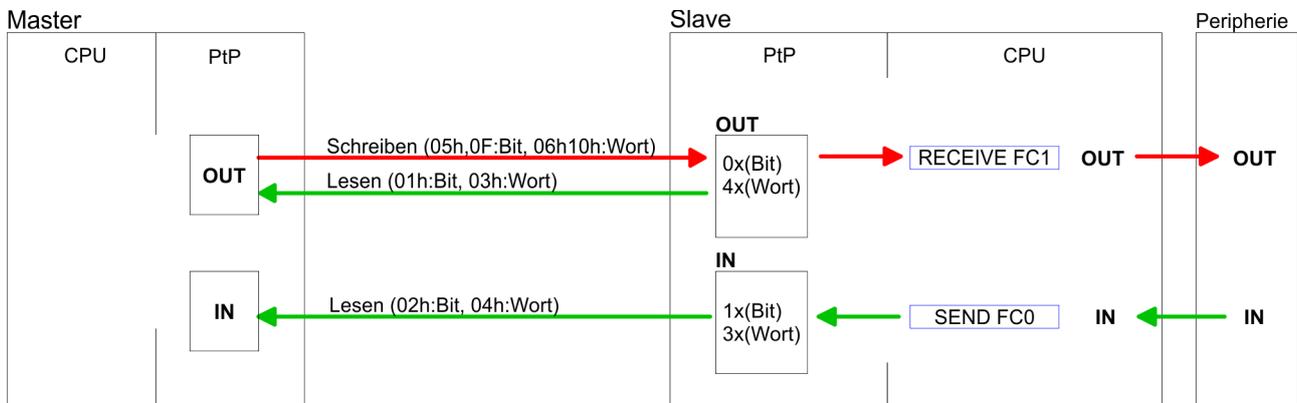
**Übersicht**

Mit folgenden Funktionscodes können Sie von einem Modbus-Master auf einen Slave zugreifen. Die Beschreibung erfolgt immer aus Sicht des Masters:

Code	Befehl	Beschreibung
01h	Read n Bits	n Bit lesen von Master-Ausgabe-Bereich 0x
02h	Read n Bits	n Bit lesen von Master-Eingabe-Bereich 1x
03h	Read n Words	n Worte lesen von Master-Ausgabe-Bereich 4x
04h	Read n Words	n Worte lesen von Master-Eingabe-Bereich 3x
05h	Write 1 Bit	1 Bit schreiben in Master-Ausgabe-Bereich 0x
06h	Write 1 Word	1 Wort schreiben in Master-Ausgabe-Bereich 4x
0Fh	Write n Bits	n Bit schreiben in Master-Ausgabe-Bereich 0x
10h	Write n Words	n Worte schreiben in Master-Ausgabe-Bereich 4x

*Sichtweise für "Eingabe"- und "Ausgabe"-Daten*

Die Beschreibung der Funktionscodes erfolgt immer aus Sicht des Masters. Hierbei werden Daten, die der Master an den Slave schickt, bis zu ihrem Ziel als "Ausgabe"-Daten (OUT) und umgekehrt Daten, die der Master vom Slave empfängt als "Eingabe"-Daten (IN) bezeichnet.

**Antwort des Slaves**

Liefert der Slave einen Fehler zurück, wird der Funktionscode mit 80h "verodert" zurückgesendet.

Ist kein Fehler aufgetreten, wird der Funktionscode zurückgeliefert.

Slave-Antwort:	Funktionscode OR 80h	→ Fehler
	Funktionscode	→ OK

**Byte-Reihenfolge im Wort**

1 Wort	
High-Byte	Low-Byte

**Prüfsumme CRC, RTU, LRC** Die aufgezeigten Prüfsummen CRC bei RTU- und LRC bei ASCII-Modus werden automatisch an jedes Telegramm angehängt. Sie werden nicht im Datenbaustein angezeigt.

**Read n Bits 01h, 02h** Code 01h: n Bit lesen von Master-Ausgabe-Bereich 0x  
Code 02h: n Bit lesen von Master-Eingabe-Bereich 1x

**Kommandotelegramm**

Slave-Adresse	Funktions-Code	Adresse 1. Bit	Anzahl der Bits	Prüfsumme CRC/LRC
1Byte	1Byte	1Wort	1Wort	1Wort

**Antworttelegramm**

Slave-Adresse	Funktions-Code	Anzahl der gelesenen Bytes	Daten 1. Byte	Daten 2. Byte	...	Prüfsumme CRC/LRC
1Byte	1Byte	1Byte	1Byte	1Byte		1Wort
			max. 250Byte			

**Read n Words 03h, 04h** 03h: n Worte lesen von Master-Ausgabe-Bereich 4x  
04h: n Worte lesen von Master-Eingabe-Bereich 3x

**Kommandotelegramm**

Slave-Adresse	Funktions-Code	Adresse 1.Bit	Anzahl der Worte	Prüfsumme CRC/LRC
1Byte	1Byte	1Wort	1Wort	1Wort

**Antworttelegramm**

Slave-Adresse	Funktions-Code	Anzahl der gelesenen Bytes	Daten 1. Wort	Daten 2. Wort	...	Prüfsumme CRC/LRC
1Byte	1Byte	1Byte	1Wort	1Wort		1Wort
			max. 125Worte			

**Write 1 Bit 05h**

Code 05h: 1 Bit schreiben in Master-Ausgabe-Bereich 0x

Eine Zustandsänderung erfolgt unter "Zustand Bit" mit folgenden Werten:

"Zustand Bit" = 0000h → Bit = 0

"Zustand Bit" = FF00h → Bit = 1

**Kommandotelegramm**

Slave-Adresse	Funktions-Code	Adresse Bit	Zustand Bit	Prüfsumme CRC/LRC
1Byte	1Byte	1Wort	1Wort	1Wort

**Antworttelegramm**

Slave-Adresse	Funktions-Code	Adresse Bit	Zustand Bit	Prüfsumme CRC/LRC
1Byte	1Byte	1Wort	1Wort	1Wort

**Write 1 Word 06h**

Code 06h: 1 Wort schreiben in Master-Ausgabe-Bereich 4x

**Kommandotelegramm**

Slave-Adresse	Funktions-Code	Adresse Wort	Wert Wort	Prüfsumme CRC/LRC
1Byte	1Byte	1Wort	1Wort	1Wort

**Antworttelegramm**

Slave-Adresse	Funktions-Code	Adresse Wort	Wert Wort	Prüfsumme CRC/LRC
1Byte	1Byte	1Wort	1Wort	1Wort

**Write n Bits 0Fh**

Code 0Fh: n Bit schreiben in Master-Ausgabe-Bereich 0x Bitte beachten Sie, dass die Anzahl der Bits zusätzlich in Byte anzugeben sind.

**Kommandotelegramm**

Slave-Adresse	Funktions-Code	Adresse 1. Bit	Anzahl der Bits	Anzahl der Bytes	Daten 1. Byte	Daten 2. Byte	...	Prüfsumme CRC/LRC
1Byte	1Byte	1Wort	1Wort	1Byte	1Byte	1Byte	1Byte	1Wort
					max. 250Byte			

**Antworttelegramm**

Slave-Adresse	Funktions-Code	Adresse 1. Bit	Anzahl der Bits	Prüfsumme CRC/LRC
1Byte	1Byte	1Wort	1Wort	1Wort

**Write n Words 10h**

Code 10h: n Worte schreiben in Master-Ausgabe-Bereich

**Kommandotelegramm**

Slave-Adresse	Funktions-Code	Adresse 1. Wort	Anzahl der Worte	Anzahl der Bytes	Daten 1. Wort	Daten 2. Wort	...	Prüfsumme CRC/LRC
1Byte	1Byte	1Wort	1Wort	1Byte	1Wort	1Wort	1Wort	1Wort
					max. 125Worte			

**Antworttelegramm**

Slave-Adresse	Funktions-Code	Adresse 1. Wort	Anzahl der Worte	Prüfsumme CRC/LRC
1Byte	1Byte	1Wort	1Wort	1Wort

**6.8 Modbus - Beispiel zur Kommunikation**

**Übersicht**

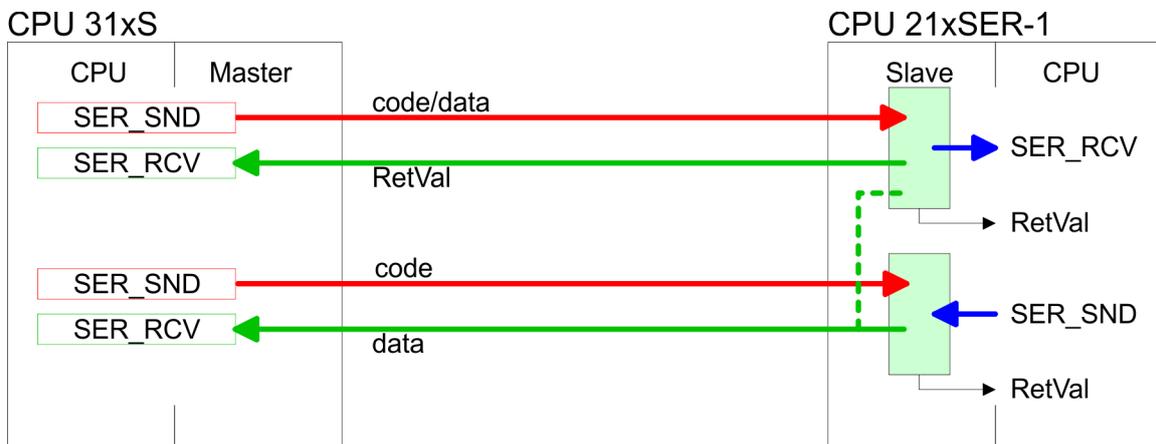
In dem Beispiel wird eine Kommunikation zwischen einem Master und einem Slave über Modbus aufgebaut. Folgende Komponenten sind für das Beispiel erforderlich:

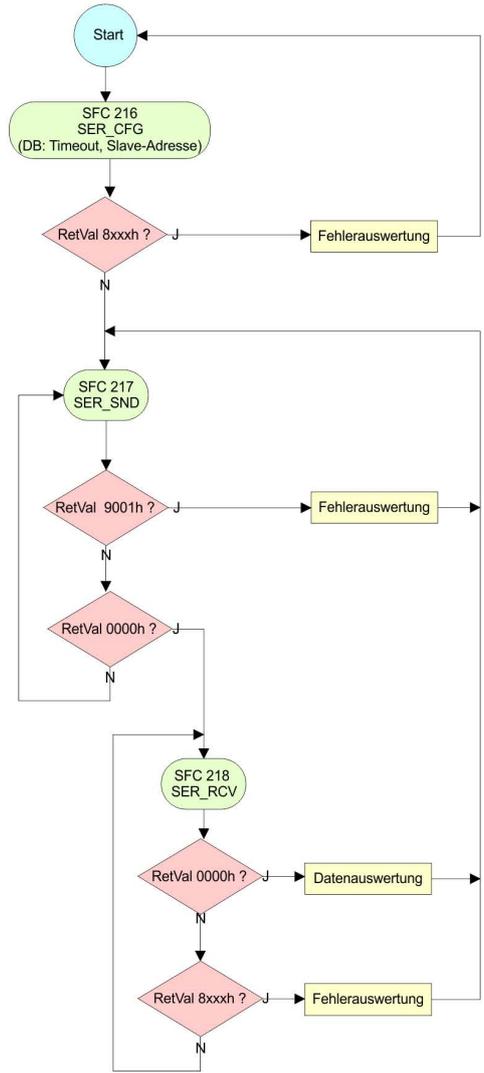
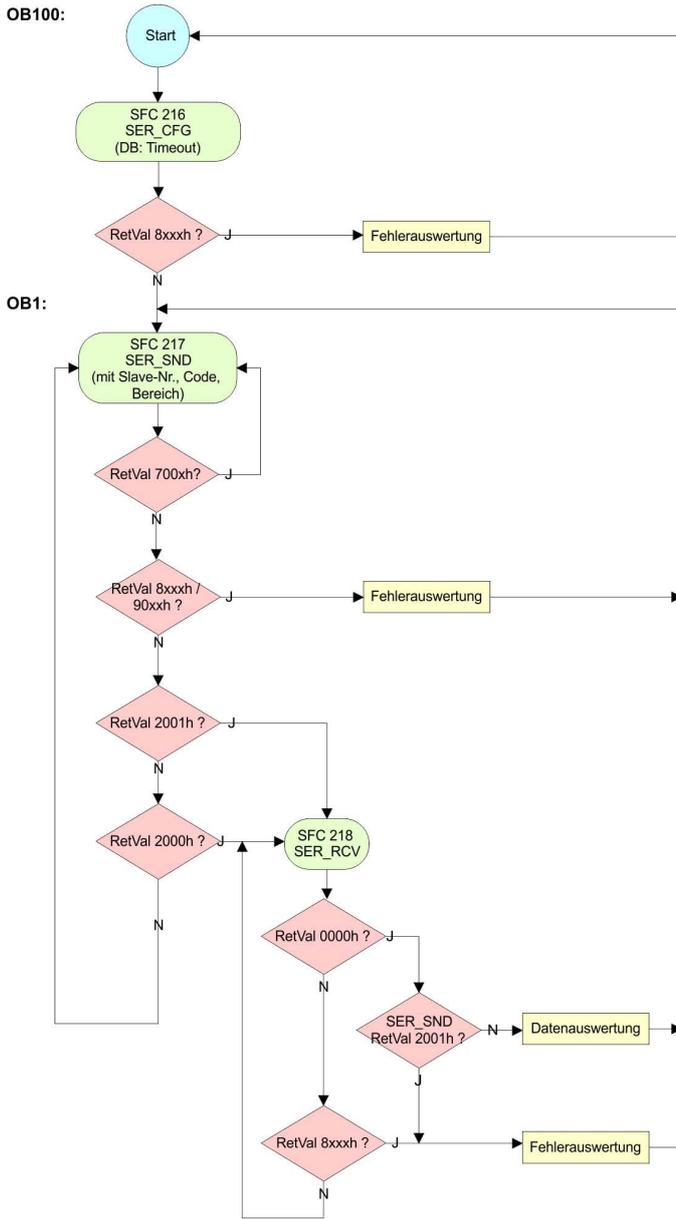
- CPU 31xS als Modbus RTU-Master
- CPU 21xSER-1 als Modbus RTU-Slave
- Siemens SIMATIC Manager und Möglichkeit für Projekttransfer
- Modbus-Kabel-Verbindung

**Vorgehensweise**

1. ► Bauen Sie ein Modbus-System bestehend aus CPU 31xS als Modbus-Master und CPU 21xSER-1 als Modbus-Slave und Modbus-Kabel auf.
2. ► Projektieren Sie die Master-Seite! Erstellen Sie hierzu ein SPS-Anwenderprogramm nach folgender Struktur:
  - OB 100:  
Aufruf SFC 216 (Konfiguration als Modbus RTU-Master) mit Timeout-Angabe und Fehlerauswertung.
  - OB 1:  
Aufruf des SFC 217 (SER\_SND) wobei mit Fehlerauswertung die Daten gesendet werden. Hierbei ist das Telegramm gemäß den Modbus-Vorgaben aufzubauen. Aufruf des SFC 218 (SER\_RECV) wobei mit Fehlerauswertung die Daten empfangen werden.
3. ► Projektieren Sie die Slave-Seite! Das SPS-Anwenderprogramm auf der Slave-Seite sollte folgenden Aufbau haben:
  - OB 100:  
Aufruf SFC 216 (Konfiguration als Modbus RTU-Slave) mit Timeout-Angabe und Modbus-Adresse im DB und Fehlerauswertung
  - OB 1:  
Aufruf des SFC 217 (SER\_SND) für den Datentransport von der Slave-CPU in den Ausgangs-Puffer. Aufruf des SFC 218 (SER\_RECV) für den Datentransport vom Eingangspuffer in die CPU. Für beide Richtungen ist eine entsprechende Fehlerauswertung vorzusehen.

Struktur für die jeweiligen SPS-Programme für Master- und Slave-Seite:





## 7 Einsatz PROFIBUS-Kommunikation

### 7.1 Übersicht

#### PROFIBUS-DP

- PROFIBUS ist ein international offener und serieller Feldbus-Standard für Gebäude-, Fertigungs- und Prozessautomatisierung im unteren (Sensor-/ Aktor-Ebene) bis mittleren Leistungsbereich (Prozessebene).
- PROFIBUS besteht aus einem Sortiment kompatibler Varianten. Die hier angeführten Angaben beziehen sich auf den PROFIBUS-DP.
- PROFIBUS-DP ist besonders geeignet für die Fertigungsautomatisierung. DP ist sehr schnell, bietet "Plug and Play" und ist eine kostengünstige Alternative zur Parallelverkabelung zwischen SPS und dezentraler Peripherie.
- Der Datenaustausch "Data Exchange" erfolgt zyklisch. Während eines Buszyklus liest der Master die Eingangswerte der Slaves und schreibt neue Ausgangsinformationen an die Slaves.

#### CPU mit DP-Master

Der PROFIBUS-DP-Master ist im Hardware-Konfigurator zu projektieren. Hierbei erfolgt die Projektierung über das Submodul X1 (MPI/DP) der Siemens-CPU.

Nach der Übertragung der Daten in die CPU, leitet diese die Projektierdaten intern weiter an den PROFIBUS-Master-Teil.

Während des Hochlaufs blendet der DP-Master automatisch seine Datenbereiche im Adressbereich der CPU ein. Eine Projektierung auf CPU-Seite ist hierzu nicht erforderlich.

#### Einsatz CPU mit DP-Master

Über den PROFIBUS-DP-Master können PROFIBUS-DP-Slaves an die CPU angekoppelt werden. Der DP-Master kommuniziert mit den DP-Slaves und blendet die Datenbereiche im Adressbereich der CPU ein.

Bei jedem NETZ EIN bzw. nach dem URLÖSCHEN holt sich die CPU vom Master die I/O-Mapping-Daten. Bei DP-Slave-Ausfall leuchtet die ER-LED und der OB 86 wird angefordert. Ist dieser nicht vorhanden, geht die CPU in STOP und BASP wird gesetzt. Sobald das BASP-Signal von der CPU kommt, stellt der DP-Master die Ausgänge der angeschlossenen Peripherie auf Null. Unabhängig von der CPU bleibt der DP-Master weiter im RUN.

#### DP-Slave-Betrieb

Für den Einsatz in einem übergeordneten Master-System projektieren Sie zuerst Ihr Slave-System als Siemens-CPU im Slave-Betrieb mit konfigurierten Ein-/Ausgabe-Bereichen. Danach projektieren Sie Ihr Master-System. Binden Sie an das Master-System Ihr Slave-System an, indem Sie die CPU 31x aus dem Hardware-Katalog unter *Bereits projektierte Stationen* auf das Master-System ziehen und Ihr Slave-System auswählen und ankoppeln

### 7.2 Schnelleinstieg

#### Übersicht

Der PROFIBUS-DP-Master ist im Hardware-Konfigurator zu projektieren. Hierbei erfolgt die Projektierung über das Submodul X1 (DP) der Siemens-CPU.

**Schritte der Projektierung**

Die Projektierung des PROFIBUS-DP-Masters sollte nach folgender Vorgehensweise erfolgen:

- **Hardware-Konfiguration - CPU**
- **Einsatz als DP-Master oder Einsatz als DP-Slave**
- **Transfer des Gesamtprojekts in die CPU** ↪ Kapitel 5.10 "Projekt transferieren" auf Seite 57



Um kompatibel mit dem Siemens SIMATIC Manager zu sein, ist die CPU 315-4EC12 von VIPA als

CPU 315-2PN (315-2EH14-0AB00 V3.2)

zu projektieren!

Über das Submodul X1 (DP) projektieren und vernetzen Sie den integrierten PROFIBUS-DP-Master (X3). Den Ethernet-PG/OP-Kanal der CPU 315-4EC12 projektieren Sie immer als 1. Modul nach den reell gesteckten Modulen am Standard-Bus als CP343-1 (343-1EX11) von Siemens.

**7.3 Hardware-Konfiguration - CPU****Voraussetzung**

Die Konfiguration der CPU erfolgt im "Hardware-Konfigurator" von Siemens. Der Hardware-Konfigurator ist Bestandteil des Siemens SIMATIC Managers. Er dient der Projektierung. Bitte verwenden Sie für die Projektierung den Siemens SIMATIC Manager ab V 5.5 SP2. Die Module, die hier projiziert werden können, entnehmen Sie dem Hardware-Katalog, ggf. müssen Sie mit "Extras → Katalog aktualisieren" den Hardware-Katalog aktualisieren.

Für die Projektierung werden fundierte Kenntnisse im Umgang Siemens SIMATIC Manager und dem Hardware-Konfigurator vorausgesetzt!



Bitte beachten Sie, dass diese SPEED7-CPU 4 AKKUs besitzt. Nach einer arithmetischen Operation (+I, -I, \*I, /I, +D, -D, \*D, /D, MOD, +R, -R, \*R, /R) wird der Inhalt des AKKUs 3 und 4 in die AKKUs 2 und 3 geladen. Dies kann bei Programmen, die einen unveränderten AKKU 2 voraussetzen, zu Konflikten führen.

Nähere Informationen hierzu finden Sie im Handbuch "VIPA Operationsliste SPEED7" unter "Unterschiede zwischen SPEED7 und 300V Programmierung".

## Vorgehensweise

Steckpl.	Modul
1	
2	<b>CPU 315-2PN/DP</b>
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 315-2EH14-0AB0 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.

## 7.4 Einsatz als PROFIBUS-DP-Master

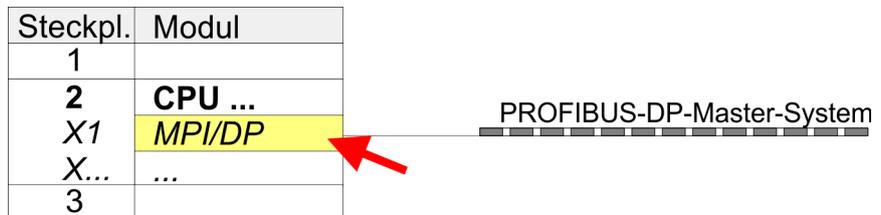
### Voraussetzung

Die zuvor beschriebene Hardware-Konfiguration ist durchgeführt.

### Vorgehensweise

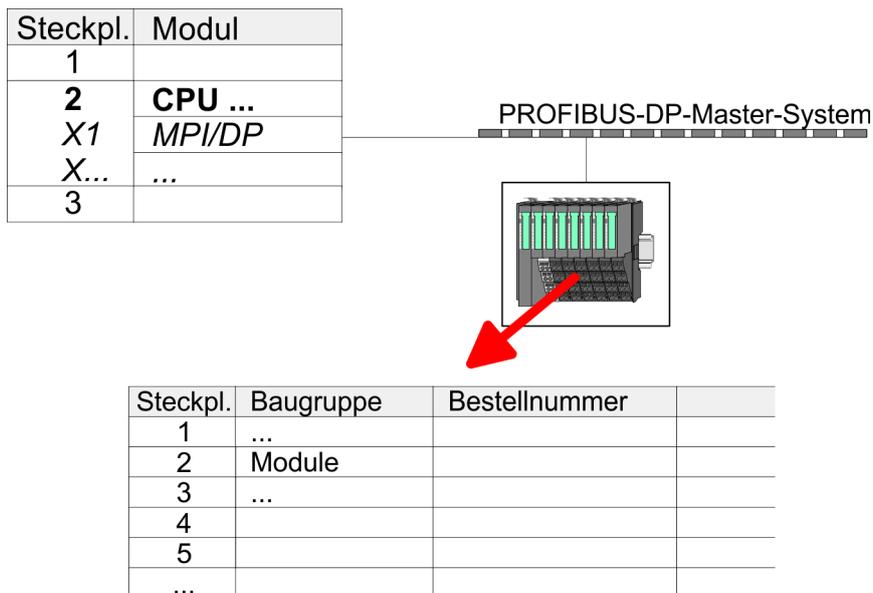
1. ▶ Öffnen Sie den Eigenschaften-Dialog der MPI/DP-Schnittstelle, indem Sie auf "MPI/DP" doppelklicken.
2. ▶ Stellen Sie unter Schnittstelle: Typ "PROFIBUS" ein.
3. ▶ Vernetzen Sie mit PROFIBUS und geben Sie eine Adresse (vorzugsweise 2) vor. Schließen Sie Ihre Eingabe mit [OK] ab.
4. ▶ Stellen Sie unter Betriebsart "DP-Master" ein und schließen Sie den Dialog mit [OK].

⇒ Ein Master-System wird eingefügt:



Sie haben jetzt ihren PROFIBUS-DP-Master projektiert. Binden Sie nun Ihre DP-Slaves mit Peripherie an Ihren DP-Master an.

1. ▶ Zur Projektierung von PROFIBUS-DP-Slaves entnehmen Sie aus dem Hardwarekatalog den entsprechenden PROFIBUS-DP-Slave und ziehen Sie diesen auf das Subnetz Ihres Masters.
2. ▶ Geben Sie dem DP-Slave eine gültige PROFIBUS-Adresse.
3. ▶ Binden Sie in der gesteckten Reihenfolge die Module Ihres DP-Slave-Systems ein und vergeben Sie die Adressen, die von den Modulen zu verwenden sind.
4. ▶ Parametrieren Sie die Module gegebenenfalls.
5. ▶ Speichern, übersetzen und transferieren Sie Ihr Projekt.



### 7.5 Einsatz als PROFIBUS-DP-Slave

#### Schnelleinstieg

Nachfolgend ist der Einsatz des PROFIBUS-Teils als "intelligenter" DP-Slave an Master-Systemen beschrieben, welche ausschließlich im Siemens SIMATIC Manager projiziert werden können. Folgende Schritte sind hierzu erforderlich:

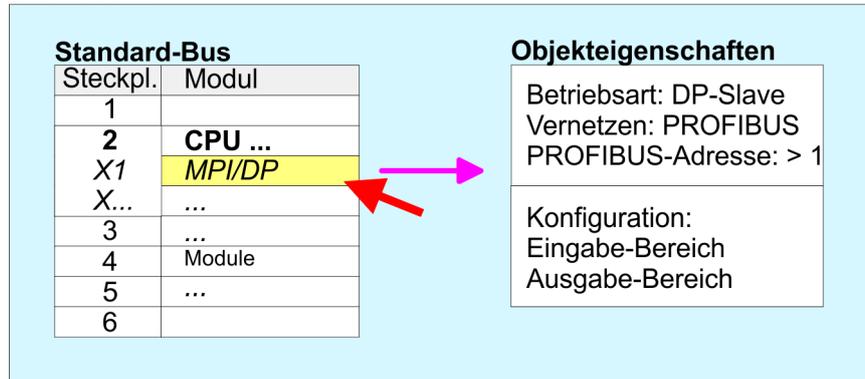
1. ▶ Projektieren Sie eine Station mit einer CPU mit der Betriebsart DP-Slave.
2. ▶ Vernetzen Sie mit PROFIBUS und konfigurieren Sie die Ein-/Ausgabe-Bereiche für die Slave-Seite.
3. ▶ Speichern und übersetzen Sie Ihr Projekt.
4. ▶ Projektieren Sie als weitere Station eine weitere CPU mit der Betriebsart DP-Master.
5. ▶ Vernetzen Sie mit PROFIBUS und konfigurieren Sie die Ein-/Ausgabe-Bereiche für die Master-Seite.
6. ▶ Speichern, übersetzen und transferieren Sie Ihr Projekt in die CPU.

#### Projektierung der Slave-Seite

1. ▶ Starten Sie den Siemens SIMATIC Manager und projektieren Sie eine CPU wie unter "Hardware-Konfiguration - CPU" beschrieben.
2. ▶ Bezeichnen Sie die Station als "...DP-Slave".
3. ▶ Binden Sie gemäß Ihrem Hardwareaufbau Ihre Module ein.
4. ▶ Öffnen Sie den Eigenschaften-Dialog der DP-Schnittstelle der CPU, indem Sie auf "*MPI/DP*" doppelklicken.
5. ▶ Stellen Sie unter Schnittstelle: Typ "PROFIBUS" ein.
6. ▶ Vernetzen Sie mit PROFIBUS und geben Sie eine Adresse (z.B. 3) vor. Schließen Sie Ihre Eingabe mit [OK] ab.
7. ▶ Stellen Sie unter Betriebsart "DP-Slave" ein.

8. ▶ Bestimmen Sie über Konfiguration die Ein-/Ausgabe-Adressbereiche der Slave-CPU, die dem DP-Slave zugeordnet werden sollen.
9. ▶ Speichern, übersetzen und transferieren Sie Ihr Projekt in die CPU.

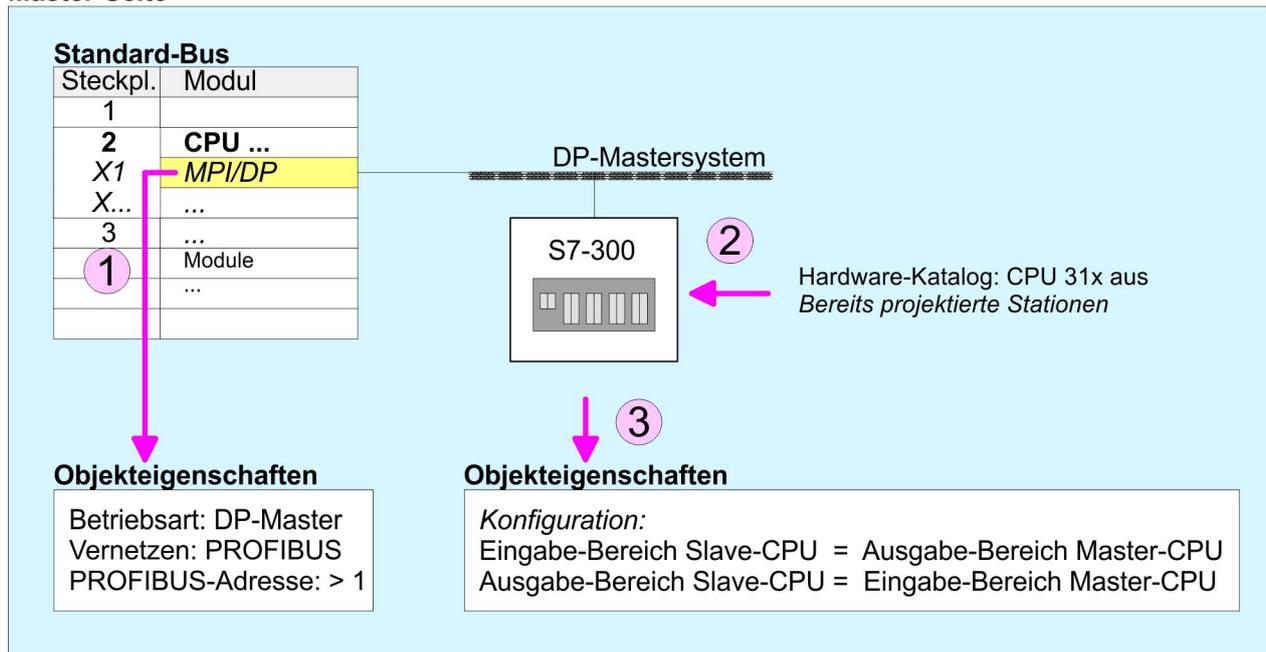
#### Slave-Seite



#### Projektierung der Master-Seite

1. ▶ Fügen Sie eine weitere Station ein und projektieren Sie eine CPU.
2. ▶ Bezeichnen Sie die Station als "...DP-Master".
3. ▶ Binden Sie gemäß Ihrem Hardwareaufbau Ihre Module ein.
4. ▶ Öffnen Sie den Eigenschaften-Dialog der DP-Schnittstelle der CPU, indem Sie auf "MPI/DP" doppelklicken.
5. ▶ Stellen Sie unter *Schnittstelle*: Typ "PROFIBUS" ein.
6. ▶ Vernetzen Sie mit PROFIBUS und geben Sie eine Adresse (z.B. 2) vor. Schließen Sie Ihre Eingabe mit [OK] ab.
7. ▶ Stellen Sie unter Betriebsart "DP-Master" ein und schließen Sie den Dialog mit [OK].
8. ▶ Binden Sie an das Master-System Ihr Slave-System an, indem Sie die "CPU 31x" aus dem Hardware-Katalog unter Bereits projektierte Stationen auf das Master-System ziehen, Ihr Slave-System auswählen und ankoppeln.
9. ▶ Öffnen Sie die *Konfiguration* unter *Objekteigenschaften* Ihres Slave-Systems.
10. ▶ Ordnen Sie durch Doppelklick auf die entsprechende Konfigurationszeile den Slave-Ausgabe-Daten den entsprechenden Eingabe-Adressbereich und den Slave-Eingabe-Daten den entsprechenden Ausgabe-Adressbereich in der Master-CPU zu.
11. ▶ Speichern, übersetzen und transferieren Sie Ihr Projekt in die CPU.

Master-Seite



## 7.6 PROFIBUS-Aufbau Richtlinien

### PROFIBUS allgemein

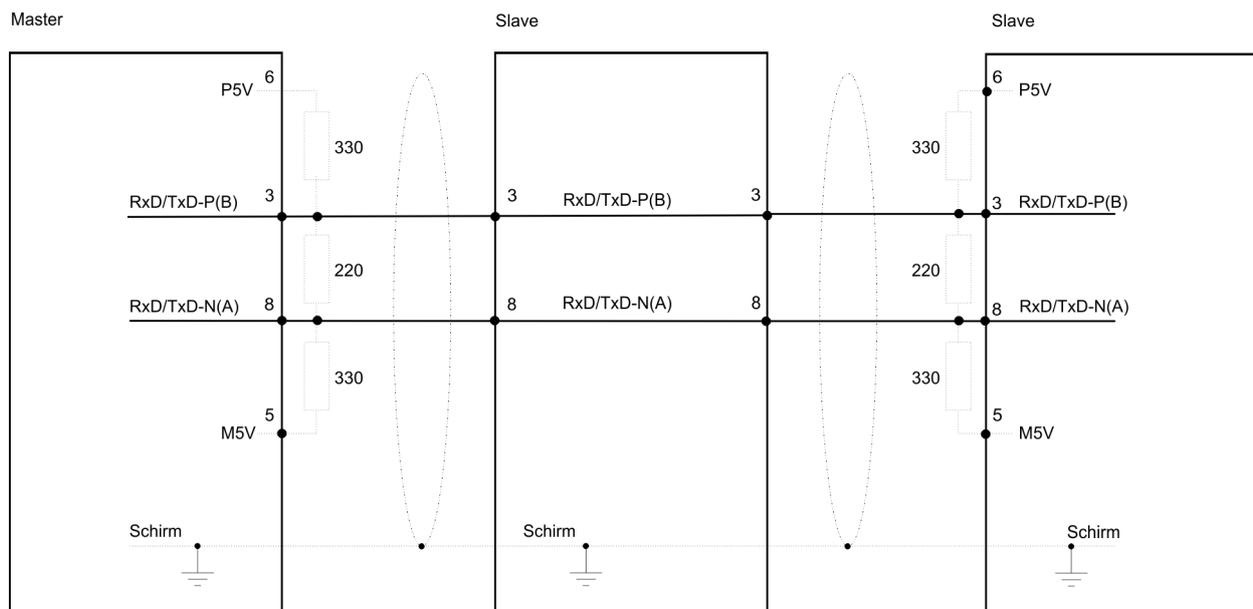
- Ein PROFIBUS-DP-Netz darf nur in Linienstruktur aufgebaut werden.
- PROFIBUS-DP besteht aus mindestens einem Segment mit mindestens einem Master und einem Slave.
- Ein Master ist immer in Verbindung mit einer CPU einzusetzen.
- PROFIBUS unterstützt max. 126 Teilnehmer.
- Pro Segment sind max. 32 Teilnehmer zulässig.
- Die maximale Segmentlänge hängt von der Übertragungsrate ab:  
 9,6 ... 187,5kBit/s → 1000m  
 500kBit/s → 400m  
 1,5MBit/s → 200m  
 3 ... 12MBit/s → 100m
- Maximal 10 Segmente dürfen gebildet werden. Die Segmente werden über Repeater verbunden. Jeder Repeater zählt als Teilnehmer.
- Der Bus bzw. ein Segment ist an beiden Enden abzuschließen.
- Alle Teilnehmer kommunizieren mit der gleichen Übertragungsrate. Die Slaves passen sich automatisch an die Übertragungsrate an.

## Übertragungsmedium

- PROFIBUS verwendet als Übertragungsmedium eine geschirmte, verdrehte Zweidrahtleitung auf Basis der RS485-Schnittstelle.
- Die RS485-Schnittstelle arbeitet mit Spannungsdifferenzen. Sie ist daher unempfindlicher gegenüber Störeinflüssen als eine Spannungs- oder Stromschnittstelle.
- Pro Segment sind maximal 32 Teilnehmer zulässig. Innerhalb eines Segment sind die einzelnen Teilnehmer über Linienstruktur zu verbinden. Die einzelnen Segmente werden über Repeater verbunden. Die max. Segmentlänge ist von der Übertragungsrate abhängig.
- Bei PROFIBUS-DP wird die Übertragungsrate aus dem Bereich zwischen 9,6kBit/s bis 12MBit/s eingestellt, die Slaves passen sich automatisch an. Alle Teilnehmer im Netz kommunizieren mit der gleichen Übertragungsrate.
- Die Busstruktur erlaubt das rückwirkungsfreie Ein- und Auskopeln von Stationen oder die schrittweise Inbetriebnahme des Systems. Spätere Erweiterungen haben keinen Einfluss auf Stationen, die bereits in Betrieb sind. Es wird automatisch erkannt, ob ein Teilnehmer ausgefallen oder neu am Netz ist.

## Busverbindung

In der nachfolgenden Abbildung sind die Abschlusswiderstände der jeweiligen Anfangs- und Endstation stilisiert dargestellt.

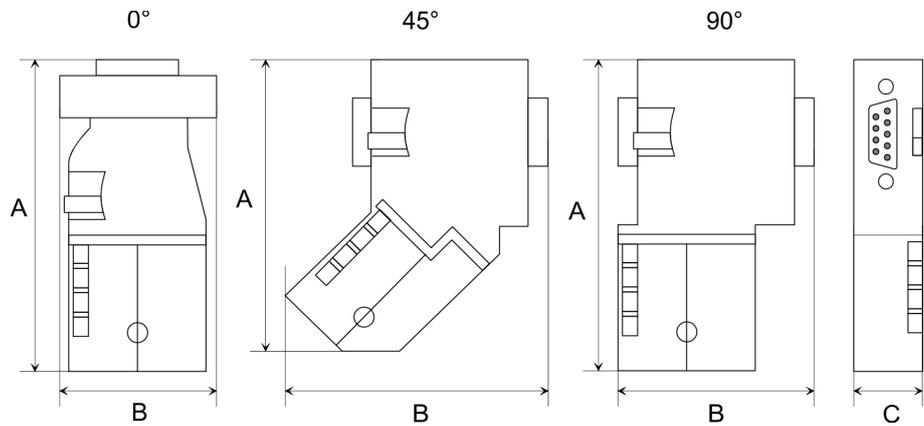


Die PROFIBUS-Leitung muss mit Ihrem Wellenwiderstand abgeschlossen werden. Bitte beachten Sie, dass Sie bei dem jeweiligen letzten Teilnehmer den Bus durch Zuschalten eines Abschlusswiderstands abschließen.

**EasyConn Busanschlusstecker**



In PROFIBUS werden alle Teilnehmer parallel verdrahtet. Hierzu ist das Buskabel durchzuschleifen. Unter der Best.-Nr. 972-0DP10 erhalten Sie von VIPA den Stecker "EasyConn". Dies ist ein Busanschlusstecker mit zuschaltbarem Abschlusswiderstand und integrierter Busdiagnose.



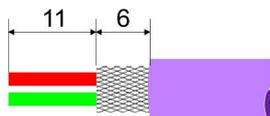
Maße in mm	0°	45°	90°
A	64	61	66
B	34	53	40
C	15,8	15,8	15,8



Zum Anschluss des EasyConn-Steckers verwenden Sie bitte die Standard PROFIBUS-Leitung Typ A (EN50170). Ab Ausgabestand 5 können auch hochflexible Bus-Kabel verwendet werden:

Lapp Kabel Best.-Nr.: 2170222, 2170822, 2170322.

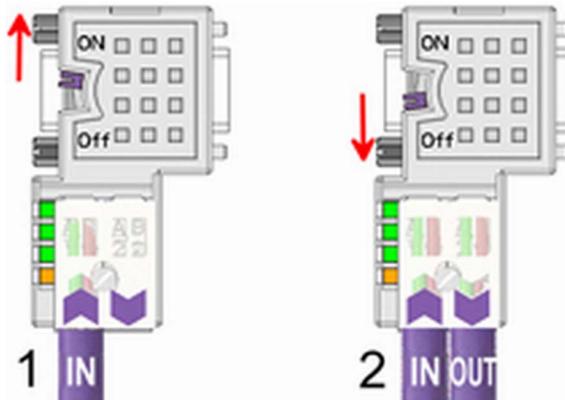
Von VIPA erhalten Sie unter der Best.-Nr. 905-6AA00 das "EasyStrip" Abisolierwerkzeug, das Ihnen den Anschluss des EasyConn-Steckers sehr vereinfacht.



Maße in mm

**Leistungsabschluss mit "EasyConn"**

Auf dem "EasyConn" Busanschlusstecker von VIPA befindet sich unter anderem ein Schalter, mit dem Sie einen Abschlusswiderstand zuschalten können.

**Verdrahtung**

- [1] Einstellung für 1./letzter Bus-Teilnehmer  
 [2] Einstellung für jeden weiteren Busteilnehmer

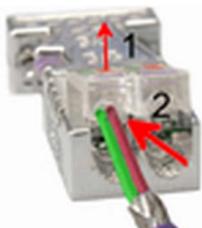
**VORSICHT!**

Der Abschlusswiderstand wird nur wirksam, wenn der Stecker an einem Bus-Teilnehmer gesteckt ist und der Bus-Teilnehmer mit Spannung versorgt wird.

Das Anzugsmoment der Schrauben zur Fixierung des Steckers an einem Teilnehmer darf 0,02Nm nicht überschreiten!



Eine ausführliche Beschreibung zum Anschluss und zum Einsatz der Abschlusswiderstände liegt dem Stecker bei.

**Montage**

1. ➤ Lösen Sie die Schraube.
2. ➤ Klappen Sie die Kontaktabdeckung hoch.
3. ➤ Stecken Sie beide Adern in die dafür vorgesehenen Öffnungen (Farbzuordnung wie unten beachten!).
4. ➤ Bitte beachten Sie, dass zwischen Schirm und Datenleitungen kein Kurzschluss entsteht!
5. ➤ Schließen Sie die Kontaktabdeckung.
6. ➤ Ziehen Sie die Schraube wieder fest (max. Anzugsmoment 0,08Nm).



Den grünen Draht immer an A, den roten immer an B anschließen!

## 7.7 Inbetriebnahme und Anlaufverhalten

### Anlauf im Auslieferungszustand

Im Auslieferungszustand ist die CPU urgelöscht. Nach Netz EIN ist der PROFIBUS-Teil deaktiviert und die LEDs des PROFIBUS-Teils sind ausgeschaltet.

### Online mit Bus-Parametern ohne Slave-Projekt

Über eine Hardware-Konfiguration können Sie den DP-Master mit Busparametern versorgen. Sobald diese übertragen sind geht der DP-Master mit den Bus-Parametern online und zeigt dies über die RUN-LED an. Der DP-Master ist durch Angabe der PROFIBUS-Adresse über PROFIBUS erreichbar. In diesem Zustand können Sie direkt über PROFIBUS Ihre CPU projektieren bzw. Ihr Slave-Projekt übertragen.

### Slave-Projektierung

Sofern der Master gültige Projektierdaten erhalten hat, geht dieser in *Data Exchange* mit den DP-Slaves und zeigt dies über die DE-LED an.

### Zustand CPU beeinflusst DP-Master

Nach NetzeIN bzw. nach der Übertragung einer neuen Hardware-Konfiguration werden automatisch die Projektierdaten und Bus-Parameter an den DP-Master übergeben. Abhängig vom CPU-Zustand zeigt der DP-Master folgendes Verhalten:

### Master-Verhalten bei CPU-STOP

- Der Master sendet an alle angebotenen Slaves das Global Control Kommando "Clear" und zeigt dies durch Blinken der DE-LED an.
- DP-Slaves im *Fail Safe Mode* bekommen die Ausgangstelegrammlänge "0" gesendet.
- DP-Slaves ohne *Fail Safe Mode* bekommen das Ausgangstelegramm in voller Länge aber mit Ausgabewerten=0 gesendet.
- Eingabe-Daten der DP-Slaves werden weiterhin zyklisch im Eingabe-Bereich der CPU abgelegt.

### Master-Verhalten bei CPU-RUN

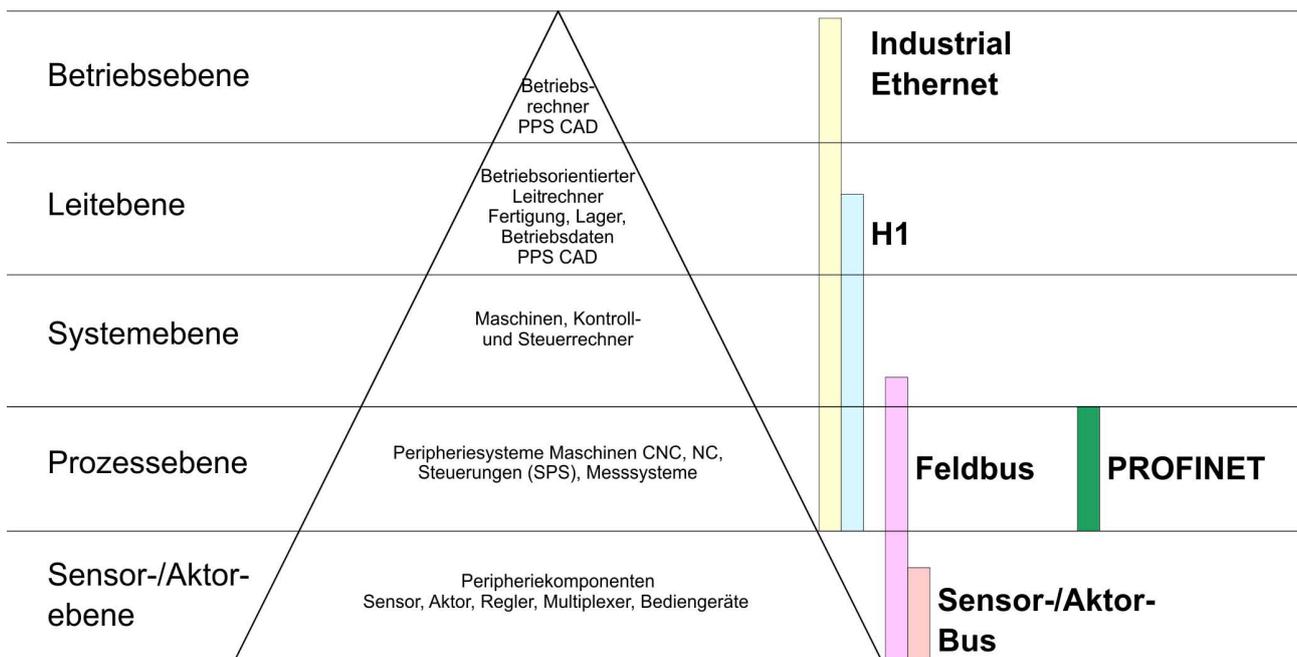
- Der Master sendet an alle angebotenen Slaves das Global Control Kommando "Operate" und zeigt dies durch Leuchten der DE-LED an.
- Alle angebotenen Slaves bekommen zyklisch ein Ausgangstelegramm mit aktuellen Ausgabedaten gesendet.
- Die Eingabe-Daten der DP-Slaves werden zyklisch im Eingabe-Bereich der CPU abgelegt.

## 8 Einsatz Ethernet-Kommunikation - Produktiv

### 8.1 Grundlagen - Industrial Ethernet in der Automatisierung

#### Übersicht

Der Informationsfluss in einem Unternehmen stellt sehr unterschiedliche Anforderungen an die eingesetzten Kommunikationssysteme. Je nach Unternehmensbereich hat ein Bussystem unterschiedlich viele Teilnehmer, es sind unterschiedlich große Datenmengen zu übertragen, die Übertragungsintervalle variieren. Aus diesem Grund greift man je nach Aufgabenstellung auf unterschiedliche Bussysteme zurück, die sich wiederum in verschiedene Klassen einteilen lassen. Eine Zuordnung verschiedener Bussysteme zu den Hierarchieebenen eines Unternehmens zeigt das folgende Modell:



#### Industrial Ethernet

Physikalisch ist Industrial Ethernet ein elektrisches Netz auf Basis einer geschirmten Twisted Pair Verkabelung oder ein optisches Netz auf Basis eines Lichtwellenleiters. Ethernet ist definiert durch den internationalen Standard IEEE 802.3.

Der Netzzugriff bei Industrial Ethernet entspricht dem in der IEEE 802.3 festgelegten CSMA/CD-Verfahren (**C**arrier **S**ense **M**ultiple **A**ccess/**C**ollision **D**etection - Mithören bei Mehrfachzugriff/ Kollisionserkennung):

- Jeder Teilnehmer "hört" ständig die Busleitung ab und empfängt die an ihn adressierten Sendungen.
- Ein Teilnehmer startet eine Sendung nur, wenn die Leitung frei ist.
- Starten zwei Teilnehmer gleichzeitig eine Sendung, so erkennen sie dies, stellen die Sendung ein und starten nach einer Zufallszeit erneut.
- Durch Einsatz von Switches wird eine kollisionsfreie Kommunikation zwischen den Teilnehmern gewährleistet.

## 8.2 Grundlagen - ISO/OSI-Schichtenmodell

### Übersicht

Das ISO/OSI-Schichtenmodell basiert auf einem Vorschlag, der von der International Standards Organization (ISO) entwickelt wurde. Es stellt den ersten Schritt zur internationalen Standardisierung der verschiedenen Protokolle dar. Das Modell trägt den Namen ISO-OSI-Schichtenmodell. OSI steht für **O**pen **S**ystem **I**nterconnection, die Kommunikation offener Systeme. Das ISO/OSI-Schichtenmodell ist keine Netzwerkarchitektur, da die genauen Dienste und Protokolle, die in jeder Schicht verwendet werden, nicht festgelegt sind. Sie finden in diesem Modell lediglich Informationen über die Aufgaben, welche die jeweilige Schicht zu erfüllen hat. Jedes offene Kommunikationssystem basiert heutzutage auf dem durch die Norm ISO 7498 beschriebenen ISO/OSI Referenzmodell. Das Referenzmodell strukturiert Kommunikationssysteme in insgesamt 7 Schichten, denen jeweils Teilaufgaben in der Kommunikation zugeordnet sind. Dadurch wird die Komplexität der Kommunikation auf verschiedene Ebenen verteilt und somit eine größere Übersichtlichkeit erreicht.

Folgende Schichten sind definiert:

- Schicht 7 - Application Layer (Anwendung)
- Schicht 6 - Presentation Layer (Darstellung)
- Schicht 5 - Session Layer (Sitzung)
- Schicht 4 - Transport Layer (Transport)
- Schicht 3 - Network Layer (Netzwerk)
- Schicht 2 - Data Link Layer (Sicherheit)
- Schicht 1 - Physical Layer (Bitübertragung)

Je nach Komplexität der geforderten Übertragungsmechanismen kann sich ein Kommunikationssystem auf bestimmte Teilschichten beschränken.

### Schicht 1 - Bitübertragungsschicht (physical layer)

Die Bitübertragungsschicht beschäftigt sich mit der Übertragung von Bits über einen Kommunikationskanal. Allgemein befasst sich diese Schicht mit den mechanischen, elektrischen und prozeduralen Schnittstellen und mit dem physikalischen Übertragungsmedium, das sich unterhalb der Bitübertragungsschicht befindet:

- Wie viel Volt entsprechen einer logischen 0 bzw. 1?
- Wie lange muss die Spannung für ein Bit anliegen?
- Pinbelegung der verwendeten Schnittstelle.

### Schicht 2 - Sicherungsschicht (data link layer)

Diese Schicht hat die Aufgabe, die Übertragung von Bitstrings zwischen zwei Teilnehmern sicherzustellen. Dazu gehören die Erkennung und Behebung bzw. Weitermeldung von Übertragungsfehlern, sowie die Flusskontrolle. Die Sicherungsschicht verwandelt die zu übertragenden Rohdaten in eine Datenreihe. Hier werden Rahmen Grenzen beim Sender eingefügt und beim Empfänger erkannt. Dies wird dadurch erreicht, dass am Anfang und am Ende eines Rahmens spezielle Bitmuster gesetzt werden. In der Sicherungsschicht wird häufig noch eine Flussregelung und eine Fehlererkennung integriert. Die Datensicherungsschicht ist in zwei Unterschichten geteilt, die LLC- und die MAC-Schicht. Die MAC (**M**edia **A**ccess **C**ontrol) ist die untere Schicht und steuert die Art, wie Sender einen einzigen Übertragungskanal gemeinsam nutzen. Die LLC (**L**ogical **L**ink **C**ontrol) ist die obere Schicht und stellt die Verbindung für die Übertragung der Datenrahmen von einem Gerät zum anderen her.

<b>Schicht 3 - Netzwerkschicht (network layer)</b>	Die Netzwerkschicht wird auch Vermittlungsschicht genannt. Die Aufgabe dieser Schicht besteht darin, den Austausch von Binärdaten zwischen nicht direkt miteinander verbundenen Stationen zu steuern. Sie ist für den Ablauf der logischen Verknüpfungen von Schicht 2-Verbindungen zuständig. Dabei unterstützt diese Schicht die Identifizierung der einzelnen Netzwerkadressen und den Auf- bzw. Abbau von logischen Verbindungskanälen. IP basiert auf Schicht 3. Eine weitere Aufgabe der Schicht 3 besteht in der priorisierten Übertragung von Daten und die Fehlerbehandlung von Datenpaketen. IP (Internet Protokoll) basiert auf Schicht 3.
<b>Schicht 4 - Transportschicht (transport layer)</b>	Die Aufgabe der Transportschicht besteht darin, Netzwerkstrukturen mit den Strukturen der höheren Schichten zu verbinden, indem sie Nachrichten der höheren Schichten in Segmente unterteilt und an die Netzwerkschicht weiterleitet. Hierbei wandelt die Transportschicht die Transportadressen in Netzwerkadressen um. Gebräuchliche Transportprotokolle sind: TCP, SPX, NWLink und NetBEUI.
<b>Schicht 5 - Sitzungsschicht (session layer)</b>	Die Sitzungsschicht wird auch Kommunikationssteuerungsschicht genannt. Sie erleichtert die Kommunikation zwischen Service-Anbieter und Requestor durch Aufbau und Erhaltung der Verbindung, wenn das Transportsystem kurzzeitig ausgefallen ist. Auf dieser Ebene können logische Benutzer über mehrere Verbindungen gleichzeitig kommunizieren. Fällt das Transportsystem aus, so ist es die Aufgabe, gegebenenfalls eine neue Verbindung aufzubauen. Darüber hinaus werden in dieser Schicht Methoden zur Steuerung und Synchronisation bereitgestellt.
<b>Schicht 6 - Darstellungsschicht (presentation layer)</b>	Auf dieser Ebene werden die Darstellungsformen der Nachrichten behandelt, da bei verschiedenen Netzsystemen unterschiedliche Darstellungsformen benutzt werden. Die Aufgabe dieser Schicht besteht in der Konvertierung von Daten in ein beiderseitig akzeptiertes Format, damit diese auf den verschiedenen Systemen lesbar sind. Hier werden auch Kompressions-/Dekompressions- und Verschlüsselungs-/ Entschlüsselungsverfahren durchgeführt. Man bezeichnet diese Schicht auch als Dolmetscherdienst. Eine typische Anwendung dieser Schicht ist die Terminalemulation.
<b>Schicht 7 - Anwendungsschicht (application layer)</b>	Die Anwendungsschicht stellt sich als Bindeglied zwischen der eigentlichen Benutzeranwendung und dem Netzwerk dar. Sowohl die Netzwerk-Services wie Datei-, Druck-, Nachrichten-, Datenbank- und Anwendungs-Service als auch die zugehörigen Regeln gehören in den Aufgabenbereich dieser Schicht. Diese Schicht setzt sich aus einer Reihe von Protokollen zusammen, die entsprechend den wachsenden Anforderungen der Benutzer ständig erweitert werden.

### 8.3 Grundlagen - Begriffe

#### Netzwerk (LAN)

Ein Netzwerk bzw. LAN (Local Area Network) verbindet verschiedene Netzwerkstationen so, dass diese miteinander kommunizieren können. Netzwerkstationen können PCs, IPCs, TCP/IP-Baugruppen, etc. sein. Die Netzwerkstationen sind, durch einen Mindestabstand getrennt, mit dem Netzkabel verbunden. Die Netzwerkstationen und das Netzkabel zusammen bilden ein Gesamtsegment. Alle Segmente eines Netzwerks bilden das Ethernet (Physik eines Netzwerks).

- Twisted Pair** Früher gab es das Triaxial- (Yellow Cable) oder Thin Ethernet-Kabel (Cheapernet). Mittlerweile hat sich aber aufgrund der Störfestigkeit das Twisted Pair Netzwerkkabel durchgesetzt. Die CPU hat einen Twisted-Pair-Anschluss. Das Twisted Pair Kabel besteht aus 8 Adern, die paarweise miteinander verdreht sind. Aufgrund der Verdrehung ist dieses System nicht so stör anfällig wie frühere Koaxialnetze. Verwenden Sie für die Vernetzung Twisted Pair Kabel, die mindestens der Kategorie 5 entsprechen. Abweichend von den beiden Ethernet-Koaxialnetzen, die auf einer Bus-Topologie aufbauen, bildet Twisted Pair ein Punkt-zu-Punkt-Kabelschema. Das hiermit aufzubauende Netz stellt eine Stern-Topologie dar. Jede Station ist einzeln direkt mit dem Sternkoppler (Hub/Switch) zu einem Ethernet verbunden.
- Hub (Repeater)** Ein Hub ist ein zentrales Element zur Realisierung von Ethernet auf Twisted Pair. Seine Aufgabe ist dabei, die Signale in beide Richtungen zu regenerieren und zu verstärken. Gleichzeitig muss er in der Lage sein, segmentübergreifende Kollisionen zu erkennen, zu verarbeiten und weiter zu geben. Er kann nicht im Sinne einer eigenen Netzwerkadresse angesprochen werden, da er von den angeschlossenen Stationen nicht registriert wird. Er bietet Möglichkeiten zum Anschluss an Ethernet oder zu einem anderen Hub bzw. Switch.
- Switch** Ein Switch ist ebenfalls ein zentrales Element zur Realisierung von Ethernet auf Twisted Pair. Mehrere Stationen bzw. Hubs werden über einen Switch verbunden. Diese können dann, ohne das restliche Netzwerk zu belasten, über den Switch miteinander kommunizieren. Eine intelligente Hardware analysiert für jeden Port in einem Switch die eingehenden Telegramme und leitet diese kollisionsfrei direkt an die Zielstationen weiter, die am Switch angeschlossen sind. Ein Switch sorgt für die Optimierung der Bandbreite in jedem einzeln angeschlossenen Segment eines Netzes. Switches ermöglichen exklusiv nach Bedarf wechselnde Verbindungen zwischen angeschlossenen Segmenten eines Netzes.

## 8.4 Grundlagen - Protokolle

### Übersicht

In Protokollen ist ein Satz an Vorschriften oder Standards definiert, der es Kommunikationssystemen ermöglicht, Verbindungen herzustellen und Informationen möglichst fehlerfrei auszutauschen. Ein allgemein anerkanntes Protokoll für die Standardisierung der kompletten Kommunikation stellt das ISO/OSI-Schichtenmodell dar.

↳ Kapitel 8.2 "Grundlagen - ISO/OSI-Schichtenmodell" auf Seite 131

Folgende Protokolle kommen im EtherCAT-Master zum Einsatz:

- Siemens S7-Verbindungen
- Offene Kommunikation
  - TCP native gemäß RFC 793
  - ISO on TCP gemäß RFC 1006
  - UDP gemäß RFC 768

### Siemens S7-Verbindungen

Mit der Siemens S7-Kommunikation können Sie auf Basis von Siemens STEP<sup>®</sup>7 größere Datenmengen zwischen SPS-Systemen übertragen. Hierbei sind die Stationen über Ethernet zu verbinden. Voraussetzung für die Siemens S7-Kommunikation ist eine projektierte Verbindungstabelle, in der die Kommunikationsverbindungen definiert werden. Hierzu können Sie beispielsweise NetPro von Siemens verwenden.

## Eigenschaften:

- Eine Kommunikationsverbindung ist durch eine Verbindungs-ID für jeden Kommunikationspartner spezifiziert.
- Die Quittierung der Datenübertragung erfolgt vom Partner auf Schicht 7 des ISO/OSI-Schichtenmodells.
- Zur Datenübertragung auf SPS-Seite sind für Siemens S7-Verbindungen die FB/SFB-VIPA-Hantierungsbausteine zu verwenden.



*Nähere Informationen zum Einsatz der Bausteine finden Sie im Handbuch Operationsliste HB00\_OPL\_SP7 in Kapitel "VIPA-spezifische Bausteine".*

## Offene Kommunikation

Bei der *"Offenen Kommunikation"* erfolgt die Kommunikation über das Anwenderprogramm bei Einsatz von Hantierungsbausteinen. Diese Bausteine sind auch Bestandteil des Siemens SIMATIC Manager. Sie finden diese in der *"Standard Library"* unter *"Communication Blocks"*.

### ■ Verbindungsorientierte Protokolle:

Verbindungsorientierte Protokolle bauen vor der Datenübertragung eine (logische) Verbindung zum Kommunikationspartner auf und bauen diese nach Abschluss der Datenübertragung ggf. wieder ab. Verbindungsorientierte Protokolle werden eingesetzt, wenn es bei der Datenübertragung insbesondere auf Sicherheit ankommt. Auch wird hier die richtige Reihenfolge der empfangenen Pakete gewährleistet. Über eine physikalische Leitung können in der Regel mehrere logische Verbindungen bestehen. Bei den FBs zur Offenen Kommunikation über Industrial Ethernet werden die folgenden verbindungsorientierten Protokolle unterstützt:

#### – TCP native gemäß RFC 793:

Bei der Datenübertragung über TCP nativ werden weder Informationen zur Länge noch über Anfang und Ende einer Nachricht übertragen. Auch besteht keine Möglichkeit zu erkennen, wo ein Datenstrom endet und der nächste beginnt. Die Übertragung ist stream-orientiert. Aus diesem Grund sollten Sie in den FBs bei Sender und Empfänger identische Datenlängen angeben. Falls die empfangene Anzahl der Daten von der parametrisierten Länge abweicht, erhalten Sie entweder Daten, welche nicht die vollständigen Telegrammdaten enthalten oder mit dem Inhalt eines nachfolgenden Telegramms aufgefüllt sind.

#### – ISO on TCP gemäß RFC 1006:

Bei der Datenübertragung werden Informationen zur Länge und zum Ende einer Nachricht übertragen. Die Übertragung ist blockorientiert. Falls Sie die Länge der zu empfangenden Daten größer gewählt haben als die Länge der gesendeten Daten, kopiert der Empfangsbaustein die gesendeten Daten vollständig in den Empfangsdatenbereich.

### ■ Verbindungslose Protokolle:

Bei den verbindungslosen Protokollen entfallen Verbindungsauf- und Verbindungsabbau zum remoten Partner. Verbindungslose Protokolle übertragen die Daten unquittiert und damit ungesichert zum remoten Partner.

#### – UDP gemäß RFC 768:

Bei Aufruf des Sendebausteins ist ein Verweis auf die Adressparameter des Empfängers (IP-Adresse und Port-Nr.) anzugeben. Auch werden Informationen zur Länge und zum Ende einer Nachricht übertragen. Analog erhalten Sie nach Abschluss des Empfangsbausteins einen Verweis auf die Adressparameter des Senders (IP-Adresse und Port-Nr.). Damit sie Sende- und Empfangsbaustein nutzen können, müssen Sie zuvor sowohl auf der Sender- als auch auf der Empfängerseite einen lokalen Kommunikationszugangspunkt einrichten. Bei jedem Sendauftrag können Sie den remoten Partner durch Angabe seiner IP-Adresse und seiner Port-Nr. neu referenzieren.

## 8.5 Grundlagen - IP-Adresse und Subnetz

### Aufbau IP-Adresse

Unterstützt wird ausschließlich IPv4. Unter IPv4 ist die IP-Adresse eine 32-Bit-Adresse, die innerhalb des Netzes eindeutig sein muss und sich aus 4 Zahlen zusammensetzt, die jeweils durch einen Punkt getrennt sind. Jede IP-Adresse besteht aus einer *Net-ID* und *Host-ID* und hat folgenden

Aufbau: **XXX . XXX . XXX . XXX**

Wertebereich: 000.000.000.000 bis 255.255.255.255

### Net-ID, Host-ID

Die **Network-ID** kennzeichnet ein Netz bzw. einen Netzbetreiber, der das Netz administriert. Über die Host-ID werden Netzverbindungen eines Teilnehmers (Hosts) zu diesem Netz gekennzeichnet.

### Subnetz-Maske

Die Host-ID kann mittels bitweiser UND-Verknüpfung mit der *Subnetz-Maske* weiter aufgeteilt werden, in eine *Subnet-ID* und eine neue *Host-ID*. Derjenige Bereich der ursprünglichen *Host-ID*, welcher von Einsen der Subnetz-Maske überstrichen wird, wird zur *Subnet-ID*, der Rest ist die neue *Host-ID*.

Subnetz-Maske	binär alle "1"		binär alle "0"
IPv4 Adresse	Net-ID	Host-ID	
Subnetz-Maske und IPv4 Adresse	Net-ID	Subnet-ID	neue Host-ID

### Adresse bei Erstinbetriebnahme

Bei der Erstinbetriebnahme der CPU besitzen der Ethernet-PG/OP-Kanal und der EtherCAT-Anschluss keine IP-Adresse.

So weisen Sie dem Ethernet-PG/OP-Kanal IP-Adress-Daten zu  
 ↪ *Kapitel 5.6 "Hardware-Konfiguration - Ethernet-PG/OP-Kanal" auf Seite 45.*

So weisen Sie dem EtherCAT-Anschluss IP-Adress-Daten zu  
 ↪ *"Konfiguration EtherCAT-Master" auf Seite 158.*

### Adress-Klassen

Für IPv4-Adressen gibt es fünf Adressformate (Klasse A bis Klasse E), die alle einheitlich 4 Byte = 32Bit lang sind.

Klasse A	0	Network-ID (1+7bit)	Host-ID (24bit)
Klasse B	10	Network-ID (2+14bit)	Host-ID (16bit)
Klasse C	110	Network-ID (3+21bit)	Host-ID (8bit)
Klasse D	1110	Multicast Gruppe	
Klasse E	11110	Reserviert	

Die Klassen A, B und C werden für Individualadressen genutzt, die Klasse D für Multicast-Adressen und die Klasse E ist für besondere Zwecke reserviert. Die Adressformate der 3 Klassen A,B,C unterscheiden sich lediglich dadurch, dass Network-ID und Host-ID verschieden lang sind.

**Private IP Netze**

Diese Adressen können von mehreren Organisationen als Netz-ID gemeinsam benutzt werden, ohne dass Konflikte auftreten, da diese IP-Adressen weder im Internet vergeben noch ins Internet geroutet werden. Zur Bildung privater IP-Netze sind gemäß RFC1597/1918 folgende Adressbereiche vorgesehen:

Netzwerk Klasse	von IP	bis IP	Standard Subnetz-Maske
A	10.0.0.0	10.255.255.255	255.0.0.0
B	172.16.0.0	172.31.255.255	255.255.0.0
C	192.168.0.0	192.168.255.255	255.255.255.0

(Die Host-ID ist jeweils unterstrichen.)

**Reservierte Host-IDs**

Einige Host-IDs sind für spezielle Zwecke reserviert.

Host-ID = "0"	Identifiziert dieses Netzwerks, reserviert!
Host-ID = maximal (binär komplett "1")	Broadcast-Adresse dieses Netzwerks



*Wählen Sie niemals eine IP-Adresse mit Host-ID=0 oder Host-ID=maximal! (z.B. ist für Klasse B mit Subnetz-Maske = 255.255.0.0 die "172.16.0.0" reserviert und die "172.16.255.255" als lokale Broadcast-Adresse dieses Netzes belegt.)*

**8.6 Schnelleinstieg**

**Übersicht**

Bei der Erstinbetriebnahme bzw. nach dem Umrücken mit erneutem PowerON der CPU besitzen der Ethernet PG/OP-Kanal und der EtherCAT-Master keine IP-Adresse. Diese sind lediglich über ihre MAC-Adresse erreichbar. Mittels der MAC-Adressen, die sich auf Aufkleber unterhalb der Frontklappe befinden in der Reihenfolge Adresse PG/OP-Kanal und darunter Adresse EtherCAT-Master, können Sie der entsprechenden Komponente IP-Adress-Daten zuweisen. Die Zuweisung erfolgt hier direkt über die Hardware-Konfiguration im Siemens SIMATIC Manager.

**Schritte der Projektierung**

Die Projektierung des EtherCAT-Master für Produktiv-Verbindungen sollte nach folgender Vorgehensweise erfolgen:

- Montage und Inbetriebnahme
- Hardware-Konfiguration - CPU

- Verbindungen projektieren
  - Siemens S7-Verbindungen  
(Projektierung erfolgt über Siemens NetPro, die Kommunikation über VIPA Handierungsbausteine)
  - Offene Kommunikation  
(Projektierung und Kommunikation erfolgen über Standard-Handierungsbausteine)
- Transfer des Gesamtprojekts in die CPU.



*Um kompatibel mit dem Siemens SIMATIC Manager zu sein, ist die CPU 315-4EC12 von VIPA als CPU 315-2 PN/DP (6ES7 315-2EH14-0AB0 V3.2) zu projektieren!*

*Der EtherCAT-Master ist über das CPU-Submodul X2 (PN-IO) zu parametrieren.*

*Den Ethernet-PG/OP-Kanal der CPU 315-4EC12 projektieren Sie immer als 1. Modul nach den reell gesteckten Modulen am Standard-Bus als CP343-1 (343-1EX11) von Siemens.*

## 8.7 Montage und Inbetriebnahme

### Voraussetzung

Für den Einsatz von Ethernet-Produktiv-Verbindungen mit dem EtherCAT-Master ist eine EtherCAT-Umsetzbaugruppe zu verwenden, welche das Protokoll EoE (**E**thernet **o**ver **E**therCAT) unterstützt.

Informationen, wie Sie die EtherCAT-Umsetzbaugruppe einbinden, finden Sie im Handbuch zu Ihrer Umsetzbaugruppe.

### Montage

1. ► Bauen Sie Ihr System 300S mit Ihrer CPU auf.
2. ► Verdrahten Sie das System, indem Sie die Leitungen für Spannungsversorgung und Signale anschließen.
3. ► Binden Sie ihren EtherCAT-Master mittels der EtherCAT-Schnittstelle in Ihr EtherCAT-Netz ein.
4. ► Binden sie Ihre EtherCAT-Umsetzbaugruppe in EtherCAT ein.
5. ► Verbinden Sie den Ethernet-Anschluss der EtherCAT-Umsetzbaugruppe mit ihrem Ethernet
6. ► Schalten Sie die Spannungsversorgung ein.
  - ⇒ Nach kurzer Hochlaufzeit befindet sich der EtherCAT-Master im Leerlauf. Bei der Erstinbetriebnahme bzw. nach dem Umröschen der CPU besitzt der EtherCAT-Master und der Ethernet-PG/OP-Kanal keine IP-Adresse.



*Die Zuweisung einer IP-Adresse für Ethernet-Produktiv-Verbindungen über den EtherCAT-Master erfolgt über die Hardware-Konfiguration der CPU über das Sub-Modul "PN-IO".*

## 8.8 Hardware-Konfiguration - CPU

### Voraussetzung

Die Konfiguration der CPU erfolgt im *"Hardware-Konfigurator"* von Siemens. Der Hardware-Konfigurator ist Bestandteil des Siemens SIMATIC Managers. Er dient der Projektierung. Bitte verwenden Sie für die Projektierung den Siemens SIMATIC Manager ab V 5.5 SP2. Die Module, die hier projiziert werden können, entnehmen Sie dem Hardware-Katalog, ggf. müssen Sie mit *"Extras → Katalog aktualisieren"* den Hardware-Katalog aktualisieren.

Für die Projektierung werden fundierte Kenntnisse im Umgang Siemens SIMATIC Manager und dem Hardware-Konfigurator vorausgesetzt!

### Vorgehensweise

Steckpl.	Modul
1	
2	<b>CPU 315-2PN/DP</b>
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 315-2EH14-0AB0 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.

### Parametrierung der IP-Adress-Daten für den EtherCAT-Master

Für den Einsatz von Ethernet-Produktiv-Verbindungen mit dem EtherCAT-Master ist eine EtherCAT-Umsetzbaugruppe zu verwenden, welche das Protokoll EoE (**E**thernet **o**ver **E**therCAT) unterstützt. Informationen, wie Sie die EtherCAT-Umsetzbaugruppe einbinden, finden Sie im Handbuch zu Ihrer Umsetzbaugruppe. Über die PN-IO-Eigenschaften können Sie IP-Adressdaten für den EtherCAT-Master vergeben. Durch Doppelklick auf die Komponente PN-IO öffnen Sie den Eigenschaften-Dialog.

- Geben Sie unter *"Allgemein"* einen Gerätenamen an.
- Geben Sie *"Eigenschaften"* IP-Adresse, Subnetz-Maske und Gateway an und wählen Sie das gewünschte Subnetz aus.

## 8.9 Siemens S7-Verbindungen projektieren

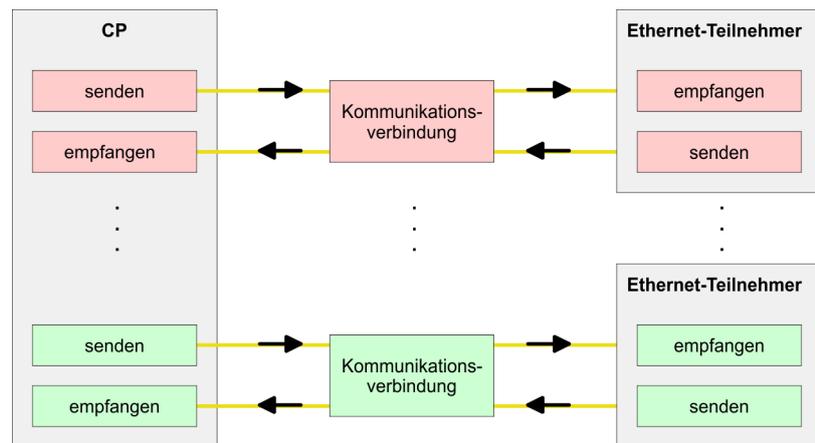
### Übersicht

Die Projektierung von S7-Verbindungen, d.h. die *"Vernetzung"* zwischen den Stationen erfolgt in NetPro von Siemens. NetPro ist eine grafische Benutzeroberfläche zur Vernetzung von Stationen. Eine Kommunikationsverbindung ermöglicht die programmgesteuerte Kommunikation zwischen zwei Teilnehmern am Industrial Ethernet. Die Kommunikationspartner können hierbei im selben Projekt oder - bei Multiprojekten - in den zugehörigen Teilprojekten verteilt angeordnet sein. Kommunikationsverbindungen zu Partnern außerhalb eines Projekts werden über das Objekt *"In unbekanntem Projekt"* oder mittels Stellvertreterobjekten wie *"Andere Stationen"* oder Siemens *"SIMATIC S5 Station"* projiziert. Die Kommunikation steuern Sie durch Einsatz von VIPA Hantierungsbausteinen in Ihrem Anwenderprogramm. Für den Einsatz dieser Bausteine sind immer projizierte Kommunikationsverbindungen auf der aktiven Seite erforderlich.

## Eigenschaften einer Kommunikationsverbindung

Folgende Eigenschaften zeichnen eine Kommunikationsverbindung aus:

- Eine Station führt immer einen aktiven Verbindungsaufbau durch.
- Bidirektionaler Datentransfer (Senden und Empfangen auf einer Verbindung).
- Beide Teilnehmer sind gleichberechtigt, d.h. jeder Teilnehmer kann ereignisabhängig den Sende- bzw. Empfangsvorgang anstoßen.
- Mit Ausnahme der UDP-Verbindung wird bei einer Kommunikationsverbindung die Adresse des Kommunikationspartners über die Projektierung festgelegt. Hierbei ist immer von einer Station der Verbindungsaufbau aktiv durchzuführen.



## Voraussetzung

- Siemens SIMATIC Manager V 5.5 SP2 oder höher und SIMATIC NET sind installiert.
- Bei der Hardware-Konfiguration wurden dem CP über die Eigenschaften von PN-IO IP-Adress-Daten zugewiesen.

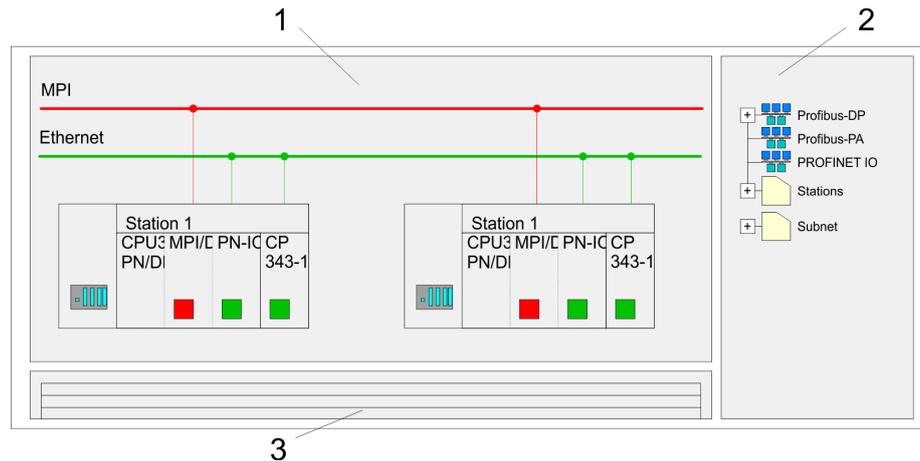


*Alle Stationen außerhalb des aktuellen Projekts müssen mit Stellvertreterobjekten, wie z.B. Siemens "SIMATIC S5" oder "Andere Station" oder mit dem Objekt "In unbekanntem Projekt" projiziert sein. Sie können aber auch beim Anlegen einer Verbindung den Partnertyp "unspezifiziert" anwählen und die erforderlichen Remote-Parameter im Verbindungsdialog direkt angeben.*

## Arbeitsumgebung von NetPro

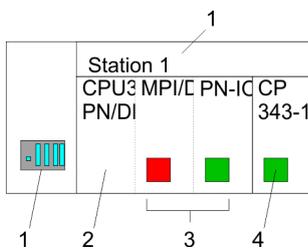
Zur Projektierung von Verbindungen werden fundierte Kenntnisse im Umgang mit NetPro von Siemens vorausgesetzt! Nachfolgend soll lediglich der grundsätzliche Einsatz von NetPro gezeigt werden. Nähere Informationen zu NetPro finden Sie in der zugehörigen Online-Hilfe bzw. Dokumentation. NetPro starten Sie, indem Sie im Siemens SIMATIC Manager auf ein "Netz" klicken oder innerhalb Ihrer CPU auf "Verbindungen".

Die Arbeitsumgebung von NetPro hat folgenden Aufbau:



- 1 **Grafische Netzansicht:** Hier werden alle Stationen und Netzwerke in einer grafischen Ansicht dargestellt. Durch Anwahl der einzelnen Komponenten können Sie auf die jeweiligen Eigenschaften zugreifen und ändern.
- 2 **Netzobjekte:** In diesem Bereich werden alle verfügbaren Netzobjekte in einer Verzeichnisstruktur dargestellt. Durch Ziehen eines gewünschten Objekts in die Netzansicht können Sie weitere Netzobjekte einbinden und im Hardware-Konfigurator öffnen.
- 3 **Verbindungstabelle:** In der Verbindungstabelle sind alle Verbindungen tabellarisch aufgelistet. Diese Liste wird nur eingeblendet, wenn Sie die CPU einer verbindungsfähigen Baugruppe ausgewählt haben. In dieser Tabelle können Sie mit dem gleichnamigen Befehl neue Verbindungen einfügen.

### SPS-Stationen

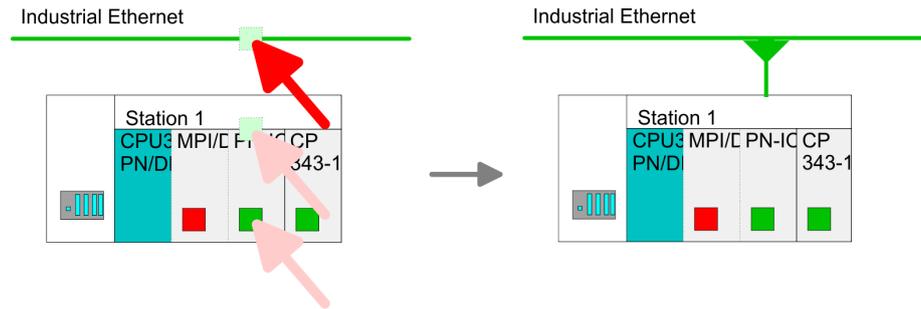


Für jede SPS-Station und ihre Komponente haben Sie folgende grafische Darstellung. Durch Anwahl der einzelnen Komponenten werden Ihnen im Kontext-Menü verschiedene Funktionen zu Verfügung gestellt:

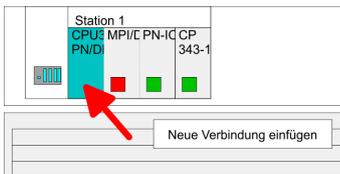
- 1 **Station:** Dies umfasst eine SPS-Station mit Rack, CPU und Kommunikationskomponenten. Über das Kontext-Menü haben Sie die Möglichkeit eine aus den Netzobjekten eingefügte Station im Hardware-Konfigurator mit den entsprechenden Komponenten zu projektieren. Nach der Rückkehr in NetPro werden die neu projektierten Komponenten dargestellt.
- 2 **CPU:** Durch Klick auf die CPU wird die Verbindungstabelle angezeigt. In der Verbindungstabelle sind alle Verbindungen aufgelistet, die für die CPU projektiert sind.
- 3 **Interne Kommunikationskomponenten:** Hier sind die Kommunikationskomponenten aufgeführt, die sich in Ihrer CPU befinden. Der EtherCAT-Master der CPU ist über die Komponente PN-IO zu projektieren.
- 4 **Ethernet-PG/OP-Kanal:** In der Hardware-Konfiguration ist der interne Ethernet-PG/OP-Kanal immer als externer CP zu projektieren. Dieser CP dient ausschließlich der PG/OP-Kommunikation. Produktiv-Verbindungen sind nicht möglich.

### Stationen vernetzen

Stationen vernetzen NetPro bietet Ihnen die Möglichkeit die kommunizierenden Stationen zu vernetzen. Die Vernetzung können Sie über die Eigenschaften in der Hardware-Konfiguration durchführen oder grafisch unter NetPro. Gehen Sie hierzu mit der Maus auf die farbliche Netzmarkierung des entsprechenden CPs und ziehen Sie diese auf das zuzuordnende Netz. Daraufhin wird Ihr CP über eine Linie mit dem gewünschten Netz verbunden



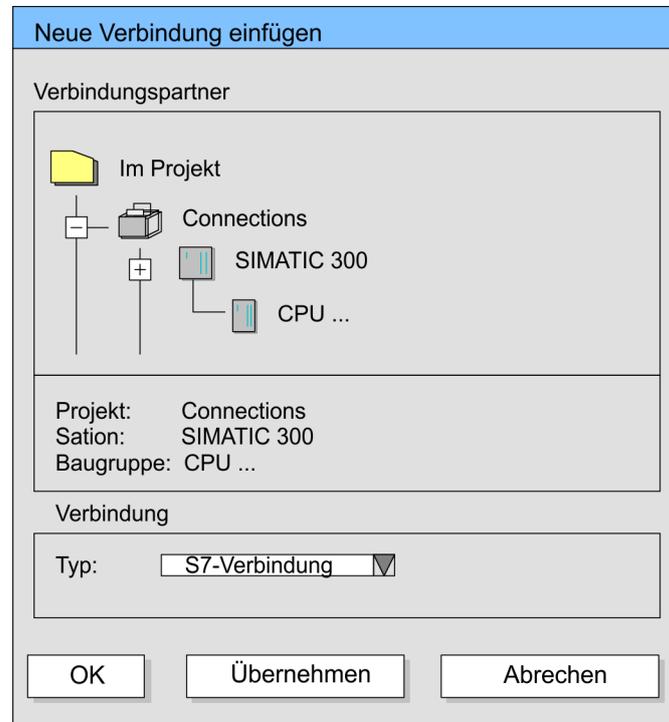
## Verbindungen projektieren



1. Zur Projektierung von Verbindungen blenden Sie die Verbindungsliste ein, indem Sie die entsprechende CPU auswählen. Rufen Sie über das Kontext-Menü *Neue Verbindung einfügen* auf:

- **Verbindungspartner (Station Gegenseite)**  
Es öffnet sich ein Dialogfenster in dem Sie den Verbindungspartner auswählen und den *Verbindungstyp* einstellen können.
- **Spezifizierte Verbindungspartner**  
Jede im Siemens SIMATIC Manager projektierte Station wird in die Liste der Verbindungspartner aufgenommen. Durch Angabe einer IP-Adresse und Subnetz-Maske sind diese Stationen eindeutig *spezifiziert*.
- **Unspezifizierte Verbindungspartner**  
Hier kann sich der Verbindungspartner im aktuellen Projekt oder in einem unbekanntes Projekt befinden. Verbindungsaufträge in ein unbekanntes Projekt sind über einen eindeutigen Verbindungs-Namen zu definieren, der für die Projekte in beiden Stationen zu verwenden ist. Aufgrund dieser Zuordnung bleibt die Verbindung selbst *unspezifiziert*.

2. ▶ Wählen Sie den Verbindungspartner und den Verbindungstyp und klicken Sie auf [OK].
- ⇒ Sofern aktiviert, öffnet sich ein Eigenschaften-Dialog der entsprechenden Verbindung als Bindeglied zu Ihrem SPS-Anwenderprogramm.



3. ▶ Nachdem Sie auf diese Weise alle Verbindungen projiziert haben, können Sie Ihr Projekt "Speichern und übersetzen" und NetPro beenden.

## Verbindungstypen

Mit NetPro können Sie ausschließlich Siemens S7-Verbindungen projektieren.



*Alle Broadcast-Teilnehmer und Alle Multicast-Teilnehmer wird von dieser CPU nicht unterstützt.*

## Siemens S7-Verbindung

- Für Siemens S7-Verbindungen sind für den Datenaustausch die FB/SFB-VIPA-Hantierungsbausteine zu verwenden, deren Gebrauch im Handbuch "Operationsliste" Ihrer CPU näher beschrieben ist.
- Bei Siemens S7-Verbindungen werden Kommunikationsverbindungen durch eine Verbindungs-ID für jeden Kommunikationspartner spezifiziert.
- Eine Verbindung wird durch den lokalen und fernen Verbindungsendpunkt spezifiziert.
- Bei Siemens S7-Verbindungen müssen die verwendeten TSAPs kreuzweise übereinstimmen.

**Folgende Parameter definieren einen Verbindungsendpunkt:**

Station A				Station B
ferner TSAP	→	Siemens	→	lokaler TSAP
lokaler TSAP	←	S7-Verbindung	←	ferner TSAP
ID A				ID B

**Kombinationsmöglichkeiten unter Einsatz der FB/SFB-VIPA-Handierungsbausteine**

Verbindungspartner	Verbindungsaufbau	Verbindung
spezifiziert in NetPro (im aktuellen Projekt)	aktiv/passiv	spezifiziert
unspezifiziert in NetPro (im aktuellen Projekt)	aktiv	spezifiziert
	passiv	unspezifiziert
unspezifiziert in NetPro (in unbekanntem Projekt)	aktiv/passiv	spezifiziert (Verbindungsname in einem anderen Projekt)

Nachfolgend sind alle relevanten Parameter für eine Siemens S7-Verbindung beschrieben:

■ **Lokaler Verbindungsendpunkt:**

Hier können Sie angeben, wie Ihre Verbindung aufgebaut werden soll. Da der Siemens SIMATIC Manager die Kommunikationsmöglichkeiten anhand der Endpunkte identifizieren kann, sind manche Optionen schon vorgelegt und können nicht geändert werden.

– **Aktiver Verbindungsaufbau:**

Für die Datenübertragung muss eine Verbindung aufgebaut sein. Durch Aktivierung der Option Aktiver Verbindungsaufbau übernimmt die lokale Station den Verbindungsaufbau. Bitte beachten Sie, dass nicht jede Station aktiv eine Verbindung aufbauen kann. In diesem Fall hat diese Aufgabe die Gegenstation zu übernehmen.

– **Einseitig:**

Im aktivierten Zustand sind nur einseitige Kommunikationsbausteine wie PUT und GET im Anwenderprogramm der CPU zur Nutzung dieser Verbindung möglich. Hier dient der Verbindungspartner als Server, der weder aktiv senden noch aktiv empfangen kann.

■ **Bausteinparameter**

– **Lokale ID:**

Die ID ist das Bindeglied zu Ihrem SPS-Programm. Die ID muss identisch sein mit der ID in der Aufrufschnittstelle des FB/SFB-VIPA-Hantierungsbausteins.

– **[Vorgabe]:**

Sobald Sie auf [Vorgabe] klicken, wird die ID auf die vom System generierte ID zurückgesetzt.

■ **Verbindungsweg:**

In diesem Teil des Dialogfensters können Sie den Verbindungsweg zwischen der lokalen Station und dem Verbindungspartner einstellen. Abhängig von der Vernetzung der Baugruppen werden Ihnen die möglichen Schnittstellen zur Kommunikation in einer Auswahlliste aufgeführt.

– **[Adressdetails]:**

Über diese Schaltfläche gelangen Sie in das Dialogfeld zur Anzeige und Einstellung der Adressinformationen für den lokalen bzw. den Verbindungspartner.

– **TSAP:**

Bei einer Siemens S7-Verbindung wird der TSAP automatisch generiert aus den Verbindungsressourcen (einseitig/zweiseitig) und Ortsangabe (Rack/Steckplatz bzw. einer systeminternen ID bei PC-Stationen).

– **Verbindungsressource:**

Die Verbindungsressource ist Teil des TSAP der lokalen Station bzw. des Partners. Nicht jede Verbindungsressource ist für jeden Verbindungstyp verwendbar. Je nach Verbindungspartner und -Typ wird bei der Projektierung der Wertebereich eingeschränkt bzw. die Verbindungsressource fest vorgegeben.

**Siemens S7-Verbindung  
- Kommunikationsfunktionen**

Bei den SPEED7-CPUs von VIPA gibt es folgende 2 Möglichkeiten für den Einsatz der Kommunikationsfunktionen:

- **Siemens S7-300-Kommunikationsfunktionen:**  
Durch Einbindung der Funktionsbausteine FB 8 ... FB 15 von VIPA können Sie auf die Siemens S7-300-Kommunikationsfunktionen zugreifen.
- **Siemens S7-400-Kommunikationsfunktionen:**  
Für die Siemens S7-400-Kommunikationsfunktionen verwenden Sie die SFB 8 ... SFB 15, die im Betriebssystem der CPU integriert sind. Hierzu kopieren Sie die Schnittstellenbeschreibung der SFBs aus der Siemens Standard-Bibliothek in das Verzeichnis "Bausteine", generieren für jeden Aufruf einen Instanzen-Datenbaustein und rufen den SFB mit dem zugehörigen Instanzen-Datenbaustein auf.

**Funktionsbausteine**

FB/SFB	Bezeichnung	Beschreibung
FB/SFB 12	BSEND	Blockorientiertes Senden: Mit dem FB/SFB 12 BSEND können Daten an einen remoten Partner-FB/SFB vom Typ BRCV (FB/SFB 13) gesendet werden. Der zu sendende Datenbereich wird segmentiert. Jedes Segment wird einzeln an den Partner gesendet. Das letzte Segment wird vom Partner bereits bei seiner Ankunft quittiert, unabhängig vom zugehörigen Aufruf des FB/SFB BRCV. Aufgrund der Segmentierung können Sie mit einem Sendeauftrag bis zu 65534Byte große Daten übertragen.
FB/SFB 13	BRCV	Blockorientiertes Empfangen: Mit dem FB/SFB 13 BRCV können Daten von einem remoten Partner-FB/SFB vom Typ BSEND (FB/SFB 12) empfangen werden, wobei darauf zu achten ist, dass der Parameter R_ID bei beiden FB/SFBs identisch ist. Nach jedem empfangenen Datensegment wird eine Quittung an den Partner-FB/SFB geschickt, und der Parameter LEN aktualisiert.
FB/SFB 14	GET	Remote CPU lesen: Mit dem FB/SFB 14 GET können Daten aus einer remoten CPU ausgelesen werden, wobei sich die CPU im Betriebszustand RUN oder STOP befinden kann.
FB/SFB 15	PUT	Remote CPU schreiben: Mit dem FB/SFB 15 PUT können Daten in eine remote CPU geschrieben werden, wobei sich die CPU im Betriebszustand RUN oder STOP befinden kann.

**8.10 Offene Kommunikation projektieren****Verbindungsorientierte  
Protokolle**

- Verbindungsorientierte Protokolle bauen vor der Datenübertragung eine (logische) Verbindung zum Kommunikationspartner auf und bauen diese nach Abschluss der Datenübertragung ggf. wieder ab.
- Verbindungsorientierte Protokolle werden eingesetzt, wenn es bei der Datenübertragung insbesondere auf Sicherheit ankommt.

- Die richtige Reihenfolge der empfangenen Pakete ist gewährleistet.
- Über eine physikalische Leitung können in der Regel mehrere logische Verbindungen bestehen.

Bei den FBs zur Offenen Kommunikation über Industrial Ethernet werden die folgenden verbindungsorientierten Protokolle unterstützt:

- *TCP native gemäß RFC 793 (Verbindungstypen 01h und 11h):*
  - Bei der Datenübertragung über TCP nativ werden weder Informationen zur Länge noch über Anfang und Ende einer Nachricht übertragen.
  - Es besteht keine Möglichkeit zu erkennen, wo ein Datenstrom endet und der nächste beginnt.
  - Die Übertragung ist stream-orientiert. Aus diesem Grund sollten Sie in den FBs bei Sender und Empfänger identische Datenlängen angeben.
  - Falls die empfangene Anzahl der Daten von der parametrisierten Länge abweicht, erhalten Sie entweder Daten, welche nicht die vollständigen Telegrammdaten enthalten oder mit dem Inhalt eines nachfolgenden Telegramms aufgefüllt sind. Der Empfangsbaustein kopiert so viele Bytes in den Empfangsbereich, wie Sie als Länge parametrisiert haben. Anschließend setzt er NDR auf TRUE und beschreibt RCVD\_LEN mit dem Wert von LEN. Mit jedem weiteren Aufruf erhalten Sie damit einen weiteren Block der gesendeten Daten.
- *ISO on TCP gemäß RFC 1006:*
  - Bei der Datenübertragung werden Informationen zur Länge und zum Ende einer Nachricht übertragen.
  - Die Übertragung ist blockorientiert.
  - Falls Sie die Länge der zu empfangenden Daten größer gewählt haben als die Länge der gesendeten Daten, kopiert der Empfangsbaustein die gesendeten Daten vollständig in den Empfangsdatenbereich. Anschließend setzt er NDR auf TRUE und beschreibt RCVD\_LEN mit der Länge der gesendeten Daten.
  - Falls Sie die Länge der zu empfangenden Daten kleiner gewählt haben als die Länge der gesendeten Daten, kopiert der Empfangsbaustein keine Daten in den Empfangsdatenbereich, sondern liefert folgende Fehlerinformation: ERROR = 1, STATUS = 8088h.

### **Verbindungsloses Protokoll**

- Bei den verbindungslosen Protokollen entfallen Verbindungsauf- und Verbindungsabbau zum remoten Partner.
- Verbindungslose Protokolle übertragen die Daten unquittiert und damit ungesichert zum remoten Partner.

Bei den FBs zur Offenen Kommunikation über Industrial Ethernet wird das folgende verbindungslose Protokoll unterstützt:

■ *UDP gemäß RFC 768 (Verbindungstyp 13h):*

- Bei Aufruf des Sendebausteins ist ein Verweis auf die Adressparameter des Empfängers (IP-Adresse und Port-Nr.) anzugeben.
- Informationen zur Länge und zum Ende einer Nachricht werden übertragen. Analog erhalten Sie nach Abschluss des Empfangsbausteins einen Verweis auf die Adressparameter des Senders (IP-Adresse und Port-Nr.).
- Damit sie Sende- und Empfangsbaustein nutzen können, müssen Sie zuvor sowohl auf der Sender- als auch auf der Empfängerseite einen lokalen Kommunikationszugangspunkt einrichten.
- Bei jedem Sendauftrag können Sie den remoten Partner durch Angabe seiner IP-Adresse und seiner Port-Nr. neu referenzieren.
- Falls Sie die Länge der zu empfangenden Daten größer gewählt haben als die Länge der gesendeten Daten, kopiert der Empfangsbaustein die gesendeten Daten vollständig in den Empfangsdatenbereich. Anschließend setzt er NDR auf TRUE und beschreibt RCVD\_LEN mit der Länge der gesendeten Daten.
- Falls Sie die Länge der zu empfangenden Daten kleiner gewählt haben als die Länge der gesendeten Daten, kopiert der Empfangsbaustein keine Daten in den Empfangsdatenbereich, sondern liefert folgende Fehlerinformation: ERROR = 1, STATUS = 8088h.

**Hantierungsbausteine**

Die nachfolgend aufgeführten UDTs und FBs dienen der "Offenen Kommunikation" mit anderen Ethernet-fähigen Kommunikationspartnern über Ihr Anwenderprogramm. Diese Bausteine sind Bestandteil des Siemens SIMATIC Manager. Sie finden diese in der "Standard Library" unter "Communication Blocks". Bitte beachten Sie, dass bei Einsatz der Bausteine für offene Kommunikation die Gegenseite nicht zwingend mit diesen Bausteinen projektiert sein muss. Diese kann mit AG\_SEND/AG\_RECEIVE oder mit IP\_CONFIG projektiert sein.

**UDTs**

FB	Bezeichnung	Verbindungsorientierte Protokolle: TCP native gemäß RFC 793, ISO on TCP gemäß RFC 1006	Verbindungsloses Protokoll: UDP gemäß RFC 768
UDT 65	TCON_PAR	Datenstruktur zur Verbindungsparametrierung	Datenstruktur zur Parametrierung des lokalen Kommunikationszugangspunktes
UDT 66	TCON_ADR		Datenstruktur der Adressierungsparameter des remoten Partners

**FBs**

FB	Bezeichnung	Verbindungsorientierte Protokolle: TCP native gemäß RFC 793, ISO on TCP gemäß RFC 1006	Verbindungsloses Protokoll: UDP gemäß RFC 768
FB 63	TSEND	Daten senden	
FB 64	TRCV	Daten empfangen	
FB 65	TCON	Verbindungsaufbau	Einrichtung des lokalen Kommunikationszugangspunktes
FB 66	TDISCON	Verbindungsabbau	Auflösung des lokalen Kommunikationszugangspunktes
FB 67	TUSEND		Daten senden
FB 68	TURCV		Daten empfangen

**8.11 NCM-Diagnose - Hilfe zur Fehlersuche****Siemens NCM S7-Diagnose**

Das Siemens NCM-Diagnosetool, als Bestandteil des Siemens SIMATIC Managers, wird unterstützt. Dieses Tool liefert dynamisch Informationen zum Betriebszustand der Kommunikationsfunktionen von online geschalteten CPs.

Folgende Diagnose-Funktionen stehen Ihnen zur Verfügung:

- Betriebszustand an Ethernet ermitteln
- Im EtherCAT-Master den Diagnosepuffer auslesen
- Verbindungen diagnostizieren



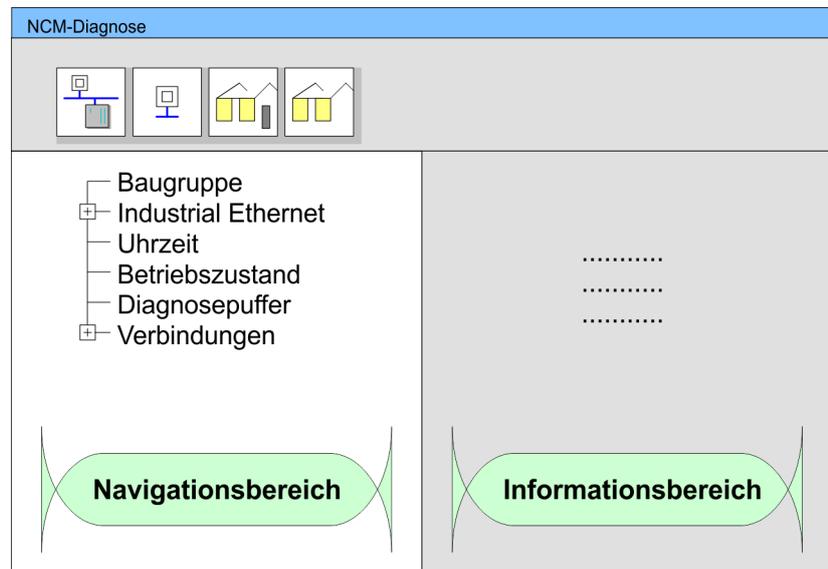
Geben Sie für den EtherCAT-Master als Zielparameter immer Baugruppenträger 0 und Steckplatz 125 an.

Auf den Folgeseiten finden Sie eine Kurzbeschreibung der NCM-Diagnose. Näheres zum Funktionsumfang und zum Einsatz des Siemens NCM-Diagnose-Tools finden Sie in der entsprechenden Online-Hilfe bzw. Dokumentation von Siemens.

## NCM-Diagnose starten

Das Diagnose-Tool starten Sie über "Windows-START-Menü  
→ SIMATIC → ... NCM S7 → Diagnose".

## Aufbau



Die Arbeitsumgebung des Diagnose-Tools hat folgenden Aufbau:

- Im "Navigationsbereich" auf der linken Seite finden Sie die hierarchisch geordneten Diagnoseobjekte. Je nach CP haben Sie eine angepasste Objektstruktur im Navigationsbereich.
- Im "Informationsbereich" auf der rechten Seite finden Sie immer das Ergebnis der von Ihnen angewählten Navigationsfunktion im Navigationsbereich.

## Keine Diagnose ohne Verbindung

Für eine Diagnose ist immer eine Online-Verbindung zu dem zu diagnostizierenden CP erforderlich. Klicken Sie hierzu in der Symbolleiste auf .

Es öffnet sich folgendes Dialogfenster:

Stellen Sie unter Zielstation folgende Parameter ein:

- **Anschluss ...:**  
Ind. Ethernet TCP/IP
- **Teilnehmer-Adr.:**  
Tragen Sie hier die IP-Adresse des CPs ein
- **Baugruppenträger/Steckplatz:**  
Geben Sie hier für den VIPA EtherCAT-Master den Baugruppenträger 0 und für Steckplatz 125 an. Stellen Sie Ihre PG/PC-Schnittstelle auf "TCP/IP -> Netzwerkkarte ...." ein. Mit [OK] starten Sie die Online-Diagnose.

### Diagnosepuffer auslesen

Der EtherCAT-Master besitzt einen Diagnosepuffer. Dieser hat die Architektur eines Ringspeichers. Hier können bis zu 100 Diagnosemeldungen festgehalten werden. In der NCM-Diagnose können Sie über das Diagnoseobjekt Diagnosepuffer die Diagnosemeldungen anzeigen und auswerten. Über einen Doppelklick auf eine Diagnosemeldung hält die NCM-Diagnose weitere Informationen bereit.

### Vorgehensweise bei der Diagnose

Sie führen eine Diagnose aus, indem Sie ein Diagnoseobjekt im Navigationsbereich anklicken. Weitere Funktionen stehen Ihnen über das Menü und über die Symbolleiste zur Verfügung.

Für den gezielten Diagnoseeinsatz ist folgende Vorgehensweise zweckmäßig:

1. ▶ Diagnose aufrufen
2. ▶ Mit  Dialog für Online-Verbindung öffnen, Verbindungsparameter eintragen und mit [OK] Online-Verbindung herstellen.
3. ▶ Den EtherCAT-Master identifizieren und über Baugruppenzustand den aktuellen Zustand des EtherCAT-Masters ermitteln.
4. ▶ Verbindungen überprüfen auf Besonderheiten wie:
  - Verbindungszustand
  - Empfangszustand
  - Sendezustand
5. ▶ Über "*Diagnosepuffer*" den Diagnosepuffer des EtherCAT-Masters einsehen und entsprechend auswerten.
6. ▶ Soweit erforderlich, Projektierung bzw. Programmierung ändern und Diagnose erneut starten.

## 9 Einsatz Ethernet-Kommunikation - EtherCAT

### 9.1 Grundlagen EtherCAT

#### 9.1.1 Allgemeines

Feldbusse haben sich seit vielen Jahren in der Automatisierungstechnik etabliert. Da einerseits die Forderung nach immer höheren Geschwindigkeiten besteht, andererseits bei dieser Technologie die technischen Grenzen bereits erreicht wurden, musste nach neuen Lösungen gesucht werden.

Das aus der Bürowelt bekannte Ethernet ist mit seinen heute überall verfügbaren 100MBit/s sehr schnell. Durch die dort verwendete Art der Verkabelung und den Regeln bei den Zugriffsrechten ist dieses Ethernet nicht echtzeitfähig. Dieser Effekt wurde mit EtherCAT® beseitigt.

#### EtherCAT®

- Für EtherCAT® gilt: EtherCAT® is a registered trademark and patented technology, licensed by Beckhoff Automation GmbH, Germany.
- EtherCAT bedeutet Ethernet for Controller and Automation Technology. Es wurde ursprünglich von der Firma Beckhoff Automation GmbH entwickelt und wird nun von der EtherCAT Technology Group (ETG) unterstützt und weiterentwickelt. Die ETG ist die weltgrößte internationale Anwender- und Herstellervereinigung für Industrial Ethernet.
- EtherCAT ist ein offenes Ethernet-basierendes Feldbus-System, das in der IEC genormt wird.
- EtherCAT erfüllt als offenes Feldbus-System das Anwenderprofil für den Bereich industrieller Echtzeitsysteme.  
Im Gegensatz zur klassischen Ethernet-Kommunikation erfolgt bei EtherCAT der Datenaustausch der I/O-Daten bei 100MBit/s im Vollduplex-Betrieb, während das Telegramm die Koppler durchläuft. Da auf diese Weise ein Telegramm in Sende- und in Empfangsrichtung die Daten vieler Teilnehmer erreicht, besitzt EtherCAT eine Nutzdatenrate von über 90%.
- Das für Prozessdaten optimierte EtherCAT-Protokoll wird direkt im Ethernet-Telegramm transportiert. Dieses wiederum kann aus mehreren Untertelegammen bestehen, die jeweils einen Speicherbereich des Prozessabbilds bedienen.

#### Übertragungsmedium

EtherCAT verwendet als Übertragungsmedium Ethernet. Es kommen Standard-CAT5-Kabel zum Einsatz. Hierbei sind Leitungslängen von bis zu 100m zwischen 2 Teilnehmern möglich.

In einem EtherCAT-Netzwerk dürfen nur EtherCAT-Komponenten verwendet werden. Für die Realisierung von Topologien abweichend von der Linienstruktur sind entsprechende EtherCAT-Komponenten erforderlich, welche dies unterstützen. Der Einsatz von Hubs ist nicht möglich.

#### Kommunikationsprinzip

Bei EtherCAT sendet der Master ein Telegramm an den ersten Teilnehmer. Dieser entnimmt aus dem laufenden Datenstrom die für ihn bestimmten Daten, fügt seine Antwortdaten in das Telegramm ein und sendet das Telegramm weiter zum nächsten Teilnehmer. Dieser verfährt auf die gleiche Weise mit dem Telegramm.

Ist das Telegramm beim letzten Teilnehmer angekommen, stellt dieser fest, dass kein weiterer Teilnehmer angeschlossen ist und sendet das Telegramm zurück an den Master. Hierbei wird das Telegramm über das andere Adernpaar durch alle Teilnehmer zum Master gesendet (Vollduplex). Durch die Steckreihenfolge und die Nutzung der Vollduplex-Technologie stellt EtherCAT einen logischen Ring dar.

**EtherCAT State Machine**

Über die EtherCAT State Machine wird der Zustand des EtherCAT-Kopplers gesteuert.

**Objektverzeichnis (SDOs)**

Im Objektverzeichnis werden alle Parameter-, Diagnose-, Prozess- oder sonstige Daten aufgeführt, die über EtherCAT gelesen oder beschrieben werden können. Über den SDO-Informations-Dienst können Sie auf das Objektverzeichnis zugreifen. Zusätzlich liegt das Objektverzeichnis in der Gerätebeschreibungsdatei ab.

**Prozessdaten (PDOs)**

Der EtherCAT Data Link Layer ist für die schnelle Übertragung von Prozessdaten optimiert. Hier wird festgelegt, wie die Prozessdaten des Gerätes den EtherCAT-Prozessdaten zugeordnet sind und wie die Applikation auf dem Gerät zum EtherCAT-Zyklus synchronisiert ist. Die Zuordnung der Prozessdaten (Mapping) erfolgt über die PDO-Mapping- und die SyncManager-PDO-Assign-Objekte. Diese beschreiben, welche Objekte aus dem Objektverzeichnis als Prozessdaten mit EtherCAT übertragen werden. Über die SyncManager-Communication-Objekte wird festgelegt, mit welcher Zykluszeit die zugehörigen Prozessdaten über EtherCAT übertragen werden und in welcher Form sie für die Übertragung synchronisiert werden.

**Emergencies**

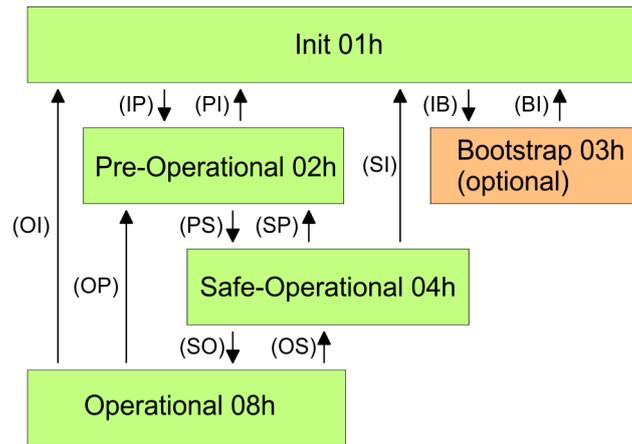
Über Emergencies können Diagnosen, Prozessereignisse und Fehler beim Zustandswechsel der State Machine übertragen werden.

Statusmeldungen dagegen, die den aktuellen Zustand des Gerätes anzeigen, sollten direkt mit den Prozessdaten übertragen werden.

## 9.1.2 EtherCAT Zustandsmaschine

**Zustände**

In jedem EtherCAT-Kommunikationsteilnehmer ist eine *Zustandsmaschine* implementiert. Für jeden Zustand ist definiert, welche Kommunikationsdienste über EtherCAT aktiv sind. Die Zustandsmaschine wird vom EtherCAT-Master gesteuert.



- IP Starte Mailbox-Kommunikation
- PI Stoppe Mailbox-Kommunikation
- PS Starte Input Update
- SP Stoppe Input Update
- SO Starte Output Update
- OS Stoppe Output Update
- OP Stoppe Input Update, stoppe Output Update
- SI Stoppe Input Update, stoppe Mailbox-Kommunikation
- OI Stoppe Output Update, stoppe Input Update, Stoppe Mailbox-Kommunikation
- IB Starte Mailbox für Firmwareupdate im Bootstrap-Mode
- BI Neustart/Stoppe Mailbox

**Init - 01h**

Nach dem Einschalten befindet sich der EtherCAT-Koppler im Zustand *Init*. Dort ist weder Mailbox- noch Prozessdatenkommunikation möglich. Der EtherCAT-Master initialisiert die SyncManager-Kanäle 0 und 1 für die Mailbox-Kommunikation.

**Pre-Operational (Pre-Op) - 02h**

Beim Übergang von *Init* nach *Pre-Op* prüft der EtherCAT-Koppler, ob die Mailbox korrekt initialisiert wurde. Im Zustand *Pre-Op* ist Mailbox-Kommunikation aber keine Prozessdaten-Kommunikation möglich. Der EtherCAT-Master initialisiert die SyncManager-Kanäle für Prozessdaten (ab SyncManager-Kanal 2), die FMMU-Kanäle und das PDO-Mapping bzw. das SyncManager-PDO-Assignment. Weiterhin werden in diesem Zustand die Einstellungen für die Prozessdatenübertragung sowie modulspezifische Parameter übertragen, die von den Defaulteinstellungen abweichen.

**Safe-Operational (Safe-Op) - 04h**

Beim Übergang von *Pre-Op* nach *Safe-Op* prüft der EtherCAT-Koppler, ob die SyncManager-Kanäle für die Prozessdatenkommunikation korrekt sind. Bevor er den Zustandswechsel quittiert, kopiert der EtherCAT-Koppler aktuelle Inputdaten in die entsprechenden DP-RAM-Bereiche des EtherCAT-Koppler-Controllers. Im Zustand *Safe-Op* ist Mailbox- und Prozessdaten-Kommunikation möglich. Hierbei werden die Inputdaten zyklisch aktualisiert aber die Ausgänge sind deaktiviert.

**Operational (Op) - 08h**

Im Zustand *Op* kopiert der EtherCAT-Koppler die Ausgangsdaten des Masters auf seine Ausgänge. Es ist Prozessdaten- und Mailbox-Kommunikation möglich.

**Bootstrap - optional (Boot) - 03h**

Im Zustand *Boot* kann ein Update der EtherCAT-Koppler-Firmware vorgenommen werden. Dieser Zustand ist nur über Init zu erreichen. Im Zustand *Boot* ist Mailbox-Kommunikation über das Protokoll File-Access over EtherCAT (FoE) möglich, aber keine andere Mailbox-Kommunikation und keine Prozessdaten-Kommunikation.

**Zustände 00h und FFh**

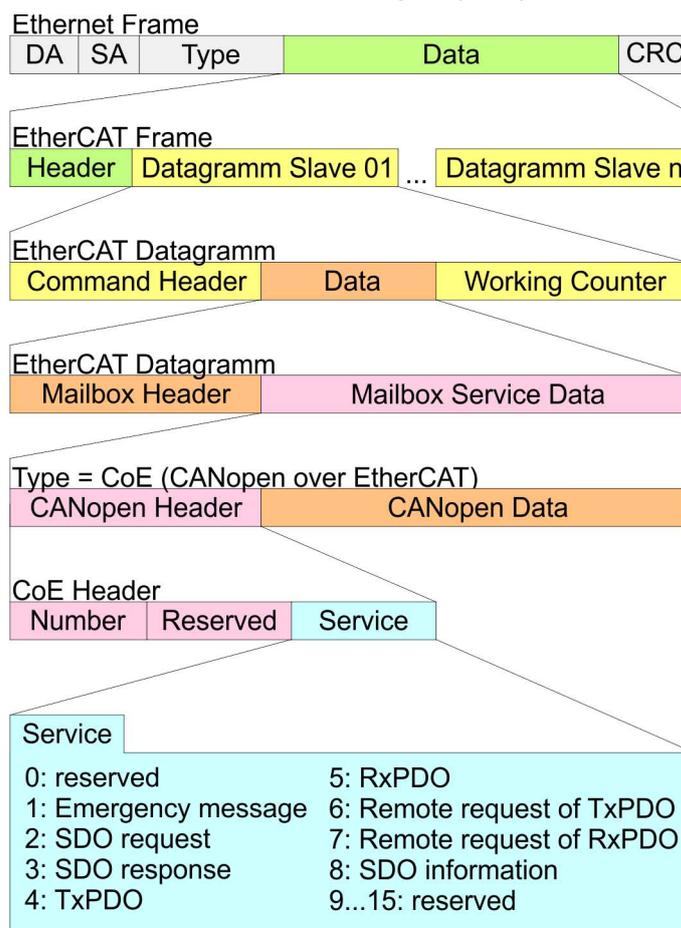
Zusätzlich gibt es noch folgende Zustände:

- 00h: Station ist physikalisch nicht vorhanden
- FFh: Station ist nicht projiziert

**9.1.3 CoE - CANopen over Ethernet**

CoE steht für CANopen over EtherCAT. Jeder intelligente EtherCAT-Koppler (mit Mikrocontroller) unterstützt das CoE-Interface.

Mit CANopen haben Sie eine einheitliche Anwenderschnittstelle, die einen vereinfachten Systemaufbau mit unterschiedlichsten Geräten ermöglicht. Mit CoE können Sie komfortabel auf alle Geräteparameter zugreifen und gleichzeitig Daten einlesen und ausgeben. Echtzeitdaten lesen Sie über PDOs und die Parametrierung führen Sie über SDOs aus. Weiter stehen Ihnen Emergency-Objekte zur Verfügung.



DA Destination address  
 SA Source address  
 CRC Checksum

### 9.1.4 ESI-Dateien

Von VIPA erhalten Sie für den EtherCAT-Koppler ESI-Dateien. Diese Dateien befinden sich entweder auf dem beiliegenden Datenträger oder im Download-Bereich von [www.vipa.com](http://www.vipa.com). Installieren Sie die ESI-Dateien in Ihrem Projektierool. Nähere Hinweise zur Installation der ESI-Dateien finden Sie im Handbuch zu Ihrem Projektierool. Zur Konfiguration in Ihrem Projektierool befinden sich in den ESI-Dateien alle System 300S Module in Form von XML-Daten.

## 9.2 Inbetriebnahme und Anlaufverhalten

### 9.2.1 Montage und Inbetriebnahme

1. ► Bauen Sie Ihr System 300S mit Ihrer CPU auf.
2. ► Verdrahten Sie das System, indem Sie die Leitungen für Spannungsversorgung und Signale anschließen.
3. ► Binden Sie ihren EtherCAT-Master an EtherCAT an.
4. ► Schalten Sie die Spannungsversorgung ein.

### 9.2.2 Anlaufverhalten

#### Bedingungen für den Anlauf

Nach PowerON und dem ANLAUF (inkl. OB100) wird die CPU nach RUN geschaltet. Dies bringt den EtherCAT-Master in den Zustand *Op* und dieser fordert den Zustand *Op* bei den angebotenen EtherCAT Slave-Stationen an. Bevor nun der OB1 aufgerufen wird, wartet die CPU, bis keine EtherCAT Slave-Station mehr im Zustand *SafeOp* befindet. Die *Wartezeit* können Sie über den CPU-Parameter "*Übertragung der Parameter an Baugruppen*" im Eigenschaftsregister "*Anlauf*" vorgeben.

Unter Einsatz des EtherCAT-Masters wird zwischen folgenden Anlaufverhalten unterschieden. Die Bedingungen hierzu können Sie der nachfolgenden Tabelle entnehmen:

#### ■ CPU geht in RUN wenn Topologie OK ist

Die CPU wartet auf alle Slaves, welche zwingend vorhanden sein müssen, maximal bis die *Wartezeit* abgelaufen ist und geht dann in RUN. Die Topologie muss OK sein.

#### ■ CPU geht in RUN unabhängig von Topologie bzw. optionalen Slaves

Die CPU wartet auf alle Slave, welche zwingend vorhanden sein müssen, maximal bis die *Wartezeit* abgelaufen ist und geht dann in RUN unabhängig von Topologie bzw. optionalen Slaves.

Ist der CPU-Parameter: " <i>Anlauf bei Sollausbau ungleich Istausbau</i> " aktiviert?	J	N	N	N	N	N	N	N	N	N
Sind alle erforderlichen Slaves projektiert?	x	J	N	J	J	J	J	N	N	J
Sind optionale Slaves projektiert (Hot-Connect-Gruppe)?	x	N	J	J	x	J	x	N	N	x
Sind alle erforderlichen Slaves vorhanden?	x	J	N	J	x	J	x	x	x	N

Sind optionale Slaves vorhanden (nicht alle müssen vorhanden sein)?	x	N	J	J	x	J	x	x	x	x
Gibt es mindestens einen erforderlichen Slave mit falschem Modul?	x	N	N	N	J	x	x	x	x	x
Gibt es mindestens einen optionalen Slave mit falschem Modul?	x	N	N	N	x	J	x	x	x	x
Ist mindestens ein nicht projektierter Slave vorhanden?	x	N	N	N	x	x	J	J	N	x
	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓
<b>CPU geht in RUN wenn Topologie OK ist.</b>	J									
<b>CPU geht in RUN unabhängig von Topologie bzw. optionalen Slaves.</b>		J	J	J	N	N	N	N	J	N
Ja: J   Nein: N   nicht relevant: X										

### 9.3 Hardware-Konfiguration - CPU

#### Voraussetzung

Die Konfiguration der CPU erfolgt im *"Hardware-Konfigurator"* von Siemens. Der Hardware-Konfigurator ist Bestandteil des Siemens SIMATIC Managers. Er dient der Projektierung. Bitte verwenden Sie für die Projektierung den Siemens SIMATIC Manager ab V 5.5 SP2. Die Module, die hier projektieren werden können, entnehmen Sie dem Hardware-Katalog, ggf. müssen Sie mit *"Extras*  
 → *Katalog aktualisieren"* den Hardware-Katalog aktualisieren.



*Für die Projektierung werden fundierte Kenntnisse im Umgang mit dem Siemens SIMATIC Manager und dem Hardware-Konfigurator vorausgesetzt!*

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 und kann mit dem VIPA-Tool *SPEED7 EtherCAT Manager* konfiguriert werden.

Folgende Voraussetzungen müssen für die Projektierung des EtherCAT-Masters erfüllt sein:

- GSDML für *"EtherCAT-Netzwerk"* ist installiert
- *SPEED7 EtherCAT Manager* für EtherCAT-Konfiguration ist installiert

#### 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](http://www.vipa.com)
2. ▶ Laden Sie die Datei Cx000151\_Vxxx
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](http://www.vipa.com).

Die Installation erfolgt nach folgender Vorgehensweise:

1. ▶ Schließen Sie den Siemens SIMATIC Manager.
2. ▶ Gehen Sie in den Service-Bereich von [www.vipa.com](http://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.

### **Vorgehensweise**

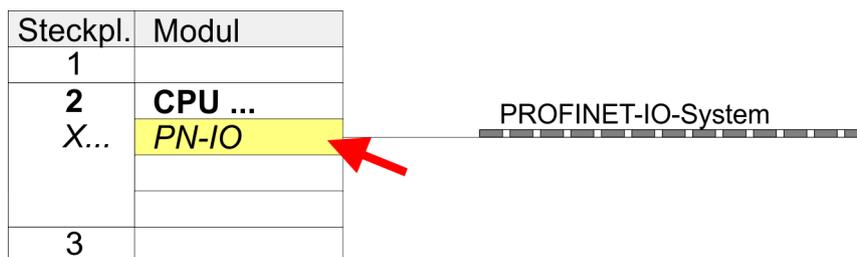
Steckpl.	Modul
1	
2	<b>CPU 315-2PN/DP</b>
X1	MPI/DP
X2	PN-IO
X2...	Port 1
X2...	Port 2
3	

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 315-2EH14-0AB0 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.

### **Konfiguration EtherCAT-Master**

1. ▶ Klicken Sie auf das Submodul "PN-IO" der CPU.
2. ▶ Wählen Sie "Kontextmenü → PROFINET IO-System einfügen".

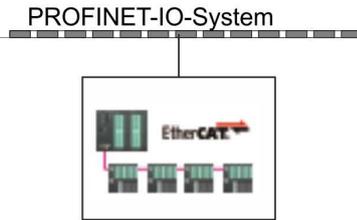


3. Legen Sie mit [Neu] ein neues Subnetz an und vergeben Sie gültige IP-Adress-Daten
4. Navigieren Sie im Hardware-Katalog in das Verzeichnis "PROFINET IO → Weitere Feldgeräte → I/O → VIPA EtherCAT System" und binden Sie das IO Device "EtherCAT-Netzwerk" an Ihr PROFINET-System an.
5. Klicken Sie auf 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.

Hierbei sind folgende Regeln zu beachten:

- Ein- und Ausgabebereiche können beliebig gemischt werden.
- Sie haben jeweils maximal 4096Byte an EtherCAT-Prozessdaten für Ein- und Ausgabe.
- Daten müssen im Siemens Hardware-Konfigurator konsistent sein, d.h. die maximale Anzahl an Bytes darf unter PROFINET nicht überschritten werden. Ansonsten müssen Sie ein weiteres "EtherCAT-Netzwerk" an Ihr PROFINET-System anbinden. Im *SPEED7 EtherCAT Manager* werden alle Teilbereiche automatisch erkannt und zusammengefasst.

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



**Katalog**

- ▼ **PROFINET IO**
- ...
- ▼ Weitere Feldgeräte
- ...
- ▼ I/O
- ...
- ▼ **VIPA EtherCAT System**
  - ▣ EtherCAT-Netzwerk
    - ▣ In 1024 byte
    - ...
    - ▣ In 128 byte
    - ▣ Out 1024 byte
    - ...
    - ▣ Out 128 byte

Steckpl.	Baugruppe	Bestellnummer
0	VIPA31x...	
1	In ... byte	
2	Out ... byte	
3		
4		
...		

**6.** Wählen Sie "Station → Speichern und übersetzen"

⇒ Sie können jetzt Ihr EtherCAT-System mit dem *SPEED7 EtherCAT Manager* projektieren.

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

**7.** Klicken Sie ein eingefügtes IO Device "EtherCAT-Netzwerk" und wählen Sie "Kontextmenü → Device Tool starten → *SPEED7 EtherCAT Manager*".

⇒ Der *SPEED7 EtherCAT Manager* wird gestartet. Hier können Sie die Konfiguration des EtherCAT-Master-System durchführen.

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

**8.** Durch Schließen des *SPEED7 EtherCAT Manager* wird die EtherCAT-Konfiguration in die Projektierung übernommen und der *SPEED7 EtherCAT Manager* geschlossen. Sie können Ihre EtherCAT-Konfiguration jederzeit im *SPEED7 EtherCAT Manager* wieder bearbeiten, da die Konfiguration in Ihrem Projekt gespeichert wird.

**9.** 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 Umräumen werden Slave- und Modul-Parameter nicht zurückgesetzt!**

## 9.4 EtherCAT Diagnose

### Übersicht

Über folgende Wege erhalten Sie Diagnose-Informationen von Ihrem System:

- Diagnose über den *SPEED7 EtherCAT Manager*
- Diagnose zur Laufzeit im Anwenderprogramm (OB 1, SFB 52)
- Diagnose über Systemzustandslisten - SZL
- Diagnose über OB-Startinformationen
- Diagnose über NCM-Diagnose
- Diagnose über Diagnosepuffer CPU bzw. CP
- Diagnose über die Status-LEDs

### 9.4.1 Diagnose über den *SPEED7 EtherCAT Manager*

#### Informationen

Der *SPEED7 EtherCAT Manager* bietet vielfältige Möglichkeiten für die Diagnose:

- Diagnose EtherCAT-Master
- Diagnose EtherCAT-Slave-Station



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

### 9.4.2 Diagnose zur Laufzeit im Anwenderprogramm (OB 1, SFB 52)

#### Hantierungsbaustein SFB 52 RDREC

Mit dem SFB 52 RDREC (read record) können Sie aus Ihrem Anwenderprogramm z.B. im OB1 auf Diagnosedaten zugreifen. Der SFB 52 ist ein asynchron arbeitender SFB, d.h. die Bearbeitung erstreckt sich über mehrere SFB-Aufrufe.



Nähere Informationen zum Einsatz des SFB 52 finden Sie in der Online-Hilfe zu ihrem Programmier-Tool und im Handbuch "SPEED7 Operationsliste" von VIPA.

Mit dem SFB 52 haben Sie Zugriff auf folgende Daten:

- CoE-Emergency-Meldungen (Datensatz 0x4000 ... 0x4003)
- EtherCAT-spezifischen Identifikationsdaten (Datensatz 0x1000)
- EtherCAT-Register von Slave-Stationen (Datensatz 0x3000)

### 9.4.2.1 Zugriff auf CoE-Emergency-Meldungen

#### Datensatz 0x4000 ... 0x4003

Mit dem SFB 52 RDREC (read record) können Sie mittels der Datensätze 0x4000 ... 0x4003 aus Ihrem Anwenderprogramm z.B. im OB 1 auf CoE-Emergency-Meldungen zugreifen. Der SFB 52 ist ein asynchron arbeitender SFB, d.h. die Bearbeitung erstreckt sich über mehrere SFB-Aufrufe. Ein Eintrag für die hier beschriebenen Datensätze 0x4000 ... 0x4003 besteht aus der CoE-Emergency selbst (8 Byte), und der Stations-Adresse, von der die CoE-Emergency kommt (2 Byte).

#### Datensatz-Struktur

Index [Byte]	Inhalt	Beschreibung
0	NumberOfEntries	Anzahl der nachfolgenden CoE-Emergency Einträge (0 ... n)
1		
2 + (n*12)	n * <i>CoE-Emergency Eintrag</i>	CoE-Emergency-Eintrag entsprechend dem angefragten Datensatz

#### CoE-Emergency Eintrag

Index [Byte]	Inhalt	Beschreibung
0	Error-Code	CoE Emergency
1		
2	Error-Register	
3	Error-Data	
4		
5		
6		
7		
8	Station Address	Adresse der Station, welche die Emergency geliefert hat.
9		
10	Reserved	
11		

**Datensätze**

<b>Datensatz</b>	<b>Beschreibung</b>
0x4000	<p>Der Datensatz liefert die zuletzt aufgetretene CoE-Emergency jedes Slave (ein CoE-Emergency Eintrag pro Slave, der eine CoE-Emergency geliefert hat). Für Slaves, bei denen keine CoE-Emergencies aufgetreten sind, werden keine Einträge geliefert.</p> <p>Parameter: Keine, NumberOfEntries: 0 ... 512</p>
0x4001	<p>Der Datensatz liefert die zuletzt aufgetretene CoE-Emergency eines bestimmten Slaves. Wird eine Slave-Id übergeben, die nicht vorhanden ist, wird ein Fehler geliefert. Wenn die Slave-ID gültig ist, aber keine CoE-Emergencies für diesen Slave vorhanden ist, ist die Anzahl der gelieferten Einträge entsprechend 0.</p> <p>Parameter: Slave-ID (1 ... 512), NumberOfEntries: 0 ... 1</p>
0x4002	<p>Der Datensatz liefert die 20 letzten CoE-Emergencies des Gesamtsystems (d.h. es können mehrere Einträge für einen Slave geliefert werden). Gibt es insgesamt weniger als 20 Einträge, ist die Anzahl der gelieferten Einträge entsprechend kleiner.</p> <p>Parameter: Keine, NumberOfEntries: 0 ... 20</p>
0x4003	<p>Der Datensatz liefert die 10 letzten CoE-Emergencies eines bestimmten Slaves. Wird eine Slave-ID übergeben, die nicht vorhanden ist, wird ein Fehler geliefert. Wenn die Slave-ID gültig ist, aber weniger als 10 CoE-Emergencies für diesen Slave vorhanden sind, ist die Anzahl der gelieferten Einträge entsprechend kleiner.</p> <p>Parameter: Slave-Id (1 ... 512), NumberOfEntries: 0 ... 10</p>

**Beispiel OB 1**

Für den zyklischen Zugriff auf einen Datensatz der Diagnosedaten einer EtherCAT Slave-Station können Sie folgendes Beispielprogramm im OB 1 verwenden:

```

UN M10.3 'Ist Lesevorgang beendet (BUSY=0)
UN M10.1 'und liegt kein Auftragsanstoß
      'an (REQ=0) dann
S M10.1 'starte Datensatz-Übertragung (REQ:=1)
L W#16#4000 'Datensatznummer (hier Datensatz
0x4000)
T MW12
CALL SFB 52, DB52 'Aufruf SFB 52 mit Instanz-DB
      REQ :=M10.1 'Anstoßmerker
      ID :=DW#16#0018 'Adresse des EtherCAT Slave
      INDEX :=MW12
      MLEN :=14 'Länge Datensatz 0x4000 bei 1 Eintrag
      VALID :=M10.2 'Gültigkeit des Datensatz
      BUSY :=M10.3 'Anzeige, ob Auftrag noch läuft
      ERROR :=M10.4 'Fehler-Bit während des Lesens
      STATUS :=MD14 'Fehlercodes
      LEN :=MW16 'Länge des gelesenen Datensatz
      RECORD :=P#M 100.0 Byte 40 'Ziel (MB100, 40Byte)
U M10.1
R M10.1 'Rücksetzen von REQ

```

**9.4.2.2 Zugriff auf EtherCAT-spezifische Identifikationsdaten****Datensatz 0x1000**

Der Datensatz 0x1000 enthält EtherCAT-spezifische Identifikationsdaten, welche mit dem SFB 52 gelesen werden können. Die Werte für *Device Type*, *Serial Number*, *Hardware Version* und *Software Version* werden direkt über CoE von der Slave-Station abgefragt. Sollte eine Slave-Station CoE oder einen dieser Werte im Objektverzeichnis nicht unterstützen, so werden die Werte mit 0xFF aufgefüllt. Der Datensatz hat folgende Struktur:

Index	Bezeichnung	Datentyp
1	Address	Unsigned32
2	Device Name	Array of char[32]
3	Vendor ID	Unsigned32
4	Product Code	Unsigned32
5	Device Type	Unsigned32
6	Serial Number	Unsigned32
7	Revision	Unsigned32
8	Hardware Version	Array of char[8]
9	Software Version	Array of char[8]

**9.4.2.3 Zugriff auf EtherCAT-Register von Slave-Stationen****Datensatz 0x3000**

Mit dem Datensatz 0x3000 können Sie auf die Register einer EtherCAT Slave-Station zugreifen, indem Sie diesen mit dem SFB 52 aufrufen. Der Datensatz hat folgende Struktur:

Byte	Inhalt	Register
0	AL Status	0x0130, 0x0131
1		
2	AL Control	0x0120, 0x0121
3		
4	AI Status Code	0x0134, 0x0135
5		
6	ESC DL Status	0x0110, 0x0111
7		
8	Processing Unit Error Counter	0x030C
9	PDI Error Counter	0x030D
10	Link Lost Counter Port A	0x0310
11	Link Lost Counter Port B	0x0311
12	Link Lost Counter Port C	0x0312
13	Link Lost Counter Port D	0x0313
14	reserviert	-
15	reserviert	-

### 9.4.3 Diagnose über Systemzustandslisten - SZL

#### SZL-Teillisten

Nachfolgend sind alle SZL-Teillisten mit zugehöriger SZL-ID aufgeführt, welche vom EtherCAT-Master System unterstützt werden.



Nähere Informationen zum Einsatz der SZLs finden Sie im Handbuch "SPEED7 Operationsliste" von VIPA.

SZL-Teillisten	SZL-ID
SZL-Inhaltsverzeichnis	xy00h
Baugruppen-Identifikation	xy11h
Zustand aller LEDs	xy19h
Zustand der LEDs	xy74h
Zustandsinfo CPU	xy91h
Stationszustandsinformation (EtherCAT)	xy94h
Baugruppenzustandsinformation (EtherCAT)	xy96h
Diagnosepuffer der CPU	xyA0h
Zustand EtherCAT-Master/Slave	xyE0h

SZL-Teillisten	SZL-ID
Zustand EtherCAT-Bus-System	xyE1h
Status der VSC-Features der System SLIO CPU	xyFCh

#### 9.4.4 Diagnose über OB-Startinformationen

Bei Auftreten eines Fehlers generiert das gestörte System eine Diagnosemeldung an die CPU. Daraufhin ruft die CPU den entsprechenden Diagnose-OB auf. Hierbei übergibt das CPU-Betriebssystem dem OB in den temporären Lokaldaten eine Startinformation. Durch Auswertung der Startinformation des entsprechenden OBs erhalten Sie Informationen über Fehlerursache und Fehlerort. Mit der Systemfunktion SFC 6 RD\_SINFO können Sie zur Laufzeit auf diese Startinformation zugreifen. Bitte beachten Sie hierbei, dass Sie die Startinformationen eines OBs nur im OB selbst lesen können, da es sich hier um temporäre Daten handelt.

Abhängig vom Fehlertyp werden folgende OBs im Diagnosefall aufgerufen:

- OB 82 bei Fehler an einem Modul an der EtherCAT-Slave-Station (Diagnosealarm) ↪ *"Alarm-Handling in der CPU" auf Seite 172*
- OB 86 Bei Ausfall bzw. Wiederkehr einer EtherCAT-Slave-Station ↪ *"OB-Startinformationen eintragen und OB aufrufen" auf Seite 171*
- OB 57 Herstellerspezifischer Alarm



Nähere Informationen zu den OBs finden Sie in der Online-Hilfe zu ihrem Programmier-Tool und im Handbuch "SPEED7 Operationsliste" von VIPA.

#### 9.4.5 Diagnose über NCM-Diagnose

↪ *"NCM-Diagnose - Hilfe zur Fehlersuche" auf Seite 149*

#### 9.4.6 Diagnose über Diagnosepuffer CPU bzw. CP

↪ *Kapitel 5.20 "VIPA-spezifische Diagnose-Einträge" auf Seite 75*

#### 9.4.7 Diagnose über Status-LEDs

##### LEDs EtherCAT-Schnittstelle X8

EC	MT	BF	Bedeutung
grün 	gelb 	rot 	
○	○	○	Master ist im Zustand INIT
BB	○	○	Master ist im Zustand Pre-Op

EC	MT	BF	Bedeutung
P	○	○	Master ist im Zustand Safe-Op
●	○	○	Master ist im Zustand OP
X	○	X	Es liegt kein Maintenance-Ereignis an
X	●	X	Ein Maintenance-Ereignis liegt an. Näheres hierzu finden Sie in der Diagnose
X	X	○	Es liegt kein Fehler am EtherCAT-Bus vor
X	X	●	<ul style="list-style-type: none"> <li>■ EtherCAT-Busfehler, keine Verbindung zu Subnetz</li> <li>■ falsche Übertragungsgeschwindigkeit</li> <li>■ Voll duplexübertragung ist nicht aktiviert</li> </ul>
X	X	B	<ul style="list-style-type: none"> <li>■ Ausfall eines angeschlossenen IO-Device</li> <li>■ Mindestens ein IO-Device ist nicht ansprechbar (Topologie-Fehler)</li> <li>■ Fehlerhafte Projektierung</li> </ul>
○	B4	B4	Fehlerhafte Projektierung: Im Diagnosepuffer wurde 0xEA64 eingetragen. Zusätzlich leuchtet die SF-LED der CPU.
○	BB*	BB*	* Das abwechselnde Blinken mit 4Hz zeigt an, dass ein Firmwareupdate des EtherCAT-Masters durchgeführt wird.
●	●	●	Firmwareupdate des EtherCAT-Masters wurde fehlerfrei durchgeführt.

an: ● | aus: ○ | blinkend (1Hz): B | blinkend (2Hz): BB | B4: blinkend (4s an, 1s aus) | pulsierend: P | flackernd: F | nicht relevant: X

**LEDs L/A, S**

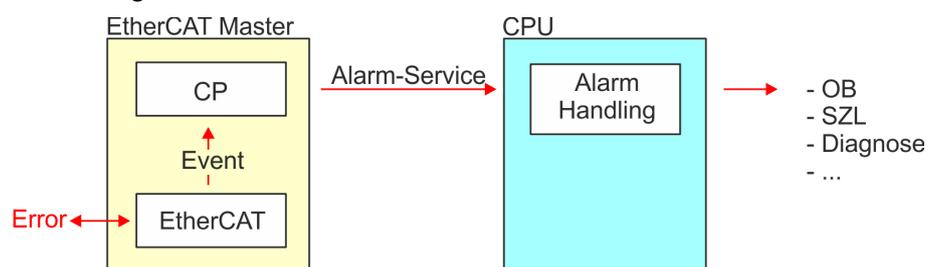
Die grüne L/A-LED (Link/Activity) zeigt an, dass der EtherCAT-Master physikalisch mit Ethernet verbunden ist. Unregelmäßiges Blinken der L/A-LED zeigt Kommunikation des EtherCAT-Masters über Ethernet an.

Leuchtet die grüne S-LED (Speed), so hat der EtherCAT-Master eine Übertragungsrate von 100MBit/s ansonsten mit 10MBit/s.

## 9.5 Alarmverhalten

### 9.5.1 Übersicht

Sobald ein Fehler auftritt, erkennt dies die EtherCAT-Elektronik und meldet intern einen Event (Notification) an den CP des EtherCAT-Masters. Im CP wird hieraus ein Alarm generiert, welcher in Form einer definierten Datenstruktur an die CPU weitergeleitet wird. Während des Alarmhandlings in der CPU wird daraufhin ermittelt, ob ein OB-Aufruf erfolgen soll, die Daten einer SZL zu aktualisieren oder weitere Aktionen erforderlich sind. Der EtherCAT-Master darf keinen Alarm an die CPU senden, solange er noch keine Konfiguration an die CPU gemeldet hat



## 9.5.2 Alarmtypen

### Alarmtypen

- MANUFACTURER\_SPECIFIC\_ALARM\_MIN (0x0020 oder 0x0021)
- PROZESS\_ALARM (0x0002) - OB 40
- BUS\_STATE\_CHANGED (0x8001) - OB 86
- DIAGNOSE\_ALARM\_GEHEND (0x000C) - OB 82
- DIAGNOSE\_ALARM\_KOMMEND (0x0001) - OB 82
- SLAVE\_STATE\_CHANGED (0x8002) - OB 86
- TOPOLOGY\_MISMATCH (0x8004) - OB 86
- TOPOLOGY\_OK (0x8003) - OB 86

### 9.5.2.1 MANUFACTURER\_SPECIFIC\_ALARM\_MIN (0x0020 oder 0x0021)

#### Eigenschaften

Auslösendes Event

- EC\_NOTIFY\_MBOXRCV - Mailbox-Nachricht erhalten - mit dem Typ eMbxTferType\_COE\_EMERGENCY

Mitgelieferte Daten

- Slave-Adresse
- CoE-Emergency

Bedingungen

- Der Error-Code der CoE-Emergency muss von einer VIPA Slave-Station stammen.
  - Der Error-Code der CoE-Emergency muss ungleich 0x0000 sein.
  - Der Error-Code der CoE-Emergency muss ungleich 0xA000 sein.
  - Der Error-Code der CoE-Emergency muss ungleich 0xA001 sein.
  - Der Error-Code der CoE-Emergency muss ungleich 0xFF00 sein.
  - Falls der Error-Code 0xFF00 ist, dann muss das 2. Byte ungleich 1 oder 2 sein.
- Der Error-Code der CoE-Emergency stammt von einer anderen Slave-Station.
  - Jede Emergency wird als OB 57 gemeldet.
- Es ist eine CoE-Emergency während einer Topologie-Änderung aufgetreten.
  - Der Error-Code der CoE-Emergency muss ungleich 0x0000 sein.
  - Der Error-Code der CoE-Emergency muss ungleich 0xA000 und 0xA001 sein.

#### Alarm-Handling in der CPU

#### OB-Startinformationen eintragen und OB aufrufen

Struktur-element	Datentyp	Beschreibung
EventClass	BYTE	0x11
FLT_ID	BYTE	0x5C
PrioLevel	BYTE	0x02
OBnr	BYTE	57

Struktur- element	Datentyp	Beschreibung
Reserved1	BYTE	0xCC
IoFlag	BYTE	0x54 oder 0x55 (abhängig vom Adresstyp des alarmauslösenden Moduls)
Info1	WORD	Diagnoseadresse des Slaves
Info2	WORD	Error-Code aus CoE-Emergency
Info3	WORD	Slavestate aus CoE-Emergency
User1	WORD	AlarmPrio, AlarmRef
User2	WORD	EtherCAT-Slave-Adresse

### SZL-Daten aktualisieren

Herstellerspezifische Alarme ändern keine SZLs

### Zwischenspeichern des Alarms

Snapshot zum Zeitpunkt des Alarmevents - kann über SFB 54 ausgewertet werden.

### Diagnosepuffer schreiben

EventId:= Eventclass, StartEvent	OBNr.	PK	Dat ID 1/2	Info1	Info2	Info3
0x115C	57	0x02	0x54CC	Diagnose- adresse Slave	Alarmtyp	Error-Code CoE-Emer- gency

#### 9.5.2.2 PROZESS\_ALARM (0x0002)

##### Eigenschaften

Auslösendes Event

- EC\_NOTIFY\_MBOXRCV - Mailbox-Nachricht erhalten - mit dem Typ eMbxTferType\_COE\_EMERGENCY

Mitgelieferte Daten

- Slave-Adresse
- CoE-Emergency

Bedingungen

- Der Error-Code der CoE-Emergency muss gleich 0xFF00 sein und die CoE-Emergency muss von einer VIPA Slave-Station stammen.
- Das 2. Byte von *MEF* muss 1 sein.

**Alarm-Handling in der CPU****OB-Startinformationen eintragen und OB aufrufen**

Struktur-element	Datentyp	Beschreibung
EventClass	BYTE	0x11
FLT_ID	BYTE	0x41
PrioLevel	BYTE	Priorität des OB 40
OBNr	BYTE	40
Reserved1	BYTE	reserviert
IoFlag	BYTE	0x54 oder 0x55 (abhängig vom Adresstyp des alarmlösenden Moduls)
Info1	WORD	Diagnoseadresse des Slaves
Info2	WORD	Error-Code aus CoE-Emergency
Info3	WORD	Slavestate aus CoE-Emergency
User1	WORD	Alarmprio, AlarmRef
User2	WORD	EtherCAT-Slave-Adresse

**SZL-Daten aktualisieren**

Prozessalarmländer keine SZLs

**Zwischenspeichern des Alarms**

Snapshot zum Zeitpunkt des Alarmevents - kann über SFB 54 ausgewertet werden.

**Diagnosepuffer schreiben**

Es erfolgt kein Diagnosepuffer-Eintrag.

**9.5.2.3 BUS\_STATE\_CHANGED (0x8001)****Eigenschaften**

Auslösendes Event

- EC\_NOTIFY\_STATECHANGED - Bus-Status wurde geändert

Mitgelieferte Daten

- Alter und neuer Status des Masters und die Anzahl der Slave-Module, welche sich nicht im Master-Status befinden.

Bedingungen

- keine

**Alarm-Handling in der CPU**

Für den Fall dass der Master nach "Operational" ↪ *Kapitel 9.1.2 "EtherCAT Zustandsmaschine" auf Seite 153* wechselt wird der OB86 ausgelöst. Über dessen Eventclass können Sie erkennen, ob alle projektierten Slave-Stationen den Statuswechsel durchgeführt haben. Sollten einzelne oder alle Slave-Stationen den Statuswechsel nach "Operational" nicht geschafft haben, so können Sie dies über eine SZL abfragen.

**OB-Startinformationen eintragen und OB aufrufen**

Struktur-element	Datentyp	Beschreibung
EventClass	BYTE	0xEC bei Wiederkehr oder 0xED bei Ausfall oder sonstigen VusState-Changed
FLT_ID	BYTE	0x10 Ausfall oder Wiederkehr mit allen Slaves, 0x11 Wiederkehr mit fehlenden Slave(s), 0x20 sonstiger BusState-Changed
PrioLevel	BYTE	Priorität des OB86
OBnr	BYTE	86
Reserved1	BYTE	1, wenn Slave verfügbar, sonst 0
IoFlag	BYTE	0x54 bei Eingangsadresse in ZInfo1, 0x55 bei Ausgangsadresse
Info1	WORD	0xXXYY: XX=OldState, YY=NewState
Info2	WORD	Diagnoseadresse des Masters
Info3	WORD	Anzahl der fehlenden Slaves
User1	WORD	0xXXYY: XX=AlarmPrio, YY=AlarmRef
User2	WORD	EtherCAT-Slave-Adresse

↪ *Kapitel 9.4.4 "Diagnose über OB-Startinformationen" auf Seite 166*

**SZL-Daten aktualisieren**

In der SZL 0x0294, 0x0694 und 0x0994 werden jeweils die entsprechenden Bits für die Slaves aktualisiert. Jeder als Alarmevent an die CPU gemeldete Zustandswechsel erzeugt einen Diagnosepuffereintrag und ist in der SZL 0xE0 auslesbar.

**E/A-Peripheriestruktur aktualisieren**

E/A-Status der Slaves und deren Module werden bei Wiederkehr auf EA\_STATUS\_BG\_VORHANDEN und bei Ausfall auf EA\_STATUS\_BG\_NICHTVORHANDEN gesetzt.

**Zwischenspeichern des Alarms**

Snapshot zum Zeitpunkt des Alarmevents - kann über SFB 54 ausgewertet werden.

**Diagnosepuffer schreiben**

EventId:= Eventclass, StartEvent	PrioLevel	OBNr.	Reserved1, IOFlag	Info1	Info2	Info3
0xEC10, 0xEC11, 0xED10 oder 0xED20 (abhängig vom Status- wechsel)	PrioLevel von OB86	86	siehe OB- Startinfo Reserved1, IOFlag	alter und neuer Status des Slaves	Diagnose Adresse Master	Anzahl der Slaves, welche vom Status des Masters abweichen

**9.5.2.4 DIAGNOSE\_ALARM\_GEHEND (0x000C)****Eigenschaften**

Auslösendes Event

- EC\_NOTIFY\_MBOXRCV - Mailbox-Nachricht erhalten - mit dem Typ eMbxTferType\_COE\_EMERGENCY

Mitgelieferte Daten

- Slave-Adresse
- CoE-Emergency

Bedingungen

- Der Error-Code der CoE-Emergency muss gleich 0x0000 ("kein Fehler" bzw. "Fehler behoben") sein und die CoE-Emergency muss von einer VIPA Slave-Station stammen.

**Alarm-Handling in der CPU****OB-Startinformationen eintragen und OB aufrufen**

Struktur- element	Datentyp	Beschreibung
EventClass	BYTE	0x38
FLT_ID	BYTE	0x42
PrioLevel	BYTE	Priorität des OB82
OBNr	BYTE	82
Reserved1	BYTE	0xC5
IoFlag	BYTE	0x54
Info1	WORD	Diagnoseadresse des Slaves
Info2	WORD	Error-Code aus CoE-Emergency
Info3	WORD	Slavestate aus CoE-Emergency

Struktur- element	Datentyp	Beschreibung
User1	WORD	Alarmprio, AlarmRef
User2	WORD	EtherCAT-Slave-Adresse

### SZL-Daten aktualisieren

In der SZL 0694 und 0692 wird jeweils das entsprechende Bit für den Slave aktualisiert.

### Zwischenspeichern des Alarms

Snapshot zum Zeitpunkt des Alarmevents - kann über SFB 54 ausgewertet werden.

### Diagnosepuffer schreiben

EventId:= Eventclass, StartEvent	PrioLevel	OBNr.	Reserved1, IOFlag	Info1	Info2	Info3
0x3842	PrioLevel von OB 82	82	0xC554	Diagnose Adresse Slaves	EtherCAT Error-Code	Slave Status

#### 9.5.2.5 DIAGNOSE\_ALARM\_Kommend (0x0001)

##### Eigenschaften

Auslösendes Event

- EC\_NOTIFY\_MBOXRCV - Mailbox-Nachricht erhalten - mit dem Typ eMbxTferType\_COE\_EMERGENCY

Mitgelieferte Daten

- Slave-Adresse
- CoE-Emergency

Bedingungen

- Der Error-Code der CoE-Emergency muss ungleich 0x0000 sein
- Der Error-Code der CoE-Emergency muss ungleich 0xA000 und 0xA001 sein

##### Alarm-Handling in der CPU

##### OB-Startinformationen eintragen und OB aufrufen

Struktur- element	Datentyp	Beschreibung
EventClass	BYTE	0x39
FLT_ID	BYTE	0x42
PrioLevel	BYTE	Priorität des OB82
OBNr	BYTE	82
Reserved1	BYTE	0xC5

Struktur- element	Datentyp	Beschreibung
IoFlag	BYTE	0x54
Info1	WORD	Diagnoseadresse des Slaves
Info2	WORD	Error-Code aus CoE-Emergency
Info3	WORD	Slavestate aus CoE-Emergency
User1	WORD	AlarmPrio, AlarmRef
User2	WORD	EtherCAT-Slave-Adresse

### SZL-Daten aktualisieren

In der SZL 0694 und 0692 wird jeweils das entsprechende Bit für den Slave aktualisiert.

### Zwischenspeichern des Alarms

Snapshot zum Zeitpunkt des Alarmevents - kann über SFB 54 ausgewertet werden.

### Diagnosepuffer schreiben

EventId:= Eventclass, StartEvent	PrioLevel	OBNr.	Reserved1, IOFlag	Info1	Info2	Info3
0x3942	PrioLevel von OB82	82	0xC554	Diagnose Adresse Slave	EtherCAT Error-Code	Slave Status

#### 9.5.2.6 SLAVE\_STATE\_CHANGED (0x8002)

##### Eigenschaften

Auslösendes Event

- EC\_NOTIFY\_SLAVE\_UNEXPECTED\_STATE - Slave ist nicht im angeforderten Status.
- Die Applikation hat einen Slave erfolgreich in einen anderen Zustand versetzt.

Mitgelieferte Daten

- aktueller neuer Status



Wenn gerade ein Master-Status-Wechsel durchgeführt wird, wird diese Meldung **nicht** zur CPU gesendet, da das Gesamtergebnis für fehlerhafte Slaves des Status-Wechsels im Event BUS\_STATE\_CHANGED übermittelt wird.

##### Alarm-Handling in der CPU

Der jeweils neue Slave-Status wird auf Seiten der CPU für jeden Slave gespeichert.

**OB-Startinformationen eintragen und OB aufrufen**

Struktur-element	Datentyp	Beschreibung
EventClass	BYTE	0xEC bei Wiederkehr oder 0xED bei Ausfall oder sonstigen VusState-Changed
FLT_ID	BYTE	0x12 Ausfall oder Wiederkehr, 0x22 sonstiger BusStateChanged
PrioLevel	BYTE	Priorität des OB 86
OBnr	BYTE	86
Reserved1	BYTE	1, wenn Slave verfügbar, sonst 0
IoFlag	BYTE	0x54 bei Eingangsadresse in ZInfo1, 0x55 bei Ausgangsadresse
Info1	WORD	0xXXYY: XX=OldState, YY=NewState
Info2	WORD	Diagnoseadresse des Slaves
Info3	WORD	AL Status Code
User1	WORD	0xXXYY: XX=AlarmPrio, YY=AlarmRef
User2	WORD	EtherCAT-Slave-Adresse

**SZL-Daten aktualisieren**

In der SZL 0x0294, 0x0694 und 0x0994 werden jeweils die entsprechenden Bits für die Slaves aktualisiert. Jeder als Alarmevent an die CPU gemeldete Zustandswechsel erzeugt einen Diagnosepuffereintrag und ist in der SZL 0xE0 auslesbar.

**E/A-Peripheriestruktur aktualisieren**

E/A-Status der Slaves und deren Module werden bei Wiederkehr auf EA\_STATUS\_BG\_VORHANDEN und bei Ausfall auf EA\_STATUS\_BG\_NICHTVORHANDEN gesetzt.

**Zwischenspeichern des Alarms**

Snapshot zum Zeitpunkt des Alarmevents - kann über SFB 54 ausgewertet werden.

**Diagnosepuffer schreiben**

EventId:= Eventclass, StartEvent	PrioLevel	OBNr.	Reserved1, IOFlag	Info1	Info2	Info3
0xEC10, 0xEC11, 0xED10 oder 0xED20 (abhängig vom Status- wechsel)	PrioLevel von OB 86	86	siehe OB- Startinfo Reserved1, IOFlag	alter und neuer Status des Slaves	Diagnose Adresse Master	Anzahl der Slaves, welche vom Status des Masters abweichen

**9.5.2.7 TOPOLOGY\_MISMATCH (0x8004)****Eigenschaften**

Auslösendes Event

- Alarm wird ausgelöst, wenn die Topology OK war und das Event EC\_NOTIFY\_SB\_MISMATCH auftritt. Der Alarm wird nur bei einer vorhandenen Konfiguration ausgelöst.

Mitgelieferte Daten

- keine

Bedingungen

- keine

**Alarm-Handling in der CPU****OB-Startinformationen eintragen und OB aufrufen**

Struktur- element	Datentyp	Beschreibung
EventClass	BYTE	0xED
FLT_ID	BYTE	0x30
PrioLevel	BYTE	Priorität des OB 86
OBNr	BYTE	86
Reserved1	BYTE	0
IoFlag	BYTE	0
Info1	WORD	0
Info2	WORD	Diagnoseadresse des Masters
Info3	WORD	0
User1	WORD	0
User2	WORD	0

**SZL Daten aktualisieren**

In der SZL xy94 wird eine Soll/Ist-Differenz eingetragen.

**Diagnosepuffer schreiben**

EventId:= Eventclass, StartEvent	PrioLevel	OBNr.	Reserved1, IOFlag	Info1	Info2	Info3
0xED30	PrioLevel von OB 86	86	0x0000	0	Diagnose Adresse Master	0

**9.5.2.8 TOPOLOGY\_OK (0x8003)**

**Eigenschaften**

Auslösendes Event

- Alarm wird ausgelöst, wenn die Topology OK war und das Event EC\_NOTIFY\_SB\_STATUS mit pScanBusStatus→dwResultCode = 0 auftritt. Der Alarm wird nur bei einer vorhandenen Konfiguration ausgelöst.

Mitgelieferte Daten

- keine

Bedingungen

- keine

**Alarm-Handling in der CPU**

**OB-Startinformationen eintragen und OB aufrufen**

Struktur- element	Datentyp	Beschreibung
EventClass	BYTE	0xED
FLT_ID	BYTE	0x30
PrioLevel	BYTE	Priorität des OB 86
OBNr	BYTE	86
Reserved1	BYTE	0
IoFlag	BYTE	0
Info1	WORD	0
Info2	WORD	Diagnoseadresse des Masters
Info3	WORD	0
User1	WORD	0
User2	WORD	0

**SZL Daten aktualisieren**

In der SZL xy94 wird eine Soll/Ist-Differenz eingetragen.

**Diagnosepuffer schreiben**

EventId:= Eventclass, StartEvent	PrioLevel	OBNr.	Reserved1, IOFlag	Info1	Info2	Info3
0xED30	PrioLevel von OB 86	86	0x0000	0	Diagnose Adresse Master	0

**9.6 Systemeigenschaften****Allgemeines**

Die Systemeigenschaften des EtherCAT-Masters sind nicht als Einschränkungen zu sehen bzw. einem Fehlverhalten gleichzustellen, vielmehr sind gewisse Funktionalitäten nicht zu erreichen, oder aus Sicht des Gesamtsystems gewollt.

**Verhalten bei Topologie-Änderungen**

Werden Topologie-Änderungen am EtherCAT-Bus durchgeführt, kann es zu Buszykluszeit-Überschreitungen kommen. Topologie-Änderungen sollten nicht im Zustand *Op* bzw. *SafeOp* durchgeführt werden, ggf. müssen Sie mittels *SPEED7 EtherCAT Manager* oder SDO-Zugriff den Status des EtherCAT-Masters manuell anpassen. Buszykluszeit-Überschreitungen können Sie mit dem OB 86 unter Einsatz des SFB 54 ermitteln. Nähere Informationen zum Einsatz des SFB 54 finden Sie in der im Handbuch "SPEED7 Operationsliste" von VIPA.

**Konfiguration von mehr als 128 EtherCAT Slave-Stationen**

Ab einer Konfiguration von mehr als 128 EtherCAT Slave-Stationen können die EtherCAT-Zustände nicht mehr korrekt aktualisiert werden, sobald eine Konfiguration auf die Baugruppe geladen wird. Hierbei zeigt die EC-LED des EtherCAT-Masters den Zustand *PreOp* an, obwohl sich dieser im *SafeOP* befindet. Auch die CPU bekommt den Zustand *PreOp* geliefert.

**Ursache:** Die CP-Applikation kann die große Anzahl an Stack-Notifications nicht mehr bearbeiten, da bei jedem Statuswechsel von jeder Slave-Station eine Notification gesendet wird.

**Abhilfe:** Indem Sie einen STOP/RUN-Übergang bei der CPU durchführen, wechselt das gesamte EtherCAT-System in den Zustand *OP*.

**Kompensation der Leitungslänge (Continuous propagation compensation)**

Der EtherCAT-Master unterstützt diese Funktion aktuell nicht.

**Verteilte Uhren (Distributed Clocks)**

Der EtherCAT-Master unterstützt *Distributed Clocks* nur bei Buszykluszeiten bis 4ms. Bei größeren Buszykluszeiten wird *Distributed Clocks* nicht unterstützt.

**SM Watchdog**

Sofern Sie lange Zykluszeiten (> 100ms) verwenden, sollten Sie im *SPEED7 EtherCAT Manager* immer den "SM Watchdog" ebenfalls entsprechend erhöhen oder ausschalten. Ansonsten wechselt Ihre Slave-Station nach Ablauf der "SM Watchdog"-Zeit in Safe-Op und löst den OB 86 aus. Von jetzt ab können Sie diesen Slave nur noch manuell in Op setzen! Ohne Anpassung der "SM Watchdog"-Zeit bekommen Sie bei Einsatz der EtherCAT Slave-Stationen von VIPA bei Zykluszeiten > 100ms immer die Fehlermeldung AISTatusCode 0x1B. Hierbei belässt die CPU die Slave-Station im aktuellen Status, d.h. dieser wird beim Polling ignoriert. Den Status können Sie aber mittels SDO-Zugriff bzw. mit dem *SPEED7 EtherCAT Manager* ändern.



*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 Umlöschen werden Slave- und Modul-Parameter nicht zurückgesetzt!***

**9.7 Firmwareupdate****EtherCAT-Master**

↪ Kapitel 5.14 "Firmwareupdate" auf Seite 66

**EtherCAT-Slave-Station**

Firmwareupdate über den *SPEED7 EtherCAT Manager*. Näheres hierzu finden Sie im zugehörigen Handbuch bzw. in der Onlinehilfe.

**9.8 Zugriff auf das Objektverzeichnis****9.8.1 Übersicht****Bausteine**

Mit folgenden Bausteinen haben Sie zur Laufzeit Zugriff auf das Objektverzeichnis von EtherCAT-Slave-Stationen und EtherCAT-Master:

- FB 52 - Read SDO - Lesezugriff auf Objektverzeichnis
- FB 53 - Write SDO - Schreibzugriff auf Objektverzeichnis



*Hierbei handelt es sich um VIPA-spezifische Bausteine. Näheres zum Einsatz dieser Bausteine finden Sie im Handbuch "Operationsliste".*

*Bitte beachten Sie beim Zugriff auf das Objektverzeichnis, dass abhängig von Ihrem Master-System, die Byte-Reihenfolge gedreht sein kann!*

## 9.8.2 FB 52 - Read SDO - Lesezugriff auf Objektverzeichnis

### Beschreibung

Mit diesem Baustein können Sie auf das Objektverzeichnis von EtherCAT-Slave-Stationen und EtherCAT-Master lesend zugreifen. Hierbei handelt es sich um einen asynchron arbeitenden Baustein, d.h. die Bearbeitung erstreckt sich über mehrere Baustein-Aufrufe. Sie starten den SDO-Auftrag, indem Sie den FB 52 mit REQ = 1 aufrufen. Über den Ausgangsparameter BUSY und den Ausgangsparameter RETVAL wird der Zustand des Auftrags angezeigt. Die Datensatzübertragung ist abgeschlossen, wenn der Ausgangsparameter BUSY den Wert FALSE angenommen hat. Die Fehlerbehandlung erfolgt über die Parameter ERROR, ERROR\_ID und RETVAL.

### Parameter

Parameter	DeklARATION	Datentyp	Beschreibung
REQ	IN	BOOL	REQ = 1: Aktiviert den SDO-Zugriff bei steigender Flanke.
ID	IN	WORD	Logische Basisadresse der EtherCAT-Slave-Station bzw. des Masters in der Hardwarekonfiguration.  Bei einer Ausgabebaugruppe muss Bit 15 gesetzt werden (Bsp. für Adresse 5: ID:=DW#16#8005). Bei einer Mischbaugruppe ist die kleinere der beiden Adressen anzugeben.
INDEX	IN	WORD	Index des Objekts für den SDO-Zugriff.
SUBINDEX	IN	BYTE	Subindex des Objekts für den SDO-Zugriff.
COMPL_ACCESS	IN	BOOL	Mit diesem Parameter wird bestimmt, ob nur ein einzelner Subindex oder das gesamte Objekt gelesen werden soll.
MLEN	IN	INT	Maximale Länge der zu lesenden Daten.
VALID	OUT	BOOL	Gibt an, ob ein neuer Datensatz empfangen wurde und gültig ist.
BUSY	OUT	BOOL	Dieser Parameter gibt den Bearbeitungsstatus des SDO-Zugriffs an.  <i>BUSY = 1</i> : SDO-Zugriff ist noch in Bearbeitung.
ERROR	OUT	BOOL	<i>ERROR = 1</i> : Beim Lesevorgang trat ein Fehler auf.
RETVAl	OUT	INT	Rückgabewert (0 = OK)
ERROR_ID	OUT	DWORD	Busspezifischer Fehlercode. Ist während der Bearbeitung des SDO-Zugriffs ein Fehler aufgetreten, so ist in diesem Parameter der SDO-Abort-Fehlercode (EtherCAT-Fehlercode) angegeben.
LEN	OUT	INT	Länge der gelesenen Daten.
RECORD	INOUT	ANY	Bereich für die gelesenen Daten.

**Besonderheiten bei *COMPL\_ACCESS* (CompleteAccess)**

Bei Aktivierung des Parameters *COMPL\_ACCESS* ist folgendes zu beachten:

- Bei *COMPL\_ACCESS* = true darf der *SUBINDEX* nur 0 oder 1 betragen! Ansonsten bekommen Sie eine Fehlermeldung.
- Bei *COMPL\_ACCESS* = true werden für *SUBINDEX* 0 2Byte ausgelesen, da *SUBINDEX* 1 einen Offset von 2Byte besitzt.

**RETVAL (Rückgabewert)**

Zusätzlich zu den hier aufgeführten modulspezifischen Fehlercodes sind auch noch die allgemeingültigen Fehlercodes für FC/SFCs als Rückgabewert möglich.

RETVAL	Description
0x80A5	Fehler beim Lesen eines SDO vom Master-System. Einen Fehlercode finden Sie in <i>ERROR_ID</i> .
0x80A6	Fehler beim Lesen eines SDO von einer EtherCAT-Slave-Station. Einen Fehlercode finden Sie in <i>ERROR_ID</i> .
0x80D2	Fehler beim Lesen eines SDO aufgrund falscher Aufruf-Parameter. Einen Fehlercode finden Sie in <i>ERROR_ID</i> .

**ERROR\_ID**

Wenn der Parameter *RETVAL* den Wert 0x80A5 oder 0x80A6 hat finden Sie in *ERROR\_ID* die entsprechende Fehlermeldung. Ansonsten ist *ERROR\_ID* 0.

**Internal error**

0x00000000	No error
0x98110001	Feature not supported
0x98110002	Invalid Index
0x98110003	Invalid Offset
0x98110005	Invalid Size
0x98110006	Invalid Data
0x98110007	Not ready
0x98110008	Busy
0x9811000A	No Memory left
0x9811000B	Invalid Parameter
0x9811000C	Not Found
0x9811000E	Invalid state
0x98110010	Timeout
0x98110011	Open Failed
0x98110012	Send Failed
0x98110014	Invalid Command

Zugriff auf das Objektverzeichnis&gt; FB 52 - Read SDO - Lesezugriff auf Objektverzeichnis

0x98110015	Unknown Mailbox Protocol Command
0x98110016	Access Denied
0x98110024	Slave error

Value	Text	Possible error cause
0x98110040	SDO: Toggle bit not alternated	CoE abort code 0x05030000 of slave
0x98110041	SDO protocol timed out	CoE abort code 0x05040000 of slave
0x98110042	SDO: Client/server command specifier not valid or unknown	CoE abort code 0x05040001 of slave
0x98110043	SDO: Invalid block size (block mode only)	CoE abort code 0x05040002 of slave
0x98110044	SDO: Invalid sequence number (block mode only)	CoE abort code 0x05040003 of slave
0x98110045	SDO: CRC error (block mode only)	CoE abort code 0x05040004 of slave
0x98110046	SDO: Out of memory	CoE abort code 0x05040005 of slave
0x98110047	SDO: Unsupported access to an object	CoE abort code 0x06010000 of slave
0x98110048	SDO: Attempt to read a write only object	CoE abort code 0x06010001 of slave
0x98110049	SDO: Attempt to write a read only object	CoE abort code 0x06010002 of slave
0x9811004A	SDO: Object does not exist in the object dictionary	CoE abort code 0x06020000 of slave
0x9811004B	SDO: Object cannot be mapped to the PDO	CoE abort code 0x06040041 of slave
0x9811004C	SDO: The number and length of the objects to be mapped would exceed PDO length	CoE abort code 0x06040042 of slave
0x9811004D	SDO: General parameter incompatibility reason	CoE abort code 0x06040043 of slave
0x9811004E	SDO: General internal incompatibility in the device	CoE abort code 0x06040047 of slave
0x9811004F	SDO: Access failed due to an hardware error	CoE abort code 0x06060000 of slave
0x98110050	SDO: Data type does not match, length of service parameter does not match	CoE abort code 0x06070010 of slave
0x98110051	SDO: Data type does not match, length of service parameter too high	CoE abort code 0x06070012 of slave
0x98110052	SDO: Data type does not match, length of service parameter too low	CoE abort code 0x06070013 of slave
0x98110053	SDO: Sub-index does not exist	CoE abort code 0x06090011 of slave
0x98110054	SDO: Value range of parameter exceeded (only for write access)	CoE abort code 0x06090030 of slave
0x98110055	SDO: Value of parameter written too high	CoE abort code 0x06090031 of slave
0x98110056	SDO: Value of parameter written too low	CoE abort code 0x06090032 of slave
0x98110057	SDO: Maximum value is less than minimum value	CoE abort code 0x06090036 of slave
0x98110058	SDO: General error	CoE abort code 0x08000000 of slave
0x98110059	SDO: Data cannot be transferred or stored to the application	CoE abort code 0x08000020 of slave
0x9811005A	SDO: Data cannot be transferred or stored to the application because of local control	CoE abort code 0x08000021 of slave

Value	Text	Possible error cause
0x9811005B	SDO: Data cannot be transferred or stored to the application because of the present device state	CoE abort code 0x08000022 of slave
0x9811005C	SDO: Object dictionary dynamic generation fails or no object dictionary is present (e.g. object dictionary is generated from file and generation fails because of an file error)	CoE abort code 0x08000023 of slave
0x9811005D	SDO: Unknown code	Unknown CoE abort code of slave
0x9811010E	Command not executed	Slave is not present at the bus

### 9.8.3 FB 53 - Write SDO - Schreibzugriff auf Objektverzeichnis

#### Beschreibung

Mit diesem Baustein können Sie auf das Objektverzeichnis von EtherCAT-Slave-Stationen und EtherCAT-Master schreibend zugreifen. Hierbei handelt es sich um einen asynchron arbeitenden Baustein, d.h. die Bearbeitung erstreckt sich über mehrere Baustein-Aufrufe. Sie starten den SDO-Auftrag, indem Sie den FB 53 mit *REQ* = 1 aufrufen. Über den Ausgangsparameter *BUSY* und den Ausgangsparameter *RETVAL* wird der Zustand des Auftrags angezeigt. Die Datensatzübertragung ist abgeschlossen, wenn der Ausgangsparameter *BUSY* den Wert *FALSE* angenommen hat. Die Fehlerbehandlung erfolgt über die Parameter *ERROR*, *ERROR\_ID* und *RETVAL*.

#### Parameter

Parameter	DeklARATION	Datentyp	Beschreibung
REQ	IN	BOOL	REQ = 1: Aktiviert den SDO-Zugriff bei steigender Flanke.
ID	IN	WORD	Logische Basisadresse der EtherCAT-Slave-Station bzw. des Masters in der Hardwarekonfiguration.  Bei einer Ausgabebaugruppe muss Bit 15 gesetzt werden (Bsp. für Adresse 5: ID:=DW#16#8005). Bei einer Mischbaugruppe ist die kleinere der beiden Adressen anzugeben.
INDEX	IN	WORD	Index des Objekts für den SDO-Zugriff.
SUBINDEX	IN	BYTE	Subindex des Objekts für den SDO-Zugriff.
COMPL_ACCESS	IN	BOOL	Mit diesem Parameter wird bestimmt, ob nur ein einzelner Subindex oder das gesamte Objekt geschrieben werden soll.
LEN	IN	INT	Maximale Länge der zu schreibenden Daten.
DONE	OUT	BOOL	Gibt an, ob ein neuer Datensatz übertragen wurde.
BUSY	OUT	BOOL	Dieser Parameter gibt den Bearbeitungsstatus des SDO-Zugriffs an.  <i>BUSY</i> = 1: SDO-Zugriff ist noch in Bearbeitung.

Zugriff auf das Objektverzeichnis&gt; FB 53 - Write SDO - Schreibzugriff auf Objektverzeichnis

Parameter	Deklaration	Datentyp	Beschreibung
ERROR	OUT	BOOL	<i>ERROR</i> = 1: Beim Schreibvorgang trat ein Fehler auf.
RETVAL	OUT	INT	Rückgabewert (0 = OK)
ERROR_ID	OUT	DWORD	Busspezifischer Fehlercode. Ist während der Bearbeitung des SDO-Zugriffs ein Fehler aufgetreten, so ist in diesem Parameter der SDO-Abort-Fehlercode (EtherCAT-Fehlercode) angegeben.
LEN	OUT	INT	Länge der zu schreibenden Daten.
RECORD	INOUT	ANY	Bereich für die zu schreibenden Daten.

### Besonderheiten bei *COMPL\_ACCESS* (CompleteAccess)

Bei Aktivierung des Parameters *COMPL\_ACCESS* ist folgendes zu beachten:

- Bei *COMPL\_ACCESS* = true darf der *SUBINDEX* nur 0 oder 1 betragen! Ansonsten bekommen Sie eine Fehlermeldung.
- Bei *COMPL\_ACCESS* = true werden für *SUBINDEX* 0 2Byte geschrieben, da *SUBINDEX* 1 einen Offset von 2Byte besitzt.

### RETVAL (Rückgabewert)

Zusätzlich zu den hier aufgeführten modulspezifischen Fehlercodes sind auch noch die allgemeingültigen Fehlercodes für FC/SFCs als Rückgabewert möglich.

RETVAL	Description
0x80A5	Fehler beim Schreiben eines SDO vom Master-System. Einen Fehlercode finden Sie in <i>ERROR_ID</i> .
0x80A6	Fehler beim Schreiben eines SDO von einer EtherCAT-Slave-Station. Einen Fehlercode finden Sie in <i>ERROR_ID</i> .
0x80D2	Fehler beim Schreiben eines SDO aufgrund falscher Aufruf-Parameter. Einen Fehlercode finden Sie in <i>ERROR_ID</i> .

### ERROR\_ID

Wenn der Parameter *RETVAL* den Wert 0x80A5 oder 0x80A6 hat finden Sie in *ERROR\_ID* die entsprechende Fehlermeldung. Ansonsten ist *ERROR\_ID* 0.

#### Internal error

0x00000000	No error
0x98110001	Feature not supported
0x98110002	Invalid Index
0x98110003	Invalid Offset

Zugriff auf das Objektverzeichnis&gt; FB 53 - Write SDO - Schreibzugriff auf Objektverzeichnis

0x98110005	Invalid Size
0x98110006	Invalid Data
0x98110007	Not ready
0x98110008	Busy
0x9811000A	No Memory left
0x9811000B	Invalid Parameter
0x9811000C	Not Found
0x9811000E	Invalid state
0x98110010	Timeout
0x98110011	Open Failed
0x98110012	Send Failed
0x98110014	Invalid Command
0x98110015	Unknown Mailbox Protocol Command
0x98110016	Access Denied
0x98110024	Slave error

Value	Text	Possible error cause
0x98110040	SDO: Toggle bit not alternated	CoE abort code 0x05030000 of slave
0x98110041	SDO protocol timed out	CoE abort code 0x05040000 of slave
0x98110042	SDO: Client/server command specifier not valid or unknown	CoE abort code 0x05040001 of slave
0x98110043	SDO: Invalid block size (block mode only)	CoE abort code 0x05040002 of slave
0x98110044	SDO: Invalid sequence number (block mode only)	CoE abort code 0x05040003 of slave
0x98110045	SDO: CRC error (block mode only)	CoE abort code 0x05040004 of slave
0x98110046	SDO: Out of memory	CoE abort code 0x05040005 of slave
0x98110047	SDO: Unsupported access to an object	CoE abort code 0x06010000 of slave
0x98110048	SDO: Attempt to read a write only object	CoE abort code 0x06010001 of slave
0x98110049	SDO: Attempt to write a read only object	CoE abort code 0x06010002 of slave
0x9811004A	SDO: Object does not exist in the object dictionary	CoE abort code 0x06020000 of slave
0x9811004B	SDO: Object cannot be mapped to the PDO	CoE abort code 0x06040041 of slave
0x9811004C	SDO: The number and length of the objects to be mapped would exceed PDO length	CoE abort code 0x06040042 of slave
0x9811004D	SDO: General parameter incompatibility reason	CoE abort code 0x06040043 of slave
0x9811004E	SDO: General internal incompatibility in the device	CoE abort code 0x06040047 of slave
0x9811004F	SDO: Access failed due to an hardware error	CoE abort code 0x06060000 of slave
0x98110050	SDO: Data type does not match, length of service parameter does not match	CoE abort code 0x06070010 of slave

Objekt-Verzeichnis &gt; Objektübersicht

Value	Text	Possible error cause
0x98110051	SDO: Data type does not match, length of service parameter too high	CoE abort code 0x06070012 of slave
0x98110052	SDO: Data type does not match, length of service parameter too low	CoE abort code 0x06070013 of slave
0x98110053	SDO: Sub-index does not exist	CoE abort code 0x06090011 of slave
0x98110054	SDO: Value range of parameter exceeded (only for write access)	CoE abort code 0x06090030 of slave
0x98110055	SDO: Value of parameter written too high	CoE abort code 0x06090031 of slave
0x98110056	SDO: Value of parameter written too low	CoE abort code 0x06090032 of slave
0x98110057	SDO: Maximum value is less than minimum value	CoE abort code 0x06090036 of slave
0x98110058	SDO: General error	CoE abort code 0x08000000 of slave
0x98110059	SDO: Data cannot be transferred or stored to the application	CoE abort code 0x08000020 of slave
0x9811005A	SDO: Data cannot be transferred or stored to the application because of local control	CoE abort code 0x08000021 of slave
0x9811005B	SDO: Data cannot be transferred or stored to the application because of the present device state	CoE abort code 0x08000022 of slave
0x9811005C	SDO: Object dictionary dynamic generation fails or no object dictionary is present (e.g. object dictionary is generated from file and generation fails because of a file error)	CoE abort code 0x08000023 of slave
0x9811005D	SDO: Unknown code	Unknown CoE abort code of slave
0x9811010E	Command not executed	Slave is not present at the bus

## 9.9 Objekt-Verzeichnis

### 9.9.1 Objektübersicht

Index	Object Dictionary Area
0x0000 ... 0x0FFF	Data Type Area Objects
0x1000 ... 0x1FFF	CoE Communication Area Objects
0x2000 ... 0x20FF	Generic Master Area Objects
0x2100 ... 0x21FF	Distributed Clocks Objects
0x3000 ... 0x3FFF	Slave Configuration / Information Objects
0x4000 ... 0x7FFF	Reserved Area
0x8000 ... 0x8FFF	CoE Slave Configuration Objects
0x9000 ... 0x9FFF	CoE Slave Information Objects
0xA000 ... 0xAFFF	CoE Slave Diagnosis Data Objects
0xB000 ... 0xEFFF	Reserved Area
0xF000 ... 0xFFFF	CoE Device Area Objects

### 9.9.2 CoE Communication Area Objects: 0x1000-0x1FFF

Index	Object Type	Name	Type
0x1000	VAR	Device Type	Unsigned32
0x1001	VAR	Error Register	Unsigned8
0x1008	VAR	Manufacturer Device Name String	VisibleString
0x1009	VAR	Manufacturer Hardware Version String	VisibleString
0x100A	VAR	Manufacturer Software Version String	VisibleString
0x1018	RECORD	Identity Object	Identity (0x23)
0x10F3	RECORD	History Object	History (0x26)

#### 9.9.2.1 Device Type 0x1000

Sub-index	Name	Type	Access	Value	Meaning
0x00	Device Type	Unsigned32	ro	0x00001389	0x00001389 means MDP

#### 9.9.2.2 Device Name 0x1008

Sub-index	Name	Type	Access	Value	Meaning
0x00	Device name	Visible string	ro	VIPA 31x	Name of the EtherCAT device

#### 9.9.2.3 Hardware Version 0x1009

Sub-index	Name	Type	Access	Value	Meaning
0x00	Hardware version	Visible string	ro	"V MM.mm.ss.bb" MM = Major Version mm = Minor Version ss = Service Pack bb = Build e.g. "V 01.05.02.02"	Hardware version of the EtherCAT device

Objekt-Verzeichnis&gt; CoE Communication Area Objects: 0x1000-0x1FFF

## 9.9.2.4 Software Version 0x100A

Sub-index	Name	Type	Access	Value	Meaning
0x00	Software version	Visible string	ro	"V MM.mm.ss.bb" MM = Major Version mm = Minor Version ss = Service Pack bb = Build e.g. "V 01.05.02.02"	Software version of the EtherCAT device

## 9.9.2.5 Identity Object 0x1018

Sub-index	Name	Type	Access	Value	Meaning
0x00	Number of Entries	Unsigned8	ro	0x04 (default)	
0x01	Vendor ID	Unsigned32	ro	0x0000022B (default)	Vendor ID of the EtherCAT device
0x02	Product Code	Unsigned32	ro	0x00001636 (default)	Product Code of the EtherCAT device
0x03	Revision Number	Unsigned32	ro	0x00000000 (default)	Revision Number (EtherCAT master software version)
0x04	Serial Number	Unsigned32	ro	0x00000000 (default)	Serial Number of the EtherCAT device

## 9.9.2.6 History Object 0x10F3

Sub-index	Name	Type	Access	Value	Meaning
0	Number of Entries	Unsigned8	ro		
1	Maximum number of Diag messages	Unsigned8	ro		
2	Subindex of newest Diag message	Unsigned8	ro		
3	Subindex of newest acknowledged Diag message	Unsigned8	r/w		
4	New Diag messages available	BOOL32	ro		

Sub-index	Name	Type	Access	Value	Meaning
5	Flags (UINT16, r/w)	Unsigned16	r/w	0	Bit 0 = 1: Enable Emergency sending (default = 0) Bit 1 = 1: Disable Storing Info Messages (default = 0) Bit 2 = 1: Disable Storing Warning Messages (default = 0) Bit 3 = 1: Disable Storing Error Messages (default = 0) Bit 4...15: reserved for future use
6 ... 255			ro		

#### 9.9.2.6.1 Diagnosis Messages Object 0x10F3: 6-255

Byte-Offset	Name	Type	Access	Value	Meaning
0	Diag-Number	Unsigned32	ro		Bit 0...11: free use Bit 12...15 = 14: to be comp. with Emergency Error Bit 16...31 = 0: reserved Bit 16...31 = 0xFFFE: free use Bit 16...31 = 0xFFFF: reserved
4	Flags	Unsigned16	ro		Bit 0...3: Diag type (0 = Info, 1 = warning, 2 = error) Bit 4...15: reserved
6	Text ID	Unsigned16	ro		0 = no Text ID 1-65535 = Reference to a Text ID with formatted string
8	Time Stamp in ns (from DC)	Unsigned64	ro		
16	Flags parameter 1	Unsigned16	ro		
18	Parameter 1	several	ro		

Objekt-Verzeichnis&gt; Generic Master Objects: 0x2000-0x20FF

Byte-Offset	Name	Type	Access	Value	Meaning
N	Flags parameter n	Unsigned16	ro		
N+2	Parameter n	several	ro		

### 9.9.3 Generic Master Objects: 0x2000-0x20FF

Index	Object Type	Name	Type
0x2000	VAR	Master State Change Command Register	Unsigned32
0x2001	VAR	Master State Summary	Unsigned32
0x2002	RECORD	Bus Diagnosis Object	BusDiagnostic (0x40)
0x2005	RECORD	MAC Address	MACAddress (0x41)
0x2010	VAR	Debug Register	Unsigned48
0x2020	RECORD	Master Init. Parameters	MasterInitParm (0x42)

#### 9.9.3.1 Master State Change Command Register 0x2000

Sub-index	Name	Type	Access	Value	Meaning
0x00	Master State	Unsigned32	r/w	0 = invalid 1 = init 2 = pre-operational 3 = bootstrap mode 4 = safe operational 8 = operational	

## 9.9.3.2 Master State Summary 0x2001

Sub-index	Name	Type	Access	Value	Meaning
0x00	Master State	Unsigned32	ro		Bit 0: = 1 Master OK Bit 1...3: reserved Bit 4...7: Master State Bit 8: Slaves in requested State Bit 9: Master in requested State Bit 10: Bus Scan Match Bit 11: reserved Bit 12: DC is enabled Bit 13: DC In-Sync Bit 14: DC Busy Bit 15: Reserved Bit 16: Link Up Bit Bit 17...31: reserved

Master ist Ok wenn Topologie Ok (Mismatch wenn nicht projektiertes Slave vorhanden). Master muss in *Op* sein, Slaves müssen im *Op* sein und *Distributed Clocks* muss *insync* sein sofern aktiv.

Parameter Flags Bit 12...15	Parameter Flags Bit 0...11	Type of Data	Data
0	CoE DataType e.g. 0x0007 = UINT32	Data Type	Data defined through CoE DataType
1	Length in Byte	Byte Array	Byte stream byData[Size]
2	Length in Byte	ASCII-String	String szString[Length] (not '\0' terminated)
3	Length in Byte	Unicode String	String wszString[Length/2] (not L'\0' terminated)
4	0	Text Id	Text Id (Word)

## 9.9.3.3 Bus Diagnosis Object 0x2002

Object Type: RECORD, Manufacturer Specific Identity 0x40

Subindex	Description	Type	Access
0x00	Number of Entries	Unsigned8	ro
0x01	Reserved	Unsigned16	ro

Objekt-Verzeichnis&gt; Generic Master Objects: 0x2000-0x20FF

Subindex	Description	Type	Access
0x02	Configuration Checksum CRC32	Unsigned32	ro
0x03	Number of found Slave	Unsigned32	ro
0x04	Number of found DC Slave	Unsigned32	ro
0x05	Number of Slaves in Configuration	Unsigned32	ro
0x06	Number of Mailbox Slaves in Configuration	Unsigned32	ro
0x07	Counter: TX frames	Unsigned32	ro
0x08	Counter: RX frames	Unsigned32	ro
0x09	Counter: Lost frames	Unsigned32	ro
0x10	Counter: Cyclic frames	Unsigned32	ro
0x11	Counter: Cyclic datagrams	Unsigned32	ro
0x12	Counter: Acyclic frames	Unsigned32	ro
0x13	Counter: Acyclic datagrams	Unsigned32	ro
0x14	Clear Counters by writing 1 to bit(s) Bit 0: Clear all Counters Bit 1: Clear Tx Frame Counter (Idx 7) Bit 2: Clear Rx Frame Counter (Idx 8) Bit 3: Clear Lost Frame Counter (Idx 9) Bit 4: Clear Cyclic Frame Counter (Idx 10) Bit 5: Clear Cyclic Datagram Counter (Idx 11) Bit 6: Clear Acyclic Frame Counter (Idx 12) Bit 7: Clear Acyclic DataGram Counter (Idx 13) Bit 8...31: Reserved	Unsigned32	r/w

#### 9.9.3.4 MAC Address 0x2005

Object Type: RECORD, Manufacturer Specific Identity 0x41

Subindex	Description	Type	Access
0x00	Number of Entries	Unsigned8	ro
0x01	Hardware	Unsigned48	ro
0x02	Red Hardware	Unsigned48	ro
0x03	Configuration Source	Unsigned48	ro
0x04	Configuration Destination	Unsigned48	

## 9.9.3.5 Debug Register 0x2010

Sub-index	Name	Type	Access	Value	Meaning
0x00	Debug Register	Unsigned38	r/w	Upper 16Bit: 0: activate LinkError Messages 1...15: reserved Lower 32Bit: Definition of parameter dwStateChangeDebug in structure EC_T_MASTER_CONFIG	

## 9.9.3.6 Master Init Parameters 0x2020

Object Type: RECORD, Manufacturer Specific Identity 0x42

Sub-index	Description	Type	Access
00	Number of Entries	Unsigned8	ro
01	EC_T_INITMASTERPARMS.dwVersion Application	Unsigned32	ro
02	dwVersion Master	Unsigned32	ro
03	EC_T_MASTER_CONFIG.nSlaveMultiplier	Unsigned32	ro
04	EC_T_MASTER_CONFIG.dwEcatCmdTimeout in millisec	Unsigned32	ro
05	EC_T_MASTER_CONFIG.dwEcatCmdMaxRetries	Unsigned32	ro
06	EC_T_MASTER_CONFIG.dwCycTimeout in millisec	Unsigned32	ro
07	EC_T_MASTER_CONFIG.dwEoeTimeout in millisec	Unsigned32	ro
08	EC_T_MASTER_CONFIG.dwFoeBusyTimeout in millisec	Unsigned32	ro
09	EC_T_MASTER_CONFIG.dwMaxQueuedEthFrames	Unsigned32	ro
10	EC_T_MASTER_CONFIG.dwMaxSlaveCmdPerFrame	Unsigned32	ro
11	EC_T_MASTER_CONFIG.dwMaxQueuedCoeSlaves	Unsigned32	ro
12	EC_T_MASTER_CONFIG.dwMaxQueuedCoeCmds	Unsigned32	ro
13	EC_T_MASTER_CONFIG.dwStateChangeDebug	Unsigned32	ro
14	EC_T_LINK_DEV_PARAM.szDriverIdent	VisibleString	ro
15	EC_T_LINK_DEV_PARAM.bPollingModeActive	Bool32	ro
16	EC_T_LINK_DEV_PARAM.bAllocSendFrameActive	Bool32	ro

Objekt-Verzeichnis &gt; Slave specific objects

**9.9.4 Distributed Clocks Objects: 0x2100-0x21FF**

Index	Object Type	Name	Type
0x2100	VAR	DC Slave Sync Deviation Limit	Unsigned32
0x2101	VAR	DC Current Deviation	Signed32
0x2102	VAR	DC Reserved	Unsigned32
0x2103	VAR	DC Reserved	Unsigned32

**9.9.4.1 Distributed Clocks Slave Sync Deviation Limit 0x2100**

Sub-index	Name	Type	Access	Value	Meaning
0x00	Master State	Unsigned32	ro	dwDevLimit	

**9.9.4.2 Distributed Clocks Current Deviation 0x2101**

Sub-index	Name	Type	Access	Value	Meaning
0x00	Master State	Unsigned32	ro	dwDeviation	

**9.9.4.3 Reserviert: 0x2102 / 0x2103**

Dieser Wert ist reserviert.

**9.9.5 Slave specific objects**

Slave Configuration / Information Objects: 0x3000-0x3FFF

Index	Object Type	Name	Type
0x3000	RECORD	Slave Configuration and Information Objects	SlaveCfgInfo (0x43)
...			
0x3FFF			

## CoE Slave Configuration Objects: 0x8000-0x8FFF

Index	Object Type	Name	Type
0x8000	RECORD	One index entry for each configured slave (from ESI)	SlaveCfg (0x45)
...			
0x8FFF			

## CoE Slave Information Objects: 0x9000-0x9FFF

Index	Object Type	Name	Type
0x9000	RECORD	One index entry for each connected BUS-slave (updated during BUS scan)	SlaveInfo (0x46)
...			
0x9FFF			

## CoE Slave Diagnosis Data Objects: 0xA000-0xAFFF

Index	Object Type	Name	Type
0xA000	RECORD	One subindex entry for each connected BUS-slave (cyclic updated)	SlaveDiag (0x47)
...			
0xAFFF			

## 9.9.5.1 Slave Configuration and Information Object 0x3000-0x3FFF

Object Type: RECORD, Manufacturer Specific Identity 0x43

Subindex	Description	Type	Access
0	Number of Entries	Unsigned8	ro
1	Entry Valid	Bool32	ro
2	VendorId (Bus)	Unsigned32	ro
3	ProductCode (Bus)	Unsigned32	ro
4	Revision No (Bus)	Unsigned32	ro
5	Serial No (Bus)	Unsigned32	ro
6	Device Name (Config)	Visible_String[80]	ro
7	Auto Increment Address (Bus)	Unsigned16	ro
8	Physical Address (Bus)	Unsigned16	ro
9	Config Physical Address (Config)	Unsigned16	ro
10	Alias Address (Bus)	Unsigned16	ro

Objekt-Verzeichnis&gt; Slave specific objects

Subindex	Description	Type	Access
11	PortState (Bus)	Unsigned16	ro
12	DC Support (Bus)	Bool32	ro
13	DC Support 64Bit (Bus)	Bool32	ro
14	Mailbox Support (Config)	Bool32	ro
15	Requested State (slave instance)	Unsigned32	r/w
16	Current State (slave instance)	Unsigned32	ro
17	Error Flag Set (slave instance)	Bool32	ro
18	Enable Linkmessages (slave instance)	Bool32	r/w
19	Error code (slave instance)	Unsigned32	ro
20	Sync Pulse active (Config, slave instance)	Bool32	ro
21	DC Sync 0 Period (Config, slave instance)	Unsigned32	ro
22	DC Sync 1 Period (Config, slave instance)	Unsigned32	ro
23	SB Error Code (Bus Topology)	Unsigned32	ro
24	RX Error Counter Port 0 (Bus)	Unsigned16	ro
25	RX Error Counter Port 1 (Bus)	Unsigned16	ro
26	RX Error Counter Port 2 (Bus)	Unsigned16	ro
27	RX Error Counter Port 3 (Bus)	Unsigned16	ro
28	Forwarded RX Error Counter Port 0 (Bus)	Unsigned8	ro
29	Forwarded RX Error Counter Port 1 (Bus)	Unsigned8	ro
30	Forwarded RX Error Counter Port 2 (Bus)	Unsigned8	ro
31	Forwarded RX Error Counter Port 3 (Bus)	Unsigned8	ro
32	EtherCAT Processing Unit Error Counter (Bus)	Unsigned8	ro
33	PDI Error Counter (Bus)	Unsigned8	ro
34	Reserved	Unsigned16	ro
35	Lost Link Counter Port 0 (Bus)	Unsigned8	ro
36	Lost Link Counter Port 1 (Bus)	Unsigned8	ro
37	Lost Link Counter Port 2 (Bus)	Unsigned8	ro
38	Lost Link Counter Port 3 (Bus)	Unsigned8	ro
39	FMMU's supported (Bus)	Unsigned8	ro

Subindex	Description	Type	Access
40	Sync Managers supported (Bus)	Unsigned8	ro
41	RAM Size in kByte (Bus)	Unsigned8	ro
42	Port Descriptor (Bus)	Unsigned8	ro
43	ECS Type (Config)	Unsigned8	ro
44	Slave is optional (Config)	Bool32	ro
45	Slave is present (Bus)	Bool32	ro
46	Hot connect group ID	Unsigned32	ro

### 9.9.5.2 CoE Slave Configuration Objects: 0x8000-0x8FFF

Object Type: RECORD, Manufacturer Specific Identity 0x45

Die Konfigurationsdaten enthalten Informationen über die EtherCAT-Slaves.

Subindex	Description	Type	Access
0	Number of Entries	Unsigned8	ro
1	Fixed Station Address	Unsigned16	ro
2	Type	Visible_String[64]	ro
3	Name	Visible_String[64]	ro
4	Device Type	Unsigned32	ro
5	Vendor ID	Unsigned32	ro
6	Product Code	Unsigned32	ro
7	Revision Number	Unsigned32	ro
8	Version Number	Unsigned32	ro
33	Mailbox Out Size (if mailbox slave)	Unsigned16	ro
34	Mailbox In Size (if mailbox slave)	Unsigned16	ro

### 9.9.5.3 CoE Slave Information Objects: 0x9000-0x9FFF

Object Type: RECORD, Manufacturer Specific Identity 0x46

Informationen über die angeschlossenen EtherCAT-Slaves erhalten sie über die Informationsdaten. Sie werden verfügbar, wenn der Scan Befehl ausgeführt wurde.

Subindex	Description	Type	Access
0	Number of Entries	Unsigned8	ro
1	<b>Fixed Station Address</b> of the Nth EtherCAT slave found (same value as 0xF040: 01)	Unsigned16	ro
5	<b>Vendor ID</b> of the Nth EtherCAT slave found (entry 0x1018: 01 of the EtherCAT slave)	Unsigned32	ro
6	<b>Product Code</b> of the Nth EtherCAT slave found (entry 0x1018: 02 of the EtherCAT slave)	Unsigned32	ro
7	<b>Revision Number</b> of the first EtherCAT slave found (entry 0x1018: 03 of the EtherCAT slave)	Unsigned32	ro
8	<b>Version Number</b> of the first EtherCAT slave found (entry 0x1018: 04 of the EtherCAT slave)	Unsigned32	ro
32	<b>DL Status</b> (Register 0x110-0x111) of the Nth EtherCAT slave found.	Unsigned16	ro

#### 9.9.5.4 CoE Slave Diagnosis Data Objects: 0xA000-0xAFFF

Object Type: RECORD, Manufacturer Specific Identity 0x47

Die Diagnosedaten beinhalten die Status- und die Diagnoseinformationen der EtherCAT-Slaves oder der Verbindungen der EtherCAT-Slaves.

Subindex	Description	Type	Access
0	<b>Number of Entries</b>	Unsigned8	ro
1	<b>AL Status</b> (Register 0x130-0x131) of the Nth EtherCAT slave configured.	Unsigned16	ro
2	<b>AL Control</b> (Register 0x120-0x121) of the Nth EtherCAT slave configured.	Unsigned16	r/w

### 9.9.6 CoE Device Area Objects: 0xF000-0xFFFF

Index	Object Type	Name	Type
0xF000	RECORD	Modular Device Profile	DeviceProfile (0x48)
0xF002	RECORD	Detect Modules Command	DetectCmd (0x49)
0xF020	RECORD	Configured Address List	ConfAddrList (0x50)
...			
0xF02F			
0xF040	RECORD	Detected Address List	ConnAddrList (0x51)
...			
0xF04F			

#### 9.9.6.1 Modular Device Profile Object 0xF000

Object Type: RECORD, Manufacturer Specific Identity 0x48

Subindex	Description	Type	Access
0	<b>Number of Entries</b>	Unsigned8	ro
1	<b>Index distance</b> between two modules. This value is always read as 1.	Unsigned16	ro
2	<b>Maximum number of EtherCAT slaves</b> connected to the EtherCAT bus. This value is read as 512.	Unsigned16	ro
3	<b>Available entries in objects 0x8xxx</b> (number of configured slaves).	Unsigned32	ro
4	<b>Available entries in objects 0x9xxx</b> (number of connected slaves).	Unsigned32	ro

#### 9.9.6.2 Configured Address List Object 0xF020-0xF02F

Object Type: RECORD, Manufacturer Specific Identity 0x50

Subindex	Description	Type	Access
0	<b>Number of Entries</b>	Unsigned8	ro
1	Fixed Station Address of the first EtherCAT slave configured.	Unsigned16	ro
2	Fixed Station Address of the second EtherCAT slave configured.	Unsigned16	ro
...	...		ro
255	Fixed Station Address of the 255. EtherCAT slave configured.	Unsigned16	ro

Objekt-Verzeichnis&gt; CoE Device Area Objects: 0xF000-0xFFFF

Subindex	Description	Type	Access
0	<b>Number of Entries</b>	Unsigned8	ro
1	Fixed Station Address of the 256. EtherCAT slave configured.	Unsigned16	ro
...	...		

### 9.9.6.3 Detected Address List Object 0xF040-0xF04F

Object Type: RECORD, Manufacturer Specific Identity 0x51

Subindex	Description	Type	Access
0	<b>Number of Entries</b>	Unsigned8	ro
1	Fixed Station Address of the first EtherCAT slave detected.	Unsigned16	ro
2	Fixed Station Address of the second EtherCAT slave detected.	Unsigned16	ro
...	...		ro
255	Fixed Station Address of the 255. EtherCAT slave detected.	Unsigned16	ro
0	<b>Number of Entries</b>	Unsigned8	ro
1	Fixed Station Address of the 256. EtherCAT slave detected.	Unsigned16	ro
...	...		