



# VIPA System 200V



## **CPU | Handbuch**

HB97D\_CPU | Rev. 12/02

Januar 2012

## **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 (91 32) 744 -0

Fax.: +49 9132 744 1864

E-Mail: [info@vipa.de](mailto:info@vipa.de)

<http://www.vipa.de>

## **Hinweis**

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.

## **CE-Konformität**

Hiermit erklärt VIPA GmbH, dass die Produkte und Systeme mit den grundlegenden Anforderungen und den anderen relevanten Vorschriften der folgenden Richtlinien übereinstimmen:

- 2004/108/EG Elektromagnetische Verträglichkeit
- 2006/95/EG Niederspannungsrichtlinie

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, 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)

## Inhaltsverzeichnis

<b>Über dieses Handbuch</b> .....	<b>1</b>
<b>Sicherheitshinweise</b> .....	<b>2</b>
<b>Teil 1 Grundlagen</b> .....	<b>1-1</b>
Sicherheitshinweise für den Benutzer .....	1-2
Übersicht.....	1-3
Komponenten.....	1-4
Allgemeine Beschreibung System 200V.....	1-5
<b>Teil 2 Montage und Aufbaurichtlinien</b> .....	<b>2-1</b>
Übersicht.....	2-2
Montage.....	2-5
Verdrahtung .....	2-8
Einbaumaße.....	2-10
Aufbaurichtlinien.....	2-12
<b>Teil 3 Hardwarebeschreibung</b> .....	<b>3-1</b>
Einleitung .....	3-2
Arbeitsweise einer CPU.....	3-3
Funktionssicherheit der VIPA CPUs .....	3-5
Einsatzbereiche.....	3-6
Systemübersicht.....	3-8
Aufbau.....	3-16
Komponenten.....	3-18
Blockschaltbild .....	3-27
Technische Daten .....	3-28
<b>Teil 4 Einsatz CPU 21x</b> .....	<b>4-1</b>
Schnelleinstieg.....	4-2
Hinweise zum Einsatz der MPI-Schnittstelle.....	4-6
Montage.....	4-7
Anlaufverhalten .....	4-8
Adressvergabe.....	4-9
Projektierung.....	4-11
Einstellung der CPU-Parameter .....	4-14
Projekt transferieren.....	4-16
Betriebszustände.....	4-20
Urlöschen.....	4-21
Firmwareupdate .....	4-23
VIPA-spezifische Diagnose-Einträge.....	4-26
Mit Testfunktionen Variablen steuern und beobachten.....	4-28
<b>Teil 5 Einsatz CPU 21x-2BT10 unter TCP/IP</b> .....	<b>5-1</b>
Industrial Ethernet in der Automatisierung.....	5-2
ISO/OSI-Schichtenmodell .....	5-3
Grundbegriffe .....	5-6
Protokolle .....	5-7
IP-Adresse und Subnetz .....	5-10
Planung eines Netzwerks.....	5-12
Kommunikationsmöglichkeiten des CP .....	5-15
Funktionsübersicht .....	5-18
Schnelleinstieg .....	5-19
Hardware-Konfiguration .....	5-23

Kommunikationsverbindungen projektieren.....	5-26
SEND/RECEIVE im SPS-Anwenderprogramm.....	5-32
Projekt transferieren.....	5-37
NCM-Diagnose - Hilfe zur Fehlersuche .....	5-39
Kopplung mit Fremdsystemen.....	5-42
Beispiel zur Kommunikation CPU 21x-2BT10.....	5-45
<b>Teil 6 Einsatz CPU 21x-2BT02 unter H1 / TCP/IP .....</b>	<b>6-1</b>
Grundlagen .....	6-2
Planung eines Netzwerks .....	6-7
MAC- und IP-Adresse .....	6-9
Projektierung der CPU 21x-2BT02 .....	6-11
Beispiel zur Kommunikation CPU 21x-2BT02.....	6-23
Anlaufverhalten .....	6-34
Systemeigenschaften der CPU 21x-2BT02 .....	6-35
Kopplung mit Fremdsystemen.....	6-37
Testprogramm für TCP/IP-Verbindungen.....	6-40
<b>Teil 7 Einsatz CPU 21xDPM .....</b>	<b>7-1</b>
Grundlagen .....	7-2
Projektierung CPU mit integriertem PROFIBUS-DP-Master .....	7-5
Projekt transferieren.....	7-9
DP-Master-Betriebsarten.....	7-12
Inbetriebnahme und Anlaufverhalten.....	7-13
<b>Teil 8 Einsatz CPU 21xDP .....</b>	<b>8-1</b>
Grundlagen .....	8-2
Projektierung der CPU 21xDP .....	8-7
DP-Slave Parameter .....	8-12
Diagnosefunktionen .....	8-15
Statusmeldung intern an CPU.....	8-18
PROFIBUS Aufbaurichtlinien.....	8-20
Inbetriebnahme .....	8-23
Beispiel .....	8-25
<b>Teil 9 Einsatz CPU 21xCAN.....</b>	<b>9-1</b>
Grundlagen CAN-Bus.....	9-2
Projektierung der CPU 21xCAN .....	9-4
Betriebsarten.....	9-13
Prozessabbild der CPU 21xCAN .....	9-14
CANopen Telegrammaufbau.....	9-16
Objekt-Verzeichnis .....	9-21
<b>Teil 10 Einsatz CPU 21xSER-1 .....</b>	<b>10-1</b>
Schnelleinstieg .....	10-2
Protokolle und Prozeduren .....	10-3
Einsatz der seriellen Schnittstelle .....	10-7
Prinzip der Datenübertragung .....	10-8
Parametrierung .....	10-10
Kommunikation .....	10-14
Modemfunktionalität .....	10-20
Modbus Slave Funktionscodes.....	10-21
Modbus - Beispiel zur Kommunikation.....	10-25

<b>Teil 11 Einsatz CPU 21xSER-2</b> .....	<b>11-1</b>
Schnelleinstieg .....	11-2
Protokolle und Prozeduren .....	11-3
RS232-Schnittstelle .....	11-7
Kommunikationsprinzip .....	11-8
Schnittstellen initialisieren .....	11-9
Schnittstellen-Parameter .....	11-11
Schnittstellenkommunikation .....	11-14



## Über dieses Handbuch

Das Handbuch beschreibt die bei VIPA erhältlichen System 200V CPUs. Hier finden Sie neben einer Produktübersicht eine detaillierte Beschreibungen der einzelnen Module. Sie erhalten Informationen für den Anschluss und die Handhabung der CPUs im System 200V.

### Überblick

#### **Teil 1: Grundlagen**

Im Rahmen dieser Einleitung erfolgt die Vorstellung des System 200V von VIPA als zentrales bzw. dezentrales Automatisierungssystem.

Des Weiteren finden Sie hier allgemeine Hinweise zum System 200V.

#### **Teil 2: Montage und Aufbaurichtlinien**

Alle Informationen, die für den Aufbau und die Verdrahtung einer Steuerung aus den Komponenten des Systems 200V erforderlich sind, finden Sie in diesem Kapitel.

#### **Teil 3: Hardwarebeschreibung CPU**

Die CPU 21x erhalten Sie in verschiedenen Ausführungen, auf die in diesem Kapitel weiter eingegangen werden soll.

#### **Teil 4: Einsatz CPU 21x**

In diesem Kapitel ist der Einsatz des CPU-Teils mit den Peripherie-Modulen des System 200V beschrieben, die sich neben der CPU auf der gleichen Busschiene befinden.

#### **Teil 5: Einsatz CPU 21x-2BT10 unter TCP/IP**

Inhalt dieses Kapitels ist der Einsatz der CPU 21x-2BT10 und die Kommunikation des CPs unter TCP/IP.

Der CP ist parametrierbar mit NetPro von Siemens.

#### **Teil 6: Einsatz CPU 21x-2BT02 unter H1 / TCP/IP**

In diesem Kapitel ist der Einsatz der CPU 21x-2BT02 und die Kommunikation des CPs unter H1 bzw. TCP/IP beschrieben.

Der CP ist parametrierbar mit WinNCS.

#### **Teil 7: Einsatz CPU 21xDPM**

Inhalt dieses Kapitels ist der Einsatz und die Projektierung der CPU 21xDPM mit integriertem PROFIBUS-DP-Master. Neben der Beschreibung der DP-Master-Betriebsarten finden Sie hier auch Angaben zum Anlaufverhalten und zur Inbetriebnahme.

**Teil 8: Einsatz CPU 21xDP**

Dieses Kapitel befasst sich mit der CPU 21xDP (intelligenter Slave). Beschrieben sind hier Einsatz, Projektierung und Parametrierung der CPU 21xDP unter PROFIBUS.

**Teil 9: Einsatz CPU 21xCAN**

Inhalt dieses Kapitels ist der Einsatz der CPU 21xCAN unter CANopen. Sie erhalten hier alle Informationen, die zum Einsatz der integrierten CAN-Komponenten erforderlich sind.

**Teil 10: Einsatz CPU 21xSER-1**

Inhalt dieses Kapitels ist der Einsatz der CPU 21xSER-1 mit RS232/RS485-Schnittstelle.

**Teil 11: Einsatz CPU 21xSER-2**

Inhalt dieses Kapitels ist der Einsatz der CPU 21x-2BS02 mit zwei RS232-Schnittstellen.

<b>Zielsetzung und Inhalt</b>	Das Handbuch beschreibt die bei VIPA erhältlichen System 200V CPUs. Beschrieben werden Aufbau, Projektierung und Technische Daten.
<b>Zielgruppe</b>	Das Handbuch ist geschrieben für Anwender mit Grundkenntnissen in der Automatisierungstechnik.
<b>Aufbau des Handbuchs</b>	Das Handbuch ist in Kapitel gegliedert. Jedes Kapitel beschreibt eine abgeschlossene Thematik.
<b>Orientierung im Dokument</b>	Als Orientierungshilfe stehen im Handbuch zur Verfügung: <ul style="list-style-type: none"><li>• Gesamt-Inhaltsverzeichnis am Anfang des Handbuchs</li><li>• Übersicht der beschriebenen Themen am Anfang jedes Kapitels</li></ul>
<b>Verfügbarkeit</b>	Das Handbuch ist verfügbar in: <ul style="list-style-type: none"><li>• gedruckter Form auf Papier</li><li>• in elektronischer Form als PDF-Datei (Adobe Acrobat Reader)</li></ul>
<b>Piktogramme Signalwörter</b>	Besonders wichtige Textteile sind mit folgenden Piktogrammen und Signalworten ausgezeichnet:

**Gefahr!**

Unmittelbar drohende oder mögliche Gefahr.  
Personenschäden sind möglich.

**Achtung!**

Bei Nichtbefolgen sind Sachschäden möglich.

**Hinweis!**

Zusätzliche Informationen und nützliche Tipps

## Sicherheitshinweise

### Bestimmungsgemäße Verwendung

Die CPU 21x ist konstruiert und gefertigt für:

- alle VIPA System-200V-Komponenten
- 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



### 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!**

## Teil 1 Grundlagen

### Übersicht

Kernthema dieses Kapitels ist die Vorstellung des System 200V von VIPA. In einer Übersicht werden die Möglichkeiten zum Aufbau von zentralen und dezentralen Systemen aufgezeigt.

Auch finden Sie hier allgemeine Angaben zum System 200V wie Maße, Hinweise zur Montage und zu den Umgebungsbedingungen.

### Inhalt

Thema	Seite
<b>Teil 1 Grundlagen</b> .....	<b>1-1</b>
Sicherheitshinweise für den Benutzer .....	1-2
Übersicht .....	1-3
Komponenten .....	1-4
Allgemeine Beschreibung System 200V .....	1-5

## Sicherheitshinweise für den Benutzer

### Handhabung elektrostatisch gefährdeter Baugruppen

VIPA-Module und 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 Komponenten wird nachfolgendes Symbol verwendet:



Das Symbol befindet sich auf Modulen, Baugruppen, Baugruppenträgern oder auf Verpackungen und weist so auf elektrostatisch gefährdete Komponenten 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 diese 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.



### Achtung!

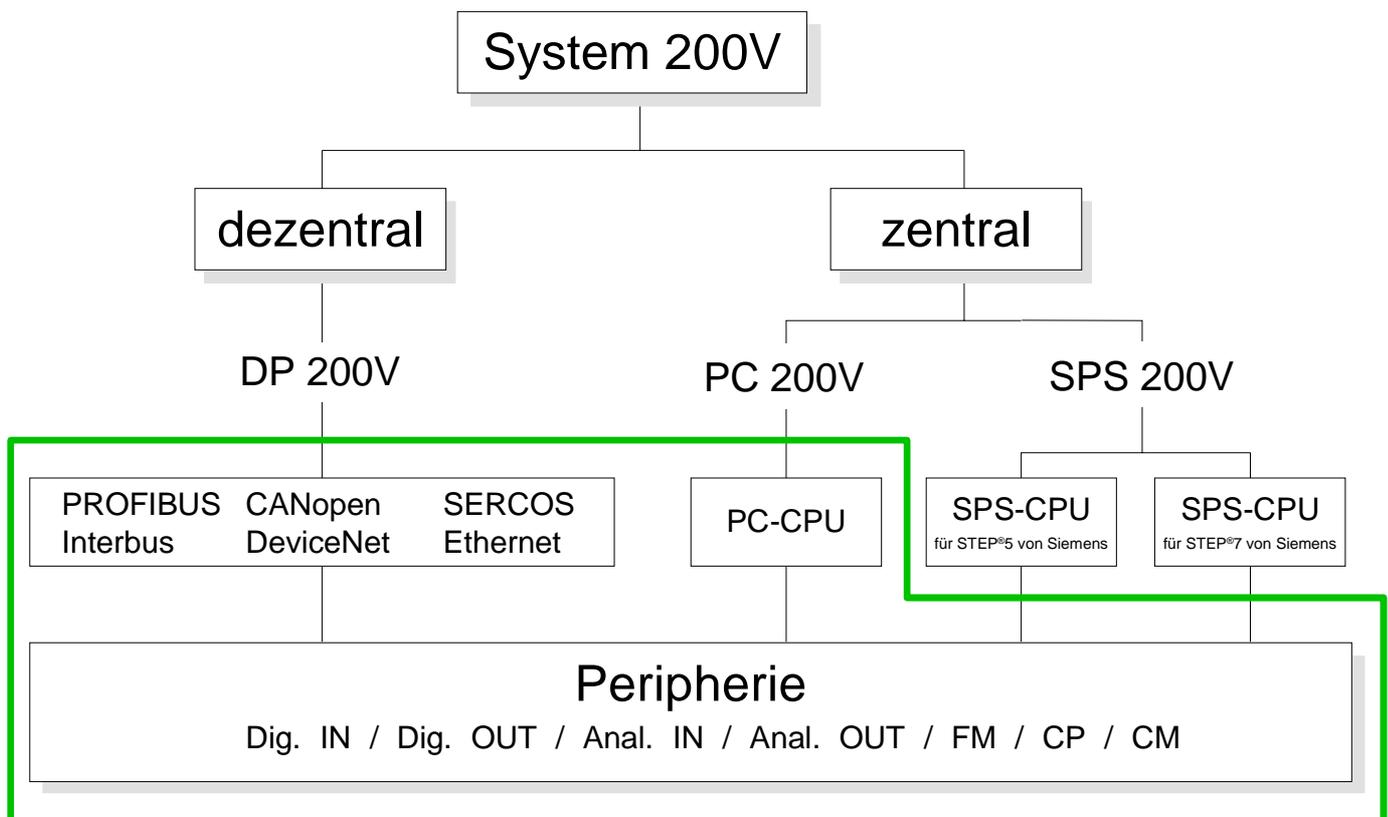
Bei Arbeiten mit und an elektrostatisch gefährdeten Baugruppen ist auf ausreichende Erdung des Menschen und der Arbeitsmittel zu achten.

## Übersicht

### Das System 200V

Das System 200V ist ein modulares, zentral wie dezentral einsetzbares Automatisierungssystem für Anwendungen im unteren und mittleren Leistungsbereich. Die einzelnen Module werden direkt auf eine 35mm-Normprofilschiene montiert und über Busverbinder, die vorher in die Profilschiene eingelegt werden, gekoppelt.

Die nachfolgende Abbildung soll Ihnen den Leistungsumfang des System 200V verdeutlichen:



## Komponenten

- Zentrales System** Im System 200V stehen verschiedene SPS-CPU's zur Verfügung. Programmiert wird in STEP<sup>®</sup>5 oder STEP<sup>®</sup>7 von Siemens.
- CPU's mit integrierter Ethernetanschaltung oder mit zusätzlichen seriellen Schnittstellen garantieren eine komfortable Integration der SPS in ein Netzwerk oder den Anschluss von zusätzlichen Endgeräten.
- Das Anwenderprogramm wird im Flash oder einem zusätzlich steckbaren Speichermodul gespeichert.
- Bedienen/Beobachten, Steuerungsaufgaben oder andere Dateiverarbeitungsaufgaben können mit der PC-basierenden CPU 288 realisiert werden.
- Programmiert wird in C++ oder Pascal.
- Die PC 288-CPU ermöglicht einen aktiven Zugriff auf den Rückwandbus und ist so mit allen Peripherie- und Funktionsmodulen des VIPA System 200V als zentrale Steuerung einsetzbar.
- Mit einer Zeilenanschaltung ist ein Aufbau des System 200V in bis zu 4 Zeilen möglich.
- Dezentrales System** Die SPS-CPU's oder die PC-CPU bilden, in Kombination mit einem PROFIBUS DP-Master, die Basis für ein PROFIBUS-DP-Netzwerk nach DIN 19245-3. Das DP-Netzwerk können Sie mit dem VIPA Projektierool WinNCS bzw. mit dem SIMATIC Manager projektieren.
- Die Anbindung an weitere Feldbusgeräte ermöglichen Slaves für Interbus, CANopen, DeviceNet, SERCOS und Ethernet.
- Peripheriemodule** Von VIPA erhalten Sie eine Vielzahl an Peripheriemodulen, wie z.B. für digitale bzw. analoge Ein-/Ausgabe, Zählerfunktionen, Wegmessung, Positionierung und serielle Kommunikation.
- Die Peripheriemodule können zentral und dezentral betrieben werden.
- Einbindung über GSD-Datei** Die Funktionalität aller Systemkomponenten von VIPA sind in Form von verschiedenen GSD-Dateien verfügbar.
- Da die PROFIBUS-Schnittstelle auch softwareseitig standardisiert ist, können wir auf diesem Weg gewährleisten, dass über die Einbindung einer GSD-Datei die Funktionalität in Verbindung mit dem Siemens SIMATIC Manager jederzeit gegeben ist.
- Für jede Systemfamilie erhalten Sie eine GSD-Datei. Aktuelle GSD-Dateien finden Sie im Service-Bereich unter [www.vipa.de](http://www.vipa.de).

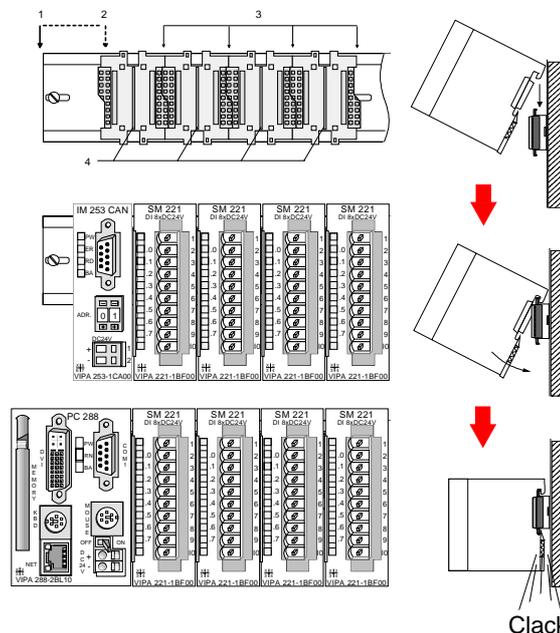
## Allgemeine Beschreibung System 200V

### Aufbau/Maße

- Normprofil-Hutschiene 35mm
- Peripherie-Module mit seitlich versenkbaren Beschriftungsstreifen
- Maße Grundgehäuse:
  - 1fach breit: (HxBxT) in mm: 76x25,4x74 in Zoll: 3x1x3
  - 2fach breit: (HxBxT) in mm: 76x50,8x74 in Zoll: 3x2x3

### Montage

Bitte beachten Sie, dass Sie Kopfmodule, wie CPUs, PC und Koppler nur auf Steckplatz 2 bzw. 1 und 2 (wenn doppelt breit) stecken dürfen.



- [1] Kopfmodul, wie PC, CPU, Buskoppler, wenn doppelt breit
- [2] Kopfmodul, wenn einfach breit
- [3] Peripheriemodule
- [4] Führungsleisten

#### Hinweis

Sie können maximal 32 Module stecken, hierbei ist zu beachten, dass der **Summenstrom** von **3,5A** am Rückwandbus nicht überschritten wird!

Bitte montieren Sie Module mit hoher Stromaufnahme direkt neben das Kopfmodul.

### Betriebssicherheit

- Anschluss über Federzugklemmen an Frontstecker, Aderquerschnitt 0,08...2,5mm<sup>2</sup> bzw. 1,5 mm<sup>2</sup> (18-fach Stecker)
- Vollisolierung der Verdrahtung bei Modulwechsel
- Potenzialtrennung aller Module zum Rückwandbus
- ESD/Burst gemäß IEC 61000-4-2 / IEC 61000-4-4 (bis Stufe 3)
- Schockfestigkeit gemäß IEC 60068-2-6 / IEC 60068-2-27 (1G/12G)
- Schutzklasse IP20

### Umgebungsbedingungen

- Betriebstemperatur: 0 ... +60°C
- Lagertemperatur: -25 ... +70°C
- Relative Feuchte: 5 ... 95% ohne Betauung
- Lüfterloser Betrieb



## Teil 2 Montage und Aufbaurichtlinien

### Überblick

In diesem Kapitel finden Sie alle Informationen, die für den Aufbau und die Verdrahtung einer Steuerung aus den Komponenten des Systems 200V erforderlich sind.

### Inhalt

Thema	Seite
<b>Teil 2 Montage und Aufbaurichtlinien</b> .....	<b>2-1</b>
Übersicht .....	2-2
Montage .....	2-5
Verdrahtung .....	2-8
Einbaumaße .....	2-10
Aufbaurichtlinien .....	2-12

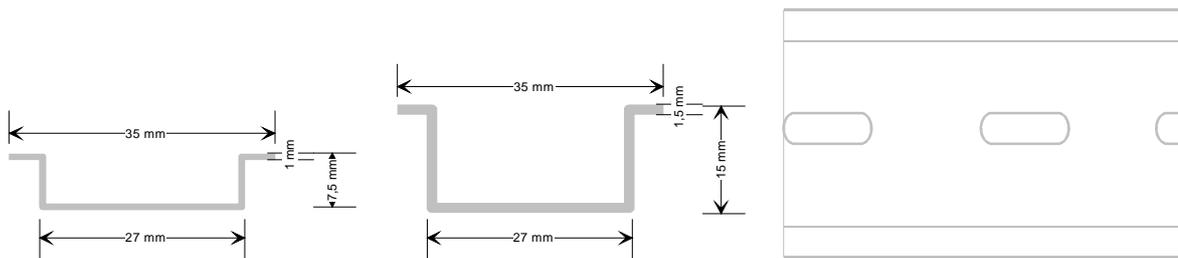
## Übersicht

### Allgemein

Die einzelnen Module werden direkt auf eine Tragschiene montiert und über Rückwandbusverbinder, die vorher in die Profilschiene eingelegt werden, gekoppelt.

### Tragschienen

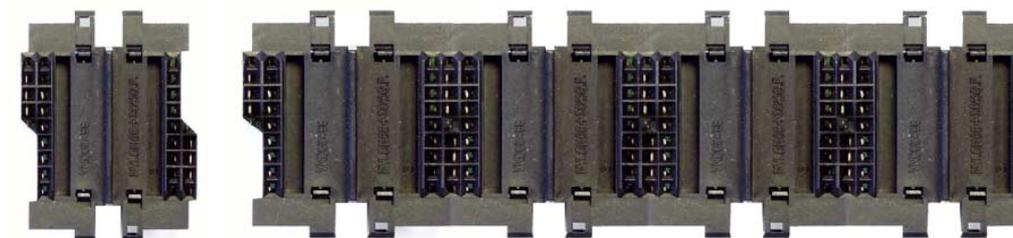
Für die Montage können Sie folgende 35mm-Normprofilschiene verwenden:



### Busverbinder

Für die Kommunikation der Module untereinander wird beim System 200V ein Rückwandbusverbinder eingesetzt. Die Rückwandbusverbinder sind isoliert und bei VIPA in 1-, 2-, 4- oder 8facher Breite erhältlich.

Nachfolgend sehen Sie einen 1fach und einen 4fach Busverbinder:

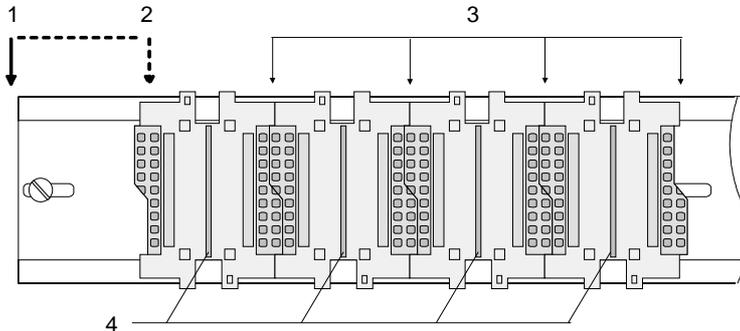


Der Busverbinder wird in die Tragschiene eingelegt, bis dieser sicher einrastet, so dass die Bus-Anschlüsse aus der Tragschiene heraussehen.

**Montage auf Tragschiene**

Die nachfolgende Skizze zeigt einen 4fach-Busverbinder in einer Tragschiene und die Steckplätze für die Module.

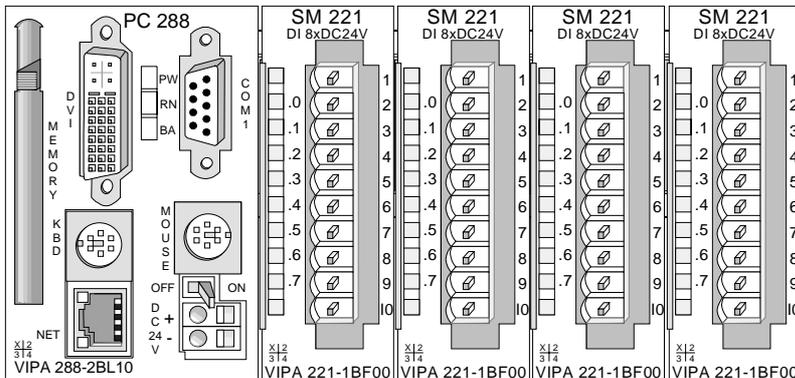
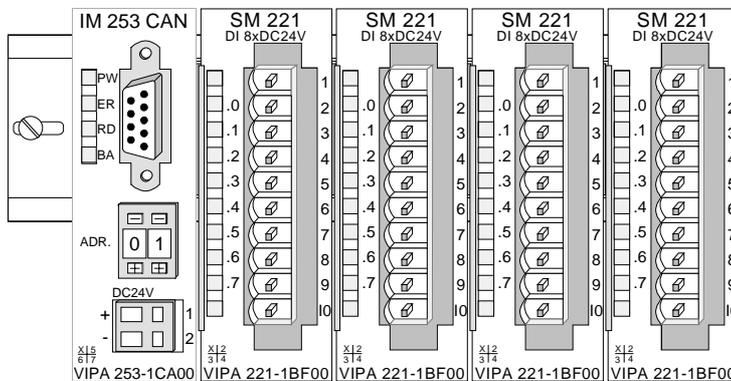
Die einzelnen Modulsteckplätze sind durch Führungsleisten abgegrenzt.



- [1] Kopfmodul, wie PC, CPU, Bus-Koppler, wenn doppelt breit
- [2] Kopfmodul (einfach breit)
- [3] Peripheriemodule
- [4] Führungsleisten

**Hinweis**

Sie können maximal 32 Module stecken. Hierbei ist zu beachten, dass der **Summenstrom** von **3,5A** am Rückwandbus nicht überschritten wird!



**Montage unter Berücksichtigung der Stromaufnahme**

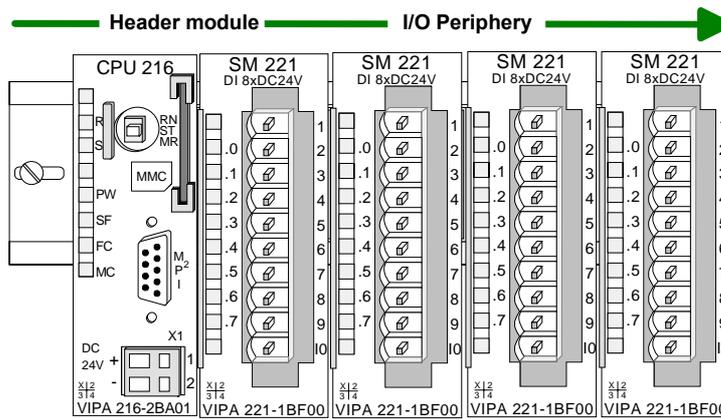
- Verwenden Sie möglichst lange Busverbinder.
- Ordnen Sie Module mit hohem Stromverbrauch direkt rechts neben Ihrem Kopfmodul an. Unter [www.vipa.de/manuals/system200v](http://www.vipa.de/manuals/system200v) finden Sie alle Stromaufnahmen des System 200V in einer Liste zusammengefasst.

**Aufbau waagrecht bzw. senkrecht**

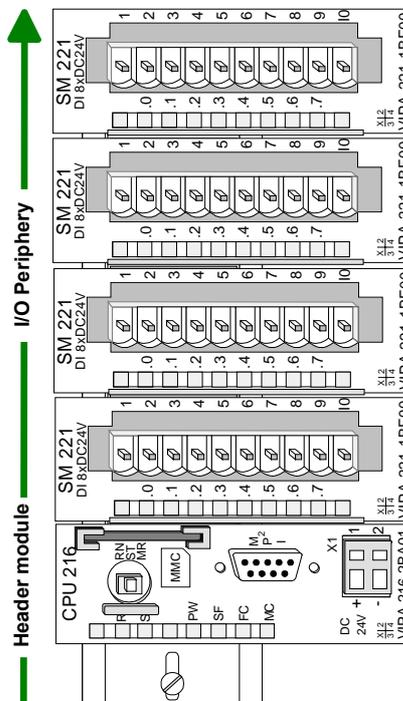
Sie haben die Möglichkeit das System 200V waagrecht oder senkrecht aufzubauen. Beachten Sie bitte die hierbei zulässigen Umgebungstemperaturen:

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

Der waagrechte Aufbau beginnt immer links mit einem Kopfmodul (CPU, Buskoppler, PC); rechts daneben sind die Peripherie-Module zu stecken. Es dürfen maximal 32 Peripherie-Module gesteckt werden.



Der senkrechte Aufbau erfolgt gegen den Uhrzeigersinn um 90° gedreht.

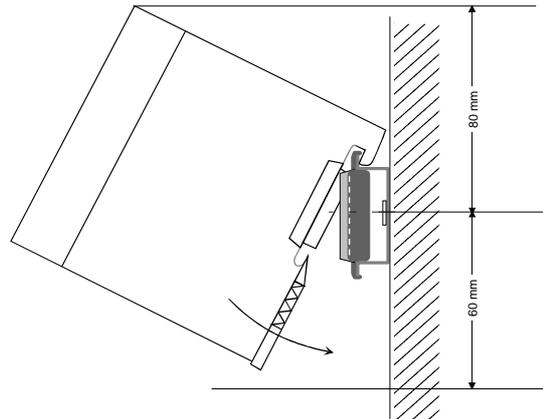


## Montage

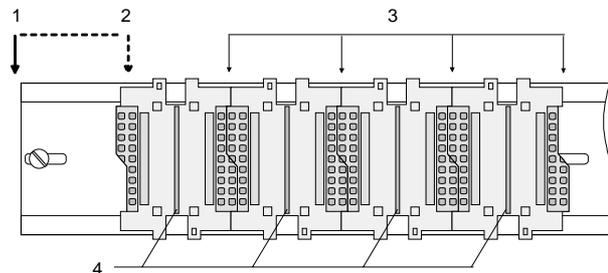


### Bitte bei der Montage beachten!

- Schalten Sie die Stromversorgung aus, bevor Sie Module stecken bzw. abziehen!
- Bitte beachten Sie, dass Sie ab der Mitte der Busschiene nach oben einen Modul-Montageabstand von mindestens 80mm und nach unten von 60mm einhalten.



- Eine Zeile wird immer von links nach rechts aufgebaut und beginnt immer mit einem Kopfmodul (PC, CPU, Buskoppler).



- [1] Kopfmodul, wie PC, CPU, Buskoppler, wenn doppelt breit
- [2] Kopfmodul (einfach breit)
- [3] Peripheriemodule
- [4] Führungsleisten

- Module müssen immer direkt nebeneinander gesteckt werden. Lücken zwischen den Modulen sind nicht zulässig, da ansonsten der Rückwandbus unterbrochen ist.
- Ein Modul ist erst dann gesteckt und elektrisch verbunden, wenn es hörbar einrastet.
- Steckplätze rechts nach dem letzten Modul dürfen frei bleiben.

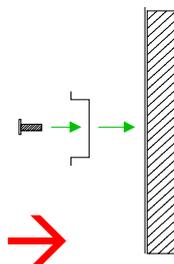


### Hinweis!

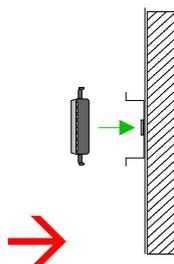
Am Rückwandbus dürfen sich maximal 32 Module befinden. Hierbei ist zu beachten, dass der **Summenstrom** von **3,5A** am Rückwandbus nicht überschritten wird!

**Montage  
Vorgehensweise**

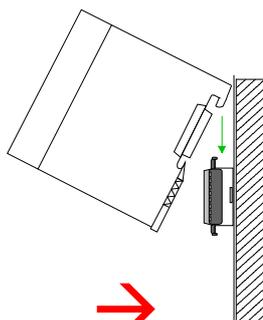
Die nachfolgende Abfolge stellt die Montageschritte in der Seitenansicht dar.



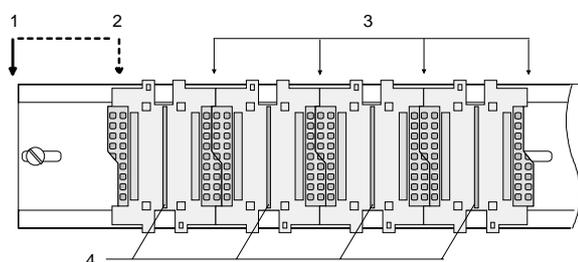
- Montieren Sie die Tragschiene! Bitte beachten Sie, dass Sie ab der Mitte der Busschiene nach oben einen Modul-Montageabstand von mindestens 80mm und nach unten von 60mm einhalten.



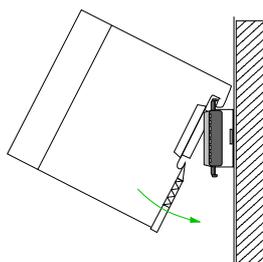
- Drücken Sie den Busverbinder in die Tragschiene, bis dieser sicher einrastet, so dass die Bus-Anschlüsse aus der Tragschiene herauschauen. Sie haben nun die Grundlage zur Montage Ihrer Module.



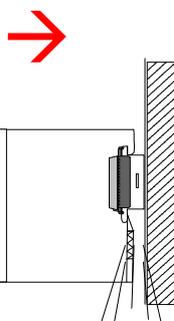
- Beginnen Sie ganz links mit dem Kopfmodul, wie CPU, PC oder Buskoppler und stecken Sie rechts daneben Ihre Peripherie-Module.



- [1] Kopfmodul, wie PC, CPU, Buskoppler, wenn doppelt breit
- [2] Kopfmodul (einfach breit)
- [3] Peripheriemodule
- [4] Führungsleisten



- Setzen Sie das zu steckende Modul von oben in einem Winkel von ca. 45Grad auf die Tragschiene und drehen Sie das Modul nach unten, bis es hörbar auf der Tragschiene einrastet. Nur bei eingerasteten Modulen ist eine Verbindung zum Rückwandbus sichergestellt.



Clack

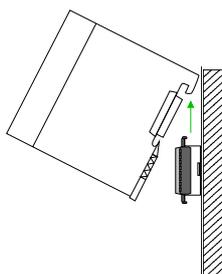
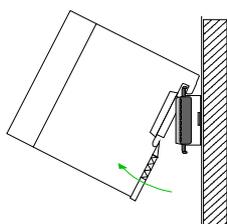
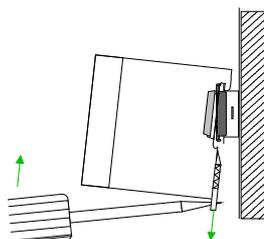
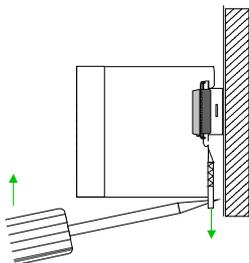
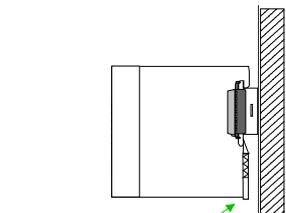


**Achtung!**

Module dürfen nur im spannungslosen Zustand gesteckt bzw. gezogen werden!

### Demontage Vorgehensweise

Die nachfolgende Abfolge stellt die Schritte zur Demontage in der Seitenansicht dar.



- Zur Demontage befindet sich am Gehäuseunterteil eine gefederter Demontageschlitz.
- Stecken Sie, wie gezeigt, einen Schraubendreher in den Demontageschlitz.

- Durch Druck des Schraubendrehers nach oben wird das Modul entriegelt.

- Ziehen Sie nun das Modul nach vorn und ziehen Sie das Modul mit einer Drehung nach oben ab.



#### Achtung!

Module dürfen nur im spannungslosen Zustand gesteckt bzw. gezogen werden!

Bitte beachten Sie, dass durch die Demontage von Modulen der Rückwandbus an der entsprechenden Stelle unterbrochen wird!

## Verdrahtung

### Übersicht

Die meisten Peripherie-Module besitzen einen 10poligen bzw. 18poligen Steckverbinder. Über diesen Steckverbinder werden Signal- und Versorgungsleitungen mit den Modulen verbunden.

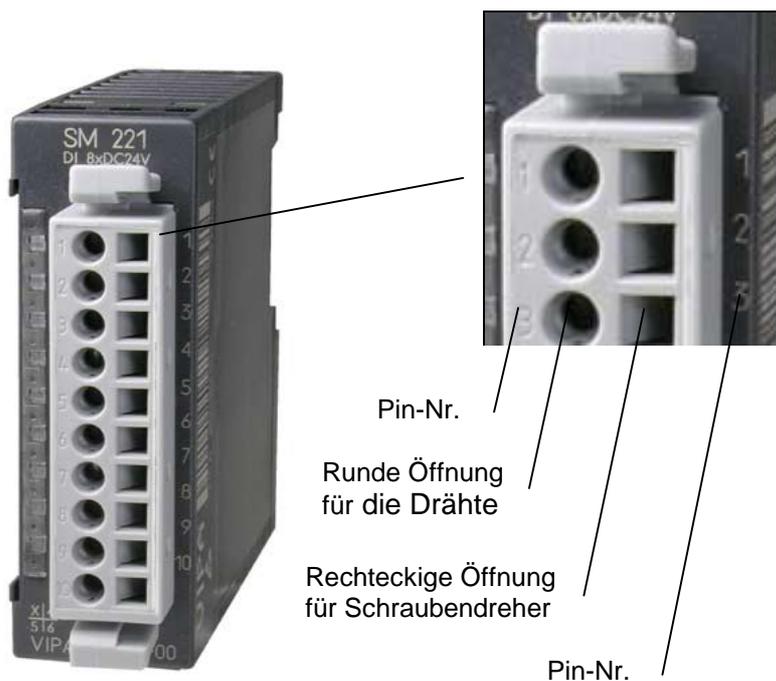
Bei der Verdrahtung werden Steckverbinder mit Federklemmtechnik eingesetzt.

Die Verdrahtung mit Federklemmtechnik ermöglicht einen schnellen und einfachen Anschluss Ihrer Signal- und Versorgungsleitungen.

Im Gegensatz zur Schraubverbindung, ist diese Verbindungsart erschütterungssicher. Die Steckerbelegung der Peripherie-Module finden Sie in der Beschreibung zu den Modulen.

Sie können Drähte mit einem Querschnitt von  $0,08\text{mm}^2$  bis  $2,5\text{mm}^2$  (bis  $1,5\text{mm}^2$  bei 18poligen) anschließen.

Folgende Abbildung zeigt ein Modul mit einem 10poligen Steckverbinder.

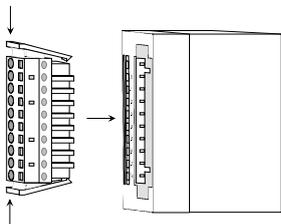


### Hinweis!

Die Federklemme wird zerstört, wenn Sie den Schraubendreher in die Öffnung für die Leitungen stecken!

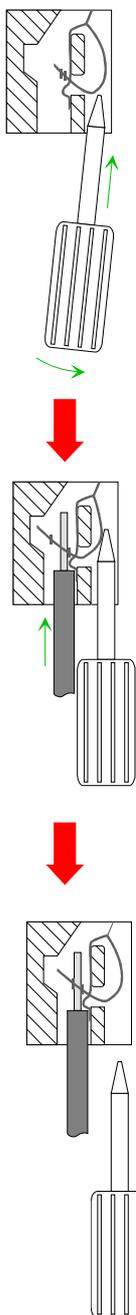
Drücken Sie den Schraubendreher nur in die rechteckigen Öffnungen des Steckverbinders!

## Verdrahtung Vorgehensweise



- Stecken Sie den Steckverbinder auf das Modul bis dieser hörbar einrastet. Drücken Sie hierzu während des Steckens, wie gezeigt, die beiden Verriegelungsklinken zusammen. Der Steckverbinder ist nun in einer festen Position und kann leicht verdrahtet werden.

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



- Zum Verdrahten stecken Sie, wie in der Abbildung gezeigt, einen passenden Schraubendreher leicht schräg in die rechteckige Öffnung.
- Zum Öffnen der Kontaktfeder müssen Sie den Schraubendreher in die entgegengesetzte Richtung drücken und halten.
- Führen Sie durch die runde Öffnung Ihren abisolierten Draht ein. Sie können Drähte mit einem Querschnitt von  $0,08\text{mm}^2$  bis  $2,5\text{mm}^2$  (bei 18poligen Steckverbindern bis  $1,5\text{mm}^2$ ) anschließen.
- Durch Entfernen des Schraubendrehers wird der Draht über einen Federkontakt sicher mit dem Steckverbinder verbunden.



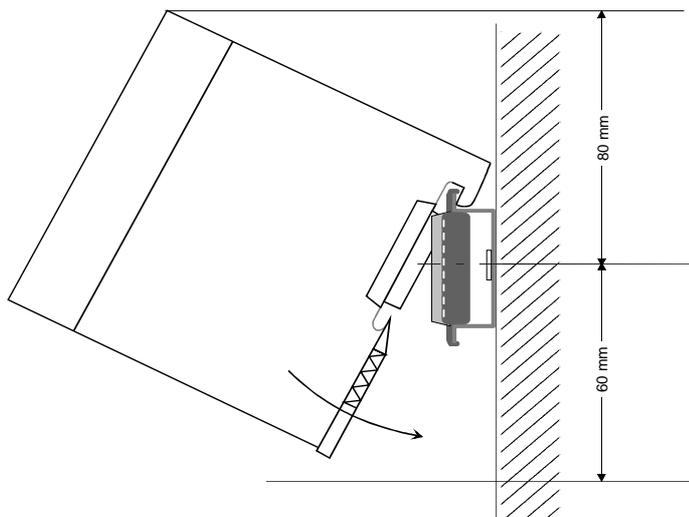
Verdrahten Sie zuerst die Versorgungsleitungen (Spannungsversorgung) und dann die Signalleitungen (Ein- und Ausgänge)!

# Einbaumaße

**Übersicht** Hier finden Sie alle wichtigen Maße des System 200V.

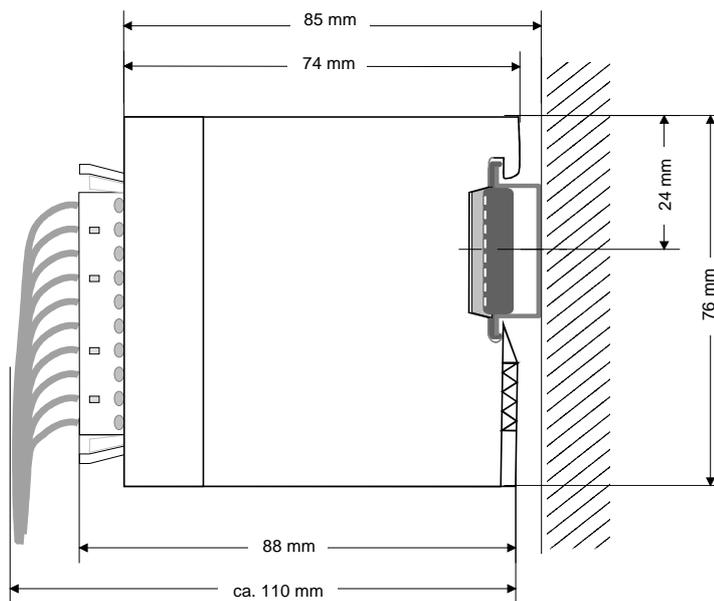
**Maße Grundgehäuse**  
 1fach breit (HxBxT) in mm: 76 x 25,4 x 74  
 2fach breit (HxBxT) in mm: 76 x 50,8 x 74

## Montagemaße

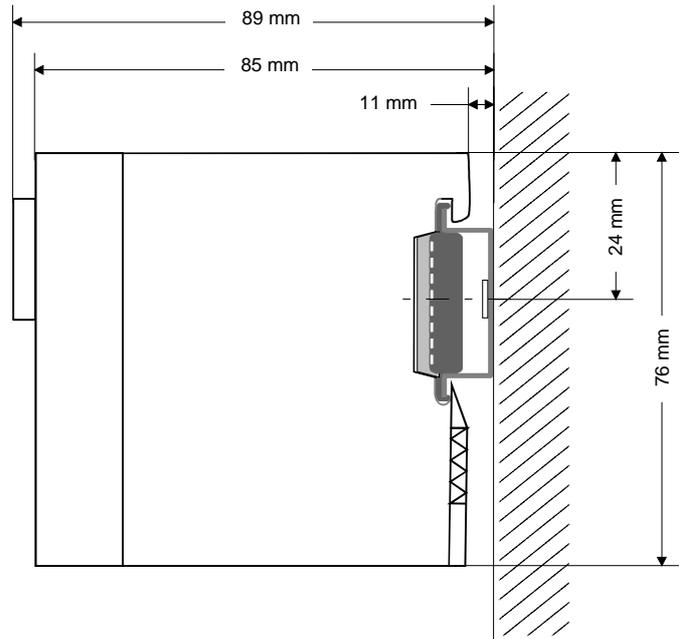


## Maße montiert und verdrahtet

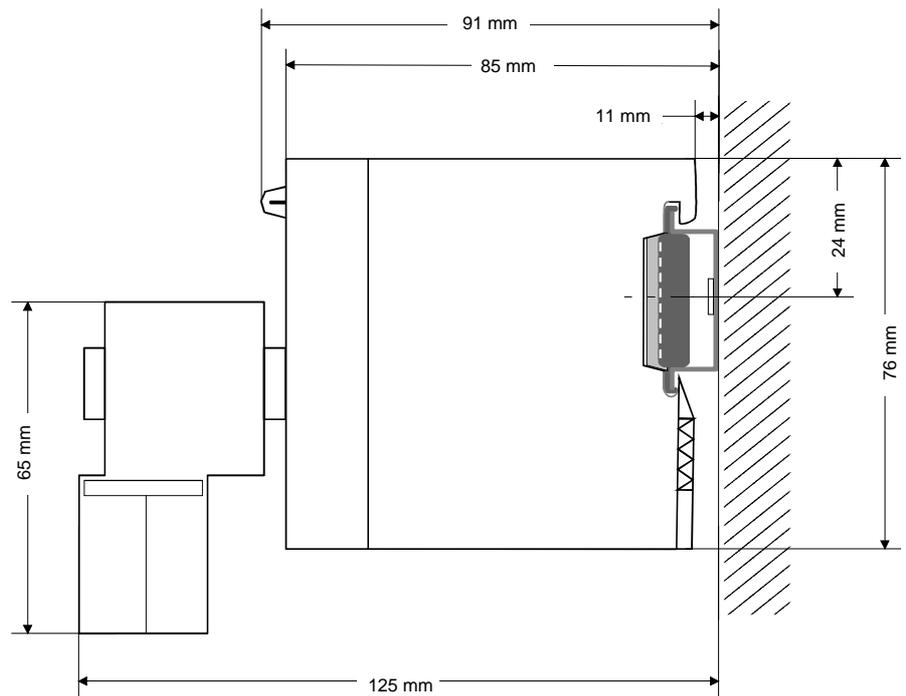
Ein- / Ausgabe-  
 module



Funktionsmodule



CPUs (hier mit  
VIPA EasyConn)



## Aufbaurichtlinien

<b>Allgemeines</b>	Die Aufbaurichtlinien enthalten Informationen über den störsicheren Aufbau des System 200V. Es wird beschrieben, wie Störungen in Ihre Steuerung gelangen können, wie die elektromagnetische Verträglichkeit (EMV) sicher gestellt werden kann und wie bei der Schirmung vorzugehen ist.
<b>Was bedeutet EMV?</b>	<p>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.</p> <p>Alle System 200V Komponenten sind für den Einsatz in rauen 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.</p>
<b>Mögliche Störeinträge</b>	<p>Elektromagnetische Störungen können sich auf unterschiedlichen Pfaden in Ihre Steuerung einkoppeln:</p> <ul style="list-style-type: none"><li>• Felder</li><li>• E/A-Signalleitungen</li><li>• Bussystem</li><li>• Stromversorgung</li><li>• Schutzleitung</li></ul> <p>Je nach Ausbreitungsmedium (leitungsgebunden oder -ungebunden) und Entfernung zur Störquelle gelangen Störungen über unterschiedliche Kopplungsmechanismen in Ihre Steuerung.</p> <p>Man unterscheidet:</p> <ul style="list-style-type: none"><li>• galvanische Kopplung</li><li>• kapazitive Kopplung</li><li>• induktive Kopplung</li><li>• Strahlungskopplung</li></ul>

**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.
  - Beschalten Sie alle Induktivitäten mit Löschgliedern, die von System 200V Modulen angesteuert werden.
  - Benutzen Sie zur Beleuchtung von Schränken Glühlampen und vermeiden Sie Leuchtstofflampen.
- 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 dem System 200V sternförmig mit dem Erde/Schutzleitersystem. Sie vermeiden so die Bildung von Erdschleifen.
  - Verlegen Sie bei Potenzialdifferenzen zwischen Anlagenteilen und Schränken ausreichend dimensionierte Potenzialausgleichsleitungen.

## 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 Potenzialausgleichsleitung 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 zum System 200V Modul weiter, legen Sie ihn dort jedoch **nicht** erneut auf!



### Bitte bei der Montage beachten!

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

Abhilfe: Potenzialausgleichsleitung

## Teil 3 Hardwarebeschreibung

### Überblick

Die CPUs 21x erhalten Sie in verschiedenen Ausführungen, auf die in diesem Kapitel weiter eingegangen werden soll. Neben einer Übersicht der Hardwarevarianten finden Sie hier auch eine Beschreibung der verschiedenen Komponenten der System 200V CPU-Familie.

Mit den technischen Daten wird das Kapitel abgeschlossen.

### Inhalt

Thema	Seite
<b>Teil 3 Hardwarebeschreibung</b> .....	<b>3-1</b>
Einleitung .....	3-2
Arbeitsweise einer CPU.....	3-3
Funktionssicherheit der VIPA CPUs .....	3-5
Einsatzbereiche.....	3-6
Systemübersicht.....	3-8
Aufbau.....	3-16
Komponenten.....	3-18
Blockschaltbild .....	3-27
Technische Daten .....	3-28

## Einleitung

### Allgemeines

Eine CPU ist ein intelligentes Modul. Hier werden Ihre Steuerungsprogramme ausgeführt. Je nachdem wie leistungsfähig Ihr System sein soll, können Sie zwischen drei CPUs wählen. Je leistungsfähiger die gewählte CPU, desto größer ist der Anwenderspeicher.

Diese CPU 21x sind für kleine und mittlere Anwendungen mit integriertem 24V-Netzteil. Die CPUs enthalten einen Standardprozessor mit internem Programmspeicher für die Speicherung des Anwenderprogramms. Weiter besitzt jede CPU 21x an der Front einen Steckplatz für ein Speicher-Modul.

Jede CPU hat einen MPI-Anschluss und ist befehlskompatibel zu STEP<sup>®</sup>7 von Siemens. Die CPU-Serie 214...216 deckt den Leistungsbereich der STEP<sup>®</sup>7 CPU-Serie von Siemens ab.

Mit dieser CPU-Serie haben Sie Zugriff auf die Peripherie-Module des System 200V. Sie können über standardisierte Befehle und Programme Sensoren abfragen und Aktoren steuern. Sie können maximal 32 Module ansprechen.

### Ausführungen

Die CPUs 21x sind bei VIPA in 3 CPU-Leistungsklassen in je 8 Varianten erhältlich:

- **CPU 21x** SPS-CPU
- **CPU 21x-2BT02** SPS-CPU mit Ethernet-Anschaltung für H1 / TCP-IP
- **CPU 21x-2BT10** SPS-CPU mit Ethernet-Anschaltung für TCP-IP
- **CPU 21xDP** SPS-CPU mit PROFIBUS-Slave
- **CPU 21xDPM** SPS-CPU mit PROFIBUS-Master
- **CPU 21xSER-1** SPS-CPU mit 1 seriellen Schnittstelle
- **CPU 21xSER-2** SPS-CPU mit 2 seriellen Schnittstellen
- **CPU 21xCAN** SPS-CPU mit CANopen-Master

Alle CPU 21x sind in 3 CPU-Leistungsklassen lieferbar als 214, 215 und 216. Innerhalb dieser 3 Leistungsklassen gibt es optisch keinen Unterschied. Mit steigender Nummer steigt der Leistungsumfang einer CPU 21x.



### Hinweis!

Soweit nichts anderes erwähnt ist, beziehen sich die in diesem Handbuch gemachten Angaben auf alle CPUs der CPU 21x-Familie von VIPA!

## Arbeitsweise einer CPU

---

### Allgemein

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 Alarmereignis schnell reagiert werden muss, unterbricht zur Bearbeitung dieser hoch-prioren Ereignisse die CPU die zyklische Bearbeitung, reagiert auf diese Ereignisse und setzt danach die zyklische Bearbeitung wieder fort. Die zyklische Bearbeitung hat daher die niedrigste Priorität.

---

### Programme der CPU 21x

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.

### Anwender- programm

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

---

**Operanden der CPU 21x**

Die CPU 21x 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

## Funktionssicherheit der VIPA CPUs

### Sicherheitsmechanismen

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.

VIPA CPUs sind funktionssicher ausgelegt und besitzen folgende Systemeigenschaften:

Ereignis	betrifft	Effekt
RUN → STOP	allgemein zentrale digitale Ausgänge zentrale analoge Ausgänge  dezentrale Ausgänge dezentrale Eingänge	BASP ( <b>B</b> efehls- <b>A</b> usgabe- <b>S</b> perre) wird gesetzt. Die Ausgänge werden auf 0V gesetzt. Die Spannungsversorgung für die Ausgabekanäle wird abgeschaltet. Die Ausgänge werden auf 0V gesetzt. Die Eingänge werden vom Slave konstant gelesen und die aktuellen Werte zur Verfügung gestellt.
STOP → RUN bzw. Netz-Ein	allgemein  zentrale analoge Ausgänge  dezentrale Eingänge	Zuerst wird das PAE gelöscht, danach erfolgt der Aufruf des OB100. Nachdem dieser abgearbeitet ist, wird das BASP zurückgesetzt und der Zyklus gestartet mit: PAA löschen → PAE lesen → OB1. Das Verhalten der Ausgänge bei Neustart kann voreingestellt werden. Die Eingänge werden vom Slave konstant gelesen und die aktuellen Werte zur Verfügung gestellt.
RUN	allgemein	Der Programmablauf ist zyklisch und damit vorhersehbar: PAE lesen → OB1 → PAA schreiben.

PAE: = Prozessabbild der Eingänge

PAA: = Prozessabbild der Ausgänge

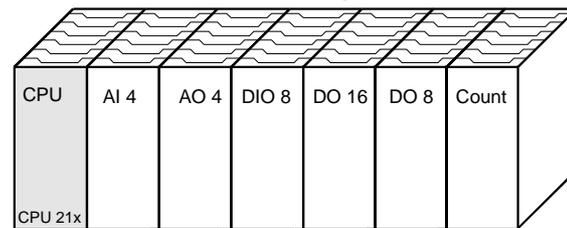
## Einsatzbereiche

### Übersicht

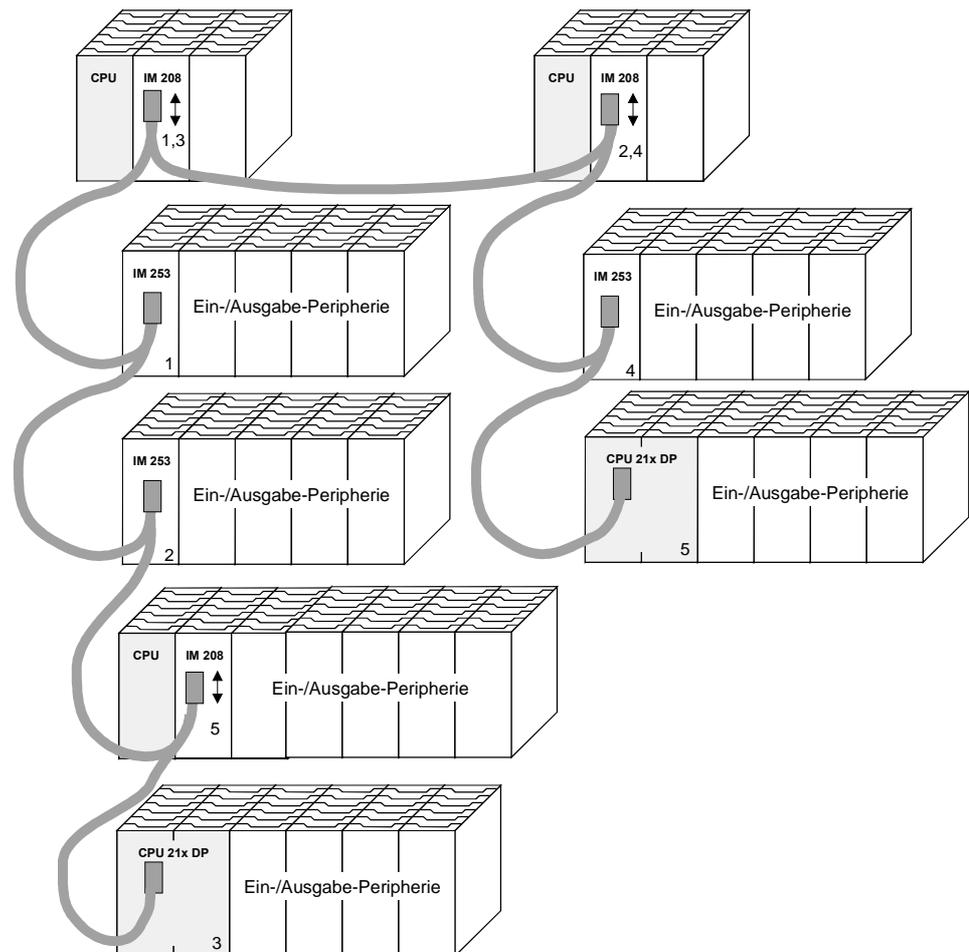
Mit dieser CPU-Serie haben Sie Zugriff auf die Peripherie-Module des System 200V von VIPA. Sie können über standardisierte Befehle und Programme Sensoren abfragen und Aktoren steuern. Eine CPU kann zentral maximal 32 Module ansprechen.

### Anwendungs- beispiel

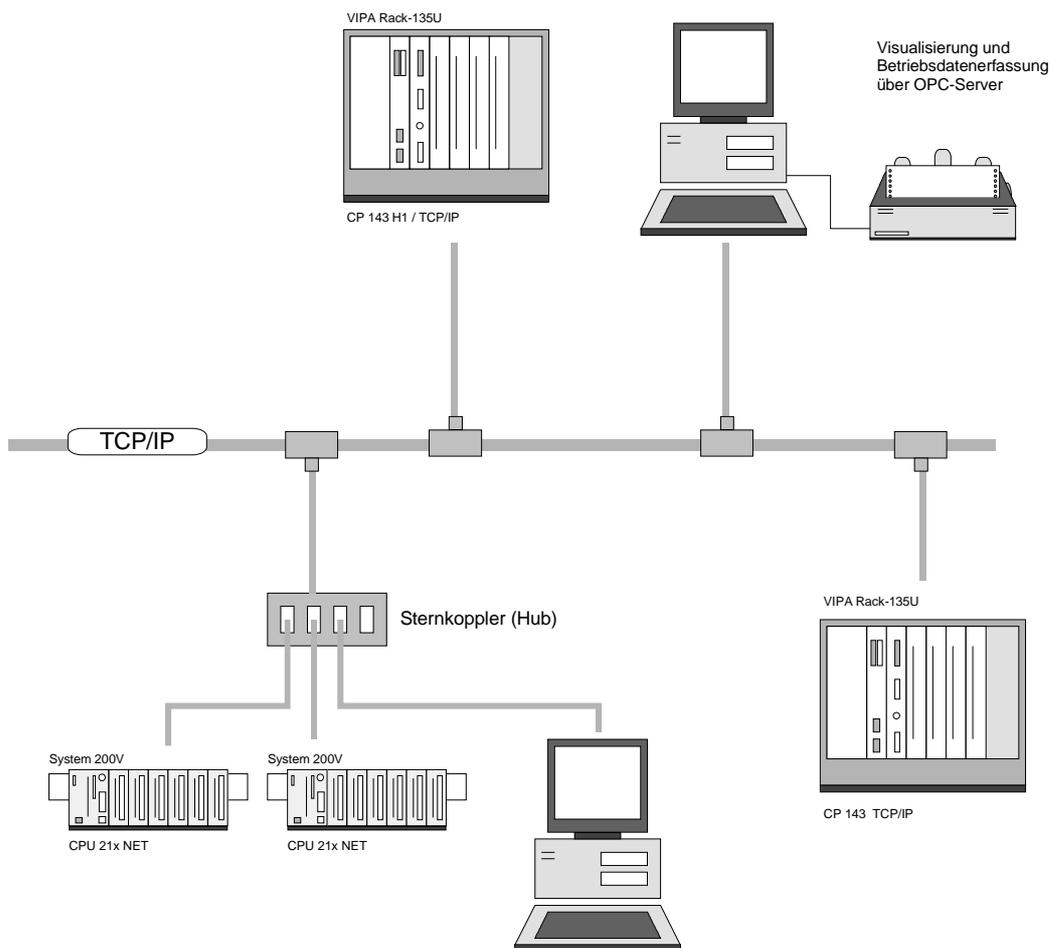
#### Zentrale Kompakt-Lösung



#### Dezentraler Einsatz unter PROFIBUS



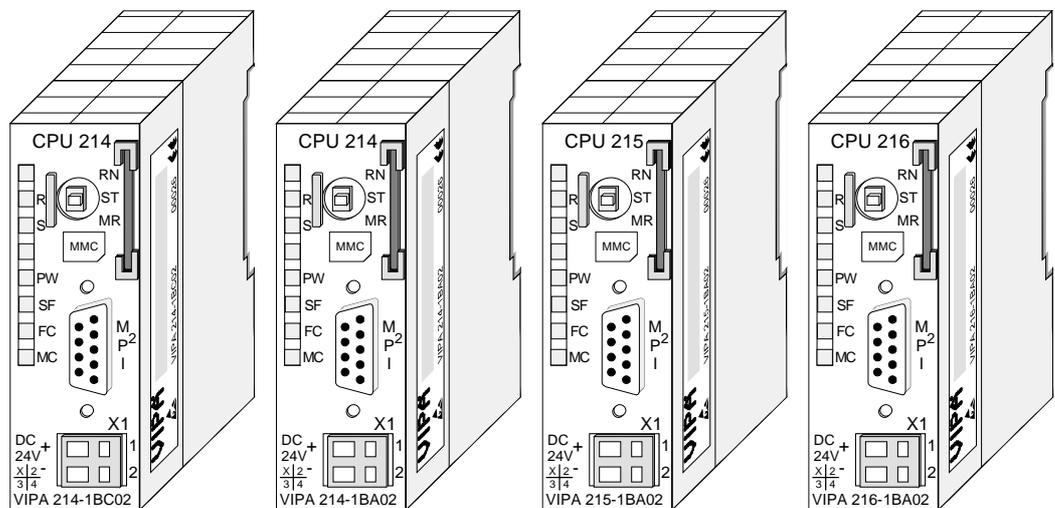
Einsatz unter TCP/IP



## Systemübersicht

### CPU 21x

- Befehlskompatibel zu STEP<sup>®</sup>7 von Siemens
- Projektierung über den Siemens SIMATIC Manager
- Integrierter V-Bus-Kontroller zur Steuerung der System 200V Peripherie-Module
- Integriertes 24V-Netzteil
- Gesamtadressraum: 1024 Byte Eingänge, 1024 Byte Ausgänge (je 128 Byte Prozessabbild)
- Arbeitsspeicher 48...128kByte "on board"
- Ladespeicher 80...192kByte "on board"
- Steckplatz für MMC (für Anwenderprogramm)
- Akkugepufferte Uhr
- MP<sup>2</sup>I-Schnittstelle zur Datenübertragung
- Status-LEDs für Betriebszustand und Diagnose
- 256 Zeiten
- 256 Zähler
- 8192 Merker-Bits

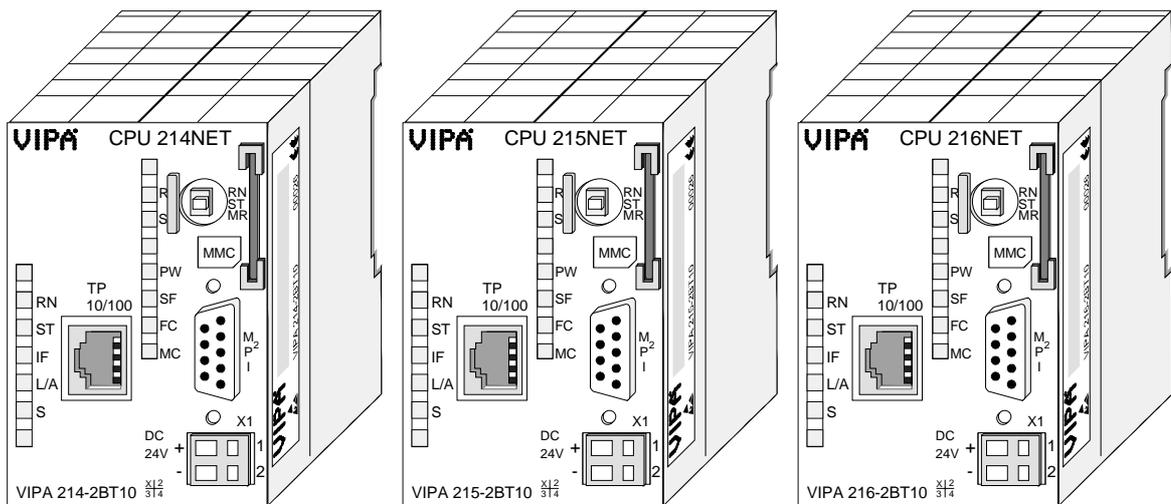


### Bestelldaten CPU 21x

Typ	Bestellnummer	Beschreibung
CPU 214C	VIPA 214-1BC02	SPS CPU 214 mit 32/40kByte A/L-Speicher
CPU 214	VIPA 214-1BA02	SPS CPU 214 mit 48/80kByte A/L-Speicher
CPU 215	VIPA 215-1BA02	SPS CPU 215 mit 96/144kByte A/L-Speicher
CPU 216	VIPA 216-1BA02	SPS CPU 216 mit 128/192kByte A/L-Speicher

**CPU 21x-2BT10** Wie CPU 21x zusätzlich mit:

- Integriertem Ethernet-CP 243 (kompatibel zum CP 343)
- Direktem Anschluss an Twisted-Pair-Ethernet über RJ45
- Protokolle TCP/IP, UDP und RFC1006
- Übertragungsrate 10/100MBit/s
- PG/OP-Kanal
- CP projektierbar mit NetPro von Siemens

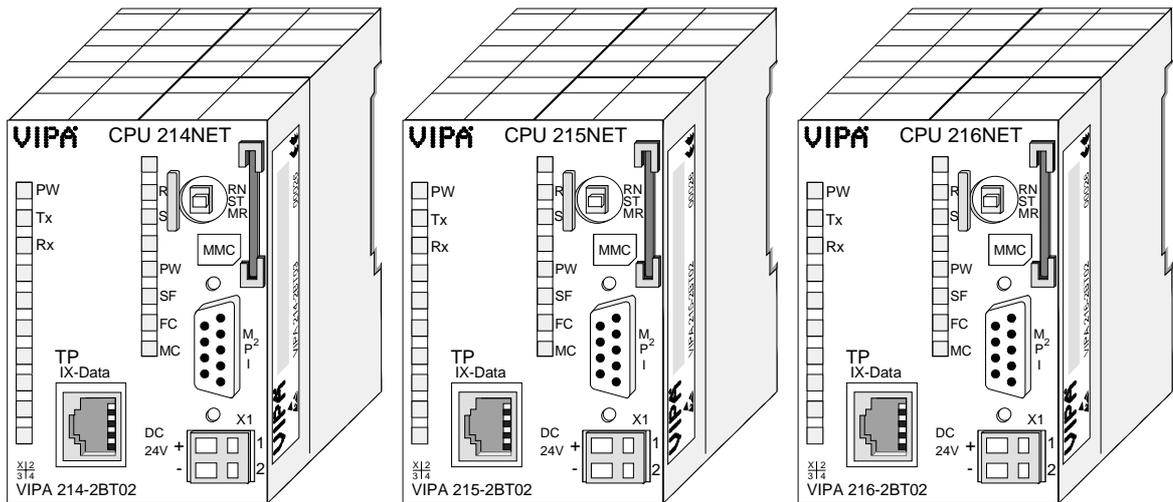


**Bestelldaten  
CPU 21xNET**

Typ	Bestellnummer	Beschreibung
CPU 214NET	VIPA 214-2BT10	SPS CPU 214 mit Ethernet-Anschaltung und 48/80kByte A/L-Speicher
CPU 215NET	VIPA 215-2BT10	SPS CPU 215 mit Ethernet-Anschaltung und 96/144kByte A/L-Speicher
CPU 216NET	VIPA 216-2BT10	SPS CPU 216 Ethernet-Anschaltung und 128/192kByte A/L-Speicher

**CPU 21x-2BT02** Wie CPU 21x zusätzlich mit:

- Integriertem Ethernet-CP 243
- Direktem Anschluss an Twisted-Pair-Ethernet über RJ45
- Protokolle H1, TCP/IP, UDP
- Übertragungsrate 10MBit/s
- CP parametrierbar mit WinNCS



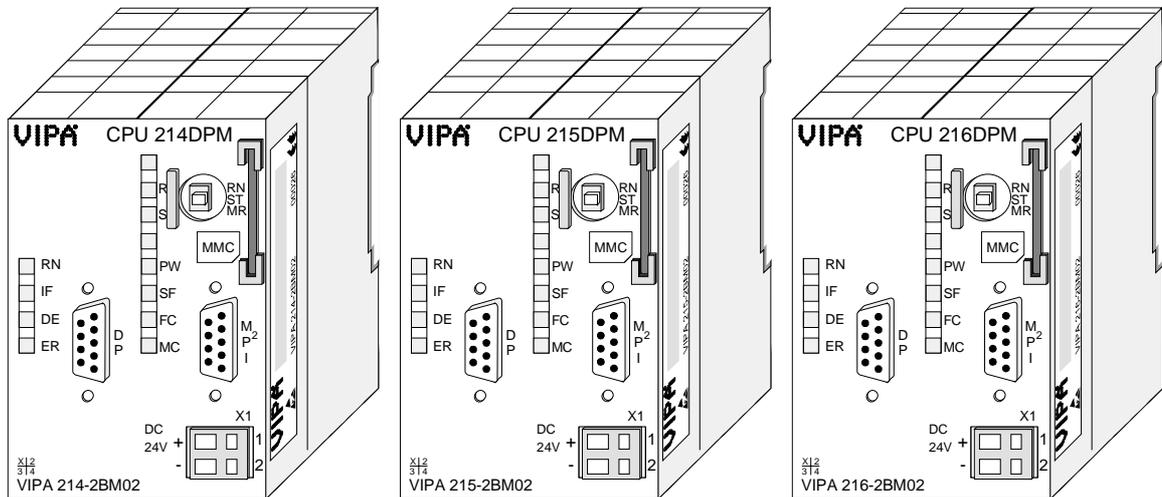
**Bestelldaten  
CPU 21xNET**

Typ	Bestellnummer	Beschreibung
CPU 214NET	VIPA 214-2BT02	SPS CPU 214 mit Ethernet-Anschaltung und 48/80kByte A/L-Speicher
CPU 215NET	VIPA 215-2BT02	SPS CPU 215 mit Ethernet-Anschaltung und 96/144kByte A/L-Speicher
CPU 216NET	VIPA 216-2BT02	SPS CPU 216 Ethernet-Anschaltung und 128/192kByte A/L-Speicher

**CPU 21xDPM**

Wie CPU 21x zusätzlich mit:

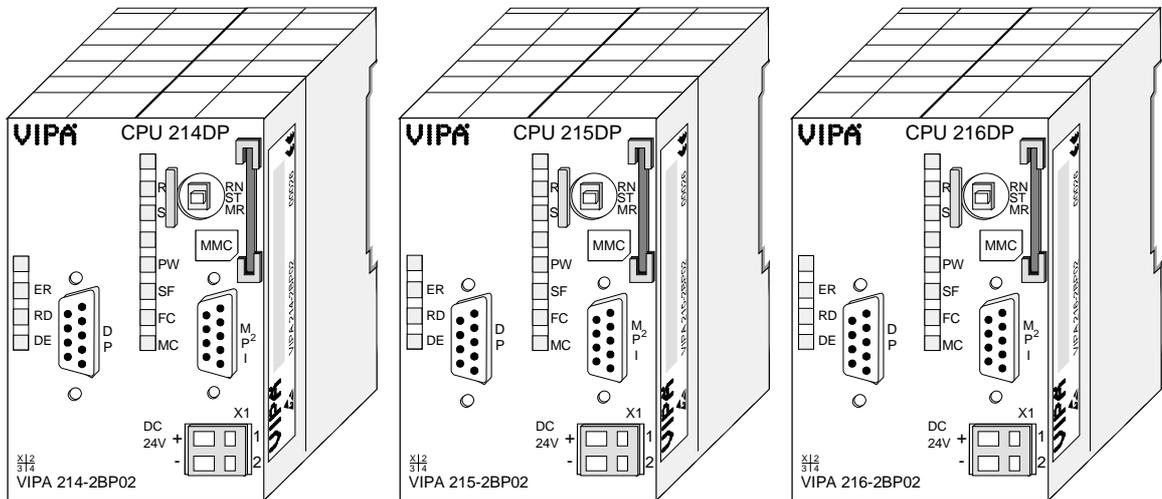
- Integriertem PROFIBUS-DP-Master
- LEDs für PROFIBUS-Status und -Diagnose

**Bestelldaten  
CPU 21xDPM**

Typ	Bestellnummer	Beschreibung
CPU 214DPM	VIPA 214-2BM02	SPS CPU 214 mit PROFIBUS-DP-Master und 48/80kByte A/L-Speicher
CPU 215DPM	VIPA 215-2BM02	SPS CPU 215 mit PROFIBUS-DP-Master und 96/144kByte A/L-Speicher
CPU 216DPM	VIPA 216-2BM02	SPS CPU 216 mit PROFIBUS-DP-Master und 128/192kByte A/L-Speicher

**CPU 21xDP** Wie CPU 21x zusätzlich mit:

- Integriertem PROFIBUS-Slave
- LEDs für PROFIBUS-Status und -Diagnose



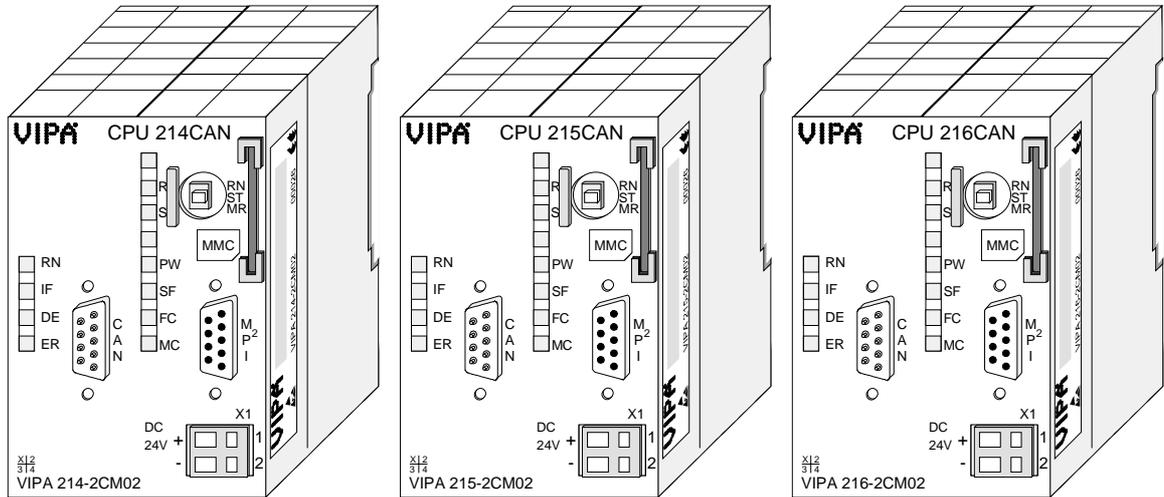
**Bestelldaten  
CPU 21xDP**

Typ	Bestellnummer	Beschreibung
CPU 214DP	VIPA 214-2BP02	SPS CPU 214 mit PROFIBUS-Slave und 48/80kByte A/L-Speicher
CPU 215DP	VIPA 215-2BP02	SPS CPU 215 mit PROFIBUS-Slave und 96/144kByte A/L-Speicher
CPU 216DP	VIPA 216-2BP02	SPS CPU 216 mit PROFIBUS-Slave und 128/192kByte A/L-Speicher

**CPU 21xCAN**

Wie CPU 21x zusätzlich mit:

- Integriertem CANopen-Master
- LEDs für CAN-Status und -Diagnose

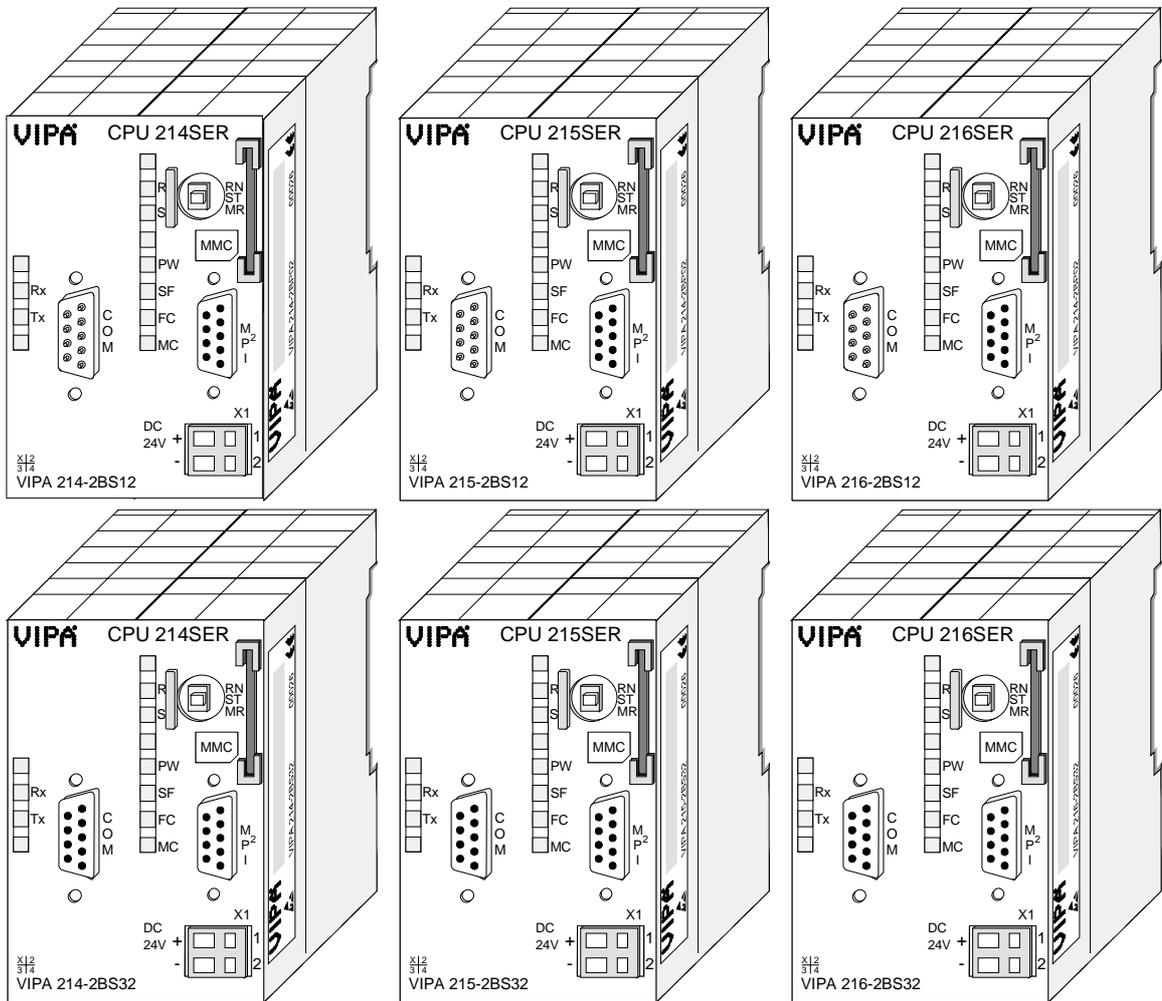


**Bestelldaten  
CPU 21xCAN**

Typ	Bestellnummer	Beschreibung
CPU 214CAN	VIPA 214-2CM02	SPS CPU 214 mit CAN-Master und 48/80kByte A/L-Speicher
CPU 215CAN	VIPA 215-2CM02	SPS CPU 215 mit CAN-Master und 96/144kByte A/L-Speicher
CPU 216CAN	VIPA 216-2CM02	SPS CPU 216 mit CAN-Master und 128/192kByte A/L-Speicher

**CPU 21xSER-1** Wie CPU 21x zusätzlich mit:

- Serieller Kommunikation über COM-Schnittstelle (RS232 oder RS485)
- LEDs für Kommunikation

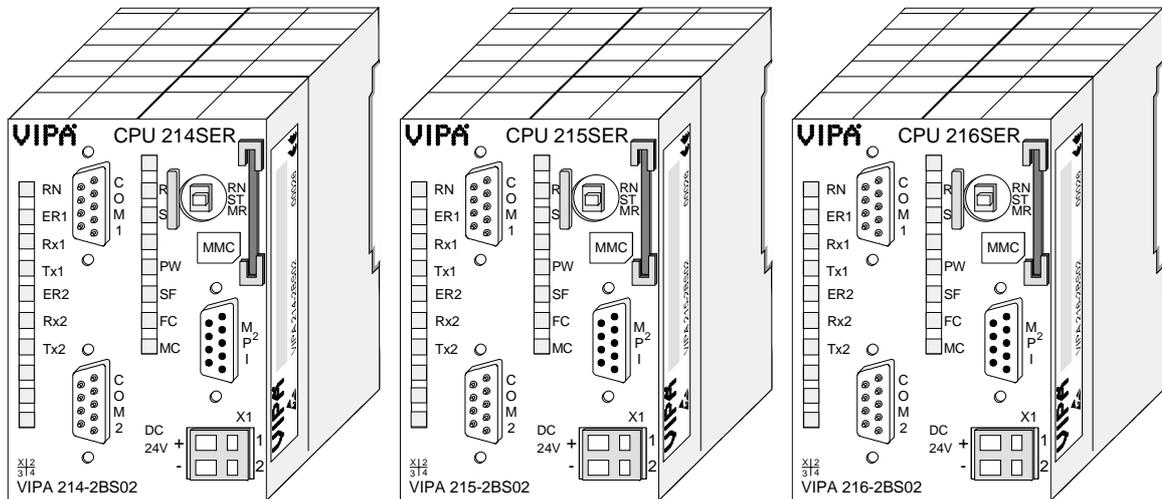


**Bestelldaten  
CPU 21xSER-1**

Typ	Bestellnummer	Beschreibung
CPU 214SER	VIPA 214-2BS12	SPS CPU 214 mit 1xRS232-Schnittstellen und 48/80kByte A/L-Speicher
CPU 215SER	VIPA 215-2BS12	SPS CPU 215 mit 1xRS232-Schnittstellen und 96/144kByte A/L-Speicher
CPU 216SER	VIPA 216-2BS12	SPS CPU 216 mit 1xRS232-Schnittstellen und 128/192kByte A/L-Speicher
CPU 214SER	VIPA 214-2BS32	SPS CPU 214 mit 1xRS485-Schnittstellen und 48/80kByte A/L-Speicher
CPU 215SER	VIPA 215-2BS32	SPS CPU 215 mit 1xRS485-Schnittstellen und 96/144kByte A/L-Speicher
CPU 216SER	VIPA 216-2BS32	SPS CPU 216 mit 1xRS485-Schnittstellen und 128/192kByte A/L-Speicher

**CPU 21xSER-2** Wie CPU 21x zusätzlich mit:

- Serieller Kommunikation über 2 RS232-Schnittstellen
- LEDs für Kommunikation

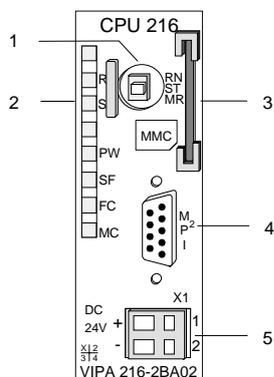


**Bestelldaten  
CPU 21xSER-2**

Typ	Bestellnummer	Beschreibung
CPU 214SER	VIPA 214-2BS02	SPS CPU 214 mit 2xRS232-Schnittstellen und 48/80kByte A/L-Speicher
CPU 215SER	VIPA 215-2BS02	SPS CPU 215 mit 2xRS232-Schnittstellen und 96/144kByte A/L-Speicher
CPU 216SER	VIPA 216-2BS02	SPS CPU 216 mit 2xRS232-Schnittstellen und 128/192kByte A/L-Speicher

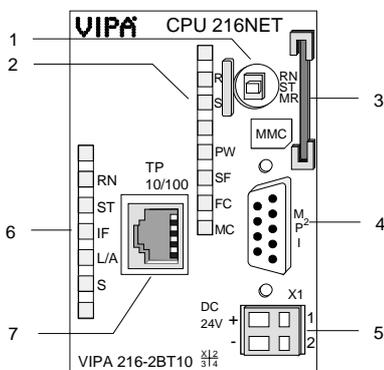
# Aufbau

## Frontansicht CPU 21x



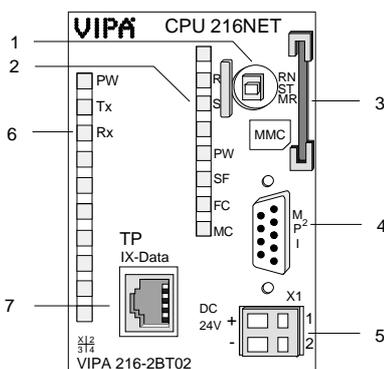
- [1] Betriebsarten-Schalter RUN/STOP/RESET
- [2] LEDs Statusanzeige CPU
- [3] Steckplatz MMC-Speicherkarte
- [4] MP<sup>2</sup>I-Schnittstelle
- [5] Anschluss für DC 24V-Spannungsversorgung

## Frontansicht CPU 21x-2BT10



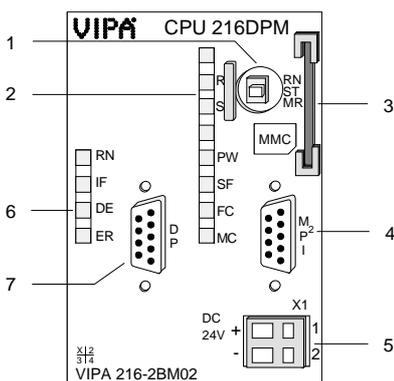
- [1] Betriebsarten-Schalter RUN/STOP/RESET
- [2] LEDs Statusanzeige CPU
- [3] Steckplatz MMC-Speicherkarte
- [4] MP<sup>2</sup>I-Schnittstelle
- [5] Anschluss für DC 24V-Spannungsversorgung
- [6] LEDs Statusanzeige Ethernet
- [7] RJ45-Buchse für Twisted Pair Ethernet

## Frontansicht CPU 21x-2BT02



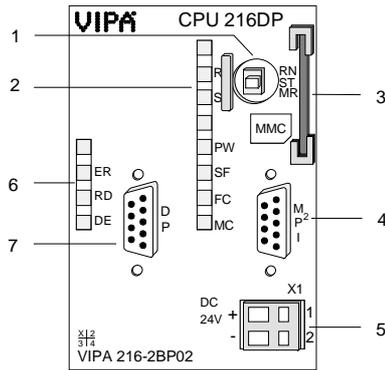
- [1] Betriebsarten-Schalter RUN/STOP/RESET
- [2] LEDs Statusanzeige CPU
- [3] Steckplatz MMC-Speicherkarte
- [4] MP<sup>2</sup>I-Schnittstelle
- [5] Anschluss für DC 24V-Spannungsversorgung
- [6] LEDs Statusanzeige Ethernet
- [7] RJ45-Buchse für Twisted Pair Ethernet

## Frontansicht CPU 21xDPM



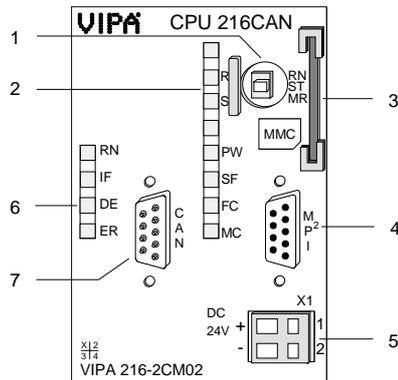
- [1] Betriebsarten-Schalter RUN/STOP/RESET
- [2] LEDs Statusanzeige CPU
- [3] Steckplatz MMC-Speicherkarte
- [4] MP<sup>2</sup>I-Schnittstelle
- [5] Anschluss für DC 24V-Spannungsversorgung
- [6] LEDs Statusanzeige PROFIBUS-DP-Master
- [7] PROFIBUS-DP-Schnittstelle

**Frontansicht  
CPU 21xDP**



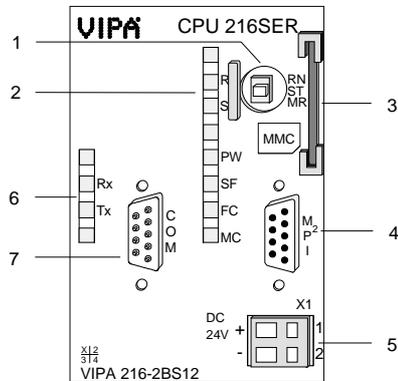
- [1] Betriebsarten-Schalter  
RUN/STOP/RESET
- [2] LEDs Statusanzeige CPU
- [3] Steckplatz MMC-Speicherkarte
- [4] MP<sup>2</sup>I-Schnittstelle
- [5] Anschluss für DC 24V-  
Spannungsversorgung
- [6] LEDs Statusanzeige PROFIBUS-  
DP-Slave
- [7] PROFIBUS-DP-Schnittstelle

**Frontansicht  
CPU 21xCAN**



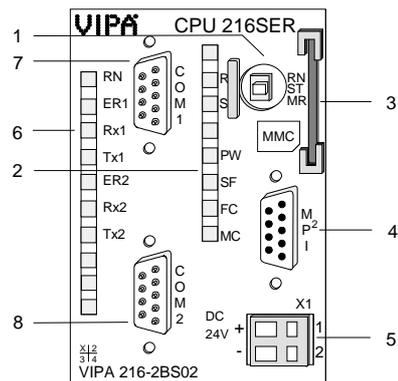
- [1] Betriebsarten-Schalter  
RUN/STOP/RESET
- [2] LEDs Statusanzeige CPU
- [3] Steckplatz MMC-Speicherkarte
- [4] MP<sup>2</sup>I-Schnittstelle
- [5] Anschluss für DC 24V-  
Spannungsversorgung
- [6] LEDs Statusanzeige CAN-  
Master
- [7] CAN-Schnittstelle

**Frontansicht  
CPU 21xSER-1**



- [1] Betriebsarten-Schalter  
RUN/STOP/RESET
- [2] LEDs Statusanzeige CPU
- [3] Steckplatz MMC-Speicherkarte
- [4] MP<sup>2</sup>I-Schnittstelle
- [5] Anschluss für DC 24V-  
Spannungsversorgung
- [6] LEDs Statusanzeige  
Kommunikation
- [7] RS232-Schnittstelle  
(nur 21x-2BS12)  
RS485-Schnittstelle  
(nur 21x-2BS32)

**Frontansicht  
CPU 21xSER-2**



- [1] Betriebsarten-Schalter  
RUN/STOP/RESET
- [2] LEDs Statusanzeige CPU
- [3] Steckplatz MMC-Speicherkarte
- [4] MP<sup>2</sup>I-Schnittstelle
- [5] Anschluss für DC 24V-  
Spannungsversorgung
- [6] LEDs Statusanzeige  
Kommunikation
- [7] RS232-Schnittstelle 1
- [8] RS232-Schnittstelle 2

## Komponenten

### CPU 21x

Die hier für die CPU 21x beschriebenen Komponenten sind auch Bestandteil aller in diesem Handbuch vorgestellten CPUs.

### LEDs

Die CPUs 21x besitzen verschiedene LEDs, die der Busdiagnose und der Programm-Statusanzeige dienen. Die Verwendung und die jeweiligen Farben dieser Diagnose-LEDs finden Sie in der nachfolgenden Tabelle.

Diese LEDs sind Bestandteil jeder in diesem Handbuch vorgestellten CPU.

Bezeichnung	Farbe	Bedeutung
PW	Grün	Signalisiert die eingeschaltete CPU.
R	Grün	CPU befindet sich in RUN-Betriebszustand.
S	Gelb	CPU befindet sich in STOP-Betriebszustand.
SF	Rot	Leuchtet bei System-Fehler (Hardware-Defekt)
FC	Gelb	Leuchtet, sobald Variablen geforced (fixiert) werden.
MC	Gelb	Ein Blinken zeigt Zugriffe auf die MMC an.

### Betriebsarten-Schalter RN/ST/MR

Mit dem Betriebsartenschalter können Sie zwischen den Betriebsarten STOP (ST) und RUN (RN) wählen. Die Betriebsart ANLAUF wird von der CPU automatisch zwischen STOP und RUN ausgeführt.

Mit der Tasterstellung Memory Reset (MR) fordern Sie das Urlöschen an.

### Steckplatz MMC-Speicherkarte

Als externes Speichermedium können Sie hier ein MMC-Speicher-Modul von VIPA einsetzen (Best.-Nr.: VIPA 953-0KX10).

Ein Zugriff auf die MMC erfolgt immer nach Urlöschen.

### Spannungsversorgung

Die CPU besitzt ein eingebautes Netzteil. Der Anschluss erfolgt über zwei Anschlussklemmen an der Frontseite.

Das Netzteil ist mit DC 24V (20,4 ... 28,8V) zu versorgen. Über die Versorgungsspannung werden neben der CPU-Elektronik auch die angeschlossenen Module über den Rückwandbus versorgt.

Die CPU-Elektronik ist nicht galvanisch von der Versorgungsspannung getrennt. Das Netzteil ist gegen Verpolung und Überstrom geschützt.



#### Hinweis!

Bitte achten Sie auf richtige Polarität bei der Spannungsversorgung.

**Batteriepufferung für Uhr und RAM**

Jede CPU 21x 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.



**Achtung!**

Der Akku muss in Ordnung sein, damit die CPU in Betrieb gehen kann. Bei einem Fehler des eingebauten Akkus geht die CPU in STOP. In diesem Fall sollte die CPU überprüft werden. Setzen Sie sich hierzu mit der VIPA in Verbindung!

**MP<sup>2</sup>I-Schnittstelle**

Die MPI-Schnittstelle dient zur Datenübertragung zwischen CPUs und PCs. In einer Buskommunikation können Sie Programme und Daten zwischen den CPUs transferieren, die über MPI verbunden sind.

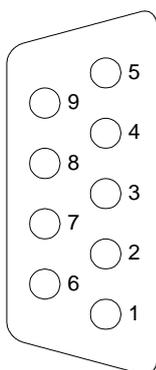
Zur seriellen Übertragung von Ihrem PC aus ist ein MPI-Umsetzer erforderlich. Sie können aber auch von VIPA das "Green Cable" (Best.-Nr. VIPA 950-0KB00) beziehen.

Hiermit können Sie nur bei Systemkomponenten von VIPA als Punkt-zu-Punkt-Verbindung seriell über die MPI-Schnittstelle Ihre Daten übertragen.

Bitte beachten Sie die "Hinweise zum Einsatz der MPI-Schnittstelle" in Teil "Einsatz CPU 21x".

Die MP<sup>2</sup>I-Buchse hat folgende Pinbelegung:

*9polige Buchse*



Pin	Belegung
1	reserviert (darf nicht belegt sein)
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.

**CPU 21x-2BT10**

Zusätzlich zu den unter CPU 21x aufgeführten Komponenten besitzt die CPU 21x-2BT10 auf der linken Seite weitere LEDs und einen Ethernet-Anschluss, die hier näher beschrieben werden.

**LEDs**

Die LEDs befinden sich auf der Front der linken Gehäusehälfte und zeigen die Kommunikation des CPs an. Die Verwendung und die jeweiligen Farben dieser LEDs finden Sie in der nachfolgenden Tabelle.

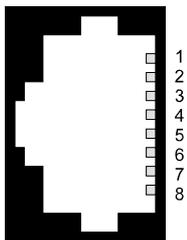
Bezeichnung	Farbe	Bedeutung
RN	Grün	CP-RUN An: CP-Projekt ist geladen Aus: CP ist urgelöscht (kein Projekt)
ST	Gelb	CP-STOP An: CP ist urgelöscht Aus: CP-Projekt ist geladen
IF	Rot	An: Internem CP-Fehler
L/A	Grün	Link/Activity: An: Physikalisch mit Ethernet verbunden Aus: Keine physikalische Ethernet-Verbindung Blinken: Ethernet-Aktivität
S	Grün	Übertragungsrate: An: 100MBit Aus: 10MBit

**Ethernet-Anschluss**

Über die RJ45-Buchse können Sie den CP an Twisted-Pair-Ethernet anbinden. Die Buchse hat folgende Belegung:

*8-polige RJ45-Buchse:*

Pin	Belegung
1	Transmit +
2	Transmit -
3	Receive +
4	-
5	-
6	Receive -
7	-
8	-



**Hinweis!**

Näheres hierzu finden Sie im Teil "Einsatz CPU 21x-2BT10 unter TCP/IP".

**CPU 21x-2BT02**

Zusätzlich zu den unter CPU 21x aufgeführten Komponenten besitzt die CPU 21x-2BT02 auf der linken Seite weitere LEDs und einen Ethernet-Anschluss, die hier näher beschrieben werden.

**LEDs**

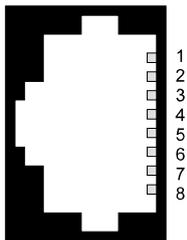
Die LEDs befinden sich auf der Front der linken Gehäusehälfte und zeigen Kommunikation an. Die Verwendung und die jeweiligen Farben dieser LEDs finden Sie in der nachfolgenden Tabelle.

Bezeichnung	Farbe	Bedeutung
PW	Grün	Signalisiert die eingeschaltete CPU
TxD	Grün	Daten senden (transmit data)
RxD	Grün	Daten empfangen (receive data)

**Ethernet-Anschluss**

Über die RJ45-Buchse haben Sie einen Twisted-Pair-Anschluss an Ethernet. Die Buchse hat folgende Belegung:

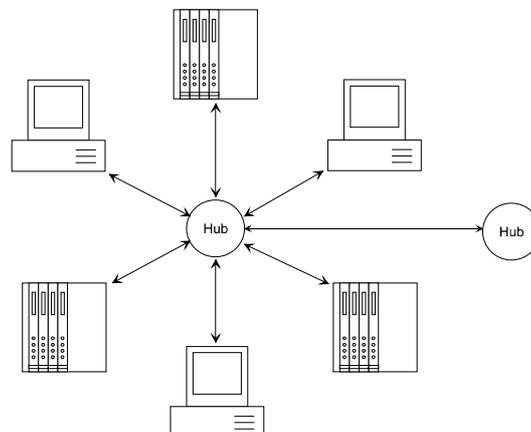
8-polige RJ45-Buchse:



Pin	Belegung
1	Transmit +
2	Transmit -
3	Receive +
4	-
5	-
6	Receive -
7	-
8	-

**Sterntopologie**

Sie können ein Twisted-Pair-Netzwerk nur sternförmig aufbauen. Für die Sterntopologie ist ein Sternkoppler erforderlich:

**Hinweis!**

Näheres hierzu finden Sie im Teil "Einsatz CPU 21x-2BT02 unter H1/TCP/IP".



**CPU 21xDPM**

Zusätzlich zu den unter CPU 21x aufgeführten Komponenten besitzt die CPU 21xDPM auf der linken Seite 4 weitere LEDs und einen PROFIBUS-Anschluss, die hier näher beschrieben werden.

**LEDs**

Die LEDs befinden sich auf der Front der linken Gehäusehälfte und dienen der Diagnose. Die Verwendung und die jeweiligen Farben dieser LEDs finden Sie in der nachfolgenden Tabelle.

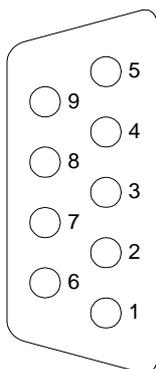
Bezeichnung	Farbe	Bedeutung
RN	Grün	DP-Master-RUN An: Master befindet sich im RUN. Die Slaves werden automatisch angesprochen und die Ausgänge sind 0 ("clear"-Zustand). An mit DE: Master befindet sich im "operate"-Zustand. Er tauscht Daten mit den Slaves aus.
IF	Rot	Initialisierungsfehler An: fehlerhafte PROFIBUS-Parametrierung.
DE	Gelb	DE (Data exchange) An: Zeigt Kommunikation über PROFIBUS
ER	Rot	Fehler An: Slave ist ausgefallen

**PROFIBUS-Anschluss**

Die CPU 21xDPM wird über eine 9-polige Buchse in das PROFIBUS-System eingebunden.

Die Anschlussbelegung dieser Schnittstelle zeigt folgende Abbildung:

*9-polige PROFIBUS-SubD-Buchse:*



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



**Hinweis!**

Näheres zum Thema PROFIBUS-DP-Master finden Sie im Teil "Einsatz CPU 21xDPM".

**CPU 21xDP**

Zusätzlich zu den unter CPU 21x aufgeführten Komponenten besitzt die CPU 21xDP auf der linken Seite 3 weitere LEDs und einen PROFIBUS-Anschluss, die hier näher beschrieben werden.

**LEDs**

Die LEDs befinden sich auf der Front der linken Gehäusehälfte und dienen der Diagnose. Die Verwendung und die jeweiligen Farben dieser LEDs finden Sie in der nachfolgenden Tabelle.

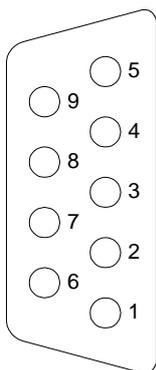
Bezeichnung	Farbe	Bedeutung
ER	Rot	Fehler An: Fehler im PROFIBUS-Teil bzw. CPU ist gestoppt Blinken (2Hz): Initialisierungsfehler Blinken (10Hz): Versorgungsspannung < DC18V Blinken <i>abwechselnd</i> mit RD: Projektierungsfehler (Konfiguration des Masters fehlerhaft) Blinken <i>gleichzeitig</i> mit RD: Parametrierungsfehler
RD	Grün	Ready An: Datenübertragung über Rückwandbus. Blinken: Positiver Selbsttest (READY) und erfolgreiche Initialisierung
DE	Grün	DE (Data exchange) An: Zeigt Kommunikation über PROFIBUS

**PROFIBUS-Anschluss**

Die CPU 21xDP wird über eine 9-polige Buchse in das PROFIBUS-System eingebunden.

Die Anschlussbelegung dieser Schnittstelle zeigt folgende Abbildung:

*9-polige PROFIBUS-SubD-Buchse:*



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

**Hinweis!**

Näheres zum Thema PROFIBUS finden Sie im Teil "Einsatz CPU 21xDP".

**CPU 21xCAN**

Zusätzlich zu den unter CPU 21x aufgeführten Komponenten besitzt die CPU 21xCAN auf der linken Seite 4 weitere LEDs und einen CAN-Anschluss, die hier näher beschrieben werden.

**LEDs**

Die LEDs befinden sich auf der Front der linken Gehäusehälfte und dienen der Diagnose. Die Verwendung und die jeweiligen Farben dieser LEDs finden Sie in der nachfolgenden Tabelle.

Bez.	Farbe	Bedeutung
RN	Grün	CAN-Master-RUN An: CAN-Master befindet sich im RUN Aus: CAN-Master befindet sich im STOP
ER	Rot	Fehler (Error) An: Während der Initialisierung und bei Slave-Ausfall Aus: Alle Slaves befinden sich im Zustand "operational"
BA	Gelb	BA (Bus aktiv) An: Kommunikation über CAN-Bus bzw. Zustand "operational" Blinken (1Hz): Zustand "pre-operational".
IF	Rot	Initialisierung An: Initialisierungsfehler bei fehlerhafter Parametrierung. Aus: Initialisierung ist OK.



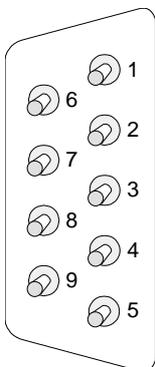
**Hinweis!**

Blinken alle LEDs mit 1Hz, erwartet der CAN-Master gültige Parameter von der CPU. Bekommt der CAN-Master keine Parameter von der CPU, gehen nach 5sec alle LEDs aus.

**CAN-Anschluss**

Die CPU 21xCAN wird über einen 9-poligen Stecker in das CAN-System eingebunden. Der Stecker hat folgende Pinbelegung:

*9-polige CAN-SubD-Stecker:*



Pin	Belegung
1	reserviert
2	CAN low
3	CAN Ground
4	reserviert
5	Schirm
6	Masse 24V
7	CAN high
8	reserviert
9	n.c.



**Hinweis!**

Näheres zum Thema CAN-Master finden Sie im Teil "Einsatz CPU 21xCAN".

**CPU 21xSER-1**

Zusätzlich zu den unter CPU 21x aufgeführten Komponenten besitzt die CPU 21x mit der Best.-Nr. 21x-2BS12 eine RS232-Schnittstelle und die CPU 21x mit der Best.-Nr. 21x-2BS32 eine RS485-Schnittstelle.

**LEDs**

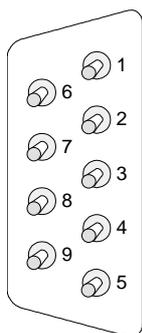
Die LEDs befinden sich auf der Front der linken Gehäusehälfte und dienen der Diagnose. Die Verwendung und die jeweiligen Farben dieser LEDs finden Sie in der nachfolgenden Tabelle.

Bezeichnung	Farbe	Bedeutung
Rx	Grün	Schnittstelle Daten empfangen (receive data)
Tx	Grün	Schnittstelle Daten senden (transmit data)

**RS232-Schnittstelle**

Mit dem 9poligen Stecker können Sie eine serielle Punkt-zu-Punkt-Verbindung über RS232 herstellen. Der Stecker hat folgende Belegung

*9poliger Stecker (CPU 21x-2BS12)*

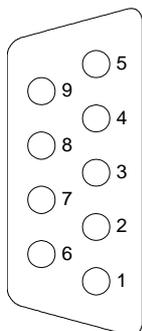


Pin	Belegung
1	CD-
2	RxD
3	TxD
4	DTR-
5	GND
6	DSR-
7	RTS-
8	CTS-
9	RI-

**RS485-Schnittstelle**

Mit der 9poligen Buchse können Sie eine serielle Busverbindung über RS485 herstellen. Die Buchse hat folgende Belegung

*9poliger Buchse (CPU 21x-2BS32)*



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

**CPU 21xSER-2**

Zusätzlich zu den unter CPU 21x aufgeführten Komponenten besitzt die CPU 21xSER-2 weitere LEDs und zwei serielle RS232-Schnittstellen, die hier näher beschrieben werden.

**LEDs**

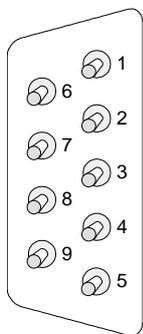
Die LEDs befinden sich auf der Front der linken Gehäusehälfte und dienen der Diagnose. Die Verwendung und die jeweiligen Farben dieser LEDs finden Sie in der nachfolgenden Tabelle.

Bezeichnung	Farbe	Bedeutung
RN	Grün	Kommunikations-Prozessor läuft
ER1	Rot	Fehler Schnittstelle 1
Rx1	Grün	Schnittstelle 1 Daten empfangen (receive data)
Tx1	Grün	Schnittstelle 1 Daten senden (transmit data)
ER2	Rot	Fehler Schnittstelle 2
Rx2	Grün	Schnittstelle 2 Daten empfangen (receive data)
Tx2	Grün	Schnittstelle 2 Daten senden (transmit data)

**RS232-Schnittstelle COM1, COM2**

Die CPU 21xSER-2 besitzt einen Kommunikationsprozessor mit 2 RS232-Schnittstellen. Die Schnittstellen haben folgende Pinbelegung:

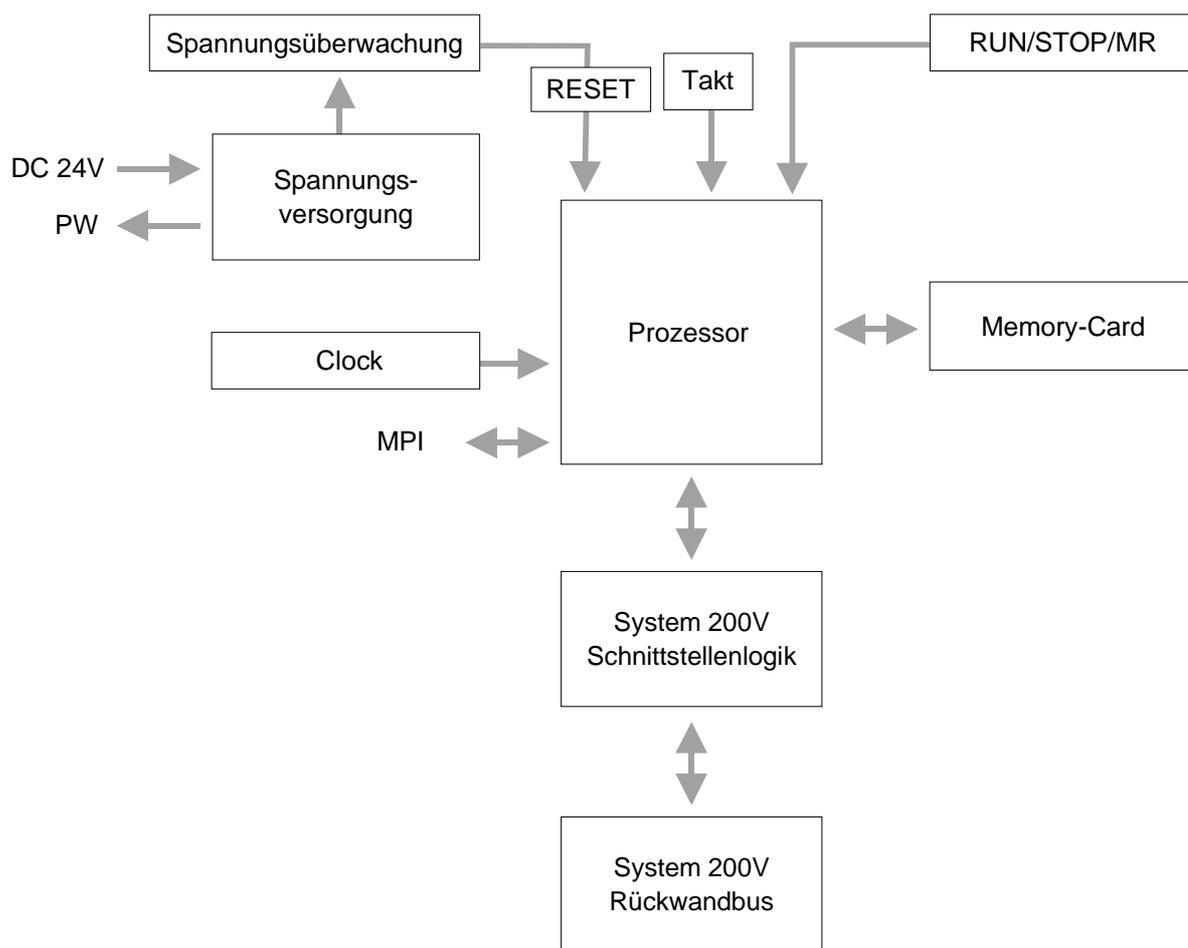
*9poliger Stecker*



Pin	Belegung
1	CD-
2	RxD
3	TxD
4	DTR-
5	GND
6	DSR-
7	RTS-
8	CTS-
9	RI-

### Blockschaltbild

Das nachfolgende Blockschaltbild zeigt den prinzipiellen Hardwareaufbau der CPU 21x:



## Technische Daten

### CPU 21x

#### Allgemein

Elektrische Daten	VIPA 214-1BC02, VIPA 214-1BA02 ... VIPA 216-1BA02
Spannungsversorgung	DC 24V (20,4 ... 28,8V)
Stromaufnahme	max. 1,5A
Verlustleistung	max. 3,5W
Statusanzeigen (LEDs)	über LEDs auf der Frontseite
Anschlüsse / Schnittstellen	MP <sup>2</sup> I-Schnittstelle zur Datenübertragung
Uhr / Pufferung Uhr und RAM	ja / Lithium-Akku, 30 Tage Pufferung
Ausgangsstrom zum Rückwandbus	max. 3A
Merker	8192 Bit (M0.0...M1023.7)
Zeiten	256 (T0...T255)
Zähler	256 (Z0...Z255)
Bausteinanzahl FB	1024 (FB0...FB1023)
FC	1024 (FC0...FC1023)
DB	2047 (DB1...DB2047)
Gesamtadressraum Ein- / Ausgänge	1024 Byte / 1024 Byte davon je 128 Byte Prozessabbild
PA Eingänge	1024Bit (E0.0...E127.7)
PA Ausgänge	1024Bit (A0.0...A127.7)
Kombination mit Peripheriemodulen	
max. Modulanzahl	32
max. digital E/A	32
max. analoge E/A	16
Adressierbare Ein-, Ausgänge	1024 (digital), 128/128 (analog)
Maße und Gewicht	
Abmessungen (BxHxT) in mm	25,4x76x80
Gewicht	80g

#### Modulspezifisch

	CPU 214C	CPU 214	CPU 215	CPU 216
Arbeitsspeicher	32kByte	48kByte	96kByte	128kByte
Ladespeicher	40kByte	80kByte	144kByte	192kByte
Bearbeitungszeit Bit-Operationen	0,18µs	0,18µs	0,18µs	0,18µs
Bearbeitungszeit Wort-Operationen	0,78µs	0,78µs	0,78µs	0,78µs
Best.-Nr.:	214-1BC02	214-1BA02	215-1BA02	216-1BA02

**CPU 21x-2BT10**

Elektrische Daten	VIPA 214-2BT10 ... VIPA 216-2BT10
Spannungsversorgung	DC 24V (20,4 ... 28,8)
Stromaufnahme	max. 1,5A
Verlustleistung	max. 6W
Potenzialtrennung	≥ 500V AC zum Ethernet
Statusanzeige (LEDs)	wie CPU 21x zusätzlich mit LEDs für den Ethernet-Teil
Anschlüsse/Schnittstellen	wie CPU 21x zusätzlich RJ45-Buchse (Ethernet)
Ethernet Schnittstelle	CP 243
Ankopplung	RJ45
Netzwerk Topologie	Sterntopologie
Medium	Twisted Pair
Übertragungsrate	10/100Mbit
Gesamtlänge	max. 100m pro Segment
PG/OP-Kanäle	8 (32 ab CP-Firmware-Version 1.7.4)
Projektierbare Verbindungen	16
Maße und Gewicht	
Abmessungen (BxHxT) in mm	50,8x76x80
Gewicht	150g

**CPU 21x-2BT02**

Elektrische Daten	VIPA 214-2BT02 ... VIPA 216-2BT02
Spannungsversorgung	DC 24V (20,4 ... 28,8)
Stromaufnahme	max. 1,5A
Verlustleistung	max. 6W
Potenzialtrennung	≥ 500V AC zum Ethernet
Statusanzeige (LEDs)	wie CPU 21x zusätzlich mit LEDs für den Ethernet-Teil
Anschlüsse/Schnittstellen	wie CPU 21x zusätzlich RJ45-Buchse (Ethernet)
Ethernet Schnittstelle	CP 243
Ankopplung	RJ45
Netzwerk Topologie	Sterntopologie
Medium	Twisted Pair
Übertragungsrate	10Mbit
Gesamtlänge	max. 100m pro Segment
PG/OP-Kanäle	-
Projektierbare Verbindungen	32
Maße und Gewicht	
Abmessungen (BxHxT) in mm	50,8x76x80
Gewicht	150g

**CPU 21xDPM**

Elektrische Daten	VIPA 214-2BM02 ... VIPA 216-2BM02
Spannungsversorgung	DC 24V (20,4 ... 28,8)
Stromaufnahme	max. 1,5A
Verlustleistung	max. 5W
Potenzialtrennung	≥ 500V AC zum Feldbus
Statusanzeige (LEDs)	wie CPU 21x zusätzlich mit LEDs für den PROFIBUS-Teil
Anschlüsse/Schnittstellen	wie CPU 21x zusätzlich 9polige SubD-Buchse (PROFIBUS)
PROFIBUS Schnittstelle	
Ankopplung	9polige SubD-Buchse
Netzwerk Topologie	Linearer Bus, aktiver Busabschluss an beiden Enden
Medium	Abgeschirmtes verdrilltes Twisted Pair-Kabel, Schirmung darf, abhängig von Umgebungsbedingungen, entfallen.
Übertragungsrate	9,6kbit/s bis 12Mbit/s
Gesamtlänge	ohne Repeater 100m (12Mbit/s), mit Repeater bis 1000m
max. Teilnehmeranzahl	32 Stationen (o. Repeater), auf 126 erweiterbar (m. Repeater).
Kombination mit Peripheriemodulen	
max. Anzahl Slaves	125
max. Anzahl Eingangs-Bytes	1024
max. Anzahl Ausgangs-Bytes	1024
Maße und Gewicht	
Abmessungen (BxHxT) in mm	50,8x76x80
Gewicht	150g

**CPU 21xDP**

Elektrische Daten	VIPA 214-2BP02 ... VIPA 216-2BP02
Spannungsversorgung	DC 24V (20,4 ... 28,8)
Stromaufnahme	max. 1,5A
Verlustleistung	max. 5W
Potenzialtrennung	≥ 500V AC zum Feldbus
Statusanzeige (LEDs)	wie CPU 21x zusätzlich mit LEDs für den PROFIBUS-Teil
Anschlüsse/Schnittstellen	wie CPU 21x zusätzlich 9polig SubD-Buchse (PROFIBUS)
PROFIBUS Schnittstelle	
Ankopplung	9polige SubD-Buchse
Netzwerk Topologie	Linearer Bus, aktiver Busabschluss an beiden Enden
Medium	Abgeschirmtes verdrilltes Twisted Pair-Kabel, Schirmung darf, abhängig von Umgebungsbedingungen, entfallen.
Übertragungsrate	9,6kbit/s bis 12Mbit/s
Gesamtlänge	ohne Repeater 100m (12Mbit/s), mit Repeater bis 1000m
max. Teilnehmeranzahl	32 Stationen (o. Repeater). auf 126 erweiterbar (m. Repeater).
Maße und Gewicht	
Abmessungen (BxHxT) in mm	50,8x76x80
Gewicht	150g

**CPU 21xCAN**

Elektrische Daten	VIPA 214-2CM02 ... VIPA 216-2CM02
Spannungsversorgung	DC 24V (20,4 ... 28,8)
Stromaufnahme	max. 1,5A
Verlustleistung	max. 5W
Potenzialtrennung	≥ 500V AC zum Feldbus
Statusanzeige (LEDs)	wie CPU 21x zusätzlich mit LEDs für den CAN-Teil
Anschlüsse/Schnittstellen	wie CPU 21x zusätzlich 9polig SubD-Buchse (CAN)
CAN Schnittstelle	
Ankopplung	9polige SubD-Buchse
Netzwerk Topologie	Linearer Bus, aktiver Busabschluss an beiden Enden
Medium	Abgeschirmtes verdrilltes Twisted Pair-Kabel, Schirmung darf, abhängig von Umgebungsbedingungen, entfallen.
Übertragungsrate	10kbit/s bis 1Mbit/s
Gesamtlänge	ohne Repeater 1000m bei 50kbit/s
max. Teilnehmeranzahl	126 Stationen
Kombination mit Peripheriemodulen	
max. Anzahl Slaves	125
max. Anzahl TxPDOs	40
max. Anzahl RxPDOs	40
max. Anzahl Eingangs-Byte	384
max. Anzahl Ausgangs-Byte	384
Maße und Gewicht	
Abmessungen (BxHxT) in mm	50,8x76x80
Gewicht	150g

**CPU 21xSER-1**

Modulspezifische Daten	VIPA 214-2BS12...216-2BS12	VIPA 214-2BS32...216-2BS32
Spannungsversorgung	DC 24V (20,4 ... 28,8)	
Stromaufnahme	max. 1,5A	
Verlustleistung	max. 5W	
Potenzialtrennung	≥ 500V AC zur seriellen Schnittstelle	
Statusanzeige (LEDs)	wie CPU 21x	
RS232-Schnittstelle integriert	ja	-
RS485-Schnittstelle integriert	-	ja
Maße und Gewicht		
Abmessungen (BxHxT) in mm	50,8x76x80	
Gewicht	150g	

**CPU 21xSER-2**

Elektrische Daten	VIPA 214-2BS02 ... VIPA 216-2BS02
Spannungsversorgung	DC 24V (20,4 ... 28,8)
Stromaufnahme	max. 1,5A
Verlustleistung	max. 5W
Potenzialtrennung	-
Statusanzeige (LEDs)	wie CPU 21x zusätzlich mit LEDs für Serielle Kommunikation
Anschlüsse/Schnittstellen	wie CPU 21x zusätzlich 2x 9polige SubD Stecker (Serielle Kommunikation)
Schnittstellen	
COM 1, COM 2	2 integrierte RS232-Schnittstellen
Maße und Gewicht	
Abmessungen (BxHxT) in mm	50,8x76x80
Gewicht	150g

## Teil 4 Einsatz CPU 21x

### Überblick

In diesem Kapitel finden Sie alle Informationen, die für den grundsätzlichen Einsatz der CPU 21x mit System 200V Modulen erforderlich sind.

Neben Inbetriebnahme und Anlauf sind hier auch Projektierung, Parametrierung, Betriebsarten und Testfunktionen beschrieben.

Die Angaben gelten auch für den grundsätzlichen Einsatz einer CPU 21x mit integriertem Kommunikations-Teil.

### Inhalt

Thema	Seite
<b>Teil 4 Einsatz CPU 21x</b> .....	<b>4-1</b>
Schnelleinstieg .....	4-2
Hinweise zum Einsatz der MPI-Schnittstelle .....	4-6
Montage .....	4-7
Anlaufverhalten .....	4-8
Adressvergabe .....	4-9
Projektierung .....	4-11
Einstellung der CPU-Parameter .....	4-14
Projekt transferieren .....	4-16
Betriebszustände .....	4-20
Urlöschen .....	4-21
Firmwareupdate .....	4-23
VIPA-spezifische Diagnose-Einträge .....	4-26
Mit Testfunktionen Variablen steuern und beobachten .....	4-28



### Hinweis!

Diese Angaben gelten für alle in diesem Handbuch vorgestellten CPUs, da die Rückwandbus-Kommunikation zwischen CPU und Peripherie-Modulen immer gleich ist!

## Schnelleinstieg

### Übersicht

Zur Projektierung der CPU 21x und der am VIPA-Bus neben der CPU befindlichen System 200V Module verwenden Sie den Hardware-Konfigurator von Siemens.

Damit die direkt gesteckten Peripheriemodule gezielt angesprochen werden können, müssen ihnen bestimmte Adressen in der CPU zugeordnet werden.

Die Adresszuordnung und die Parametrierung der Module erfolgt im Siemens SIMATIC Manager in Form eines virtuellen PROFIBUS-Systems. Da die PROFIBUS-Schnittstelle auch softwareseitig standardisiert ist, können wir auf diesem Weg gewährleisten, dass über die Einbindung einer GSD-Datei die Funktionalität in Verbindung mit dem Siemens SIMATIC Manager jederzeit gegeben ist.

Ihr Projekt wird über die MPI-Schnittstelle in Ihre CPU übertragen

### Voraussetzungen

Für die Projektierung Ihrer CPU 21x müssen folgende Voraussetzungen erfüllt sein:

- Siemens SIMATIC Manager auf PC bzw. PG installiert
- GSD-Dateien in Hardware-Konfigurator von Siemens eingebunden. Aktuelle GSD-Dateien finden Sie im Service-Bereich unter [www.vipa.de](http://www.vipa.de)
- serielle Verbindung zur CPU (z.B. über das "Green Cable" von VIPA)



### Hinweis!

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

### Kompatibilität zu Siemens SIMATIC Manager über GSD-Datei vipa\_21x.gsd

Die Projektierung einer CPU 21x erfolgt im Siemens SIMATIC Manager in Form eines virtuellen PROFIBUS-Systems auf Basis der CPU 315-2DP.

Aufgrund der standardisierten softwareseitigen PROFIBUS-Schnittstelle können wir durch Einbindung der GSD-Datei vipa\_21x.gsd die Funktionalität der System 200V Familie im Siemens SIMATIC Manager zur Verfügung stellen.

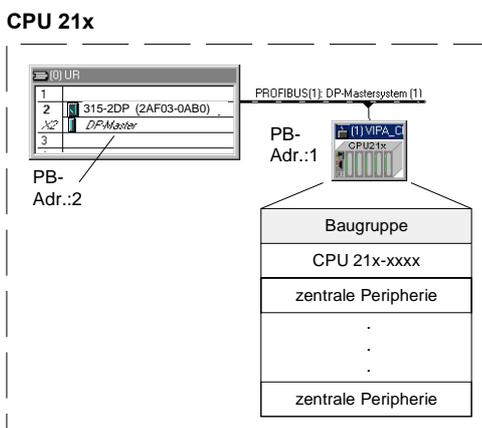
**Um kompatibel mit dem Siemens SIMATIC Manager zu sein, sind folgende Schritte durchzuführen:**

- **PROFIBUS-DP-Master System mit CPU 315-2DP (6ES7 315-2AF03 V1.2) projektieren.**
- **PROFIBUS-Slave "VIPA\_CPU21x" aus HW-Katalog mit Adresse 1 anfügen.**
- **Auf dem 1. Steckplatz des Slave-Systems Ihre CPU 21x einbinden.**

**Projektierung CPU 21x mit zentraler Peripherie**

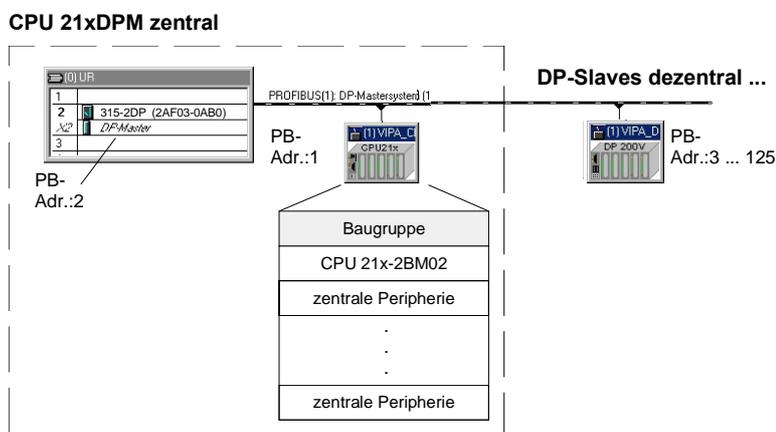
Folgende Schritte sind zur Projektierung einer CPU 21x im Hardware-Konfigurator von Siemens erforderlich:

- Starten Sie den Hardware-Konfigurator von Siemens.
- Installieren Sie die GSD VIPA\_21x.gsd.
- Projektieren Sie ein PROFIBUS-DP-Mastersystem mit der CPU 315-2DP.
- Binden Sie an das Master-System aus dem Hardware-Katalog das Slave-System "VIPA\_CPU21x" an. Sie finden das Slave-System im Hardware-Katalog unter *PROFIBUS-DP > Weitere Feldgeräte > I/O > VIPA\_System\_200V*.
- Geben Sie dem Slave-System die Adresse 1. Hiermit identifiziert die VIPA CPU das System als zentrales Peripherie-System.
- Platzieren Sie in diesem Slave-System in der gesteckten Reihenfolge Ihre Module. Beginnen sie mit der CPU auf dem 1. Steckplatz.
- Binden Sie danach Ihre System 200V Module ein.



**Master-Projektierung der CPU 21xDPM**

Bei der Projektierung einer CPU 21xDPM projektieren Sie die zentral gesteckten Module, wie oben aufgeführt. Slave-Systeme, die an den Master anzukoppeln sind, fügen Sie an das schon bestehende Master-System an:



**Projektierung der CPU 21xDP in einem Master-System**

Bei der Projektierung einer CPU 21xDP projektieren Sie die zentral gesteckten Module, wie auf der vorhergehenden Seite aufgeführt.

*Slave-Parametrierung*

Beim "intelligenten" Slave blendet der PROFIBUS-Teil seine Datenbereiche im Speicherbereich der CPU 21xDP ein. Die Zuordnung der Bereiche führen Sie in den "Eigenschaften" der CPU 21xDP durch. Diese Bereiche sind mit einem entsprechenden SPS-Programm zu versorgen.



**Achtung!**

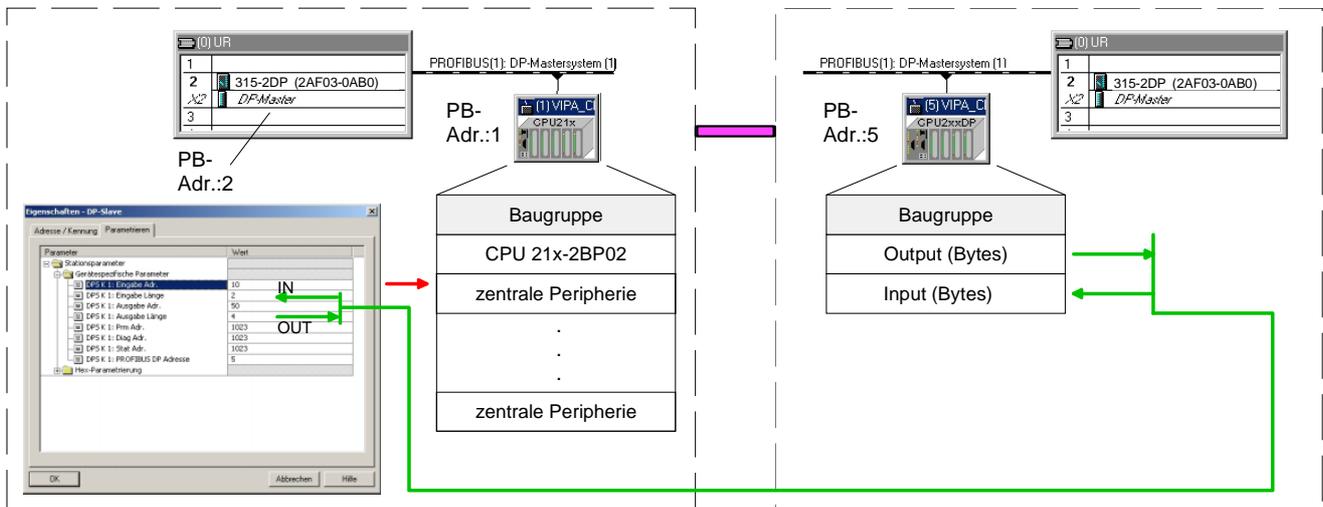
Die Längenangaben für Ein- und Ausgabe-Bereich müssen mit den Byteangaben bei der Projektierung auf Master-Seite übereinstimmen. Ansonsten kann keine PROFIBUS-Kommunikation stattfinden (Slave-Ausfall).

*Schritte der Projektierung im DP-Master*

- CPU mit DP-Master-System projektieren (Adresse 2)
- PROFIBUS-Slave VIPA\_CPU2xxDP aus VIPA04d5.gsd anfügen
- Die PROFIBUS Ein- und Ausgabebereiche ab Steckplatz 0 angeben

**Slave: (VIPA\_CPU21x aus VIPA\_21x.gsd)**

**Master: (VIPA\_CPU21xDP aus VIPA04d5.gsd)**



**Achtung!**

Bei Einsatz eines IM 208DP Master an einer CPU 21x mit Firmwarestand ab V 3.0 ist darauf zu achten, dass der DP-Master ebenfalls einen Firmwarestand ab V 3.0 besitzt; ansonsten ist der Betrieb des IM 208DP nicht möglich.

Die Firmwarestände entnehmen Sie bitte dem Aufkleber, der sich auf der Rückseite der jeweiligen Module befindet.

**Projektierung der CPU 21xNET**

Die Projektierung von Netzwerkverbindungen über Ethernet erfolgt bei der CPU 21x-2BT02 unter WinNCS von VIPA und bei der CPU 21x-2BT10 im Siemens SIMATIC Manager unter NetPro.

**Projektierung CPU 21xSER-1**

Bei der CPU 21x-2BS12 bzw. 21x-2BS32 erfolgt die CP-Kommunikation mittels eines 256Byte großen Sende und Empfangspuffers.

Die Parametrierung zur Laufzeit ist mit dem SFC 216 (SER\_CFG) durchzuführen. Hierbei sind für alle Protokolle mit Ausnahme von ASCII die Parameter in einem DB abzulegen. Das Beschreiben bzw. Lesen des Sende- bzw. Empfangspuffers steuern Sie mit SFC 217 (SER\_SND) bzw. SFC 218 (SER\_RCV).

**Projektierung CPU 21xSER-2**

Der CP der CPU 21x-2BS02 ist über ein Dual-Port-RAM auch "Kachel" genannt, direkt mit dem CPU-Teil verbunden. Diese Kachel steht auf der CPU-Seite als Standard-CP-Interface zur Verfügung. Die Kommunikation über die entsprechenden Protokolle regeln Verbindungsaufträge, die im Anwenderprogramm zu programmieren sind.

Hierbei kommen die VIPA-SFCs 230...238 zum Einsatz.

Die Parameterübertragung an den Kommunikationsprozessor (CP) erfolgt zur Laufzeit mit einem SEND (SFC 230) mit Auftrags-Nr. 201. Hierbei sind die Parameter in einem DB abzulegen, dessen Aufbau sich nach dem gewünschten Protokoll richtet.

Zur Aktivierung der Parameter ist nach dem SEND ein RESET (SFC 234) mit Auftrags-Nr. 0 auszuführen.

**Hinweis!**

Bitte beachten Sie, dass den Aufträgen SEND, RECEIVE, FETCH und RESET immer ein "VKE"=1 voranzustellen ist, da ansonsten diese nicht ausgeführt werden.

**Projektierung der CPU 21xCAN**

Die Projektierung des CANopen-Masters erfolgt unter WinCoCT (**Windows CANopen Configuration Tool**) von VIPA. Aus WinCoCT exportieren Sie Ihr Projekt als wld-Datei, die Sie in den Hardware-Konfigurator von Siemens importieren können.

Legen Sie hierzu ein virtuelles PROFIBUS-System "VIPA\_CPU21x" an und binden Sie auf Steckplatz 0 die CPU21xCAN (VIPA 21x-2CM02) ein.

## Hinweise zum Einsatz der MPI-Schnittstelle

### Was ist MP<sup>2</sup>I

Die MP<sup>2</sup>I-Schnittstelle hat 2 Schnittstellen in einer Schnittstelle vereint:

- MPI-Schnittstelle
- RS232-Schnittstelle

Bitte beachten Sie, dass die MP<sup>2</sup>I-Schnittstelle nur bei Einsatz des Green Cable von VIPA als RS232-Schnittstelle benutzt werden kann.

### Einsatz als MPI-Schnittstelle

Die MPI-Schnittstelle dient zur Datenübertragung zwischen CPUs und PCs. In einer Buskommunikation können Sie Daten zwischen den CPUs transferieren, die über MPI verbunden sind.

Bei Anschluss eines handelsüblichen MPI-Kabels bietet die MPI-Buchse die volle MPI-Funktionalität.



### Wichtige Hinweise zum Einsatz von MPI-Kabeln

Bei Einsatz eines MPI-Kabels an den CPUs von VIPA ist darauf zu achten, dass der Pin 1 nicht verbunden ist. Dies kann zu Transferproblemen führen und ggf. an der CPU einen Defekt herbeiführen!

Insbesondere PROFIBUS-Kabel von Siemens wie beispielsweise das Kabel mit der Best.-Nr. 6XV1 830-1CH30 darf an der MP<sup>2</sup>I-Buchse nicht betrieben werden.

Für Schäden, die aufgrund der Nichtbeachtung dieser Hinweise und bei unsachgemäßem Einsatz entstehen, übernimmt die VIPA keinerlei Haftung!

### Einsatz als RS232-Schnittstelle nur über "Green Cable"



Zur seriellen Übertragung von Ihrem PC aus ist ein MPI-Umsetzer erforderlich. Sie können aber auch das "Green Cable" von VIPA verwenden. Sie erhalten es unter der Best.-Nr. VIPA 950-0KB00.

Hiermit können Sie Ihre Daten, ausschließlich bei VIPA CPUs mit MP<sup>2</sup>I-Buchse, als Punkt-zu-Punkt-Verbindung seriell über die MP<sup>2</sup>I-Buchse übertragen.

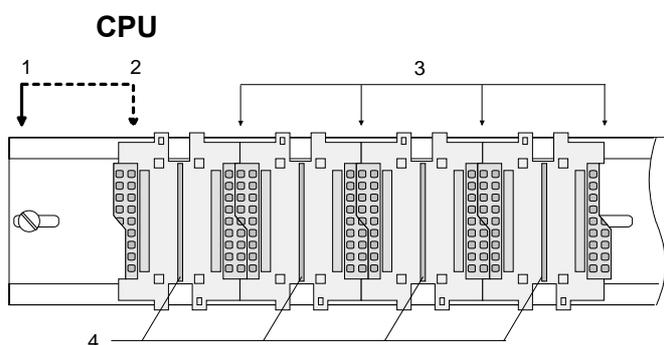
# Montage



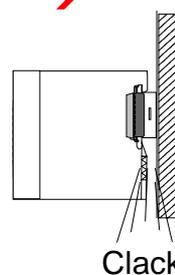
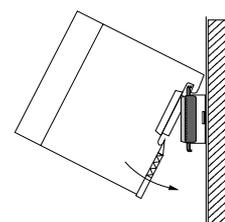
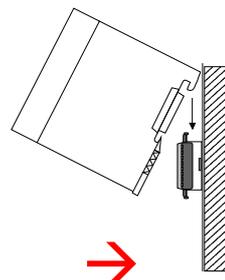
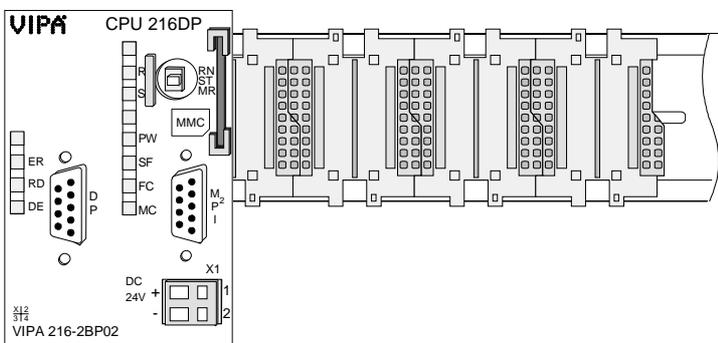
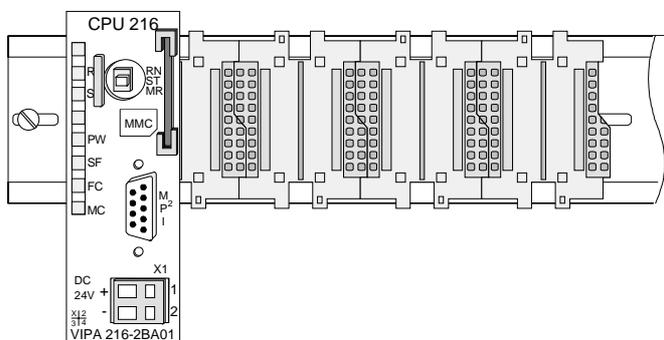
## Achtung!

Schalten Sie die Stromversorgung aus, bevor Sie Module stecken bzw. abziehen!

Bitte beachten Sie, dass Sie die CPU nur auf Steckplatz 1 bzw. 2 stecken dürfen (siehe nachfolgende Abbildung).



- [1] CPU, wenn doppelt breit
- [2] CPU einfach breit
- [3] Peripheriemodule
- [4] Führungsleisten



Näheres zur Montage von System 200V-Modulen finden Sie im Teil "Montage und Aufbaurichtlinien".

## Anlaufverhalten

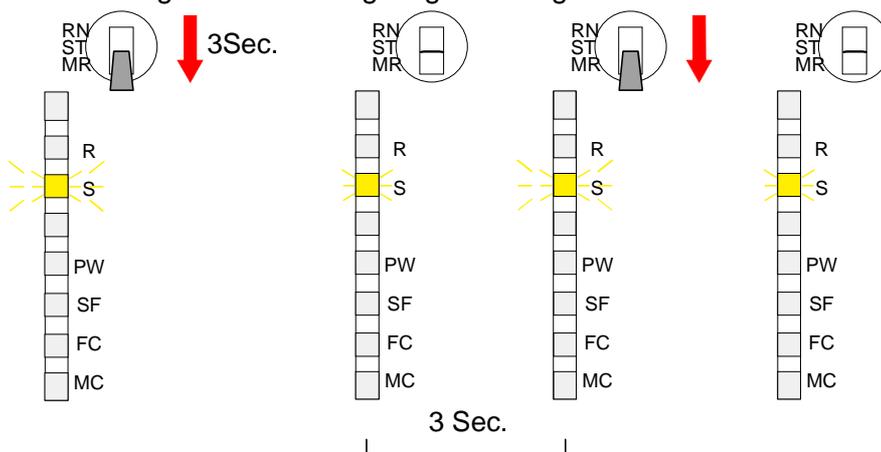
### Stromversorgung einschalten

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

Sie können jetzt aus Ihrem Projektier-Tool heraus über MPI Ihr Projekt in die CPU übertragen bzw. eine MMC mit Ihrem Projekt stecken und Urlöschen ausführen.

### Urlöschen

Die nachfolgende Abbildung zeigt die Vorgehensweise:



### Hinweis!

Das Übertragen des Anwenderprogramms von der MMC in die CPU erfolgt immer nach Urlöschen!

### Anlauf im Auslieferungszustand

Im Auslieferungszustand ist die CPU urgelöscht.

Nach einem STOP→RUN Übergang geht die CPU ohne Programm in RUN.

### Anlauf mit gültigen Daten 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 max. 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, wird das Programm auf der MMC in das RAM übertragen.

Abhängig von der eingestellten Betriebsart geht die CPU in RUN bzw. bleibt im STOP.

Dieser Vorgang wird im Diagnosepuffer unter folgendem Eintrag festgehalten: "Start Urlöschen automatisch (ungepuffert NETZ-EIN)".

## Adressvergabe

### Automatische Adressierung

Damit die gesteckten Peripheriemodule gezielt angesprochen werden können, müssen ihnen bestimmte Adressen in der CPU zugeordnet werden.

Bei der CPU gibt es einen Peripheriebereich (Adresse 0 ... 1023) und ein Prozessabbild der Ein- und Ausgänge (je Adresse 0 ... 127).

Beim Hochlauf der CPU vergibt diese automatisch von 0 an aufsteigend Peripherieadressen für digitale Ein-/Ausgabe-Module.

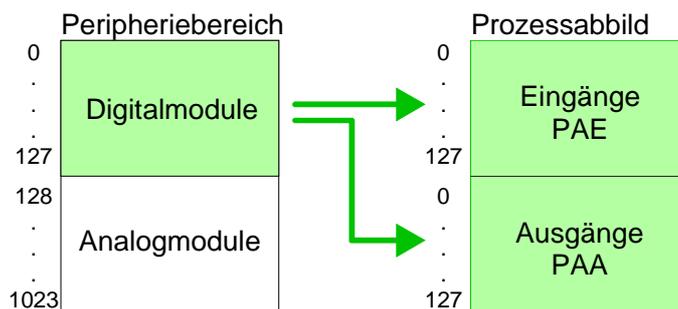
Sofern keine Hardwareprojektierung vorliegt, werden Analog-Module bei der automatischen Adressierung auf gerade Adressen ab Adresse 128 abgelegt.

### Signalzustände in Prozessabbild

Die Signalzustände der unteren Adresse (0 ... 127) werden zusätzlich in einem besonderen Speicherbereich, dem *Prozessabbild* gespeichert.

Das Prozessabbild ist in zwei Teile gegliedert:

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



Nach jedem Zyklusdurchlauf wird das Prozessabbild automatisch aktualisiert.

### Lese- und Schreibzugriffe

Über Lese- bzw. Schreibzugriffe auf die Peripheriebytes oder auf das Prozessabbild können Sie die Module ansprechen.



#### Hinweis!

Bitte beachten Sie, dass durch den lesenden und schreibenden Zugriff auf dieselbe Adresse unterschiedliche Module angesprochen werden können.

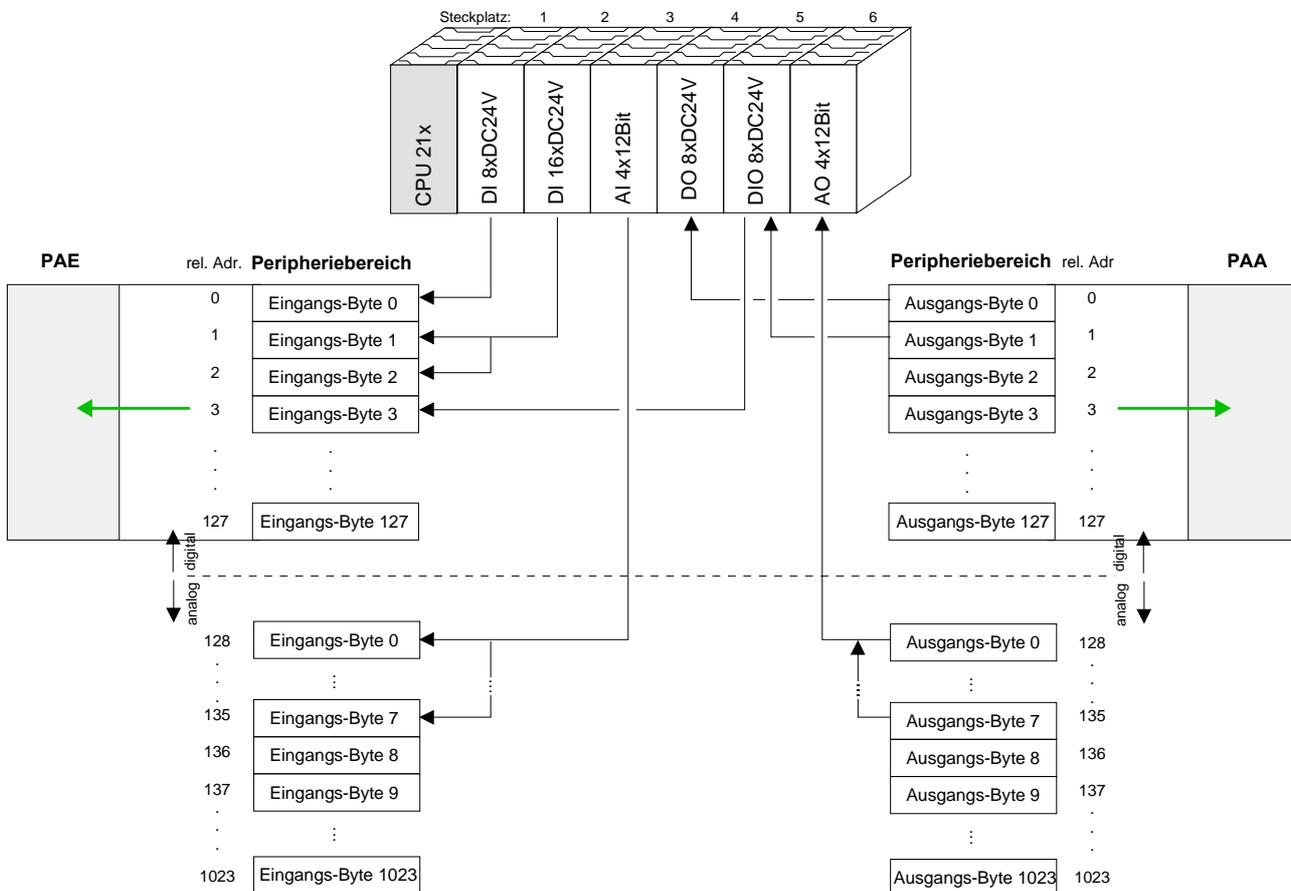
Digitale und analoge Module haben bei der automatischen Adressierung getrennte Adressbereiche.

Digitalmodule: 0 ... 127

Analogmodule: 128 ... 1023

**Beispiel zur automatischen Adresszuordnung**

Die nachfolgende Abbildung soll die automatische Adresszuordnung nochmals verdeutlichen:



**Adresszuordnung durch Projektierung ändern**

Sie können jederzeit durch Einsatz des Siemens SIMATIC Managers die Adresszuordnung ändern. Somit können Sie auch Analogmodule in den Prozessabbildbereich (0 ... 127) legen und Digitalmodule oberhalb von 127. Die Vorbereitung für die Projektierung und die Vorgehensweise bei der Projektierung sind auf den Folgeseiten beschrieben.

## Projektierung

### Allgemein

Die hier gemachten Angaben beziehen sich immer auf Module, die auf dem gleichen Bus neben der CPU gesteckt sind.

Damit die gesteckten Peripheriemodule gezielt angesprochen werden können, müssen ihnen bestimmte Adressen in der CPU zugeordnet werden.

Die Adresszuordnung und die Parametrierung der direkt gesteckten Module erfolgt im Siemens SIMATIC Manager in Form eines virtuellen PROFIBUS-Systems. Da die PROFIBUS-Schnittstelle softwareseitig standardisiert ist, können wir auf diesem Weg gewährleisten, dass über die Einbindung einer GSD-Datei die Funktionalität in Verbindung mit dem Siemens SIMATIC Manager jederzeit gegeben ist

Ihr Projekt übertragen Sie seriell über die MPI-Schnittstelle oder über MMC in Ihre CPU.

---

### Schnelleinstieg

Für den Einsatz der CPU 21x von VIPA in Verbindung mit dem Siemens SIMATIC Manager ist die Einbindung des System 200V über die GSD-Datei von VIPA im Hardware-Katalog erforderlich.

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

- Hardware-Konfigurator von Siemens starten und vipa\_21x.gsd von VIPA einbinden.
- CPU 315-2DP (315-2AF03 0AB00 V1.2) von Siemens projektieren und hierbei ein neues PROFIBUS-Subnetz anlegen.
- An das Subnetz das Slave-System "VIPA\_CPU21x" mit der **PROFIBUS-Adresse 1** anhängen, zu finden im Hardware-Katalog unter *PROFIBUS DP > Weitere Feldgeräte > IO > VIPA\_System\_200V*.
- Platzieren Sie **immer auf dem 1. Steckplatz** die entsprechende CPU 21x, indem Sie diese dem Hardware-Katalog entnehmen.
- Binden Sie danach Ihre System 200V Module in der gesteckten Reihenfolge ein.
- Sichern und übertragen Sie Ihr Projekt.

---

### Voraussetzung

Folgende Voraussetzungen müssen für die Projektierung erfüllt sein

- Siemens SIMATIC Manager auf PC bzw. PG installiert
- GSD-Dateien in Hardware-Konfigurator von Siemens eingebunden
- Serielle Verbindung zur CPU (z.B. "Green Cable" von VIPA)



### Hinweis!

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

**GSD-Datei einbinden**

- Gehen Sie auf [www.vipa.de](http://www.vipa.de) > Service > Download > GSD- und EDS-Files > PROFIBUS und laden Sie die Datei Cx000023\_Vxxx.
- Extrahieren Sie die Datei in Ihr Arbeitsverzeichnis. Die vipa\_21x.gsd (deutsch) bzw. vipa\_21x.gse (englisch) befinden sich im Verzeichnis VIPA\_System\_200V.
- Starten Sie den Hardware-Konfigurator von Siemens und schließen Sie alle Projekte.
- Gehen Sie auf **Extras** > *Neue GSD-Datei installieren*.
- Navigieren Sie in das Verzeichnis *System\_200V* und geben Sie die entsprechende Datei **vipa\_cpu21x.gsd** (deutsch) oder **vipa\_cpu21x.gse** (englisch) an.

Die Module des System 200V von VIPA befinden sich im Hardwarekatalog unter *PROFIBUS-DP \ Weitere Feldgeräte \ I/O \ VIPA\_System\_200V*.

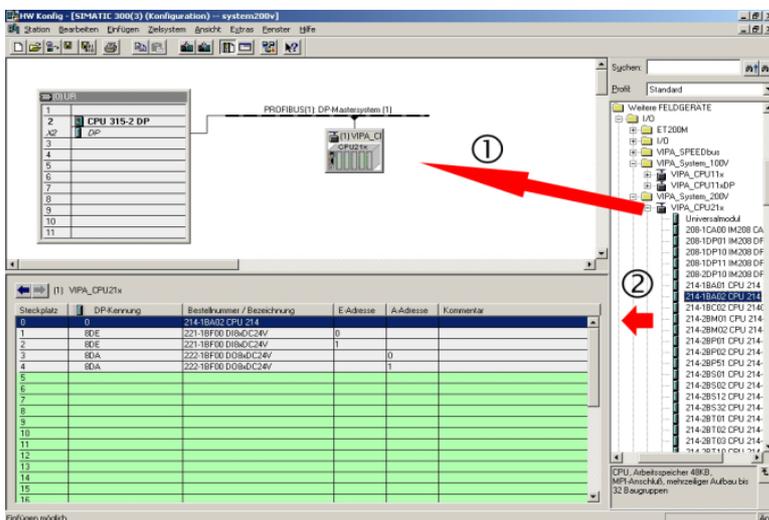
**Projektierung der CPU 21x**

- Starten Sie den Hardware-Konfigurator von Siemens mit einem neuen Projekt und fügen Sie aus dem Hardware-Katalog eine Profilschiene ein.
- Platzieren Sie auf dem ersten möglichen Steckplatz die CPU 315-2DP (6ES7 315-2AF03 V1.2) von Siemens.
- Sofern Ihre CPU 21x einen PROFIBUS-DP-Master integriert hat, können Sie diesen jetzt mit PROFIBUS vernetzen und Ihre DP-Slaves anbinden.

**Projektierung der System 200V Module**

Die System 200V Module sind, nachdem Sie die CPU 21x projiziert haben, nach folgender Vorgehensweise zu projektieren:

- Erzeugen Sie ein PROFIBUS-Subnetz (falls noch nicht vorhanden)
- Hängen Sie an das Subnetz das System "VIPA\_CPU21x". Sie finden es im Hardware-Katalog unter *PROFIBUS DP > Weitere Feldgeräte > IO > VIPA\_System\_200V*. Geben Sie diesem Slave die **PROFIBUS-Adresse 1**.
- Platzieren Sie in Ihrem Konfigurator **immer auf dem 1. Steckplatz** die CPU 21x, die Sie einsetzen, indem Sie diese dem Hardware-Katalog entnehmen.
- Binden Sie danach Ihre System 200V Module in der gesteckten Reihenfolge ein.
- Sichern Sie Ihr Projekt.

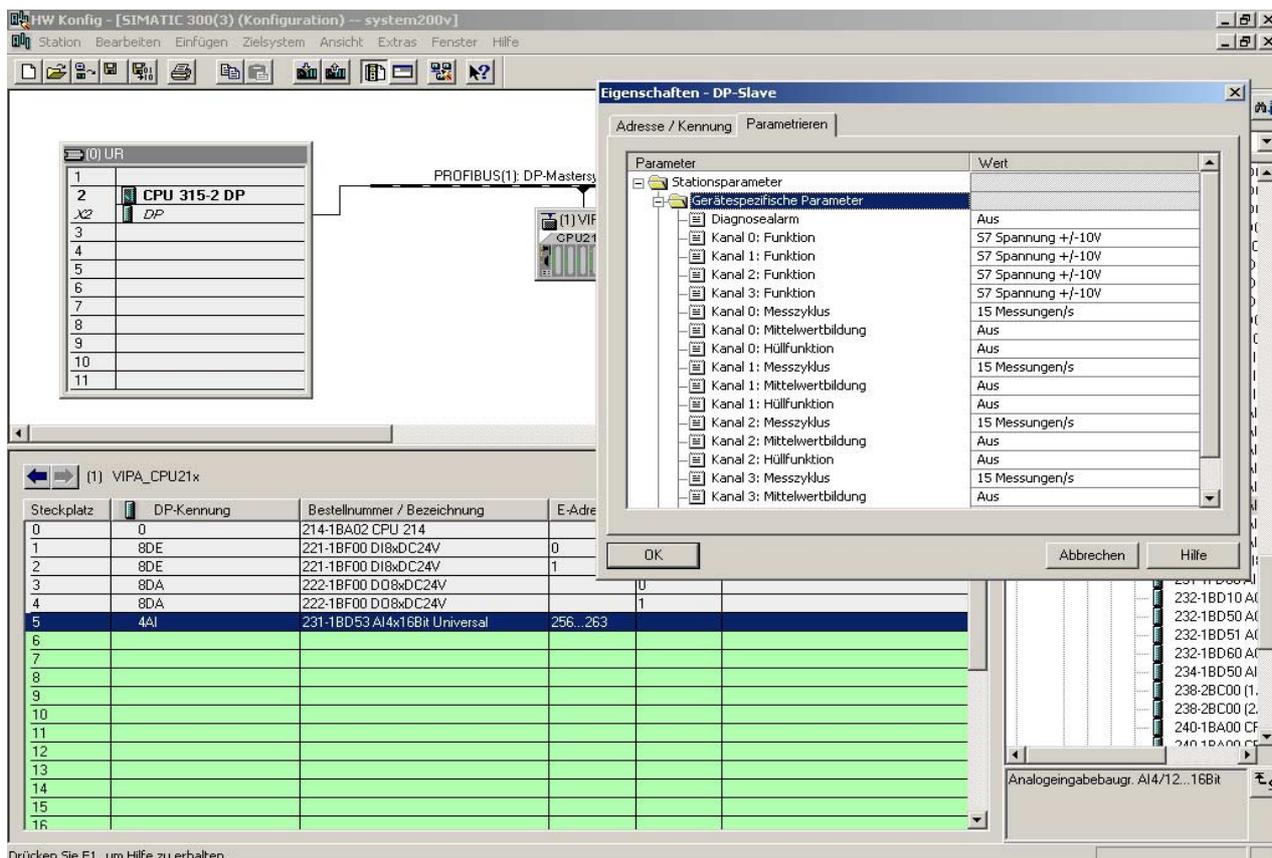


Parametrierung System 200V Module

System 200V Module können bis zu 16Byte Parameterdaten von der CPU erhalten. Durch Einsatz des Siemens SIMATIC Managers können Sie jederzeit für parametrierbare System 200V Module Parameter vorgeben.

Doppelklicken Sie hierzu bei der Projektierung in Ihrer Steckplatzübersicht auf das zu parametrierende Modul.

Die nachfolgende Abbildung zeigt die Parametrierung eines Analogen Eingabe-Moduls:



Projekt übertragen

Die Datenübertragung erfolgt über MPI. Sollte Ihr Programmiergerät keine MPI-Schnittstelle besitzen, können Sie für eine serielle Punkt-zu-Punkt-Übertragung von Ihrem PC an MPI das "Green Cable" von VIPA verwenden.

Das "Green Cable" hat die Best.-Nr. VIPA 950-0KB00 und darf nur bei den VIPA CPUs mit MP<sup>2</sup>I-Schnittstelle eingesetzt werden.

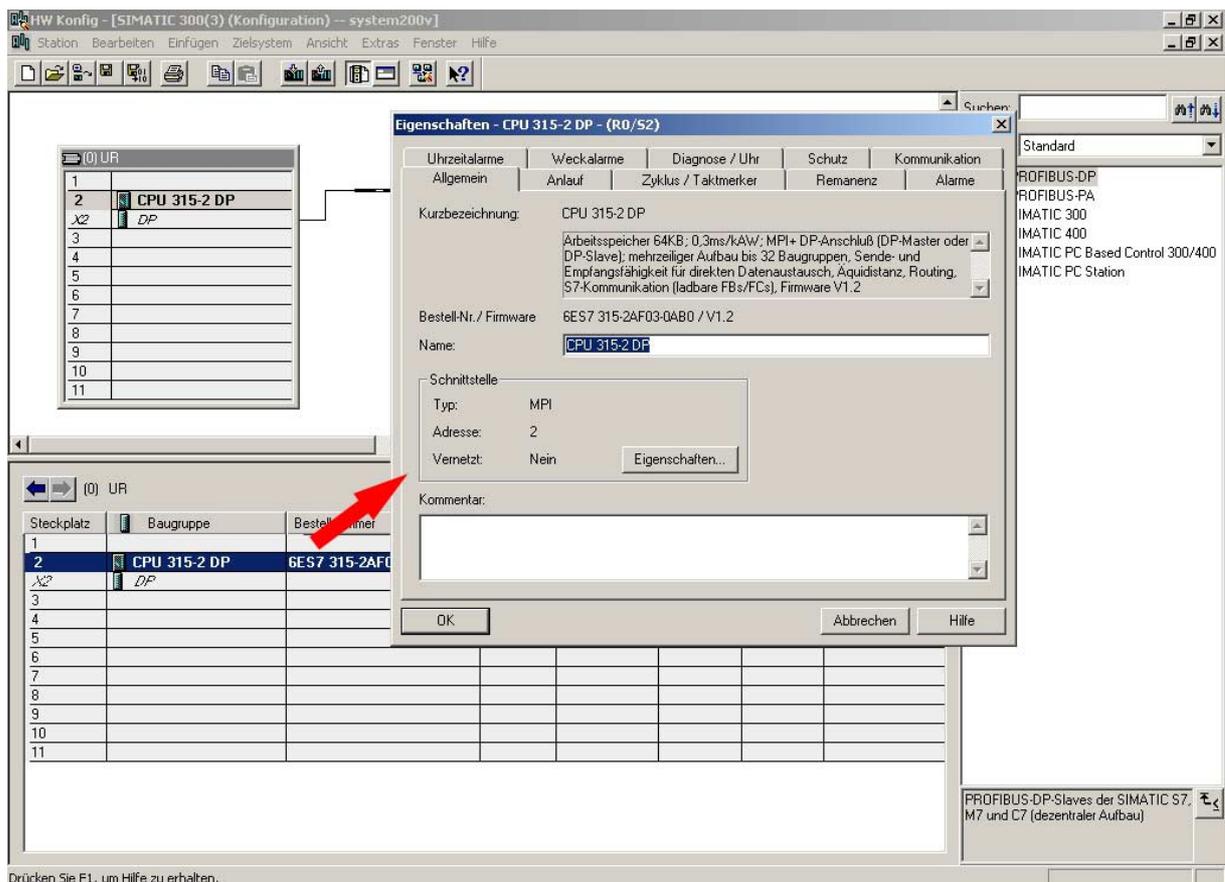
Bitte beachten Sie hierzu die Hinweise zum Green Cable in den Grundlagen!

- Verbinden Sie Ihr PG mit der CPU.
- Mit **Zielsystem** > *Laden in Baugruppe* in Ihrem Projektierool übertragen Sie Ihr Projekt in die CPU.
- Stecken Sie eine MMC und übertragen Sie mit **Zielsystem** > *RAM nach ROM kopieren* Ihr Anwenderprogramm auf die MMC.
- Während des Schreibvorgangs blinkt die "MC"-LED auf der CPU. Systembedingt wird zu früh ein erfolgter Schreibvorgang gemeldet. Der Schreibvorgang ist erst beendet, wenn die LED erlischt.

## Einstellung der CPU-Parameter

### Übersicht

Mit Ausnahme der Parameter für den PROFIBUS-Teil der CPU 21xDP erfolgt die **CPU-Parametrierung im Parameter-Dialog der CPU 315-2DP**. Die Parametrierung des PROFIBUS-Teils der CPU 21xDP erfolgt im Parameter-Dialog der CPU 21xDP.



### Parametrierung CPU 21x unter CPU 315-2DP

Durch Doppelklick auf die CPU 315-2DP gelangen Sie in das Parametrierfenster für Ihre CPU 21x. Über die Register haben Sie Zugriff auf alle Parameter der CPU 315-2DP. Die Parameter entsprechen denen der CPU 21x von VIPA.

Bitte beachten Sie, dass zur Zeit nicht alle Parameter unterstützt werden.

Parameter, die  
unterstützt werden

Die CPU wertet nicht alle Parameter aus, die Sie in Ihrem Projektiertool einstellen können.

Folgende Parameter werden zur Zeit in der CPU ausgewertet:

*Allgemein :*

MPI-Adresse der CPU  
Baudrate (19,2kBaud, 187kBaud)  
maximale MPI-Adresse

*Uhrzeitalarm :*

OB10 : Ausführung  
Aktiv  
Startdatum  
Uhrzeit

*Anlauf :*

Anlauf bei Sollausbau ungleich...  
Fertigmeldung durch Baugruppe  
Übertragung der Parameter an...

*Weckalarm :*

OB35 : Ausführung

*Remanenz :*

Anzahl Merkerbytes ab MB0  
Anzahl S7-Timer ab T0  
Anzahl S7-Zähler ab Z0

*Zyklus / Taktmerker :*

Zyklusüberwachungszeit  
Zyklusbelastung durch  
Kommunikation  
OB85-Aufruf bei Peripherie-  
zugriffsfehler  
Taktmerker mit Merkerbytenummer

*Schutz :*

Schutzstufe durch Passwort ...

**Parametrierung  
PROFIBUS-Teil  
der  
CPU 21xDP**

Sie gelangen in das Parametrierfenster für den PROFIBUS-Teil der CPU 21x, indem Sie unter VIPA\_CPU21x auf die eingefügte System 200V-CPU doppelklicken.

Über die Register haben Sie Zugriff auf alle Parameter des PROFIBUS-Teils Ihrer CPU 21x.

Näheres hierzu finden Sie im Kapitel "Einsatz CPU 21xDP".

## Projekt transferieren

### Übersicht

Es bestehen 2 Möglichkeiten für den Transfer Ihres Projekts in die CPU:

- Transfer über MPI
- Transfer über MMC bei Einsatz eines MMC-Lesers

---

### Transfer über MPI

Der Aufbau eines MPI-Netzes ist prinzipiell gleich dem Aufbau eines 1,5Mbaud PROFIBUS-Netzes. Das heißt, es gelten dieselben Regeln und Sie verwenden für beide Netze die gleichen Komponenten zum Aufbau.

Defaultmäßig wird das MPI-Netz mit 187kbaud betrieben.

Jeder Busteilnehmer identifiziert sich mit einer eindeutigen MPI-Adresse am Bus.

Sie verbinden die einzelnen Teilnehmer über Busanschlussstecker und das PROFIBUS-Buskabel.

### 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, während des Hochlaufs und des Betriebs immer mit Spannung versorgt sind.

### Vorgehensweise

- Verbinden Sie Ihr PG bzw. Ihren PC über MPI mit Ihrer CPU.  
Sollte Ihr Programmiergerät keine MPI-Schnittstelle besitzen, können Sie für eine serielle Punkt-zu-Punkt-Übertragung von Ihrem PC an MPI das "Green Cable" von VIPA verwenden.  
Das "Green Cable" hat die Best.-Nr. VIPA 950-0KB00 und darf nur bei VIPA CPUs mit MP<sup>2</sup>I-Schnittstelle eingesetzt werden.
- Konfigurieren Sie die MPI-Schnittstelle Ihres PC.
- Mit **Zielsystem** > *Laden in Baugruppe* in Ihrem Projektierwerkzeug übertragen Sie Ihr Projekt in die CPU.
- Zur zusätzlichen Sicherung Ihres Projekts auf MMC stecken Sie eine MMC und übertragen Sie mit **Zielsystem** > *RAM nach ROM kopieren* Ihr Anwenderprogramm auf die MMC.  
Während des Schreibvorgangs blinkt die "MC"-LED auf der CPU. Systembedingt wird zu früh ein erfolgreicher Schreibvorgang gemeldet. Der Schreibvorgang ist erst beendet, wenn die LED erlischt.

## MPI konfigurieren

Hinweise zur Konfiguration einer MPI-Schnittstelle finden Sie in der Dokumentation zu Ihrer Programmiersoftware.

An dieser Stelle soll lediglich der Einsatz des "Green Cable" von VIPA in Verbindung mit dem Programmierwerkzeug von Siemens gezeigt werden.

Das "Green Cable" stellt über MPI eine serielle Verbindung zwischen der COM-Schnittstelle des PCs und der MP<sup>2</sup>I-Schnittstelle der CPU her.



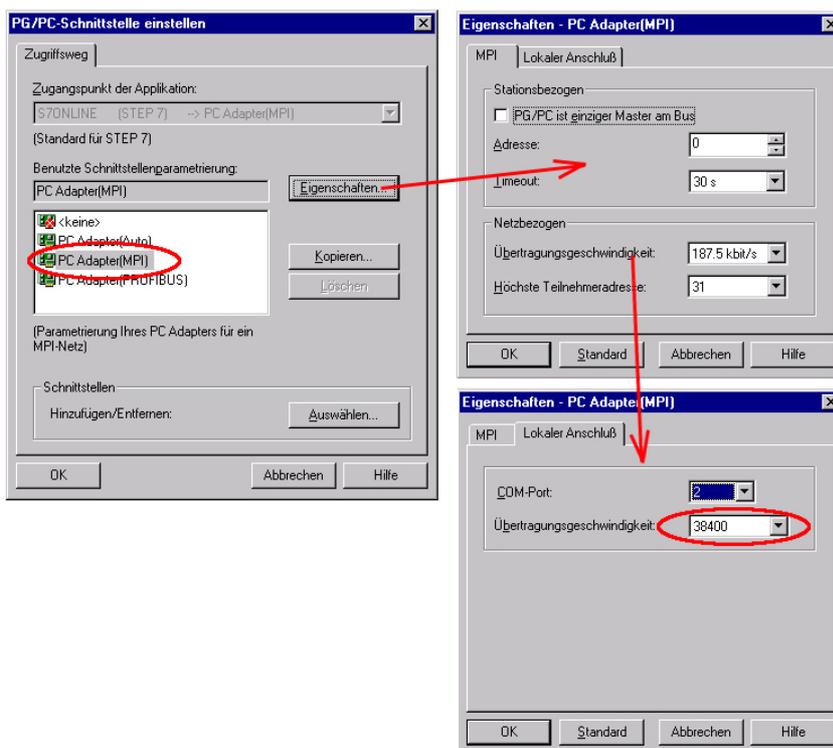
### Achtung!

Das "Green Cable" dürfen Sie ausschließlich bei VIPA CPUs mit MP<sup>2</sup>I-Schnittstellen einsetzen.

Bitte beachten Sie hierzu die Hinweise zum Einsatz der MPI-Buchse und des Green Cable!

### Vorgehensweise

- Starten Sie den Siemens SIMATIC Manager.
- Wählen Sie unter **Extras** > *PG/PC-Schnittstelle einstellen*
  - Es öffnet sich folgendes Dialogfenster, in dem Sie die zu verwendende MPI-Schnittstelle konfigurieren können:



- Wählen Sie in der Auswahlliste "PC Adapter (MPI)" aus; ggf. müssen Sie diesen erst hinzufügen.
- Klicken auf [Eigenschaften].
  - In den folgenden 2 Unterdialogen können Sie, wie in der Abbildung gezeigt, Ihren PC-Adapter konfigurieren.



### Hinweis!

Bitte beachten Sie, dass Sie bei Einsatz des Green Cable die Übertragungsgeschwindigkeit auf 38400 Baud einstellen.

## Hinweise zum Green Cable

Das Green Cable ist ein grünes Verbindungskabel, das ausschließlich zum Einsatz an VIPA System-Komponenten konfektioniert ist.

Das Green Cable ist ein Programmier- und Downloadkabel für VIPA CPUs mit MP<sup>2</sup>I-Buchse sowie VIPA Feldbus-Master. Sie erhalten das Green Cable von VIPA unter der Best.-Nr.: VIPA 950-0KB00.



Mit dem Green Cable können Sie:

- *Projekte seriell übertragen*  
Unter Umgehung aufwändiger Hardware (MPI-Adapter, etc.) können Sie über das Green Cable eine serielle Punkt-zu-Punkt-Verbindung über die MP<sup>2</sup>I-Schnittstelle realisieren.
- *Firmware-Updates der CPUs und Feldbus-Master durchführen*  
Über das Green Cable können Sie unter Einsatz eines Upload-Programms die Firmware aller aktuellen VIPA CPUs mit MP<sup>2</sup>I-Buchse sowie bestimmte Feldbus-Master (s. Hinweis) aktualisieren.



### Wichtige Hinweise zum Einsatz des Green Cable

Bei Nichtbeachtung der nachfolgenden Hinweise können Schäden an den System-Komponenten entstehen.

Für Schäden, die aufgrund der Nichtbeachtung dieser Hinweise und bei unsachgemäßem Einsatz entstehen, übernimmt die VIPA keinerlei Haftung!



### Hinweis zum Einsatzbereich

Das Green Cable darf ausschließlich direkt an den hierfür vorgesehenen Buchsen der VIPA-Komponenten betrieben werden (Zwischenstecker sind nicht zulässig). Beispielsweise ist vor dem Stecken des Green Cable ein gestecktes MPI-Kabel zu entfernen.

Zurzeit unterstützen folgende Komponenten das Green Cable:

VIPA CPUs mit MP<sup>2</sup>I-Buchse sowie die Feldbus-Master von VIPA.



### Hinweis zur Verlängerung

Die Verlängerung des Green Cable mit einem weiteren Green Cable bzw. die Kombination mit weiteren MPI-Kabeln ist nicht zulässig und führt zur Beschädigung der angeschlossenen Komponenten!

Das Green Cable darf nur mit einem 1:1 Kabel (alle 9 Pin 1:1 verbunden) verlängert werden.

**Transfer über MMC**

Als externes Speichermedium kommt eine MMC zum Einsatz. Die MMC (**M**ulti **M**edia **C**ard) dient auch als externes Transfermedium für Programme und Firmware, da Sie unter anderem das PC-kompatible FAT16 Filesystem besitzt. Mit Urlöschen oder PowerON wird automatisch von der MMC gelesen. Es dürfen sich mehrere Projekte und Unterverzeichnisse auf einem MMC-Speichermodul befinden. Bitte beachten Sie, dass sich Ihr aktuelles Projekt im Root-Verzeichnis befindet und einen der nachfolgend beschriebenen Dateinamen hat.

**Transfer MMC→RAM→ROM**

Immer nach Urlöschen und PowerON versucht die CPU ein Anwenderprogramm von der MMC in das batteriegepufferte RAM bzw. in den Flash-Speicher zu laden.

Hierbei können Sie je nach gewünschter Funktionalität folgende Dateinamen für Ihr Projekt vergeben:

- **S7PROG.WLD**  
Nach Urlöschen wird das Anwenderprogramm S7PROG.WLD in das batteriegepufferte RAM übertragen.
- **S7PROGF.WLD (ab Firmware-Version V. 3.8.6)**  
Nach Urlöschen wird das Anwenderprogramm S7PROGF.WLD in das batteriegepufferte RAM und zusätzlich in den Flash-Speicher übertragen. Ein Zugriff auf den Flash-Speicher erfolgt nur bei leerer Pufferbatterie, sofern keine MMC mit Anwenderprogramm gesteckt ist.
- **AUTOLOAD.WLD**  
Nach PowerON wird das Anwenderprogramm AUTOLOAD.WLD in das batteriegepufferte RAM übertragen.

**Transfer RAM→MMC→ROM**

Bei einer in der CPU gesteckten MMC wird durch ein Schreibbefehl der Inhalt des batteriegepufferten RAMs als **S7PROG.WLD** auf die MMC übertragen. Den Schreibbefehl starten Sie aus dem Hardware-Konfigurator von Siemens über **Zielsystem > RAM nach ROM kopieren**. Während des Schreibvorgangs blinkt die "MC"-LED. Erlischt die LED, ist der Schreibvorgang beendet. Gleichzeitig erfolgt ein Schreibvorgang in den interne Flash-Speicher der CPU. Sollte keine MMC gesteckt sein, so erhalten Sie eine Fehlermeldung über ungenügenden Speicher.

**Kontrolle des Transfervorgangs**

Nach einem Schreibvorgang auf die MMC wird ein entsprechendes ID-Ereignis im Diagnosepuffer der CPU eingetragen. Zur Anzeige der Diagnoseeinträge gehen Sie im Siemens SIMATIC Manager auf **Zielsystem > Baugruppenzustand**. Über das Register "Diagnosepuffer" gelangen Sie in das Diagnosefenster. Folgende Ereignisse können auftreten:

Ereignis-ID	Bedeutung
0xE100	MMC-Zugriffsfehler
0xE101	MMC-Fehler Filesystem
0xE102	MMC-Fehler FAT
0xE200	MMC Schreiben beendet
0xE300	Internes Flash Schreiben beendet
0xE310	Internes Flash Lesen beendet (Nachladen nach Batterieausfall)

Nähere Informationen zu den Ereignis-IDs finden am Ende des Kapitels.



**Hinweis!**

Ist das Anwenderprogramm größer als der Anwenderspeicher in der CPU, wird der Inhalt der MMC nicht in die CPU übertragen. Führen Sie vor der Übertragung eine Komprimierung durch, da keine automatische Komprimierung durchgeführt wird.

## Betriebszustände

### Übersicht

Die CPU kennt 3 Betriebszustände:

- Betriebszustand STOP
- Betriebszustand ANLAUF
- Betriebszustand RUN

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 Prozessabbilds 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 Übergang von STOP nach RUN erfolgt ein Sprung in den Anlauf-Organisationsbaustein OB100. Die Länge des OBs ist nicht beschränkt. Auch wird der Ablauf 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
- STOP-LED    aus

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

### Betriebszustand RUN

- Das Anwenderprogramm im OB1 wird zyklisch bearbeitet, wobei zusätzlich alarmgesteuert weitere Programmteile eingeschachtelt werden können.
- Alle im Programm gestarteten Zeiten und Zähler laufen und das Prozessabbild wird zyklisch aktualisiert.
- Das BASP-Signal (Befehlsausgabesperre) wird deaktiviert, d.h. alle digitalen Ausgänge sind freigegeben.
- RUN-LED     an
- STOP-LED    aus

# Urlöschen

## Übersicht

Beim Urlöschen wird der komplette Anwenderspeicher gelöscht. Ihre Daten auf der Memory Card bleiben erhalten.

Sie haben 2 Möglichkeiten zum Urlöschen:

- Urlöschen über Betriebsartenschalter
- Urlöschen über Siemens SIMATIC Manager



### Hinweis!

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

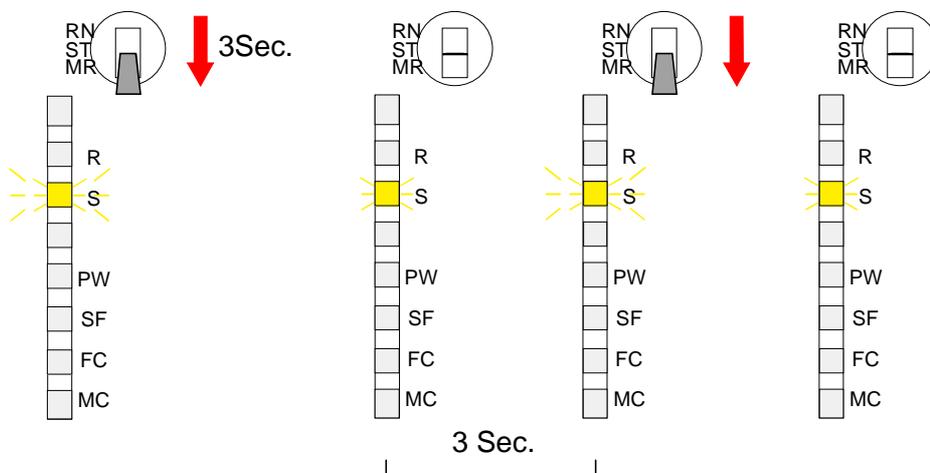
### Voraussetzung

Ihre CPU muss sich im STOP-Zustand befinden. Stellen Sie hierzu den CPU-Betriebsartenschalter auf "ST" → die S-LED leuchtet.

### Urlöschen

- Bringen Sie den Betriebsartenschalter in Stellung MR und halten Sie ihn ca. 3 Sekunden. → Die S-LED geht von Blinken über in Dauerlicht.
- Bringen Sie den Betriebsartenschalter in Stellung ST und innerhalb von 3 Sekunden kurz in MR dann wieder auf ST. → Die S-LED blinkt (Urlösch-Vorgang).
- Das Urlöschen ist abgeschlossen, wenn die S-LED in Dauerlicht übergeht → Die S-LED leuchtet.

Die nachfolgende Abbildung zeigt nochmals die Vorgehensweise:



**Automatisch nachladen**

Nun versucht die CPU Daten von der MMC neu zu laden. → Die unterste LED (MC) blinkt.

Nach dem Nachladen erlischt die LED. Abhängig von der Einstellung des Betriebsartenschalters bleibt die CPU in STOP bzw. geht in RUN.

**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 S-LED.

Geht die S-LED in Dauerlicht über, ist der Urlöschvorgang abgeschlossen.

**Automatisch nachladen**

Nun versucht die CPU Daten von der MMC neu zu laden. → Die unterste LED (MC) blinkt.

Nach dem Nachladen erlischt die LED. Abhängig von der Einstellung des Betriebsartenschalters bleibt die CPU in STOP bzw. geht in RUN.

**Rücksetzen auf Werkseinstellung**

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 defaultmäßig auf 2 zurückgestellt wird!

- Drücken Sie den Reset-Schalter für ca. 30 Sekunden nach unten. Hierbei blinkt die ST-LED. Nach ein paar Sekunden wechselt die Anzeige zu statischem Licht. Zählen Sie die Anzahl des statischen Lichts, da die Anzeige jetzt wechselt zwischen statischem Licht und Blinken.
- Nach dem 6. mal statischem Licht 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.
- Schalten Sie die Stromzufuhr aus und wieder an.

## Firmwareupdate

### Übersicht

Ab der CPU-Firmware-Version 3.3.3 haben Sie die Möglichkeit mittels einer MMC einen Firmwareupdate für CPU, Bus-Master und CP durchzuführen. Die 2 aktuellsten Firmwarestände finden Sie im Service-Bereich auf [www.vipa.de](http://www.vipa.de).



### Achtung!

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

### CPU Firmware-Version ermitteln

Den ausgelieferten Firmwarestand können Sie einem Aufkleber entnehmen, der sich auf der Rückseite der entsprechenden Komponente befindet.

Sie haben auch die Möglichkeit im Siemens SIMATIC Manager den aktuellen Firmwarestand Ihrer CPU auszulesen. Gehen Sie hierzu über Ihr PG bzw. Ihren PC mit der CPU online und starten den Siemens SIMATIC Manager.

Über **Zielsystem** > *Baugruppenzustand*, Register "Allgemein" wird der aktuelle CPU-Firmwarestand ermittelt und angezeigt.

**MMC-Update mittels reservierter Dateinamen**

In der CPU 21x können mittels reservierter Dateinamen per MMC-Karte die Firmware-Updates folgender Komponenten aufgespielt werden:

Komponente	möglich ab CPU-Version	Dateiname	Neuer Dateiname auf MMC
CPU	V. 3.3.3	Best-Nr._Ausgabestand.Version	firmware.bin
PROFIBUS-Master	V. 3.0.5	Bb000089.Version	dpm00.bin <sup>*)</sup>
CANopen-Master	V. 3.4.8 und CAN-Master V. 1.0.7	VIPA_Best.-Nr._Version.bin	can00.bin
CP (21x-2BT10)	V. 3.6.7	px000018_Version.zip	px000018.pkg

<sup>\*)</sup> Bei Einsatz der CPU-Firmware V 3.3.3 muss der Firmwareupdate-Dateiname dpm.bin lauten!

**Firmware laden und mit reserviertem Namen auf MMC übertragen**

- Gehen Sie auf [www.vipa.de/support/firmware](http://www.vipa.de/support/firmware)
- Navigieren Sie zu *System 200V*. Hier finden Sie für die gewünschte Komponente die entsprechende Firmware abgelegt.
- Downloaden Sie die entsprechende Datei. Zur Kontrolle können Sie den Aufbau des Dateinamens der Tabelle oben entnehmen.
- Kopieren Sie die Datei auf Ihre MMC und benennen Sie die Datei gemäß der Tabelle um.

**Firmware von MMC in CPU übertragen**

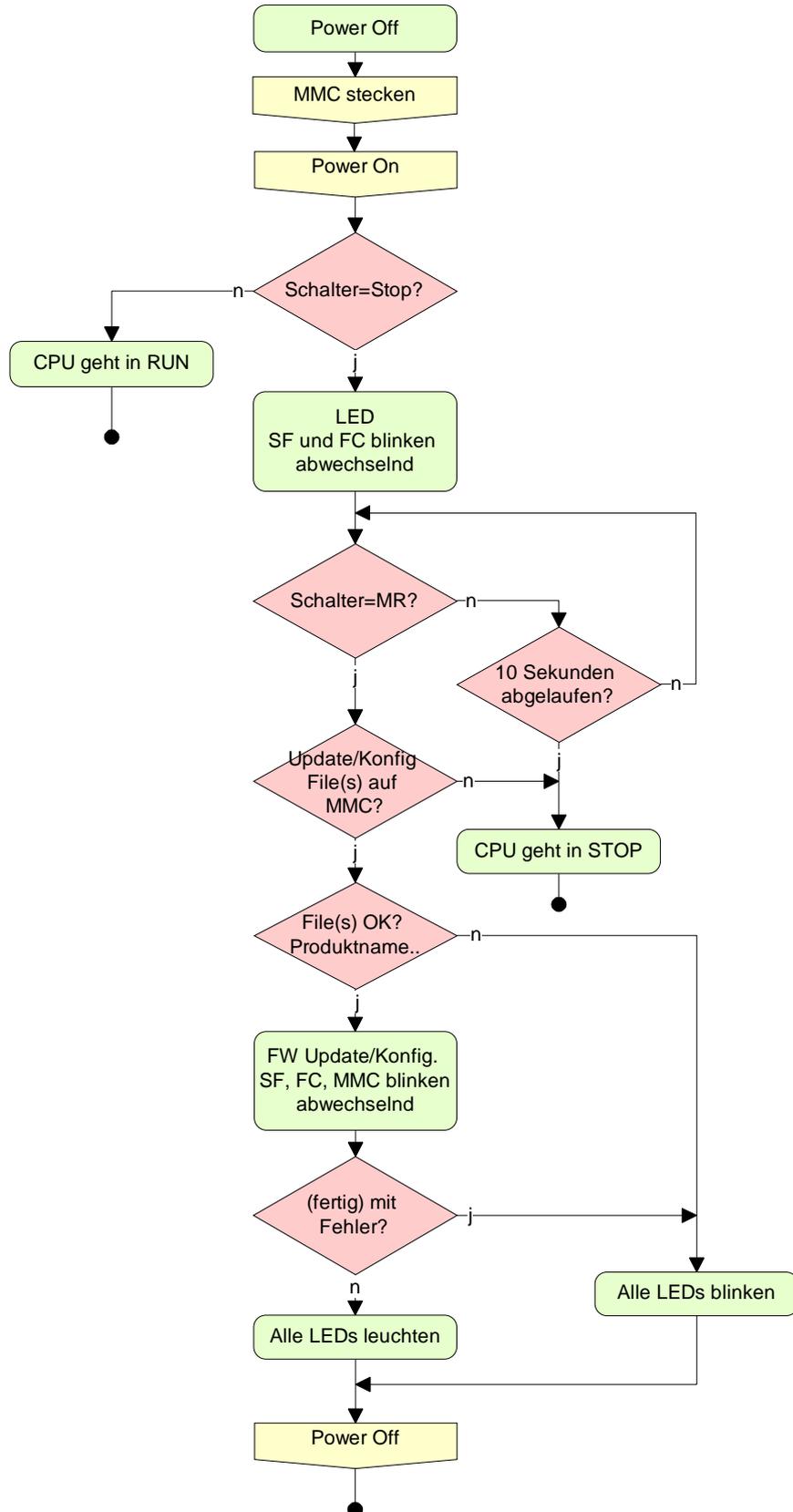
- Bringen Sie den RUN-STOP-Schalter Ihrer CPU in Stellung STOP.
- Schalten Sie die Spannungsversorgung aus.
- Stecken Sie die MMC mit der Firmware in die CPU. Achten Sie hierbei auf die Steckrichtung der MMC.
- Schalten Sie die Spannungsversorgung ein.
- Nach einer kurzen Hochlaufzeit zeigt das abwechselnde Blinken der LEDs SF und FC an, dass auf der MMC die Firmware-Datei gefunden wurde.
- Sie starten die Übertragung der Firmware, sobald Sie innerhalb von 10s den RUN/STOP-Schalter kurz nach MRES tippen. Die CPU zeigt die Übertragung über ein LED-Lauflicht an.
- Während des Update-Vorgangs blinken die LEDs SF, FC und MC abwechselnd. Dieser Vorgang kann mehrere Minuten dauern.
- Das Update ist fehlerfrei beendet, wenn alle CPU-LEDs leuchten. Blinken diese schnell, ist ein Fehler aufgetreten.

**Hinweis!**

Ab der WinNCS-Version 3.1.1 können Sie über das Parametrierfenster einen Firmwareupdate des CP-Teils online durchführen. Diese Funktion unterstützt der CP-Teil ab der CP-Firmware-Version 2.1.2.

**Flussdiagramm zum Firmware-update**

Das nachfolgende Flussdiagramm soll das CPU-Verhalten beim Firmware-update nochmals verdeutlichen:



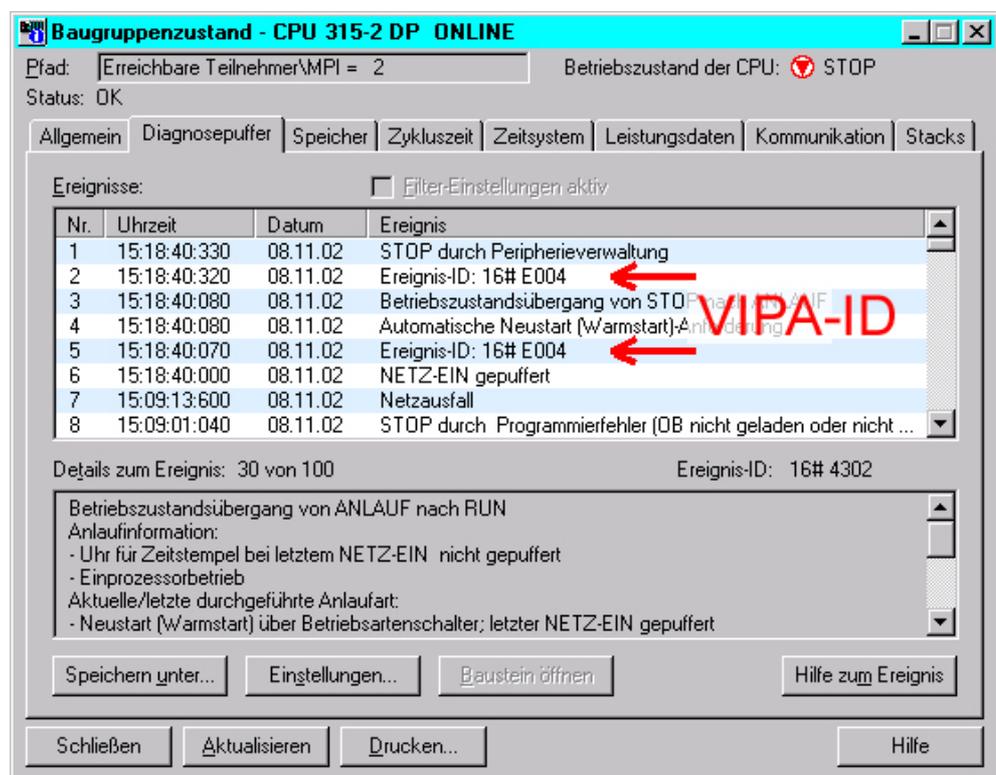
## VIPA-spezifische Diagnose-Einträge

### 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, die ausschließlich in Form einer Ereignis-ID angezeigt werden.

### Anzeige der Diagnoseeinträge

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



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

Auf der Folgeseite finden Sie eine Übersicht der VIPA-spezifischen Ereignis-IDs.

## Übersicht der Ereignis-ID

Ereignis-ID	Bedeutung
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 Slavekonfiguration
0xE012	Fehler bei Parametrierung
0xE013	Fehler bei Schieberegisterzugriff auf VBUS 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 Masterperipherie
0xE019	Fehler bei Erkennung des Standard Rückwandbus Systems
0xE01A	Fehler bei Erkennung der Betriebsart (8 / 9 Bit)
0xE0CC	Kommunikationsfehler MPI / Seriell
0xE100	MMC-Zugriffsfehler
0xE101	MMC-Fehler Filesystem
0xE102	MMC-Fehler FAT
0xE104	MMC Fehler beim Speichern
0xE200	MMC Schreiben beendet (Copy Ram2Rom)
0xE210	MMC Lesen beendet (Nachladen nach Urlöschen)
0xE300	Internes Flash Schreiben beendet (Copy RAM to ROM)
0xE310	Internes Flash Lesen beendet (Nachladen nach Batterieausfall)

## 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.



### Hinweis!

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

Die Statusbearbeitung kann durch Sprungbefehle oder Zeit- und Prozessalarme unterbrochen werden. 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:

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

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.

**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.

*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 21x 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.



## Teil 5 Einsatz CPU 21x-2BT10 unter TCP/IP

### Überblick

Im folgenden Kapitel ist der Einsatz der CPU 21x-2BT10 und die Kommunikation unter TCP/IP beschrieben. Bitte beachten Sie den Abschnitt "Schnelleinstieg". Hier finden Sie in komprimierter Form alle Informationen, die für die Projektierung der CPU 21x-2BT10 erforderlich sind. Nach dem "Schnelleinstieg" sind diese Punkte näher beschrieben.

### Inhalt

Thema	Seite
<b>Teil 5 Einsatz CPU 21x-2BT10 unter TCP/IP</b> .....	<b>5-1</b>
Industrial Ethernet in der Automatisierung.....	5-2
ISO/OSI-Schichtenmodell .....	5-3
Grundbegriffe .....	5-6
Protokolle .....	5-7
IP-Adresse und Subnetz .....	5-10
Planung eines Netzwerks .....	5-12
Kommunikationsmöglichkeiten des CP .....	5-15
Funktionsübersicht .....	5-18
Schnelleinstieg .....	5-19
Hardware-Konfiguration .....	5-23
Kommunikationsverbindungen projektieren.....	5-26
SEND/RECEIVE im SPS-Anwenderprogramm.....	5-32
Projekt transferieren.....	5-37
NCM-Diagnose - Hilfe zur Fehlersuche .....	5-39
Kopplung mit Fremdsystemen.....	5-42
Beispiel zur Kommunikation CPU 21x-2BT10.....	5-45

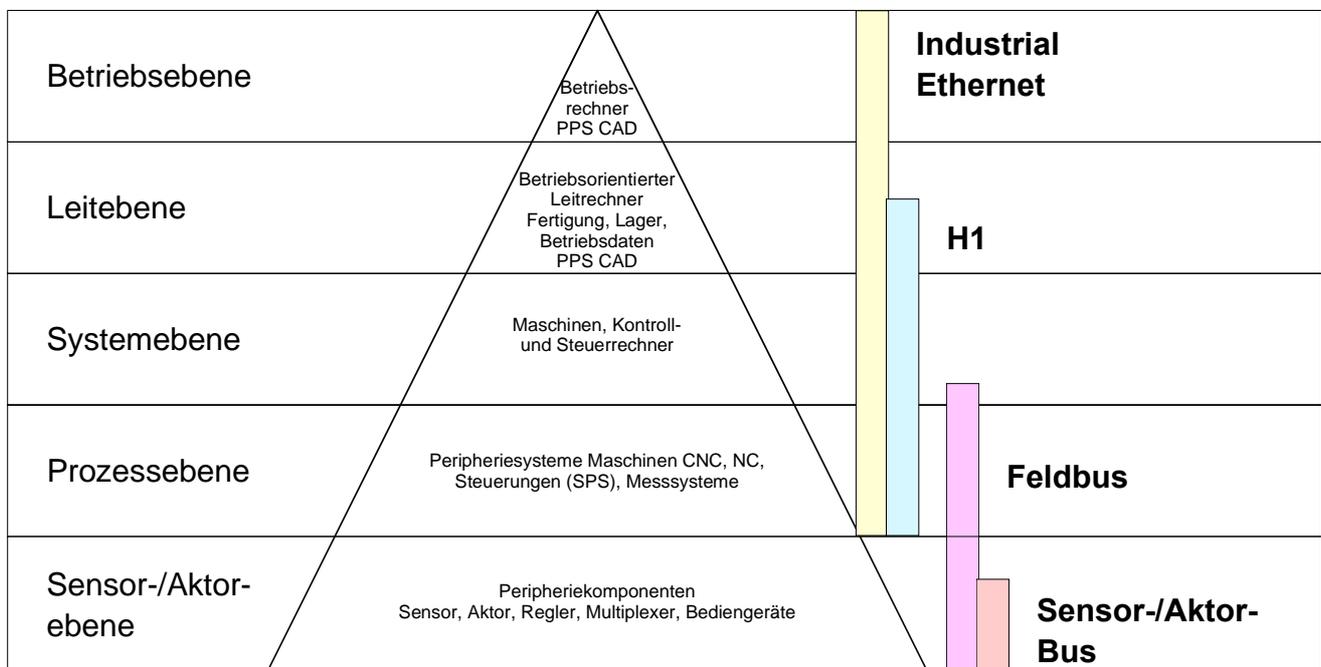
# Industrial Ethernet in der Automatisierung

## 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.

## 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, die 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	Funktion
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 (Sicherung)
Schicht 1	Physical Layer (Bitübertragung)

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

**Schichten****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 Rahmengrenzen 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 (**Media Access Control**) ist die untere Schicht und steuert die Art, wie Sender einen einzigen Übertragungskanal gemeinsam nutzen

Die LLC (**Logical Link Control**) ist die obere Schicht und stellt die Verbindung für die Übertragung der Datenrahmen von einem Gerät zum anderen her.

**Schicht 3** Netzwerkschicht (network layer)

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.

**Schicht 4** Transportschicht (transport layer)

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.

**Schichten**  
**Fortsetzung ...****Schicht 5** Sitzungsschicht (session layer)

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.

**Schicht 6** Darstellungsschicht (presentation layer)

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.

**Schicht 7** Anwendungsschicht (application layer)

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.

## Grundbegriffe

- Netzwerk (LAN)** Ein Netzwerk bzw. LAN (**L**ocal **A**rea **N**etwork) verbindet verschiedene Netzwerkstationen so, dass diese miteinander kommunizieren können. Netzwerkstationen können PCs, IPCs, TCP/IP-Baugrupen, etc. sein. Die Netzwerkstationen sind, durch einen Mindestabstand getrennt, mit dem Netzwerkkabel verbunden. Die Netzwerkstationen und das Netzwerkkabel 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 21xNET hat einen Twisted-Pair-Anschluss. Das Twisted Pair Kabel besteht aus 8 Adern, die paarweise miteinander verdrillt sind. Aufgrund der Verdrillung 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.

## 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 (siehe "ISO/OSI-Schichtenmodell" weiter oben).

Folgende Protokolle kommen in der CPU 21xNET zum Einsatz:

- TCP/IP
- UDP
- RFC1006 (ISO-ON-TCP)

Nachfolgend sind diese Protokolle kurz aufgeführt:

---

### TCP/IP

TCP/IP-Protokolle stehen auf allen derzeit bedeutenden Systemen zur Verfügung. Dies gilt am unteren Ende für einfache PCs, über die typischen Mini-Rechner, bis hinauf zu Großrechnern (auch für IBM-Systeme existieren TCP/IP-Implementierungen) und Spezialrechnern wie Vektorrechner und Parallelrechner.

Durch die weite Verbreitung von Internetzugängen und -Anschlüssen wird TCP/IP sehr häufig für den Aufbau heterogener Systemverbunde verwendet.

Hinter TCP/IP, das für die Abkürzungen **T**ransmission **C**ontrol **P**rotocol und **I**nternet **P**rotocol steht, verbirgt sich eine ganze Familie von Protokollen und Funktionen.

TCP und IP sind nur zwei der für den Aufbau einer vollständigen Architektur erforderlichen Protokolle. Die Anwendungsschicht stellt Programme wie "FTP" und "Telnet" auf PC-Seite zur Verfügung.

Die Anwendungsschicht des Ethernet-Teils der CPU 21xNET ist mit dem Anwenderprogramm unter Verwendung der Standardhantierungsbausteine definiert.

Diese Anwendungsprogramme nutzen für den Datenaustausch die Transportschicht mit den Protokollen TCP oder UDP, die wiederum mit dem IP-Protokoll der Internetschicht kommunizieren.

- IP**
- Das IP (Internet **P**rotokoll) deckt die Netzwerkschicht (Schicht 3) des ISO/OSI-Schichtmodells ab.
- Die Aufgabe des IP besteht darin, Datenpakete von einem Rechner über mehrere Rechner hinweg zum Empfänger zu senden. Diese Datenpakete sind sogenannte Datagramme. Das IP gewährleistet weder die richtige Reihenfolge der Datagramme, noch die Ablieferung beim Empfänger.
- Zur eindeutigen Unterscheidung zwischen Sender und Empfänger kommen 32Bit-Adressen (IP-Adressen) zum Einsatz, die bei *IPv4* in vier Oktetts (genau 8Bit) geschrieben werden, z.B. 172.16.192.11.
- Diese Internetadressen werden weltweit eindeutig vergeben, so dass jeder Anwender von TCP/IP mit allen anderen TCP/IP Anwendern kommunizieren kann.
- Ein Teil der Adresse spezifiziert das Netzwerk, der Rest dient zur Identifizierung der Rechner im Netzwerk. Die Grenze zwischen Netzwerkanteil und Host-Anteil ist fließend und hängt von der Größe des Netzwerkes ab.
- Um IP-Adressen zu sparen, werden sogenannte *NAT-Router* eingesetzt, die eine einzige offizielle IP-Adresse besitzen und das Netzwerk hinter diesem Rechner abschotten. Somit können im privaten Netzwerk dann beliebige IP-Adressen vergeben werden.
- TCP**
- Das TCP (**T**ransmission **C**ontrol **P**rotokoll) setzt direkt auf dem IP auf, somit deckt das TCP die Transportschicht (Schicht 4) auf dem OSI-Schichtenmodell ab. TCP ist ein verbindungsorientiertes End-to-End-Protokoll und dient zur logischen Verbindung zwischen zwei Partnern.
- TCP gewährleistet eine folgerichtige und zuverlässige Datenübertragung. Hierzu ist ein relativ großer Protokoll-Overhead erforderlich, der folglich die Übertragung verlangsamt.
- Jedes Datagramm wird mit einem mindestens 20 Byte langen Header versehen. In diesem Header befindet sich auch eine Folgenummer, mit der die richtige Reihenfolge erkannt wird. So können in einem Netzwerkverbund die einzelnen Datagramme auf unterschiedlichen Wegen zum Ziel gelangen.
- Bei TCP-Verbindungen wird die Gesamtdatenlänge nicht übermittelt. Aus diesem Grund muss der Empfänger wissen, wie viele Bytes zu einer Nachricht gehören. Zur Übertragung von Daten mit variabler Länge können Sie die Längenangabe den Nutzdaten voranstellen und diese Längenangabe entsprechend auf der Gegenseite auswerten.
- Eigenschaften**
- Zur Adressierung werden neben der IP-Adresse Ports verwendet. Eine Port-Adresse sollte im Bereich 2000...65535 liegen. Ferne und lokale Ports dürfen bei nur 1 Verbindung identisch sein.
  - Unabhängig vom eingesetzten Protokoll sind zur Datenübertragung auf SPS-Seite die VIPA-Hantierungsbausteine AG\_SEND (FC5) und AG\_RECV (FC6) erforderlich.

---

**UDP**

Das UDP (**U**ser **D**atagramm **P**rotocol) ist ein verbindungsloses Transportprotokoll. Es wurde im RFC768 (Request for Comment) definiert. Im Vergleich zu TCP hat es wesentlich weniger Merkmale.

Die Adressierung erfolgt durch Portnummern.

UDP ist ein schnelles ungesichertes Protokoll, da es sich weder um fehlende Datenpakete kümmert, noch um die Reihenfolge der Pakete.

---

**ISO-on-TCP  
RFC1006**

Da der TCP-Transportdienst streamorientiert ist, bedeutet dies, dass einzelne vom Anwender zusammengestellte Datenpakete nicht unbedingt in der gleichen Paketierung beim Teilnehmer ankommen. Je nach Datenvolumen können Pakete zwar in der gleichen Reihenfolge aber anders paketierte Pakete ankommen, so dass der Empfänger die einzelnen Paketgrenzen nicht mehr erkennen kann. Beispielsweise werden 2x 10Byte-Pakete geschickt, die auf der Gegenseite als 20Byte-Paket ankommen. Aber gerade die richtige Paketierung ist für die meisten Anwendungen unerlässlich.

Dies bedeutet, dass oberhalb von TCP ein zusätzliches Protokoll erforderlich ist. Diese Aufgabe erfüllt der Protokollaufsatz RFC1006 (ISO-on-TCP). Der Protokollaufsatz beschreibt die Arbeitsweise einer ISO Transportschnittstelle (ISO 8072) auf der Basis des Transportinterfaces TCP (RFC793).

Das dem RFC1006 zugrunde liegende Protokoll ist in seinen wesentlichen Teilen identisch zu TP0 (Transport Protokoll, Class 0) in ISO 8073.

Da RFC1006 als Protokollaufsatz zu TCP gefahren wird, erfolgt die Dekodierung im Datenteil des TCP-Pakets.

**Eigenschaften**

- Im Gegensatz zu TCP wird hier der Empfang eines Telegramms bestätigt.
- Zur Adressierung werden neben der IP-Adresse anstelle von Ports TSAPs verwendet. Die TSAP-Länge kann 1 ... 16Byte betragen. Die Eingabe kann im ASCII- oder Hex-Format erfolgen. Ferne und lokale TSAPs dürfen bei nur 1 Verbindung identisch sein.
- Unabhängig vom eingesetzten Protokoll sind zur Datenübertragung auf SPS-Seite die VIPA-Hantierungsbausteine AG\_SEND (FC5) und AG\_RECV (FC6) erforderlich.
- Im Gegensatz zu TCP können über RFC1006 unterschiedliche Telegrammlängen empfangen werden.

## IP-Adresse und Subnetz

**Aufbau IP-Adresse** Industrial Ethernet unterstützt 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  
 Die IP-Adressen werden vom Netzwerkadministrator vergeben.

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

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

Subnet-Maske	binär alle "1"		binär alle "0"
IPv4 Adresse	Net-ID	Host-ID	
Subnet-Maske und IPv4 Adresse	Net-ID	Subnet-ID	<i>neue</i> Host-ID

**Subnetz** Eine TCP-basierte Kommunikation per Punkt-zu-Punkt-, Hub- oder Switch-Verbindung ist nur zwischen Stationen mit identischer Network-ID und Subnet-ID möglich! Unterschiedliche Bereiche sind mit einem Router zu verknüpfen.  
 Über die Subnet-Maske haben Sie die Möglichkeit, die Ressourcen ihren Bedürfnissen entsprechend zu ordnen. So erhält z.B. jede Abteilung ein eigenes Subnetz und stört damit keine andere Abteilung.

**Adresse bei Erst-inbetriebnahme** Bei der Erstinbetriebnahme besitzt der CP keine IP-Adresse. Für die Adresszuweisung haben Sie folgende Möglichkeiten:

- Im Siemens SIMATIC Manager die PG/PC-Schnittstelle auf "TCP/IP... RFC1006" einstellen, über "Ethernet-Adresse vergeben..." den CP suchen und diesem IP-Parameter zuweisen. Nach der Zuweisung werden die IP-Parameter sofort ohne CPU-Neustart übernommen.
- Über ein "Minimalprojekt" dem CP IP-Adresse und Subnet-Maske zuweisen und das Projekt über MMC oder MPI in die CPU übertragen. Nach dem Neustart der CPU und nach Umstellen der PG/PC-Schnittstelle auf "TCP/IP... RFC1006" können Sie nun online über den CP Ihre CPU projektieren.

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

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

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** Zur Bildung privater IP-Netze sind gemäß RFC1597/1918 folgende Adressbereiche vorgesehen:

Netzwerk Klasse	von IP	bis IP	Standard Subnet-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.)

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.

### Reservierte Host-IDs

Einige Host-IDs sind für spezielle Zwecke reserviert.

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



### Hinweis!

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

## Planung eines Netzwerks

### Normen und Richtlinien

Zur Kommunikation zwischen einzelnen Stationen gibt es gewisse Vorschriften und Regeln die einzuhalten sind. Hierbei werden die Form des Datenprotokolls, das Zugriffsverfahren auf den Bus und weitere, für die Kommunikation wichtige Grundlagen definiert.

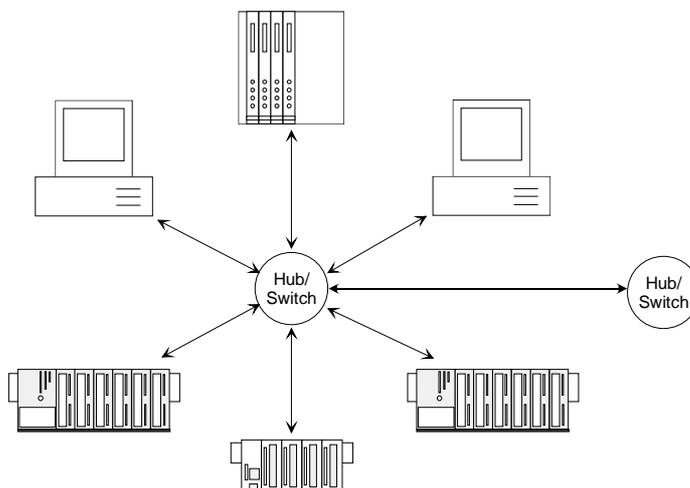
Basierend auf den von ISO festgelegten Standards und Normen wurde die CPU 21xNET von VIPA entwickelt.

In den folgenden internationalen und nationalen Gremien sind Normen und Richtlinien für Netzwerktechnologien festgelegt worden:

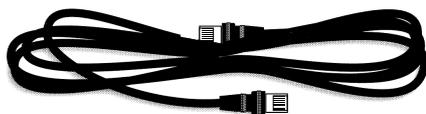
ANSI	American National Standards Institute Hier werden zur Zeit in der ANSI X3T9.5 Vereinbarungen für LANs mit hohen Übertragungsgeschwindigkeiten (100 MB/s) auf Glasfaserbasis formuliert. (FDDI) Fibre Distributed Data Interface.
CCITT	Committee Consultative Internationale de Telephone et Telegraph. Von diesem beratenden Ausschuss werden unter anderem die Vereinbarungen für die Anbindung von Industriekommunikationsnetzen (MAP) und Büronetzen (TOP) an Wide Area Networks (WAN) erstellt.
ECMA	European Computer Manufacturers Association. Hier werden verschiedene Standards für MAP und TOP erarbeitet.
EIA	Electrical Industries Association (USA) Standardfestlegungen wie RS-232 (V.24) und RS-511 sind in diesem Ausschuss erarbeitet worden.
IEC	International Electrotechnical Commission. Hier werden einzelne spezielle Standards festgelegt. z.B. für Feld Bus.
ISO	International Organisation for Standardization. In diesem Verband der nationalen Normungsstellen wurde das OSI-Modell entwickelt (ISO/TC97/SC16). Es gibt den Rahmen vor, an den sich die Normungen für die Datenkommunikation halten sollen. ISO Standards gehen über in die einzelnen nationalen Standards wie z.B. UL und DIN.
IEEE	Institute of Electrical and Electronic Engineers (USA). In der Projektgruppe 802 werden die LAN-Standards für Übertragungsraten von 1 bis 1000MB/s festgelegt. IEEE Standards bilden häufig die Grundlage für ISO-Standards z.B. IEEE 802.3 = ISO 8802.3.

## Übersicht der Komponenten

Der CP ist ausschließlich für den Einsatz in einem Twisted-Pair-Netz geeignet. Bei einem Twisted-Pair-Netz werden alle teilnehmenden Stationen sternförmig über Twisted-Pair-Kabel mit einem Hub/Switch verbunden, der seinerseits mit weiteren Hub/Switch kommunizieren kann. Zwei verbundene Stationen bilden ein Segment, wobei die Länge des Twisted-Pair-Kabels zwischen den Stationen max. 100m betragen darf.



### Twisted Pair Kabel



Bei einem Twisted Pair-Kabel handelt es sich um ein Kabel mit 8 Adern, die paarweise miteinander verdreht sind.

Die einzelnen Adern haben einen jeweiligen Durchmesser von 0,4 bis 0,6mm. Verwenden Sie zur Vernetzung Twisted Pair Kabel, die mindestens der Kategorie 5 entsprechen.

## Ermitteln des Netzwerkbedarfs

- Welche Fläche muss mit dem Kabelsystem abgedeckt werden?
- Wie viele Netzwerksegmente lösen am besten die physikalischen (räumlich, störungsbedingt) Gegebenheiten der Anlage?
- Wie viele Netzwerkstationen (SPS, IPC, PC, Transceiver, evtl. Bridges) sollen an das Kabelsystem angeschlossen werden?
- In welchem Abstand stehen die Netzwerkstationen voneinander getrennt?
- Welches "Wachstum" in Größe und Anzahl der Verbindungen muss das System bewältigen können?
- Welches Datenaufkommen ist zu bewältigen (Bandbreite, Zugriffe/Sec.)?

## Zeichnen des Netzwerkplans

Zeichnen Sie Ihren Netzwerkplan. Bezeichnen Sie jedes Stück Hardware, das verwendet wird (wie Stationskabel, Hub, Switch). Halten Sie die Regeln und Grenzwerte im Auge.

Messen Sie die Distanz zwischen allen Komponenten um sicher zu gehen, dass jeweils die maximale Länge nicht überschritten wird.

**Vernetzung unter NetPro**

Bitte beachten Sie, dass zur Projektierung die folgenden Software-Pakete installiert sein müssen:

- Siemens SIMATIC Manager ab V.5.1 und vipa\_21x.gsd (im Lieferumfang)
- Siemens SIMATIC NET

Damit Stationen miteinander kommunizieren können, sind die hierzu erforderlichen (Sub-)Netze nach folgenden Schritten im Siemens SIMATIC Manager bzw. NetPro zu projektieren:

- Legen Sie in Ihrem Projekt ein oder mehrere Subnetze des jeweils gewünschten Typs an.
- Passen Sie die Eigenschaften der Subnetze an.
- Schließen Sie Ihre Teilnehmer logisch an das Subnetz an.
- Richten Sie Kommunikationsverbindungen zwischen den einzelnen Stationen ein.

**Netz-Projektvarianten**

In einem Projekt können mehrere Subnetze verwaltet werden. Jede Station ist einmal anzulegen. Eine Station kann mehreren Subnetzen zugeordnet sein, indem Sie die CPs entsprechend zuordnen.

Nachfolgend sind typische Projektvarianten für Netzwerke aufgeführt:

**1 Subnetz -  
1 Projekt**

Im einfachsten Fall besteht Ihre Anlage aus Stationen, die über 1 Subnetz vom Typ Industrial Ethernet vernetzt werden sollen.

Legen Sie hierzu ein Objekt "Ethernet" an. Stationen, die im selben Projekt angelegt werden, beziehen sich auf dieses Objekt, sobald sie als Netzknoten konfiguriert werden. Diese können dann direkt ausgewählt werden. Fremdgeräte sind in diesem Subnetz bei der Projektierung als "Andere Station" einzutragen".

**2 oder mehr  
Subnetze -  
1 Projekt**

Aufgrund unterschiedlicher Aufgaben der Stationen oder aufgrund der Ausdehnung Ihrer Anlage kann es erforderlich sein, mehrere Netze zu betreiben. Hierbei können Sie mehrere Subnetze in einem Projekt anlegen und die Stationen auf einfache Weise für die Kommunikation projektieren.

**1 oder mehrere  
Subnetz - mehrere  
Teilprojekte**

Bei komplexen vernetzten Anlagen ist es sinnvoll, Anlagenteile in unterschiedlichen Teilprojekten zu verwalten. Hierbei kann es erforderlich sein, dass Sie projektübergreifende Verbindungen anzulegen haben. Hierzu steht Ihnen im Siemens SIMATIC Manager ab V 5.2 die Multiprojekt-Funktion zu Verfügung. Mit dieser Funktion können Sie unter anderem Projekte auftrennen und wieder zusammenfügen. Näheres hierzu finden Sie in Ihrer Beschreibung zum Siemens SIMATIC Manager.

**Subnetzüber-  
greifende  
Verbindungen**

Dies sind Verbindungen die aufgrund der Anlagenkomplexität in ein anderes Subnetz greifen. Die Subnetze untereinander sind über Router verbunden. Durch Angabe einer Router-Adresse bei der Hardware-Konfiguration Ihres CP, können Sie Ihren CP anweisen über diesen Router das entsprechende Subnetz für die Kommunikation einzubinden.

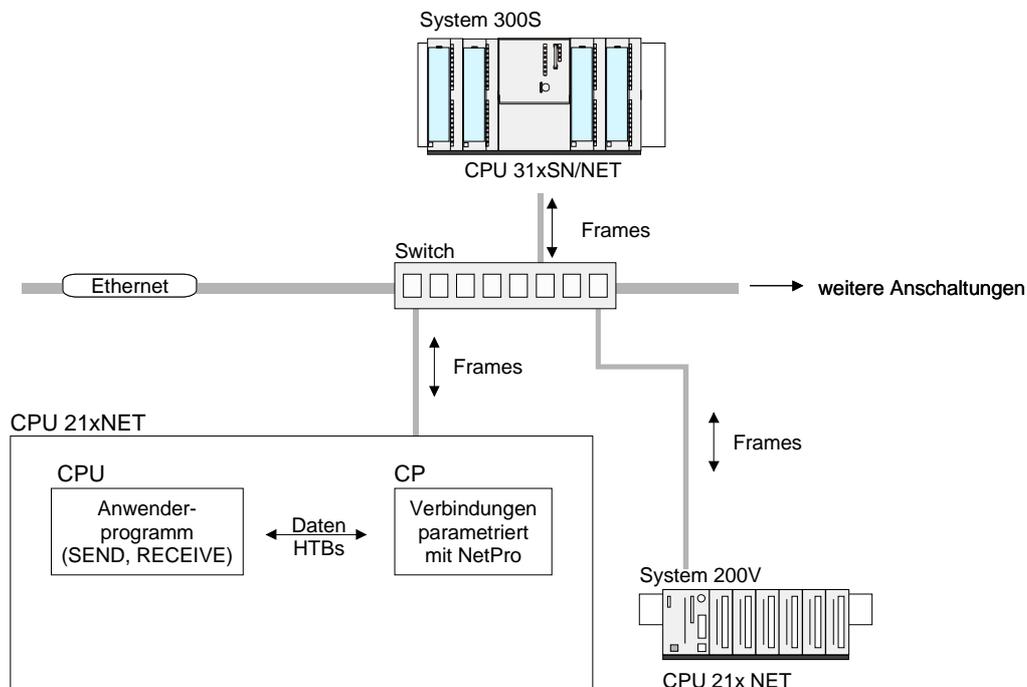
## Kommunikationsmöglichkeiten des CP

### Kommunikation zwischen CP 243 und CPU

Der interne CP der CPU 21x-2BT10 ist über ein Dual-Port-RAM direkt mit der CPU verbunden. Auf CPU-Seite findet der Datenaustausch über die VIPA-Hantierungsbausteine AG\_SEND (FC5) und AG\_RECV (FC6) statt.

Die Kommunikation über die entsprechenden Protokolle regeln Verbindungen, die unter dem Siemens-Projektier-Tool NetPro zu parametrieren sind und über MMC, MPI oder direkt über Ethernet in die CPU übertragen werden können.

Zur Übertragung über Ethernet muss sich Ihr CP mit gültigen IP-Parametern am Ethernet befinden. Hierbei kann die Zuweisung entweder über den entsprechenden Menüpunkt im Siemens SIMATIC Manager erfolgen oder über ein Minimalprojekt in dem die IP-Parameter definiert sind. Dieses Projekt können Sie über MMC oder MPI in die CPU übertragen.



### Kommunikationsarten

Der CP unterstützt folgende Kommunikationsarten:

- PG/OP-Kommunikation
- Projektierbare Verbindungen

### PG/OP-Kommunikation

Die PG/OP-Kommunikation dient zum Laden von Programmen und Konfigurationsdaten, für Test und Diagnosefunktionen sowie zum Bedienen und Beobachten einer Anlage. Hierbei können Sie über den CP (Ethernet) auf die CPU online zugreifen.

Ab der CP-Firmware-Version 1.7.4 ist ein gleichzeitiger Zugriff von bis zu 32 Teilnehmern möglich. Bitte beachten Sie, dass je eine Verbindung für PG- und OP-Kommunikation reserviert ist.

**Projektierbare Verbindungen**

Bei projektierbaren Verbindungen handelt es sich um Verbindungen zur Kommunikation zwischen SPS-Stationen. Die Verbindungen können mit dem Siemens Projektierool NetPro projiziert werden.

Die nachfolgende Tabelle zeigt die Kombinationsmöglichkeiten mit den verschiedenen Betriebsarten:

Kombinationsmöglichkeiten

Verbindungspartner	Verbindungstyp	Verbindungsaufbau	Verbindung	Betriebsart
spezifiziert in NetPro (im aktuellen Projekt)	ISO-Transport / TCP / ISO-on-TCP	aktiv/passiv	spezifiziert	SEND/RECEIVE
	UDP	-		
unspezifiziert in NetPro (im aktuellen Projekt)	TCP / ISO-on-TCP	aktiv	spezifiziert	SEND/RECEIVE
		passiv	teilspezifiziert (Port)	SEND/RECEIVE
	UDP		-	unspezifiziert
unspezifiziert in NetPro (in "unbekanntem Projekt")	TCP / ISO-on-TCP	aktiv	spezifiziert (Verbindungsname)	SEND/RECEIVE
		passiv	unspezifiziert (Verbindungsname)	SEND/RECEIVE FETCH PASSIV WRITE PASSIV
	UDP	-	unspezifiziert (Verbindungsname)	SEND/RECEIVE
Alle Broadcast-Teilnehmer	UDP	-	spezifiziert (Port, Broadcast-Adr.)	SEND
Alle Multicast-Teilnehmer	UDP	-	spezifiziert (Port, Multicast-Gruppe)	SEND/RECEIVE

Verbindungspartner Verbindungspartner sind Stationen auf der Gegenseite.

*Spezifizierte Verbindungspartner*

Jede im Siemens SIMATIC Manager projizierte Station wird in die Liste der Verbindungspartner aufgenommen. Durch Angabe einer IP-Adresse und Subnet-Maske sind diese Stationen eindeutig *spezifiziert*.

*Unspezifizierte Verbindungspartner*

Sie können aber auch einen *unspezifizierten* Verbindungspartner angeben. Hierbei kann sich der Verbindungspartner im *aktuellen Projekt* oder in einem *unbekanntem Projekt* befinden. Verbindungs-Aufträge in ein *unbekanntes Projekt* sind über einen eindeutigen Verbindungs-Namen zu definieren, der für die Projekte in beiden Stationen zu verwenden ist. Aufgrund der Zuordnung über einen Verbindungs-Namen bleibt die Verbindung selbst *unspezifiziert*.

*Alle Broadcast-Teilnehmer*

Ausschließlich bei UDP-Verbindungen können Sie hier an alle erreichbaren Broadcast-Teilnehmer senden. Der Empfang ist nicht möglich. Über einen Port und eine Broadcast-Adresse bei Sender und Empfänger werden die Broadcast-Teilnehmer spezifiziert.

*Alle Multicast-Teilnehmer*

Über diese Einstellung können Multicast-Telegramme zwischen den Multicast-Teilnehmern gesendet und empfangen werden. Durch Angabe eines Ports und einer Multicast-Gruppe für Sender und Empfänger sind die Multicast-Teilnehmer zu spezifizieren.

Verbindungstypen	<p>Für die Kommunikation stehen Ihnen folgende Verbindungstypen zur Verfügung:</p> <ul style="list-style-type: none"><li>• <b>TCP</b> bzw. <b>ISO-on-TCP</b> zur gesicherten Datenübertragung zusammenhängender Datenblöcke zwischen zwei Ethernet-Teilnehmern</li><li>• <b>UDP</b> zur ungesicherten Datenübertragung zusammenhängender Datenblöcke zwischen zwei Ethernet-Teilnehmer</li></ul>
Verbindungsaufbau	<p>Bei projektierbaren Verbindungen gibt es immer eine Station, die <i>aktiv</i> eine Verbindung aufbaut. Auf der Gegenseite wird <i>passiv</i> auf die aktive Verbindung gewartet. Erst dann können Produktiv-Daten übertragen werden.</p>
Verbindung	<p>Durch Angabe von IP-Adresse und Port/TSAP der Gegenseite wird eine Verbindung <i>spezifiziert</i>. Aktive Verbindungen sind immer spezifiziert anzugeben. Bei einer <i>unspezifizierten</i> Verbindung, die nur bei passivem Verbindungs-Aufbau möglich ist, sind IP-Adresse und Port/TSAP der Gegenseite für die Telegrammauswertung nicht erforderlich.</p> <p>Es besteht auch die Möglichkeit für <i>teilspezifizierte</i> Verbindungen. Die Teilspezifikation erfolgt hierbei über die Port-Angabe. Die Angabe einer IP-Adresse ist nicht erforderlich.</p>
Betriebsarten	<p>Je nach Verbindung stehen Ihnen folgende Betriebsarten zur Verfügung:</p> <p><i>SEND/RECEIVE</i></p> <p>Die SEND/RECEIVE-Schnittstelle ermöglicht die programmgesteuerte Kommunikation über eine projektierte Verbindung zu beliebigen Fremdstationen. Die Datenübertragung erfolgt hierbei durch Anstoß durch Ihr Anwenderprogramm. Als Schnittstelle dienen Ihnen FC5 und FC6, die Bestandteil der VIPA-Baustein-Bibliothek sind.</p> <p>Hiermit wird Ihre Steuerung in die Lage versetzt, abhängig von Prozessereignissen Nachrichten zu versenden.</p> <p><i>FETCH/WRITE PASSIV</i></p> <p>Mit den FETCH/WRITE-Diensten haben Fremdsysteme direkten Zugriff auf Speicherbereiche der CPU. Es handelt sich hierbei um "passive" Kommunikationsverbindungen, die zu projektieren sind. Die Verbindungen werden "aktiv" vom Verbindungspartner (z.B. Siemens-S5) aufgebaut.</p> <p><i>FETCH PASSIV (Daten anfordern)</i></p> <p>Mit FETCH kann ein Fremdsystem Daten anfordern.</p> <p><i>WRITE PASSIV (Daten schreiben)</i></p> <p>Hiermit kann ein Fremdsystem in den Datenbereich der CPU schreiben.</p>

## Funktionsübersicht

### Übersicht

Nachfolgend sind die Funktionen aufgeführt die ab der CP-Firmware-Version 1.7.4 vom CP-Teil der CPU 21x-2BT10 unterstützt werden:

### Projektierbare Verbindungen

Funktion	Eigenschaft
Maximale Anzahl projektierbarer Verbindungen	16
TCP-Verbindungen	SEND, RECEIVE, FETCH PASSIV, WRITE PASSIV Verbindungsaufbau aktiv und passiv, unterstützt un spezifizierten Verbindungspartner
ISO-on-TCP-Verbindungen (RFC1006)	SEND, RECEIVE, FETCH PASSIV, WRITE PASSIV Verbindungsaufbau aktiv und passiv, unterstützt un spezifizierten Verbindungspartner
ISO-Transport UDP-Verbindungen	SEND und RECEIVE SEND und RECEIVE Die Übertragung der Telegramme erfolgt nicht quittiert, d.h. der Verlust von Nachrichten wird vom Sendebaustein nicht erkannt.
UDP-Broadcast-Verb.	SEND
UDP-Multicast-Verb.	SEND und RECEIVE (max. 16 Multicast-Kreise)
Datenblocklänge	max. 64kByte (max. 2KByte bei UDP)
VIPA-Hantierungsbausteine	Für Verbindungsaufträge auf SPS-Seite: AG_SEND (FC5) / AG_RECEIVE (FC6) Beliebiger Aufruf ohne Verriegelung in allen OBs

### PG-Verbindungen und Diagnose

Funktion	Eigenschaft
Maximale Anzahl PG/OP-Verbindungen	32 (je 1 Verbindung ist für PG und OP reserviert)
Diagnose	unterstützt NCM-Diagnose über Ethernet
Suche im Netzwerk	unterstützt Siemens SIMATIC Manager Suche
10/100MBit	Umschaltung erfolgt automatisch

## Schnelleinstieg

### Übersicht

Bei der Erstinbetriebnahme einer CPU 21x-2BT10 besitzt der CP der CPU 21xNET keine IP-Adresse. Die Zuweisung erfolgt hier direkt über die Hardware-Konfiguration im Siemens SIMATIC Manager. Die Projektierung einer CPU 21xNET sollte nach folgender Vorgehensweise erfolgen:

- **Montage und Inbetriebnahme**
- **Hardware-Konfiguration** (Einbindung CP in CPU)
- **CP-Projektierung** über NetPro (Verbindung zum Ethernet).
- **SPS-Programmierung** über Anwender-Programm (Verbindung zur SPS).
- **Transfer des Gesamtprojekts in die CPU**

### Hinweis!

Um kompatibel mit dem Siemens SIMATIC Manager zu sein, sind die CPU 21x von VIPA als

**CPU 315-2DP (6ES7 315-2AF03-0AB0) V1.2**

zu projektieren!

Der CP-Teil der CPU 21xNET ist virtuell als CP343-1 (343-1EX11) von Siemens auf Steckplatz 4 zu projektieren.

Damit die System 200V-Module gezielt angesprochen werden können, sind diese im Hardware-Konfigurator von Siemens in Form eines virtuellen PROFIBUS-Systems zu projektieren. Hierbei können Sie durch Einbindung der GSD-Datei VIPA\_21X.GSD auf den Funktionsumfang der System 200V Module zurückgreifen.

### Montage und Inbetriebnahme

- Bauen Sie Ihr System 200V mit der CPU 21xNET auf.
- Verdrahten Sie das System, indem Sie die Leitungen für Spannungsversorgung, Signale und Ethernet anschließen. Eine detaillierte Beschreibung zu diesem Thema finden Sie im Teil "Montage und Aufbaurichtlinien".
- Schalten Sie die Spannungsversorgung ein. → Nach kurzer Hochlaufzeit befindet sich der CP im Leerlauf. Bei der Erstinbetriebnahme bzw. nach dem Umrösten der CPU besitzt der CP keine IP-Adresse. Zur Kontrolle können Sie den CP jetzt über die MAC-Adresse erreichen. Die MAC-Adresse finden Sie auf einem Aufkleber seitlich am Modul.

**IP-Parameter zuweisen**

Für die Zuweisung der IP-Parameter wie IP-Adresse, Subnet-Maske usw. haben Sie folgende Möglichkeiten:

- Online mit dem Siemens SIMATIC Manager über "Ethernet-Adresse vergeben" (ab CP-Firmware 1.7.4)
- Über ein Minimalprojekt mit IP-Adresse und IP-Parameter, das über MMC bzw. MPI in die CPU übertragen wird. Nach dem Neustart der CPU und nach Umstellen der PG/PC-Schnittstelle auf "TCP/IP... RFC1006" können Sie nun online über den CP Ihre CPU projektieren.

## Adressierung mit "Ethernet-Adresse vergeben"

Bitte beachten Sie, dass diese Funktionalität ab der CP-Firmware-Version 1.7.4 unterstützt wird.

- Starten Sie den Siemens SIMATIC Manager.
- Stellen Sie über **Extras** > *PG/PC-Schnittstelle einstellen* auf "TCP/IP... RFC1006" ein.
- Öffnen Sie mit **Zielsystem** > *Ethernet-Adresse vergeben* das Dialogfenster zur "Taufe" einer Station.
- 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 einem Aufkleber seitlich an der CPU.
- Wählen Sie ggf. bei der Netzwerksuche aus der Liste die Baugruppe mit der Ihnen bekannten MAC-Adresse aus.
- Stellen Sie nun die IP-Konfiguration ein, indem Sie IP-Adresse, Subnet-Maske und den Netzübergang eintragen. Sie können aber auch über einen DHCP-Server eine IP-Adresse beziehen. Hierzu ist dem DHCP-Server je nach gewählter Option die MAC-Adresse, der Gerätenamen oder die hier eingebare Client-ID zu übermitteln. Die Client-ID ist eine Zeichenfolge aus maximal 63 Zeichen. Hierbei dürfen folgende Zeichen verwendet werden: Bindestrich "-", 0-9, a-z, A-Z
- Bestätigen Sie Ihre Eingabe mit der Schaltfläche [... zuweisen].

Direkt nach der Zuweisung ist der CP über die angegebenen IP-Parameter online erreichbar.

## Adressierung über Minimalprojekt

- Starten Sie den Siemens SIMATIC Manager mit einem neuen Projekt.
- Fügen Sie mit **Einfügen** > *Station* > *SIMATIC 300-Station* eine neue System 300 Station ein.
- Aktivieren Sie die Station "SIMATIC 300" und öffnen Sie den Hardware-Konfigurator indem Sie auf "Hardware" klicken.
- Projektieren Sie ein Rack (SIMATIC 300 \ Rack-300 \ Profilschiene).
- Projektieren Sie stellvertretend für Ihre CPU 21xNET die Siemens CPU 315-2DP mit der Best.-Nr. 6ES7 315-2AF03-0AB0 V1.2., zu finden unter SIMATIC 300 \ CPU 300 \ CPU 315-2 DP. Parametrieren Sie ggf. die CPU 315-2DP.
- Projektieren Sie stellvertretend für den CP den Siemens-CP **CP343-1 (343-1EX11)** auf Steckplatz 4, zu finden unter SIMATIC 300 \ CP 300 \ Industrial Ethernet \ CP 343-1.
- Geben Sie in den CP-Eigenschaften die gewünschte IP-Adresse und Subnet-Maske an. Hier endet das *Minimalprojekt*. Nach der Übertragung dieses *Minimalprojekts* in die CPU können Sie über die im Projekt angegebene IP-Adresse und Subnet-Maske auf den CP zugreifen.

## Hardware-Konfiguration

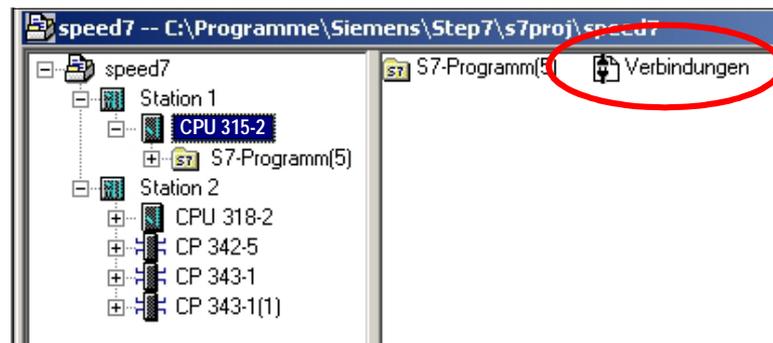
Die Hardware-Konfiguration erfolgt nach den unter "Adressierung über Minimalprojekt" aufgeführten Schritten, wobei die Projektierung noch nicht in die CPU übertragen wird.

Zur Projektierung der System 200V Module fahren Sie folgendermaßen fort:

- Legen Sie ein neues PROFIBUS-Subnetz mit der **PROFIBUS-Adresse >1** an.
- Hängen Sie das System "VIPA\_CPU21x" (VIPA\_21X.GSD erforderlich) mit der **PROFIBUS-Adresse 1** an, zu finden unter *PROFIBUS DP \ Weitere Feldgeräte \ IO \ VIPA\_System\_200V*.
- Platzieren Sie hier auf dem **1. Steckplatz** die CPU 21xNET.
- Binden Sie Ihre System 200V Module in der gesteckten Reihenfolge beginnend mit Steckplatz 1 ein.
- Sichern Sie Ihr Projekt.

## Verbindungen mit NetPro projektieren

Die Vernetzung zwischen den Stationen erfolgt mit der grafischen Benutzeroberfläche NetPro. Starten Sie NetPro, indem Sie in Ihrem Projekt auf ein Netz klicken bzw. im CPU-Verzeichnis auf Verbindungen.

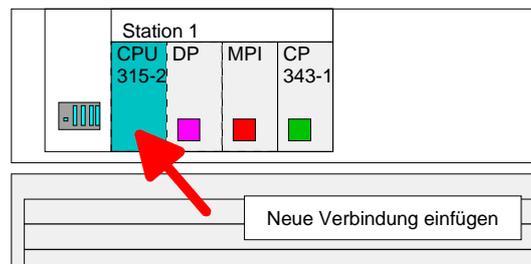


## Stationen vernetzen

Zur Projektierung von Verbindungen werden vernetzte Stationen vorausgesetzt. Zur Vernetzung von Stationen gehen Sie mit der Maus auf die farbliche Netzmarkierung des entsprechenden CP und ziehen Sie diese auf das zuzuordnende Netz. Die Verbindung wird grafisch über eine Linie dargestellt.

## Verbindungen projektieren

Klicken Sie zur Projektierung neuer Verbindungen auf die entsprechende CPU und wählen Sie über das Kontextmenü "Neue Verbindung einfügen".



Über das Dialogfenster können Sie die Parameter für eine Verbindung vorgeben. Die Parameter ID und LADDR sind für den Einsatz der AG\_SEND- bzw. AG\_RECV-Bausteine (FC5 bzw. FC6) erforderlich.

Verbindungen  
speichern und  
übersetzen

Speichern und übersetzen Sie Ihr Projekt und beenden Sie NetPro.  
Damit die CP-Projektierdaten in den Systemdaten abgelegt werden, müssen Sie in den der Hardware-Konfiguration des CP unter *Objekteigenschaften* im Bereich *Optionen* die Option "Projektierungsdaten in der CPU speichern" aktivierten (Standardeinstellung).

**SPS-Anwender-  
programm**

Zur Verarbeitung der Verbindungsaufträge auf SPS-Seite ist ein Anwenderprogramm in der CPU erforderlich. Hierbei kommen ausschließlich die VIPA Hantierungsbausteine AG\_SEND (FC5) und AG\_RECV (FC6) zum Einsatz. Die Bausteine sind Bestandteil der VIPA-Library, die sich als CD (SW830) im Lieferumfang befindet.

Den entsprechenden CP spezifizieren Sie über die Parameter *ID* und *LADDR* beim Aufruf der FC5 bzw. FC6.

**Projekt  
transferieren**

Es bestehen 3 Möglichkeiten für den Transfer Ihres Projekts in die CPU:

- Transfer über MPI
- Transfer über MMC bei Einsatz eines Schreib-/Lesegeräts
- Transfer über CP (Minimalprojekt erforderlich)

Näheres hierzu finden Sie weiter unten unter "Projekt transferieren".

Auf den Folgeseiten sind die in diesem Schnelleinstieg aufgeführten Schritte näher erläutert.

## Hardware-Konfiguration

### Übersicht

Zur Hardware-Konfiguration setzen Sie den Hardware-Konfigurator von Siemens ein. Hier geben Sie unter anderem die IP-Adresse des CPs an und projektieren die Hardware-Komponenten Ihrer SPS.

Für den Einsatz der System 200V Module ist die Einbindung der Module über die GSD-Datei `vipa_21x.gsd` von VIPA im Hardwarekatalog erforderlich.

### Voraussetzung

Bitte beachten Sie, dass zur Hardware-Konfiguration die folgenden Software-Pakete installiert sein müssen:

- Siemens SIMATIC Manager V. 5.1 oder höher und `vipa_21x.gsd`
- Siemens SIMATIC NET



### Hinweis!

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

### VIPA\_21x-GSD-Datei einbinden

- Kopieren Sie die mitgelieferte VIPA-GSD-Datei `vipa_21x.gsd` in Ihr Arbeitsverzeichnis.
- Starten Sie den Hardware-Konfigurator von Siemens.
- Schließen Sie alle Projekte.
- Gehen Sie auf **Extras** > *Neue GSD-Datei installieren*.
- Geben hier Ihr Arbeitsverzeichnis mit der **VIPA\_21X.GSD** an.

Die Module des System 200V von VIPA sind jetzt im Hardwarekatalog integriert und können projiziert werden.

### Hinweis

**Um kompatibel mit dem Siemens SIMATIC Manager zu sein, sind die CPU 21x von VIPA als**

**CPU 315-2DP (6ES7 315-2AF03-0AB0) V1.2**

**zu projektieren!**

**Der CP-Teil der CPU 21xNET ist virtuell als CP343-1 (343-1EX11) von Siemens auf Steckplatz 4 zu projektieren.**

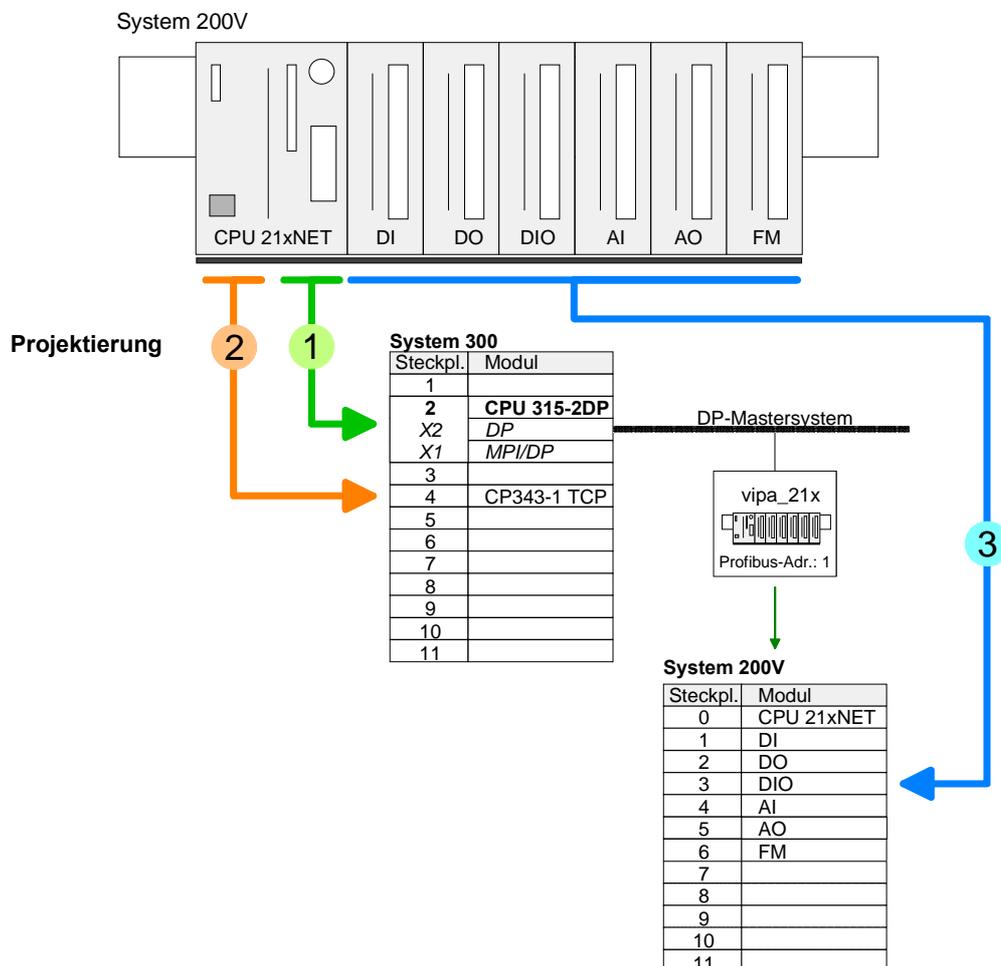
**Damit die System 200V-Module gezielt angesprochen werden können, sind diese im Hardware-Konfigurator von Siemens in Form eines virtuellen PROFIBUS-Systems zu projektieren. Hierbei können Sie durch Einbindung einer GSD-Datei von VIPA auf den Funktionsumfang der System 200V Module zurückgreifen.**

**Schritte der Projektierung**

Nachfolgend wird die Vorgehensweise der Projektierung im Hardware-Konfigurator von Siemens an einem abstrakten Beispiel gezeigt.

Projektierung

Hardwareaufbau

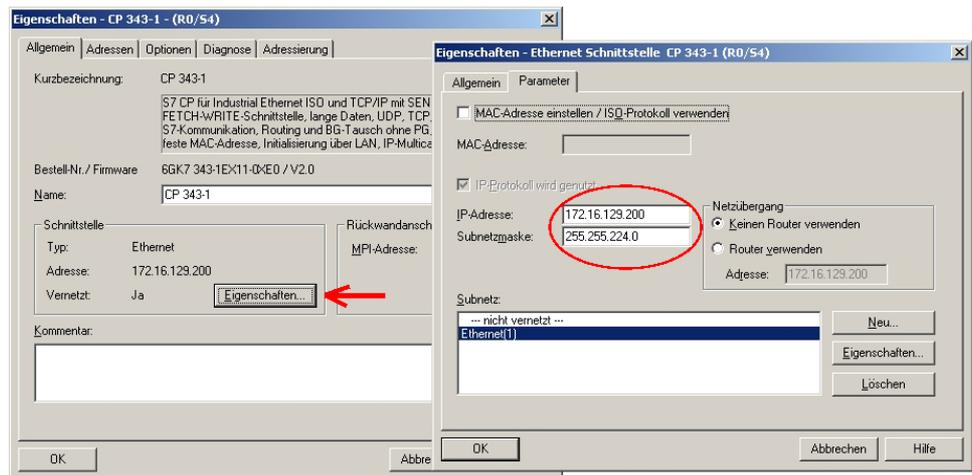


zu **1**

- Starten Sie den Siemens SIMATIC Manager und legen Sie ein neues Projekt an.
- Fügen Sie mit **Einfügen** > *Station* > *SIMATIC 300-Station* eine neue System 300 Station ein.
- Aktivieren Sie die Station "SIMATIC 300" und öffnen Sie den Hardware-Konfigurator indem Sie auf "Hardware" klicken.
- Projektieren Sie ein Rack (SIMATIC 300 \ Rack-300 \ Profilschiene).
- Alle CPU 21x von VIPA sind als Siemens CPU 315-2DP zu projektieren. Sie finden diese im Hardwarekatalog unter: SIMATIC 300 \ CPU 300 \ CPU 315-2 DP \ 6ES7 315-2AF03-0AB0 V1.2

zu 2

- Da sich der CP-Teil der CPU 21xNET von VIPA in der Projektierung gleich verhält wie der CP343-1 von Siemens, projektieren Sie auf Steckplatz 4 einen virtuellen CP343-1 (SIMATIC 300 \ CP 300 \ Industrial Ethernet \ CP 343-1 \ 6GK7 343-1EX11 0XE0).
- Öffnen Sie durch Doppelklick auf den CP 343-1 das "Eigenschaften"-Fenster und geben Sie unter Eigenschaften die IP-Adresse und Subnet-Maske an und wählen Sie das gewünschte Subnetz aus.



zu 3

- Parametrieren Sie die CPU 315-2DP nach Bedarf und legen Sie ein neues PROFIBUS-Subnetz mit einer **PROFIBUS-Adresse >1** an.
- Hängen Sie an das Subnetz das System "VIPA\_CPU21x". Sie finden dies im Hardware-Katalog unter *PROFIBUS DP \ Weitere Feldgeräte \ IO \ VIPA\_System\_200V*. Geben Sie diesem Slave die **PROFIBUS-Adresse 1**.
- Platzieren Sie in Ihrem Konfigurator auf dem **1. Steckplatz** die CPU 21xNET, die Sie einsetzen, indem Sie diese dem Hardware-Katalog entnehmen.
- Binden Sie Ihre System 200V Module in der gesteckten Reihenfolge beginnend mit Steckplatz 1 ein.
- Sichern und übersetzen Sie Ihr Projekt.

## Kommunikationsverbindungen projektieren

### Übersicht

Die Projektierung von Verbindungen, d.h. die "Vernetzung" zwischen den Stationen erfolgt in NetPro von Siemens. NetPro ist eine grafische Benutzeroberfläche zur Vernetzung von Stationen.

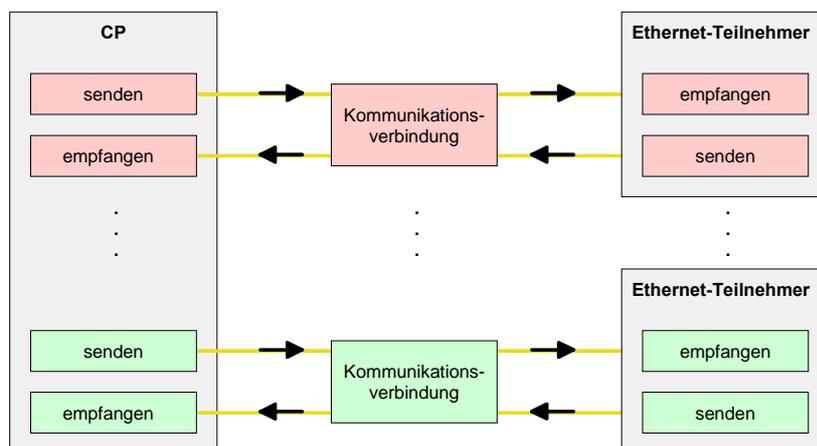
Eine Kommunikationsverbindung ermöglicht die programmgesteuerte Kommunikation zwischen zwei Teilnehmern am Industrial Ethernet. Die Kommunikationspartner können hierbei im selben Projekt oder - bei Multiprojekten - in den zugehörigen Teilprojekten verteilt angeordnet sein.

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.

### Eigenschaften

Folgende Eigenschaften zeichnen eine Kommunikationsverbindung aus:

- 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 Siemens V. 5.1 oder höher und SIMATIC NET sind installiert
- Der CP wurde bei der Hardware-Konfiguration projiziert, in die Hardware-Konfiguration eingetragen und mit dem Ethernet-Subnetz vernetzt.
- Der CP besitzt als Busteilnehmer eine IP-Adresse.



### Hinweis!

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 eingeben.

### 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 von Siemens.

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.

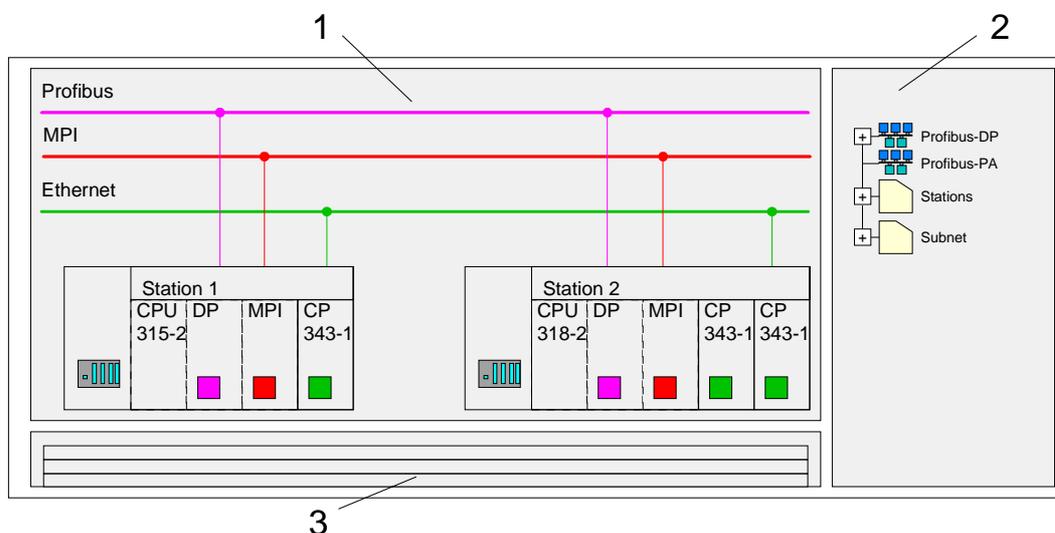
#### 2 Netzobjekte

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

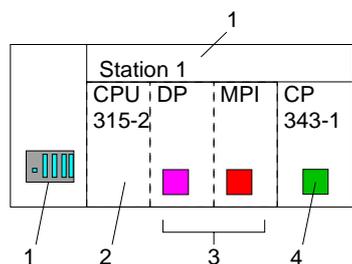
In der Verbindungstabelle sind alle Verbindungen tabellarisch aufgelistet. Diese Liste wird nur eingeblendet, wenn Sie die CPU einer verbindungsfähigen Baugruppe angewählt haben.

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. Da die 21xNET-CPU's als CPU 315-2DP projektiert werden, wird bei den internen Komponenten kein CP angezeigt.

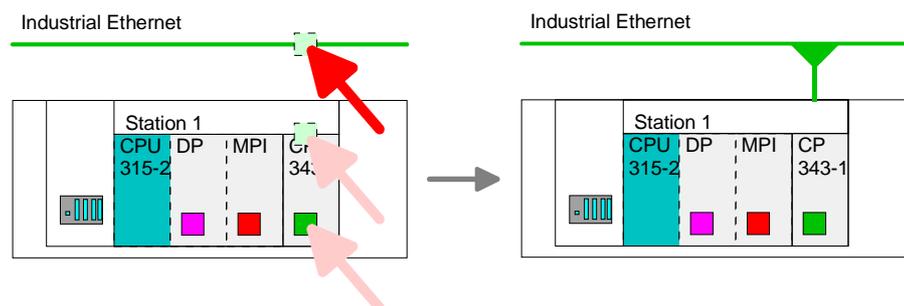
Aus diesem Grund ist der CP, der sich in der 21xNET-CPU befindetet als externer CP auf Steckplatz 4 zu projektieren. Der CP wird dann auch in NetPro als externer CP in der Station eingeleudet.

4 *CP*

In der Hardware-Konfiguration ist der CP als Siemens CP 343-1 zu projektieren.

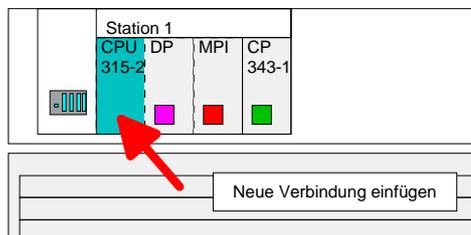
**Stationen vernetzen**

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 CP und ziehen Sie diese auf das zuzuordnende Netz. Daraufhin wird Ihr CP über eine Linie mit dem gewünschten Netz verbunden.



## Verbindungen projektieren

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



Es öffnet sich ein Dialogfenster in dem Sie den Verbindungspartner auswählen und den Typ der Verbindung einstellen können.

Markieren Sie die Partnerstation, zu der Sie eine Verbindung aufbauen möchten.

Wählen Sie unter "Typ" den Verbindungstyp aus, den Sie verwenden möchten. Folgende Verbindungstypen werden zur Zeit vom CP unterstützt:

ISO-on-TCP (SEND-RECEIVE, FETCH-WRITE PASSIV)

TCP (SEND-RECEIVE, FETCH-WRITE PASSIV)

UDP (SEND-RECEIVE)

## Allgemein

ID

LADDR

Sofern aktiviert öffnet sich ein Eigenschaften-Dialog der entsprechenden Verbindung. Dieses Dialogfenster ist das Bindeglied zu Ihrem SPS-Programm. Hier können Sie die *Lokale ID* einstellen und die *LADDR* ermitteln.

Beides sind Parameter, die in Ihrem SPS-Programm bei Verwendung der FC5 und FC6 (AG\_SEND, AG\_RECEIVE) anzugeben sind. Bitte hier immer die VIPA FCs verwenden, die als Bibliothek auf der SW830 mitgeliefert werden.



### Hinweis!

Bitte beachten Sie, dass den Verbindungen der SEND/RECEIVE-Schnittstelle eine CP-abhängige ID zugewiesen wird. Daher kann es bei Änderungen im Projekt zu Anpassungen der ID kommen. In diesem Fall ist im Anwenderprogramm die Schnittstellenversorgung von AG\_SEND bzw. AG\_RECV ebenfalls anzupassen.

Wird ein CP durch einen anderen ersetzt, muss dieser mindestens die gleichen Dienste bereitstellen und mindestens den gleichen Versionsstand haben. Nur so ist gewährleistet, dass die über den CP projektierten Verbindungen konsistent erhalten bleiben und genutzt werden können.

## Wegewahl

Mit der *Wegewahl* können Sie den lokalen und Partner-CP anwählen, über den die Verbindung laufen soll.

**Adressen**

Im Register Adressen werden die relevanten lokalen und fernen Adressinformationen als Vorschlagswerte angezeigt. Je nach Kommunikationsart können Sie Adressinformationen unspezifiziert lassen.

Die nachfolgende Tabelle zeigt die Kombinationsmöglichkeiten mit den verschiedenen Betriebsarten:

Verbindungspartner	Verbindungstyp	Verbindungsaufbau	Verbindung	Betriebsart
spezifiziert in NetPro (im aktuellen Projekt)	TCP / ISO-on-TCP	aktiv/passiv	spezifiziert	SEND/RECEIVE
	UDP	-		
unspezifiziert in NetPro (im aktuellen Projekt)	TCP / ISO-on-TCP	aktiv	spezifiziert	SEND/RECEIVE
		passiv	teilspezifiziert (Port)	SEND/RECEIVE FETCH PASSIV WRITE PASSIV
	UDP	-	unspezifiziert	SEND/RECEIVE
unspezifiziert in NetPro (in "unbekanntem Projekt")	TCP / ISO-on-TCP	aktiv	unspezifiziert (Verbindungsname)	SEND/RECEIVE
		passiv	unspezifiziert (Verbindungsname)	SEND/RECEIVE FETCH PASSIV WRITE PASSIV
	UDP	-	unspezifiziert (Verbindungsname)	SEND/RECEIVE
Alle Broadcast-Teilnehmer	UDP	-	spezifiziert (Port, Broadcast-Adr.)	SEND
Alle Multicast-Teilnehmer	UDP	-	spezifiziert (Port, Multicast-Gruppe)	SEND/RECEIVE

**Adressparameter**

Eine Verbindung wird durch den *lokalen* und *fernen* Verbindungsendpunkt spezifiziert. Mit Ausnahme von Broadcast- und Multicast-Verbindungen müssen bei der Projektierung von Verbindungen Ports/TSAPs kreuzweise übereinstimmen. Bei Broadcast- bzw. Multicast-Verbindungen müssen Sender und Empfänger den gleichen Port verwenden. Abhängig vom Protokoll definieren folgende Parameter einen Verbindungsendpunkt:

IP-Adresse Station A		IP-Adresse Station B
ferner TSAP	→	ISO-on-TCP-Verbindung
lokaler TSAP	←	lokaler TSAP
		← ferner TSAP

IP-Adresse Station A		IP-Adresse Station B
ferner Port	→	TCP-Verbindung
lokaler Port	←	lokaler Port
		← ferner Port

IP-Adresse Station A		IP-Adresse Station B
ferner Port	→	UDP-Verbindung
lokaler Port	←	lokaler Port
		← ferner Port

**TSAP**

ISO-on-TCP unterstützt TSAP-Längen (**T**ransport **S**ervice **A**ccess **S** Point) von 1...16Byte. Sie können den TSAP im ASCII- oder im hexadezimalen Format eingeben. Die Längenberechnung erfolgt automatisch.

**Port**

Ports bzw. Port-Adressen definieren den Zugangspunkt zum Anwenderprogramm innerhalb der Station/CPU. Diese müssen eindeutig sein. Eine Port-Adresse sollte im Bereich 2000...65535 liegen. Ferne und lokale Ports dürfen bei nur 1 Verbindung identisch sein.

**Verbindungen  
speichern und  
übersetzen**

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

Damit die CP-Projektierdaten in den Systemdaten abgelegt werden, müssen Sie in den der Hardware-Konfiguration des CP unter *Objekteigenschaften* im Bereich *Optionen* die Option "Projektierungsdaten in der CPU speichern" aktivierten (Standardeinstellung).

---

**Broadcast-/  
Multicast-  
Verbindungen**

Der Begriff "Verbindung" wird auch bei UDP verwendet, obwohl im Betrieb der Stationen kein expliziter Verbindungsaufbau zwischen den Kommunikationspartnern erfolgt.

Bei der Projektierung werden aber wie z.B. bei TCP die Kommunikationspartner einander zugeordnet und somit auch logisch verbunden.

Bei der Auswahl des Verbindungspartners haben Sie ausschließlich bei UDP noch folgende Optionen:

- Alle Broadcast-Teilnehmer
- Alle Muticast-Teilnehmer

**Broadcast-  
Teilnehmer**

Indem Sie als Verbindungspartner *Alle Broadcast-Teilnehmer* anwählen, bestimmen Sie, dass UDP-Telegramme an alle erreichbaren Broadcast-Teilnehmer zu senden sind. Bitte beachten Sie, dass der CP ausschließlich Broadcast-Telegramme senden kann. Empfang von Nutzdaten über Broadcast ist nicht möglich. Standardmäßig werden Broadcasts, die ausschließlich der Ethernet-Kommunikation dienen, wie z.B. ARP-Requests (Suche MAC <> IP-Adresse), empfangen und entsprechend bearbeitet.

Zur Identifikation der Broadcast-Teilnehmer im Netz ist bei der Projektierung einer Broadcast-Verbindung eine gültige Broadcast-Adresse als Partner-IP vorzugeben. Zusätzlich zur Broadcast-Adresse müssen Sie für Sender und Empfänger einen gemeinsamen Port angeben.

**Multicast-  
Teilnehmer**

Durch Anwahl von *Alle Multicast-Teilnehmer* bestimmen Sie, dass UDP-Telegramme an Teilnehmern einer Multicast-Gruppe zusenden bzw. von diesen zu empfangen sind. Im Gegensatz zu Broadcast ist hier der Empfang möglich.

Zur Identifikation der Multicast-Teilnehmer im Netz ist bei der Projektierung einer Multicast-Verbindung eine gültige Multicast-Gruppen-Adresse als Partner-IP vorzugeben. Zusätzlich zu dieser Adresse müssen Sie für Sender und Empfänger einen gemeinsamen Port angeben.

Die maximale Anzahl der Multicast-Kreise, die vom Ethernet CP unterstützt werden, ist identisch mit der maximalen Anzahl an Verbindungen.

## SEND/RECEIVE im SPS-Anwenderprogramm

### Übersicht

Für die Verarbeitung der Verbindungsaufträge auf SPS-Seite ist ein SPS-Anwenderprogramm in der CPU erforderlich. Hierbei kommen ausschließlich die VIPA Hantierungsbausteine AG\_SEND (FC5) und AG\_RECV (FC6) zum Einsatz. Beispielsweise durch Einbindung dieser Bausteine in den Zyklus-Baustein OB1 können Sie zyklisch Daten senden und empfangen. Die beiden FCs sind Bestandteil der VIPA-Library, die sich als CD (SW830) im Lieferumfang der CPU befindet.



### Hinweis!

Bitte beachten Sie, dass Sie in Ihrem Anwenderprogramm für die Kommunikation mit VIPA-CPs ausschließlich die SEND/RECV-FCs von VIPA einsetzen dürfen. Bei Wechsel zu VIPA-CPs in einem schon bestehenden Projekt können die bestehenden AG\_SEND/AG\_LSEND bzw. AG\_RECV/AG\_LRECV durch AG\_SEND bzw. AG\_RECV von VIPA ohne Anpassung ersetzt werden. Da sich der CP automatisch an die Länge der zu übertragenden Daten anpasst ist die L-Variante von SEND bzw. RECV bei VIPA nicht erforderlich.

### Kommunikationsbausteine

Für die Kommunikation zwischen CPU und CP stehen Ihnen folgende FCs zur Verfügung:

#### AG\_SEND (FC5)

Dieser Baustein übergibt die Nutzdaten aus dem über *SEND* angegebenen Datenbereich an den über *ID* und *LADDR* spezifizierten CP. Als Datenbereich können Sie einen PA-, Merker- oder Datenbaustein-Bereich angeben. Wurde der Datenbereich fehlerfrei übertragen, so wird "Auftrag fertig ohne Fehler" zurückgemeldet.

#### AG\_RECV (FC6)

Der Baustein übernimmt vom CP die Nutzdaten und legt sie in dem über *RECV* definierten Datenbereich ab. Als Datenbereich können Sie einen PE-, Merker- oder Datenbaustein-Bereich angeben. Wurde der Datenbereich fehlerfrei übernommen, so wird "Auftrag fertig ohne Fehler" zurückgemeldet.

### Statusanzeigen

Der CP bearbeitet Sende- und Empfangsaufträge unabhängig vom CPU-Zyklus und benötigt hierzu eine Übertragungszeit. Die Schnittstelle mit den FC-Bausteinen zum Anwenderprogramm wird hierbei über Quittungen synchronisiert.

Für die Statusauswertung liefern die Kommunikationsbausteine Parameter zurück, die Sie in Ihrem Anwenderprogramm direkt auswerten können.

Diese Statusanzeigen werden bei jedem Baustein-Aufruf aktualisiert.

### Einsatz unter hoher Kommunikationslast

Verwenden Sie keine zyklischen Aufrufe der Kommunikationsbausteine im OB1. Dies führt zu einer ständigen Kommunikation zwischen CPU und CP. Programmieren Sie stattdessen Ihre Kommunikationsbausteine in einem Zeit-OB, deren Zykluszeit größer ist als die des OB1 bzw. ereignisgesteuert.

Aufruf FC schneller als CP-Übertragungszeit

Wird ein Baustein im Anwenderprogramm erneut aufgerufen, bevor die Daten vollständig gesendet oder empfangen wurden, wird an der Schnittstelle der FC-Bausteine wie folgt verfahren:

**AG\_SEND**

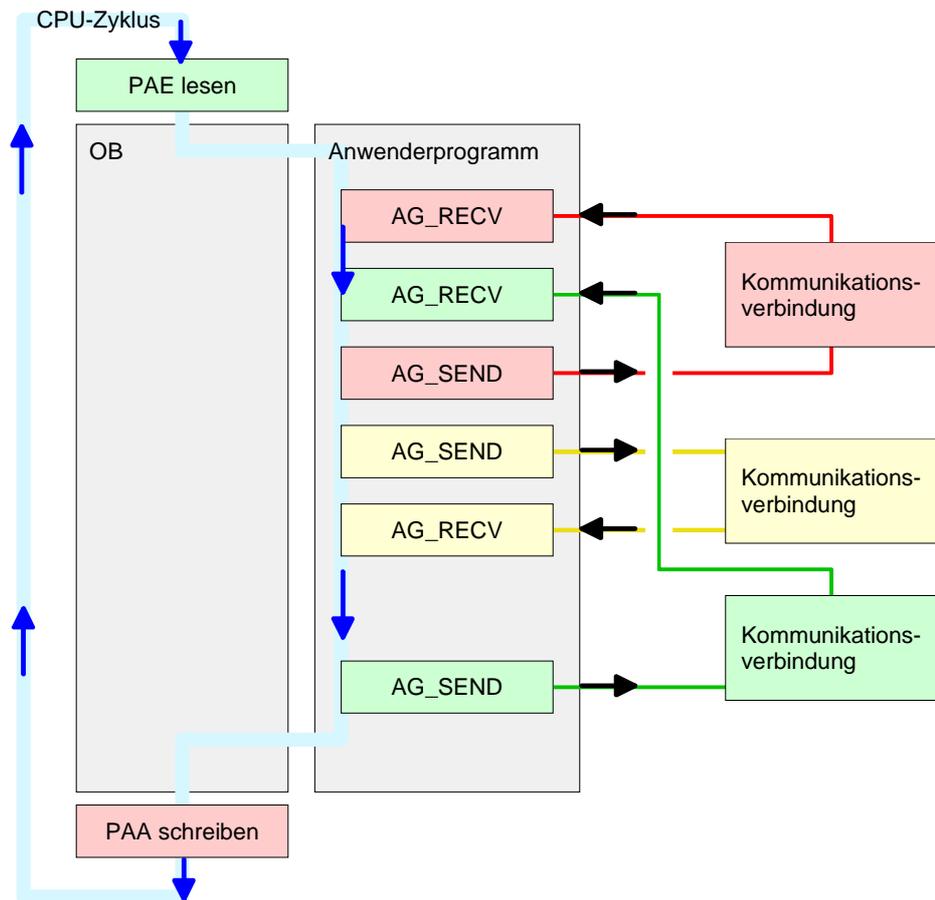
Es wird kein Auftrag entgegen genommen, bis die Datenübertragung über die Verbindung vom Partner quittiert wurde. Solange erhalten Sie die Meldung "Auftrag läuft", bis der CP den nächsten Auftrag für die gleiche Verbindung übernehmen kann.

**AG\_RECV**

Der Auftrag wird mit der Meldung "Es liegen noch keine Daten vor" quittiert, solange der CP die Empfangsdaten noch nicht vollständig empfangen hat.

**AG\_SEND, AG\_RECV im Anwenderprogramm**

Eine mögliche Ablaufsequenz für die FC-Bausteine zusammen mit den Organisations- und Programmbausteinen im CPU-Zyklus ist nachfolgend dargestellt:



Die FC-Bausteine mit zugehöriger Kommunikationsverbindung sind farblich zusammengefasst. Hier können Sie auch erkennen, dass Ihr Anwenderprogramm aus beliebig vielen Bausteinen bestehen kann. Somit können Sie ereignis- bzw. programmgesteuert an beliebiger Stelle im CPU-Zyklus mit AG\_SEND Daten senden bzw. mit AG\_RECV Daten empfangen.

Sie können die Bausteine für **eine** Kommunikationsverbindung auch mehrmals in einem Zyklus aufrufen.

**AG\_SEND (FC5)** Mit AG\_SEND werden die zu sendenden Daten an den CP übertragen.

Parameter

Parameter	Deklaration	Datentyp	Beschreibung
ACT	Input	BOOL	Aktivierung des Senders 0: Aktualisiert die DONE, ERROR und STATUS 1: Der unter SEND mit der Länge LEN abgelegte Datenbereich wird gesendet
ID	Input	INT	Verbindungsnummer 1 ... 16 (identisch mit ID aus NetPro)
LADDR	Input	WORD	Logische Basisadresse des CPs (identisch mit LADDR aus NetPro)
SEND	Input	ANY	Datenbereich
LEN	Input	INT	Anzahl der Bytes, die aus dem Datenbereich zu übertragen sind
DONE	Output	BOOL	Zustandsparameter für den Auftrag 0: Auftrag läuft 1: Auftrag fertig ohne Fehler
ERROR	Output	BOOL	Fehleranzeige 0: Auftrag läuft (bei DONE = 0) 0: Auftrag fertig ohne Fehler (bei DONE = 1) 1: Auftrag fertig mit Fehler
STATUS	Output	WORD	Statusanzeige, die in Verbindung mit DONE und ERROR zurückgeliefert wird. Näheres hierzu finden Sie in der nachfolgenden Tabelle.

**AG\_RECV (FC6)** Mit AG\_RECV werden die Daten, die der CP empfangen hat, in die CPU übertragen.

Parameter

Parameter	Deklaration	Datentyp	Beschreibung
ID	Input	INT	Verbindungsnummer 1 ... 16 (identisch mit ID aus NetPro)
LADDR	Input	WORD	Logische Basisadresse des CPs (identisch mit LADDR aus NetPro)
RCV	Input	ANY	Datenbereich für die empfangenen Daten
NDR	Output	BOOL	Zustandsparameter für den Auftrag 0: Auftrag läuft 1: Auftrag fertig Daten wurden ohne Fehler übernommen
ERROR	Output	BOOL	Fehleranzeige 0: Auftrag läuft (bei NDR = 0) 0: Auftrag fertig ohne Fehler (NDR = 1) 1: Auftrag fertig mit Fehler
STATUS	Output	WORD	Statusanzeige, die in Verbindung mit NDR und ERROR zurückgeliefert wird. Näheres hierzu finden Sie in der nachfolgenden Tabelle.
LEN	Output	INT	Anzahl der Bytes, die empfangen wurden

**DONE, ERROR,  
STATUS**

In der nachfolgenden Tabelle sind alle Meldungen aufgeführt, die der CP nach einem SEND-Auftrag bzw. RECV-Auftrag zurückliefern kann.

Ein "-" bedeutet, dass diese Meldung für den entsprechenden SEND- bzw. RECV-Auftrag nicht existiert.

DONE (SEND)	NDR (RECV)	ERROR	STATUS	Bedeutung
1	-	0	0000h	Auftrag fertig ohne Fehler
-	1	0	0000h	Neue Daten wurden ohne Fehler übernommen
0	-	0	0000h	Kein Auftrag in Bearbeitung
-	0	0	8180h	Es liegen noch keine Daten vor
0	0	0	8181h	Auftrag läuft
0	0	1	8183h	Für diesen Auftrag gibt es keine CP-Projektierung
0	-	1	8184h	Es ist ein Systemfehler aufgetreten
-	0	1	8184h	Es ist ein Systemfehler aufgetreten (Quelldatenbereich fehlerhaft)
0	-	1	8185h	Parameter LEN größer als Quell-Bereich SEND
	0	1	8185h	Ziel-Puffer (RECV) ist zu klein
0	0	1	8186h	Parameter ID ungültig (nicht im Bereich 1 ...16)
0	-	1	8302h	keine Empfangsressourcen bei Ziel-Station, Empfänger-Station kann empfangene Daten nicht schnell genug verarbeiten bzw. hat keine Empfangsressourcen bereitgestellt.
0	-	1	8304h	Die Verbindung ist nicht aufgebaut. Der Sendeauftrag sollte erst nach einer Wartezeit >100 ms erneut abgesetzt werden.
-	0	1	8304h	Die Verbindung ist nicht aufgebaut. Der Empfangsauftrag sollte erst nach einer Wartezeit > 100ms erneut abgesetzt werden.
0	-	1	8311h	Zielstation ist unter der angegebenen Ethernet-Adresse nicht erreichbar.
0	-	1	8312h	Ethernet-Fehler im CP
0		1	8F22h	Quell-Bereich ungültig, wenn beispielsweise Bereich im DB nicht vorhanden Parameter LEN < 0
-	0	1	8F23h	Quell-Bereich ungültig, wenn beispielsweise Bereich im DB nicht vorhanden Parameter LEN < 0
0	-	1	8F24h	Bereichsfehler beim Lesen eines Parameters.
-	0	1	8F25h	Bereichsfehler beim Schreiben eines Parameters.
0	-	1	8F28h	Ausrichtungsfehler beim Lesen eines Parameters.
-	0	1	8F29h	Ausrichtungsfehler beim Schreiben eines Parameters.
-	0	1	8F30h	Parameter liegt im schreibgeschützten 1. akt. Datenbaustein
-	0	1	8F31h	Parameter liegt im schreibgeschützten 2. akt. Datenbaustein
0	0	1	8F32h	Parameter enthält zu große DB-Nummer.
0	0	1	8F33h	DB-Nummer Fehler
0	0	1	8F3Ah	Bereich nicht geladen (DB)

Fortsetzung ...

... Fortsetzung *DONE, ERROR, STATUS*

DONE (SEND)	NDR (RECV)	ERROR	STATUS	Bedeutung
0	-	1	8F42h	Quittungsverzug beim Lesen eines Parameters aus dem Peripheriebereich.
-	0	1	8F43h	Quittungsverzug beim Schreiben eines Parameters in den Peripheriebereich.
0	-	1	8F44h	Adresse des zu lesenden Parameters in der Zugriffsspur gesperrt
-	0	1	8F45h	Adresse des zu schreibenden Parameters in der Zugriffsspur gesperrt
0	0	1	8F7Fh	Interner Fehler z.B. unzulässige ANY-Referenz z.B. Parameter LEN = 0 .
0	0	1	8090h	Baugruppe mit dieser Baugruppen-Anfangsadresse nicht vorhanden oder CPU in STOP.
0	0	1	8091h	Baugruppen-Anfangsadresse nicht auf Doppel-Wort-Raster.
0	0	1	8092h	In ANY-Referenz ist eine Typangabe ungleich BYTE angegeben.
-	0	1	80A0h	Negative Quittung beim Lesen von Baugruppe
0	0	1	80A4h	reserviert
0	0	1	80B0h	Baugruppe kennt den Datensatz nicht.
0	0	1	80B1h	Die Längenangabe (im Parameter LEN) ist falsch.
0	0	1	80B2h	reserviert
0	0	1	80C0h	Datensatz kann nicht gelesen werden.
0	0	1	80C1h	Der angegebene Datensatz ist gerade in Bearbeitung.
0	0	1	80C2h	Es liegt ein Auftragsstau vor.
0	0	1	80C3h	Die Betriebsmittel (Speicher) der CPU sind temporär belegt.
0	0	1	80C4h	Kommunikationsfehler (tritt temporär auf; daher ist eine Wiederholung im Anwenderprogramm sinnvoll.)
0	0	1	80D2h	Baugruppen-Anfangsadresse ist falsch.

Status-Parameter bei Neuanlauf

Bei einem Neuanlauf des CP werden die Ausgabe-Parameter wie folgt zurückgesetzt:

- DONE = 0
- NDR = 0
- ERROR = 8180h (bei AG\_RECV)  
ERROR = 8181h (bei AG\_SEND)

## Projekt transferieren

### Übersicht

Es bestehen 3 Möglichkeiten für den Transfer Ihres Projekts in die CPU:

- Transfer über MPI
- Transfer über MMC bei Einsatz eines Schreib-/Lesegeräts
- Transfer über CP



### Hinweis!

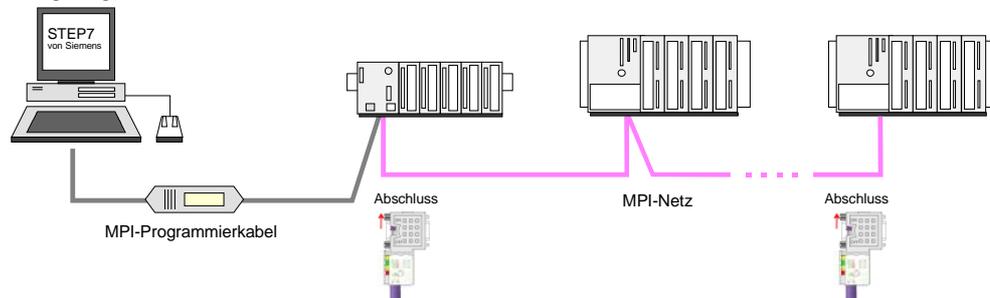
Nachfolgend sind die Transfermethoden kurz aufgeführt. Eine weitere Beschreibung dieser Methoden finden Sie unter "Einsatz CPU 21x" im Kapitel "Projekt transferieren".

### Transfer über MPI

#### Transfer mit MPI-Programmierkabel (MPI-Kommunikation)

Die MPI-Programmierkabel von VIPA bieten einen busfähigen RS485-Anschluss für die MP<sup>2</sup>I-Buchse der CPU und einen RS232 bzw. USB-Anschluss für den PC. Standardmäßig besitzt Ihre CPU die MPI-Adresse 2.

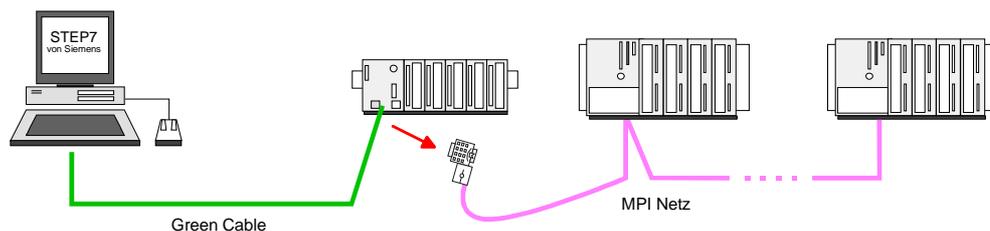
- Wählen Sie im Menü **Extras** > *PG/PC-Schnittstelle einstellen*.
- Stellen Sie Ihre *MPI-Übertragungsparameter* und *-Adresse* ein.
- Geben unter *Lokaler Anschluss* den PC-COM-Port und die Übertragungsrate 38400Baud ein.



#### Transfer mit Green Cable (Serielle Kommunikation)

Durch ausschließlich direktes Stecken des Green Cable auf einer MP<sup>2</sup>I-Buchse können Sie eine serielle Verbindung zwischen PC und CPU herstellen.

- Geben Sie unter *Lokaler Anschluss* den PC-COM-Port und die Übertragungsrate 38400Baud ein. Die Einstellungen im Register *MPI* werden bei Green Cable Einsatz ignoriert.



---

**Transfer über MMC**

Die MMC (**M**emory **C**ard) dient als externes Speichermedium und besitzt ein FAT16 Filesystem.

Ihre Projektierung muss sich im Root-Verzeichnis befinden und folgenden Dateinamen haben: **S7PROG.WLD**.

Mit URLÖSCHEN wird automatisch von der MMC gelesen (falls gesteckt).

---

**Transfer über Ethernet**

Für den Zugriff über Ethernet ist es erforderlich, dass sich in der CPU eine Hardware-Projektierung (Minimalprojekt) befindet, in der IP-Adresse und Subnet-Maske über einen virtuellen CP angegeben sind.

Dieses Projekt ist per MMC oder MPI in die CPU zu übertragen. Nach der Übertragung können Sie über die "Zielsystemfunktionen" über Ethernet auf den CP zugreifen.

**Vorgehensweise (Minimalprojekt)**

- Projektieren Sie im Hardware-Konfigurator eine CPU 315-2 mit der Best.-Nr. 6ES7 315-2AF03-0AB0 V1.2.
- Binden Sie auf Steckplatz 4 den Siemens CP 343-1 ein (343-1EX11).
- Geben Sie im Dialogfenster die gewünschte IP-Adresse und Subnet-Maske an und vernetzen Sie den CP mit "Ethernet".
- Übertragen Sie Ihr Projekt via MPI oder MMC in Ihre CPU.

**Zugriff auf CP**

Die nachfolgend aufgeführte Vorgehensweise setzt voraus, dass der CP online erreichbar ist, d.h. Sie haben ihm über eine Hardware-Konfiguration eine IP-Adresse und Subnet-Maske zugeteilt und befinden sich mit Ihrem Projektier-PC im gleichen IP-Nummernkreis.

Verbinden Sie Ihren Projektier-PC und Ihre CPU über die Twisted-pair-Buchse mit Ethernet. Stellen Sie im Siemens SIMATIC Manager unter **Extras > PG/PC Schnittstelle...** folgendes ein:

TCP/IP -> Netzwerkkarte...Protokoll RFC 1006

Nun können Sie beispielsweise über die Zielsystemfunktionen auf den CP zugreifen.

**CPU über CP projektieren**

- Wechseln Sie in Ihr Projekt im Hardware-Konfigurator und starten Sie die Übertragung mit **Zielsystem > Laden in Baugruppe**
- Wählen Sie die gewünschte Baugruppe aus und geben Sie als "Teilnehmeradresse" die IP-Adresse des CP an.  
Vor der Übertragung bekommen Sie eine Fehlermeldung, dass sich die "Online-" von der "Offline-" Baugruppe unterscheidet. Diese Meldung können Sie ignorieren und mit [OK] die Übertragung starten.

Nun können Sie über Ihr zuvor erstelltes Projekt auf den CP zugreifen und mit NetPro die gewünschten Verbindungen für den CP projektieren.

Sofern keine neue Hardware-Konfiguration in die CPU übertragen wird, wird der oben angegebene CP dauerhaft als Transferkanal im Projekt gespeichert.

## NCM-Diagnose - Hilfe zur Fehlersuche

### Checkliste zur Fehlersuche

Diese Seite soll Ihnen bei der Fehlersuche dienen. Die nachfolgende Checkliste soll Ihnen helfen, einige typische Problemstellungen und deren mögliche Ursachen zu erkennen:

Frage	Abhilfe bei "nein"
CPU im Run?	DC 24V-Spannungsversorgung überprüfen. RUN/STOP-Schalter in Stellung RUN bringen. SPS-Programm überprüfen und neu übertragen.
AG_SEND, AG_RECV im Anwenderprogramm?	Für den Datentransfer zwischen CP und CPU sind diese 2 Bausteine im Anwenderprogramm erforderlich. Auch bei einer passiven Verbindung sind beide Bausteine aufzurufen
Kann CP verbinden?	Ethernetleitung überprüfen (bei Punkt-zu-Punkt-Verbindung ist ein gekreuztes Ethernetkabel zu verwenden). IP-Adresse überprüfen.
Können Daten transferiert werden?	Port-Nr. für Lesen und Schreiben überprüfen. Die Quell- und Zielbereiche überprüfen. Prüfen, ob der richtige CP in der Wegewahl angewählt ist. Den mit dem ANY-Pointer angegebenen Empfangs- bzw. Sendepuffer vergrößern.
Wird der komplette Datenblock bei ISO-on-TCP gesendet?	Überprüfen Sie den LEN-Parameter bei AG_SEND. Den mit dem ANY-Pointer angegebenen Empfangs- bzw. Sendepuffer auf die erforderliche Größe einstellen.

### Siemens NCM S7-Diagnose

Der CP unterstützt das Siemens NCM-Diagnosetool. Das NCM-Diagnosetool ist Bestandteil des Siemens SIMATIC Managers. 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 CP den Diagnosepuffer auslesen
- Verbindungen diagnostizieren

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

Für den Aufruf des Diagnose-Tools haben Sie folgende 2 Möglichkeiten:

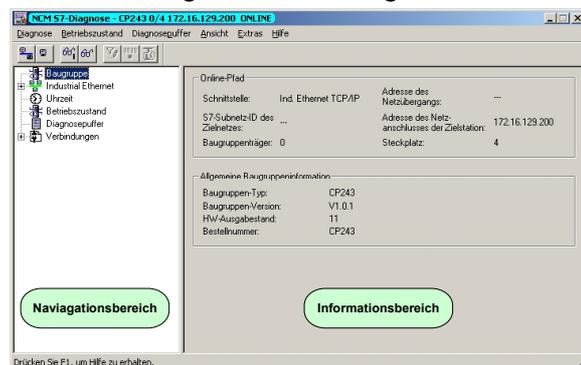
- Über *Windows-START-Menü > SIMATIC ... NCM S7 > Diagnose*
- Innerhalb der Projektierung bzw. Hardware-Konfiguration über das Register "Diagnose" im "Eigenschaften"-Dialog mit [Ausführen] die Diagnose aufrufen.

## Aufbau

Die Arbeitsumgebung des Diagnose-Tools hat folgenden Aufbau:

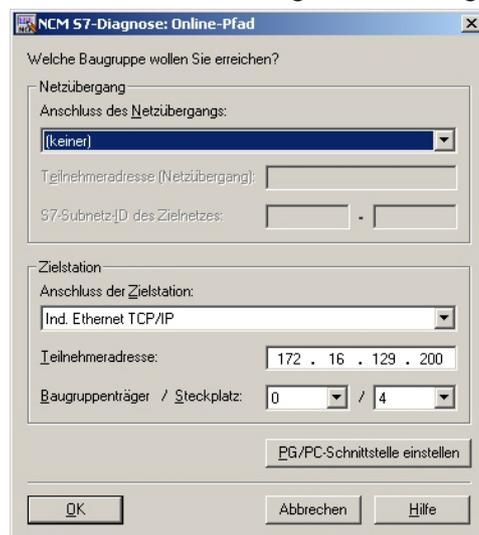
Im *Navigationsbereich* auf der linken Seite finden Sie die hierarchisch geordneten Diagnoseobjekte. Je nach CP-Typ und projektierte Verbindungen haben Sie eine angepasste Objektstruktur im Navigationsbereich.

Im *Inhaltsbereich* 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:**

Für das System 200V ist immer *Baugruppenträger 0* und *Steckplatz 0* auszuwählen.

Stellen Sie Ihre PG/PC-Schnittstelle auf TCP/IP...RFC1006 ein. Mit [OK] starten Sie die Online-Diagnose.

**Diagnosepuffer  
auslesen**

Der CP 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 CP-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.

**Hinweis!**

Überprüfen Sie immer anhand der Checkliste am Kapitelanfang die Voraussetzungen für eine funktionsfähige Kommunikation.

Für den gezielten Diagnoseeinsatz ist folgende Vorgehensweise zweckmäßig:

- Diagnose aufrufen.
- Mit  Dialog für Online-Verbindung öffnen, Verbindungsparameter eintragen und mit [OK] Online-Verbindung herstellen.
- Den CP identifizieren und über Baugruppenzustand den aktuellen Zustand des CPs ermitteln.
- Verbindungen überprüfen auf Besonderheiten wie:
  - Verbindungszustand
  - Empfangszustand
  - Sendezustand
- Über *Diagnosepuffer* den Diagnosepuffer des CP einsehen und entsprechend auswerten.
- Soweit erforderlich, Projektierung bzw. Programmierung ändern und Diagnose erneut starten.

## Kopplung mit Fremdsystemen

### Übersicht

Die bei TCP- bzw. ISO-on\_TCP unterstützte Betriebsart FETCH/WRITE können Sie prinzipiell für Zugriffe von Fremdgeräten auf den SPS-Systemspeicher verwenden. Damit Sie diesen Zugriff z.B. auch für PC-Anwendungen implementieren können, müssen Sie den Telegramm-Aufbau für die Aufträge kennen. Die spezifischen Header für Anforderungs- und Quittungstelegramme sind standardmäßig 16 Byte lang und werden auf den Folgeseiten beschrieben.

### ORG-Format

Das Organisationsformat ist die Kurzbeschreibung einer Datenquelle bzw. eines Datenziels in SPS-Umgebung. Die verwendbaren ORG-Formate sind in der nachfolgenden Tabelle aufgelistet.

Die ERW-Kennung ist bei der Adressierung von Datenbausteinen relevant. In diesem Fall wird hier die Datenbaustein-Nummer eingetragen. Die Anfangsadresse und Anzahl adressieren den Speicherbereich und sind im HIGH-/LOW- Format abgelegt (Motorola - Adressformat).

Beschreibung	Typ	Bereich
ORG-Kennung	BYTE	1..x
ERW-Kennung	BYTE	1..255
Anfangsadresse	HILOWORD	0..y
Länge	HILOWORD	1..z

In der nachfolgenden Tabelle sind die verwendbaren ORG-Formate aufgelistet. Die "Länge" darf nicht mit -1 (FFFFh) angegeben werden.

### ORG-Kennung 01h-04h

CPU-Bereich	DB	MB	EB	AB
ORG-Kennung	01h	02h	03h	04h
Beschreibung	Quell-/Zieldaten aus/in Datenbaustein im Hauptspeicher.	Quell-/Zieldaten aus/in Merkerbereich.	Quell-/Zieldaten aus/in Prozessabbild der Eingänge (PAE).	Quell-/Zieldaten aus/in Prozessabbild der Ausgänge (PAA).
ERW-Kennung (DBNR)	DB, aus dem die Quelldaten entnommen werden bzw. in den die Zieldaten transferiert werden.	irrelevant	irrelevant	irrelevant
Anfangsadresse Bedeutung	DBB-Nr., ab der die Daten entnommen bzw. eingeschrieben werden.	MB-Nr., ab der die Daten entnommen bzw. eingeschrieben werden.	EB-Nr., ab der die Daten entnommen bzw. eingeschrieben werden.	AB-Nr., ab der die Daten entnommen bzw. eingeschrieben werden.
Länge Bedeutung	Länge des Quell-/Zieldatenblocks in <u>Worten</u> .	Länge des Quell-/Zieldatenblocks in Bytes.	Länge des Quell-/Zieldatenblocks in Bytes.	Länge des Quell-/Zieldatenblocks in Bytes.



**Hinweis!**

Informationen zu den erlaubten Bereichen finden Sie in den Technischen Daten ihrer CPU.

*ORG-Kennung 05h-07h*

CPU-Bereich	PB	ZB	TB
ORG-Kennung	05h	06h	07h
Beschreibung	Quell-/Zieldaten aus/in Peripheriebaugruppen. Bei Quelldaten Eingabebaugruppen, bei Zieldaten Ausgabebaugruppen.	Quell-/Zieldaten aus/in Zählerzellen.	Quell-/Zieldaten aus/in Zeitenzellen.
ERW-Kennung (DBNR)	irrelevant	irrelevant	irrelevant
Anfangsadresse Bedeutung	PB-Nr., ab der die Daten entnommen bzw. eingeschrieben werden.	ZB-Nr., ab der die Daten entnommen bzw. eingeschrieben werden.	TB-Nr., ab der die Daten entnommen bzw. eingeschrieben werden.
Länge Bedeutung	Länge des Quell-/Zieldatenblocks in Bytes.	Länge des Quell-/Zieldatenblocks in Worten (Zählerzelle = 1 Wort).	Länge des Quell-/Zieldatenblocks in Worten (Zählerzelle = 1 Wort).

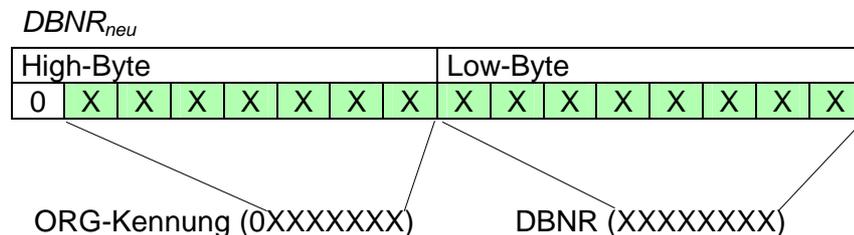
**Übertragen von Bausteinen mit Nummern >255**

*ORG-Kennung 81h-FFh*

Zur Übertragung von Datenbausteinen im Nummernbereich 256 ... 32768 können Sie die ORG-Kennung 81h-FFh verwenden.

Da die Angabe einer DB-Nr. >255 ein Wort als Länge erfordert, setzt sich  $DBNR_{neu}$  aus dem Inhalt von ORG-Kennung und DBNR zusammen.

$DBNR_{neu}$  wird als Wort auf folgende Weise generiert:



Ist das höchste Bit der ORG-Kennung gesetzt, so ergibt sich das Low-Byte von  $DBNR_{neu}$  aus der DBNR und das High-Byte von  $DBNR_{neu}$  aus der ORG-Kennung, wobei das höchste Bit der ORG-Kennung eliminiert wird.

Folgende Formel soll dies nochmals verdeutlichen:

$$DBNR_{neu} = 256 \times (ORGKennung \text{ AND } 7Fh) + DBNR$$

**Aufbau SPS-Header**

Bei FETCH und WRITE generiert der CP SPS-Header für Anforderungs- und Quittungstelegramme. Diese Header sind 16Byte lang und haben folgende Struktur:

**WRITE**

*Anforderungstelegramm Remote Station*

Systemkennung	= "S5"	(Wort)
Länge Header	= 10h	(Byte)
Kenn. OP-Code	= 01h	(Byte)
Länge OP-Code	= 03h	(Byte)
<b>OP-Code</b>	<b>= 03h</b>	(Byte)
ORG-Block	= 03h	(Byte)
Länge ORG-Block	= 08h	(Byte)
ORG-Kennung*		(Byte)
ERW-Kennung		(Byte)
Anfangsadresse		(Wort)
Länge		(Wort)
Leerblock	= FFh	(Byte)
Länge Leerblock	= 02h	(Byte)
Daten bis zu 64kByte (nur wenn Fehler-Nr.=0)		

*Quittungstelegramm CP*

Systemkennung	= "S5"	(Wort)
Länge Header	= 10h	(Byte)
Kenn. OP-Code	= 01h	(Byte)
Länge OP-Code	= 03h	(Byte)
<b>OP-Code</b>	<b>= 04h</b>	(Byte)
Quittungsblock	= 0Fh	(Byte)
Länge Q-Block	= 03h	(Byte)
Fehler-Nr.		(Byte)
Leerblock	= FFh	(Byte)
Länge Leerblock	= 07h	(Byte)
5 leere Bytes angehängt		

**FETCH**

*Anforderungstelegramm Remote Station*

Systemkennung	= "S5"	(Wort)
Länge Header	= 10h	(Byte)
Kenn. OP-Code	= 01h	(Byte)
Länge OP-Code	= 03h	(Byte)
<b>OP-Code</b>	<b>= 05h</b>	(Byte)
ORG-Block	= 03h	(Byte)
Länge ORG-Block	= 08h	(Byte)
ORG-Kennung*		(Byte)
ERW-Kennung		(Byte)
Anfangsadresse		(Wort)
Länge		(Wort)
Leerblock	= FFh	(Byte)
Länge Leerblock	= 02h	(Byte)

*Quittungstelegramm CP*

Systemkennung	= "S5"	(Wort)
Länge Header	= 10h	(Byte)
Kenn. OP-Code	= 01h	(Byte)
Länge OP-Code	= 03h	(Byte)
<b>OP-Code</b>	<b>= 06h</b>	(Byte)
Quittungsblock	= 0Fh	(Byte)
Länge Q-Block	= 03h	(Byte)
Fehler-Nr.		(Byte)
Leerblock	= FFh	(Byte)
Länge Leerblock	= 07h	(Byte)
5 leere Bytes angehängt		
Daten bis zu 64kByte (nur wenn Fehler-Nr.=0)		

\*) Nähere Angaben zum Datenbereich finden Sie unter "ORG-Format" weiter oben.



**Hinweis!**

Bitte beachten Sie, dass im Gegensatz zu Siemens-S5-Systemen hier bei der Daten-Baustein-Adressierung die Anfangsadresse als Byte-Nummer interpretiert wird.

**Meldungen von Fehler-Nr.**

Folgende Meldungen können über Fehler-Nr. zurückgeliefert werden:

Fehler-Nr	Meldung
00h	Kein Fehler aufgetreten
01h	Der angegebene Bereich kann nicht gelesen bzw. beschrieben werden.

## Beispiel zur Kommunikation CPU 21x-2BT10

### Übersicht

Dieses Kapitel soll in den Umgang mit dem Bussystem TCP/IP für das System 200V einführen. Ziel dieses Kapitels ist es, eine Kommunikation zwischen zwei VIPA CPUs 21xNET aufzubauen, die auf einfache Weise die Kontrolle der Kommunikationsvorgänge erlaubt.

### Voraussetzungen

Kenntnisse über die VIPA-CP-Hantierungsbausteine AG\_SEND und AG\_RECV sind erforderlich. Die CP-Hantierungsbausteine ermöglichen die Nutzung der Kommunikationsfunktionen durch Programme in den Automatisierungsgeräten.

Für die Durchführung des Beispiels sollten Sie mindestens die folgenden technischen Einrichtungen besitzen:

#### *Hardware*

- 2 CPUs 21x-2BT10 von VIPA
- 1 PC oder PG mit Twisted Pair Ethernet-Anschluss

#### *Übertragungsstrecke*

- 3 Buskabel
- 1 Switch/Hub

#### *Adressen*

- 2 IP Adressen und Subnet-Masken für 2 CPs

#### *Software-Pakete*

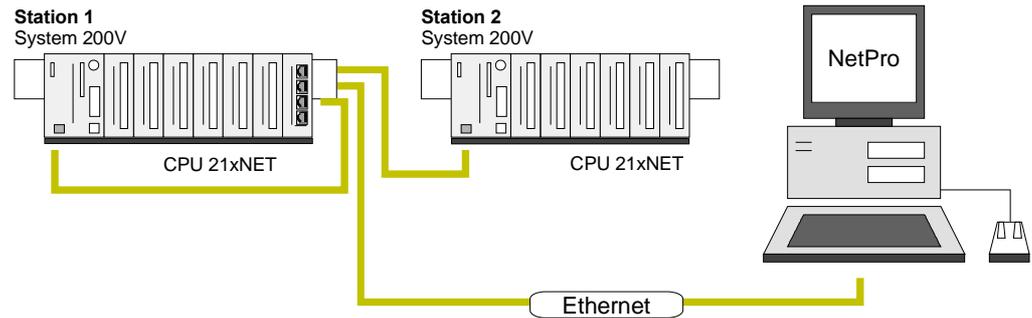
- Siemens SIMATIC Manager V. 5.1 oder höher
- Siemens SIMATIC NET

Zur Realisierung des Beispiels ist die Programmierung der zwei CPUs sowie die Parametrierung der Kommunikationsprozessoren unter NetPro von Siemens erforderlich.



### **Hinweis!**

Das Beispiel finden Sie auf [ftp.vipa.de/support/demofiles](ftp://ftp.vipa.de/support/demofiles) als ZIP-Datei. Das SPS-Programm können Sie direkt in beide CPUs übertragen.

**Aufbau****Aufgaben für die Stationen**

Dem Beispiel wird eine Kommunikationsaufgabe zugrunde gelegt, die im Folgenden näher erläutert wird:

In beiden CPUs läuft das gleiche SPS-Programm, lediglich die Projektierung der CP-Teile ist auf die jeweilige Station anzupassen.

Beide Stationen senden und empfangen im Sekundentakt 16 Datenworte.

- Im Datenbaustein DB 11 werden die Datenbyte DBB 0 bis DBB 32 im Takt von 1s übertragen. Das Datenbyte DBB 0 im DB 11 dient hierbei als Telegrammzähler. Es wird nur dann inkrementiert, wenn der vorhergegangene Sendeauftrag korrekt (fertig ohne Fehler) abgearbeitet wurde. Über die restlichen Datenbyte (DBB 2 bis DBB 32) könnten Nutzdaten übertragen werden.
- Die empfangende Station legt die Daten in DB12 ab (DBB 0 bis DBB 31).
- Über NetPro ist eine aktive SEND/RECEIVE-Verbindung mit der ID 1 für den CP zu projektieren. Diese Verbindung erscheint bei der 2. Station als passive SEND/RECEIVE-Verbindung.
- Die Quell- und Zielparameter sind direkt zu parametrieren.

Die Aufgabenstellung und die erforderlichen Voreinstellungen sind somit umrissen. Weitere Einzelheiten zur Projektierung finden Sie auf den Folgeseiten

**Schritte der Projektierung**

Die Projektierung gliedert sich in folgende Teile:

- Hardware-Konfiguration
- CP-Projektierung unter NetPro
- SPS-Anwenderprogramm
- Projekt transferieren

**Hardware-Konfiguration Station 1**

- Starten Sie den Siemens SIMATIC Manager mit einem neuen Projekt.
- Fügen Sie mit **Einfügen** > *Station* > *SIMATIC 300-Station* eine neue System 300 Station ein und geben Sie Ihr den Namen "Station 1".
- Aktivieren Sie die Station "SIMATIC 300" und öffnen Sie den Hardware-Konfigurator indem Sie auf "Hardware" klicken.
- Projektieren Sie ein Rack (Simatic300 \ Rack-300 \ Profilschiene).
- Projektieren Sie stellvertretend für Ihre CPU 21xNET die Siemens CPU 315-2DP mit der Best.-Nr. 6ES7 315-2AF03-0AB0 V1.2., zu finden unter SIMATIC 300 \ CPU 300 \ CPU 315-2 DP. Parametrieren Sie ggf. die CPU 315-2DP.
- Projektieren Sie stellvertretend für den CP den Siemens-CP **CP343-1 (343-1EX11)** auf Steckplatz 4, zu finden unter SIMATIC 300 \ CP 300 \ Industrial Ethernet \ CP 343-1.
- Geben Sie in den "CP-Eigenschaften" die gewünschte IP-Adresse und Subnet-Maske an.

Für dieses Beispiel ist eine Projektierung des System 200V in Form eines virtuellen PROFIBUS-System nicht erforderlich.

**Hardware-Konfiguration Station 2**

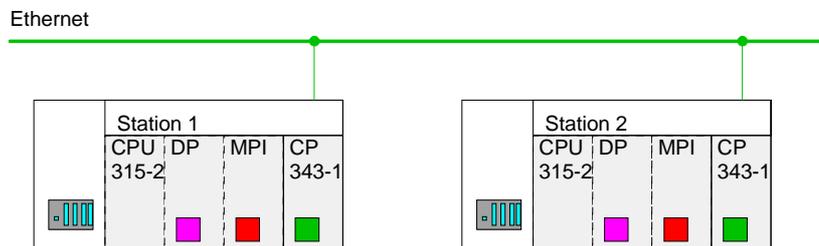
Erzeugen Sie gemäß der oben gezeigten Vorgehensweise eine Hardware-Konfiguration für die Ziel-CPU und geben Sie dieser den Namen "Station 2".

Verwenden Sie hierbei für den CP die für Station 2 zugeteilte IP-Adresse, Subnet-Maske und Gateway.

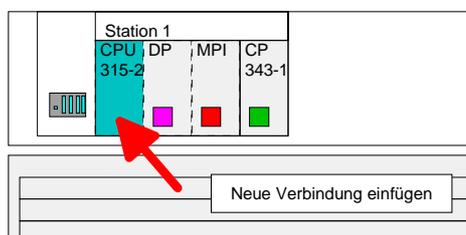
Speichern und übersetzen Sie Ihr Projekt.

**CP-Projektierung unter NetPro**

Starten Sie NetPro indem Sie unter *Station 1* die CPU anwählen und auf das Objekt "Verbindungen" klicken.  
 In NetPro werden "Station 1" und "Station 2" aufgelistet verbunden mit Ethernet dargestellt.



Zur Projektierung der Verbindung blenden Sie die Verbindungsliste ein. Wählen Sie hierzu die CPU von Station 1 an und rufen Sie über das Kontextmenü *Neue Verbindung einfügen* auf:



Es öffnet sich ein Dialogfenster in dem Sie den Verbindungspartner auswählen und den Typ der Verbindung einstellen können.  
 Projektieren Sie folgende Verbindung:

*Neue Verbindung*

Verbindung: TCP-Verbindung  
 Verbindungspartner: Station 2 > CPU 315-2

*Eigenschaften TCP-Verbindung*

ID: 1  
*ID* und *LADDR* sind Parameter, die in Ihrem SPS-Programm bei Verwendung der FC5 (AG\_SEND) und FC6 (AG\_RECEIVE) anzugeben sind.  
 Wegewahl: Mit der Wegewahl können Sie den entsprechenden CP anwählen, über den die Verbindung laufen soll.  
 Zur Kommunikation zwischen den CPU 21xNET ist die Wegewahl "CP 343-1 - (R0/S4)" schon richtig eingestellt.  
 Aktiver Verbindungsaufbau: aktiviert

Speichern und Übersetzen Sie Ihre Verbindung.

**SPS-Anwenderprogramm**

Für die Verarbeitung der Verbindungsaufträge auf SPS-Seite ist ein SPS-Anwenderprogramm in der jeweiligen CPU erforderlich. Hierbei kommen ausschließlich die Hantierungsbausteine AG\_SEND (FC5) und AG\_RECV (FC6) zum Einsatz. Durch Einbindung dieser Bausteine in den Zyklus-Baustein OB1 mit den Parametern *ID* und *LADDR* können Sie zyklisch Daten senden und empfangen.

Die beiden FCs sind Bestandteil der VIPA-Library, die sich als CD im Lieferumfang der CPU befindet.

**OB 1  
Zyklus**

Über den Zyklus-OB OB1 wird das Senden und Empfangen der Daten gesteuert. Der OB1, den Sie in beide CPUs transferieren können, hat folgenden Aufbau:

```

UN      T      1          // Timer 1 getriggertes Senden
L      S5T#1S          // alle 1 Sec ein Sendeanstoß
SV     T      1
S      M      10.0      // Anstoß-Merker
CALL   "AG_SEND"
ACT    :=M10.0          // Anstoß-Merker
ID     :=1             // Verbindungsnummer
LADDR :=W#16#110       // Baugruppenadresse
SEND   :=P#DB11.DBX0.0 BYTE 100 // Sendepuffer Bereich DB11
LEN    :=32            // 32 Byte (16 Worte) aus DB11 senden
DONE   :=M10.1
ERROR  :=#Senderror    // Temporärer Fehler-Merker
STATUS:=MW12           // Auftrags- bzw. Verbindungsstatus
U      M      10.1      // Senden fertig?
SPBN   nDon
U      M      10.1      // Senden fertig
R      M      10.0      // Anstoß rücksetzen
U      #Senderror      // Bei Sendeerror
SPB    nDon            // Sendezähler nicht erhöhen
L      DB11.DBW      0  // Sendezähler in den Nutzdaten (DBW0)
L      1              // um eins inkrementieren und
+I     // wieder im Sendepuffer ablegen
T      DB11.DBW      0

nDon:  NOP      0          // Senden noch nicht fertig

// Zyklischer Aufruf des ReceiveBausteins

CALL   "AG_RECV"
ID     :=1             // Verbindungsnummer
LADDR :=W#16#110       // Baugruppenadresse
RECV   :=P#DB12.DBX100.0 BYTE 32 //Empfangspuffer
NDR    :=#Newdata      // NewDataReceived?
ERROR  :=M0.1          // RecError
STATUS:=MW2            // Auftrags- bzw. Verbindungsstatus
LEN    :=#Reclen       // tatsächlich empfangene Länge
NOP    0                // Reclen kann bei IsoOnTCP < 32 sein
U      #Newdata        // wenn neue Daten empfangen
ZV     Z      1          // Empfangszähler Zähler1 inkrementieren
L      Z      1          // Zähler 1 bei Überlauf zurücksetzen
L      999
==I
R      Z      1

```

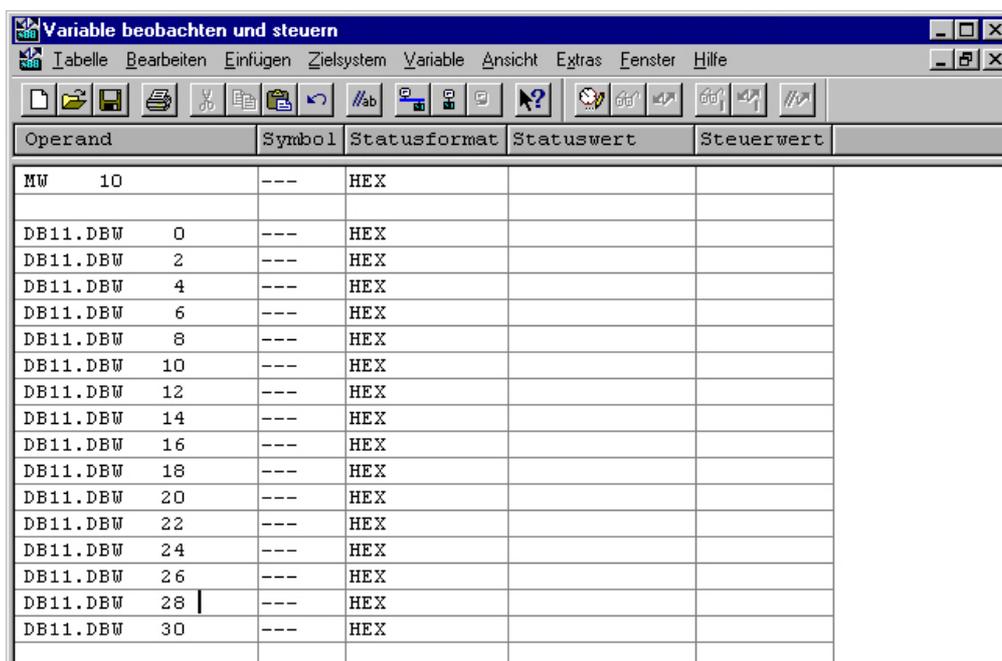
**Beobachtung der Übertragung im Siemens SIMATIC Manager**

Als Ausgangspunkt werden parametrisierte CPs und urgelöschte CPUs, deren RUN/STOP-Schalter in der Grundstellung STOP steht, vorausgesetzt.

Übertragen Sie das zuvor beschriebene Kommunikationsprojekt in beide CPUs und bringen Sie diese in RUN.

Starten Sie den Siemens SIMATIC Manager und führen Sie zur Beobachtung des Sendeauftrags die folgenden Schritte aus:

- **Zielsystem** > *Variable beobachten/steuern*
- Tragen Sie unter "Operand" die entsprechende Datenbaustein-Nr. und das Datenwort ein (DB11.DBB 0-31).
- Stellen Sie eine Verbindung her und klicken Sie auf "beobachten" .



Operand	Symbol	Statusformat	Statuswert	Steuerwert
MW 10	---	HEX		
DB11.DBB 0	---	HEX		
DB11.DBB 2	---	HEX		
DB11.DBB 4	---	HEX		
DB11.DBB 6	---	HEX		
DB11.DBB 8	---	HEX		
DB11.DBB 10	---	HEX		
DB11.DBB 12	---	HEX		
DB11.DBB 14	---	HEX		
DB11.DBB 16	---	HEX		
DB11.DBB 18	---	HEX		
DB11.DBB 20	---	HEX		
DB11.DBB 22	---	HEX		
DB11.DBB 24	---	HEX		
DB11.DBB 26	---	HEX		
DB11.DBB 28	---	HEX		
DB11.DBB 30	---	HEX		

**Nutzdaten eingeben**

Ab DBB 2 können Nutzdaten eingetragen werden. Gehen Sie hierzu mit dem Cursor auf *Steuerwert* und tragen Sie einen zu übertragenden Wert ein, wie z.B. W#16#1111.

Mit  übertragen Sie den Steuerwert bei jedem Zyklusdurchlauf bzw. mit  einmalig.

## Teil 6 Einsatz CPU 21x-2BT02 unter H1 / TCP/IP

### Überblick

In folgendem Kapitel ist der Einsatz der CPU 21x-2BT02 und die Kommunikation unter H1 bzw. TCP/IP beschrieben. Sie finden hier auch eine Einführung in die Projektierung unter WinNCS mit einem konkreten Kommunikationsbeispiel.

### Inhalt

Thema	Seite
<b>Teil 6 Einsatz CPU 21x-2BT02 unter H1 / TCP/IP</b> .....	<b>6-1</b>
Grundlagen .....	6-2
Planung eines Netzwerks .....	6-7
MAC- und IP-Adresse .....	6-9
Projektierung der CPU 21x-2BT02 .....	6-11
Beispiel zur Kommunikation CPU 21x-2BT02.....	6-23
Anlaufverhalten .....	6-34
Systemeigenschaften der CPU 21x-2BT02 .....	6-35
Kopplung mit Fremdsystemen.....	6-37
Testprogramm für TCP/IP-Verbindungen.....	6-40

## Grundlagen

- Netzwerk** Ein Netzwerk verbindet verschiedene Netzwerkstationen so, dass diese miteinander kommunizieren können.  
Netzwerkstationen können PCs, IPCs, H1/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 Netzkabel durchgesetzt. Die CPU 21xNET hat einen Twisted-Pair-Anschluss.  
Das Twisted Pair Kabel besteht aus 4 Adern, die paarweise miteinander verdreht sind. Aufgrund der Verdrehung ist dieses System nicht so stör anfällig wie frühere Koaxialnetze.  
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.
- Sternkoppler (Hub)** Der Hub ist das zentrale 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.
- 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.

**Zugriffssteuerung**

Bei Ethernet gibt es das Prinzip des zufälligen Buszugriffs: Jeder Teilnehmer greift bei Bedarf von sich aus auf den Bus zu. Koordiniert wird der Buszugriff dabei durch das Verfahren CSMA/CD (Carrier Sense Multiple Access/Collision Detection - 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.

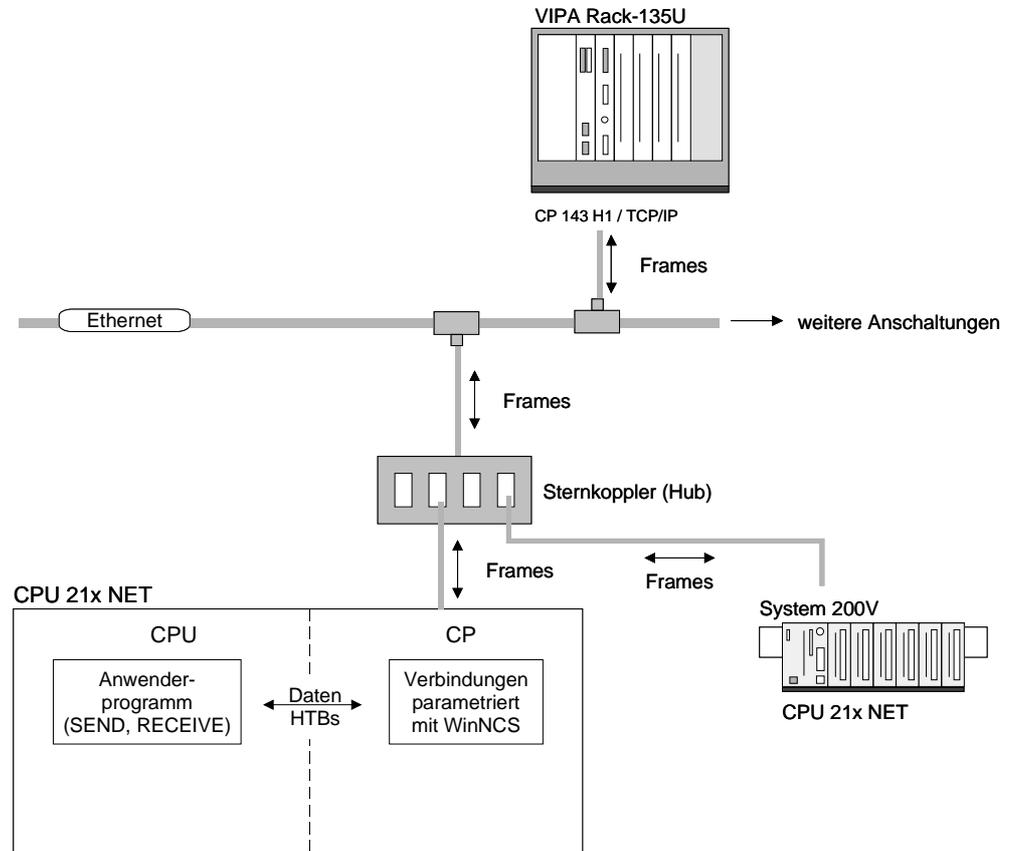
**Kommunikation**

Der interne CP der CPU 21x-2BT02 ist über ein Dual-Port-RAM, auch "Kachel" genannt, direkt mit der CPU 21xNET verbunden.

Diese Kachel steht auf der CPU-Seite als Standard-CP-Interface zur Verfügung. Der Datenaustausch findet über die Standard Handierungsbausteine (SEND und RECEIVE) statt.

Die Kommunikation unter ISO-Transport (H1) bzw. TCP/IP regeln Verbindungen, die unter dem VIPA-Parametriertool WinNCS zu parametrieren sind und direkt über den Twisted-Pair-Anschluss in die CPU übertragen werden.

Näheres zur Vorgehensweise bei der Parametrierung finden Sie im Handbuch von WinNCS (HB91).



**ISO-Transport (H1)**

ISO-Transport ist ein Protokoll, das auf dem Ethernet-Standard aufsetzt. Der Nachrichtenaustausch zwischen den Stationen erfolgt in ISO-Transport über ISO-Transport-Frames, die über Transportverbindungen übertragen werden.

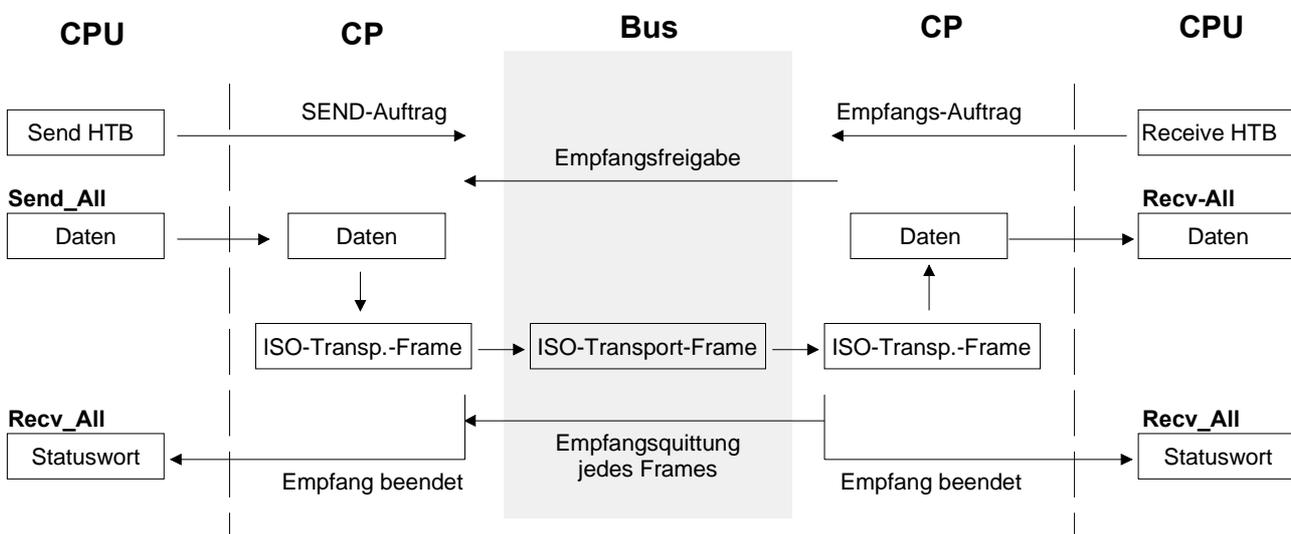
Eine Transportverbindung ist eine logische Verbindung zwischen zwei Zugangspunkten für Transport-Dienste auf verschiedenen Stationen. Eine Transportverbindung basiert auf Adressinformationen, die eindeutig den Transportweg zwischen den beiden Zugangspunkten beschreiben.

Eine Transportverbindung wird über folgende Parameter eindeutig beschrieben:

- *MAC-Adresse* auch Stationsadresse, kennzeichnet eindeutig den Zugang zu einer Station.
- *TSAP* Transport-Service-Access-Points kennzeichnet Zugangskanäle für Dienste des Transportprotokolls.

Der CP stellt einen Datenpuffer bereit und transferiert über die Hintergrundkommunikation SEND\_ALL die zu sendenden Daten in den Datenpuffer. Danach bildet der CP einen ISO-Transport-Frame und überträgt, sobald eine Empfangsfreigabe erfolgt, dieses Frame an die Partnerstation. Nach dem Empfang des ISO-Transport-Frames in der Partnerstation erhält der CP eine Empfangsquittung und transferiert über die Hintergrundkommunikation RECEIVE-ALL den Status des SEND-Auftrags in das zugehörige Anzeigenwort.

Auf diese Weise ist eine fehlerfreie Übertragung sichergestellt. Die nachfolgende Abbildung soll dies nochmals verdeutlichen:



**Hinweis!**

Aufgrund der vielen Quittungstelegramme, die bei einer ISO-Transport-Verbindung übertragen werden, ist die Netzbelastung unter ISO-Transport wesentlich höher als unter TCP/IP, wobei unter TCP/IP die Datensicherheit geringer ist!

**TCP/IP**

TCP/IP-Protokolle stehen auf allen derzeit bedeutenden Systemen zur Verfügung. Dies gilt am unteren Ende für einfache PCs, über die typischen Mini-Rechner, bis hinauf zu Großrechnern (auch für IBM-Systeme existieren TCP/IP-Implementierungen) und Spezialrechnern wie Vektorrechner und Parallelrechner. TCP/IP wird deshalb häufig für den Aufbau heterogener Systemverbunde verwendet.

Mit TCP/IP lassen sich umfassende, offene Netzwerklösungen zwischen den einzelnen Bereichen eines Unternehmens realisieren.

Mit TCP/IP lassen sich zum Beispiel folgende Aufgaben lösen:

- zentrale Steuerung und Überwachung von Fertigungsanlagen,
- Übermittlung von Maschinenzuständen,
- Managementinformationen,
- Produktionsstatistik,
- Übertragung großer Datenmengen.

TCP und IP sind nur zwei der für den Aufbau einer vollständigen Architektur erforderlichen Protokolle. Die Anwendungsschicht stellt Programme wie "FTP" und "Telnet" auf PC-Seite zur Verfügung.

Die Anwendungsschicht des CPs der CPU 21x2BT02 ist mit dem Anwenderprogramm unter Verwendung der Standardhantierungsbausteine definiert.

Diese Anwendungsprogramme nutzen für den Datenaustausch die Transportschicht mit den Protokollen TCP oder UDP, die wiederum mit dem IP-Protokoll der Internetschicht kommunizieren.

*IP*

Die Hauptaufgabe von IP besteht in der Adressierung der Datenpakete. IP hat dieselbe Funktion wie ein Briefumschlag. Anhand der Adresse erkennt das Netzwerk den Bestimmungsort und kann die Datenpakete innerhalb des Netzwerks entsprechend weiterleiten.

Das Protokoll teilt die Daten in kleine Einheiten auf, da verschiedene Netzwerke unterschiedliche Datenlängen verwenden können.

Jedes Paket erhält eine Nummer, so dass der Empfang bestätigt werden kann und die Daten wieder rekonstruiert werden können. Um diese Folgenummern über das Netz senden zu können, hat TCP wie auch IP einen eigenen Umschlag, auf denen es die benötigten Nummern notiert.

*TCP*

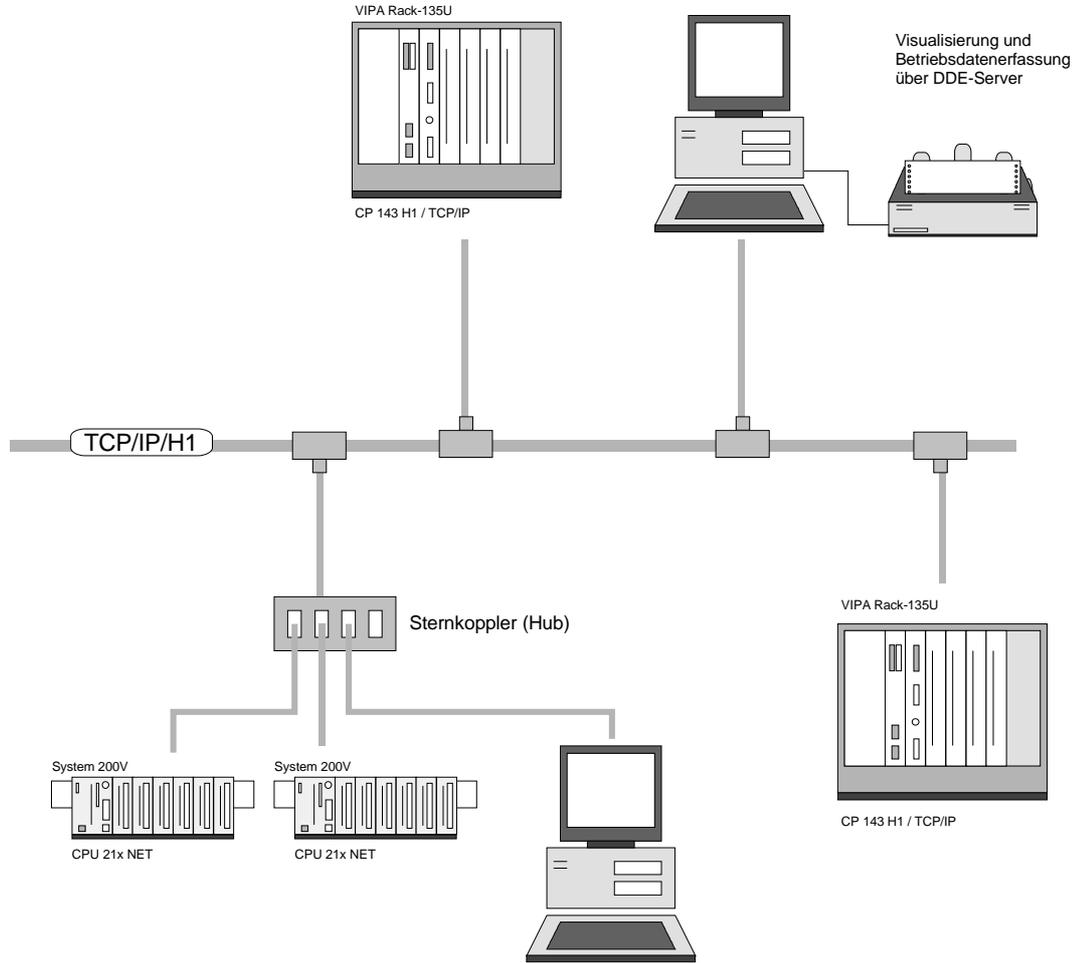
Ein Paket mit Daten wird in einen TCP-Umschlag gepackt. Dieser wird seinerseits in einen IP-Umschlag gesteckt und an das Netzwerk gesendet. Hier sorgt TCP für den sicheren Transport der Daten durch das Netzwerk. TCP erkennt Übertragungsfehler und korrigiert diese.

Dieses Sicherheitssystem macht TCP-Verbindungen verhältnismäßig sicher.

UDP ist in diesem Fall wesentlich schneller. Es kümmert sich aber weder um fehlende Datenpakete, noch um die Reihenfolge der Pakete. UDP ist ein ungesichertes Protokoll.

**Anwendungs-  
beispiel für ISO-  
Transport bzw.  
TCP/IP**

*Einsatz unter TCP/IP bzw. ISO-Transport*



## Planung eines Netzwerks

### Normen und Richtlinien

Zur Kommunikation zwischen einzelnen Stationen gibt es gewisse Vorschriften und Regeln, die einzuhalten sind. Hierbei werden die Form des Datenprotokolls, das Zugriffsverfahren auf den Bus und weitere, für die Kommunikation wichtige Grundlagen definiert.

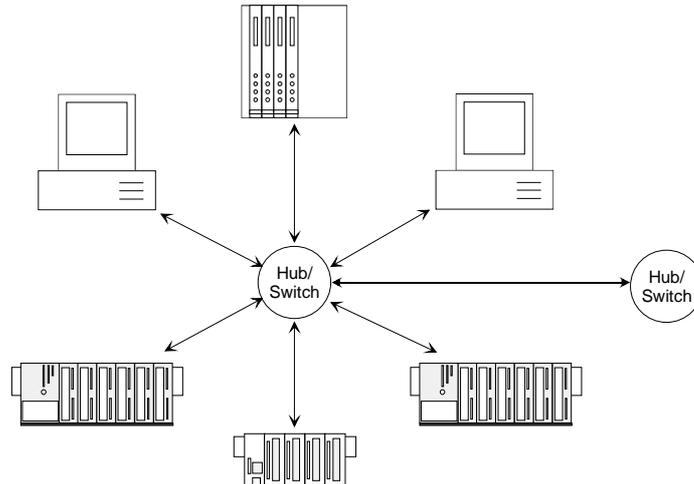
Basierend auf den von ISO festgelegten Standards und Normen wurde die CPU 21xNET von VIPA entwickelt.

In den folgenden internationalen und nationalen Gremien sind Normen und Richtlinien für Netzwerktechnologien festgelegt worden:

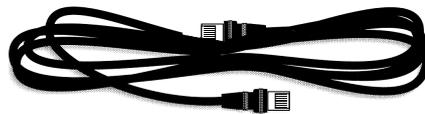
ANSI	American National Standards Institute Hier werden zur Zeit in der ANSI X3T9.5 Vereinbarungen für LANs mit hohen Übertragungsgeschwindigkeiten (100 MB/s) auf Glasfaserbasis formuliert. (FDDI) Fibre Distributed Data Interface.
CCITT	Committee Consultative Internationale de Telephone et Telegraph. Von diesem beratenden Ausschuss werden unter anderem die Vereinbarungen für die Anbindung von Industriekommunikationsnetzen (MAP) und Büronetzen (TOP) an Wide Area Networks (WAN) erstellt.
ECMA	European Computer Manufacturers Association. Hier werden verschiedene Standards für MAP und TOP erarbeitet.
EIA	Electrical Industries Association (USA) Standardfestlegungen wie RS-232 (V.24) und RS-511 sind in diesem Ausschuss erarbeitet worden.
IEC	International Electrotechnical Commission. Hier werden einzelne spezielle Standards festgelegt. z.B. für Feld Bus.
ISO	International Organisation for Standardization. In diesem Verband der nationalen Normungsstellen wurde das OSI-Modell entwickelt (ISO/TC97/SC16). Es gibt den Rahmen vor, an den sich die Normungen für die Datenkommunikation halten sollen. ISO Standards gehen über in die einzelnen nationalen Standards wie z.B. UL und DIN.
IEEE	Institute of Electrical and Electronic Engineers (USA). In der Projektgruppe 802 werden die LAN-Standards für Übertragungsraten von 1 bis 1000MB/s festgelegt. IEEE Standards bilden häufig die Grundlage für ISO-Standards z.B. IEEE 802.3 = ISO 8802.3.

## Übersicht der Komponenten

Die CPU 21x-2BT02 ist ausschließlich für den Einsatz in einem Twisted-Pair-Netz geeignet. Bei einem Twisted-Pair-Netz werden alle teilnehmenden Stationen sternförmig über Twisted-Pair-Kabel mit einem Hub/Switch verbunden, der seinerseits mit weiteren Hubs/Switches kommunizieren kann. Zwei verbundene Stationen bilden ein Segment, wobei die Länge des Twisted-Pair-Kabels zwischen den Stationen max. 100m betragen darf.



Twisted Pair Kabel



Bei einem Twisted Pair-Kabel handelt es sich um ein Kabel mit 8 Adern, die paarweise miteinander verdreht sind.

Die einzelnen Adern haben einen jeweiligen Durchmesser von 0,4 bis 0,6mm.

## Ermitteln des Netzwerkbedarfs

- Welche Fläche muss mit dem Kabelsystem abgedeckt werden?
- Wie viele Netzwerksegmente lösen am besten die physikalischen (räumlich, störungsbedingt) Gegebenheiten der Anlage?
- Wie viele Netzwerkstationen (SPS, IPC, PC, Transceiver, evtl. Bridges) sollen an das Kabelsystem angeschlossen werden?
- In welchem Abstand stehen die Netzwerkstationen voneinander getrennt?
- Welches "Wachstum" in Größe und Anzahl der Verbindungen muss das System bewältigen können?
- Welches Datenaufkommen ist zu bewältigen (Bandbreite, Zugriffe/Sec.)?

## Zeichnen des Netzwerkplans

Zeichnen Sie Ihren Netzwerkplan. Bezeichnen Sie jedes Stück Hardware, das verwendet wird (wie Stationskabel, Hub, Switch). Halten Sie die Regeln und Grenzwerte im Auge.

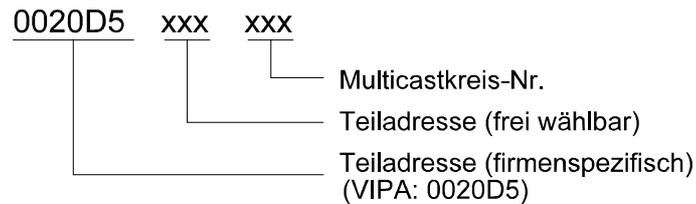
Messen Sie die Distanz zwischen allen Komponenten um sicher zu gehen, dass jeweils die maximale Länge nicht überschritten wird.

## MAC- und IP-Adresse

### Aufbau MAC-Adresse

Mit der MAC-Adresse, auch Stationsadresse genannt, wird eine Station adressiert. In einem Netz dürfen nicht mehrere Stationen mit der gleichen MAC-Adresse existieren.

Die MAC-Adresse hat folgenden Aufbau:



Die MAC-Adresse hat eine Länge von 6Byte. Die ersten 3Byte legt der Hersteller fest. Diese 3Byte werden vom IEEE-Komitee vergeben. Die letzten 3Byte können frei vergeben werden.

In Anlagen werden die MAC-Adressen vom Netzwerkadministrator vergeben.

Die Broadcastadresse (Nachricht an alle Teilnehmer) ist immer:

`FFFFFFFFFFFFh`

### Aufbau IP-Adresse

Die IP-Adresse ist eine 32Bit-Adresse, die innerhalb des Netzes eindeutig sein muss. Die IP-Adresse setzt sich aus 4 Zahlen zusammen, die jeweils durch einen Punkt getrennt sind.

Die IP-Adresse hat folgenden Aufbau: `XXX.XXX.XXX.XXX`

Wertebereich: `000.000.000.000` bis `255.255.255.255`

Auch die IP-Adressen werden vom Netzwerkadministrator vergeben.

Die Broadcastadresse (Nachricht an alle Teilnehmer) ist immer:

`255.255.255.255`



### Achtung!

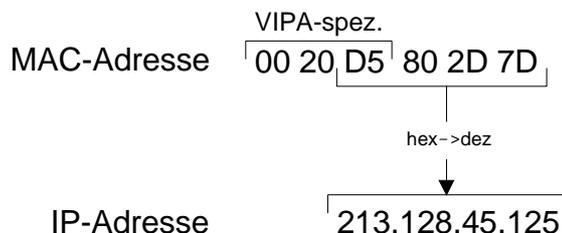
Es gibt IP-Adressen, die nicht verwendet werden dürfen! Diese Adressen sind für spezielle Dienste reserviert!

**Adresse bei Erst-  
inbetriebnahme**

Bei der Erstinbetriebnahme einer CPU 21x-2BT02 besitzt das Modul seine ursprüngliche MAC-Adresse (Stationsadresse).

Sie finden diese Adresse auf einem Aufkleber, der sich an der Seite des Moduls befindet.

Aus dieser MAC-Adresse wird nur bei der Erstinbetriebnahme eine eindeutige IP-Adresse nach folgender Umrechnung gebildet.

**Hinweis!**

Eine Beziehung zwischen MAC- und IP-Adresse besteht nur bei der Erstinbetriebnahme.

Sie können jederzeit in WinNCS unter CP-Init eine andere Ethernet- und IP-Adresse zuweisen.

**Achtung!**

Die ursprüngliche Beziehung zwischen Ethernet- und IP-Adresse besteht nur im Auslieferungszustand!

## Projektierung der CPU 21x-2BT02

### Übersicht

Die Projektierung des CP-Teils besteht aus 3 Teilen:

- **CP-Projektierung** über WinNCS von VIPA (Verbindung zum Ethernet).
- **Hardware-Konfiguration** (Einbindung CP in CPU)
- **SPS-Programmierung** über Anwender-Programm (Verbindung zur SPS).

### Schnelleinstieg

#### Voraussetzung

CP ist an Ethernet angebunden, wird mit Spannung versorgt und ist hochgelaufen.

#### CP-Projektierung unter WinNCS

Starten Sie WinNCS und stellen Sie über  die Funktionalität "Ethernet" ein.

- Legen Sie über **Datei** > *Projekt anlegen/öffnen* ein Projekt mit der Funktionsgruppe "Ethernet" an.

- Aktivieren Sie mit  die Onlinefunktionen.

- Durch Angabe der MAC-Adresse (siehe Aufkleber) können Sie auf den CP online zugreifen. Stellen Sie hierzu unter  "H1-Protokoll" für ISO-Transport ein und geben Sie die MAC-Adresse als Stationsadresse an.

- Stellen Sie mit  eine Verbindung her.

- Bringen Sie mit  Ihren CP in den Software-STOP-Zustand.
- Sie können jetzt entweder die Konfiguration des CP in WinNCS importieren, anpassen und wieder hochladen oder mit "CP-Init" die Stationseigenschaften einstellen und entsprechende Verbindungen projektieren.

### ... Fortsetzung Schnelleinstieg

#### Transfer der CP-Projektierung

Klicken Sie vor der Übertragung im "Netzwerk"-Fenster auf die Station, deren Projekt Sie transferieren möchten.

- Übertragen Sie mit  Ihr Projekt in den CP. Sollte stattdessen eine Abfrage nach einem NCS-File erfolgen, haben Sie im "Netzwerk"-Fenster keinen CP angewählt! Wählen Sie diesen an und starten Sie nochmals den Transfer.
- Da die Daten im ungesicherten RAM abgelegt werden, können Sie diese mit  dauerhaft im Flash-ROM speichern. Durch erneutes "Flashen" können Sie diese jederzeit überschreiben.
- Sobald der Transfer fertig ist, ist mit  der CP neu zu booten.

Danach befindet sich der CP mit den projektierten Adresse-Parametern am Netz.

#### Hardware-Konfiguration

Voraussetzung: Siemens SIMATIC Manager ab V. 5.1 und Siemens SIMATIC NET

- Starten Sie den Siemens SIMATIC Manager mit einem neuen Projekt.
- Fügen Sie mit **Einfügen** > *Station* > *SIMATIC 300-Station* eine neue System 300 Station ein.
- Aktivieren Sie die Station "SIMATIC 300" und öffnen Sie den Hardware-Konfigurator indem Sie auf "Hardware" klicken.
- Projektieren Sie ein Rack (Simatic300 > Rack-300 > Profilschiene).
- Da alle 21x CPUs von VIPA als CPU 315-2DP projektiert werden, projektieren Sie aus dem Hardwarekatalog die CPU 315-2DP mit der Best.-Nr. 6ES7 315-2AF03-0AB0 V1.2.  
Sie finden diese unter Simatic300 > CPU 300 > CPU 315-2 DP.
- Parametrieren Sie ggf. die CPU 315-2DP und legen Sie ein neues PROFIBUS-Subnetz an.
- Hängen Sie an das Subnetz das System "VIPA\_CPU21x". Sie finden dies im Hardware-Katalog unter *PROFIBUS DP* > *Weitere Feldgeräte* > *IO* > *VIPA\_System\_200V*. Geben Sie diesem Slave die PROFIBUS-Adresse 1.
- Platzieren Sie in Ihrem Konfigurator auf dem 1. Steckplatz die CPU 21x-2BT02, die Sie einsetzen, indem Sie diese dem Hardware-Katalog entnehmen.
- Binden Sie dahinter Ihre System 200V Module in der gesteckten Reihenfolge beginnend mit Steckplatz 1 ein.
- Sichern Sie Ihr Projekt.

**... Fortsetzung  
Schnelleinstieg**

Die Übertragung der Hardware-Konfiguration erfolgt zusammen mit dem Anwenderprogramm.

**Anwenderprogramm**

- Erstellen Sie mit dem SYNCHRON-Baustein einen Kommunikationskanal zwischen CPU und CP.
- Programmieren Sie zum Anstoß von Sende- und Empfangsaufträgen die entsprechenden SEND und RECEIVE-Bausteine.
- Programmieren Sie zur Datenübertragung die Bausteine SEND\_ALL bzw. RECEIVE\_ALL.

**Transfer von Anwenderprogramm und Hardware-Konfiguration**

Zur Übertragung Ihres Anwenderprogramms und der Hardware-Konfiguration stehen Ihnen folgende Möglichkeiten zur Verfügung:

**a) Transfer über MPI****b) Transfer über MMC****zu a) Transfer über MPI**

- Verbinden Sie Ihr PG bzw. Ihren PC über MPI mit Ihrer CPU. Für serielle Punkt-zu-Punkt-Verbindungen können Sie das Green Cable von VIPA verwenden (Näheres hierzu siehe Teil "Einsatz CPU 21X" unter "Projekt transferieren").
- Konfigurieren Sie im Siemens SIMATIC Manager unter **Extras** > *PG/PC-Schnittstelle einstellen* die MPI-Schnittstelle Ihres PC. Stellen Sie bei Einsatz des Green Cable eine Übertragungsgeschwindigkeit von 38400Baud ein.
- Mit **Zielsystem** > *Laden in Baugruppe* übertragen Sie Ihr Projekt in die CPU.

**zu b) Transfer über MMC**

Das Lesen der MMC erfolgt immer nach URLÖSCHEN.

Beschrieben wird die MMC entweder mit **Zielsystem** > *RAM nach ROM* oder mit einem MMC-Lesegerät von VIPA (Best.-Nr: VIPA 950-0AD00).

Bitte beachten Sie, dass sich Ihre aktuelle Projektierung im Root-Verzeichnis befindet und folgenden Dateinamen hat: **S7PROG.WLD**.

Während des Schreibvorgangs blinkt die gelbe "MMC"-LED der CPU.

Gleichzeitig erfolgt ein Schreibvorgang in das interne Flash der CPU.

Bei einem erfolgreichen Schreibvorgang finden Sie 0xE200 und 0xE300 im Diagnosepuffer.

Ist kein gültiges Anwenderprogramm auf der gesteckten MMC oder scheitert die Übertragung, so erfolgt das URLÖSCHEN der CPU und die STOP-LED blinkt dreimal.

Hier endet der Schnelleinstieg. Auf den Folgeseiten finden Sie nun eine ausführliche Beschreibung der Projektierung.

## CP-Projektierung mit WinNCS

Die Projektierung des CP-Teils der CPU 21x-2BT02 findet ausschließlich unter WinNCS statt und gliedert sich in folgende 3 Teile:

- CP-Grundprojektierung,
- Verbindungsbausteine projektieren,
- Projektierdaten in den CP übertragen.

## CP-Grundprojektierung

Hier werden Adresse und weitere Identifikationsparameter einer Station parametrieren.

Fügen Sie unter der Funktionalität "Ethernet" eine neue Station im Netzwerkfenster ein und führen Sie im Parameterfenster die Parametrierung Ihrer Station durch.

The screenshot shows a 'Parameter' dialog box with the following fields and values:

Field	Value
Datum	29.05.00
Version	V 1.0
Stationsname	SPS 1 mit H1
Kachelbasisadresse	0
Kachelanzahl	1
Stationsadresse	0020d5000001
IP-Adresse	172.016.129.148
Subnet-Maske	255.255.224.000
Router1	000.000.000.000
Router2	000.000.000.000
Router3	000.000.000.000

Buttons: Übernehmen, Verwerfen, Hilfe

Über die CP-Grundprojektierung bestimmen Sie das Verhalten Ihrer Station im Netz.

## Verbindungsbausteine projektieren

Ein Verbindungsbaustein beinhaltet die fernen, d.h. zum Partner im Netz orientierten und die lokalen, d.h. zum SPS-Programm orientierten Parameter einer Verbindung.

Abhängig von dem gewünschten Protokoll können Sie ISO-Transport-(H1)- bzw. TCP/IP-Verbindungen parametrieren, indem Sie bei angewähltem Stationssymbol die gewünschte Verbindung einfügen und parametrieren.

ISO-Transport-(H1)-Verbindung

Parameter											
H1-Transport Verbindung   Multiverbindungen   Systemparameter											
Verbindungsname: Transport											
Kacheloffset: 0	Auftragsart: Send										
Auftragsnummer: 1											
Priorität: 2											
<table border="0"> <tr> <td><b>Lokaler TSAP</b></td> <td><b>Fremder TSAP</b></td> </tr> <tr> <td>Asc: nordpol</td> <td>Asc: südpol</td> </tr> <tr> <td>Länge: 8</td> <td>Länge: 8</td> </tr> <tr> <td>Hex: 6E6F7264706F6C</td> <td>Hex: 73FC64706F6C</td> </tr> <tr> <td></td> <td>Adresse: 0020D5000000</td> </tr> </table>		<b>Lokaler TSAP</b>	<b>Fremder TSAP</b>	Asc: nordpol	Asc: südpol	Länge: 8	Länge: 8	Hex: 6E6F7264706F6C	Hex: 73FC64706F6C		Adresse: 0020D5000000
<b>Lokaler TSAP</b>	<b>Fremder TSAP</b>										
Asc: nordpol	Asc: südpol										
Länge: 8	Länge: 8										
Hex: 6E6F7264706F6C	Hex: 73FC64706F6C										
	Adresse: 0020D5000000										
Übernehmen   Verwerfen   Hilfe											

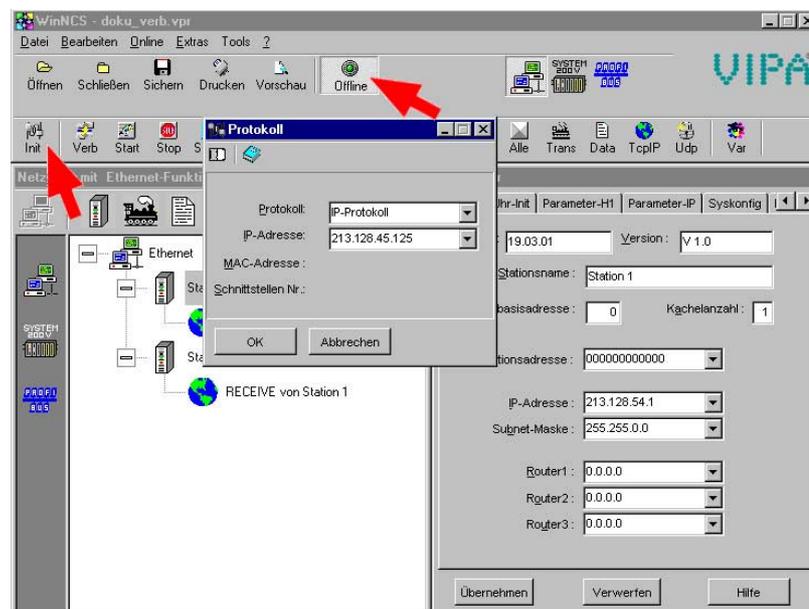
TCP/IP-Verbindung

Parameter													
TCP Verbindung   Multiverbindung   Systemparameter													
Verbindungsname: Tcp ...													
Kacheloffset: 0	Auftragsart: Send												
Auftragsnummer: 5	Auftragstyp: Einzelauftrag												
Priorität: 2													
<table border="0"> <tr> <td><b>Lokale Station :</b></td> <td><b>Fremde Station :</b></td> </tr> <tr> <td>Port: 1002</td> <td>Port: 1004</td> </tr> <tr> <td></td> <td>IP-Adresse: <input checked="" type="checkbox"/> 213.128.000.000</td> </tr> <tr> <td></td> <td>Host-Name</td> </tr> <tr> <td></td> <td>Fördereinheit</td> </tr> <tr> <td></td> <td>Versuche: 0</td> </tr> </table>		<b>Lokale Station :</b>	<b>Fremde Station :</b>	Port: 1002	Port: 1004		IP-Adresse: <input checked="" type="checkbox"/> 213.128.000.000		Host-Name		Fördereinheit		Versuche: 0
<b>Lokale Station :</b>	<b>Fremde Station :</b>												
Port: 1002	Port: 1004												
	IP-Adresse: <input checked="" type="checkbox"/> 213.128.000.000												
	Host-Name												
	Fördereinheit												
	Versuche: 0												
Übernehmen   Verwerfen   Hilfe													

## Projektierdaten in den CP übertragen

Sind alle Verbindungen parametrieren, müssen diese in den CP-Teil übertragen werden. Hierzu finden Sie unter WinNCS die "Online-Funktionen".

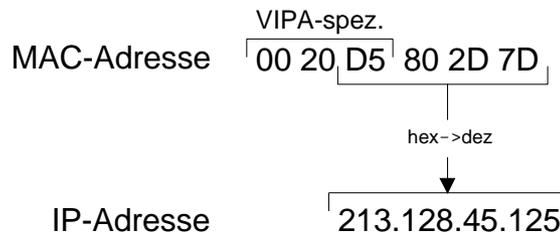
Zur Übertragung der Projektierdaten aktivieren Sie die Online-Funktionen und klicken Sie auf INIT:



Wie schon weiter oben beschrieben besitzt die CPU 21x-2BT02 bei der Erstinbetriebnahme ihre ursprüngliche MAC-Adresse.

Sie finden diese Adresse auf einem Aufkleber, der sich an der Seite des Moduls befindet.

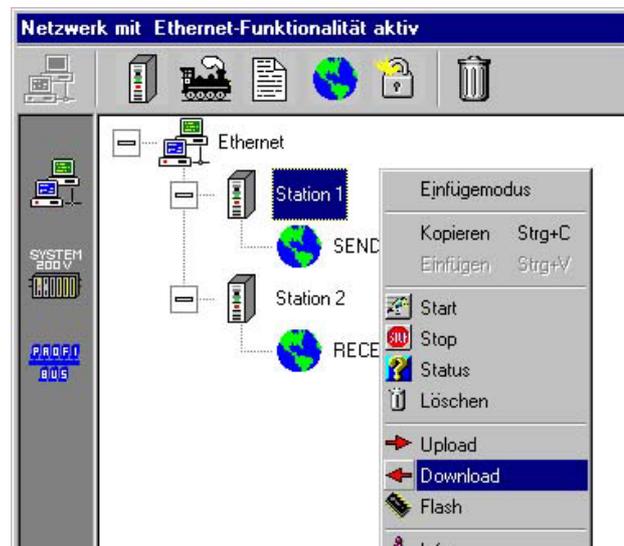
Aus dieser MAC-Adresse wird nur bei der Erstinbetriebnahme eine eindeutige IP-Adresse nach folgender Umrechnung gebildet.



Stellen Sie im *Protokoll*-Fenster "IP-Protokoll" ein und tragen Sie die ermittelte IP-Adresse ein. Bestätigen Sie Ihre Eingabe mit [OK].

Klicken Sie im *Netzwerk*-Fenster auf die entsprechende Station. Betätigen Sie die rechte Maustaste und wählen Sie den "Download"-Befehl.

Ihr Projekt wird nun direkt in das RAM des CPs übertragen.



Da die Daten im ungesicherten RAM abgelegt werden, können Sie diese

mit dauerhaft im Flash-ROM speichern. Durch erneutes "Flashen" können Sie diese jederzeit überschreiben.

**SPS-Anwender-  
Programmierung**

Für die Verarbeitung der Verbindungsaufträge auf SPS-Seite ist ein SPS-Anwenderprogramm in der CPU erforderlich. Hierbei kommen die Hantierungsbausteine (SEND, RECEIVE, ...) zum Einsatz, die u.a. in der CPU 21x-2BT02 integriert sind.

Auch bei der SPS-Programmierung ist zuerst ein Kommunikationskanal zwischen CPU und CP zu spezifizieren ("Synchronisation"). Hierzu dient der SYNCHRON-Baustein.

Sende- und Empfangsanstöße erfolgen über SEND bzw. RECEIVE. Und die Datenübertragung über SEND\_ALL bzw. RECV\_ALL.

Eventuelle Fehlermeldungen erscheinen im Anzeigenwort.

**Synchronisation**

In der CPU ist im Anlauf-OB OB 100 die benutzte Schnittstelle des CPs mittels des Hantierungsbausteins SYNCHRON zu synchronisieren.

Nach einem Netzein benötigt die CPU 21x-2BT02 ca. 15s für den Hochlauf. Wenn während dieser Anlaufphase von der CPU synchronisiert wird, erhält man im Parametrierfehlerbyte PAFE einen Fehler zurückgemeldet. Diese Meldung verschwindet, sobald der CP hochgelaufen ist.

Der Timer in diesem Baustein wird anfangs auf 20s gesetzt. Wenn innerhalb dieser Zeit nicht ordentlich synchronisiert wird, stoppt die Bearbeitung.

**Blockgröße**

Die einstellbare Blockgröße entnehmen Sie der nachfolgenden Tabelle.

Blockgröße	CP-Blockgröße in Byte
0	Default
1	16
2	32
3	64
4	128
5	256
6	512
255	512

**Zyklus**

Im Zyklusprogramm OB1 sind die Sende- und Empfangsbausteine SEND und RECEIVE für den Sende- und Empfangsanstoß zu parametrieren. Hiermit wird der Datentransfer zwischen CP und CPU gesteuert. Die Datenübertragung erfolgt mit den Bausteinen SEND\_ALL und RECV\_ALL.

Bei einer rein passiven Verbindung sind nur die Bausteine SEND\_ALL bzw. RECV\_ALL erforderlich.

Zur Sicherung der Datenübertragung sollten Sie Kontrollinstanzen durch Auswertung des Anzeigenworts integrieren.

**Hantierungs-  
bausteine**

In der nachfolgenden Tabelle sind die erforderlichen Hantierungsbausteine aufgelistet. Nähere Angaben hierzu finden Sie im Handbuch CPU 21x - Operationsliste.

SFC	Bezeichnung	Beschreibung
SFC 228	RW_Kachel	Kachel lesen/schreiben
SFC 230	Send	Senden über Kachel an CP
SFC 231	Receive	Empfangen über Kachel von CP
SFC 232	Fetch	Fetch veranlasst das Holen von Daten über Kachel. FETCH ist ausschließlich mit der RW-Kennung zugelassen und liefert den Auftragsanstoß zum Lesen.
SFC 233	Control	Der CONTROL-Baustein wird für die Statusabfrage bzgl. eines Auftrags verwendet, d.h. das ANZW eines definierten Auftrags wird aktualisiert.
SFC 234	Reset	Der RESET-Baustein veranlasst das Zurücksetzen eines Auftrags der angegebenen Verbindung.
SFC 235	Synchron	Der SYNCHRON stellt im Anlauf die Synchronisation zwischen CPU und CP her. Gleichzeitig werden die Kachel gelöscht und die Blockgröße zwischen CPU und CP ausgehandelt. Aktive Datenkommunikation kann nur über synchronisierte Kacheln stattfinden.
SFC 236	Send_All	Anstoß der Datenübergabe von CPU an CP.
SFC 237	Recv_All	Anstoß der Datenübernahme von CP zu CPU.
SFC 238	Control1	Control für Kachelkommunikation mit Typ ANZW : Zeiger und Parameter IND.

**Transfer von Anwenderprogramm und Hardware-Konfiguration**

Zur Übertragung Ihres Anwenderprogramms und der Hardware-Konfiguration stehen Ihnen folgende Möglichkeiten zur Verfügung:

- a) **Transfer über MPI**
- b) **Transfer über MMC**

zu a)  
Transfer über MPI

- Verbinden Sie Ihr PG bzw. Ihren PC über MPI mit Ihrer CPU. Sollte Ihr Programmiergerät keine MPI-Schnittstelle besitzen, können Sie für eine serielle Punkt-zu-Punkt-Übertragung von Ihrem PC an MPI das "Green Cable" von VIPA verwenden.  
Das "Green Cable" hat die Best.-Nr. VIPA 950-0KB00 und darf nur bei den VIPA CPUs mit MP<sup>2</sup>I-Schnittstelle eingesetzt werden. Bitte beachten Sie hierzu auch die Hinweise in den "Grundlagen".
- Konfigurieren Sie die MPI-Schnittstelle Ihres PCs.
- Mit **Zielsystem** > *Laden in Baugruppe* in Ihrem Projektierool übertragen Sie Ihr Projekt in die CPU.
- Zur zusätzlichen Sicherung Ihres Projekts auf MMC stecken Sie eine MMC und übertragen Sie mit **Zielsystem** > *RAM nach ROM kopieren* Ihr Anwenderprogramm auf die MMC.  
Während des Schreibvorgangs blinkt die "MC"-LED auf der CPU. Systembedingt wird zu früh ein erfolgter Schreibvorgang gemeldet. Der Schreibvorgang ist erst beendet, wenn die LED erlischt.

Hinweise zur Konfiguration einer MPI-Schnittstelle finden Sie in der Dokumentation zu Ihrer Programmiersoftware.

... Fortsetzung zu a)  
Transfer über MPI

An dieser Stelle soll lediglich der Einsatz des "Green Cable" von VIPA, in Verbindung mit dem Programmierool von Siemens, gezeigt werden.

Das "Green Cable" stellt über MPI eine serielle Verbindung zwischen der COM-Schnittstelle des PCs und der MP<sup>2</sup>I-Schnittstelle der CPU her.



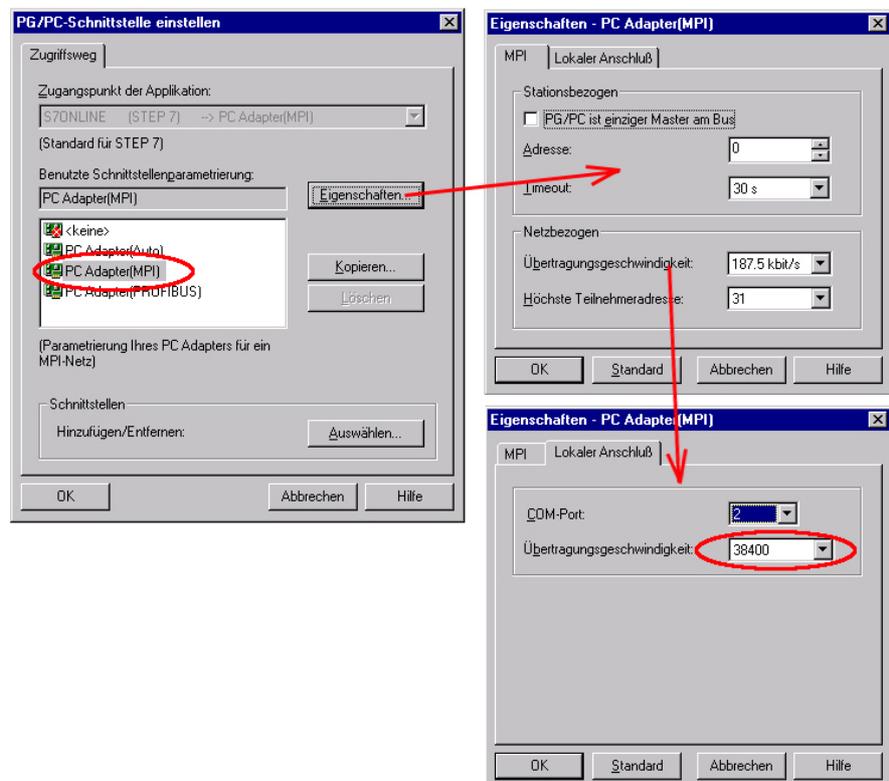
### Achtung!

Bitte beachten Sie, dass Sie das "Green Cable" ausschließlich auf einer MP<sup>2</sup>I-Schnittstelle einer VIPA CPU einsetzen dürfen!

### Vorgehensweise

- Starten Sie den Siemens SIMATIC Manager.
- Wählen Sie unter **Extras** > **PG/PC-Schnittstelle einstellen**.

→ Es öffnet sich folgendes Dialogfenster, in dem Sie die zu verwendende MPI-Schnittstelle konfigurieren können:



- Wählen Sie in der Auswahlliste "PC Adapter (MPI)" aus; ggf. müssen Sie diesen erst hinzufügen.
- Klicken Sie auf [Eigenschaften].

→ In den folgenden 2 Unterdialogen können Sie, wie in der Abbildung gezeigt, Ihren PC-Adapter konfigurieren:



### Hinweis!

Bitte beachten Sie, dass Sie bei Einsatz des Green Cables die Übertragungsgeschwindigkeit auf 38400 Baud einstellen.

zu b)  
Transfer über MMC

Als externes Speichermedium kommt die Multi Media Card (MMC) zum Einsatz (Best.-Nr. VIPA 953-0KX10).

Das Lesen der MMC erfolgt immer nach URLÖSCHEN.

Beschrieben wird die MMC entweder über einen Schreibbefehl aus dem Hardware-Konfigurator von Siemens oder mit einem MMC-Lesegerät von VIPA (Best.-Nr: VIPA 950-0AD00). Somit ist es möglich, Programme am PC zu erstellen, diese auf die MMC zu kopieren und durch Stecken in die VIPA CPU zu übertragen. Die MMC-Module werden mit dem File-System FAT16 vorformatiert von VIPA ausgeliefert.

#### *Erforderliche Dateien*

Es dürfen sich mehrere Projekte und Unterverzeichnisse auf einem MMC-Speichermodul befinden.

Bitte beachten Sie, dass sich Ihre aktuelle Projektierung im Root-Verzeichnis befindet und folgenden Dateinamen hat: **S7PROG.WLD**.

#### *Transfer CPU → MMC*

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

Den Schreibbefehl starten Sie aus dem Hardware-Konfigurator von Siemens über **Zielsystem > RAM nach ROM kopieren**.

Während des Schreibvorgangs blinkt die gelbe "MC"-LED der CPU.

Gleichzeitig erfolgt ein Schreibvorgang in das interne Flash der CPU.

#### *Kontrolle des Transfervorgangs*

Nach einem Schreibvorgang auf die MMC wird ein entsprechendes ID-Ereignis im Diagnosepuffer der CPU eingetragen. Zur Anzeige der Diagnoseeinträge gehen Sie in Ihrem Siemens SIMATIC Manager auf **Zielsystem > Baugruppenzustand**. Über das Register "Diagnosepuffer" gelangen Sie in das Diagnosefenster. Näheres hierzu finden Sie im Handbuch HB97D\_OPList am Ende des Teils "Befehlsliste".

Beim Schreiben auf eine MMC können folgende Ereignisse auftreten:

Ereignis-ID	Bedeutung
0xE100	MMC-Zugriffsfehler
0xE101	MMC-Fehler Filesystem
0xE102	MMC-Fehler FAT
0xE200	MMC schreiben beendet
0xE300	Internes Flash schreiben beendet



#### **Hinweis!**

Bei einem erfolgreichen Schreibvorgang finden Sie 0xE200 und 0xE300 im Diagnosepuffer.

... Fortsetzung zu b)  
Transfer über MMC

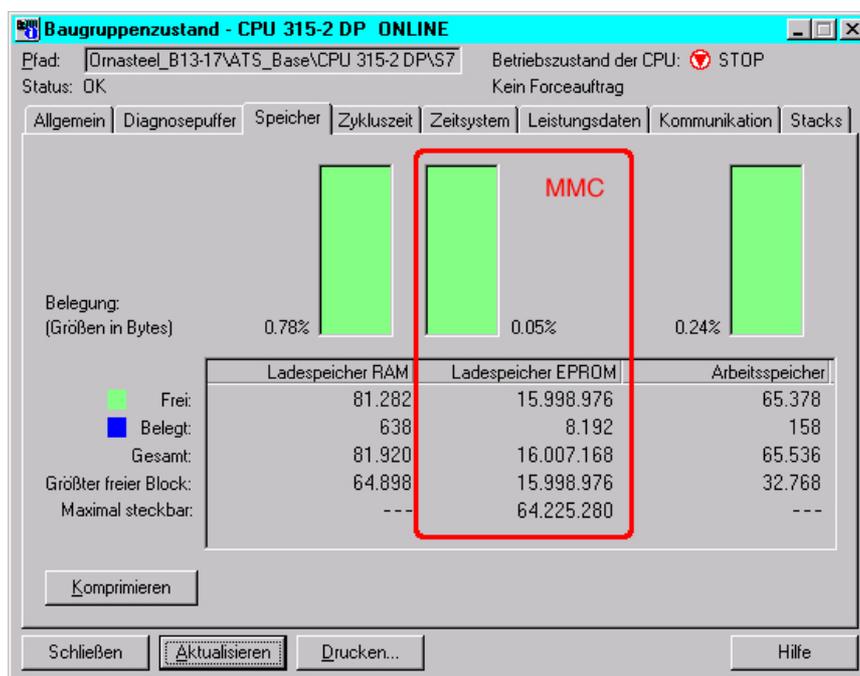
### Transfer MMC → CPU

Das Übertragen des Anwenderprogramms von der MMC in die CPU erfolgt immer nach URLÖSCHEN. Das Blinken der gelben LED "MC" der CPU kennzeichnet den Übertragungsvorgang.

Ist kein gültiges Anwenderprogramm auf der gesteckten MMC oder scheitert die Übertragung, so erfolgt das URLÖSCHEN der CPU und die STOP-LED blinkt dreimal.

### Speichergröße der MMC ausgeben

Zur Anzeige der Speichergröße gehen Sie in Ihrem Siemens SIMATIC Manager auf **Zielsystem** > *Baugruppenzustand*. Über das Register "Speicher" gelangen Sie in das Fenster, das den aktuellen Speicherausbau der CPU zeigt.



### Zugriff auf das interne Flash

Wie schon weiter oben beschrieben, wird durch einen Schreibbefehl der Inhalt des batteriegepufferten RAMs auf eine MMC und in das interne Flash übertragen.

Den Schreibbefehl starten Sie aus dem Hardware-Konfigurator von Siemens über **Zielsystem** > *RAM nach ROM kopieren*.

Lesender Zugriff auf das Flash erfolgt nur bei leerer Pufferbatterie, sofern keine MMC gesteckt ist.



### Hinweis!

Ist das Anwenderprogramm größer als der Anwenderspeicher in der CPU, wird der Inhalt der MMC nicht in die CPU übertragen.

Führen Sie vor der Übertragung eine Komprimierung durch, da keine automatische Komprimierung durchgeführt wird.

**Projektierung  
überprüfen**

- Aktivieren Sie in WinNCS über  die Onlinefunktionen.
- Stellen Sie unter  "IP-Protokoll" ein und geben Sie jetzt die neue IP-Adresse an.
- Stellen Sie mit  eine Verbindung her. Sie kommunizieren jetzt über die unter CP-Init vorgegebene IP-Adresse.
- Der CP muss sich im RUN befinden. Überprüfen Sie dies mit . Befindet sich der CP im *Idle-Mode*, ist die Synchronisation mit der CPU fehlgeschlagen. In diesem Fall ist der SYNCHRON-Baustein in OB 100 zu überprüfen.
- Zur Kontrolle können Sie sich über  den Gesamtstatus der TCP-Verbindungen ausgeben lassen.

Hiermit ist die Projektierung auf CPU- und CP-Seite beendet.

## Beispiel zur Kommunikation CPU 21x-2BT02

### Übersicht

Dieses Kapitel soll in den Umgang mit dem Bussystem TCP/IP für das System 200V einführen. Ziel dieses Kapitels ist es, eine Kommunikation zwischen zwei VIPA CPUs 21x-2BT02 aufzubauen, die auf einfache Weise die Kontrolle der Kommunikationsvorgänge erlaubt.

### Voraussetzungen

Kenntnisse über die CP-Hantierungsbausteine sind erforderlich. CP-Hantierungsbausteine sind Standardfunktionsbausteine. Sie ermöglichen die Nutzung der Kommunikationsfunktionen durch Programme in den Automatisierungsgeräten.

Für die Durchführung des Beispiels sollten Sie mindestens die folgenden technischen Einrichtungen besitzen:

#### *Hardware*

- 2 CPUs 21x-2BT02 von VIPA
- 1 PC oder PG mit Twisted Pair Ethernet-Anschluss

#### *Übertragungsstrecke*

- 3 Buskabel
- 1 Mini-Switch CM 240

#### *Software-Pakete*

- Parametriersoftware WinNCS von VIPA
- Programmierpaket WinPLC7 von VIPA bzw. Siemens SIMATIC Manager für CPU 21xNET

Zur Realisierung des Beispiels ist die Programmierung der zwei CPUs sowie die Parametrierung der Kommunikationsprozessoren unter WinNCS erforderlich.

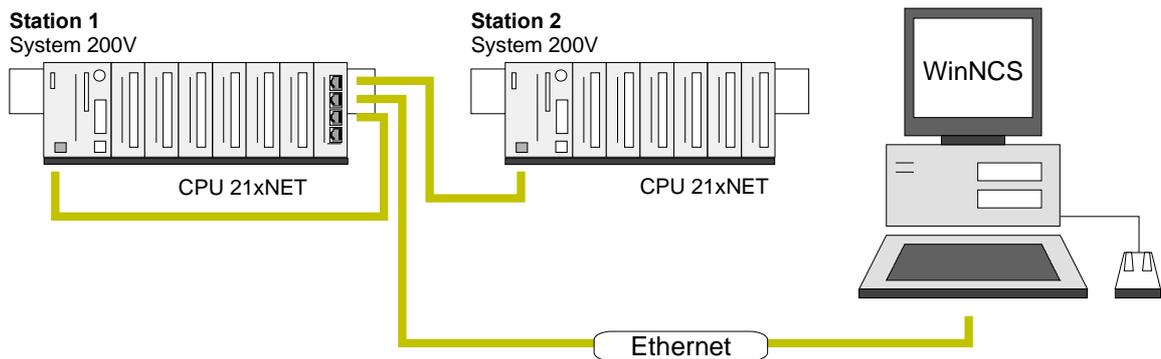


### **Hinweis!**

Das komplette Beispiel befindet sich auf unserem ftp-Server unter: [ftp.vipa.de/support/software/demo\\_files](ftp.vipa.de/support/software/demo_files).

Das SPS-Programm können Sie direkt in beide CPUs übertragen.

## Aufbau



### Aufgaben für die Stationen

Dem Beispiel wird eine Kommunikationsaufgabe zugrunde gelegt, die im Folgenden näher erläutert wird:

In beiden CPUs läuft das gleiche SPS-Programm, lediglich die Projektierung der CP-Teile ist auf die jeweilige Station anzupassen.

Beide Stationen senden und empfangen im Sekundentakt 16 Datenworte.

- Im Datenbaustein DB11 werden die Datenworte DW0 bis DW15 im Takt von 1s übertragen. Das Datenwort DW0 im DB11 dient hierbei als Telegrammzähler. Es wird nur dann inkrementiert, wenn der vorhergegangene Sendeauftrag korrekt (fertig ohne Fehler) abgearbeitet wurde. Über die restlichen Datenworte (DW1 bis DW15) könnten Nutzdaten übertragen werden.
- Die empfangende Station legt die Daten in DB12 ab (DW0 bis DW15).
- SEND wird mit der Auftragsnummer A-Nr. = 1 und dem Kacheloffset SSNR = 0 parametrier.
- RECEIVE wird mit der Auftragsnummer A-Nr. = 11 und dem Kacheloffset SSNR = 0 parametrier.
- Die Quell- und Zielparameter sind direkt zu parametrieren.

Die Aufgabenstellung und die erforderlichen Voreinstellungen sind somit umrissen. Weitere Einzelheiten zur Parametrierung der Hantierungsbausteine gehen aus den Programmen hervor.

Die hierzu passende Parametrierung der CPs unter ISO-Transport (H1) bzw. TCP/IP ist ausführlich beschrieben.

## Projektierung unter WinNCS

Die Projektierung beider CPs findet ausschließlich unter WinNCS statt. Starten Sie WinNCS und legen Sie ein Projekt mit der Funktionsgruppe "Ethernet\_H1" an. Die Vorgehensweise ist bei beiden CPUs die gleiche. Sie unterscheidet sich nur in den einzustellenden Parametern und gliedert sich in folgende 3 Teile:

- CP-Grundprojektierung
- Verbindungsbausteine projektieren
- Projektierdaten in den CP übertragen

## CP-Grundprojektierung

Fügen Sie zwei Stationen ein und stellen Sie folgende Werte ein:

Station 1

Station 2

Die erforderlichen Stationsadressen erhalten Sie ggf. von Ihrem Systemadministrator.

Führen Sie gegebenenfalls noch weitere Einstellungen in den Parametrierfenstern durch. Näheres hierzu erfahren Sie von Ihrem Systemadministrator.

## Verbindungsbausteine projektieren

### ISO-Transport-(H1)-Verbindungen

Zur Projektierung Ihrer ISO-Transport-Verbindungen fügen Sie unterhalb



der Stationen mit Ihre ISO-Transport-Verbindungen ein und geben für die Stationen folgende Parameter ein:

#### Station 1

Senden an Adr.: 0020D5000002

Parameter											
H1-Transport Verbindung   Multiverbindungen   Systemparameter											
Verbindungsname: <input type="text" value="Send zu SPS2"/>											
Kacheloffset: <input type="text" value="0"/>	Auftragsart: <input type="text" value="Send"/>										
Auftragsnummer: <input type="text" value="1"/>											
Priorität: <input type="text" value="2"/>											
<table border="1"> <thead> <tr> <th>Lokaler TSAP</th> <th>Fremder TSAP</th> </tr> </thead> <tbody> <tr> <td>Asc: <input type="text" value="senden"/></td> <td>Asc: <input type="text" value="empfangen"/></td> </tr> <tr> <td>Länge: <input type="text" value="8"/></td> <td>Länge: <input type="text" value="8"/></td> </tr> <tr> <td>Hex: <input type="text" value="73656E64656E2020"/></td> <td>Hex: <input type="text" value="656D7066616E6765"/></td> </tr> <tr> <td></td> <td>Adresse: <input type="text" value="0020D5000002"/></td> </tr> </tbody> </table>		Lokaler TSAP	Fremder TSAP	Asc: <input type="text" value="senden"/>	Asc: <input type="text" value="empfangen"/>	Länge: <input type="text" value="8"/>	Länge: <input type="text" value="8"/>	Hex: <input type="text" value="73656E64656E2020"/>	Hex: <input type="text" value="656D7066616E6765"/>		Adresse: <input type="text" value="0020D5000002"/>
Lokaler TSAP	Fremder TSAP										
Asc: <input type="text" value="senden"/>	Asc: <input type="text" value="empfangen"/>										
Länge: <input type="text" value="8"/>	Länge: <input type="text" value="8"/>										
Hex: <input type="text" value="73656E64656E2020"/>	Hex: <input type="text" value="656D7066616E6765"/>										
	Adresse: <input type="text" value="0020D5000002"/>										
<input type="button" value="Übernehmen"/> <input type="button" value="Verwerfen"/> <input type="button" value="Hilfe"/>											

#### Station 2

Empfang von Station 1

Parameter											
H1-Transport Verbindung   Multiverbindungen   Systemparameter											
Verbindungsname: <input type="text" value="Receive von SPS1"/>											
Kacheloffset: <input type="text" value="0"/>	Auftragsart: <input type="text" value="Receive"/>										
Auftragsnummer: <input type="text" value="11"/>											
Priorität: <input type="text" value="2"/>											
<table border="1"> <thead> <tr> <th>Lokaler TSAP</th> <th>Fremder TSAP</th> </tr> </thead> <tbody> <tr> <td>Asc: <input type="text" value="empfangen"/></td> <td>Asc: <input type="text" value="senden"/></td> </tr> <tr> <td>Länge: <input type="text" value="8"/></td> <td>Länge: <input type="text" value="8"/></td> </tr> <tr> <td>Hex: <input type="text" value="656D7066616E6765"/></td> <td>Hex: <input type="text" value="73656E64656E2020"/></td> </tr> <tr> <td></td> <td>Adresse: <input type="text" value="0020D5000001"/></td> </tr> </tbody> </table>		Lokaler TSAP	Fremder TSAP	Asc: <input type="text" value="empfangen"/>	Asc: <input type="text" value="senden"/>	Länge: <input type="text" value="8"/>	Länge: <input type="text" value="8"/>	Hex: <input type="text" value="656D7066616E6765"/>	Hex: <input type="text" value="73656E64656E2020"/>		Adresse: <input type="text" value="0020D5000001"/>
Lokaler TSAP	Fremder TSAP										
Asc: <input type="text" value="empfangen"/>	Asc: <input type="text" value="senden"/>										
Länge: <input type="text" value="8"/>	Länge: <input type="text" value="8"/>										
Hex: <input type="text" value="656D7066616E6765"/>	Hex: <input type="text" value="73656E64656E2020"/>										
	Adresse: <input type="text" value="0020D5000001"/>										
<input type="button" value="Übernehmen"/> <input type="button" value="Verwerfen"/> <input type="button" value="Hilfe"/>											

#### Empfang von Station 2

Parameter											
H1-Transport Verbindung   Multiverbindungen   Systemparameter											
Verbindungsname: <input type="text" value="Receive von SPS2"/>											
Kacheloffset: <input type="text" value="0"/>	Auftragsart: <input type="text" value="Receive"/>										
Auftragsnummer: <input type="text" value="11"/>											
Priorität: <input type="text" value="2"/>											
<table border="1"> <thead> <tr> <th>Lokaler TSAP</th> <th>Fremder TSAP</th> </tr> </thead> <tbody> <tr> <td>Asc: <input type="text" value="empfangen"/></td> <td>Asc: <input type="text" value="senden"/></td> </tr> <tr> <td>Länge: <input type="text" value="8"/></td> <td>Länge: <input type="text" value="8"/></td> </tr> <tr> <td>Hex: <input type="text" value="656D7066616E6765"/></td> <td>Hex: <input type="text" value="73656E64656E2020"/></td> </tr> <tr> <td></td> <td>Adresse: <input type="text" value="0020D5000002"/></td> </tr> </tbody> </table>		Lokaler TSAP	Fremder TSAP	Asc: <input type="text" value="empfangen"/>	Asc: <input type="text" value="senden"/>	Länge: <input type="text" value="8"/>	Länge: <input type="text" value="8"/>	Hex: <input type="text" value="656D7066616E6765"/>	Hex: <input type="text" value="73656E64656E2020"/>		Adresse: <input type="text" value="0020D5000002"/>
Lokaler TSAP	Fremder TSAP										
Asc: <input type="text" value="empfangen"/>	Asc: <input type="text" value="senden"/>										
Länge: <input type="text" value="8"/>	Länge: <input type="text" value="8"/>										
Hex: <input type="text" value="656D7066616E6765"/>	Hex: <input type="text" value="73656E64656E2020"/>										
	Adresse: <input type="text" value="0020D5000002"/>										
<input type="button" value="Übernehmen"/> <input type="button" value="Verwerfen"/> <input type="button" value="Hilfe"/>											

#### Senden an Adr.: 0020D5000001

Parameter											
H1-Transport Verbindung   Multiverbindungen   Systemparameter											
Verbindungsname: <input type="text" value="Send zu SPS1"/>											
Kacheloffset: <input type="text" value="0"/>	Auftragsart: <input type="text" value="Send"/>										
Auftragsnummer: <input type="text" value="1"/>											
Priorität: <input type="text" value="2"/>											
<table border="1"> <thead> <tr> <th>Lokaler TSAP</th> <th>Fremder TSAP</th> </tr> </thead> <tbody> <tr> <td>Asc: <input type="text" value="senden"/></td> <td>Asc: <input type="text" value="empfangen"/></td> </tr> <tr> <td>Länge: <input type="text" value="8"/></td> <td>Länge: <input type="text" value="8"/></td> </tr> <tr> <td>Hex: <input type="text" value="73656E64656E2020"/></td> <td>Hex: <input type="text" value="656D7066616E6765"/></td> </tr> <tr> <td></td> <td>Adresse: <input type="text" value="0020D5000001"/></td> </tr> </tbody> </table>		Lokaler TSAP	Fremder TSAP	Asc: <input type="text" value="senden"/>	Asc: <input type="text" value="empfangen"/>	Länge: <input type="text" value="8"/>	Länge: <input type="text" value="8"/>	Hex: <input type="text" value="73656E64656E2020"/>	Hex: <input type="text" value="656D7066616E6765"/>		Adresse: <input type="text" value="0020D5000001"/>
Lokaler TSAP	Fremder TSAP										
Asc: <input type="text" value="senden"/>	Asc: <input type="text" value="empfangen"/>										
Länge: <input type="text" value="8"/>	Länge: <input type="text" value="8"/>										
Hex: <input type="text" value="73656E64656E2020"/>	Hex: <input type="text" value="656D7066616E6765"/>										
	Adresse: <input type="text" value="0020D5000001"/>										
<input type="button" value="Übernehmen"/> <input type="button" value="Verwerfen"/> <input type="button" value="Hilfe"/>											

**TCP/IP-Verbindungen**

Zur Projektierung Ihrer TCP/IP-Verbindungen fügen Sie unterhalb der



Stationen mit Ihre TCP-Verbindungen ein und geben für die Stationen folgende Parameter ein:

**Station 1**

Senden an IP: 172.16.129.149

Parameter	
ICP Verbindung   <b>Multiverbindung</b>   Systemparameter	
Verbindungsname: <input type="text" value="Send zu SPS2"/>	
Kachelloffset: <input type="text" value="0"/>	Auftragsart: <input type="text" value="Send"/>
Auftragsnummer: <input type="text" value="1"/>	Auftragstyp: <input type="text" value="Einzelauftrag"/>
Priorität: <input type="text" value="2"/>	
<b>Lokale Station :</b> Port: <input type="text" value="0"/>	<b>Fremde Station :</b> Port: <input type="text" value="3000"/>
<input type="checkbox"/> IP-Adr: <input type="text" value="172.16.129.149"/>	
Host-Name: <input type="text"/>	
Versuche: <input type="text" value="0"/>	
<input type="button" value="Übernehmen"/> <input type="button" value="Verwerfen"/> <input type="button" value="Hilfe"/>	

**Station 2**

Empfang von Station 1

Parameter	
ICP Verbindung   <b>Multiverbindung</b>   Systemparameter	
Verbindungsname: <input type="text" value="Receive von SPS1"/>	
Kachelloffset: <input type="text" value="0"/>	Auftragsart: <input type="text" value="Receive"/>
Auftragsnummer: <input type="text" value="11"/>	Auftragstyp: <input type="text" value="Einzelauftrag"/>
Priorität: <input type="text" value="2"/>	
<b>Lokale Station :</b> Port: <input type="text" value="3000"/>	<b>Fremde Station :</b> Port: <input type="text" value="0"/>
<input type="checkbox"/> IP-Adr: <input type="text" value="0.0.0.0"/>	
Host-Name: <input type="text"/>	
Versuche: <input type="text" value="0"/>	
<input type="button" value="Übernehmen"/> <input type="button" value="Verwerfen"/> <input type="button" value="Hilfe"/>	

**Empfang von Station 2**

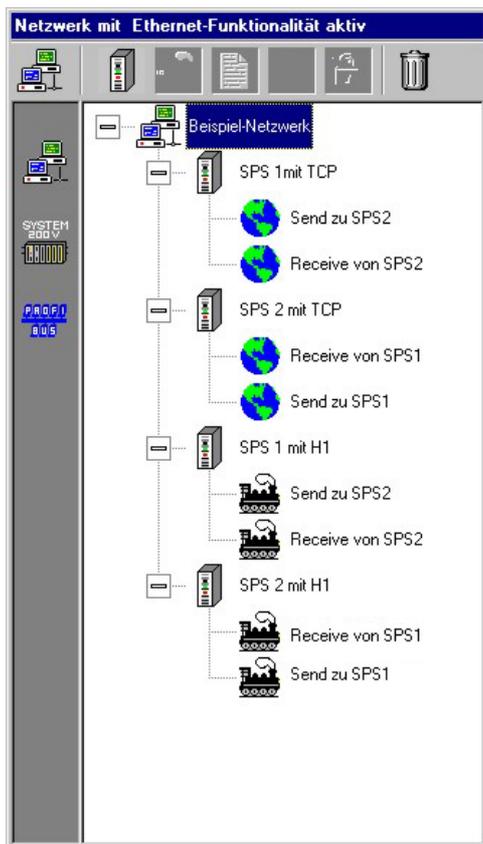
Parameter	
ICP Verbindung   <b>Multiverbindung</b>   Systemparameter	
Verbindungsname: <input type="text" value="Receive von SPS2"/>	
Kachelloffset: <input type="text" value="0"/>	Auftragsart: <input type="text" value="Receive"/>
Auftragsnummer: <input type="text" value="11"/>	Auftragstyp: <input type="text" value="Einzelauftrag"/>
Priorität: <input type="text" value="2"/>	
<b>Lokale Station :</b> Port: <input type="text" value="3001"/>	<b>Fremde Station :</b> Port: <input type="text" value="0"/>
<input type="checkbox"/> IP-Adr: <input type="text" value="0.0.0.0"/>	
Host-Name: <input type="text"/>	
Versuche: <input type="text" value="0"/>	
<input type="button" value="Übernehmen"/> <input type="button" value="Verwerfen"/> <input type="button" value="Hilfe"/>	

**Senden an IP: 172.16.129.148**

Parameter	
ICP Verbindung   <b>Multiverbindung</b>   Systemparameter	
Verbindungsname: <input type="text" value="Send zu SPS1"/>	
Kachelloffset: <input type="text" value="0"/>	Auftragsart: <input type="text" value="Send"/>
Auftragsnummer: <input type="text" value="1"/>	Auftragstyp: <input type="text" value="Einzelauftrag"/>
Priorität: <input type="text" value="2"/>	
<b>Lokale Station :</b> Port: <input type="text" value="0"/>	<b>Fremde Station :</b> Port: <input type="text" value="3001"/>
<input type="checkbox"/> IP-Adr: <input type="text" value="172.16.129.148"/>	
Host-Name: <input type="text"/>	
Versuche: <input type="text" value="0"/>	
<input type="button" value="Übernehmen"/> <input type="button" value="Verwerfen"/> <input type="button" value="Hilfe"/>	

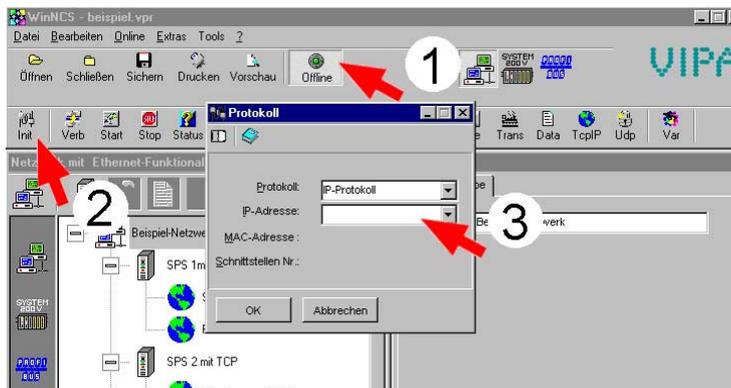
Speichern Sie Ihr Projekt!

**Netzwerkfenster** Ihr Netzwerkfenster sollte folgenden Inhalt zeigen:



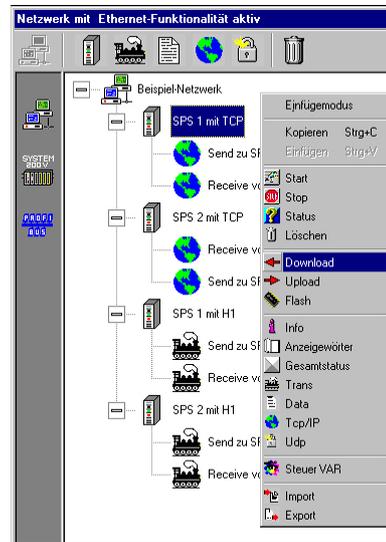
**Projektierdaten in die CPUs übertragen**

Sie können online über das Netzwerk Ihre Projektierung in die zwei CPUs übertragen. Bauen Sie hierzu, wie weiter oben gezeigt, die Anlagenstruktur auf und fahren Sie beide CPUs hoch. Zur Übertragung der Projektierdaten aktivieren Sie die Online-Funktionen und klicken Sie auf INIT:



Stellen Sie im *Protokoll*-Fenster "IP-Protokoll" ein und tragen Sie die entsprechende IP-Adresse ein. Bestätigen Sie Ihre Eingabe mit [OK].

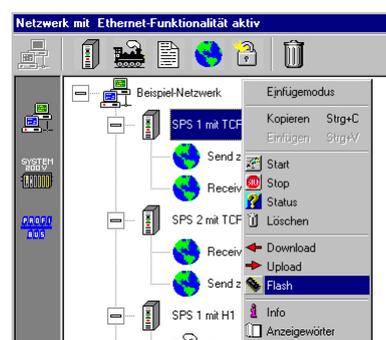
Klicken Sie im *Netzwerk*-Fenster auf die entsprechende Station. Betätigen Sie die rechte Maustaste und wählen Sie den "Download"-Befehl. Ihr Projekt wird nun in das RAM der CPU übertragen.



### Achtung!

Nach der Übertragung müssen Sie mit dem "Flash"-Befehl die RAM-Inhalte in das Flash-ROM transferieren. Ansonsten werden bei Ausschalten der Spannungsversorgung Ihrer CPU die Daten wieder gelöscht.

Klicken Sie hierzu im *Netzwerk*-Fenster auf die entsprechende Station. Betätigen Sie die rechte Maustaste und wählen Sie den "Flash"-Befehl. Die RAM-Inhalte werden in das Flash-ROM transferiert.



Verfahren Sie auf die gleiche Weise mit der 2. CPU, indem Sie unter INIT die entsprechende IP-Adresse einstellen, die Station im Netzwerk-Fenster anwählen, mit Download Ihr Projekt übertragen und mit dem "Flash"-Befehl im Flash-ROM ablegen.

Die Projektierung auf CP-Seite ist jetzt abgeschlossen. Auf der Folgeseite finden Sie die Programmierung der SPS-Seite.

## SPS-Programme für die CPUs

Die SPS-Programmierung ist in diesem Beispiel protokollunabhängig und kann für ISO-Transport und TCP/IP verwendet werden.

Das SPS-Programm kommt in beiden CPUs zum Einsatz.

## OB100 Synchronisation der Schnittstellen

### **Synchronisation der Schnittstellen**

Im Anlauf-OB OB100 der CPU ist die benutzte Schnittstelle des CPs mittels des Hantierungsbausteins SYNCHRON zu synchronisieren.

Im OB100 wird geprüft, ob die Synchronisation fehlerfrei durchgeführt wurde. Trat ein Fehler auf, wird ein Parametrierfehler in Merkerbyte 200. eingetragen.

Operationsbaustein OB100:

```

OB100 : Titel:
Kommentar:

Netzwerk 1 : Titel:
Kommentar:

      L      S5T#20S           // Timerwert 20 sec
      CLR                               // VKE 0
      SV     T      1
      SET                               // VKE 1
      SV     T      1           // Timer Start

loop: CALL  "Synchron"         // CP synchronisieren
      SSMR:=0                   // Schnittstellennr. 0
      BLGR:=6                   // Blockgröße 6 : 512 Byte
      PAFE:=MB200               // Parametrierfehlerbyte MB200

      UN     M      200.0       // kein Fehler -> fertig
      BEB
      U     T      1           // Timer läuft noch
      SPB   loop               // Synchron nochmal aufrufen

```

**OB1 - Zyklus**  
**FC1 - SEND**  
**FC2 - RECEIVE**

Über den Zyklus-OB OB1 wird das Senden und Empfangen der Daten gesteuert. Innerhalb des OBs erfolgt ein Sendeanstoß über den FC1. Die Gegenstation antwortet mit RECEIVE (FC2). Daraufhin werden die Daten mit SEND\_ALL übermittelt und in der Gegenstation mit RECEIVE\_ALL empfangen.

**Zyklus-Operationsbaustein OB1:**

OB1 : Zyklus	
Send-All, Rec-All, Applikation	
<b>Netzwerk 1: All Bausteine</b>	
Send- und Rec-All	
CALL "Receive_All"	// übernimmt Daten vom CP
SSNR:=0	// wird auch vom Write passiv benötigt
PAFE:=MB196	// Parametrierfehlerbyte im MB196
ANZW:=MD180	// Anzeigewort vom All-Baustein : enthält
//	die Nummer des momentan bearbeiteten Auftrags
CALL "Send_All"	// übergibt Daten an den CP
SSNR:=0	// ist auch für Fetch passiv erforderlich
PAFE:=MB197	// Parametrierfehlerbyte im MB197
ANZW:=MD184	// Anzeigewort vom All-Baustein : enthält
//	die Nummer des momentan bearbeiteten Auftrags
<b>Netzwerk 2: Titel:</b>	
Senden	
CALL FC 1	// hier wird das Senden ausgeführt
<b>Netzwerk 3: Titel:</b>	
Empfangen	
CALL FC 2	// Empfangsbaustein

**FC1 - SEND**

FC1 : Senden	
Sendebaustein	
<b>Netzwerk 1: Titel:</b>	
timergesteuertes Senden	
CALL "Control"	//control sendeauftrag
SSNR:=0	// schnittstellennr. 0
AMR :=1	// auftragmr. 1
PAFE:=MB195	//parametrierfehlerbyte MB195
ANZW:=MD174	//anzeigewort im mdl74
0 M 175.1	//auftrag läuft
0 T 1	//oder timer läuft
BEB	//ende
L SST#15	// timerwert 1 sekunde
CLR	// vke 0
SV T 1	// vke 1
SET	// Timerstart
SV T 1	// Timerstart
CALL "Send"	// sendeauftrag
SSNR:=0	// schnittstellennr. 0
AMR :=1	// auftragmr. 1
IND :=0	// sendeart : 0 send-direkt
QANF:=P#DB11.DEX 0.0 BYTE 16	// sendedaten : db11, ab dw0, anzahl 16 byte
PAFE:=MB196	//parametrierfehlerbyte MB196
ANZW:=MD174	//anzeigewort im mdl74
L DB11.DEW 0	// telegrammzähler
+ 1	// im db11,dw0
T DB11.DEW 0	// um eins erhöhen

**FC2 - RECEIVE**

FC2 : Empfang	
Empfangsbaustein	
<b>Netzwerk 1: Titel:</b>	
zyklischer Aufruf	
CALL "Control"	// SFC233
SSNR:=0	// Schnittstellen(Kachel)-Nr. 0
AMR :=11	// Auftrag-Nr. 11
PAFE:=MB195	// Parametrierfehlerbyte im MB195
ANZW:=MD184	// Anzeigewort im MD184
UN M 185.0	// wenn Handshake nicht sinnvoll
BEB	// dann Ende
CALL "Receive"	// SFC231
SSNR:=0	// Schnittstellen(Kachel)-Nr. 0
AMR :=11	// Auftrag-Nr. 12
IND :=0	// Adressierart
ZANF:=P#DB12.DEX0.0 BYTE 16	// Parametrierfehlerbyte im MB195
PAFE:=MB195	// Anzeigewort im MD184
ANZW:=MD184	// Anzeigewort im MD184
L MB 188	// Telegrammzähler um eins erhöhen
+ 1	
T MB 188	

**Datenbausteine  
DB11, DB12**

Für die Anstoßhäufigkeit eines Send-Auftrags ist der parametrisierte Zeitwert beim FC1-Aufruf maßgebend. Der Timer ist in diesem Beispiel auf 1000ms programmiert. Damit stößt das Beispielprogramm den Send-Auftrag im 1000ms-Takt an.

Vor jedem SEND-Aufruf, der tatsächlich ein Telegramm absendet, wird das Datenwort DW0 des Datenbausteins DB11 inkrementiert. Dies geschieht im Funktionsbaustein FC1. Es werden insgesamt 16Byte Daten übertragen. Die Gegenseite empfängt die Daten und legt diese in DB12 ab.

Zusammen mit DW0 können noch 15Byte Nutzdaten übertragen werden.

Die Datenbausteine DB11 und DB12 haben den gleichen Aufbau:

Adresse	Name	Typ	Anfangswert	Kommentar
0.0		STRUCT		
+0.0	STAT0	BYTE	B#16#0	
+1.0	STAT1	BYTE	B#16#0	
+2.0	STAT2	BYTE	B#16#0	
+3.0	STAT3	BYTE	B#16#0	
+4.0	STAT4	BYTE	B#16#0	
+5.0	STAT5	BYTE	B#16#0	
+6.0	STAT6	BYTE	B#16#0	
+7.0	STAT7	BYTE	B#16#0	
+8.0	STAT8	BYTE	B#16#0	
+9.0	STAT9	BYTE	B#16#0	
+10.0	STAT10	BYTE	B#16#0	
+11.0	STAT11	BYTE	B#16#0	
+12.0	STAT12	BYTE	B#16#0	
+13.0	STAT13	BYTE	B#16#0	
+14.0	STAT14	BYTE	B#16#0	
+15.0	STAT15	BYTE	B#16#0	
+16.0	STAT16	BYTE	B#16#0	
=18.0		END_STRUCT		

**Programm  
übertragen**

Die Datenübertragung erfolgt über MPI. Sollte Ihr Programmiergerät keine MPI-Schnittstelle besitzen, können Sie für eine serielle Punkt-zu-Punkt-Übertragung von Ihrem PC an MPI das "Green Cable" (VIPA 950-0KB00) von VIPA verwenden.

Das "Green Cable" darf nur bei den VIPA CPUs der Systeme 100V, 200V, 300V und 500V eingesetzt werden.

Näheres hierzu siehe Teil "Einsatz CPU 21x" unter "Projekt transferieren".

- Verbinden Sie Ihr PG mit der CPU.
- Mit **Zielsystem** > *Laden in Baugruppe* in Ihrem Projektierool übertragen Sie Ihr Projekt in die CPU.
- Stecken Sie eine MMC und übertragen Sie mit **Zielsystem** > *RAM nach ROM kopieren* Ihr Anwenderprogramm auf die MMC.
- Während des Schreibvorgangs blinkt die "MC"-LED auf der CPU. Systembedingt wird zu früh ein erfolgter Schreibvorgang gemeldet. Der Schreibvorgang ist erst beendet, wenn die LED erlischt.
- Bringen Sie beide CPUs in RUN.

### Beobachtung der Übertragung im Siemens SIMATIC Manager

Als Ausgangspunkt werden parametrierte CPs und urgelöschte CPUs, deren RUN/STOP-Schalter in der Grundstellung STOP steht, vorausgesetzt.

Laden Sie die zuvor beschriebene SPS-Programme in Ihre CPUs und bringen Sie diese in RUN.

Die Kommunikation zwischen den beiden Stationen ist aufgebaut. Angezeigt wird dies durch die Kommunikations-LEDs.

Starten Sie den Siemens SIMATIC Manager und führen Sie zur Beobachtung des Sendeauftrags die folgenden Schritte aus:

- **Zielsystem** > *Variable beobachten/steuern*
- Tragen Sie unter "Operand" die entsprechende Datenbaustein-Nr. und das Datenwort ein (DB11.DW0-15).

- Stellen Sie eine Verbindung her und klicken Sie auf "beobachten"



Operand	Symbol	Statusformat	Statuswert	Steuerwert
MW 10	---	HEX		
DB11.DBW 0	---	HEX		
DB11.DBW 2	---	HEX		
DB11.DBW 4	---	HEX		
DB11.DBW 6	---	HEX		
DB11.DBW 8	---	HEX		
DB11.DBW 10	---	HEX		
DB11.DBW 12	---	HEX		
DB11.DBW 14	---	HEX		
DB11.DBW 16	---	HEX		
DB11.DBW 18	---	HEX		
DB11.DBW 20	---	HEX		
DB11.DBW 22	---	HEX		
DB11.DBW 24	---	HEX		
DB11.DBW 26	---	HEX		
DB11.DBW 28	---	HEX		
DB11.DBW 30	---	HEX		

### Nutzdaten eingeben

Ab DW1 können Nutzdaten eingetragen werden. Gehen Sie hierzu mit dem Cursor auf *Steuerwert* und tragen Sie einen zu übertragenden Wert ein wie z.B. W#16#1111.

Mit  übertragen Sie den Steuerwert bei jedem Zyklusdurchlauf bzw.

mit  einmalig.

## Anlaufverhalten

### Übersicht

Nach dem Einschalten der Stromversorgung durchlaufen CPU und CP ihre BIOS-Routinen (Hardware-, Treiberinitialisierung und Speichertest).

Während die CPU die Module am Rückwandbus ermittelt und das Anwenderprogramm lädt, beginnt der CP-Teil mit der Kachelverwaltung.

Nach ca. 15s wartet der CP auf die Synchronisation mit der CPU. In diesem Zustand ist der Datenverkehr mit der CPU gesperrt und wird erst mit der Synchronisation freigegeben.

Die Hochlaufzeit der CPU 21x-2BT02 einschließlich CP-Teil beträgt ca. 18s.

### Status nach CP-Anlauf

Die CPU 21x-2BT02 vollzieht bei jedem Zustandswechsel von STOP nach RUN, sowie von RUN über STOP nach RUN einen Kalt-/Warmstart. Alle bisher aufgebauten Verbindungen werden gelöscht und nach dem Boot des CP-Job erneut aufgebaut.

Solche Zustandswechsel-Anforderungen können drei Quellen haben:

- Neusynchronisation eines bereits synchronen CPs durch die SYNCHRON-HTBs der CPU (Warmstart)
- STOP/START-Funktion des Parametrierwerkzeugs WinNCS (Warmstart)
- RESET\_ALL (Warmstart)



### Hinweis!

Sobald in den CP eine fehlerhafte Projektierung übertragen wurde, sodass dieser nicht anlaufen kann, wird die Projektierung gelöscht und eine Default-Projektierung verwendet. Hier bekommt der CP wieder seine ursprüngliche aufgedruckte IP-Adresse.

Auf diese Weise können Sie den CP unabhängig von der CPU urlöschen.

## Systemeigenschaften der CPU 21x-2BT02

- Hinweis** Systemeigenschaften eines CPs sind nicht als Einschränkungen zu sehen bzw. einem Fehlverhalten gleichzustellen, vielmehr sind gewisse Funktionalitäten nicht zu erreichen oder aus Sicht des Gesamtsystems gewollt.
- Allgemein** Die Hochlaufzeit des CP-Teils der CPU 21x-2BT02 beträgt ca. 18 Sekunden. Diese Hochlaufzeit wird im integrierten SYNCHRON-Baustein berücksichtigt (Wartezeit 30s).
- Nur bei ISO-Transport (H1) beachten**
- Aufträge der Priorität 0/1 können maximal so viele Daten senden und/oder empfangen, wie dies durch den SYNCHRON-HTB definiert wurde. Die Aufträge dieser Priorität werden nicht geblockt. Daraus ergibt sich eine maximale Daten-Transferrate von 512Byte pro Auftrag bei einer Blockgröße von 255 (siehe hierzu auch Blockgröße).
  - RECEIVE-Aufträge, die auf die Kommunikationsart "Broadcast" abgebildet werden, können bei einem schnellen zyklischen Sender nicht alle Datentelegramme empfangen. Die nicht empfangenen Telegramme werden verworfen.
- Nur bei TCP/IP beachten**
- Die Jokerlänge (-1, 0xFFFF) ist bei der ORG-Format-Längenangabe nicht zulässig, d.h. der Anwender muss eine genaue Länge seiner zu empfangenen Daten definieren.
  - Aufträge der Priorität 1 können maximal so viele Daten senden und/oder empfangen, wie dies durch den SYNCHRON-HTB definiert wurde. Die Aufträge dieser Priorität werden nicht geblockt. Daraus ergibt sich eine maximale Daten-Transferrate von 512Byte pro Auftrag bei einer Blockgröße von 255 (siehe hierzu auch Blockgröße).
  - RECEIVE-Aufträge, die auf die Kommunikationsart UDP abgebildet werden, können bei einem schnellen zyklischen Sender nicht alle Datentelegramme empfangen. Die nicht empfangenen Telegramme werden verworfen.

Der Protokollstack TCP/IP besitzt einen globalen Pufferpool, in dem sich die Empfangs- und Sende-Puffer befinden. Hier kann es zu Systemkollisionen kommen wenn:

- Daten für einen Empfangsauftrag nicht abgeholt werden. Nach geraumer Zeit werden die Ressourcen knapp und die anderen Verbindungen führen nach einer definierten Zeit Verbindungsabbrüche durch.  
Eine ausgewogene Kommunikation kann allerdings erst wieder aufgenommen werden, wenn die Empfangspuffer der einen Verbindung freigegeben (Verbindungsabbruch) bzw. mit den RECEIVE-HTBs die Daten abgeholt wurden.
- Ein oder mehrere zyklische Sender einen CP belasten. Es werden bei Ressourcen-Engpässen ebenfalls Verbindungsabbrüche vom CP initiiert.
- Ein Sender zwei oder mehrere Telegramme sendet und der Empfänger hatte noch nicht die Möglichkeit es abzuholen, so würde es bei einem dem Empfänger unbekanntem Datentyp im Empfänger zu Datenkollisionen kommen. Dies wird aber auf der CP-Seite verhindert. Die SPS-Applikation fordert eine definierte Empfangsgröße, wobei die Jokerlänge nicht zulässig ist. Die Empfangsgrößen der Prio1 - RECEIVE-Aufträge ist implizit durch die eingestellte Blockgröße definiert (16, 32, 64, 128, 256, 512Byte).
- Zur Gewährleistung einer hundertprozentigen Datenübertragungssicherheit, empfiehlt VIPA bei der Datenübertragung Quittungstelegramme auf Anwender Ebene einzusetzen.
- Bitte beachten Sie, dass der **Port 7777** von WinNCS zur Kommunikation verwendet wird. Dieser darf von anderen Anwender-Applikationen nicht belegt sein!

## Kopplung mit Fremdsystemen

### ORG-Format

Das Organisationsformat ist die Kurzbeschreibung einer Datenquelle bzw. eines Datenziels in SPS-Umgebung. Die verwendbaren ORG-Formate sind in der nachfolgenden Tabelle aufgelistet.

Der ORG-Block ist bei READ und WRITE optional.

Die ERW-Kennung ist bei der Adressierung von Datenbausteinen relevant. In diesem Fall wird hier die Datenbaustein-Nummer eingetragen. Die Anfangsadresse und Anzahl adressieren den Speicherbereich und sind im HIGH-/LOW-Format abgelegt (Motorola - Adressformat)

Beschreibung	Typ	Bereich
ORG-Kennung	BYTE	1..x
ERW-Kennung	BYTE	1..255
Anfangsadresse	HILOWORD	0..y
Länge	HILOWORD	1..z

In der nachfolgenden Tabelle sind die verwendbaren ORG-Formate aufgelistet:

### ORG-Kennung 01h-04h

CPU-Bereich	DB	MB	EB	AB
ORG-Kennung	01h	02h	03h	04h
Beschreibung	Quell-/Zieldaten aus/in Datenbaustein im Hauptspeicher.	Quell-/Zieldaten aus/in Merkerbereich.	Quell-/Zieldaten aus/in Prozessabbild der Eingänge (PAE).	Quell-/Zieldaten aus/in Prozessabbild der Ausgänge (PAA).
ERW-Kennung (DBNR)	DB, aus dem die Quelldaten entnommen werden bzw. in den die Zieldaten transferiert werden.	irrelevant	irrelevant	irrelevant
erlaubter Bereich:	1...255			
Anfangsadresse Bedeutung	DB-Nr., ab der die Daten entnommen bzw. eingeschrieben werden.	MB-Nr., ab der die Daten entnommen bzw. eingeschrieben werden.	EB-Nr., ab der die Daten entnommen bzw. eingeschrieben werden.	AB-Nr., ab der die Daten entnommen bzw. eingeschrieben werden.
erlaubter Bereich:	0...2047	0...255	0...127	0...127
Länge Bedeutung	Länge des Quell-/Zieldatenblocks in Worten.	Länge des Quell-/Zieldatenblocks in Bytes.	Länge des Quell-/Zieldatenblocks in Bytes.	Länge des Quell-/Zieldatenblocks in Bytes.
erlaubter Bereich:	1...2048	1...256	1...128	1...128

**Aufbau SPS-Header**

Bei READ und WRITE generiert der CP SPS-Header für Anforderungs- und Quittungstelegramme. Diese Header sind in der Regel 16Byte lang und haben folgende Struktur:

**bei WRITE**

Anforderungstelegramm

Systemkennung	= "S"
	= "5"
Länge Header	=16d
Kenn. OP-Code	=01
Länge OP-Code	=03
<b>OP-Code</b>	<b>=03</b>
ORG-Block	=03
Länge ORG-Block	=08
<i>ORG-Kennung</i>	
<i>ERW-Kennung</i>	
Anfangsadresse	H
	L
Länge	H
	L
Leerblock	=FFh
Länge Leerbbl.	=02
Daten bis zu 64K jedoch nur wenn Fehler-Nr.=0	

Quittungstelegramm

Systemkennung	= "S"
	= "5"
Länge Header	=16d
Kenn. OP-Code	=01
Länge OP-Code	=03
<b>OP-Code</b>	<b>=04</b>
Quittungsblock	=0Fh
Länge Q-Block	=03
Fehler Nr.	=Nr.
Leerblock	=FFh
Länge Leerbblock	=07
	frei

**bei READ**

Anforderungstelegramm

Systemkennung	= "S"
	= "5"
Länge Header	=16d
Kenn. OP-Code	=01
Länge OP-Code	=03
<b>OP-Code</b>	<b>=05</b>
ORG-Block	=03
Länge ORG-Block	=08
<i>ORG-Kennung</i>	
<i>ERW-Kennung</i>	
Anfangsadresse	H
	L
Länge	H
	L
Leerblock	=FFh
Länge Leerbbl.	=02

Quittungstelegramm

Systemkennung	= "S"
	= "5"
Länge Header	=16d
Kenn. OP-Code	=01
Länge OP-Code	=03
<b>OP-Code</b>	<b>=06</b>
Quittungsblock	=0Fh
Länge Q-Block	=03
Fehler Nr.	=Nr.
Leerblock	=FFh
Länge Leerbblock	=07
	frei
Daten bis zu 64K jedoch nur wenn Fehler-Nr.=0	

**SEND / RECEIVE  
vom Typ TRADA**

TRADA steht für **Transparenter Datenaustausch**. Beim transparenten Datenaustausch können Nutzdaten mit unterschiedlicher Länge übertragen werden. Den zu übertragenden Nutzdaten wird ein 16Byte-Header vorangestellt, der die Länge der Nutzdaten angibt.

Bei Einsatz von TRADA ist im SPS-Anwenderprogramm die Angabe einer Jokerlänge bei der Längenangabe zulässig.

Durch Eingabe von -1 als Längenangabe im RECEIVE-FB (Parameter: ZLAE) stellen Sie eine variable Nutzdatenlänge ein (Jokerlänge). Bei Einsatz einer Jokerlänge wird die jeweilige Datenlänge aus dem TRADA-Header übernommen.

Der nachfolgend beschriebene Header wird bei TRADA-Funktionalität einem SEND-Auftrag vorangestellt und von RECEIVE ausgewertet.

SEND vom Typ TRADA  
OP-Code = 07

Systemkennung	= "S"
	= "5"
Länge Header	=16d
Kenn. OP-Code	=01
Länge OP-Code	=03
<b>OP-Code</b>	<b>=07</b>
ORG-Block	=03
Länge ORG-Block	=08
<i>ORG-Kennung</i>	
<i>ERW-Kennung (irrelevant)</i>	
<i>Anfangsadresse</i>	H
	L
<b>Länge</b>	<b>H</b>
	<b>L</b>
Leerblock	=FFh
Länge Leerbbl.	=02
Daten bis zu 64K jedoch nur wenn Fehler-Nr.=0	

→ Längenangabe der Nutzdaten

**Längenangabe**

In der Längenangabe steht die Anzahl der Bytes eines Datenblocks.

Wird mit einer Blockgröße von 6 synchronisiert (512Byte), erfolgt die Längenangabe in Worten.

## Testprogramm für TCP/IP-Verbindungen

### Übersicht

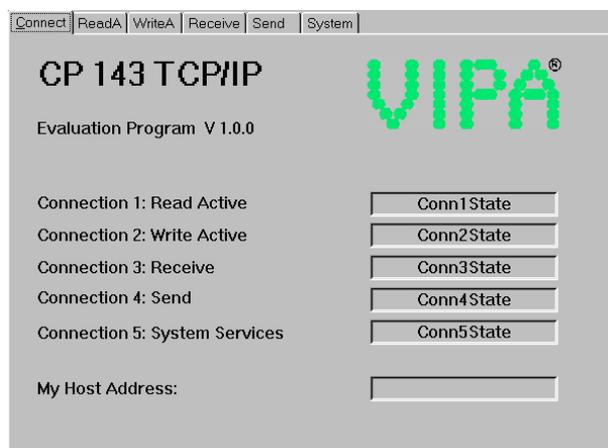
Auf dem ftp-Server von VIPA finden Sie unter ftp.vipa.de/support/software das Programm TCPTTest.exe. Mit diesem Testprogramm können Sie einfache TCP/IP-Verbindungen aufbauen und analysieren.

TCPTTest bedarf keiner weiteren Installation, ist auf allen gängigen Betriebssystemen ablauffähig und kommuniziert über Ethernet.

Der Einsatz dieses Testprogramms soll hier kurz gezeigt werden.

Starten Sie hierzu TCPTTEST.EXE. Das Testprogramm wird aufgerufen und meldet sich mit folgendem Fenster:

### Startbildschirm



### Registerblätter

Das Menü ist in Form von Registerblättern aufgebaut. Durch Anklicken mit der linken Maustaste kommt das ausgewählte Dialogfenster in den Vordergrund.

#### Registerblätter

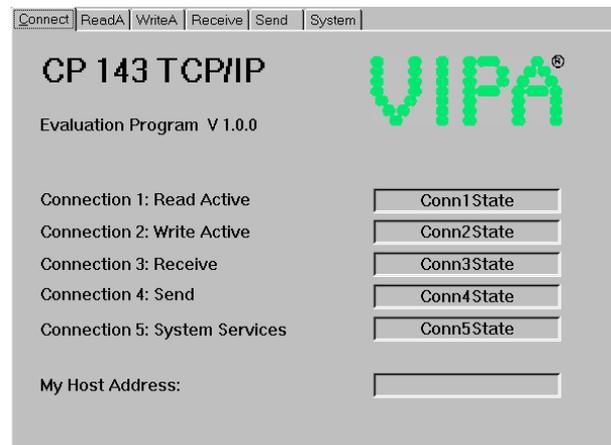
<i>Connect</i>	Fenster mit Statusanzeige der Verbindungen und der eigenen IP-Adresse
<i>ReadA</i>	Parametrierfenster für READ AKTIV-Verbindung (FETCH)
<i>WriteA</i>	Parametrierfenster für WRITE AKTIV-Verbindung
<i>Receive</i>	Parametrierfenster für RECEIVE-Auftrag
<i>Send</i>	Parametrierfenster für SEND-Auftrag
<i>System</i>	Steuerfenster für Statusabfrage und RUN/STOP-Schaltung des CPs

**Kontextmenü  
(rechte Maustaste)**

In jedem Registerblatt haben Sie die Möglichkeit ein Kontextmenü zu aktivieren. Die Aktivierung erfolgt mit der rechten Maustaste.

Durch Betätigung der rechten Maustaste haben Sie immer Zugriff auf ein Kontextmenü, das folgende Auswahl bietet:

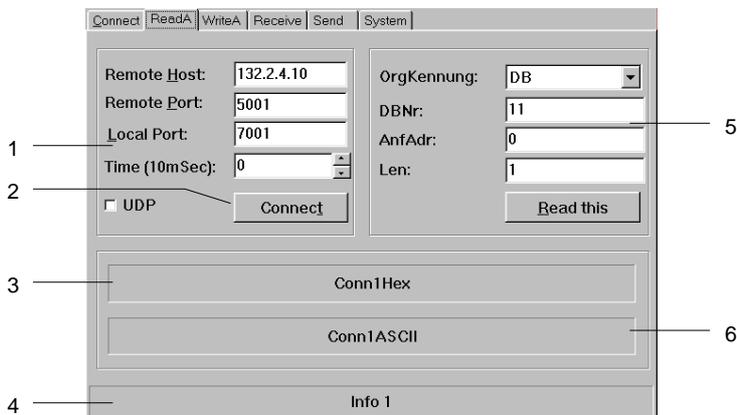
<i>Save All</i>	Alle Parameter speichern.
<i>Save Conn 1</i> bis	Speichert die jeweilige Verbindung.
<i>Save Conn5</i>	
<i>Save Win Pos</i>	Speichert die aktuelle Fensterposition.
<i>Show Hints</i>	Wenn Sie mit der Maus auf ein Eingabefeld oder auf eine Schaltfläche fahren, so wird, wenn "Show Hints" ausgewählt ist, ein Kurzhilfetext eingeblendet.

**Connect-Register  
(Status)**

Das Fenster zeigt den Status aller in diesem Programm parametrierbaren Verbindungen an. Sie können hier auf einen Blick erkennen, welche Verbindungen stabil und welche instabil sind. Sobald sich in einem Register ein Status ändert, wird dies in diesem Fenster angezeigt.

Hier finden Sie auch zur Kontrolle Ihre eigene IP-Adresse.

## ReadA-Register



- [1] Verbindungsdaten
- [2] Verbindung aufbauen
- [3] hexadezimale Darstellung
- [4] Informationsfenster für Verbindungsstatus
- [5] Quelldaten
- [6] ASCII-Darstellung der empfangenen Daten

Das Fenster bietet Ihnen die Möglichkeit zur Parametrierung einer aktiven Leseverbindung.

Neben den Daten für den Verbindungsaufbau geben Sie hier die Quelle an, aus der Daten gelesen werden sollen.

### Eingabefelder

- Remote Host* IP-Adresse der Station, aus der Daten gelesen werden sollen
- Remote Port* Verbindungsadresse der fremden Station
- Local Port* Verbindungsadresse der eigenen Station - Zur Vereinfachung können Sie für Remote und Local die gleiche Adresse verwenden.
- Time (10mSec)* Einstellbarer Zeitintervall für zyklisches Lesen
- OrgKennung* Typ des Quellbausteins
- DBNr* Nummer des Quellbausteins
- AnfAdr* Anfangsadresse des Quellbausteins
- Len* Wortlänge des Quellbausteins

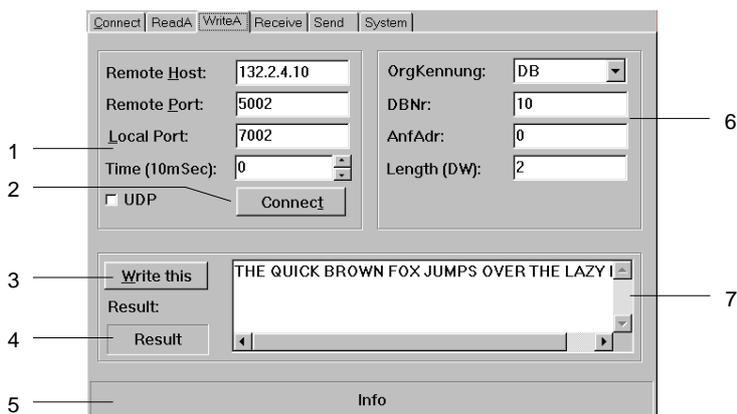
### Kontrollkästchen

- UDP* Hier wählen Sie eine ungesicherte Kommunikation an. Bei der ungesicherten Kommunikation wird mit keiner virtuellen Verbindung gearbeitet. Auf diese Weise können Sie nur UDP-Telegramme anzeigen.

### Schaltflächen

- Connect* Die Verbindung wird aufgebaut und für den Lesevorgang vorbereitet.
- Read this* Über diese Verbindung werden die angeforderten Daten gelesen.

### WriteA-Register



- [1] Verbindungsdaten
- [2] Verbindung aufbauen
- [3] Daten über Verbindung schicken
- [4] Ergebnis-Code des Schreibauftrags
- [5] Informationsfenster für Verbindungsstatus
- [6] Quelldaten
- [7] ASCII-Text, der an den CP zu übertragen ist

Hier aktivieren Sie eine aktive Schreibverbindung.

Analog zum READ AKTIV-Befehl geben Sie hier neben den Daten für den Verbindungsaufbau den Zielbaustein an, in den die Daten zu übertragen sind.

#### Eingabefelder

- Remote Host* IP-Adresse der Station, in die Daten zu schreiben sind
- Remote Port* Verbindungsadresse der fremden Station
- Local Port* Verbindungsadresse der eigenen Station - Zur Vereinfachung können Sie für Remote und Local die gleiche Adresse verwenden.
- Time (10mSec)* Einstellbarer Zeitintervall für zyklisches Schreiben - Für das zyklische Schreiben muss der Timerwert mindestens 5 betragen.
- OrgKennung* Typ des Zielbausteins
- DBNr* Nummer des Zielbausteins
- AnfAdr* Anfangsadresse des Zielbausteins
- Len* Wortlänge des Zielbausteins

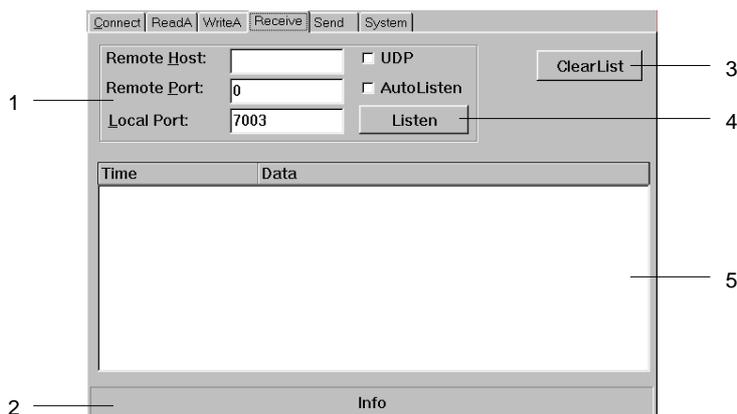
#### Kontrollkästchen

- UDP* Hier wählen Sie eine ungesicherte Kommunikation an. Bei der ungesicherten Kommunikation wird mit keiner virtuellen Verbindung gearbeitet. Das Telegramm wird als UDP-Telegramm geschickt.

#### Schaltflächen

- Connect* Die Verbindung wird aufgebaut und für den Schreibvorgang vorbereitet.
- Write this* Über die durch *Connect* aufgebaute Verbindung werden die im ASCII-Feld eingegebenen Daten in den CP geschrieben.

## Receive-Register



- [1] Verbindungsdaten
- [2] Informationsfenster für Verbindungsstatus
- [3] Empfangsliste löschen
- [4] Telegramme auflisten
- [5] Liste der empfangenen Telegramme

In diesem Dialogfenster können Sie den Empfang von Telegrammen eines bestimmten Hostrechners parametrieren.

### Eingabefelder

- Remote Host* IP-Adresse der Station, in die Daten zu schreiben sind
- Remote Port* Verbindungsadresse der fremden Station
- Local Port* Verbindungsadresse der eigenen Station - Zur Vereinfachung können Sie für Remote und Local die gleiche Adresse verwenden.

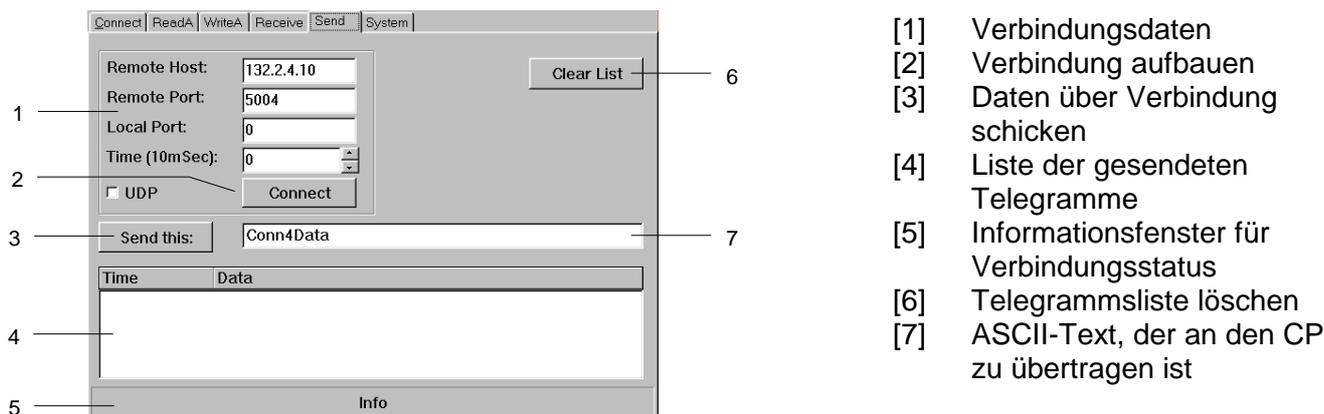
### Kontrollkästchen

- UDP* Hier wählen Sie eine ungesicherte Kommunikation an. Bei der ungesicherten Kommunikation wird mit keiner virtuellen Verbindung gearbeitet. Es werden nur UDP-Telegramme empfangen und angezeigt.
- AutoListen* Durch Anwahl von "AutoListen" springt das Programm in den Empfangsmodus. Es wird jedes empfangene Telegramm des Remote-CPs in der Liste angezeigt. Verbindungsunterbrechungen werden erkannt und angezeigt, das Programm bleibt aber empfangsbereit. Sobald die Verbindung wieder steht, wird die Telegrammauflistung fortgesetzt.

### Schaltflächen

- Listen* Die empfangenen Telegramme werden aufgelistet. Sobald Sie auf die Schaltfläche "STOP" klicken oder die Verbindung unterbrochen wird, wird die Auflistung beendet. Sie können auch durch die Eingabe neuer Verbindungsparameter die Auflistung beenden.
- ClearList* Die Empfangsliste wird gelöscht und die Auflistung in der obersten Zeile der Liste fortgesetzt.

## Send-Register



- [1] Verbindungsdaten
- [2] Verbindung aufbauen
- [3] Daten über Verbindung schicken
- [4] Liste der gesendeten Telegramme
- [5] Informationsfenster für Verbindungsstatus
- [6] Telegrammsliste löschen
- [7] ASCII-Text, der an den CP zu übertragen ist

In diesem Dialogfenster können Sie ein Telegramm an einen spezifizierten Hostrechner schicken.

### Eingabefelder

- Remote Host* IP-Adresse der Station, in die Daten zu schreiben sind
- Remote Port* Verbindungsadresse der fremden Station
- Local Port* Verbindungsadresse der eigenen Station - Zur Vereinfachung können Sie für Remote und Local die gleiche Adresse verwenden.
- Time (10mSec)* Einstellbarer Zeitintervall für zyklisches Schreiben - Für das zyklische Schreiben muss der Timerwert mindestens 5 betragen.

### Kontrollkästchen

- UDP* Hier wählen Sie eine ungesicherte Kommunikation an. Bei der ungesicherten Kommunikation wird mit keiner virtuellen Verbindung gearbeitet. Das Telegramm wird als UDP-Telegramm gesendet.

### Schaltflächen

- Connect* Die Verbindung wird aufgebaut und für den Schreibvorgang vorbereitet.
- Send this* Über die durch *Connect* aufgebaute Verbindung werden die im ASCII-Feld eingegebenen Daten in den CP geschrieben.

## System-Register



In diesem Dialogfenster erhalten Sie Informationen über Ihren spezifizierten Host-CP.

### Eingabefelder

<i>Remote Host</i>	IP-Adresse der Station, in die Daten zu schreiben sind
<i>Remote Port</i>	Verbindungsadresse der fremden Station
<i>Local Port</i>	Verbindungsadresse der eigenen Station - Zur Vereinfachung können Sie für Remote und Local die gleiche Adresse verwenden.

### Schaltflächen

<i>Connect</i>	Die Verbindung wird aufgebaut und für die Kommunikation vorbereitet.
<i>GetState</i>	Über die durch <i>Connect</i> aufgebaute Verbindung wird der Status des CPs übertragen und im Statusfenster angezeigt. Angezeigt werden kann: <ul style="list-style-type: none"> <li>- Hardware-Stop (Run/Stop-Schalter am CP steht in Stop-Stellung) Der CP ist mit dem Testprogramm nicht fernbedienbar.</li> <li>- Hardware-Run (Run/Stop-Schalter am CP steht in Run-Stellung) Der CP ist mit dem Testprogramm fernbedienbar.</li> <li>- Software-Stop (Run/Stop-Schalter am CP muss in Run-Stellung stehen) Der CP wurde mit <i>SetStop</i> in Stop geschaltet.</li> <li>- Software-Run (Run/Stop-Schalter am CP muss in Run-Stellung stehen) Der CP wurde mit <i>SetRun</i> in Run geschaltet.</li> </ul>
<i>SetStop</i>	Der CP wird in Stop geschaltet. Diese Funktion ist nur möglich, wenn am CP der Run/Stop-Schalter in Run-Stellung steht.
<i>SetRun</i>	Der CP wird in Run geschaltet. Auch diese Funktion ist nur möglich, wenn am CP der Run/Stop-Schalter in Run-Stellung steht.

## Teil 7 Einsatz CPU 21xDPM

### Überblick

Inhalt dieses Kapitels ist der Einsatz der CPU 21xDPM unter PROFIBUS. Nach einer kurzen Einführung in das PROFIBUS-System wird die Projektierung und der Einsatz unter MPI gezeigt.

Mit Angaben zu den Betriebsarten des DP-Masters und zur Inbetriebnahme endet dieses Kapitel.

### Inhalt

Thema	Seite
<b>Teil 7 Einsatz CPU 21xDPM</b> .....	<b>7-1</b>
Grundlagen .....	7-2
Projektierung CPU mit integriertem PROFIBUS-DP-Master .....	7-5
Projekt transferieren .....	7-9
DP-Master-Betriebsarten .....	7-12
Inbetriebnahme und Anlaufverhalten .....	7-13

## Grundlagen

### Allgemein

PROFIBUS ist ein internationaler offener Feldbus-Standard für Gebäude-, Fertigungs- und Prozessautomatisierung. PROFIBUS legt die technischen und funktionellen Merkmale eines seriellen Feldbus-Systems fest, mit dem verteilte digitale Feldautomatisierungsgeräte im unteren (Sensor-/Aktor-Ebene) bis mittleren Leistungsbereich (Prozessebene) vernetzt werden können.

PROFIBUS besteht aus einem Sortiment kompatibler Varianten. Die hier angeführten Angaben beziehen sich auf den PROFIBUS-DP.

### 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. PROFIBUS-DP ist für den schnellen Datenaustausch auf der Sensor-Aktor-Ebene konzipiert.

Der Datenaustausch "Data Exchange" erfolgt zyklisch. Während eines Buszyklus liest der Master die Eingangswerte der Slaves und schreibt neue Ausgangsinformationen an die Slaves.

### Master und Slaves

PROFIBUS unterscheidet zwischen aktiven Stationen (Master) und passiven Stationen (Slave).

#### *Master-Geräte*

Master-Geräte bestimmen den Datenverkehr auf dem Bus. Es dürfen auch mehrere Master an einem PROFIBUS eingesetzt werden. Man spricht dann vom Multi-Master-Betrieb. Durch das Busprotokoll wird ein logischer Tokenring zwischen den intelligenten Geräten aufgebaut. Nur der Master, der in Besitz des Tokens ist, kommuniziert mit seinen Slaves gerade.

Ein Master, wie in der CPU 21xDPM darf Nachrichten ohne externe Aufforderung aussenden, wenn er im Besitz der Buszugriffsberechtigung (Token) ist. Master werden im PROFIBUS-Protokoll auch als aktive Teilnehmer bezeichnet.

#### *Slave-Geräte*

Ein PROFIBUS-Slave stellt Daten von Peripheriegeräten, Sensoren, Aktoren und Messumformern zur Verfügung. Die VIPA PROFIBUS-Koppler sind modulare Slave-Geräte, die Daten zwischen der System 200V Peripherie und dem übergeordneten Master transferieren.

Diese Geräte haben gemäß der PROFIBUS-Norm keine Buszugriffsberechtigung. Sie dürfen nur Nachrichten quittieren oder auf Anfrage eines Masters Nachrichten an diesen übermitteln. Slaves werden auch als passive Teilnehmer bezeichnet.

---

**Kommunikation**

Das Busübertragungsprotokoll bietet zwei Verfahren für den Buszugriff:

**Master mit Master**

Die Master-Kommunikation wird auch als Token-Passing-Verfahren bezeichnet. Das Token-Passing-Verfahren garantiert die Zuteilung der Buszugriffsberechtigung. Das Zugriffsrecht auf den Bus wird zwischen den Geräten in Form eines "Token" weitergegeben. Der Token ist ein spezielles Telegramm, das über den Bus übertragen wird.

Wenn ein Master den Token besitzt, hat er das Buszugriffsrecht auf den Bus und kann mit allen anderen aktiven und passiven Geräten kommunizieren. Die Tokenhaltezeit wird bei der Systemkonfiguration bestimmt. Nachdem die Tokenhaltezeit abgelaufen ist, wird der Token zum nächsten Master weitergegeben, der dann den Buszugriff hat und mit allen anderen Geräten kommunizieren kann.

**Master-Slave-Verfahren**

Der Datenverkehr zwischen dem Master und den ihm zugeordneten Slaves wird in einer festgelegten, immer wiederkehrenden Reihenfolge automatisch durch den Master durchgeführt. Bei der Projektierung bestimmen Sie die Zugehörigkeit des Slaves zu einem bestimmten Master. Weiter können Sie definieren, welche DP-Slaves für den zyklischen Nutzdatenverkehr aufgenommen oder ausgenommen werden.

Der Datentransfer zwischen Master und Slave gliedert sich in Parametrierungs-, Konfigurierungs- und Datentransfer-Phasen. Bevor ein DP-Slave in die Datentransfer-Phase aufgenommen wird, prüft der Master in der Parametrierungs- und Konfigurationsphase, ob die projektierte Konfiguration mit der Ist-Konfiguration übereinstimmt. Überprüft werden Gerätetyp, Format- und Längeninformatoren und die Anzahl der Ein- und Ausgänge. Sie erhalten so einen zuverlässigen Schutz gegen Parametrierfehler.

Zusätzlich zum Nutzdatentransfer, den der Master selbständig durchführt, können Sie neue Parametrierdaten an einen Bus-Koppler schicken.

Im Zustand DE "Data Exchange" sendet der Master neue Ausgangsdaten an den Slave und im Antworttelegramm des Slaves werden die aktuellen Eingangsdaten an den Master übermittelt.

**Datenkonsistenz**

Daten bezeichnet man als konsistent, wenn sie inhaltlich zusammengehören. Inhaltlich gehören zusammen: das High- und Low-Byte eines Analogwerts (wortkonsistent) und das Kontroll- und Status-Byte mit zugehörigem Parameterwort für den Zugriff auf die Register.

Die Datenkonsistenz ist im Zusammenspiel von Peripherie und Steuerung grundsätzlich nur für 1Byte sichergestellt. Das heißt, die Bits eines Bytes werden zusammen eingelesen bzw. ausgegeben. Für die Verarbeitung digitaler Signale ist eine byteweise Konsistenz ausreichend.

---

**Übertragungs-  
medium**

PROFIBUS verwendet als Übertragungsmedium eine geschirmte, verdrehte Zweidrahtleitung auf Basis der RS485-Schnittstelle oder eine Duplex-Lichtwellenleitung (LWL). Die Übertragungsrate liegt bei beiden Systemen bei maximal 12MBaud.

**Elektrisches  
System über  
RS485**

Die RS485-Schnittstelle arbeitet mit Spannungsdifferenzen. Sie ist daher unempfindlicher gegenüber Störeinflüssen als eine Spannungs- oder Stromschnittstelle. Sie können das Netz sowohl als Linien- als auch als Baumstruktur konfigurieren. Auf Ihrem VIPA PROFIBUS-Koppler befindet sich eine 9polige Buchse. Über diese Buchse koppeln Sie den PROFIBUS-Koppler als Slave direkt in Ihr PROFIBUS-Netz ein.

Die Busstruktur unter RS485 erlaubt das rückwirkungsfreie Ein- und Auskoppeln 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.

**Optisches  
System über  
Lichtwellenleiter**

Das Lichtwellenleitersystem arbeitet mit Lichtimpulsen von monochromatischem Licht. Der Lichtwellenleiter ist völlig unempfindlich gegenüber Störspannungen von außen. Ein Lichtwellenleitersystem wird in Linienstruktur aufgebaut. Jedes Gerät ist mit einem Hin- und Rückleiter zu verbinden. Ein Abschluss am letzten Gerät ist nicht erforderlich.

Das rückwirkungsfreie Ein- und Auskoppeln von Stationen ist aufgrund der Linienstruktur nicht möglich.

---

**Adressierung**

Jeder Teilnehmer am PROFIBUS identifiziert sich mit einer Adresse. Diese Adresse darf nur einmal in diesem Bussystem vergeben sein und kann zwischen 0 und 125 liegen.

Bei der CPU 21xDPM stellen Sie die Adresse über Ihr Projektiertool ein.

**GSD-Datei**

Zur Konfiguration einer Slave-Anschaltung in Ihrem eigenen Projektiertool bekommen Sie die Leistungsmerkmale der PROFIBUS-Komponenten in Form einer GSD-Datei mitgeliefert. Installieren Sie diese GSD-Datei in Ihrem Projektiertool.

Aufbau und Inhalt der GSD-Datei sind durch die PROFIBUS Nutzerorganisation (PNO) genormt und können dort jederzeit abgerufen werden.

Nähere Hinweise zur Installation der GSD-Datei finden Sie im Handbuch zu Ihrem Projektiertool.

## Projektierung CPU mit integriertem PROFIBUS-DP-Master

### Übersicht

Zur Projektierung des PROFIBUS-DP-Masters ist der Hardware-Konfigurator von Siemens zu verwenden. Ihre PROFIBUS-Projekte übertragen Sie mit den "Zielsystem"-Funktionen über MPI auf die CPU 21xDPM. Diese reicht die Daten weiter an den PROFIBUS-DP-Master.

### Möglichkeit zur Projektierung im Siemens SIMATIC Manager

Die Adresszuordnung und die Parametrierung der direkt gesteckten Module erfolgt im Siemens SIMATIC Manager in Form eines virtuellen PROFIBUS-Systems.

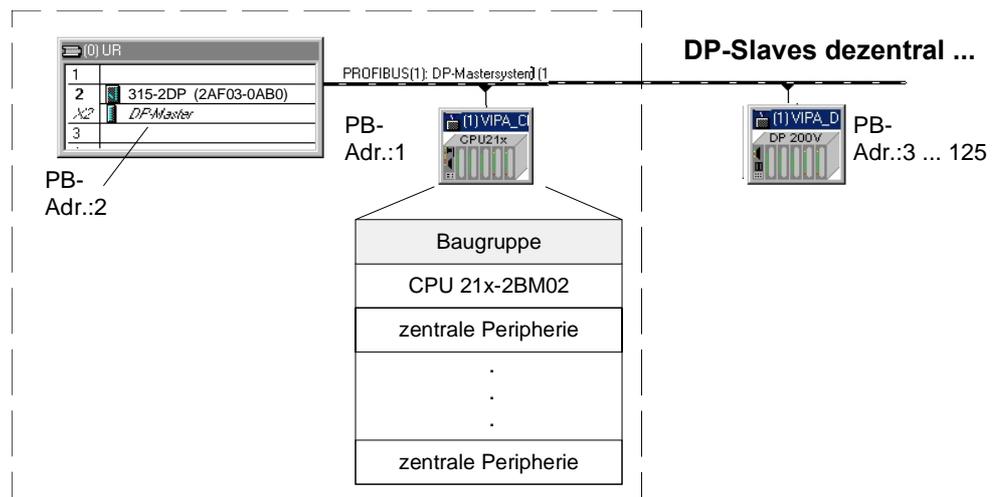
Da die PROFIBUS-Schnittstelle auch softwareseitig standardisiert ist, können wir auf diesem Weg gewährleisten, dass über die Einbindung einer GSD-Datei die Funktionalität in Verbindung mit dem Siemens SIMATIC Manager jederzeit gegeben ist.

### Schritte der Projektierung

Um kompatibel mit dem Siemens SIMATIC Manager zu sein, sind für das System 200V folgende Schritte durchzuführen:

- CPU 315-2DP mit DP-Master-System projektieren (Adresse 2).
- PROFIBUS-Slave **VIPA\_CPU21x** mit Adresse 1 anfügen.
- Auf dem 1. Steckplatz des Slave-Systems den CPU-Typ **21xDPM** einbinden.
- Direkt gesteckte Peripherie-Module ebenfalls über diesen Slave über die nachfolgenden Steckplätze einbinden.

### CPU 21xDPM zentral



### Hinweis!

Zur Projektierung der CPU und des PROFIBUS-DP-Masters werden fundierte Kenntnisse im Umgang mit dem Siemens SIMATIC Manager und dem Hardware-Konfigurator von Siemens vorausgesetzt!

**Voraussetzungen**

Für die Projektierung des PROFIBUS-DP-Masters auf der CPU 21xDPM müssen folgende Voraussetzungen erfüllt sein:

- Siemens SIMATIC Manager ist installiert.
- Bei Einsatz von PROFIBUS-DP-Slaves der Systeme 100V, 200V und 300V von VIPA: GSD-Dateien im Hardware-Konfigurator sind eingebunden.
- Transfermöglichkeit zwischen Projektiertool und CPU 21xDPM ist vorhanden

**Hardware-Konfigurator von Siemens installieren**

Der Hardware-Konfigurator ist Bestandteil des Siemens SIMATIC Managers. Er dient der Projektierung. Die Module, die hier projiziert werden können, entnehmen Sie dem Hardware-Katalog.

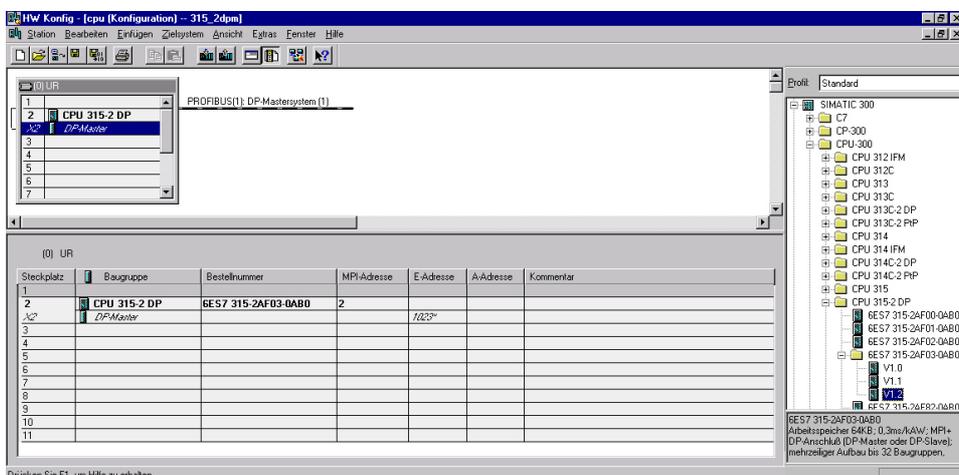
Für den Einsatz der PROFIBUS-DP-Slaves der Systeme 100V, 200V und 300V von VIPA ist die Einbindung der Module über die GSD-Datei von VIPA im Hardwarekatalog erforderlich.

**Projektierung CPU 21xDPM**

Nachfolgend werden die einzelnen Schritte der Projektierung aufgezeigt.

**Virtuelles PROFIBUS-System erzeugen**

- Legen Sie ein neues Projekt System 300V an.
- Fügen Sie aus dem Hardwarekatalog eine Profilschiene ein.
- Sie finden die CPU mit PROFIBUS-Master im Hardwarekatalog unter: *Simatic300 > CPU-300 > CPU315-2DP > 6ES7 315-2AF03-0AB0*
- Fügen Sie die CPU 315-2DP (**6ES7 315-2AF03-0AB0 V1.2**) ein.
- Geben Sie eine PROFIBUS-Adresse >1 für Ihren Master an.
- Klicken Sie auf DP und stellen Sie in unter *Objekteigenschaften* die Betriebsart "DP-Master" ein und bestätigen Sie Ihre Eingabe mit OK.
- Durch Klick mit der rechten Maustaste auf "DP" öffnet sich das Kontextmenü. Wählen Sie "Mastersystem einfügen" aus. Legen Sie über NEU ein neues PROFIBUS-Subnetz an.



**CPU-Teil projektieren**

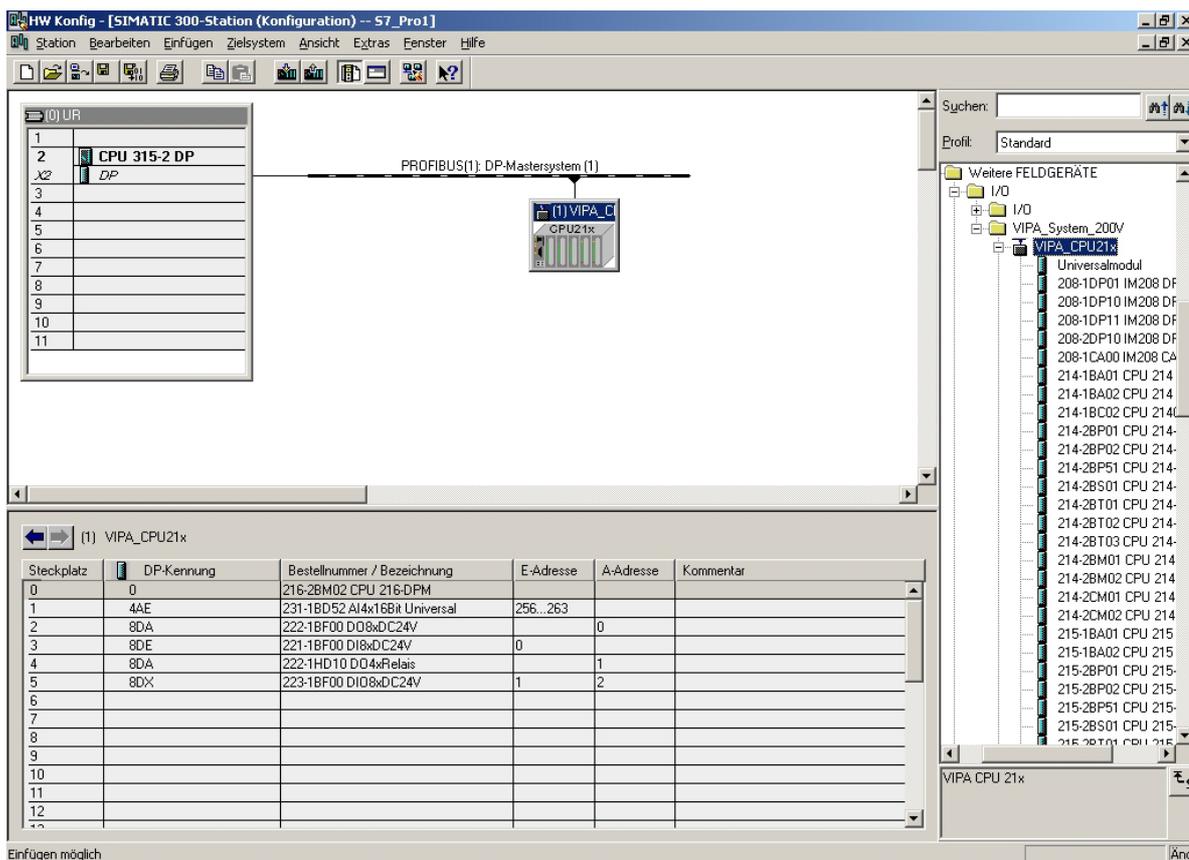
Um, wie schon weiter oben erwähnt, zum Siemens SIMATIC Manager kompatibel zu sein, müssen Sie den CPU-Teil explizit einbinden.

- Hängen Sie hierzu an das Subnetz das System "VIPA\_CPU21x". Sie finden dies im Hardware-Katalog unter PROFIBUS DP > Weitere Feldgeräte > IO > VIPA\_System\_200V.
- Geben Sie diesem Slave die PROFIBUS-Adresse 1.
- Platzieren Sie auf dem 1. Steckplatz die CPU 21x-2BM02 von VIPA, indem Sie diese im Hardware-Katalog unter VIPA\_CPU21x auswählen.

Sie haben jetzt ihren PROFIBUS-DP-Master und Ihre CPU projektiert. Nachfolgend wird gezeigt, wie Sie die direkt gesteckten System 200V Module einbinden.

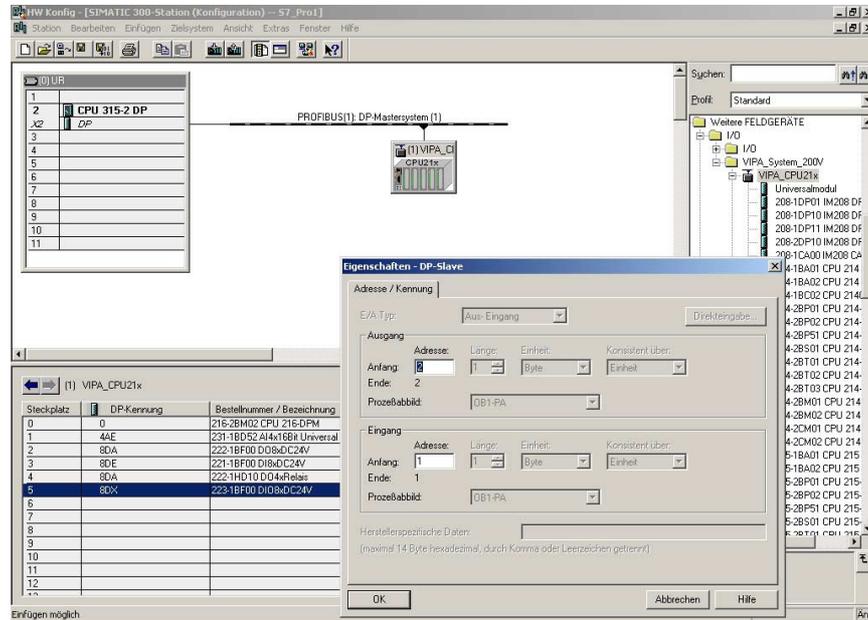
**Zentrale Peripherie projektieren**

Zur Einbindung der am VIPA-Bus befindlichen Module ziehen Sie aus dem Hardware-Katalog unter VIPA\_CPU21x die entsprechenden System 200V Module auf die Steckplätze unterhalb der CPU.



**Module parametrieren**

System 200V Module können bis zu 16Byte Parameterdaten von der CPU erhalten. Durch Einsatz des Siemens SIMATIC Manager können Sie jederzeit für parametrierbare System 200V Module Parameter vorgeben. Doppelklicken Sie hierzu bei der Projektierung in Ihrer Steckplatzübersicht auf das zu parametrierende Modul.

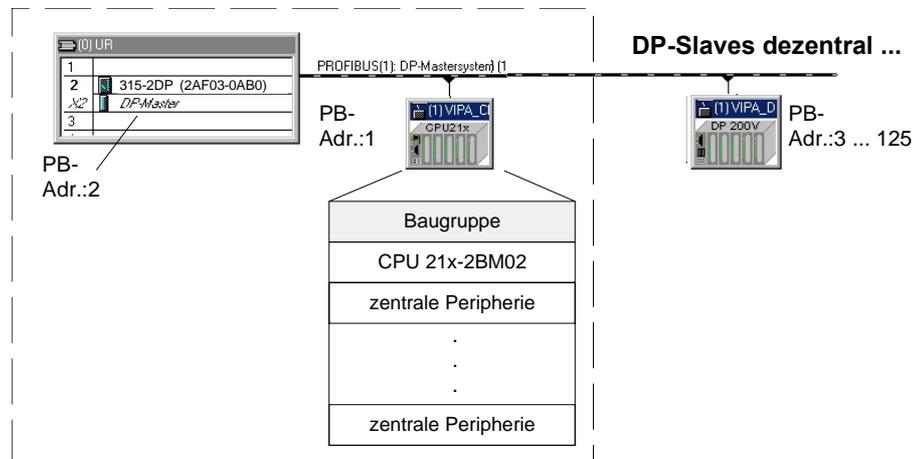


**DP-Slaves projektieren**

Zur Projektierung von DP-Slaves, die an den DP-Master der CPU 21xDPM angekoppelt sind, verfahren Sie auf die gleiche Weise wie bei der Projektierung des DP200V-Systems.

Entnehmen Sie aus dem Hardware-Katalog das entsprechende PROFIBUS-System und ziehen Sie dies auf das DP-Master Subnetz. Vergeben Sie dem Slave eine Adresse > 3.

**CPU 21xDPM zentral**



## Projekt transferieren

### Übersicht

Es bestehen 2 Möglichkeiten für den Transfer Ihres Projekts in die CPU:

- Transfer über MPI
- Transfer über MMC bei Einsatz eines MMC-Lesers

---

### Transfer über MPI

Der Aufbau eines MPI-Netzes ist prinzipiell gleich dem Aufbau eines 1,5MBaud PROFIBUS-Netzes. Das heißt, es gelten dieselben Regeln und Sie verwenden für beide Netze die gleichen Komponenten zum Aufbau.

Defaultmäßig wird das MPI-Netz mit 187kBAud betrieben.

Jeder Busteilnehmer identifiziert sich mit einer eindeutigen MPI-Adresse am Bus.

Sie verbinden die einzelnen Teilnehmer über Busanschlussstecker und das PROFIBUS-Buskabel.

### 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, während des Hochlaufs und des Betriebs immer mit Spannung versorgt sind.

### Vorgehensweise

- Verbinden Sie Ihr PG bzw. Ihren PC über MPI mit Ihrer CPU. Sollte Ihr Programmiergerät keine MPI-Schnittstelle besitzen, können Sie für eine serielle Punkt-zu-Punkt-Übertragung von Ihrem PC an MPI das "Green Cable" von VIPA verwenden. Das "Green Cable" hat die Best.-Nr. VIPA 950-0KB00 und darf nur bei den VIPA CPUs mit MP<sup>2</sup>I-Schnittstelle eingesetzt werden. Näheres hierzu siehe Teil "Einsatz CPU 21x" unter "Projekt transferieren".
- Konfigurieren Sie die MPI-Schnittstelle Ihres PCs.
- Mit **Zielsystem** > *Laden in Baugruppe* in Ihrem Projektierool übertragen Sie Ihr Projekt in die CPU.
- Zur zusätzlichen Sicherung Ihres Projekts auf MMC stecken Sie eine MMC und übertragen Sie mit **Zielsystem** > *RAM nach ROM kopieren* Ihr Anwenderprogramm auf die MMC. Während des Schreibvorgangs blinkt die "MC"-LED auf der CPU. Systembedingt wird zu früh ein erfolgter Schreibvorgang gemeldet. Der Schreibvorgang ist erst beendet, wenn die LED erlischt.

## MPI konfigurieren

Hinweise zur Konfiguration einer MPI-Schnittstelle finden Sie in der Dokumentation zu Ihrer Programmiersoftware.

An dieser Stelle soll lediglich der Einsatz des Green Cable von VIPA in Verbindung mit dem Programmierwerkzeug von Siemens gezeigt werden.

Das Green Cable stellt über MPI eine serielle Punkt-zu-Punkt Verbindung zwischen der COM-Schnittstelle des PCs und der MP<sup>2</sup>I-Schnittstelle der CPU her.

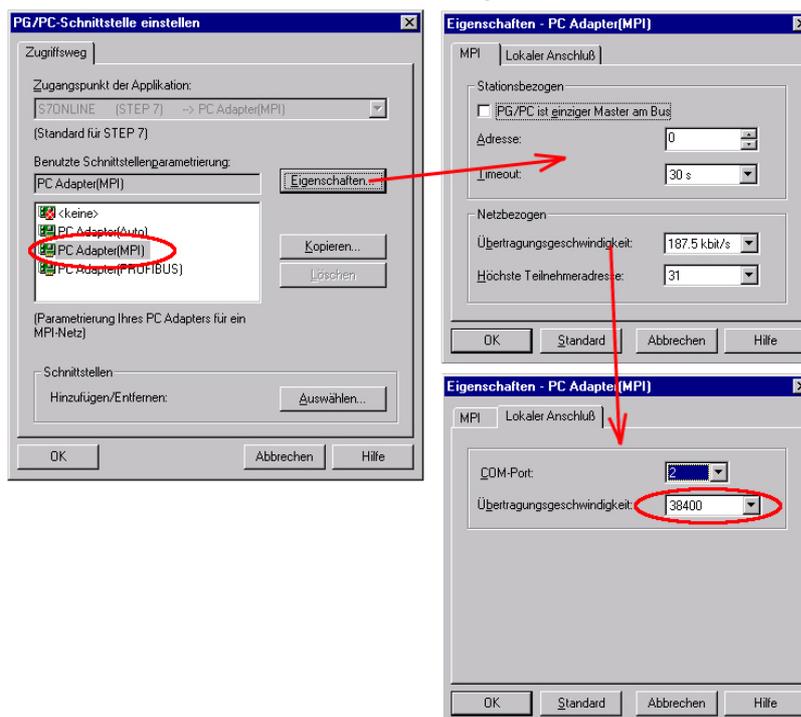


### Achtung!

Das Green Cable darf nur an VIPA-CPU mit MP<sup>2</sup>I-Schnittstelle eingesetzt werden. Näheres hierzu siehe Teil "Einsatz CPU 21x" Kapitel "Projekt transferieren".

### Vorgehensweise

- Starten Sie den Siemens SIMATIC Manager.
- Wählen Sie unter **Extras** > *PG/PC-Schnittstelle einstellen*  
→ Es öffnet sich folgendes Dialogfenster, in dem Sie die zu verwendende MPI-Schnittstelle konfigurieren können:



- Wählen Sie in der Auswahlliste "PC Adapter (MPI)" aus; ggf müssen Sie diesen erst hinzufügen.
  - Klicken auf [Eigenschaften].
- In den folgenden 2 Unterdialogen können Sie, wie in der Abbildung gezeigt, Ihren PC-Adapter konfigurieren:



### Hinweis!

Bitte beachten Sie, dass Sie bei Einsatz des "Green Cable" die Übertragungsgeschwindigkeit auf 38400Baud einstellen.

---

**Einsatz der MMC**

Als externes Speichermedium kommt die Multi Media Card (MMC) zum Einsatz (Best.-Nr. VIPA 953-0KX10).

Das Lesen der MMC erfolgt immer nach URLÖSCHEN.

Beschrieben wird die MMC entweder über einen Schreibbefehl aus dem Hardware-Konfigurator von Siemens oder mit einem MMC-Lesegerät von VIPA (Best.-Nr: VIPA 950-0AD00). Somit ist es möglich Programme am PC zu erstellen, diese auf die MMC zu kopieren und durch Stecken in die VIPA CPU zu übertragen.

Die MMC-Module werden mit dem File-System FAT16 vorformatiert von VIPA ausgeliefert.

**Erforderliche Dateien**

Es dürfen sich mehrere Projekte und Unterverzeichnisse auf einem MMC-Speichermodul befinden.

Bitte beachten Sie, dass sich Ihre aktuelle Projektierung im Root-Verzeichnis befindet und folgenden Dateinamen hat: **S7PROG.WLD**.

**Transfer CPU → MMC**

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

Den Schreibbefehl starten Sie aus dem Hardware-Konfigurator von Siemens über **Zielsystem** > *RAM nach ROM kopieren*.

Während des Schreibvorgangs blinkt die gelbe "MC"-LED der CPU.

**Transfer MMC → CPU**

Das Übertragen des Anwenderprogramms von der MMC in die CPU erfolgt immer nach URLÖSCHEN. Das Blinken der gelben LED "MC" der CPU kennzeichnet den Übertragungsvorgang.

Ist kein gültiges Anwenderprogramm auf der gesteckten MMC oder scheitert die Übertragung, so erfolgt das URLÖSCHEN der CPU und die STOP-LED blinkt dreimal.

Der Master befindet sich nun mit folgenden Default-Parametern am Netz:

**Default-Bus-Parameter: Adresse: 1, Übertragungsrate: 1,5MBAud**

**Hinweis!**

Ist das Anwenderprogramm größer als der Anwenderspeicher in der CPU, wird der Inhalt der MMC nicht in die CPU übertragen.

Ist bei einem Schreibbefehl keine MMC gesteckt, führt dies zu einer Fehlermeldung über unzureichenden Speicher.

Führen Sie vor der Übertragung eine Komprimierung durch.

## DP-Master-Betriebsarten

### STOP → RUN (automatisch)

Nach NETZ EIN und bei gültigen Projektierdaten in der CPU geht der Master automatisch in RUN über. Auf einen Betriebsarten-Schalter für den Master wurde verzichtet.

Nun wird die Kommunikation zu den DP-Slaves aufgebaut. Während dieser Zeit brennt nur die RUN-LED. Bei erfolgter Kommunikation und gültigen Bus-Parametern, geht der DP-Master in Data Exchange (DE). Die LEDs RUN und DE leuchten.

Bei fehlerhaften Parametern geht der DP-Master in RUN und zeigt über die IF-LED einen Parametrierfehler an.

Der DP-Master befindet sich nun mit folgenden Default-Bus-Parametern am Bus:

**Default-Bus-Parameter:   Adresse:1, Übertragungsrate:1,5 MBaud.**

### RUN

Im RUN leuchten die RUN- und DE-LEDs. Jetzt können Daten ausgetauscht werden. Im Fehlerfall wie z.B. DP-Slave-Ausfall, wird dies am DP-Master über die ERR-LED angezeigt und ein Alarm an die CPU abgesetzt.



#### **Hinweis!**

Sollte die CPU während des Betriebs in STOP gehen, bleibt der DP-Master im RUN.

Aufgrund des BASP-Signals werden alle Ausgänge der über DP-Slaves angebotenen Peripherie-Module auf Null gesetzt.

Das Prozessabbild der Eingänge behält nach einem Slaveausfall den Zustand wie vor dem Slaveausfall.

## Inbetriebnahme und Anlaufverhalten

### Checkliste für die Inbetriebnahme

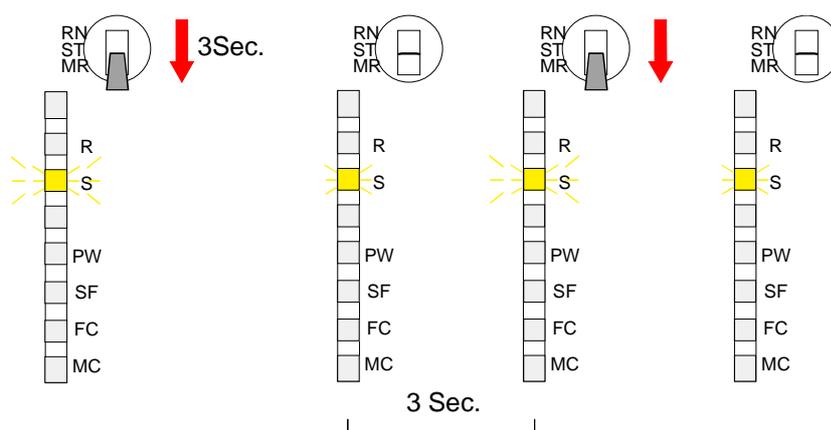
- Schalten Sie die Stromversorgung ab.
- Bauen Sie Ihr System auf.
- Verdrahten Sie Ihr System.
- Stecken Sie die MMC mit CPU-Programm und PROFIBUS-Projekt.
- Schalten Sie die Spannungsversorgung ein.
- Zur Übernahme Ihres Projekts von der MMC in die CPU führen Sie URLÖSCHEN durch.

### Stromversorgung einschalten

Schalten Sie die Stromversorgung ein. Es laufen danach Vorgänge auf der CPU ab, die nachfolgend unter "Anlauf" näher beschrieben sind.

### URLÖSCHEN

Die nachfolgende Abbildung zeigt nochmals die Vorgehensweise:



### Anlauf im Auslieferungszustand

Im Auslieferungszustand ist die CPU urlöscht.

Nach einem STOP → RUN Übergang geht die CPU ohne Programm in RUN.

Nach NETZ EIN versucht der DP-Master Parameter von der CPU zu erhalten.

Da der Master keine gültigen Parameter von der CPU erhält, startet der Master mit Defaultparametern (Adr.:1, 1,5MBit) aus seinem ROM und zeigt dies über die "IF"-LED an.

**Anlauf mit gültigen Daten in der CPU**

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

Zuvor bekommt der DP-Master gültige Parameter und startet mit diesen.

**Anlauf mit einer gültigen Memory-Card**

Das Lesen von einer MMC erfolgt nur nach URLÖSCHEN.

Nach dem URLÖSCHEN überprüft der DP-Master die Gültigkeit der Parameter in der CPU.

Sind diese gültig, werden sie übernommen.

Sind die Daten ungültig, geht die CPU in STOP und der DP-Master startet mit seinen Defaultwerten.

**Anlauf bei leerem Akku**

Der Akku wird direkt über die eingebaute Spannungsversorgung über eine Ladeelektronik geladen und gewährleistet eine Pufferung für max. 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 wird das Programm auf der MMC in des RAM übertragen und der DP-Master mit Parametern versorgt.

Sind diese gültig, geht der DP-Master mit diesen Parametern an den Bus.

Bei ungültigen Parametern startet der Master mit Defaultparametern (Adr.:1, 1,5 MBit) aus seinem ROM und zeigt dies über die "IF"-LED an.

Abhängig von der eingestellten Betriebsart geht die CPU in RUN bzw. bleibt im STOP.

Dieser Vorgang wird im Diagnosepuffer unter folgendem Eintrag festgehalten: "Start URLÖSCHEN automatisch (ungepuffert NETZ-EIN)".

## Teil 8 Einsatz CPU 21xDP

### Überblick

Inhalt dieses Kapitels ist der Einsatz der CPU 21xDP unter PROFIBUS. Sie erhalten hier alle Informationen, die zum Einsatz eines intelligenten PROFIBUS-DP-Slaves erforderlich sind.

Mit einem ausführlichen Beispiel für die CPU 21xDP und 21xDPM endet das Kapitel.

### Inhalt

Thema	Seite
<b>Teil 8 Einsatz CPU 21xDP</b> .....	<b>8-1</b>
Grundlagen .....	8-2
Projektierung der CPU 21xDP.....	8-7
DP-Slave Parameter .....	8-12
Diagnosefunktionen .....	8-15
Statusmeldung intern an CPU.....	8-18
PROFIBUS Aufbaurichtlinien.....	8-20
Inbetriebnahme .....	8-23
Beispiel .....	8-25

## Grundlagen

### Allgemein

PROFIBUS ist ein internationaler offener Feldbus-Standard für Gebäude-, Fertigungs- und Prozessautomatisierung. PROFIBUS legt die technischen und funktionellen Merkmale eines seriellen Feldbus-Systems fest, mit dem verteilte digitale Feldautomatisierungsgeräte im unteren (Sensor-/Aktor-Ebene) bis mittleren Leistungsbereich (Prozessebene) vernetzt werden können.

PROFIBUS besteht aus einem Sortiment kompatibler Varianten. Die hier angeführten Angaben beziehen sich auf den PROFIBUS-DP.

### 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. PROFIBUS-DP ist für den schnellen Datenaustausch auf der Sensor-Aktor-Ebene konzipiert.

Der Datenaustausch "Data Exchange" erfolgt zyklisch. Während eines Buszyklus liest der Master die Eingangswerte der Slaves und schreibt neue Ausgangsinformationen an die Slaves.

### Master und Slaves

PROFIBUS unterscheidet zwischen aktiven Stationen (Master) und passiven Stationen (Slave).

#### *Master-Geräte*

Master-Geräte bestimmen den Datenverkehr auf dem Bus. Es dürfen auch mehrere Master an einem PROFIBUS eingesetzt werden. Man spricht dann vom Multi-Master-Betrieb. Durch das Busprotokoll wird ein logischer Tokenring zwischen den intelligenten Geräten aufgebaut. Nur der Master, der in Besitz des Tokens ist, kommuniziert mit seinen Slaves gerade.

Ein Master darf Nachrichten ohne externe Aufforderung aussenden, wenn er im Besitz der Buszugriffsberechtigung (Token) ist. Master werden im PROFIBUS-Protokoll auch als aktive Teilnehmer bezeichnet.

#### *Slave-Geräte*

Ein PROFIBUS-Slave stellt Daten von Peripheriegeräten, Sensoren, Aktoren und Messumformern zur Verfügung. Die VIPA PROFIBUS-Koppler sind modulare Slave-Geräte, die Daten zwischen der System 200V Peripherie und dem übergeordneten Master transferieren.

Diese Geräte haben gemäß der PROFIBUS-Norm keine Buszugriffsberechtigung. Sie dürfen nur Nachrichten quittieren oder auf Anfrage eines Masters Nachrichten an diesen übermitteln. Slaves werden auch als passive Teilnehmer bezeichnet.

---

**Kommunikation**

Das Busübertragungsprotokoll bietet zwei Verfahren für den Buszugriff:

**Master mit Master**

Die Master-Kommunikation wird auch als Token-Passing-Verfahren bezeichnet. Das Token-Passing-Verfahren garantiert die Zuteilung der Buszugriffsberechtigung. Das Zugriffsrecht auf den Bus wird zwischen den Geräten in Form eines "Token" weitergegeben. Der Token ist ein spezielles Telegramm, das über den Bus übertragen wird.

Wenn ein Master den Token besitzt, hat er das Buszugriffsrecht auf den Bus und kann mit allen anderen aktiven und passiven Geräten kommunizieren. Die Tokenhaltezeit wird bei der Systemkonfiguration bestimmt. Nachdem die Tokenhaltezeit abgelaufen ist, wird der Token zum nächsten Master weitergegeben, der dann den Buszugriff hat und mit allen anderen Geräten kommunizieren kann.

**Master-Slave-Verfahren**

Der Datenverkehr zwischen dem Master und den ihm zugeordneten Slaves wird in einer festgelegten, immer wiederkehrenden Reihenfolge automatisch durch den Master durchgeführt. Bei der Projektierung bestimmen Sie die Zugehörigkeit des Slaves zu einem bestimmten Master. Weiter können Sie definieren, welche DP-Slaves für den zyklischen Nutzdatenverkehr aufgenommen oder ausgenommen werden.

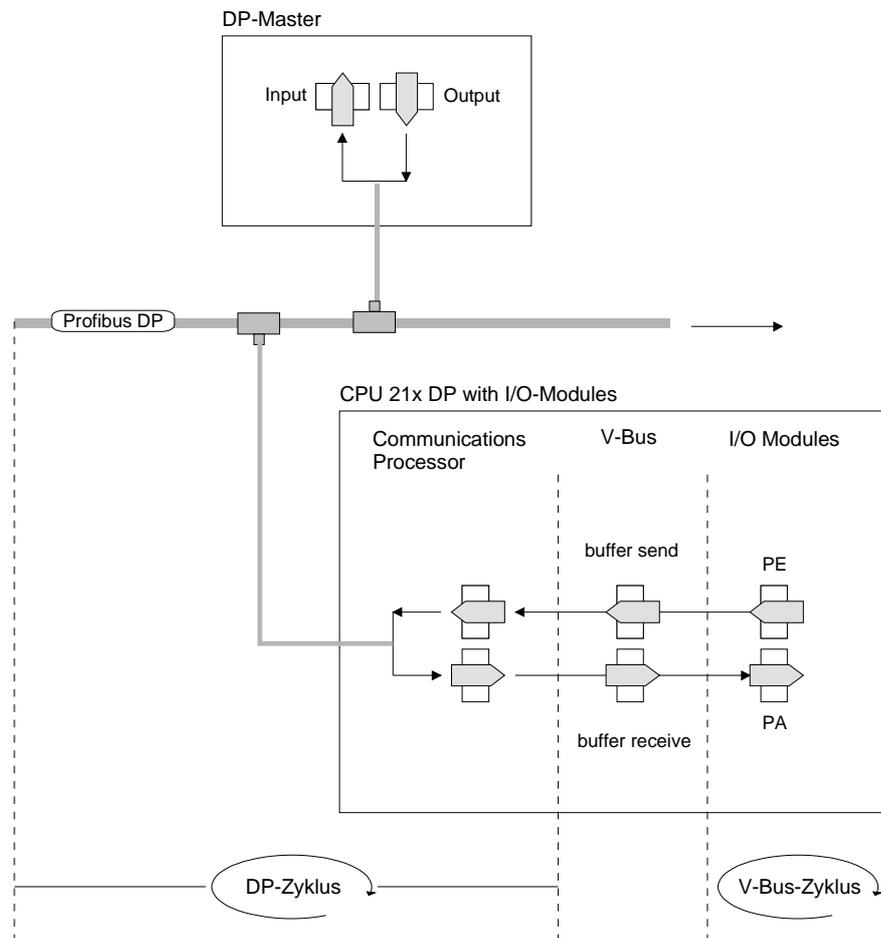
Der Datentransfer zwischen Master und Slave gliedert sich in Parametrierungs-, Konfigurierungs- und Datentransfer-Phasen. Bevor ein DP-Slave in die Datentransfer-Phase aufgenommen wird, prüft der Master in der Parametrierungs- und Konfigurationsphase, ob die projektierte Konfiguration mit der Ist-Konfiguration übereinstimmt. Überprüft werden Gerätetyp, Format- und Längeninformationen und die Anzahl der Ein- und Ausgänge. Sie erhalten so einen zuverlässigen Schutz gegen Parametrierfehler.

Zusätzlich zum Nutzdatentransfer den der Master selbständig durchführt, können Sie neue Parametrierdaten an einen Bus-Koppler schicken.

Im Zustand DE "Data Exchange" sendet der Master neue Ausgangsdaten an den Slave und im Antworttelegramm des Slaves werden die aktuellen Eingangsdaten an den Master übermittelt.

**Funktionsweise der Datenübertragung**

Der Datenaustausch zwischen DP-Master und DP-Slave erfolgt zyklisch über Sende- und Empfangspuffer.



PE: Prozessabbild der Eingänge  
 PA: Prozessabbild der Ausgänge

**V-Bus-Zyklus**

In einem V-Bus-Zyklus (V-Bus = VIPA-Rückwandbus) werden alle Eingangsdaten der Module im PE gesammelt und alle Ausgangsdaten des PA an die Ausgabe-Module geschrieben. Nach erfolgtem Datenaustausch wird das PE in den Sendepuffer (buffer send) übertragen und die Inhalte des Empfangspuffers (buffer receive) nach PA transferiert.

**DP-Zyklus**

In einem PROFIBUS-Zyklus spricht der Master alle seine Slaves der Reihe nach mit einem Data Exchange an. Beim Data Exchange werden die dem PROFIBUS zugeordneten Speicherbereiche geschrieben bzw. gelesen. Danach wird der Inhalt des PROFIBUS-Eingangsbereichs in den Empfangspuffer (buffer receive) geschrieben und die Daten des Sendepuffers (buffer send) in den PROFIBUS-Ausgangsbereich übertragen. Der Datenaustausch zwischen DP-Master und DP-Slave über den Bus erfolgt zyklisch, unabhängig vom V-Bus-Zyklus

- V-Bus-Zyklus  $\leq$  DP-Zyklus** Zur Gewährleistung einer zeitgleichen Datenübertragung sollte die V-Bus-Zykluszeit immer kleiner oder gleich der DP-Zykluszeit sein.  
In der mitgelieferten GSD-Datei befindet sich der Parameter **min\_slave\_interval = 3ms**.  
Für einen durchschnittlichen Aufbau wird garantiert, dass spätestens nach 3ms die PROFIBUS-Daten am V-Bus aktualisiert wurden. Sie dürfen also alle 3ms einen Data Exchange mit dem Slave ausführen.
- Datenkonsistenz** Daten bezeichnet man als konsistent, wenn sie inhaltlich zusammengehören. Inhaltlich gehören zusammen: das High- und Low-Byte eines Analogwerts (wortkonsistent) und das Kontroll- und Status-Byte mit zugehörigem Parameterwort für den Zugriff auf die Register.  
Die Datenkonsistenz ist im Zusammenspiel von Peripherie und Steuerung ist grundsätzlich nur für 1Byte sichergestellt. Das heißt, die Bits eines Bytes werden zusammen eingelesen bzw. ausgegeben. Für die Verarbeitung digitaler Signale ist eine byteweise Konsistenz ausreichend.  
Für Daten, deren Länge ein Byte überschreiten, wie z.B. bei Analogwerten muss die Datenkonsistenz erweitert werden. PROFIBUS garantiert die Konsistenz mit der erforderlichen Länge.  
Bitte beachten Sie, dass Sie die konsistenten Daten auf die richtige Art vom PROFIBUS-Master in Ihre SPS übernehmen.  
Hinweise hierzu finden Sie im Handbuch zu ihrem PROFIBUS-Master.
- Einschränkungen** Bei Ausfall eines übergeordneten Masters wird dies von der CPU nicht automatisch erkannt. Für diesen Fall sollten Sie immer ein Kontroll-Byte mitschicken, das die Präsenz des Masters mitteilt und somit gültige Masterdaten kennzeichnet.  
Den Einsatz des Kontroll-Bytes finden Sie auch im Beispiel am Ende dieses Kapitels.
- Diagnose** Die umfangreichen Diagnosefunktionen unter PROFIBUS-DP ermöglichen eine schnelle Fehlerlokalisierung. Die Diagnosedaten werden über den Bus übertragen und beim Master zusammengefasst.

---

**Übertragungs-  
medium**

PROFIBUS verwendet als Übertragungsmedium eine geschirmte, verdrehte Zweidrahtleitung auf Basis der RS485-Schnittstelle oder eine Duplex-Lichtwellenleitung (LWL). Die Übertragungsrate liegt bei beiden Systemen bei maximal 12MBaud.

**Elektrisches  
System über  
RS485**

Die RS485-Schnittstelle arbeitet mit Spannungsdifferenzen. Sie ist daher unempfindlicher gegenüber Störeinflüssen als eine Spannungs- oder Stromschnittstelle. Sie können das Netz sowohl als Linien-, als auch als Baumstruktur konfigurieren. Auf Ihrem VIPA PROFIBUS-Koppler befindet sich eine 9-polige Buchse. Über diese Buchse koppeln Sie den PROFIBUS-Koppler als Slave direkt in Ihr PROFIBUS-Netz ein.

Die Busstruktur unter RS485 erlaubt das rückwirkungsfreie Ein- und Auskoppeln 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.

**Optisches  
System über  
Lichtwellenleiter**

Das Lichtwellenleitersystem arbeitet mit Lichtimpulsen von monochromatischem Licht. Der Lichtwellenleiter ist völlig unempfindlich gegenüber Störspannungen von außen. Ein Lichtwellenleitersystem wird in Linienstruktur aufgebaut. Jedes Gerät ist mit einem Hin- und Rückleiter zu verbinden. Ein Abschluss am letzten Gerät ist nicht erforderlich.

Das rückwirkungsfreie Ein- und Auskoppeln von Stationen ist aufgrund der Linienstruktur nicht möglich.

---

**Adressierung**

Jeder Teilnehmer am PROFIBUS identifiziert sich mit einer Adresse. Diese Adresse darf nur einmal in diesem Bussystem vergeben sein und kann zwischen 0 und 125 liegen.

Bei der CPU 21xDP stellen Sie die Adresse über Ihr Projektiertool ein.

**GSD-Datei**

Zur Konfiguration einer Slave-Anschaltung in Ihrem eigenen Projektiertool bekommen Sie die Leistungsmerkmale der VIPA-Komponenten in Form einer GSD-Datei mitgeliefert.

Aufbau und Inhalt der GSD-Datei sind durch die PROFIBUS Nutzerorganisation (PNO) genormt und können dort jederzeit abgerufen werden.

Installieren Sie diese GSD-Datei in Ihrem Projektiertool. Nähere Hinweise zur Installation der GSD-Datei finden Sie auf den Folgeseiten unter "Projektierung der CPU 21xDP" oder im Handbuch zu Ihrem Projektiertool.

## Projektierung der CPU 21xDP

### Übersicht

Im Gegensatz zum VIPA PROFIBUS-Slave IM 253DP, ist der PROFIBUS-Koppler in der CPU 21xDP ein "intelligenter Koppler".

Der "Intelligente Koppler" verarbeitet Daten, die in einem Ein- bzw. Ausgabe-Bereich der CPU stehen. Diesen Bereich und einen Bereich für Status- und Diagnose-Daten geben Sie in den CPU 21xDP-Eigenschaften an. Für Ein- bzw. Ausgabe-Daten werden getrennte Speicher-Bereiche genutzt. Die Bereiche sind mit Ihrem SPS-Programm zu bedienen.

Die Adress-Bereiche, die der Koppler belegt, werden systembedingt im Hardware-Konfigurator von Siemens nicht angezeigt. Da die direkt gesteckten System 200V Module auch in den Peripherie-Adressraum eingebunden werden, kann dies bei der automatischen Adressierung zu Adressüberschneidungen führen.



### Hinweis!

Zur Projektierung der CPU und des PROFIBUS-DP-Masters werden fundierte Kenntnisse im Umgang mit dem Siemens SIMATIC Manager und dem Hardware-Konfigurator von Siemens vorausgesetzt!

### Möglichkeit zur Projektierung im Siemens SIMATIC Manager

Die Adresszuordnung und die Parametrierung der direkt gesteckten Module erfolgt im Siemens SIMATIC Manager in Form eines virtuellen PROFIBUS-Systems.

Da die PROFIBUS-Schnittstelle auch softwareseitig standardisiert ist, können wir auf diesem Weg gewährleisten, dass über die Einbindung einer GSD-Datei vipa\_21x.gsd die Funktionalität in Verbindung mit dem Siemens SIMATIC Manager jederzeit gegeben ist.

### Schritte der CPU 21xDP-Projektierung

Um kompatibel mit dem Siemens SIMATIC Manager zu sein, sind für das System 200V folgende Schritte durchzuführen:

- CPU 315-2DP mit DP-Master-System projektieren (Adresse 2).
- Aus VIPA\_21x.gsd den PROFIBUS-Slave "VIPA\_CPU21x" mit Adresse 1 anfügen.
- Auf dem 1. Steckplatz des Slave-Systems die CPU 21xDP einbinden.
- PROFIBUS-Parameter der CPU 21xDP einstellen.
- Dahinter die direkt gesteckte Peripherie-Module einbinden.
- Projektierung via MPI in die CPU 21xDP übertragen.

### Schritte der Master-Projektierung

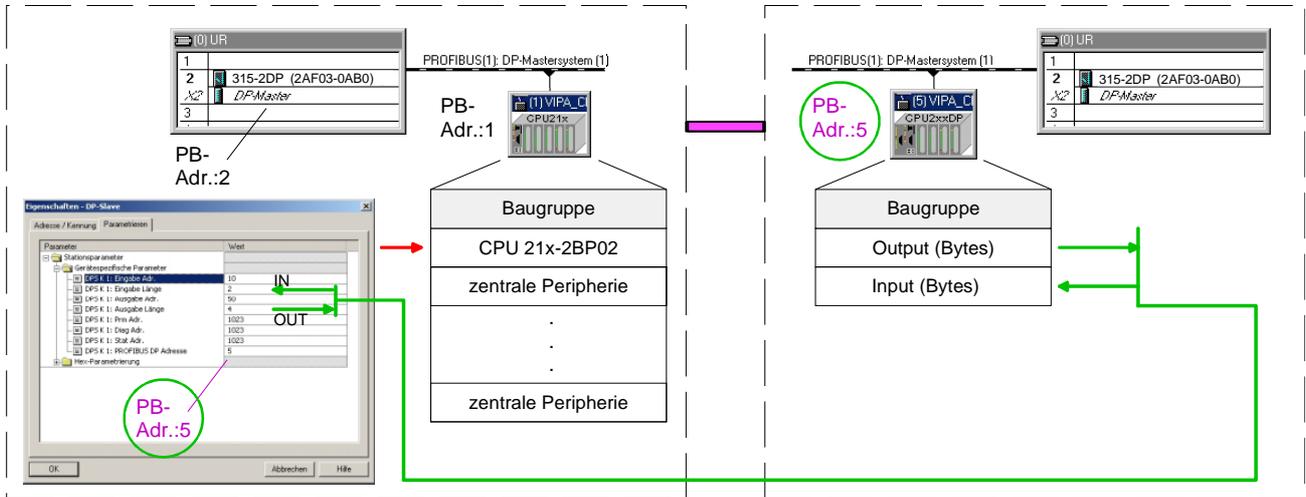
- CPU mit DP-Master-System projektieren (Adresse 2).
- Aus VIPA04D5.gsd den PROFIBUS-Slave "VIPA\_CPU2xxDP" anfügen.
- Die PROFIBUS Ein- und Ausgabebereiche ab dem 1. Steckplatz angeben. Hierbei müssen die Längenangaben auf Slave-Seite mit den Byteangaben auf der Master-Seite übereinstimmen!

**Bezug zwischen Master und Slave**

In der nachfolgenden Abbildung ist die Projektierung auf Slave- und Masterseite nochmals zusammengefasst:

**Slave: (VIPA\_CPU21x aus VIPA\_21x.gsd)**

**Master: (VIPA\_CPU21xDP aus VIPA04d5.gsd)**



**Projektierung CPU 21xDP**

Nachfolgend werden die einzelnen Schritte der Slave-Projektierung aufgezeigt.

**Voraussetzungen**

Für die Projektierung der CPU 21xDP in einem System 200V bzw. System 300 Master-System müssen folgende Voraussetzungen erfüllt sein:

- Siemens SIMATIC Manager ist installiert.
- GSD-Datei der CPU21xDP in Hardware-Konfigurator ist eingebunden.
- Transfermöglichkeit zwischen Projektiertool und CPUs ist vorhanden.

**Hardware-Konfigurator von Siemens installieren**

Der Hardware-Konfigurator ist Bestandteil des Siemens SIMATIC Manager. Er dient der Projektierung. Die Module, die hier projiziert werden können, entnehmen Sie dem Hardware-Katalog.

Für den Einsatz der PROFIBUS-DP-Slaves von VIPA ist die Einbindung der Module über die GSD-Dateien von VIPA im Hardwarekatalog erforderlich.

**GSD einbinden**

Für die Installation einer GSD-Datei wird vorausgesetzt, dass die GSD-Datei in entpackter Form als .gsd vorliegt.

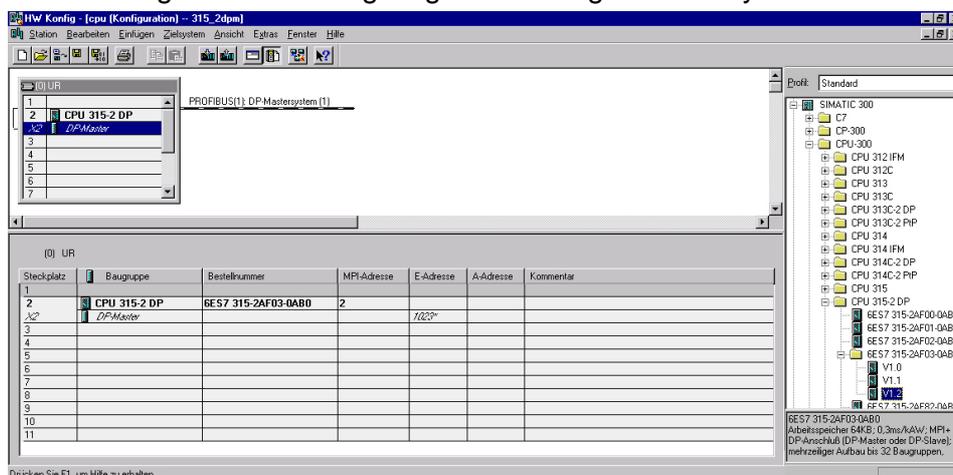
- Starten Sie den Hardware-Konfigurator von Siemens. Zur Einbindung einer neuen GSD darf kein Projekt geöffnet sein.
- Öffnen Sie über **Extras** > *Neue GSD installieren* das Dialogfenster
- Navigieren Sie zur gewünschten GSD-Datei und Installieren Sie die GSD-Datei mit [Öffnen]

Die Module der GSD-Datei finden Sie nun im Hardware-Katalog unter *PROFIBUS-DP* > *Weitere Feldgeräte* > *I/O* > *VIPA*.

## Virtuelles PROFIBUS-System erzeugen

- Legen Sie ein neues Projekt System 300 an und fügen Sie aus dem Hardwarekatalog eine Profilschiene ein.
- Fügen Sie die CPU 315-2DP ein. Sie finden die CPU mit PROFIBUS-Master im Hardwarekatalog unter: *Simatic300 > CPU-300 > CPU315-2DP > 6ES7 315-2AF03-0AB0 V1.2*
- Geben Sie Ihrem Master die PROFIBUS-Adresse 2
- Klicken Sie auf DP und stellen Sie in unter *Objekteigenschaften* die Betriebsart "DP-Master" ein und bestätigen Sie Ihre Eingabe mit [OK].
- Durch Klick mit der rechten Maustaste auf "DP" öffnet sich das Kontextmenü. Wählen Sie "Mastersystem einfügen" aus. Legen Sie über NEU ein neues PROFIBUS-Subnetz an.

Die nachfolgende Abbildung zeigt das erzeugte Mastersystem:



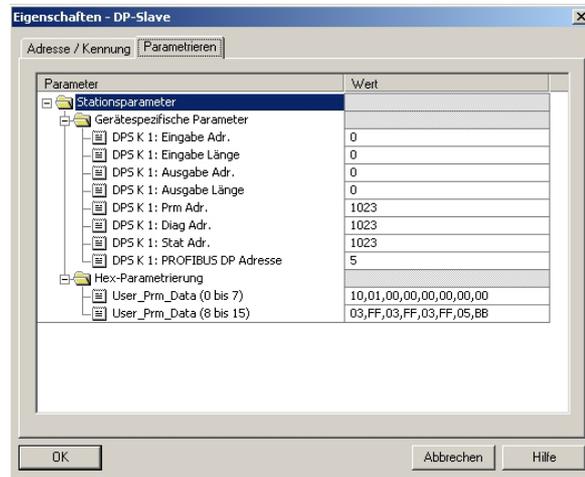
## CPU 21xDP und Module projektieren

Um, wie schon weiter oben erwähnt, zum Siemens SIMATIC Manager kompatibel zu sein, müssen Sie die CPU 21xDP explizit einbinden.

- Hängen Sie an das Subnetz das System "VIPA\_CPU21x". Sie finden dies im Hardware-Katalog unter *PROFIBUS DP > Weitere Feldgeräte > I/O > VIPA\_System\_200V*. Geben Sie dem DP-Slave die PROFIBUS-Adresse 1.
- Platzieren Sie in Ihrem Hardware-Konfigurator auf dem 1. Steckplatz die CPU 21x-2BP02 von VIPA.
- Im CPU-Parameterfenster können Sie die Datenbereiche des PROFIBUS-Teils einstellen. Näheres hierzu finden Sie unter "PROFIBUS-Teil einbinden".
- Binden Sie Ihre System 200V Module in der gesteckten Reihenfolge ein.
- Falls sich zusätzlich ein DP-Master auf dem Rückwandbus befindet, ist dieser ebenfalls auf dem entsprechenden Steckplatz einzubinden.
- Sichern Sie Ihr Projekt und übertragen Sie dieses via MPI in die CPU 21xDP.

**PROFIBUS-Teil parametrieren**

Der PROFIBUS-Teil blendet seine Datenbereiche im Speicherbereich der CPU 21xDP ein. Die Zuordnung der Bereiche führen Sie in den Eigenschaften der CPU 21xDP durch. Über einen Doppelklick auf die CPU 21xDP gelangen Sie in das Dialogfenster zur Parametrierung der Datenbereiche für den PROFIBUS-Slave. Näheres hierzu finden Sie im Kapitel "DP-Slave Parameter".

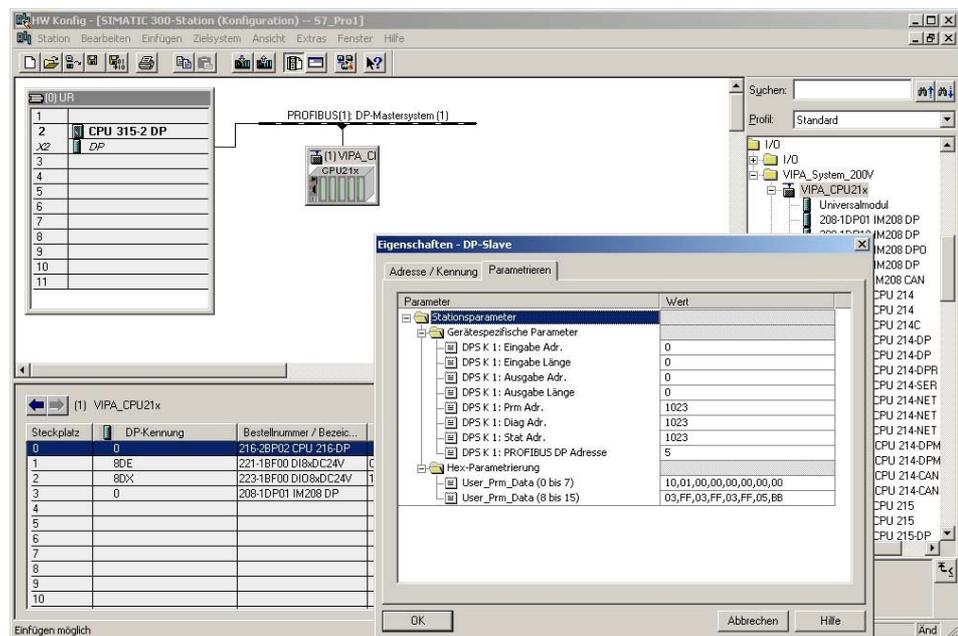


**Achtung!**

Bitte beachten Sie, dass die Längenangaben der Datenbereiche bei Master- und Slave-Projektierung identisch sind.

Die Datenbereiche, die der PROFIBUS-Teil in der CPU belegt, können systembedingt nur im CPU-Parametrierfenster angezeigt werden.

Nachfolgend sind alle relevanten Dialogfenster der Slave-Projektierung aufgeführt. Hier sehen Sie auch, auf welche Weise ihr System 200V einzubinden ist:

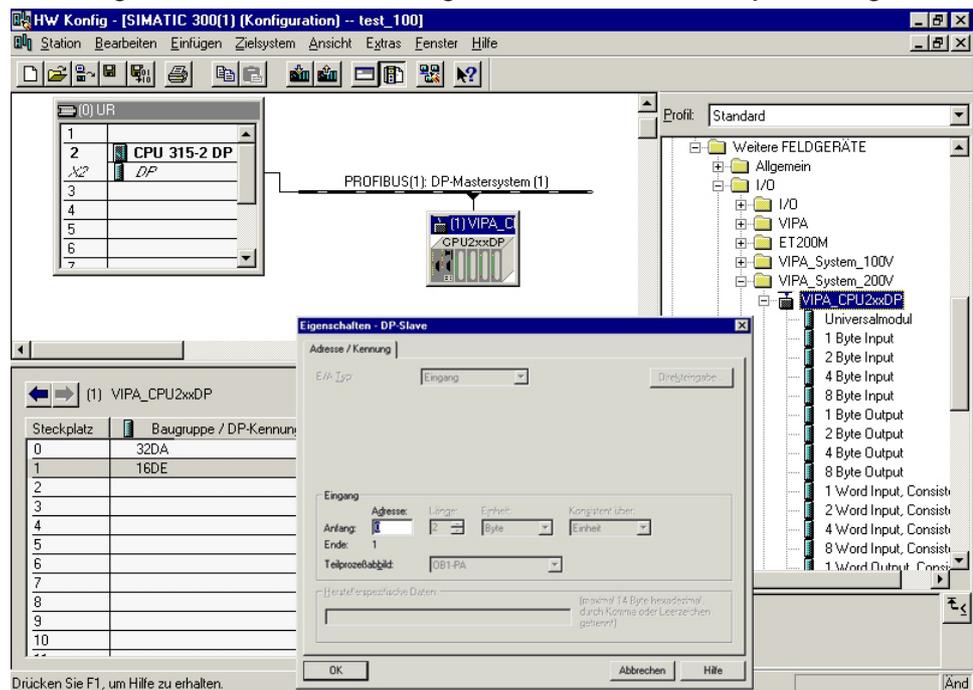


## Projektierung in einem Master-System

Zur Projektierung in einem übergeordneten Master-System ist die Einbindung der GSD VIPA04d5.gsd erforderlich.

- Starten Sie Ihr Projektierool und projektieren Sie Ihren PROFIBUS-DP-Master, der Ihrer CPU 21xDP übergeordnet ist.
- Fügen Sie an das Mastersystem ein DP-Slave-System "CPU2xxDP" an. Sie finden, nach Einbindung der GSD, das DP-Slave-System im Hardware-Katalog unter:  
*PROFIBUS-DP > Weitere Feldgeräte > I/O > VIPA > VIPA\_CPU2xxDP.*
- Vergeben Sie für den DP-Slave die gleiche PROFIBUS-Adresse, die Sie auf der Slave-Seite parametriert haben.
- Legen Sie in Form von "Modulen" für die PROFIBUS-Kommunikation den gleichen E/A-Bereich an, den Sie auf der Slave-Seite parametriert haben. Beachten Sie, dass sich ein Slave-Ausgabe- auf einen Master-Eingabe-Bereich bezieht und umgekehrt.
- Speichern Sie Ihr Projekt und übertragen Sie dieses in die CPU Ihres Master-Systems.

Nachfolgend sehen Sie die Dialogfenster der Master-Projektierung:



### Hinweis!

Sollte es sich bei Ihrem DP-Master-System um ein System 200V von VIPA handeln, so können Sie durch Anbindung eines "DP200V"-Slave-Systems die direkt gesteckten Module projektieren. Damit dieses Projekt von der VIPA-CPU als zentrales System erkannt wird, müssen Sie dem "DP200V"-Slave-System die PROFIBUS-Adresse 1 zuweisen!

**Bitte beachten Sie bei Einsatz des IM 208 PROFIBUS-DP-Master, dass dieser einen Firmwarestand ab V 3.0 besitzt; ansonsten kann dieser an der CPU 21x mit Firmwarestand ab V 3.0 nicht betrieben werden. Die Firmwarestände entnehmen Sie bitte dem Aufkleber, der sich auf der Rückseite des jeweiligen Moduls befindet.**

# DP-Slave Parameter

## Übersicht

Beim "intelligenten" Slave blendet der PROFIBUS-Teil seine Datenbereiche im Speicherbereich der CPU 21xDP ein. Die Zuordnung der Bereiche führen Sie in den "Eigenschaften" der CPU 21xDP durch. Die Ein- bzw. Ausgabe-Bereiche sind mit einem entsprechenden SPS-Programm zu versorgen.

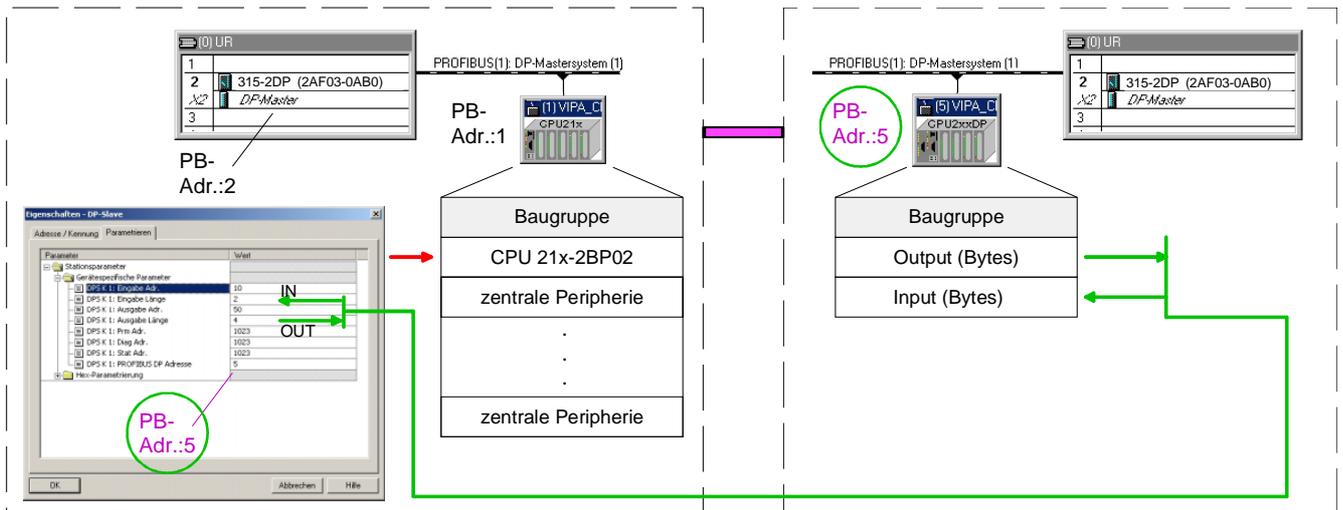


### Achtung!

Die Längenangaben für Ein- und Ausgabe-Bereich müssen mit den Byteangaben bei der Master-Projektierung übereinstimmen. Ansonsten kann keine PROFIBUS-Kommunikation stattfinden und der Master meldet Slave-Ausfall!

Slave: (VIPA\_CPU21x aus VIPA\_21x.gsd)

Master: (VIPA\_CPU21xDP aus VIPA04d5.gsd)



## Bereiche in CPU freigeben

Sobald Sie bei einer Längenangabe 0 angeben, wird für die zugehörigen Daten kein Speicherplatz in der CPU belegt.

Durch Eingabe von 255 (Speichergrenze) bei den Parametern PRN, DIAG und STAT können Sie ebenfalls Speicherbereiche in der CPU freigeben.



### Hinweis!

Bis zur CPU-Firmware-Version V 2.2.0 wird von der CPU 21x und dem PROFIBUS-DP-System ein Adressbereich von 0 bis 255 unterstützt.

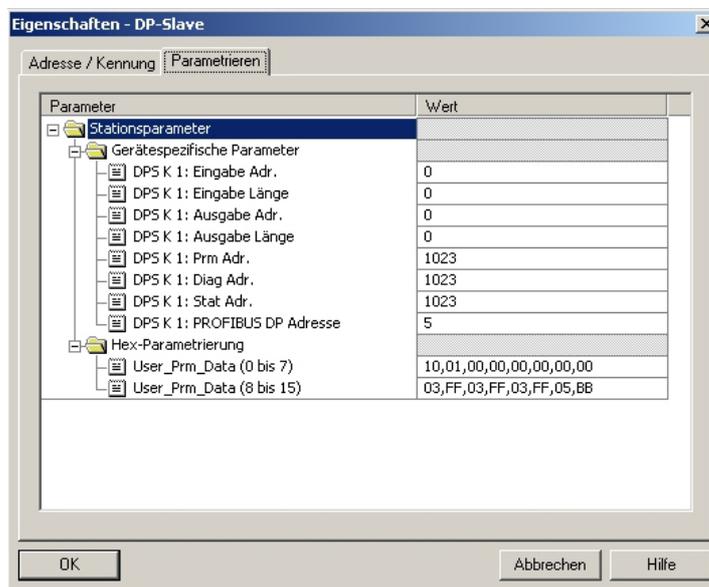
Ab der Firmware-Version V. 3.0 unterstützen CPU 21x und PROFIBUS-DP-System von VIPA einen Adressbereich von 0 bis 1023.

Den Firmwarestand entnehmen Sie bitte dem Aufkleber, der sich an der Rückseite des jeweiligen Moduls befindet.

Hier ist 1023 der Wert zur Deaktivierung von PRN, DIAG und STAT.

## Beschreibung der Parameterdaten

Über einen Doppelklick im Hardware-Konfigurator auf die CPU 21xDP öffnet sich folgendes Dialogfenster zur Parametrierung der Datenbereiche für den PROFIBUS-Slave:



### input adr, len

Adresse, ab der die über PROFIBUS kommenden Daten mit der entsprechenden "Länge" in der CPU abzulegen sind.

Die Längenangabe von 0 belegt für den Eingabe-Bereich keinen Speicherbereich in der CPU.

### output adr, len

Adresse, ab der die Daten ablegen, die über PROFIBUS zu senden sind. Auch hier geben Sie über *len* die Datenbreite vor.

Die Längenangabe von 0 belegt für den Eingabe-Bereich keinen Speicherbereich in der CPU.

### prm. adr.

Die Parameterdaten sind ein Auszug der Parameter-Telegramms. Das Parameter-Telegramm wird bei der Masterprojektierung erzeugt und an den Slave geschickt wenn:

- sich die CPU 21xDP im Hochlauf befindet
- die Verbindung zwischen CPU 21xDP und Master gestört war, wie z.B. kurzzeitiges Abziehen des Bus-Steckers.

Ein Parameter-Telegramm besteht aus profibusspezifischen Daten (Busparameter) und benutzerspezifischen Daten in denen bei der CPU 21xDP die Ein- und Ausgabe-Bytes definiert sind.

Die benutzerspezifischen Daten (Byte 7 ... 31) werden mit einer fixen Länge von 24Byte ab der unter *prm* eingestellten Adresse im Speicherbereich der CPU eingeblendet.

Hiermit können Sie die Parameter überprüfen, die Ihr Slave vom Master erhält.

**diag. adr.**

Die umfangreichen Diagnosefunktionen von PROFIBUS-DP ermöglichen eine schnelle Fehlerlokalisierung. Die Diagnosemeldungen werden über den Bus übertragen und beim Master zusammengefasst.

Die CPU 21xDP sendet auf Anforderung vom Master oder im Fehlerfall Diagnosedaten. Die Diagnosedaten bestehen aus:

- Norm-Diagnose-Daten (Byte 0 ... 5),
- Gerätebezogene Diagnose-Daten (Byte 6 ... 10)
- **Anwenderspezifische Diagnose-Daten (Byte 11 ... 15)**

Über *diag* bestimmen Sie die Startadresse, ab der die 6 Byte breiten anwenderspezifischen Diagnose-Daten in der CPU abzulegen sind.

Durch gezielten Zugriff auf diesen Bereich können Sie Diagnosen auslösen und beeinflussen.

**Hinweis!**

Näheres zum Aufbau und zur Beeinflussung von Diagnosemeldungen finden Sie unter "Diagnosefunktionen".

**stat. adr.**

Den aktuellen Status der PROFIBUS-Kommunikation können Sie einem 2Byte breiten Statusbereich entnehmen, der ab der Statusadresse im Peripherieadressbereich der CPU abliegt.

**Hinweis!**

Näheres zum Aufbau einer Statusmeldung finden Sie unter "Statusmeldung intern an CPU".

**PROFIBUS DP  
address**

Über diesen Parameter weisen Sie Ihrem PROFIBUS-Slave ein PROFIBUS-Adresse zu.

**Bereiche in CPU  
freigeben**

Sobald Sie bei einer Längenangabe 0 angeben, wird für die zugehörigen Daten kein Speicherplatz in der CPU belegt.

Durch Eingabe der Adressbereichsgrenze (255 bzw. 1023 ab CPU-Versionen > 2.2.0) bei den Parametern *PRN*, *DIAG* und *STAT* können Sie ebenfalls Speicherbereiche in der CPU freigeben.

## Diagnosefunktionen

### Übersicht

Die umfangreichen Diagnosefunktionen von PROFIBUS-DP ermöglichen eine schnelle Fehlerlokalisierung. Die Diagnosemeldungen werden über den Bus übertragen und beim Master zusammengefasst.

Die CPU 21xDP sendet auf Anforderung vom Master oder im Fehlerfall Diagnosedaten. Da ein Teil der Diagnosedaten (Byte 11 ... 15) im Peripherieadressbereich der CPU liegt, können Sie eine Diagnose auslösen und Diagnosedaten beeinflussen. Die Diagnosedaten bestehen aus:

- Norm-Diagnose-Daten (Byte 0 ... 5),
- Gerätebezogene Diagnose-Daten (Byte 6 ... 15).

### Aufbau

Die Diagnosedaten haben folgenden Aufbau:

#### *Norm-Diagnosedaten*

Byte 0	Stationsstatus 1
Byte 1	Stationsstatus 2
Byte 2	Stationsstatus 3
Byte 3	Master-Adresse
Byte 4	Ident-Nummer (low)
Byte 5	Ident-Nummer (high)

#### *Gerätebezogene Diagnosedaten*

Byte 6	Länge und Code gerätebezogene Diagnose
Byte 7	Gerätebezogene Diagnosemeldungen
Byte 8 ... Byte 10	reserviert
Byte 11 ... Byte 15	Anwenderspezifische Diagnosedaten werden in CPU-Peripherieadressbereich eingeblendet und können bearbeitet und an den Master geschickt werden.

**Norm-Diagnosedaten**

Nähere Angaben zum Aufbau der Norm-Diagnosedaten finden Sie in den PROFIBUS-Norm-Schriften. Die Normschriften sind bei der PROFIBUS Nutzer Organisation erhältlich.

Die Slave-Normdiagnosedaten haben folgenden Aufbau:

Byte	Bit 7 ... Bit 0
0	Bit 0: fest auf 0 Bit 1: Slave nicht bereit für Datenaustausch Bit 2: Konfigurationsdaten stimmen nicht überein Bit 3: Slave hat externe Diagnosedaten Bit 4: Slave unterstützt angeforderte Funktion nicht Bit 5: fest auf 0 Bit 6: Falsche Parametrierung Bit 7: fest auf 0
1	Bit 0: Slave muss neu parametrieren werden Bit 1: Statistische Diagnose Bit 2: fest auf 1 Bit 3: Ansprechüberwachung aktiv Bit 4: Freeze-Kommando erhalten Bit 5: Sync-Kommando erhalten Bit 6: reserviert Bit 7: fest auf 0
2	Bit 0 ... Bit 6: reserviert Bit 7: Diagnosedaten Überlauf
3	Masteradresse nach Parametrierung FFh: Slave ist ohne Parametrierung
4	Identnummer High Byte
5	Identnummer Low Byte

**gerätebezogene  
Diagnosedaten**

Die gerätebezogenen Diagnosedaten geben detaillierte Auskunft über den Slave und die Peripherie-Module. Die Länge der gerätebezogenen Diagnosedaten ist fest auf 10Byte eingestellt.

Byte	Bit 7 ... Bit 0
6	Bit 0 ... 5: Länge gerätebezogene Diagnosedaten 001010: Länge 10Byte (fest) Bit 6 ... 7: Code für gerätebezogene Diagnose 00: Code 00 (fest)
7	Bit 0 ... Bit 7: Gerätebezogene Diagnosemeldung 12h: Fehler: Parameterdatenlänge 13h: Fehler: Konfigurationsdatenlänge 14h: Fehler: Konfigurationseintrag 15h: Fehler: VPC3 Pufferberechnung 16h: Fehler: fehlende Konfigurationsdaten 17h: Fehler: Abgleich DP-Parametrierung mit Projektierung 40h: Benutzerdefinierte Diagnose gültig
8 ... 10	reserviert
<b>11 ... 15</b>	<b>Anwenderspezifische Diagnosedaten, die nach dem Diagnose-Statusbyte im Prozessabbild der CPU abgelegt werden. Diese können überschrieben und an den Master weitergeleitet werden.</b>

**Diagnose  
auslösen**

Im Diagnosefall werden die Inhalte von Byte 11...15 der gerätebezogenen Diagnosedaten in das Prozessabbild der CPU übertragen und diesen ein Statusbyte vorangestellt. Die Lage dieses 6Byte langen Diagnoseblocks im Prozessabbild der CPU können Sie in der CPU Parameter-Einstellung definieren.

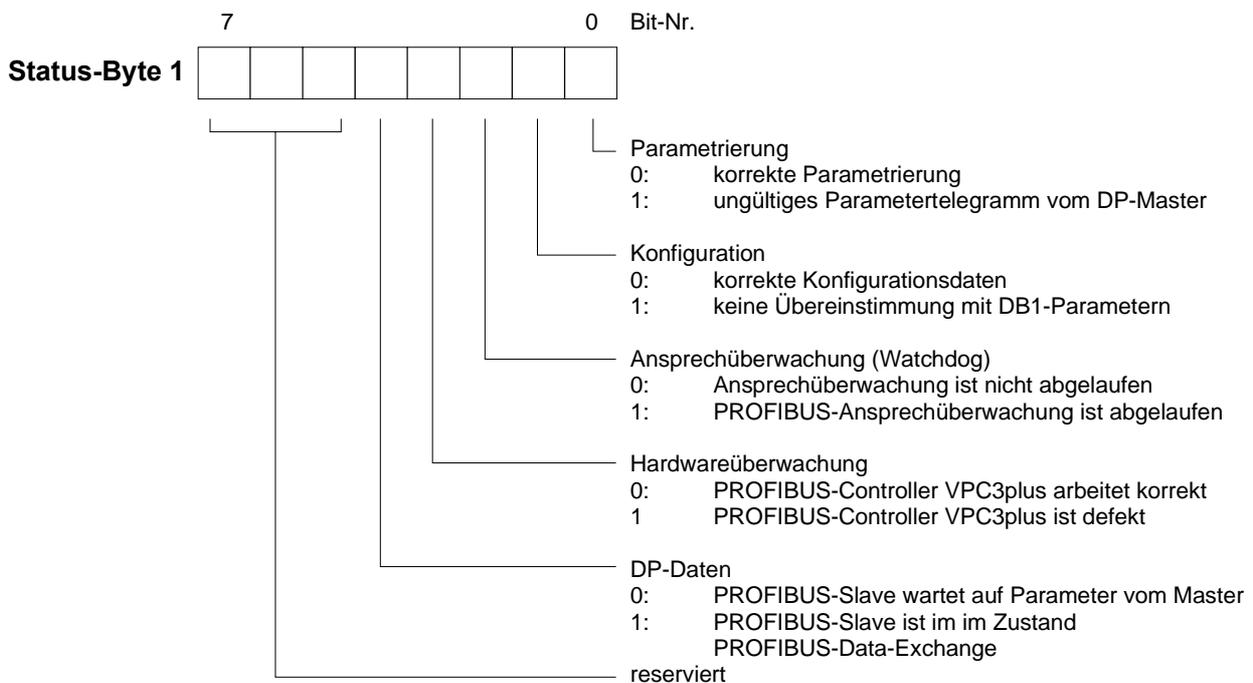
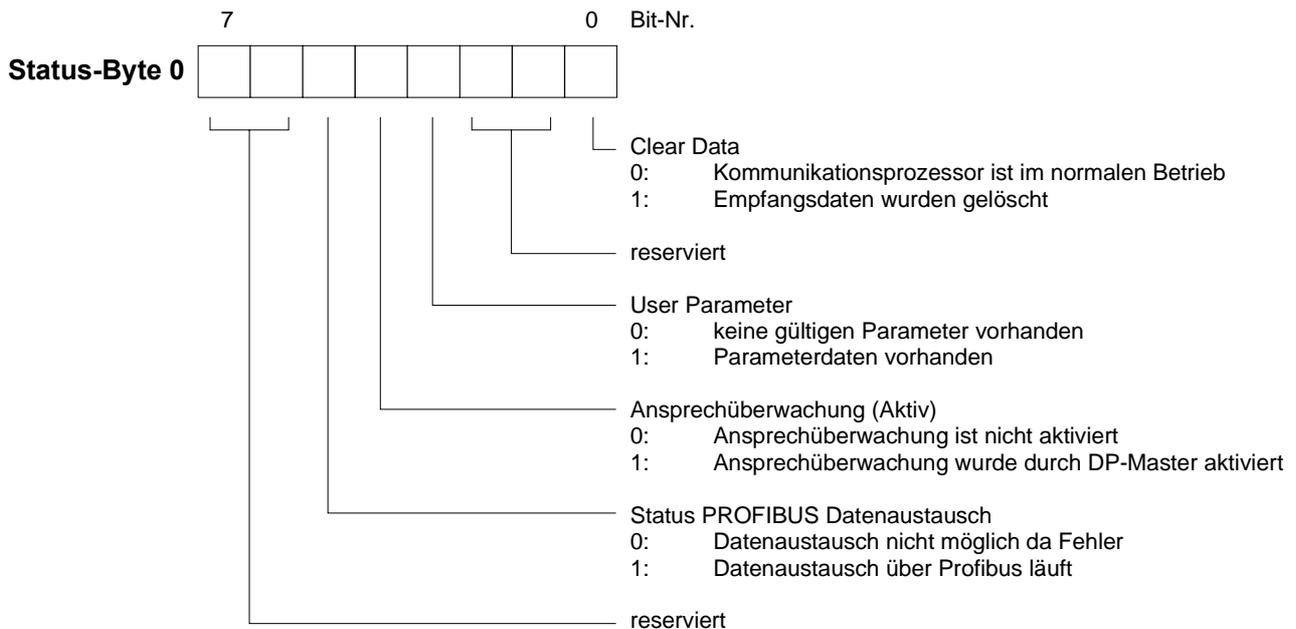
Durch Zustandswechsel von 0 → 1 im Diagnose-Statusbyte lösen Sie eine Diagnose aus und das entsprechende Diagnose-Telegramm wird an den Master übertragen. **Der Zustand 0000 0011 wird ignoriert!**

Der Diagnoseblock in der CPU hat folgenden Aufbau:

Byte	Bit 7 ... Bit 0
0	Diagnose-Statusbyte: Bit 0: anwenderspezifische Diagnosedaten 0: ungültige Diagnosedaten 1: gültige Diagnosedaten (Auslösen einer Diagnose) Bit 1: Diagnose löschen 0: Diagnose löschen ungültig 1: Diagnose löschen gültig Bit 2 ... Bit 7: reserviert
1 ... 5	Bit 0 ... Bit 7: Anwenderspezifische Diagnosedaten entspricht Byte 11 ... 15 der gerätebezogenen Diagnose

## Statusmeldung intern an CPU

Den aktuellen Status der PROFIBUS-Kommunikation finden Sie in den Statusmeldungen, die in den Peripherieadressbereich der CPU eingebunden sind. Die Statusmeldungen bestehen aus 2Byte und haben folgenden Aufbau:



---

**Parameter**

<b>Clear Data</b>	Im Fehlerfall werden die Sende- und Empfangspuffer gelöscht.
<b>reserviert</b>	Diese zwei Bits sind für zukünftige Erweiterungen reserviert.
<b>User Parameter</b>	Zeigt die Gültigkeit der Parameterdaten an. Die Parameterdaten werden im Master-Parametriertool eingegeben.
<b>Ansprechüberwachung (Aktiv)</b>	Zeigt den Zustand der Aktivierung der Ansprechüberwachung im übergeordneten PROFIBUS-Master an. Bei überschrittener Ansprechüberwachungszeit bricht der Slave die Kommunikation ab.
<b>Status PROFIBUS Datenaustausch</b>	Statusanzeige über die Kommunikation mit dem übergeordneten Master. Bei fehlerhafter Konfiguration oder bei fehlerhaften Parametern wird die Kommunikation unterbrochen und der Fehler über dieses Bit angezeigt.
<b>Parametrierung</b>	Zeigt den Status der Parametrierdaten an. Die Länge der Parametrierdaten und die Anzahl der Parametrier-Bytes wird ausgewertet. Nur wenn diese gleich sind und nicht mehr als 31Byte Parameterdaten übertragen werden, ist die Parametrierung korrekt.
<b>Konfiguration</b>	Statusanzeige der Konfigurationsdaten, die vom PROFIBUS-Master geschickt werden. Die Konfiguration erstellen Sie im Master Projektier-Tool.
<b>Ansprechüberwachung (Watchdog)</b>	Hier wird der Zustand der Ansprechüberwachung im PROFIBUS-Master angezeigt. Bei aktivierter Ansprechüberwachung und überschrittener Ansprechzeit im Slave wird hier ein Fehler angezeigt.
<b>Hardwareüberwachung</b>	Ein gesetztes Bit zeigt hier an, dass der PROFIBUS-Controller in der CPU 21xDP defekt ist. Kontaktieren Sie in diesem Fall die VIPA Hotline.
<b>DP-Daten</b>	Bei jedem Transferfehler über PROFIBUS wird dieses Bit gesetzt.

## PROFIBUS Aufbaurichtlinien

### 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 ... 187,5kBaud	→	1000m
500kBaud	→	400m
1,5Mbaud	→	200m
3 ... 12Mbaud	→	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 Baudrate. Die Slaves passen sich automatisch an die Baudrate an.



#### Hinweis!

Sie sollten bei optischen Teilnehmern am Busende die Buchse für den nachfolgenden Teilnehmer abdecken, ansonsten besteht Blendungsgefahr und das Empfangsteil kann durch Fremdeinstrahlung gestört werden. Verwenden Sie hierzu die mitgelieferten Gummi-Stäbchen und stecken Sie die Stäbchen in die übrigen zwei Öffnungen des LWL-Anschlusses.

### Übertragungs- medium

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.

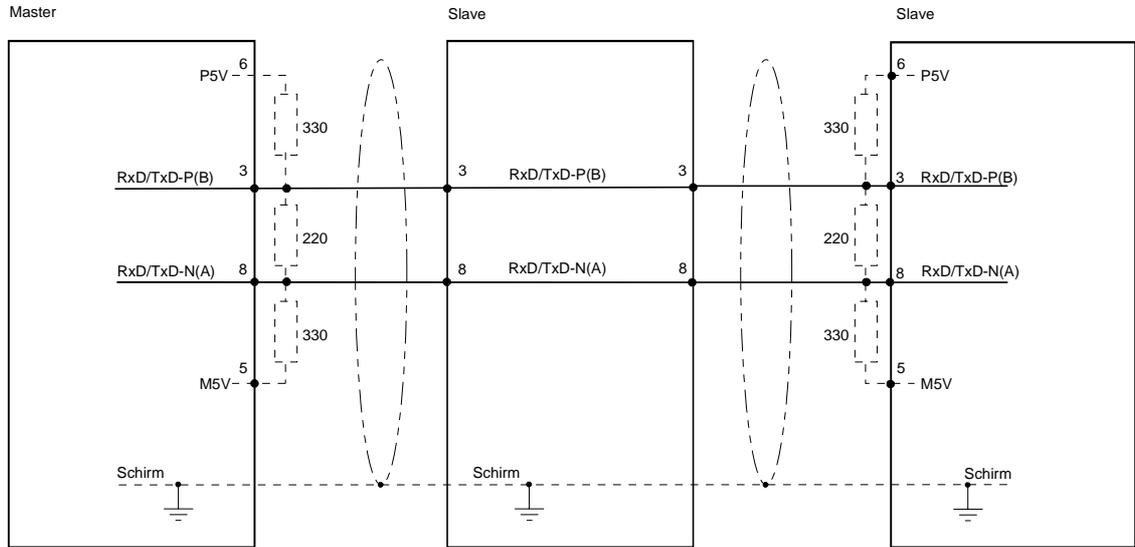
Pro Segment sind maximal 32 Teilnehmer zulässig. Innerhalb eines Segment sind die einzelnen Teilnehmer über Linienstruktur zu verbinden. Die einzelnen Segmente werden über Repeater verbunden. Die max. Segmentlänge ist von der Übertragungsrate abhängig.

Bei PROFIBUS-DP wird die Übertragungsrate aus dem Bereich zwischen 9,6kBaude bis 12Mbaud eingestellt, die Slaves passen sich automatisch an. Alle Teilnehmer im Netz kommunizieren mit der gleichen Übertragungsrate.

Die Busstruktur erlaubt das rückwirkungsfreie Ein- und Auskoppeln 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.



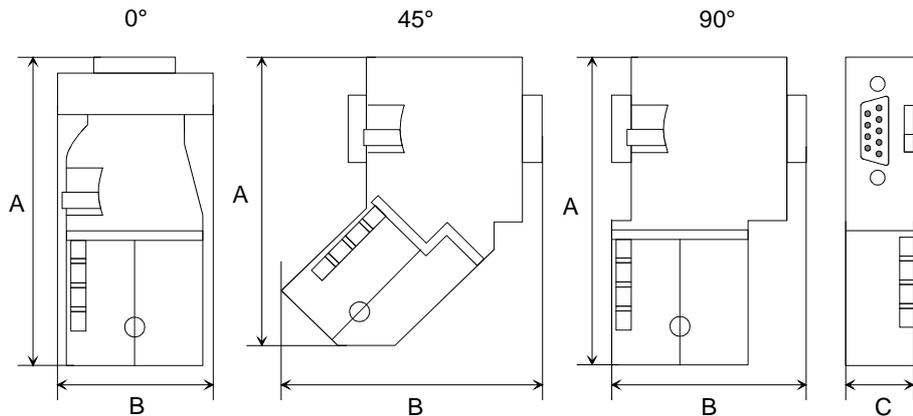
**Hinweis!**

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 Busanschlussstecker**

In PROFIBUS werden alle Teilnehmer parallel verdrahtet. Hierzu ist das Buskabel durchzuschleifen.

Unter der Best.-Nr. VIPA 972-0DP10 erhalten Sie von VIPA den Stecker "EasyConn". Dies ist ein Busanschlussstecker mit zuschaltbarem Abschlusswiderstand und integrierter Busdiagnose.



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

Maße in mm



**Hinweis!**

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. VIPA 905-6AA00 das "EasyStrip" Abisolierwerkzeug, das Ihnen den Anschluss des EasyConn-Steckers sehr vereinfacht.



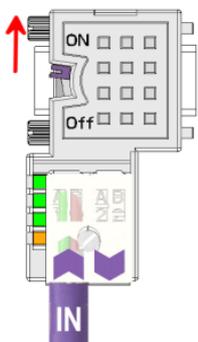
Maße in mm

Leitungsabschluss mit "EasyConn"

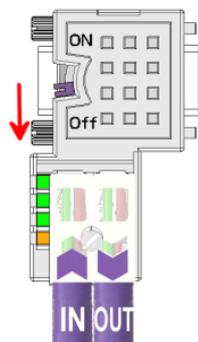
Auf dem "EasyConn" Busanschlussstecker von VIPA befindet sich unter anderem ein Schalter, mit dem Sie einen Abschlusswiderstand zuschalten können.

**Verdrahtung**

1./letzter Bus-Teilnehmer



weiterer Bus-Teilnehmer



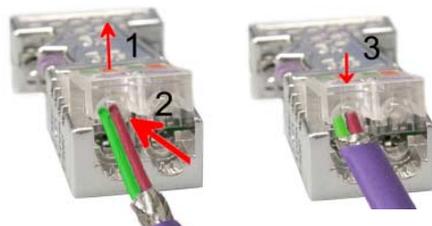
**Achtung!**

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

**Hinweis!**

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

**Montage**



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

**Bitte beachten:**

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

## Inbetriebnahme

### Übersicht

- Bauen Sie Ihre CPU 21xDP auf.
- Projektieren Sie die CPU 21xDP in Ihrem Mastersystem.
- Projektieren Sie die über den Rückwandbus verbundene E/A-Peripherie.
- Verbinden Sie Ihre CPU 21xDP mit Ihrem PROFIBUS.
- Schalten Sie die Spannungsversorgung ein.
- Übertragen Sie Ihr Projekt in Ihre CPUs.

### Aufbau

Bauen Sie Ihre CPU 21xDP mit den gewünschten Peripherie-Modulen auf. Bitte beachten Sie hierbei die maximale Strombelastung Ihrer Spannungsversorgung.



#### Hinweis!

An den Leitungsenden muss das Buskabel immer mit dem Wellenwiderstand abgeschlossen werden um Reflexionen und damit Übertragungsprobleme zu vermeiden!

### Projektierung im Mastersystem

Projektieren Sie Ihre CPU 21xDP in Ihrem Master-System. Zur Projektierung der System 200V PROFIBUS-Slaves von VIPA ist die Einbindung der zugehörigen GSD erforderlich.

### Projektierung CPU 21xDP und E/A-Peripherie

Die System 200V Peripherie-Module, die direkt über den Rückwandbus mit der CPU 21xDP verbunden sind, werden automatisch in den CPU-Adressbereich eingeblendet. Die Adresszuweisung können Sie jederzeit im Hardware-Konfigurator von Siemens ändern.

### Spannungsversorgung

Die CPU 21xDP besitzt ein eingebautes Netzteil. Das Netzteil ist mit 24V Gleichspannung zu versorgen.

Über die Versorgungsspannung werden neben der CPU und dem Buskoppler auch die angeschlossenen Module über den Rückwandbus versorgt. Bitte beachten Sie, dass das integrierte Netzteil den Rückwandbus mit maximal 3A versorgen kann.

PROFIBUS und Rückwandbus sind galvanisch voneinander getrennt.

- Projekt übertragen** Die Übertragung der Hardware-Konfiguration in Ihre CPU erfolgt unter MPI.
- Verbinden Sie Ihr PG bzw. Ihren PC über MPI mit Ihrer CPU.  
Sollte Ihr Programmiergerät keine MPI-Schnittstelle besitzen, können Sie, für eine serielle Punkt-zu-Punkt-Übertragung von Ihrem PC an MPI, das "Green Cable" von VIPA verwenden.  
Das "Green Cable" hat die Best.-Nr. VIPA 950-0KB00 und darf nur bei den VIPA CPUs mit MP<sup>2</sup>I-Schnittstelle eingesetzt werden.
  - Konfigurieren Sie die MPI-Schnittstelle Ihres PCs.
  - Mit **Zielsystem** > *Laden in Baugruppe* in Ihrem Projektierool übertragen Sie Ihr Projekt in die CPU.
  - Zur zusätzlichen Sicherung Ihres Projekts auf MMC stecken Sie eine MMC und übertragen Sie mit **Zielsystem** > *RAM nach ROM kopieren* Ihr Anwenderprogramm auf die MMC.  
Während des Schreibvorgangs blinkt die "MC"-LED auf der CPU. Systembedingt wird ein zu früh ein erfolgter Schreibvorgang gemeldet. Der Schreibvorgang ist erst beendet, wenn die LED erlischt.

**Achtung!**

Bitte beachten Sie die Hinweise im Teil "Einsatz CPU 21x" zum Einsatz der MP<sup>2</sup>I-Buchse und des Green Cables!

**Initialisierungsphase**

Nach dem Einschalten durchläuft der PROFIBUS-Koppler einen Selbsttest. Hierbei überprüft er seine internen Funktionen, die Kommunikation über den Rückwandbus und die Kommunikation zum PROFIBUS.

Bei erfolgreichem Test werden die Parameter aus der CPU gelesen und die PROFIBUS-Slave-Parameter geprüft.

Nach fehlerfreiem Hochlauf geht der Buskoppler in den Zustand "READY" über.

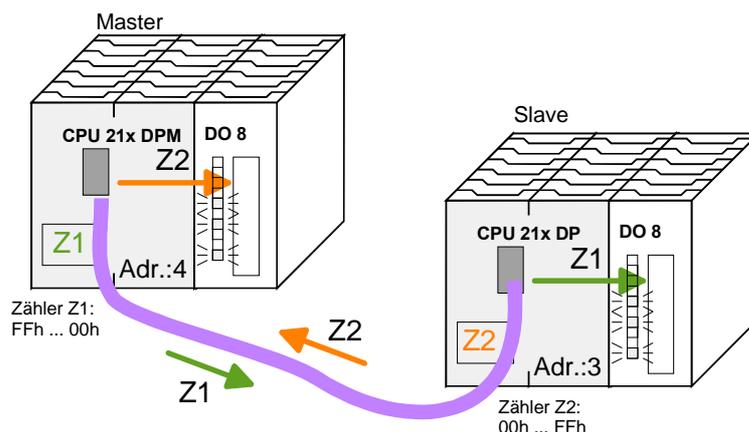
Bei Kommunikationsstörungen am Rückwandbus geht der PROFIBUS-Koppler zunächst in STOP und läuft nach ca. 2 Sekunden erneut hoch. Sobald der Test positiv abgeschlossen ist, blinkt die RD-LED.

Bei beginnender Kommunikation leuchtet die DE-LED.

## Beispiel

**Aufgabenstellung** In diesem Beispiel soll eine Kommunikation zwischen einer Master-CPU CPU 214DPM und einer Slave-CPU CPU 214DP gezeigt werden. Hierbei sollen Zählerstände über den PROFIBUS ausgetauscht und diese auf dem Ausgabe-Modul des jeweiligen Partners dargestellt werden.

**Aufgabenstellung im Detail** Die CPU 214DPM soll von FFh ... 00h zählen und den Zählerstand zyklisch in den Ausgabebereich des PROFIBUS-Masters übertragen. Der Master hat diesen Wert an den Slave der CPU 214DP zu schicken. Der empfangene Wert soll in der CPU im Eingangs-Peripheriebereich abgelegt und über den Rückwandbus auf dem Ausgabe-Modul (auf Adresse 0) ausgegeben werden. Umgekehrt soll die CPU 214DP von 00h bis FFh zählen. Auch dieser Zählerstand ist im Ausgabe-Bereich des CPU-Slaves abzulegen und über den PROFIBUS in den Master zu transferieren. Dieser Wert ist auf dem Ausgabe-Modul (Adresse 0) der CPU 214DPM auszugeben.



### Projektierdaten

#### CPU 21xDPM

Zählerstand: MB 0 (FFh ... 00h)  
 PROFIBUS-Adresse: 4  
 Eingangsbereich: Adresse 10                      Länge: 2 Byte  
 Ausgangsbereich: Adresse 20                      Länge: 2 Byte

#### CPU 21xDP

Zählerstand: MB 0 (00h...FFh)  
 PROFIBUS-Adresse: 3  
 Eingangsbereich: Adresse 30                      Länge: 2 Byte  
 Ausgangsbereich: Adresse 40                      Länge: 2 Byte  
 Parameterdaten: Adresse 50                      Länge: 24 Byte (fest)  
 Diagnosedaten: Adresse 60                      Länge: 6 Byte (fest)  
 Statusdaten: Adresse 100                      Länge: 2 Byte (fest)

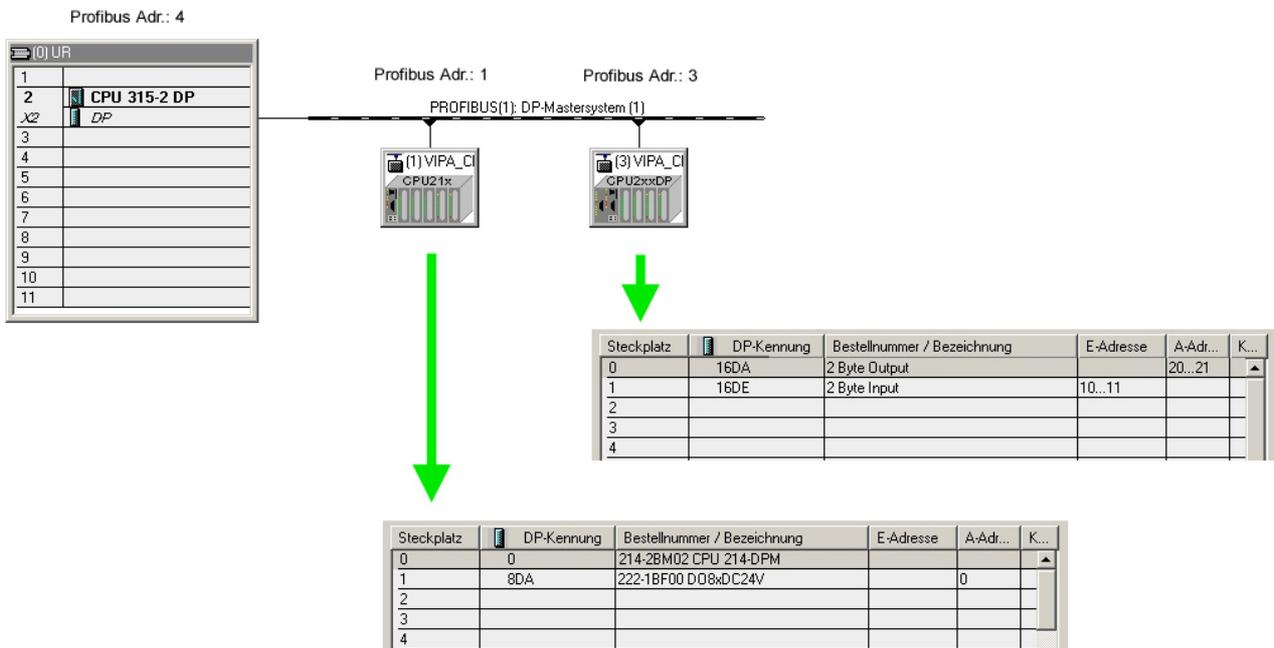
**Projektierung  
CPU 21xDPM  
(Master)**

Um kompatibel mit dem Siemens SIMATIC Manager zu sein, sind für das System 200V folgende Schritte durchzuführen:

- Starten Sie den Hardware-Konfigurator von Siemens.
- Installieren Sie die GSD-Datei VIPA\_21x.gsd.
- Projektieren Sie eine CPU 315-2 DP mit DP-Master-System (Adresse 4).
- Fügen Sie einen PROFIBUS-Slave "**VIPA\_CPU21x**" mit Adresse 1 an .
- Binden Sie auf dem 1. Steckplatz des Slave-Systems die CPU **214-2BM02** ein.
- Binden Sie danach das Ausgabe-Modul 222-1BF00 ein.

Zur Ankopplung Ihrer CPU 21xDP sind, nachdem Sie die GSD-Datei VIPA04d5.gsd eingebunden haben, folgende Schritte erforderlich:

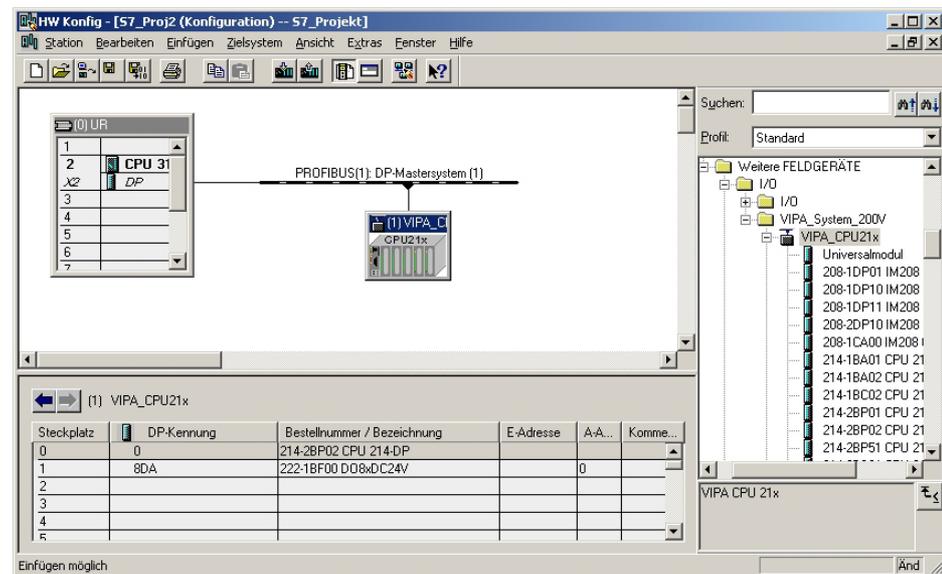
- Fügen Sie den PROFIBUS-Slave "**VIPA\_CPU2xxDP**" an (Adresse 3). Sie finden den DP-Slave im Hardware-Katalog unter: *PROFIBUS-DP > Weitere Feldgeräte > I/O > VIPA\_System\_200V > VIPA\_CPU2xxDP*.
- Teilen Sie für Ein- und Ausgabe dem PROFIBUS-DP-Master-Teil in Form von Byte-Blöcken Speicherbereiche aus dem Adressbereich der CPU zu. Binden Sie hierzu auf dem 1. Steckplatz das "2 Byte Output" Element ein und stellen Sie die Ausgabe-Adresse 20 ein. Binden Sie auf dem nachfolgenden Steckplatz das "2 Byte Input" Element ein und stellen Sie die Eingabe-Adresse 10 ein.
- Speichern Sie Ihr Projekt!



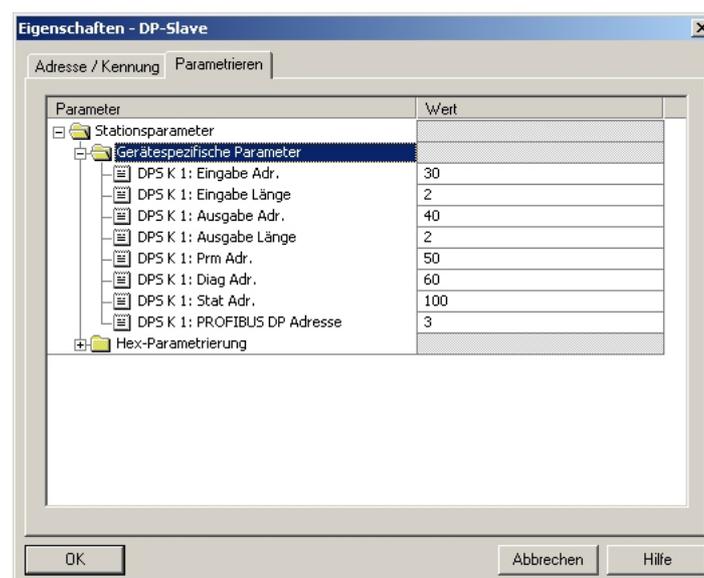
## Projektierung CPU 21xDP (Slave)

Um kompatibel mit dem Siemens SIMATIC Manager zu sein, sind für das System 200V folgende Schritte durchzuführen:

- Starten Sie den Hardware-Konfigurator von Siemens.
- Projektieren Sie eine CPU 315-2DP mit DP-Master-System (Adresse 2).
- Fügen Sie einen PROFIBUS-Slave "VIPA\_CPU21x" mit Adresse 1 an.
- Binden Sie auf dem 1. Steckplatz des Slave-Systems die CPU **214-2BP02** ein.
- Binden Sie auf dem nächsten Steckplatz 1 das Ausgabe-Modul 222-1BF00 ein.



- Stellen Sie folgende Parameter im Parametrierfenster der CPU 214-2BP02 ein:



- Speichern Sie Ihr Projekt!

**Anwender-  
programm in  
CPU 214DPM**

Das Anwenderprogramm in der CPU 214DPM hat zwei Aufgaben, die auf zwei OBs verteilt werden:

- Über Kontrollbyte die Kommunikation testen.  
Vom PROFIBUS das Eingangs-Byte laden und den Wert auf dem Ausgabe-Modul ausgeben.

**OB 1 (zyklischer Aufruf)**

```

L   B#16#FF
T   AB  20           Kontrollbyte für Slave-CPU

L   B#16#FE           Kontrollwert 0xFE laden
L   EB  10           Wurde Kontrollbyte von der Slave
<>I CPU richtig übermittelt?
BEB                Nein -> Ende

-----
                Datenaustausch via PROFIBUS

L   EB  11           Lade Eingangsbyte 11 (Ausgangsdaten
T   AB  0           der CPU214DP) und
                transferiere ins Ausgangsbyte 0

BE

```

- Zählerstand aus dem MB 0 lesen, dekrementieren, in MB 0 speichern und über PROFIBUS an CPU 214DP ausgeben.

**OB 35 (Zeit-OB)**

```

L   MB  0           Zähler von 0xFF bis 0x00
L   1
-I
T   MB  0

T   AB  21           Transferiere ins Ausgangsbyte 21
                (Eingangsdaten der CPU214DP)

BE

```

Auf Seiten der CPU 214DPM ist jetzt alles programmiert. Auch die PROFIBUS-Kommunikation ist nun auf beiden Seiten festgelegt.

Übertragen Sie mit den Zielsystemfunktionen via MPI Ihr Projekt in die CPU 214DPM.

**Anwender-  
programm in  
CPU 214DP**

Das Anwenderprogramm hat wie schon weiter oben gezeigt zwei Aufgaben, die auch bei dieser CPU auf zwei OBs verteilt werden:

- Vom PROFIBUS-Slave das Eingangs-Byte laden und den Wert auf dem Ausgabe-Modul ausgeben.

**OB 1 (zyklischer Aufruf)**

```

L   PEW 100           Statusdaten laden und in Merker-
T   MW 100           wort speichern

UN  M 100.5          Inbetriebnahme durch DP-Master
BEB                               erfolgt? Nein -> Ende

U   M 101.4          Empfangsdaten gültig?
BEB                               Nein -> Ende
L   B#16#FF          Kontrollwert laden und mit
L   PEB 30           Kontrollbyte (1. Eingangsbyte)
<>I                               vergleichen
BEB                               Empfangene Daten haben keine
                                   gültigen Werte

L   B#16#FE          Kontrollbyte für Master-CPU
T   PAB 40           -----
                                   Datenaustausch via PROFIBUS

L   PEB 31           Lade Peripheriebyte 31 (Eingangs-
T   AB 0             daten vom PROFIBUS-Slave) und
                                   transferiere ins Ausgangsbyte 0

BE

```

- Zählerstand aus dem MB 0 lesen, inkrementieren, in MB 0 speichern und über PROFIBUS an CPU 21x ausgeben.

**OB 35 (Zeit-OB)**

```

L   MB 0             Zähler von 0x00 bis 0xFF
L   1
+I
T   MB 0

T   PAB 41          Transferiere Zählerwert ins
                                   Peripheriebyte 41 (Ausgangsdaten
                                   des PROFIBUS-Slaves)

BE

```



## Teil 9 Einsatz CPU 21xCAN

### Überblick

Inhalt dieses Kapitels ist der Einsatz der CPU 21xCAN unter CANopen. Sie erhalten hier alle Informationen, die zum Einsatz des integrierten CAN-Masters erforderlich sind.

### Inhalt

Thema	Seite
<b>Teil 9 Einsatz CPU 21xCAN</b> .....	<b>9-1</b>
Grundlagen CAN-Bus.....	9-2
Projektierung der CPU 21xCAN .....	9-4
Betriebsarten.....	9-13
Prozessabbild der CPU 21xCAN .....	9-14
CANopen Telegrammaufbau.....	9-16
Objekt-Verzeichnis .....	9-21

## Grundlagen CAN-Bus

### Allgemeines

Der CAN-Bus (**C**ontrol **A**rea **N**etwork) ist ein international offener Feldbus-Standard für Gebäude-, Fertigungs- und Prozessautomatisierung und wurde ursprünglich für die Automobiltechnik entwickelt.

Aufgrund der umfassenden Fehlererkennungs-Maßnahmen gilt der CAN-Bus als das sicherste Bussystem mit einer Restfehlerwahrscheinlichkeit von weniger als  $4,7 \times 10^{-11}$ . Fehlerhafte Meldungen werden signalisiert und automatisch neu übertragen.

Im Gegensatz zu PROFIBUS und INTERBUS-S sind beim CAN-Bus auch verschiedene Schicht-7-Anwenderprofile unter dem CAL-Schicht-7-Protokoll definiert (CAL=CAN application layer). Ein solches Anwenderprofil ist CANopen, dessen Standardisierung der CiA (CAN in Automation) e.V. übernimmt.

### CANopen

CANopen ist das Anwenderprofil für den Bereich industrieller Echtzeitsysteme und wird zur Zeit von vielen Herstellern implementiert. CANopen wurde als Profil DS-301 von der CAN-Nutzerorganisation (C.i.A) veröffentlicht. Das Kommunikationsprofil DS-301 dient zur Standardisierung der Geräte. Somit werden die Produkte verschiedener Hersteller austauschbar. Weiter sind zur Gewährleistung der Austauschbarkeit in dem Geräteprofil DS-401 die gerätespezifischen Daten und die Prozessdaten standardisiert. DS-401 standardisiert die digitalen und analogen Ein-/Ausgabe-Module.

CANopen besteht aus dem Kommunikationsprofil (communication profile) das festlegt, welche Objekte für die Übertragung bestimmter Daten zu verwenden sind, und den Geräteprofilen (device profiles), die die Art der Daten spezifizieren, die mit den Objekten übertragen werden.

Das CANopen-Kommunikationsprofil basiert auf einem Objektverzeichnis ähnlich dem des PROFIBUS. Im Kommunikationsprofil DS-301 sind zwei Objektarten sowie einige Spezialobjekte definiert:

- Prozessdatenobjekte (PDO)  
PDOs dienen der Übertragung von Echtzeitdaten
- Servicedatenobjekte (SDO)  
SDOs ermöglichen den lesenden und schreibenden Zugriff auf das Objektverzeichnis

**Übertragungs-  
medium**

CAN basiert auf einer linienförmigen Topologie. Sie haben die Möglichkeit mittels Routerknoten eine Netzstruktur aufzubauen. Die Anzahl der Teilnehmer pro Netz wird nur durch die Leistungsfähigkeit des eingesetzten Bustreiberbausteins begrenzt.

Die maximale Netzausdehnung ist durch Signallaufzeiten begrenzt. Bei 1MBaud ist z.B. eine Netzausdehnung von 40m und bei 80kBAud von 1000m möglich.

CAN-Bus verwendet als Übertragungsmedium eine abgeschirmte Dreidrahtleitung (Fünfdraht optional).

Der CAN-Bus arbeitet mit Spannungsdifferenzen. Er ist daher unempfindlicher gegenüber Störeinflüssen als eine Spannungs- oder Stromschnittstelle. Das Netz sollte als Linie konfiguriert sein, mit einem 120Ω Abschlusswiderstand am Ende.

Auf dem VIPA CAN-Master befindet sich ein 9poliger Stecker. Über diesen Stecker koppeln Sie den CAN-Master als Slave direkt in das CAN-Bus-Netz ein.

Alle Teilnehmer im Netz kommunizieren mit der gleichen Baudrate.

Die Bus Struktur erlaubt das rückwirkungsfreie Ein- und Auskoppeln 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.

**Buszugriffs-  
verfahren**

Man unterscheidet bei Buszugriffsverfahren generell zwischen kontrolliertem (deterministischem) und unkontrolliertem (zufälligen) Buszugriff.

CAN arbeitet nach dem Verfahren Carrier-Sense Multiple Access (CSMA), d.h. jeder Teilnehmer ist bezüglich des Buszugriffs gleichberechtigt und kann auf den Bus zugreifen, sobald dieser frei ist (zufälliger Buszugriff).

Der Nachrichtenaustausch ist nachrichtenbezogen und nicht teilnehmerbezogen. Jede Nachricht ist mit einem priorisierenden Identifier eindeutig gekennzeichnet. Es kann immer nur ein Teilnehmer für seine Nachricht den Bus belegen.

Die Buszugriffssteuerung bei CAN geschieht mit Hilfe der zerstörungsfreien, bitweisen Arbitrierung. Hierbei bedeutet zerstörungsfrei, dass der Gewinner der Arbitrierung sein Telegramm nicht erneut senden muss. Beim gleichzeitigen Mehrfachzugriff von Teilnehmern auf den Bus wird automatisch der wichtigste Teilnehmer ausgewählt. Erkennt ein sendebereiter Teilnehmer, dass der Bus belegt ist, so wird sein Sendewunsch bis zum Ende der aktuellen Übertragung verzögert.

## Projektierung der CPU 21xCAN

### Übersicht

Die Projektierung des CANopen-Masters erfolgt unter WinCoCT (**Windows CANopen Configuration Tool**) von VIPA. Aus WinCoCT exportieren Sie Ihr Projekt als wld-Datei. Die wld-Datei können Sie in Ihren Hardware-Konfigurator von Siemens importieren.

Legen Sie hierzu ein virtuelles PROFIBUS-System "VIPA\_CPU21x" an und binden Sie auf Steckplatz 0 die CPU21xCAN (VIPA 21x-2CM02) ein.

### Schnelleinstieg

Für den Einsatz von CAN-Master und System 200V Modulen ist die Einbindung der Module über die GSD-Datei als virtuelles PROFIBUS-System im Hardware-Konfigurator von Siemens erforderlich. Hierzu sind folgende Schritte durchzuführen:

- WinCoCT starten und CANopen-Netzwerk projektieren.
- Hierzu mit  eine "Master"-Gruppe anlegen und mit  einen CANopen-Master einfügen.
- Über "Device Access" mit "Device is NMT Master" die Master-Funktion aktivieren.
- Aktivieren Sie im Register "CANopen Manager" Device is NMT Master und bestätigen Sie Ihre Eingabe.
- Mit "Set PLC Parameters" Parameter vorgeben, wie Diagnose-Verhalten und CPU-Adress-Bereiche.
- Eine "Slave"-Gruppe mit  anlegen und mit  Ihre CANopen-Slaves hinzufügen.
- Den Slaves über "Module" Module hinzufügen und ggf. parametrieren.
- Unter "Connections" Prozessdatenverbindungen in der Matrix einstellen, ggf. Eingabe im Prozessabbild des Master überprüfen.
- Projekt speichern und als wld-Datei exportieren.
- Im Hardware-Konfigurator von Siemens vipa\_21x.gsd einbinden.
- In den Siemens SIMATIC Manager wechseln und Datenbaustein von CAN-wld-Datei in Bausteine-Verzeichnis kopieren.
- Im Hardware-Konfigurator PROFIBUS-DP-Master-System mit folgender Siemens-CPU projektieren: CPU 315-2DP (6ES7 315-2AF03-0AB0 V1.2)
- DP-Master bekommt Adresse >1.
- An Master-System aus dem Hardware-Katalog das DP-Slave-System "VIPA\_CPU21x" anbinden.
- Slave-System bekommt Adresse 1.
- Beginnend mit CPU 21xCAN auf dem 1. Steckplatz die System 200V Module in gesteckter Reihenfolge platzieren.
- Alles speichern und SPS-Projekt via MPI in die CPU übertragen.

Nachfolgend sind diese Schritte näher erläutert.

### Voraussetzungen zur Projektierung

Der Hardware-Konfigurator ist Bestandteil des Siemens SIMATIC Managers. Er dient der Projektierung. Die Module, die hier projiziert werden können, entnehmen Sie dem Hardware-Katalog.

Für den Einsatz der System 200V Module ist die Einbindung der System 200V Module über die GSD-Datei vipa\_21x.gsd von VIPA im Hardwarekatalog erforderlich.



### Hinweis!

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

GSD-Datei einbinden

- Kopieren Sie die mitgelieferte VIPA-GSD-Datei vipa\_21x.gsd in Ihr GSD-Verzeichnis ... \siemens\step7\s7data\gsd.
- Starten Sie den Hardware-Konfigurator von Siemens.
- Schließen Sie alle Projekte.
- Gehen Sie auf **Extras** > *Neue GSD-Datei installieren*.
- Geben hier **VIPA\_21x.GSD** an.

Die Module des System 200V von VIPA sind jetzt im Hardwarekatalog integriert und können projiziert werden.

### Hinweis

**Um kompatibel mit dem Siemens SIMATIC Manager zu sein, sind die System 200V CPUs von VIPA als**

**CPU 315-2DP (6ES7 315-2AF03-0AB0 V1.2)**

**zu projektieren!**

**Damit die Module gezielt angesprochen werden können, sind diese im Hardware-Konfigurator von Siemens in Form eines virtuellen PROFIBUS-Systems zu projektieren. Hierbei können Sie durch Einbindung einer GSD-Datei von VIPA auf den Funktionsumfang der System 200V Module zurückgreifen.**

**Den CAN-Master projektieren Sie in Ihrem virtuellen PROFIBUS-System durch Platzieren einer CPU 21xCAN auf dem 1. Steckplatz.**

**Die eigentliche Projektierung führen Sie mit dem CANopen-Konfigurations-Tool WinCoCT durch. Ihr Projekt können Sie in Form einer wld-Datei exportieren und als DB in Ihr SPS-Programm übernehmen.**

## WinCoCT

WinCoCT (**Windows CANopen Configuration Tool**) ist ein von VIPA entwickeltes Konfigurations-Tool zur komfortablen Projektierung von CANopen-Netzwerken.

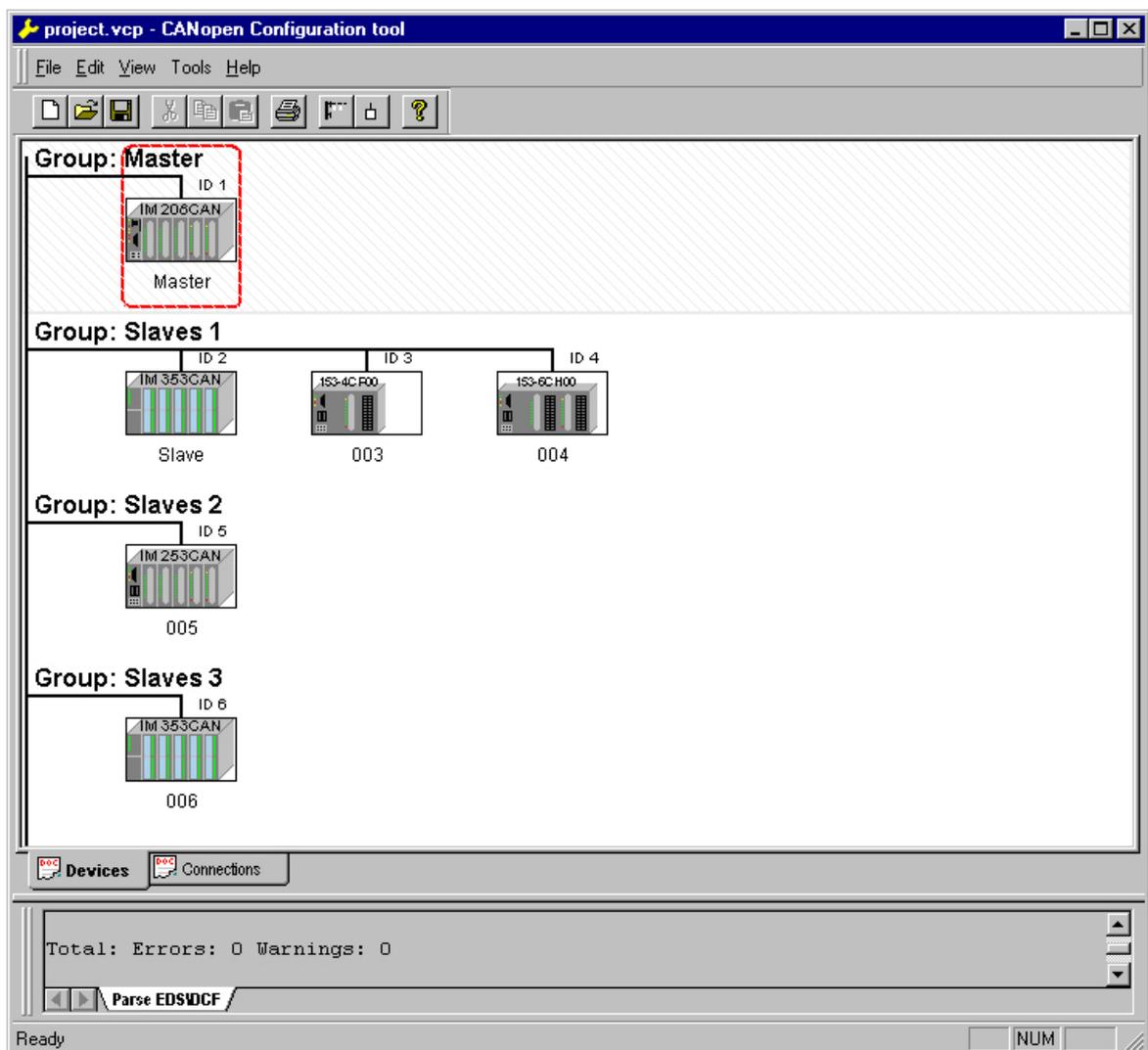
WinCoCT stellt auf einer grafischen Benutzeroberfläche die Topologie Ihres CANopen-Netzwerks dar. Hier können Sie Feldgeräte und Steuerungen platzieren, parametrieren, gruppieren und Verbindungen projektieren.

Die Auswahl der Geräte erfolgt über eine Liste, die jederzeit über eine EDS-Datei (**E**lectronic **D**ata **S**heet) beliebig erweitert werden kann.

Durch Klick mit der rechten Maustaste auf ein Gerät, erscheint für dieses Gerät ein Menü, das zu einem Teil aus statischen und zum anderen Teil aus dynamischen Komponenten besteht.

Zur Konfiguration des Prozessdatenaustauschs werden alle Prozessdaten in Form einer Matrix dargestellt, wobei Geräte-Eingänge als Zeile und -Ausgänge als Spalte ausgegeben werden. Durch einfaches Markieren der Kreuzungspunkte stellen Sie die gewünschte Verbindung her.

Das Zusammenstellen und Optimieren der Telegramme führt WinCoCT selbständig durch.



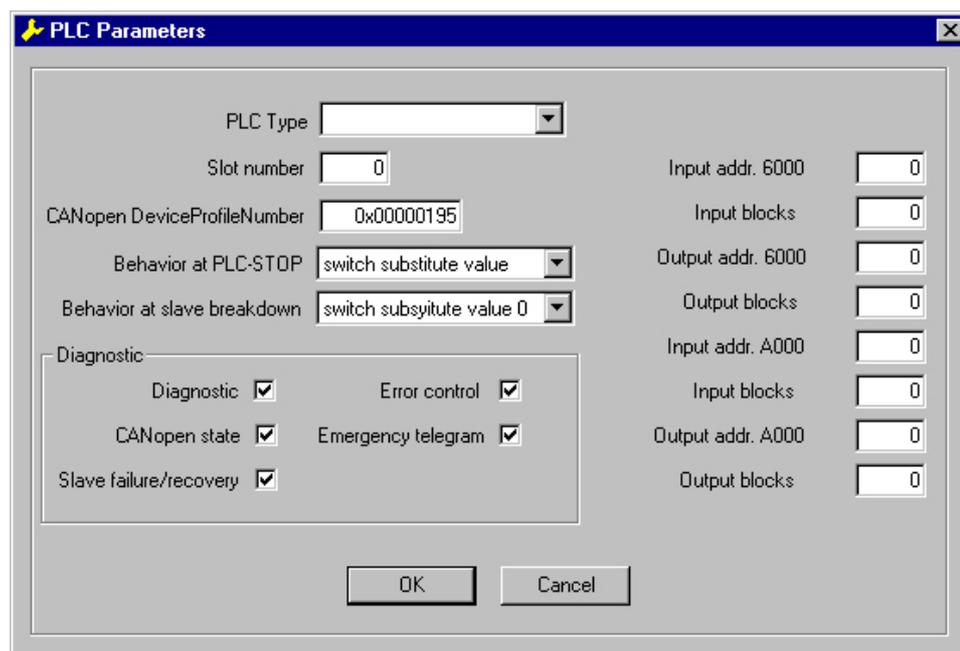
**Projektparameter einstellen**

Über **Tools** > *Project options* können Sie CAN-spezifischen Parameter wie Baud-Rate, Auswahl des Masters usw. vorgeben.

Näheres hierzu finden Sie in der Beschreibung von WinCoCT.

**Parameter CAN-Master**

Sie haben die Möglichkeit über WinCoCT VIPA-spezifische Parameter für den CAN-Master vorzugeben, indem Sie mit der rechten Maustaste auf den Master klicken und mit Set PLC-Parameters den nachfolgenden Dialog aufrufen:



PLC Type

Reserviert für zukünftige Erweiterungen

Slot number.

Steckplatz-Nr. auf dem Bus

0: Zur Adressierung des in die CPU integrierten CAN-Masters

1 ... 32: Zur Adressierung von stand-alone System 200V CAN-Master

CANopen DeviceProfileNumber

Fest eingestellt auf 0x195

Behavior at PLC-STOP

Hier können Sie das Verhalten der Ausgabe-Kanäle einstellen, sobald die CPU in STOP geht. Folgende Werte stehen zur Auswahl:

*Switch substitute value 0*: Schaltet alle Ausgänge auf 0

*Keep last value*: Friert den aktuellen Zustand der Ausgänge ein.

Behavior at Slave breakdown

Geben Sie hier an, wie die Handhabung der Slave-Eingangsdaten sein soll, wenn ein Slave ausfällt.

*Switch substitute value 0*: Die Daten werden auf 0 gesetzt.

*Keep the last value*: Die aktuellen Daten bleiben unverändert.

Diagnostic	<p>In diesem Bereich können Sie das Diagnose-Verhalten des CAN-Masters einstellen.</p> <p><i>Diagnostic:</i> Aktiviert die Diagnosefunktion</p> <p><i>CANopen state:</i> Im aktivierten Zustand sendet der CAN-Master seinen Status "preoperational" oder "operational" an die CPU. Den Status können Sie über SFC 13 abrufen.</p> <p><i>Slave failure/recovery:</i> Wenn Sie diese Option aktiviert haben wird bei Slave-Ausfall und -Wiederkehr der OB 86 in der CPU aufgerufen.</p> <p><i>Error control:</i> Ist diese Option angewählt, so sendet der NMT-Master alle Guarding-Fehler als Diagnose an die CPU, die den OB 82 aufruft.</p> <p><i>Emergency Telegram:</i> Bei aktivierter Option sendet der NMT-Master alle Emergency-Telegramme als Diagnose an die CPU, die den OB 82 aufruft.</p>
Adressbereich in der CPU	<p>Über die nachfolgend aufgeführten Felder können Sie die Adressbereiche vorgeben, die der CAN-Master für seine Ein- und Ausgabe-Bereiche in der CPU belegt. Jeder Block besteht aus 4Byte.</p> <p><i>Input addr. 6000, Input blocks</i> PE-Basis-Adresse in der CPU, die von 0x6000-CAN-Eingangsdaten belegt werden. Für Input blocks können max. 16 (64Byte) eingetragen werden.</p> <p><i>Output addr. 6000, Output blocks</i> PA-Basis-Adresse in der CPU, die von 0x6000-CAN-Ausgangsdaten belegt werden. Für Output blocks können max. 16 (64Byte) eingetragen werden.</p> <p><i>Input addr. A000, Input blocks</i> PE-Basis-Adresse in der CPU, die von 0xA000-CAN-Eingangs-Netzwerk-Variablen belegt werden. Für Input blocks können max. 80 (320Byte) eingetragen werden.</p> <p><i>Output addr. A000, Output blocks</i> PA-Basis-Adresse in der CPU, die von 0xA000-CAN-Ausgangs-Netzwerk-Variablen belegt werden. Für Output blocks können max. 80 (320Byte) eingetragen werden.</p>
<b>CANopenSlave in CANopen Manager aktivieren</b>	<p>Damit ein CANopen-Slave vom Master bearbeitet werden kann, ist dieser über WinCoCT bei dem entsprechenden Master anzumelden. Klicken Sie hierzu mit der rechten Maustaste auf Ihren CAN-Master, wählen Sie "Device access" an und gehen Sie in das Register "CANopen Manager".</p> <p>Hier können Sie über [Change] jeden Slave einzeln bzw. über [Global] alle Slaves bei Ihrem Master anmelden und das Fehlerverhalten einstellen.</p> <p><b>Bitte vergessen Sie nicht, nachdem Sie Ihre Einstellungen durchgeführt haben, diese mit [Apply to slaves] in Ihre Projektierung zu übernehmen.</b></p>

## Schritte der Projektierung

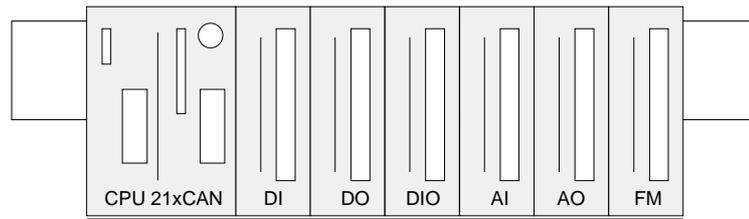
Nachfolgend wird die Vorgehensweise der Projektierung an einem abstrakten Beispiel gezeigt:

Die Projektierung gliedert sich in folgende drei Teile:

- CAN-Master-Projektierung in WinCoCT und Export als wld-Datei
- CAN-Master-Projektierung importieren
- Hardware-Konfiguration von CPU 21xCAN und System 200V Modulen

## Hardwareaufbau

System 200V

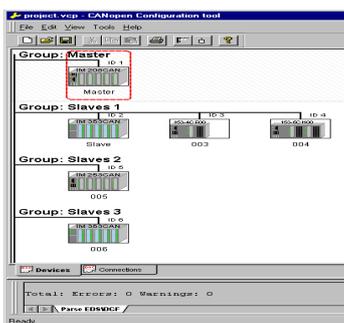


## Voraussetzungen

Zur Projektierung eines CANopen-Systems ist die aktuellste EDS-Datei in das EDS-Verzeichnis von WinCoCT zu übertragen.

Für den Einsatz der System 200V Module ist die Einbindung der System 200V Module über die GSD-Datei vipa\_cpu21x.gsd von VIPA im Hardwarekatalog erforderlich.

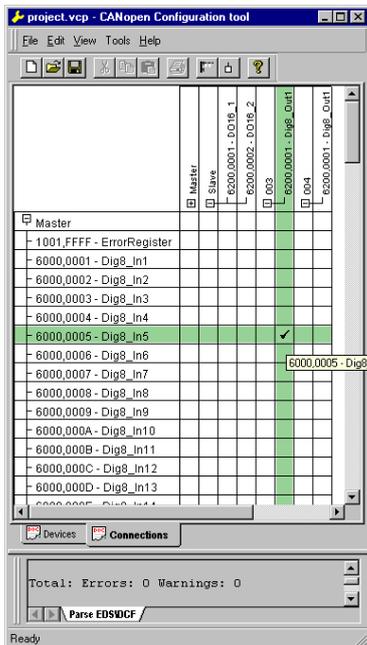
## CAN-Master-Projektierung unter WinCoCT



- Kopieren Sie die erforderlichen EDS-Dateien in das EDS-Verzeichnis und starten Sie WinCoCT.
- Legen Sie mit  eine "Master"-Gruppe an und fügen Sie mit  einen CANopen-Master ein (VIPA\_21x\_2CM02.eds).
- Legen Sie mit  eine "Slave"-Gruppe an und fügen Sie mit  Ihre CANopen-Slaves hinzu.
- Klicken mit der rechten Maustaste auf den entsprechenden Slave und fügen über "Module" Sie die entsprechenden Module hinzu.
- Parametrieren Sie Ihre Module mit [Parameter] bzw. über das entsprechende Objekt-Verzeichnis
- Klicken Sie mit der rechten Maustaste auf den Master und öffnen Sie den Dialog "Device Access".
- Aktivieren Sie im Register "CANopen Manager" Device is NMT Master und melden Sie die entsprechenden Slaves beim Master an. Vergessen Sie nicht Ihre Eingaben mit [Apply to slaves] in Ihre Projektierung zu übernehmen!



- Klicken Sie mit der rechten Maustaste auf den Master und öffnen Sie den VIPA-spezifischen Dialog "Set PLC Parameters". Hier können Sie das Diagnose-Verhalten einstellen und die Adress-Bereiche vorgeben, die vom Master in der CPU belegt werden. Unter "Slot number" geben mit 0 den Steckplatz 0 Ihrer CPU 21xCAN an. Hieraus generiert WinCoCT beim Export den DB 2000.



- Wechseln Sie im Hauptfenster in das Register "Connections". Hier werden die Prozessdaten als Eingänge (1. Spalte) und als Ausgänge (1. Zeile) in einer Matrix dargestellt. Zur Anzeige der Prozessdaten eines Geräts, dem ein "+" vorangestellt ist, klicken Sie auf das entsprechende Gerät.
  - Zu Ihrer Hilfe können Sie immer nur dann eine Verbindung definieren, wenn das Fadenkreuz grün erscheint. Stellen Sie mit der Maus in Zeile und Spalte der Matrix die entsprechende Zelle ein und klicken Sie mit der linken Maustaste → die Zelle wird mit einem "☒" gekennzeichnet. Sie können die projektierte Verbindung überprüfen, indem Sie wieder in "Devices" wechseln, auf den Master klicken und über "Device Access" das Prozessabbild des Masters ausgeben.
  - Speichern Sie Ihr Projekt.
  - Über **File** > **Export** wird Ihr CANopen-Projekt in eine wld-Datei exportiert. Der Name setzt sich zusammen aus Projektname + Knotenadresse (0)+ Kennung **Master/Slave**.
- Hiermit ist die CANopen-Projektierung unter WinCoCT abgeschlossen.

### Import in SPS-Programm

- Starten Sie den Siemens SIMATIC Manager mit Ihrem SPS-Projekt für die CPU 21xCAN.
- Öffnen Sie mit **Datei** > **Memory Card Datei** > **öffnen** die wld-Datei.
- Kopieren Sie den DB 2000 in Ihr Bausteine-Verzeichnis.

Sobald Sie diesen Baustein an Ihre CPU übertragen, wird dieser von der CPU erkannt und die entsprechenden Parameter werden an den CAN-Master übertragen.

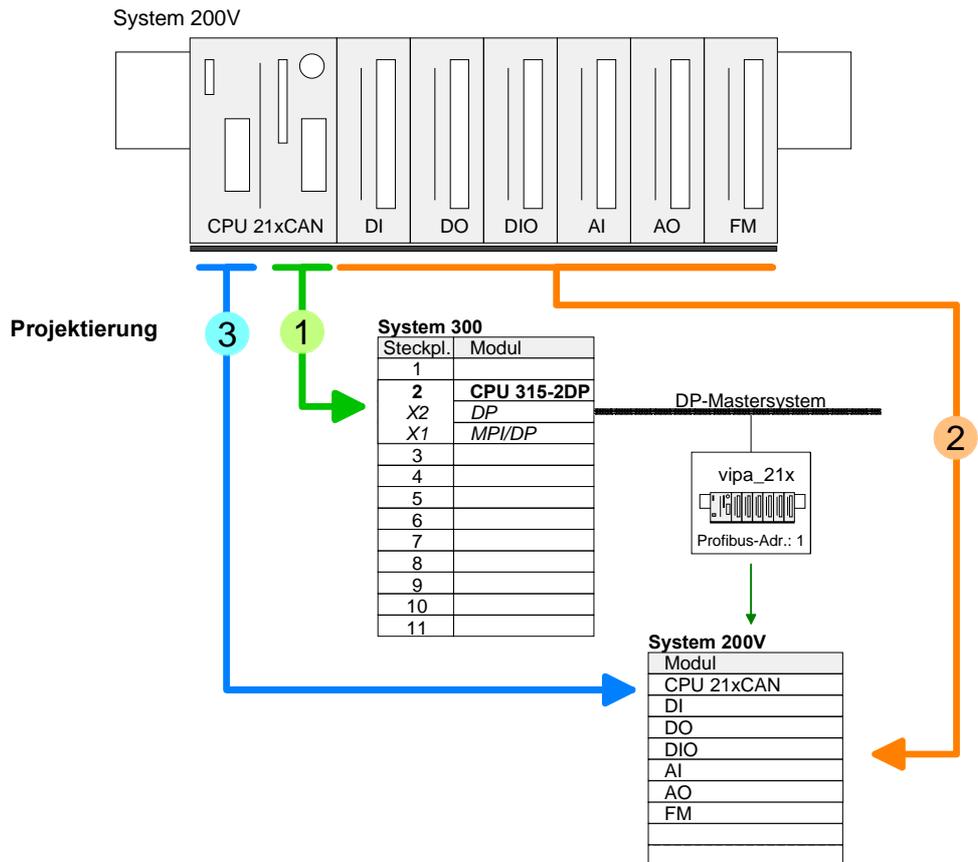
Dies ist aber nur möglich, wenn Ihre CAN-Master-CPU in der Hardware-Konfiguration als virtuelles PROFIBUS-System eingebunden wird. Die Vorgehensweise hierzu sehen Sie auf den Folgeseiten.

Hardware-Konfiguration CPU 21xCAN und System 200V Module

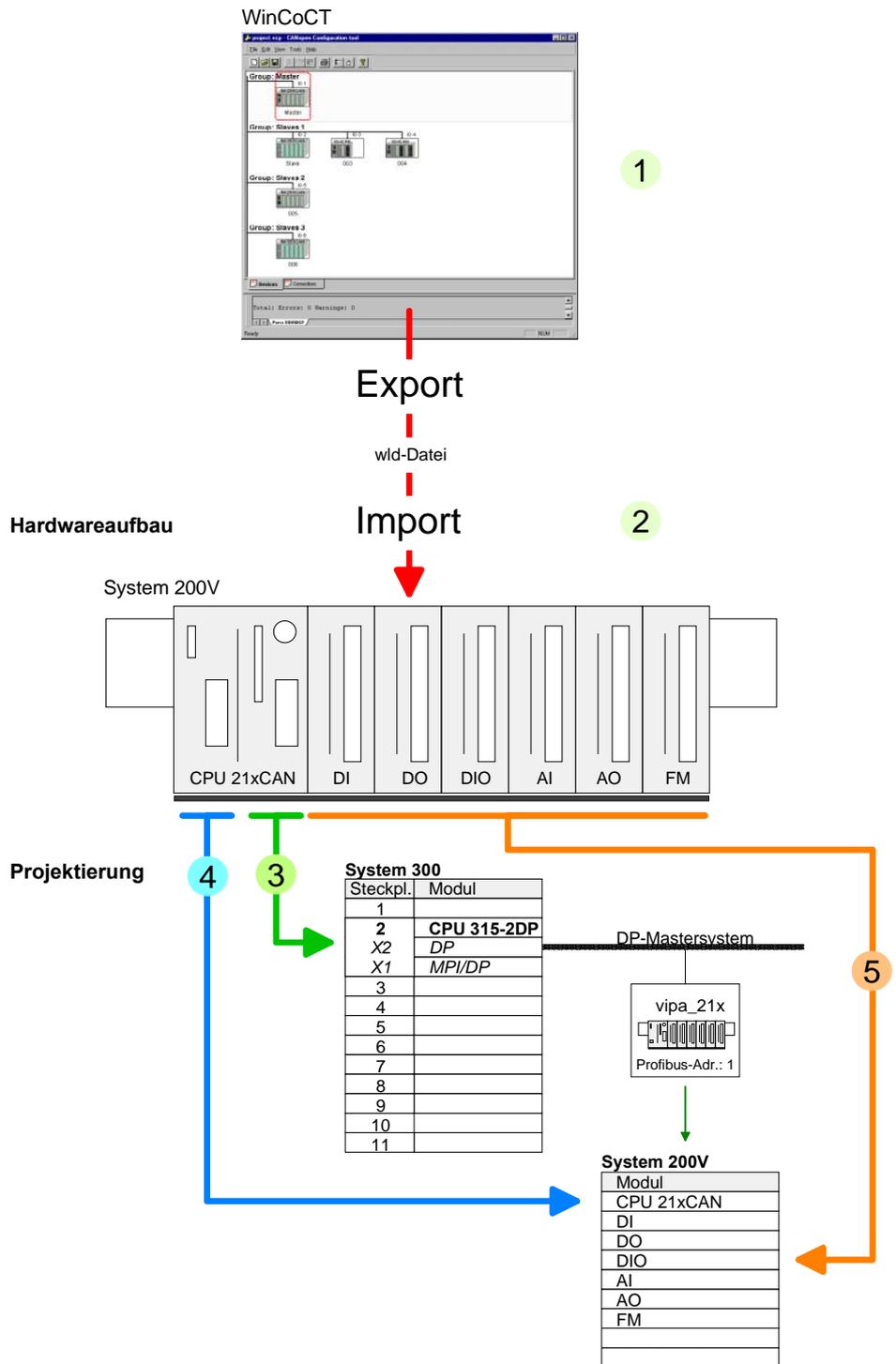
Die Hardware-Konfiguration des System 200V erfolgt nach folgender Vorgehensweisen:

- Starten Sie den Hardware-Konfigurator von Siemens mit einem neuen Projekt und fügen Sie aus dem Hardware-Katalog eine Profilschiene ein.
- Fügen Sie die CPU 315-2DP (6ES7 315-2AF03-0AB0 V1.2) ein. Hierbei ist ein neues PROFIBUS-Subnetz anzulegen.
- Hängen Sie an das Subnetz das System "VIPA\_CPU21x". Sie finden dies im Hardware-Katalog unter *PROFIBUS DP > Weitere Feldgeräte > IO > VIPA\_System\_200V*. Geben Sie diesem Slave die PROFIBUS-Adresse 1.
- Platzieren Sie in Ihrem Konfigurator auf dem 1. Steckplatz die CPU 21xCAN, die Sie einsetzen, indem Sie diese dem Hardware-Katalog entnehmen.
- Binden Sie danach Ihre System 200V Module in der gesteckten Reihenfolge ein.
- Parametrieren Sie ggf. CPU bzw. die Module. Das Parameterfenster wird geöffnet, sobald Sie auf das entsprechende Modul doppelklicken.
- Sichern Sie Ihr Projekt!

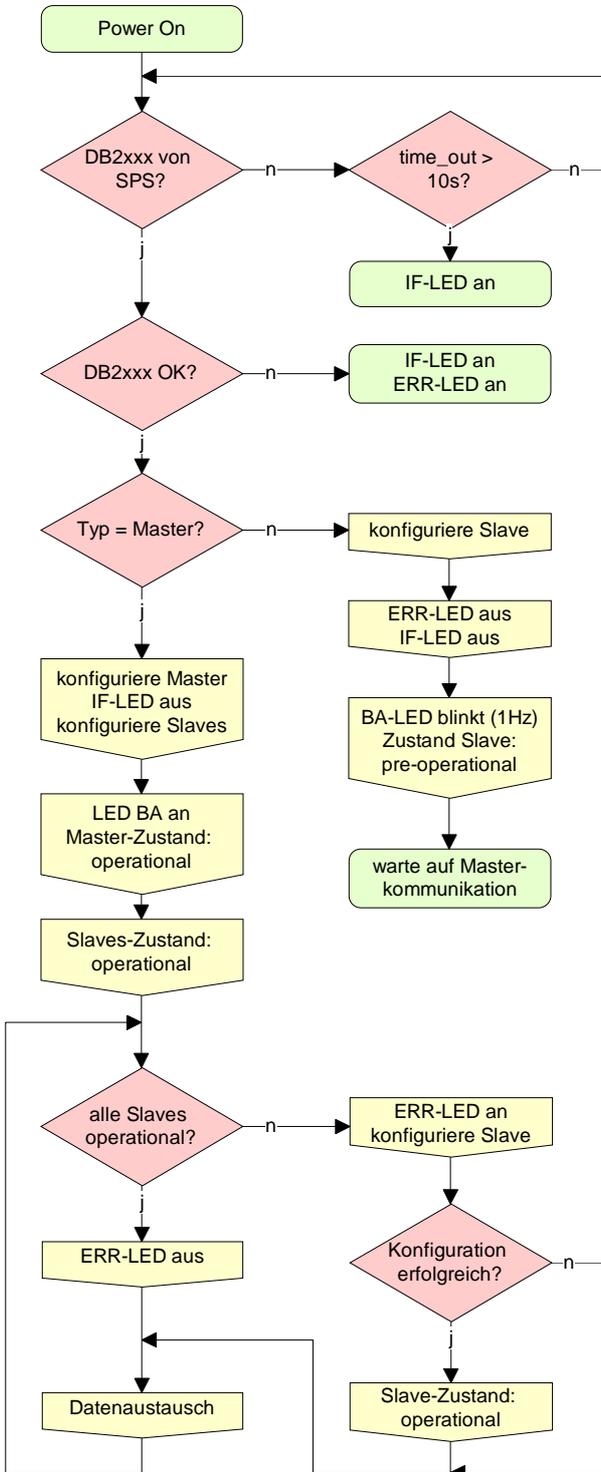
Hardwareaufbau



**Zusammenfassung** In der nachfolgenden Abbildung sind alle Projektierschritte nochmals zusammengefasst:



# Betriebsarten



## STOP → RUN (automatisch)

Nach NETZ EIN und bei gültigen Projektierdaten in der CPU geht der Master automatisch in RUN über. Auf einen Betriebsarten-Schalter für den Master wurde verzichtet.

Nach einem NETZ EIN werden automatisch die Projektierdaten von der CPU an den CAN-Master geschickt. Dieser baut eine Kommunikation zu den CAN-Slaves auf.

Bei erfolgter Kommunikation und gültigen Bus-Parametern, geht der CAN-Master in den Zustand "operational" über. Die LEDs RUN und BA leuchten.

Bei fehlerhaften Parametern bleibt der CAN-Master in STOP und zeigt über die IF-LED einen Parametrierfehler an.

## RUN

Im RUN leuchten die RUN- und BA-LEDs. Jetzt können Daten ausgetauscht werden.

Im Fehlerfall wie z.B. Slave-Ausfall, wird dies am CAN-Master über die ERR-LED angezeigt und ein Alarm an die CPU abgesetzt.

## Prozessabbild der CPU 21xCAN

Das Prozessabbild setzt sich aus folgenden Teilen zusammen:

- Prozessabbild für Eingangs-Daten (PE) für RPDOs
- Prozessabbild für Ausgangsdaten (PA) für TPDOs

Hiervon besteht jeder Teil aus einem 64Byte großen "Digital-Data"- und 320Byte großen "Network Variables"-Bereich.

### Eingabe-Daten

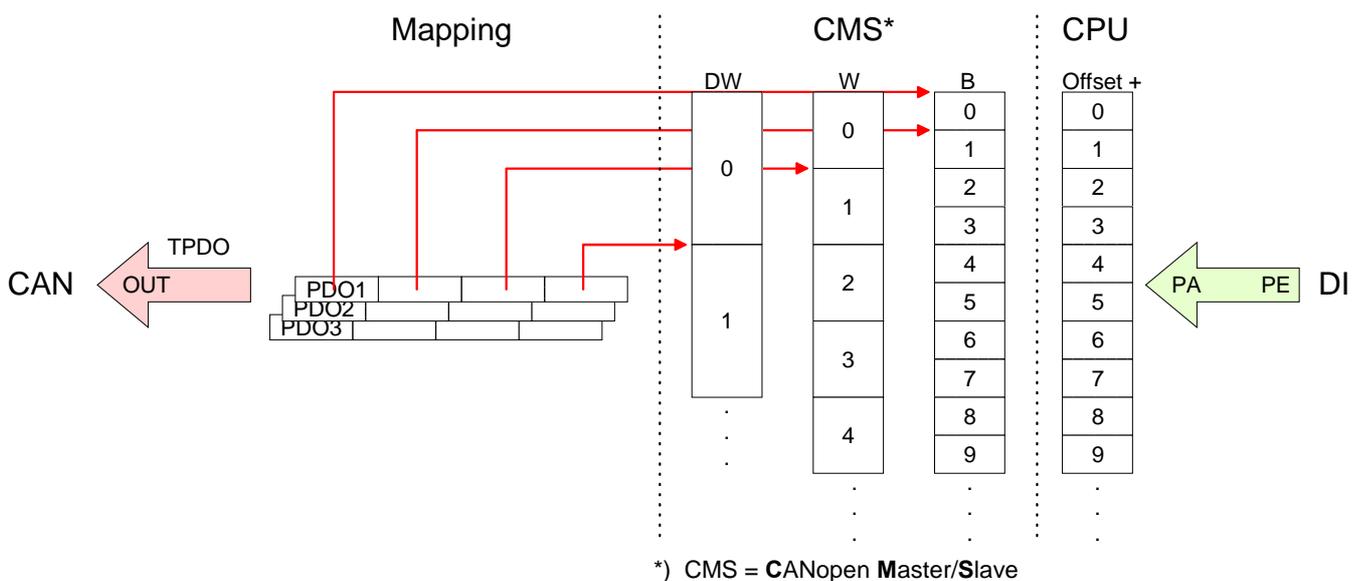
Für Eingabe-Daten gibt es folgende Objekte:

- 8 Bit Digitale Eingabe (Objekt 0x6000)
- 16 Bit Digitale Eingabe (Objekt 0x6100)
- 32 Bit Digitale Eingabe (Objekt 0x6120)
- 8 Bit Eingangs-Netzwerk-Variablen (Objekt 0xA040)
- 16 Bit Eingangs-Netzwerk-Variablen (Objekt 0xA100)
- 32 Bit Eingangs-Netzwerk-Variablen (Objekt 0xA200)
- 64 Bit Eingangs-Netzwerk-Variablen (Objekt 0xA440)

Wie in der nachfolgenden Abbildung zu erkennen ist, wird für die Objekte der digitalen Eingangsdaten der gleiche Speicherbereich in der CPU verwendet.

Beispielsweise würde ein Zugriff auf Index 0x6000 mit Subindex 2 einem Zugriff auf Index 0x6100 mit Subindex 1 entsprechen. Beide Objekte belegen die gleiche Speicherzelle in der CPU.

Bitte beachten Sie, dass auch die Eingangs-Netzwerk-Variablen den gleichen Speicherbereich benutzen.



**Ausgabe-Daten**

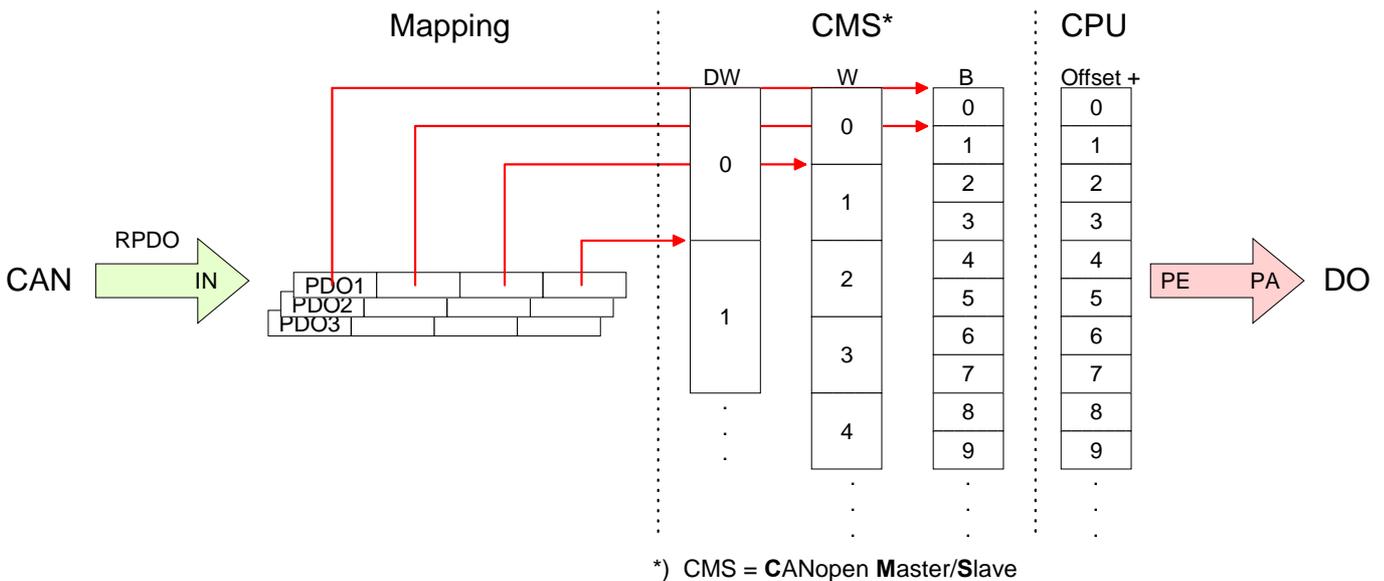
Für die digitalen Ausgabe-Daten wird die Zuordnung ähnlich durchgeführt. Für Ausgabe-Daten gibt es folgende Objekte:

- 8 Bit Digitale Ausgabe (Objekt 0x6200)
- 16 Bit Digitale Ausgabe(Objekt 0x6300)
- 32 Bit Digitale Ausgabe(Objekt 0x6320)
- 8 Bit Ausgangs-Netzwerk-Variablen (Objekt 0xA400)
- 16 Bit Ausgangs-Netzwerk-Variablen (Objekt 0xA580)
- 32 Bit Ausgangs-Netzwerk-Variablen (Objekt 0xA680)
- 64 Bit Ausgangs-Netzwerk-Variablen (Objekt 0xA8C0)

Wie in der nachfolgenden Abbildung zu erkennen ist, wird für die Objekte der digitalen Ausgangsdaten der gleiche Speicherbereich in der CPU verwendet.

Beispielsweise würde ein Zugriff auf Index 0x6200 mit Subindex 2 einem Zugriff auf Index 0x6300 mit Subindex 1 entsprechen. Beide Objekte belegen die gleiche Speicherzelle in der CPU.

Bitte beachten Sie, dass auch die Ausgangs-Netzwerk-Variablen den gleichen Speicherbereich benutzen.



## CANopen Telegrammaufbau

### Identifizier

Alle CANopen Telegramme besitzen nach CiA DS-301 folgenden Aufbau:  
*Identifizier*

Byte	Bit 7 ... Bit 0
1	Bit 3 ... Bit 0: Höchstwertige 4 Bits der Modul-ID Bit 7 ... Bit 4: CANopen Funktionscode
2	Bit 3 ... Bit 0: Datenlänge (DLC) Bit 4: RTR-Bit: 0: keine Daten (Anforderungstelegramm) 1: Daten vorhanden Bit 7 ... Bit 5: Niederwertige 3 Bits der Modul-ID

### Data

*Data*

Byte	Bit 7 ... Bit 0
3 ... 10	Daten

Der Unterschied zu einem Schicht-2-Telegramm besteht in einer zusätzlichen Unterteilung des 2 Byte Identifiers in einen Funktionsteil und eine Modul-ID. Im Funktionsteil wird die Art des Telegramms (Objekt) festgelegt und mit der Modul-ID wird der Empfänger adressiert.

Der Datenaustausch bei CANopen-Geräten erfolgt in Form von Objekten. Im CANopen-Kommunikationsprofil sind zwei Objektarten sowie einige Spezialobjekte definiert.

Der VIPA CAN-Master unterstützt folgende Objekte:

- 40 Transmit PDOs (PDO Linking, PDO Mapping)
- 40 Receive PDOs (PDO Linking, PDO Mapping)
- 2 Standard SDOs (1 Server, 127 Clients)
- 1 Emergency Objekt
- 1 Netzwerkmanagement Objekt NMT
- Node Guarding
- Heartbeat

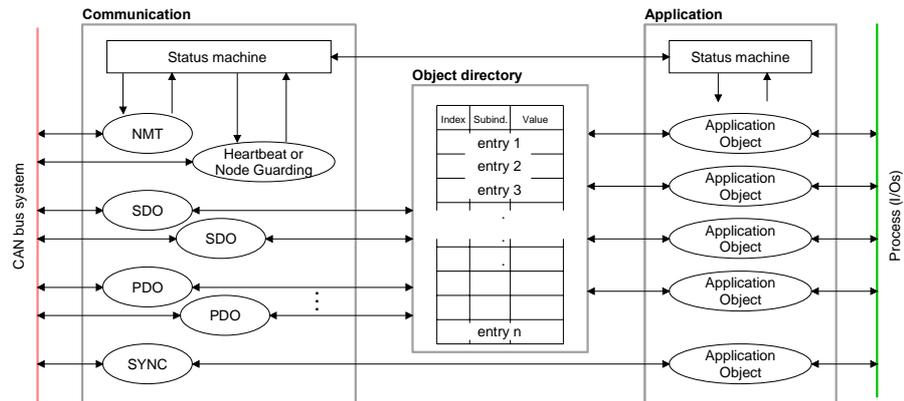


### Hinweis!

Der genaue Aufbau und Dateninhalt aller Objekte ist in den CiA-Profilen DS-301, DS-302, DS-401 und DS-405 beschrieben.

## Struktur des Gerätemodells

Ein CANopen Gerät kann wie folgt strukturiert werden:



### Communication

Stellt die Kommunikationsdatenobjekte und die zugehörige Funktionalität zum Datenaustausch über das CANopen Netzwerk zur Verfügung.

### Application

Die Applikationsdatenobjekte enthalten z.B. Ein- und Ausgangsdaten. Eine Applikationsstatusmaschine überführt die Ausgänge im Fehlerfall in einen sicheren Zustand.

Das Objektverzeichnis ist wie eine zweidimensionale Tabelle organisiert. Die Daten werden über Index und Subindex adressiert.

### Object directory

Dieses enthält alle Datenobjekte (Applikationsdaten + Parameter), die von außen zugänglich sind und die das Verhalten von Kommunikation, Applikation und Statusmaschinen beeinflussen.

## PDO

Bei vielen Feldbussystemen wird ständig das gesamte Prozessabbild übertragen - meist mehr oder weniger zyklisch. CANopen ist nicht auf dieses Kommunikationsprinzip beschränkt, da CAN durch die Multi-Master Buszugriffsregelung andere Möglichkeiten bietet.

Bei CANopen werden die Prozessdaten in Segmente zu maximal 8Byte aufgeteilt. Diese Segmente heißen **Prozessdaten-Objekte (PDOs)**. Die PDOs entsprechen jeweils einem CAN-Telegramm und werden über dessen spezifischen CAN-Identifizier zugeordnet und in ihrer Priorität bestimmt.

Für den Prozessdatenaustausch stehen beim CAN-Master insgesamt 80 PDOs zur Verfügung. Jedes PDO besteht dabei aus maximal 8 Datenbytes. PDOs werden unbestätigt übertragen, da das CAN-Protokoll die Übertragung sicherstellt.

Für Eingangsdaten stehen 40Tx Transmit-PDOs und für Ausgangsdaten 40Rx Receive-PDOs zur Verfügung. Die PDOs werden aus Sicht des CAN-Masters bezeichnet:

Receive-PDOs (RxPDOs) werden vom CAN-Master empfangen und enthalten Eingangsdaten.

Transmit-PDOs (TxPDOs) werden vom CAN-Master gesendet und enthalten Ausgangsdaten.

Die Belegung dieser PDOs mit Ein- bzw. Ausgangsdaten erfolgt unter WinCoCT automatisch.



<b>Index</b>	CANopen Index
<b>Subindex</b>	CANopen Subindex
<b>CanOpenError</b>	Liegt kein Fehler vor, so liefert CanOpenError eine 0 zurück. Im Fehlerfall beinhaltet CanOpenError eine der nachfolgend aufgeführten Fehlermeldungen, die vom CAN-Master generiert wird:

Code	Bedeutung
0x05030000	Toggle-Bit nicht geändert
0x05040000	SDO Protokoll Time-out
0x05040001	Client/server Befehlsspezifizierung nicht gültig oder unbekannt
0x05040002	Ungültige Blockgröße (nur Block-Modus)
0x05040003	Ungültige Sequenznummer (nur Block-Modus)
0x05040004	CRC Fehler (nur Block-Modus)
0x05040005	Unzureichender Speicher
0x06010000	Nicht unterstützter Zugriff auf ein Objekt
0x06010001	Lesezugriff auf ein Nur-Schreiben-Objekt
0x06010002	Schreibzugriff auf ein Nur-Lesen-Objekt
0x06020000	Objekt nicht im Objektverzeichnis vorhanden
0x06040041	Objekt kann nicht ins PDO gemappt werden
0x06040042	Anzahl und Länge der zu mappenden Objekte überschreitet PDO-Länge
0x06040043	Generelle Parameterinkompatibilität
0x06040047	Generelle interne Inkompatibilität im Gerät
0x06060000	Zugriffsfehler wegen Hardwareausfall
0x06070010	Datentyp nicht korrekt, Länge der Serviceparameter nicht korrekt
0x06070012	Datentyp nicht korrekt, Serviceparameter zu lang
0x06070013	Datentyp nicht korrekt, Serviceparameter zu kurz
0x06090011	Subindex existiert nicht
0x06090030	Wertebereich der Parameter überschritten (nur für Schreibzugriff)
0x06090031	Zu schreibender Parameterwert ist zu hoch
0x06090032	Zu schreibender Parameterwert ist zu niedrig
0x06090036	Maximumwert ist kleiner als Minimumwert
0x08000000	Genereller Fehler
0x08000020	Die Daten können entweder nicht transferiert oder nicht in der SPS gespeichert werden.
0x08000021	Die Daten können wegen lokaler Kontrollen entweder nicht transferiert oder nicht in der SPS gespeichert werden.
0x08000022	Die Daten können wegen aktuellem Modulstatus entweder nicht transferiert oder nicht in der SPS gespeichert werden.
0x08000023	Dynamische Objektverzeichnisgenerierung fehlgeschlagen oder kein Objektverzeichnis gefunden (z.B. Objektverzeichnis wird aus Datei generiert und ein Dateifehler ist aufgetreten).

**RetVal** Wird die Funktion fehlerfrei ausgeführt, enthält der Rückgabewert die gültige Länge der Antwortdaten: 1: Byte, 2: Wort, 4: Doppelwort  
Tritt während der Bearbeitung der Funktion ein Fehler auf, enthält der Rückgabewert einen der nachfolgend aufgeführten Fehlercodes.

Code	Bedeutung
0xF021	Ungültige Slave-Adresse (Aufrufparameter gleich 0 oder größer 127)
0xF022	Ungültiger Transfertyp (Wert ungleich 60h, 61h)
0xF023	Ungültige Datenlänge (der Datenpuffer ist zu klein, beim SDO-Lesezugriff sollte dieser mindestens 4 Byte groß sein, beim SDO-Schreibzugriff sollte dieser 1Byte, 2Byte oder 4 Byte groß sein).
0xF024	Der SFC wird nicht unterstützt.
0xF025	Schreibpuffer im CANopen-Master ist voll, Service kann zur Zeit nicht bearbeitet werden.
0xF026	Lesebuffer im CANopen-Master ist voll, Service kann zur Zeit nicht bearbeitet werden.
0xF027	Der SDO-Lese- oder Schreibzugriff wurde fehlerhaft beantwortet, siehe CANopen Error Codes.
0xF028	SDO-Timeout (es wurde kein CANopen-Teilnehmer mit der Node-Id gefunden)

**Busy** Solange Busy = 1 ist der aktuelle Auftrag ist noch nicht beendet.

**DataBuffer** Datenbereich, über den der SFC kommuniziert. Geben Sie hier einen ANY-Pointer vom Typ Byte an.  
SDO-Lesezugriff: Zielbereich für die gelesenen Nutzdaten.  
SDO-Schreibzugriff: Quellbereich für die zu schreibenden Nutzdaten.



**Hinweis!**

Sofern eine SDO-Anforderung fehlerfrei abgearbeitet wurde, enthält RetVal die Länge der gültigen Antwortdaten in 1, 2 oder 4 Byte und CanOpenError den Wert 0.

## Objekt-Verzeichnis

### Struktur

Im CANopen-Objektverzeichnis werden alle für das Gerät relevanten CANopen Objekte eingetragen. Jeder Eintrag im Objektverzeichnis ist durch einen 16Bit-Index gekennzeichnet.

Falls ein Objekt aus mehreren Komponenten besteht (z.B. Objekttyp Array oder Record), sind die Komponenten über einen 8Bit-Subindex gekennzeichnet.

Der Objektname beschreibt die Funktion eines Objekts. Das Datentyp-Attribut spezifiziert den Datentyp des Eintrags.

Über das Zugriffsattribut ist spezifiziert, ob ein Eintrag nur gelesen werden kann, nur geschrieben werden oder gelesen und geschrieben werden darf.

Das Objektverzeichnis ist in folgende 3 Bereiche aufgeteilt:

### Kommunikationsspezifischer Profilbereich (0x1000 – 0x1FFF)

Dieser Bereich beinhaltet die Beschreibung aller spezifischen Parameter für die Kommunikation.

0x1000 – 0x1011	allgemeine kommunikationsspezifische Parameter (z.B. der Gerätename)
0x1400 – 0x1427	Kommunikationsparameter (z.B. Identifier) der Receive-PDOs
0x1600 – 0x1627	Mappingparameter der Receive-PDOs Die Mappingparameter enthalten die Querverweise auf die Applikationsobjekte, die in die PDOs gemappt sind und die Datenbreite des entsprechenden Objektes
0x1800 – 0x1827 0x1A00 – 0x1A27	Kommunikations- und Mappingparameter der Transmit-PDOs

### Herstellerspezifischer Profilbereich (0x2000 – 0x5FFF)

Hier finden Sie die herstellerspezifischen Einträge. Der CAN-Master von VIPA besitzt keine herstellerspezifischen Einträge.

### Standardisierter Geräteprofilbereich (0x6000 – 0x9FFF)

In diesem Bereich liegen die Objekte für das Geräteprofil nach DS-401.



### Hinweis!

Da die CiA Normen ausschließlich in englischer Sprache vorliegen, wurden die Tabelleneinträge der Objekte zum eindeutigen Verständnis in englischer Sprache übernommen.

Eine nähere Beschreibung der Tabelleneinträge in Deutsch finden Sie jeweils unterhalb der Tabellen.

**Objektverzeichnis  
Übersicht**

Index	Content of Object
1000h	Device type
1001h	Error register
1005h	COB-ID SYNC
1006h	Communication Cycle Period
1007h	Synchronous Window Length
1008h	Manufacturer Hardware Version
1009h	Hardware Version
100Ah	Software Version
100Ch	Guard Time
100Dh	Life Time Factor
1016h	Consumer Heartbeat Time
1017h	Producer Heartbeat Time
1018h	Identity Object
1400h bis 1427h	Receive PDO Communication Parameter
1600h bis 1627h	Receive PDO Mapping Parameter
1800h bis 1827h	Transmit PDO Communication Parameter
1A00h bis 1A27h	Transmit PDO Mapping Parameter
1F22h	Concise DCF
1F25h	Post Configuration
1F80h	NMT StartUp
1F81h	Slave Assignment
1F82h	Request NMT
1F83h	Request Guarding
6000h	Digital-Input-8-Bit Array (see DS 401)
6100h	Digital-Input-16-Bit Array (see DS 401)
6120h	Digital-Input-32Bit Array (see DS 401)
6200h	Digital-Output-8-Bit Array (see DS 401)
6300h	Digital-Output-16-Bit Array (see DS 401)
6320h	Digital-Output-32-Bit Array (see DS 401)
A040h	Dynamic Unsigned8 Input
A100h	Dynamic Unsigned16 Input
A200h	Dynamic Unsigned32 Input
A4400h	Dynamic Unsigned64 Input
A4C0h	Dynamic Unsigned8 Output
A580h	Dynamic Unsigned16 Output
A680h	Dynamic Unsigned32 Output
A8C0h	Dynamic Unsigned64 Output

**Device Type**

Index	Sub-index	Name	Type	Attr.	Map.	Default value	Meaning
0x1000	0	Device Type	Unsigned32	ro	N	0x00050191	Statement of device type

Der 32Bit-Wert ist in zwei 16Bit-Felder unterteilt:

MSB	LSB
<b>Additional information Device</b>	<b>profile number</b>
0000 0000 0000 wxyz (bit)	405dec=0x0195

Die "Additional Information" enthält Angaben über die Signalarten des I/O-Gerätes:

z=1 bedeutet digitale Eingänge

y=1 digitale Ausgänge

x=1 analoge Eingänge

w=1 analoge Ausgänge

**Error register**

Index	Sub-Index	Name	Type	Attr.	Map.	Default value	Meaning
0x1001	0	Error Register	Unsigned8	ro	Y	0x00	Error register

Bit 7							Bit 0
ManSpec	reserved	reserved	Comm.	reserved	reserved	reserved	Generic

ManSpec.: Herstellerspezifischer Fehler, wird in Objekt 0x1003 genauer spezifiziert.

Comm.: Kommunikationsfehler (Overrun CAN)

Generic: Ein nicht näher spezifizierter Fehler ist aufgetreten (Flag ist bei jeder Fehlermeldung gesetzt)

**SYNC identifier**

Index	Sub-Index	Name	Type	Attr.	Map.	Default value	Meaning
0x1005	0	COB-Id sync message	Unsigned32	ro	N	0x80000080	Identifier of the SYNC message

Die unteren 11Bit des 32Bit Wertes enthalten den Identifier (0x80=128dez), das MSBit gibt Auskunft, ob das Gerät das SYNC-Telegramm empfängt (1) oder nicht (0).

Achtung: Im Gegensatz zu den PDO-Identifiern signalisiert das gesetzte MSB, dass dieser Identifier für den Knoten relevant ist.

**SYNC interval**

Index	Sub-index	Name	Type	Attr.	Map.	Default value	Meaning
0x1006	0	Communication cycle period	Unsigned32	rw	N	0x00000000	Maximum length of the SYNC interval in $\mu$ s.

Wenn hier ein Wert ungleich Null eingetragen wird, so geht der Koppler in den Fehlerzustand, wenn beim synchronen PDO-Betrieb innerhalb der "Watchdog-Zeit" kein SYNC-Telegramm empfangen wurde.

**Synchronous Window Length**

Index	Sub-index	Name	Type	Attr.	Map.	Default value	Meaning
0x1007	0	Synchronous window length	Unsigned32	rw	N	0x00000000	Contains the length of time window for synchronous PDOs in $\mu$ s.

**Device name**

Index	Sub-index	Name	Type	Attr.	Map.	Default value	Meaning
0x1008	0	Manufacturer device name	Visible string	ro	N		Device name of the bus coupler

VIPA 21x-2CM02

Da der zurückgelieferte Wert größer als 4Byte ist, wird das segmentierte SDO-Protokoll zur Übertragung verwendet.

**Hardware version**

Index	Sub-index	Name	Type	Attr.	Map.	Default value	Meaning
0x1009	0	Manufacturer Hardware version	Visible string	ro	N		Hardware version number of bus coupler

1.00

Da der zurückgelieferte Wert größer als 4Byte ist, wird das segmentierte SDO-Protokoll zur Übertragung verwendet.

**Software version**

Index	Sub-index	Name	Type	Attr.	Map.	Default value	Meaning
0x100A	0	Manufacturer Software version	Visible string	ro	N		Software version number CANopen software

1.xx

Da der zurückgelieferte Wert größer als 4Byte ist, wird das segmentierte SDO-Protokoll zur Übertragung verwendet.

**Guard time**

Index	Sub-index	Name	Type	Attr.	Map.	Default value	Meaning
0x100C	0	Guard time [ms]	Unsigned16	rw	N	0x0000	Interval between two guard telegrams. Is set by the NMT master or configuration tool.

**Life time factor**

Index	Sub-index	Name	Type	Attr.	Map.	Default value	Meaning
0x100D	0	Life time factor	Unsigned8	rw	N	0x00	Life time factor x guard time = life time (watchdog for life guarding)

Wenn innerhalb der Life Time kein Guarding-Telegramm empfangen wurde, geht der Knoten in den Fehlerzustand. Wenn "Life Time Factor" und / oder "Guard Time" = 0 sind, so führt der Knoten kein Lifeguarding durch, kann aber dennoch vom Master überwacht werden (Node Guarding).

**Consumer Heartbeat Time**

Index	Sub-index	Name	Type	Attr.	Map.	Default value	Meaning
0x1016	0	Consumer heartbeat time	Unsigned8	ro	N	0x05	Number of entries
	1...127		Unsigned32	rw	N	0x00000000	Consumer heartbeat time

Struktur des "Consumer Heartbeat Time" Eintrags:

Bits	31-24	23-16	15-0
Value	Reserved	Node-ID	Heartbeat time
Encoded as	Unsigned8	Unsigned8	Unsigned16

Sobald Sie versuchen, für die gleiche Node-ID eine "consumer heartbeat time" ungleich 0 zu konfigurieren, bricht der Knoten den SDO-Download ab und bringt den Fehlercode 0604 0043hex.

### Producer Heartbeat Time

Index	Sub-index	Name	Type	Attr.	Map.	Default value	Meaning
0x1017	0	Producer heartbeat time	Unsigned16	rw	N	0x0000	Defines the cycle time of heartbeat in ms

### Identity Object

Index	Sub-index	Name	Type	Attr.	Map.	Default value	Meaning
0x1018	0	Identity Object	Unsigned8	ro	N	0x04	Contains general Informations about the device (number of entries)
	1	Vendor ID	Unsigned32	ro	N	0xAFFEAF00	Vendor ID
	2	Product Code	Unsigned32	ro	N	0x2142CA02	Product Code
	3	Revision Number	Unsigned32	ro	N		Revision Number
	4	Serial Number	Unsigned32	ro	N		Serial Number

### Communication parameter RxPDO

Index	Sub-index	Name	Type	Attr.	Map.	Default value	Meaning
0x1400 ... 0x1427	0	Number of Elements	Unsigned8	ro	N	0x02	Communication parameter for the first receive PDOs, Subindex 0: number of following parameters
	1	COB-ID	Unsigned32	rw	N	0xC0000200 + NODE_ID	COB-ID RxPDO1
	2	Transmission type	Unsigned8	rw	N	0xFF	Transmission type of the PDO

Subindex 1 (COB-ID): Die unteren 11Bit des 32Bit-Wertes (Bits 0-10) enthalten den CAN-Identifizier, das MSBit (Bit 31) gibt Auskunft, ob das PDO aktiv ist (1) oder nicht (0), Bit 30 teilt mit, ob ein RTR-Zugriff auf dieses PDO zulässig ist (0) oder nicht (1).

Der Subindex 2 enthält die Übertragungsart.

**Mapping RxPDO**

Index	Sub-index	Name	Type	Attr.	Map.	Default value	Meaning
0x1600 ... 0x1627	0	Number of Elements	Unsigned8	rw	N	0x01	Mapping parameter of the first receive PDO; subindex 0: number of mapped objects
	1	1 <sup>st</sup> mapped object	Unsigned32	rw	N	0x62000108	(2 byte index, 1 byte subindex, 1 byte bit-width)
	2	2 <sup>nd</sup> mapped object	Unsigned32	rw	N	0x62000208	(2 byte index, 1 byte subindex, 1 byte bit-width)
	... 8	... 8 <sup>th</sup> mapped	... Unsigned32	... rw	... N	... 0x62000808	... (2 byte index, 1 byte subindex, 1 byte bit-width)

Die Empfangs-PDOs erhalten automatisch über den Koppler ein Default-Mapping abhängig von den angeschlossenen Modulen.

**Communication parameter TxPDO1**

Index	Sub-index	Name	Type	Attr.	Map.	Default value	Meaning
0x1800 ... 0x1827	0	Number of Elements	Unsigned8	ro	N	0x05	Communication parameter of the first transmit PDO, subindex 0: number of following parameters
	1	COB-ID	Unsigned32	rw	N	0x80000180 + NODE_ID	COB-ID TxPDO1
	2	Transmission type	Unsigned8	rw	N	0xFF	Transmission type of the PDO
	3	Inhibit time	Unsigned16	rw	N	0x0000	Repetition delay [value x 100 µs]
	5	Event time	Unsigned16	rw	N	0x0000	Event timer [value x 1 ms]

Subindex 1 (COB-ID): Die unteren 11Bit des 32Bit Wertes (Bits 0-10) enthalten den CAN-Identifizier, das MSBit (Bit 31) gibt Auskunft, ob das PDO aktiv ist (1) oder nicht (0), Bit 30 teilt mit, ob ein RTR-Zugriff auf dieses PDO zulässig ist (0) oder nicht (1). Der Subindex 2 enthält die Übertragungsart, Subindex 3 die Wiederholungsverzögerung zwischen zwei gleichen PDOs. Wenn ein "Event Timer" mit einem Wert ungleich 0 existiert, wird nach Ablauf dieses Timers das PDO übertragen.

Existiert ein "Inhibit Timer", wird das Ereignis um diese Zeit verzögert.

**Mapping TxPDO1**

Index	Sub-index	Name	Type	Attr.	Map.	Default value	Meaning
0x1A00 ...	0	Number of Elements	Unsigned8	rw	N	depending on the components fitted	Mapping parameter of the first transmit PDO;
0x1A27	1	1 <sup>st</sup> mapped object	Unsigned32	rw	N	0x60000108	subindex 0: number of mapped objects (2 byte index, 1 byte subindex, 1 byte bit-width)
	2	2 <sup>nd</sup> mapped object	Unsigned32	rw	N	0x60000208	(2 byte index, 1 byte subindex, 1 byte bit-width)
	...	...	...	...	...	...	...
	8	8 <sup>th</sup> mapped object	Unsigned32	rw	N	0x60000808	(2 byte index, 1 byte subindex, 1 byte bit-width)

Die Sende-PDOs erhalten automatisch über den Koppler ein Default-Mapping, abhängig von den angeschlossenen Modulen.

**Concise DCF**

Index	Sub-index	Name	Type	Attr.	Map.	Default value	Meaning
0x1F22	Array	Concise DCF	Domain	rw	N		

Dieses Objekt ist für den Configuration Manager erforderlich. Das Concise-DCF ist eine Kurzfassung des DCF (**D**evice **C**onfiguration **F**ile).

**Post Configuration**

Index	Sub-index	Name	Type	Attr.	Map.	Default value	Meaning
0x1F25	Array	ConfigureSlave	Unsigned32	rw	N	0x00000000	

Der Configuration Manager kann über diesen Eintrag angewiesen werden, eine gespeicherte Konfiguration in das Netz zu übertragen.

Die Konfiguration kann zu jeder Zeit über Index 0x1F25 für einen bestimmten Knoten ausgelöst werden.

Subindex 0 hat den Wert 128.

Subindex x (mit x = 1..127): Löst Rekonfiguration für Knoten mit der Node ID x aus.

Subindex 128: Rekonfiguration aller Knoten.

Soll z.B. für den Knoten 2 die Konfiguration ausgelöst werden und sind für diesen Knoten Konfigurationsdaten vorhanden, so ist der Wert 0x666E6F63 (ASCII = "conf") auf das Objekt 1F25h Subindex 2 zu schreiben.

**NMT Start-up**

Index	Sub-index	Name	Type	Attr.	Map.	Default value	Meaning
0x1F80	0x00	NMTStartup	Unsigned32	rw	N	0x00000000	

Hier geben Sie an, ob das Gerät der NMT-Master ist.

Bit	Meaning
Bit 0	0 : Device is NOT the NMT Master. All other bits have to be ignored. The objects of the Network List have to be ignored. 1 : Device is the NMT Master.
Bit 1	0 : Start only explicitly assigned slaves. 1 : After boot-up perform the service NMT Start Remote Node All Nodes
Bit 2..31	Reserved by CiA, always 0

**Slave Assignment**

Index	Sub-index	Name	Type	Attr.	Map.	Default value	Meaning
0x1F81	0x00	SlaveAssignment	Unsigned32	rw	N	0x00000000	

Hier erfolgt ein Eintrag der Knoten, die vom Master überwacht, kontrolliert und gesteuert werden sollen. Für jeden zugeordneten Knoten ist hier ein Eintrag vorzunehmen.

Subindex 0 hat den Wert 127. Jeder andere Subindex korrespondiert mit der Node-ID des Knotens.

Byte	Bit	Meaning
Byte 0	Bit 0	0: Node with this ID is not a slave 1: Node with this ID is a slave. After configuration (with Configuration Manager) the Node will be set to state Operational.
	Bit 1	0: On Error Control Event or other detection of a booting slave inform the application. 1: On Error Control Event or other detection of a booting slave inform the application and automatically start Error Control service.
	Bit 2	0: On Error Control Event or other detection of a booting slave do NOT automatically configure and start the slave. 1: On Error Control Event or other detection of a booting slave do start the process Start Boot Slave.
	Bit 3..7	Reserved by CiA, always 0
Byte 1		8 Bit Value for the RetryFactor
Byte 2,3		16 Bit Value for the GuardTime

**Request NMT**

Index	Sub-Index	Name	Type	Attr.	Map.	Default value	Meaning
0x1F82	0x00	RequestNMT	Unsigned32	rw	N	0x00000000	

Ist ein vollständig autonomer Start des Stacks nicht gewünscht, so können die Funktionalitäten:

- Statusumschaltung
- Starten des Guardings
- Konfiguration über CMT

auch für jeden Knoten einzeln auf Anfrage durchgeführt werden. Die Anfrage erfolgt immer über Objekte im Objektverzeichnis.

Die Umschaltung des Kommunikationsstatus aller im Netz vorhandenen Knoten (einschließlich des lokalen Slaves) wird dabei über den Eintrag 1F82h im lokalen Objektverzeichnis bewerkstelligt:

Subindex 0 hat den Wert 128.

Subindex x (with x=1..127): Löst NMT-Service für Knoten mit der Node ID x aus.

Subindex 128: Löst NMT-Service für alle Knoten aus.

Beim Schreiben wird der gewünschte Status als Wert angegeben

State	Value
Prepared	4
Operational	5
ResetNode	6
ResetCommunication	7
PreOperational	127

**Request Guarding**

Index	Sub-index	Name	Type	Attr.	Map.	Default value	Meaning
0x1F83	0x00	RequestGuarding	Unsigned32	rw	N	0x00000000	

Subindex 0 hat den Wert 128.

Subindex x (with x=1..127): Löst Guarding für den Slave mit Node ID x aus.

Value	Write Access	Read Access
1	Start Guarding	Slave actually is guarded
0	Stop Guarding	Slave actually is not guarded

Subindex 128: Request Start/Stop Guarding für alle Knoten.

**8bit Digital inputs**

Index	Sub-index	Name	Type	Attr.	Map.	Default value	Meaning
0x6000	0x00	8bit digital input block	Unsigned8	ro	N	0x01	Number of available digital 8bit input blocks
	0x01	1 <sup>st</sup> input block	Unsigned8	ro	Y		1 <sup>st</sup> digital input block
	... 0x40	... 64 <sup>th</sup> input block	... Unsigned8	... ro	... Y	...	... 64 <sup>th</sup> digital input block

**16bit Digital inputs**

Index	Sub-Index	Name	Type	Attr.	Map.	Default value	Meaning
0x6100	0x00	16bit digital input block	Unsigned8	ro	N	depending on the fitted components	Number of available digital 16bit input blocks
	0x01	1 <sup>st</sup> input block	Unsigned16	ro	N		1 <sup>st</sup> digital input block
	... 0x20	... 32 <sup>nd</sup> input block	... Unsigned16	... ro	... N	...	... 32 <sup>nd</sup> digital input block

**32bit Digital inputs**

Index	Sub-index	Name	Type	Attr.	Map.	Default value	Meaning
0x6120	0x00	32bit digital input block	Unsigned8	ro	N	depending on the components fitted	Number of available digital 32bit input blocks
	0x01	1 <sup>st</sup> input block	Unsigned32	ro	N		1 <sup>st</sup> digital input block
	... 0x10	... 16 <sup>th</sup> input block	... Unsigned32	... ro	... N	...	... 16 <sup>th</sup> digital input block

**8bit Digital outputs**

Index	Sub-index	Name	Type	Attr.	Map.	Default value	Meaning
0x6200	0x00	8bit digital output block	Unsigned8	ro	N	0x01	Number of available digital 8bit output blocks
	0x01	1 <sup>st</sup> output block	Unsigned8	rw	Y		1 <sup>st</sup> digital output block
	... 0x40	... 64 <sup>th</sup> output block	... Unsigned8	... rw	... Y	...	... 64 <sup>th</sup> digital output block

**16bit Digital outputs**

Index	Sub-index	Name	Type	Attr.	Map.	Default value	Meaning
0x6300	0x00	16bit digital input block	Unsigned8	ro	N	Depending on the components fitted	Number of available digital 16bit output blocks
	0x01	1 <sup>st</sup> output block	Unsigned16	rw	N		1 <sup>st</sup> digital output block
	... 0x20	... 32 <sup>nd</sup> output block	... Unsigned16	... rw	... N	...	... 32 <sup>nd</sup> digital output block

**32bit Digital outputs**

Index	Sub-index	Name	Type	Attr.	Map.	Default value	Meaning
0x6320	0x00	32bit digital input block	Unsigned8	ro	N	Depending on the components fitted	Number of available digital 32bit output blocks
	0x01	1 <sup>st</sup> output block	Unsigned32	rw	N		1 <sup>st</sup> digital output block
	... 0x10	... 16 <sup>th</sup> output block	... Unsigned32	... rw	... N	...	... 16 <sup>th</sup> digital output block

**8bit Network input variables**

Index	Sub-index	Name	Type	Attr.	Map.	Default value	Meaning
0xA040	0x00	8bit digital input block	Unsigned8	ro	N	0x01	Number of available digital 8bit input blocks
	0x01	1 <sup>st</sup> input block	Unsigned8	ro	Y		1 <sup>st</sup> digital input block
	...	...	...	...	...	...	...
	0x140	320 <sup>th</sup> input block	Unsigned8	ro	Y		320 <sup>th</sup> digital input block

**16bit Network input variables**

Index	Sub-index	Name	Type	Attr.	Map.	Default value	Meaning
0xA100	0x00	16bit digital input block	Unsigned8	ro	N	depending on the fitted components	Number of available digital 16bit input blocks
	0x01	1 <sup>st</sup> input block	Unsigned16	ro	N		1 <sup>st</sup> digital input block
	...	...	...	...	...	...	...
	0xA0	160 <sup>th</sup> input block	Unsigned16	ro	N		160 <sup>th</sup> digital input block

**32bit Network input variables**

Index	Sub-index	Name	Type	Attr.	Map.	Default value	Meaning
0xA200	0x00	32bit digital input block	Unsigned8	ro	N	depending on the components fitted	Number of available digital 32bit input blocks
	0x01	1 <sup>st</sup> input block	Unsigned32	ro	N		1 <sup>st</sup> digital input block
	...	...	...	...	...	...	...
	0x50	80 <sup>th</sup> input block	Unsigned32	ro	N		80 <sup>th</sup> digital input block

**64bit Network input variables**

Index	Sub-index	Name	Type	Attr.	Map.	Default value	Meaning
0xA440	0x00	64bit digital input block	Unsigned8	ro	N	depending on the components fitted	Number of available digital 64bit input blocks
	0x01	1 <sup>st</sup> input block	Unsigned32	ro	N		1 <sup>st</sup> digital input block
	...	...	...	...	...	...	...
	0x28	40 <sup>th</sup> input block	Unsigned32	ro	N		40 <sup>th</sup> digital input block

**8bit Network output variables**

Index	Sub-index	Name	Type	Attr.	Map.	Default value	Meaning
0xA400	0x00	8bit digital output block	Unsigned8	ro	N	0x01	Number of available digital 8bit output blocks
	0x01	1 <sup>st</sup> output block	Unsigned8	rw	Y		1 <sup>st</sup> digital output block
	...	...	...	...	...	...	...
	0x140	320 <sup>th</sup> output block	Unsigned8	rw	Y		320 <sup>th</sup> digital output block

**16bit Network output variables**

Index	Sub-index	Name	Type	Attr.	Map.	Default value	Meaning
0xA580	0x00	16bit digital input block	Unsigned8	ro	N	Depending on the components fitted	Number of available digital 16bit output blocks
	0x01	1 <sup>st</sup> output block	Unsigned16	rw	N		1 <sup>st</sup> digital output block
	...	...	...	...	...	...	...
	0xA0	160 <sup>th</sup> output block	Unsigned16	rw	N		160 <sup>th</sup> digital output block

**32bit Network output variables**

Index	Sub-index	Name	Type	Attr.	Map.	Default value	Meaning
0xA680	0x00	32bit digital input block	Unsigned8	ro	N	Depending on the components fitted	Number of available digital 32bit output blocks
	0x01	1 <sup>st</sup> output block	Unsigned32	rw	N		1 <sup>st</sup> digital output block
	...	...	...	...	...	...	...
	0x50	80 <sup>th</sup> output block	Unsigned32	rw	N		80 <sup>th</sup> digital output block

**64bit Network output variables**

Index	Sub-index	Name	Type	Attr.	Map.	Default value	Meaning
0xA8C0	0x00	64bit digital input block	Unsigned8	ro	N	Depending on the components fitted	Number of available digital 64bit output blocks
	0x01	1 <sup>st</sup> output block	Unsigned32	rw	N		1 <sup>st</sup> digital output block
	...	...	...	...	...	...	...
	0x50	40 <sup>th</sup> output block	Unsigned32	rw	N		40 <sup>th</sup> digital output block

## Teil 10 Einsatz CPU 21xSER-1

### Überblick

Inhalt dieses Kapitels ist der Einsatz der CPU 21xSER-1 mit RS232/RS485-Schnittstelle.

Sie erhalten hier alle Informationen, die zum Einsatz der seriellen Schnittstelle, der CPU 21xSER-1 erforderlich sind.

### Inhalt

Thema	Seite
<b>Teil 10 Einsatz CPU 21xSER-1</b> .....	<b>10-1</b>
Schnelleinstieg .....	10-2
Protokolle und Prozeduren .....	10-3
Einsatz der seriellen Schnittstelle .....	10-7
Prinzip der Datenübertragung .....	10-8
Parametrierung .....	10-10
Kommunikation .....	10-14
Modemfunktionalität .....	10-20
Modbus Slave Funktionscodes.....	10-21
Modbus - Beispiel zur Kommunikation.....	10-25

## Schnelleinstieg

**Allgemein** Die CPU 21xSER-1 ermöglicht die serielle Prozessanbindung zu verschiedenen Ziel- oder Quellsystemen. Zur seriellen Kommunikation besitzt die CPU 21x-2BS12 eine RS232-Schnittstelle und die CPU 21x-2BS32 eine RS485-Schnittstelle.

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

**Parametrierung** Die Parametrierung erfolgt zur Laufzeit unter Einsatz des SFC 216 (SER\_CFG). Hierbei sind für alle Protokolle mit Ausnahme von ASCII die Parameter in einem DB abzulegen.

**Kommunikation** Mit SFCs steuern Sie die Kommunikation. Das Senden erfolgt unter Einsatz des SFC 217 (SER\_SND) und das Empfangen über SFC 218 (SER\_RCV).

Durch erneuten Aufruf des 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 SFC 218 SER\_RCV nach einem SER\_SND das Quittungstelegramm auslesen.

Die SFCs befinden sich im Lieferumfang der CPU 21xSER-1.

### Übersicht der SFCs für die serielle Kommunikation

Folgende SFCs kommen für die serielle Kommunikation zum Einsatz:

SFC		Beschreibung
SFC 216	SER_CFG	RS232/RS485 Parametrieren
SFC 217	SER_SND	RS232/RS485 Senden
SFC 218	SER_RCV	RS232/RS485 Empfangen
SFC 207	SER_CTRL	Modemfunktionalität

## Protokolle und Prozeduren

### Übersicht

Die CPU 21xSER-1 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-SFC die zum Zeitpunkt des Aufrufs im Puffer enthaltenen Daten im parametrierten 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.de](http://www.vipa.de).

### 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.

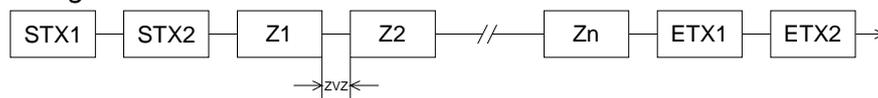
Die Prozedur STX/ETX wird zur Übertragung von ASCII-Zeichen eingesetzt. Sie arbeitet ohne Blockprüfung (BCC). Sollen Daten von der Peripherie eingelesen werden, muss das Start-Zeichen vorhanden sein, anschließend folgen die zu übertragenden Zeichen. Danach muss das Ende-Zeichen vorliegen.

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 SFC 217 (SER\_SND) übergeben und von dort mit angefügten Start- und Endezeichen über die serielle Schnittstelle an den Kommunikationspartner übertragen.

*Telegrammaufbau:*



Sie können bis zu 2 Anfangs- und Endezeichen frei definieren.

Es kann mit 1, 2 oder keiner Start- und mit 1, 2 oder keiner Ende-Kennung gearbeitet werden. Als Start- bzw. Ende-Kennung sind alle Hex-Werte von 01h 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). Wird kein Ende-Zeichen definiert, so werden alle gelesenen Zeichen nach Ablauf einer parametrierbaren Zeichenverzugszeit (Timeout) an die CPU übergeben.

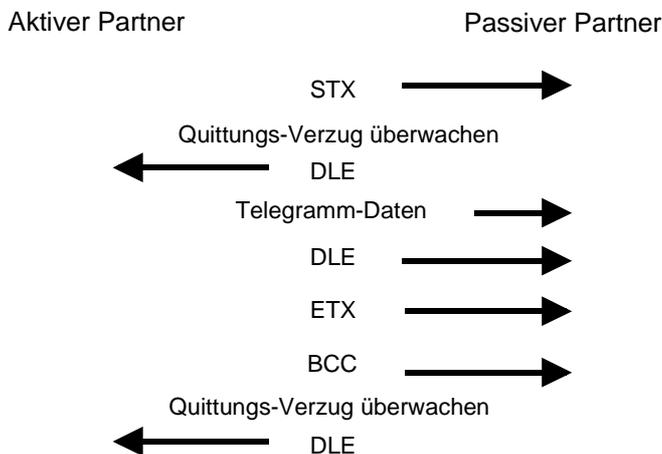
**3964R**

Die Prozedur 3964R steuert die Datenübertragung bei einer Punkt-zu-Punkt-Kopplung zwischen der CPU 21xSER-1 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            **Start of Text**
- DLE           **Data Link Escape**
- ETX           **End of Text**
- BCC           **Block Check Character**
- NAK           **Negative Acknowledge**

**Prozedurablauf**



Sie können pro Telegramm maximal 255Byte übertragen.



**Hinweis!**

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

Am Bus können 1 Master und max. 31 Slaves angebunden sein, wobei die einzelnen Slaves vom Master über ein Adresszeichen im Telegramm angewählt werden. Die Kommunikation erfolgt ausschließlich über den Master im Halbduplex-Betrieb.

Nach einem Sendeauftrag ist das Quittungstelegramm durch Aufruf des SFC 218 SER\_RCV auszulesen.

Die Telegramme für Senden und Empfangen haben folgenden Aufbau:

*Master-Slave-Telegramm*

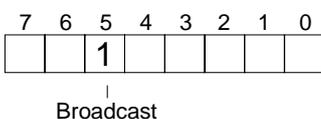
STX	LGE	ADR	PKE		IND		PWE		STW		HSW		BCC
02h			H	L	H	L	H	L	H	L	H	L	

*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: Hauptwert             |
|     | IND: Index            | BCC: Block Check Character |
|     | PWE: Parameterwert    |                            |

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" Sendeauftrag ist beim Broadcast keine Telegrammauswertung über 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 SFC 218 SER\_RCV auszulesen.

Die Anforderungs-Telegramme, die ein Master sendet und die Antwort-Telegramme eines Slaves haben den gleichen Aufbau:

Start- zeichen	Slave- Adresse	Funktions- Code	Daten	Fluss- kontrolle	Ende- zeichen
-------------------	-------------------	--------------------	-------	---------------------	------------------

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 SFC 218 SER\_RCV erforderlich.

Nur Schreibaufträge dürfen als Broadcast gesendet werden.

ASCII-, RTU-Modus

Bei Modbus gibt es zwei unterschiedliche Übertragungsmodi

- 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.

Die Modus-Wahl erfolgt zur Laufzeit unter Einsatz des SFC 216 SER\_CFG.

# Einsatz der seriellen Schnittstelle

## Übersicht

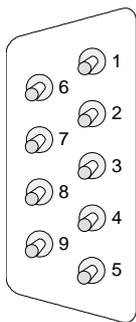
Je nach verwendeter Hardware besitzt Ihre CPU eine RS232- (Best.-Nr.: 21x-2BS12) oder RS485-Schnittstelle (Best.-Nr.: 21x-2BS32). Die beiden Schnittstellen sind nachfolgend beschrieben.

### RS232-Schnittstelle

- Schnittstelle ist kompatibel zur COM Schnittstelle eines PCs
- Logische Zustände als Spannungspegel
- Punkt-zu-Punkt-Kopplung mit serieller Vollduplex-Übertragung bis zu einer Entfernung von 15m
- Datenübertragungsrate bis 115,2kBaude
- Unterstützt werden ASCII, STX/ETX, 3964R, USS und Modbus
- Empfangs- und Sendepuffer haben jeweils eine Größe von 2x256Byte.
- Die maximale Telegrammlänge beträgt 255Byte.

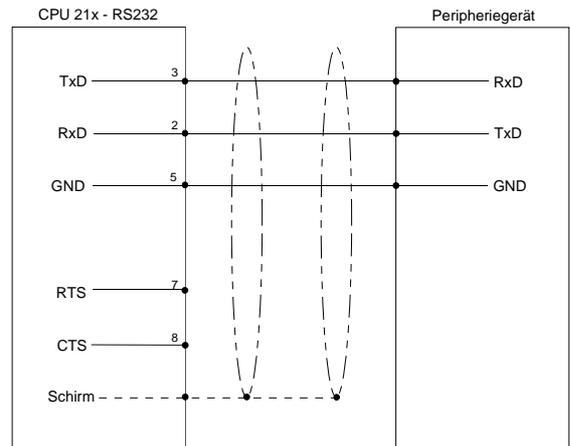
### Anschluss RS232

9poliger Stecker



Pin	RS232C
1	CD-
2	RxD
3	TxD
4	DTR-
5	GND
6	DSR-
7	RTS-
8	CTS-
9	RI-

### Anschluss RS232

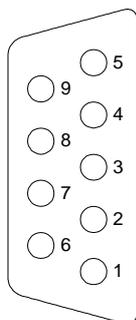


### RS485-Schnittstelle

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

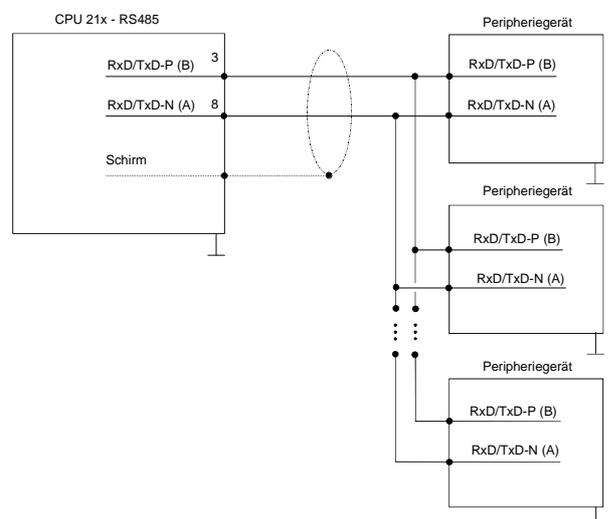
### Anschluss RS485

9poliger Buchse



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

### Anschluss RS485



# Prinzip der Datenübertragung

## Übersicht

Die Datenübertragung wird zur Laufzeit über SFCs gehandhabt. Das Prinzip der Datenübertragung ist bis auf Modbus-Slave für alle Protokolle identisch und soll hier kurz gezeigt werden.

## Prinzip für ASCII, STX/ETX, 3964R, Modbus-Master und USS

Daten, die von der CPU in den entsprechenden Datenkanal geschrieben werden, werden in einen FIFO-Sendepuffer (first in first out) mit einer Größe von 2x256Byte abgelegt und von dort über die Schnittstelle ausgegeben.

Empfängt die Schnittstelle Daten, werden diese in einem FIFO-Empfangspuffer mit einer Größe von 2x256Byte abgelegt und können dort von der CPU gelesen werden.

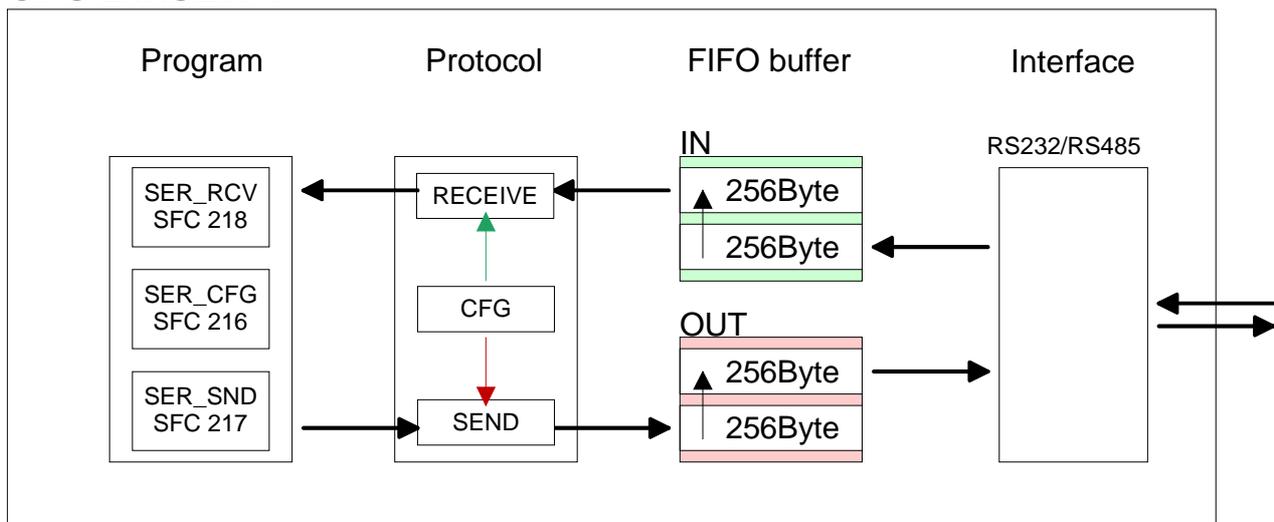
Sofern Daten mittels eines Protokolls übertragen werden, erfolgt die Einbettung der Daten in das entsprechende Protokoll automatisch.

Im Gegensatz zu ASCII- und STX/ETX erfolgt bei den Protokollen 3964R, Modbus-Master und USS die Datenübertragung mit Quittierung der Gegenseite.

Durch erneuten Aufruf des 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 Modbus-Master und USS nach einem SER\_SND das Quittungstelegramm durch Aufruf des SFC 218 SER\_RCV auszulesen.

## CPU 21xSER-1



**Prinzip für Modbus-Slave**

Daten, die von der CPU dem Modbus-Master zur Verfügung zu stellen sind, werden in einem FIFO-Sendepuffer (first in first out) mit einer Größe von 2x256Byte abgelegt. Im Gegensatz zu den anderen Protokollen bleiben die Daten im Sendepuffer, bis diese vom Modbus-Master über einen Lesebefehl (Funktionscode 01h, 03h) anfordert werden.

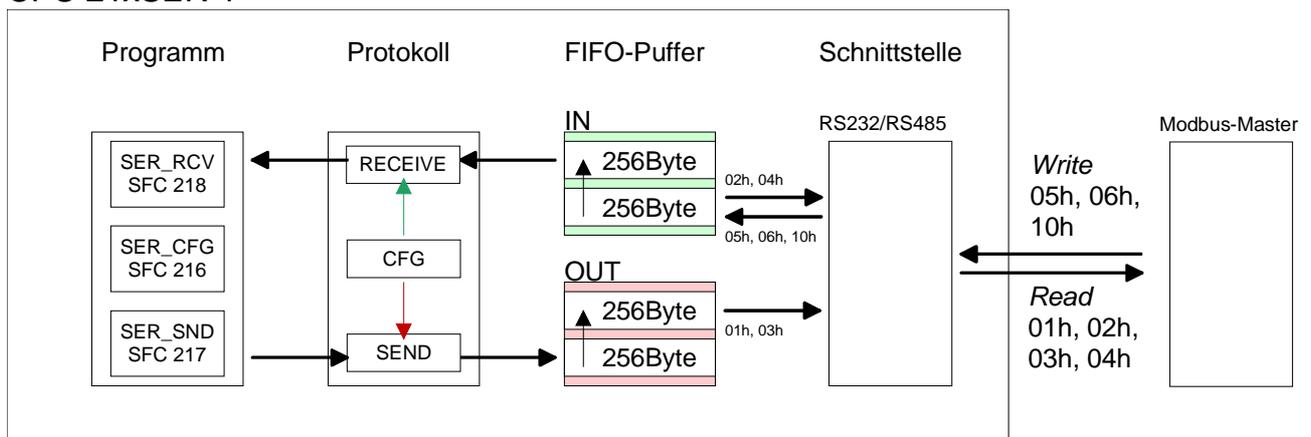
Empfängt die Schnittstelle Daten vom Master (Funktionscode 05h, 06h, 10h), werden diese in einem FIFO-Empfangspuffer mit einer Größe von 2x256Byte abgelegt und können dort von der CPU gelesen werden.

Die Einbettung der Daten in das Modbus-Protokoll erfolgt automatisch.

Bitte beachten Sie, dass der Modbus-Master durch entsprechende Vorgabe der Lese-Funktionscodes auf den IN- bzw. OUT-Puffer zugreifen kann. Mit einem Lesezugriff auf den IN-Puffer (Funktionscode 02h, 04h) kann der Master die Daten lesen, die er zuvor an den Modbus-Slave geschickt hat. Diese Daten bleiben solange im Puffer, bis diese vom Modbus-Master überschrieben werden.

In der nachfolgenden Abbildung ist das Kommunikationsprinzip aufgeführt. Weitere Informationen finden Sie auch im Kapitel "Modbus Slave Funktionscodes" weiter unten.

CPU 21xSER-1



## Parametrierung

### SFC 216 (SER\_CFG)

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

Bitte beachten Sie, dass nicht für alle Protokolle der gesamte Wertebereich der Parameter unterstützt wird. Näheres hierzu finden Sie direkt bei der Beschreibung des entsprechenden Parameters.



#### Hinweis!

Bitte beachten Sie, dass der SFC216 während einer Kommunikation nicht mehr aufgerufen wird, da hierdurch alle Puffer gelöscht werden.

Sollen keine Kommunikations-Parameter mehr geändert werden, sollten Sie den Aufruf des SFC 216 in den Anlauf-OB OB 100 legen.

Name	Deklaration	Typ	Beschreibung
Protocol	IN	BYTE	No. of protocol
Parameter	IN	ANY	Pointer to protocol-parameters
Baudrate	IN	BYTE	No of Baudrate
CharLen	IN	BYTE	0=5Bit, 1=6Bit, 2=7Bit, 3=8Bit
Parity	IN	BYTE	0=None, 1=Odd, 2=Even
StopBits	IN	BYTE	1=1Bit, 2=1,5Bit, 3=2Bit
FlowControl	IN	BYTE	Handshake
RetVal	OUT	WORD	Return Code ( 0 = OK )

### 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
- 7: Modbus RTU Slave
- 8: Modbus ASCII Slave

**Parameter (als DB)** Bei eingestelltem ASCII-Protokoll wird dieser Parameter ignoriert.  
Für die Protokolle STX/ETX, 3964R, USS und Modbus geben Sie hier einen DB an, der die Kommunikationsparameter beinhaltet und für die jeweiligen Protokolle 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 im Zeitraster von 10ms)



**Hinweis!**

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

*Datenbaustein bei 3964R*

DBB0:	Prio	BYTE	(Die Priorität beider Partner muss unterschiedlich sein. Prio 0 und 1 ist möglich)
DBB1:	ConnAttmptNr	BYTE	(Anzahl der Verbindungsaufbauversuche)
DBB2:	SendAttmptNr	BYTE	(Anzahl der Telegrammwiederholungen)
DBW4:	CharTimeout	WORD	(Zeichenverzugszeit in 10ms Zeitraster)
DBW6:	ConfTimeout	WORD	(Quittungsverzugszeit in 10ms Zeitraster)

*Datenbaustein bei USS*

DBW0:	Timeout	WORD	(Verzugszeit in 10ms Zeitraster)
-------	---------	------	----------------------------------

*Datenbaustein bei Modbus-Master*

DBW0:	Timeout	WORD	(Antwort-Verzugszeit in 10ms Zeitraster)
-------	---------	------	--

*Datenbaustein bei Modbus-Slave*

DBB0:	Adresse	BYTE	(Adresse 1...247 im Modbus-Netz)
DBW2:	Timeout	WORD	(Antwort-Verzugszeit in 10ms Zeitraster)

**Baudrate** Geschwindigkeit der Datenübertragung in Bit/s (Baud).  
 01h: 150 Baud 05h: 1800 Baud 09h: 9600 Baud 0Dh: 57600 Baud  
 02h: 300 Baud 06h: 2400 Baud 0Ah: 14400 Baud 0Eh: 115200 Baud  
 03h: 600 Baud 07h: 4800 Baud 0Bh: 19200 Baud  
 04h: 1200 Baud 08h: 7200 Baud 0Ch: 38400 Baud

**CharLen** Anzahl der Datenbits, auf die ein Zeichen abgebildet wird.  
 0: 5Bit 1: 6Bit 2: 7Bit 3: 8Bit

Unterstützte Werte:

Bit	ASCII	STX/ETX	3964R	USS	Modbus RTU	Modbus ASCII
5	x		x			
6	x	x	x			
7	x	x	x			x
8	x	x	x	x	x	x

**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  
 Die 1,5Bit können ausschließlich bei einer CharLen von 5 verwendet werden, bei dieser Datenlänge sind 2Bit sind nicht möglich.

**FlowControl** Mit diesem Bit beeinflussen Sie das Verhalten der Request to send-Leitung.  
 0: RTS wird nicht beeinflusst  
 1: RTS ist 1 beim "Senden" (AutoRTS)  
    RTS ist 0 beim "Empfangen" (AutoRTS)  
 2: HW Flow (nur bei ASCII Protokollen)

Hinweis: Bei RS485 wird FlowControl nicht ausgewertet.  
 FlowControl = "1" (AutoRTS).

**RetVal  
(Rückgabewert)**

Wert	Beschreibung
0000h	kein Fehler
809Ah	Schnittstelle ist nicht vorhanden
8x24h	Fehler in SFC-Parameter x, mit x: 1: Fehler in "Protokoll" 2: Fehler in "Parameter" 3: Fehler in "Baudrate" 4: Fehler in "CharLength" 5: Fehler in "Parity" 6: Fehler in "StopBits" 7: Fehler in "FlowControl"
809xh	Fehler in Wert des SFC-Parameter x, mit x: 1: Fehler in "Protokoll" 3: Fehler in "Baudrate" 4: Fehler in "CharLength" 5: Fehler in "Parity" 6: Fehler in "StopBits" 7: Fehler in "FlowControl"
8092h	Zugriffsfehler auf Parameter-DB (DB zu kurz)
828xh	Fehler in Parameter x von DB-Parameter mit x: 1: Fehler 1. Parameter 2: Fehler 2. Parameter ...

## Kommunikation

<b>Übersicht</b>	Die Kommunikation erfolgt über die Sende- und Empfangsbausteine SFC 217 (SER_SND) und SFC 218 (SER_RCV). Sofern Daten mittels eines Protokolls übertragen werden, erfolgt die Einbettung der Daten in das entsprechende Protokoll automatisch. Je nach Protokoll sind folgende Dinge zu beachten
ASCII STX/ETX	Bei ASCII bzw. STX/ETX erfolgt das Senden der Daten ohne Quittierung der Gegenseite.
3964R	Durch erneuten Aufruf des SFC 217 SER_SND bekommen Sie über RetVal einen Rückgabewert geliefert, der unter anderem auch aktuelle Informationen über die Quittierung der Gegenseite beinhaltet.
Modbus-Master USS	Das Senden erfolgt mit Quittierung der Gegenseite. Durch erneuten Aufruf des SFC 217 SER_SND bekommen Sie über RetVal einen Rückgabewert geliefert, der unter anderem auch aktuelle Informationen über die Quittierung der Gegenseite beinhaltet. Nach erfolgter Übertragung mit SER_Send erhalten Sie durch Aufruf des SFC 218 SER_RCV das Quittungstelegramm der Gegenseite.



### Hinweis!

Bitte beachten Sie, dass während einer Kommunikation der SFC 216 (SER\_CFG) nicht mehr aufgerufen wird, da hierdurch alle Puffer gelöscht werden.

### SFC 217 (SER\_SND)

Mit diesem Baustein werden Daten über die serielle Schnittstelle gesendet.

Name	Deklaration	Typ	Beschreibung
DataPtr	IN	ANY	Pointer to Data Buffer for sending data
DataLen	OUT	WORD	Length of data sent
RetVal	OUT	WORD	Error Code ( 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.  
Bei **STX/ETX** und **3964R** wird immer die unter DataPtr angegebene Länge oder 0 eingetragen.

Werden unter **ASCII** die Daten intern mittels 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!

**RetVal  
(Rückgabewert)**

Wert	Beschreibung
0000h	Daten gesendet - fertig
1000h	Nichts gesendet (Datenlänge 0)
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 SFC-Parameter x, mit x: 1: Fehler in "DataPtr" 2: Fehler in "DataLen"
8122h	Fehler in Parameter "DataPtr" (z.B. DB zu kurz)
807Fh	Interner Fehler
809Ah	Schnittstelle nicht vorhanden
809Bh	Schnittstelle nicht konfiguriert

**Protokollspezi-  
fische 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 (>256Byte)
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)
9001h	Daten sind zu lang (>256Byte)
9002h	Daten sind zu kurz (0Byte)

## USS

Wert	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 (>256Byte)
9002h	Daten sind zu kurz (<2Byte)

*Modbus RTU/ASCII Master*

Wert	Beschreibung
2000h	Senden fertig ohne Fehler
2001h	Senden fertig mit Fehler
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 (>256Byte)
9002h	Daten sind zu kurz (<2Byte)

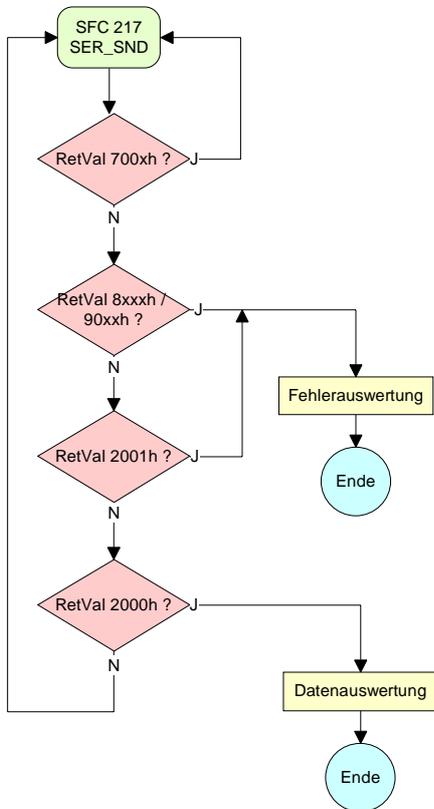
*Modbus RTU/ASCII Slave*

Wert	Beschreibung
0000h	Senden fertig ohne Fehler
9001h	Daten sind zu lang (>256Byte)

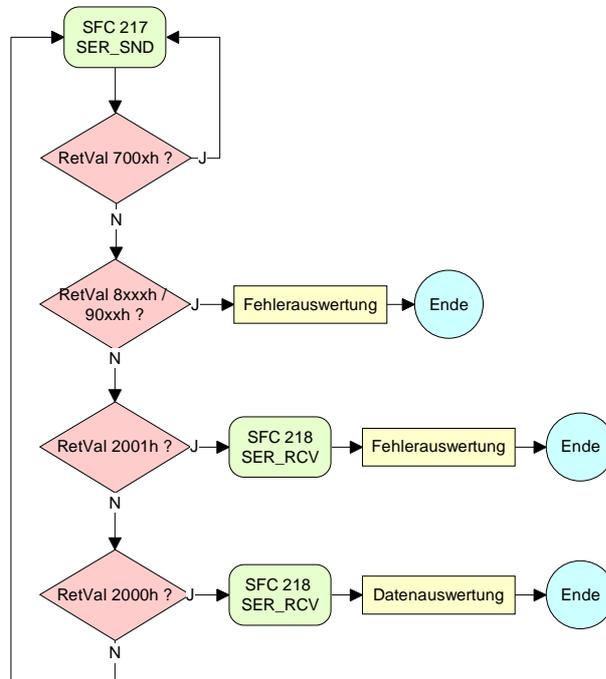
**Prinzip der Programmierung**

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

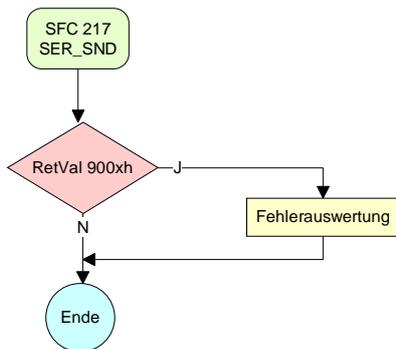
**3964R**



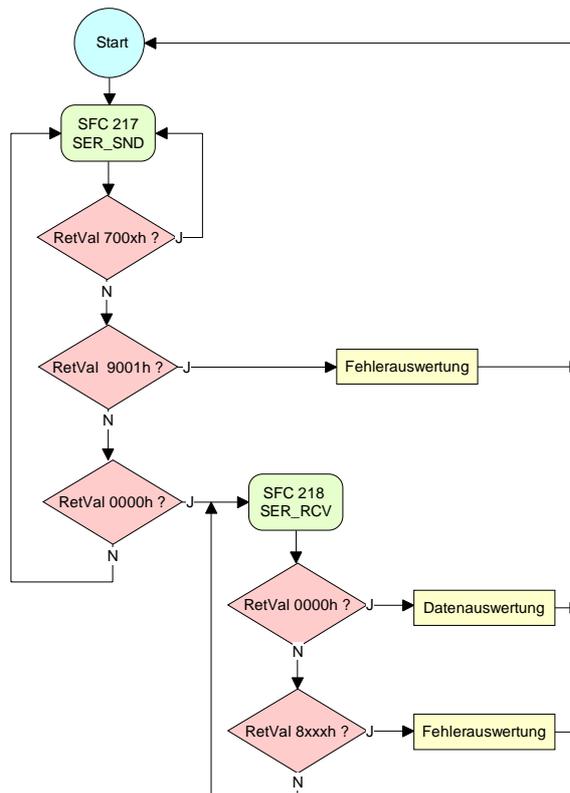
**USS / Modbus-Master**



**ASCII / STX/ETX**



**Modbus Slave**



**SFC 218  
(SER\_RCV)**

Mit diesem Baustein werden Daten über die serielle Schnittstelle empfangen.

Name	Deklaration	Typ	Beschreibung
DataPtr	IN	ANY	Pointer to Data Buffer for received data
DataLen	OUT	WORD	Length of received data
Error	OUT	WORD	Error Number
RetVal	OUT	WORD	Error Code ( 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 gelesenen Telegrammlänge abweichen.

**Error**

In diesem Wort erfolgt ein Eintrag im Fehlerfall unter ASCII. Folgende Fehlermeldungen können generiert werden:

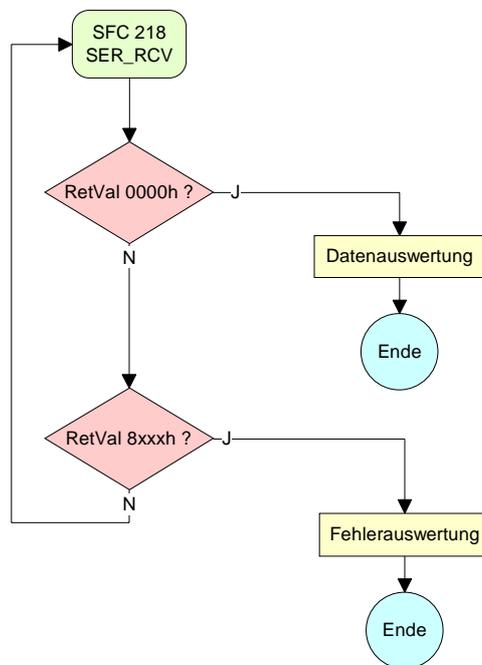
Bit	Fehler	Beschreibung
1	overrun	Überlauf, ein Zeichen konnte nicht schnell genug aus der Schnittstelle gelesen werden.
2	parity	Paritätsfehler
3	framing error	Fehler, der anzeigt, dass ein definierter Bitrahmen nicht übereinstimmt, die zulässige Länge überschreitet oder eine zusätzliche Bitfolge enthält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

**RetVal  
(Rückgabewert)**

Wert	Beschreibung
0000h	kein Fehler
1000h	Empfangspuffer ist zu klein (Datenverlust)
8x24h	Fehler in SFC-Parameter x, mit x: 1: Fehler in "DataPtr" 2: Fehler in "DataLen" 3: Fehler in "Error"
8122h	Fehler in Parameter "DataPtr" (z.B. DB zu kurz)
809Ah	Schnittstelle nicht vorhanden
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.



## Modemfunktionalität

### SFC 207 SER\_CTRL

Bei Einsatz des ASCII-Protokolls über die RS232-Schnittstelle haben Sie mit diesem Baustein zur Laufzeit Zugriff auf die seriellen Modemleitungen. Abhängig vom Parameter *FlowControl*, den Sie über *SFC 216 (SER\_CFG)* vorgeben, bietet der Baustein folgende Funktionalität:

*FlowControl=0*: Lesen: DTR, RTS, DSR, RI, CTS, CD  
Schreiben: DTR, RTS

*FlowControl>0*: Lesen: DTR, RTS, DSR, RI, CTS, CD  
Schreiben: nicht möglich

#### Parameter

Name	Deklaration	Typ	Beschreibung
Write	IN	BYTE	Bit 0: New state DTR Bit 1: New state RTS
MaskWrite	IN	BYTE	Bit 0: Set state DTR Bit 1: Set state RTS
Read	OUT	BYTE	Status flags (CTS, DSR, RI, CD, DTR, RTS)
ReadDelta	OUT	BYTE	Status flags of change between 2 accesses
RetVal	OUT	WORD	Return Code ( 0 = OK )

#### Write

Mit diesem Parameter geben Sie den Status für DTR und RTS vor, den Sie über *MaskWrite* aktivieren können. Das Byte hat folgende Belegung:

Bit 0 = DTR  
Bit 1 = RTS  
Bit 2 ... Bit 7: reserviert

#### MaskWrite

Hier wird mit "1" der Status des entsprechenden Parameters übernommen. Das Byte hat folgende Belegung:

Bit 0 = DTR  
Bit 1 = RTS  
Bit 2 ... Bit 7: reserviert

#### Read

*Read* liefert den aktuellen Status der Modem-Leitungen zurück. *ReadDelta* liefert den Status der Modem-Leitungen zurück, die sich seit dem letzten Zugriff geändert haben. Die Bytes haben folgenden Aufbau:

Bit-Nr.	7	6	5	4	3	2	1	0
Read	x	x	RTS	DTR	CD	RI	DSR	CTS
ReadDelta	x	x	x	x	CD	RI	DSR	CTS

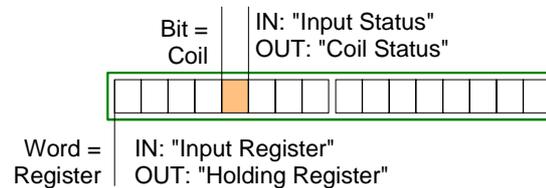
#### RetVal (Rückgabewert)

Return Code	Beschreibung
0000h	kein Fehler
8x24h	Fehler in SFC-Parameter x, mit x: 1: Fehler in <i>Write</i> 2: Fehler in <i>MaskWrite</i> 3: Fehler in <i>Read</i> 4: Fehler in <i>ReadDelta</i>
809Ah	Schnittstelle ist nicht vorhanden
809Bh	Schnittstelle ist nicht konfiguriert (SFC 216)

## Modbus Slave 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.

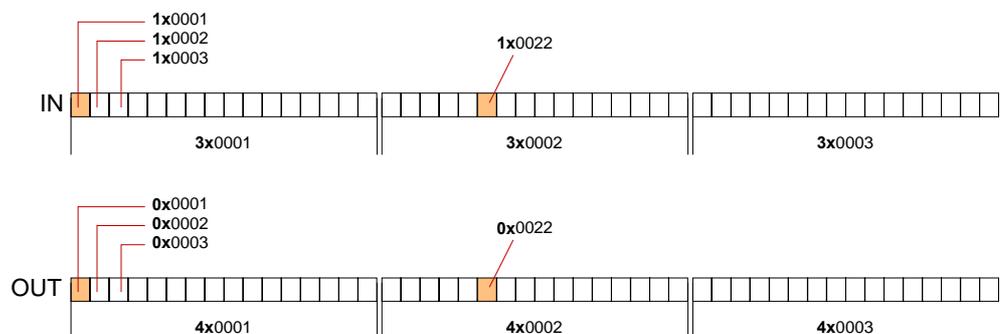
### Bereichsdefinitionen

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 der CPU 21xSER-1 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
- 1x: Bit-Bereich für Eingabe-Daten des Masters  
Zugriff über Funktions-Code 02h
- 3x: Wortbereich-Bereich für Eingabe-Daten des Masters  
Zugriff über Funktions-Code 04h
- 4x: Wortbereich-Bereich für Ausgabe-Daten des Masters  
Zugriff über Funktions-Code 03h, 06h, 10h



Eine Beschreibung der Funktions-Codes finden Sie auf den Folgeseiten.

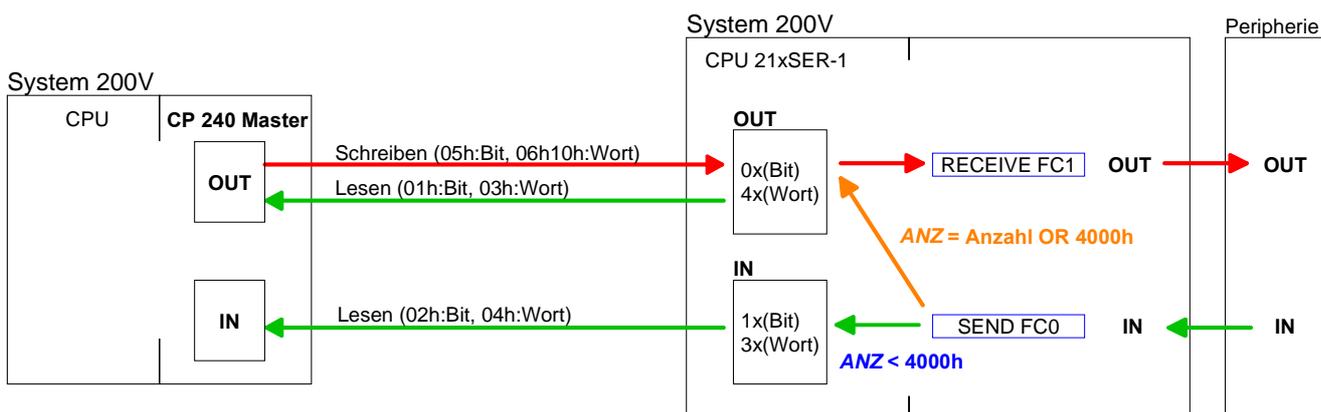
**Übersicht**

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
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.



**Master-Ausgabe-Bereich beschreiben**

Durch "Ver-Oderung" des FC 0 Parameters ANZ mit 4000h werden zu sendende Slave-Daten nicht im Master-Eingabe- sondern im Master-Ausgabe-Bereich abgelegt. Da der Master unter Einsatz von Funktionscodes diesen Bereich lesen kann, können Sie diese Funktionalität beispielsweise zur direkten Fehlerübermittlung an den Master verwenden.

**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**

Für die Byte-Reihenfolge im Wort gilt immer: *1 Wort*  
 High- Low-  
 Byte Byte

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

## Kommandotelegramm

RTU/ASCII-Rahmen	Slave-Adresse	Funktions-Code	Adresse 1. Bit	Anzahl der Bits	RTU/ASCII-Rahmen
	1Byte	1Byte	1Wort	1Wort	1Wort

## Antworttelegramm

RTU/ASCII-Rahmen	Slave-Adresse	Funktions-Code	Anzahl der gelesenen Bytes	Daten 1. Byte	Daten 2. Byte	...	RTU/ASCII-Rahmen
	1Byte	1Byte	1Byte	1Byte	1Byte		1Wort
					max. 252Byte		

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

## Kommandotelegramm

RTU/ASCII-Rahmen	Slave-Adresse	Funktions-Code	Adresse 1. Wort	Anzahl der Worte	RTU/ASCII-Rahmen
	1Byte	1Byte	1Wort	1Wort	1Wort

## Antworttelegramm

RTU/ASCII-Rahmen	Slave-Adresse	Funktions-Code	Anzahl der gelesenen Bytes	Daten 1. Wort	Daten 2. Wort	...	RTU/ASCII-Rahmen
	1Byte	1Byte	1Byte	1Wort	1Wort		1Wort
					max. 125Worte		

**Write 1 Bit** Code 05h: 1 Bit schreiben in Master-Ausgabe-Bereich 0x  
**05h** Eine Zustandsänderung erfolgt unter "Zustand Bit" mit folgenden Werten:

"Zustand Bit" = 0000h → Bit = 0

"Zustand Bit" = FF00h → Bit = 1

## Kommandotelegramm

RTU/ASCII-Rahmen	Slave-Adresse	Funktions-Code	Adresse Bit	Zustand Bit	RTU/ASCII-Rahmen
	1Byte	1Byte	1Wort	1Wort	1Wort

## Antworttelegramm

RTU/ASCII-Rahmen	Slave-Adresse	Funktions-Code	Adresse Bit	Zustand Bit	RTU/ASCII-Rahmen
	1Byte	1Byte	1Wort	1Wort	1Wort

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

Kommandotelegramm

RTU/ASCII-Rahmen	Slave-Adresse	Funktions-Code	Adresse Wort	Wert Wort	RTU/ASCII-Rahmen
	1Byte	1Byte	1Wort	1Wort	1Wort

Antworttelegramm

RTU/ASCII-Rahmen	Slave-Adresse	Funktions-Code	Adresse Wort	Wert Wort	RTU/ASCII-Rahmen
	1Byte	1Byte	1Wort	1Wort	1Wort

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

Kommandotelegramm

RTU/ASCII-Rahmen	Slave-Adresse	Funktions-Code	Adresse 1. Wort	Anzahl Worte	Anzahl Bytes	Daten 1. Wort	Daten 2. Wort	...	RTU/ASCII-Rahmen
	1Byte	1Byte	1Wort	1Wort	1Byte	1Wort	1Wort	1Wort	1Wort
						max. 124Worte			

Antworttelegramm

RTU/ASCII-Rahmen	Slave-Adresse	Funktions-Code	Adresse 1. Wort	Anzahl Worte	RTU/ASCII-Rahmen
	1Byte	1Byte	1Wort	1Wort	1Wort

## Modbus - Beispiel zur Kommunikation

### Übersicht

In dem Beispiel wird eine Kommunikation zwischen einem Master und einem Slave über Modbus aufgebaut. Gezeigt werden folgende Kombinationsmöglichkeiten:

Modbus-Master (M)	Modbus-Slave (S)
CPU 21xSER-1	CPU 21xSER-1
CPU 21xSER-1	CP 240

### M: CPU 21xSER-1 S: CPU 21xSER-1

Folgende Komponenten sind für das Beispiel erforderlich:

- 2 CPU 21xSER-1 als Modbus RTU-Master bzw. Modbus RTU-Slave
- Siemens SIMATIC Manager und Möglichkeit für Projekttransfer
- Modbus-Kabel-Verbindung

### Vorgehensweise

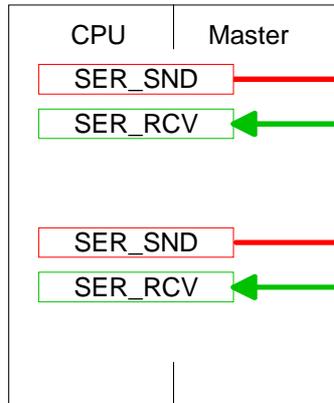
- Bauen Sie ein Modbus-System bestehend aus CPU 21xSER-1 als Modbus-Master und CPU 21xSER-1 als Modbus-Slave und Modbus-Kabel auf.
- Projektieren Sie die Master-Seite!  
Erstellen Sie hierzu ein SPS-Anwenderprogramm nach folgender Struktur:
  - OB 100: Aufruf SFC 216 (Konfiguration als Modbus RTU-Master) 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.
- 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.

Auf der Folgeseite ist die Struktur für die jeweiligen SPS-Programme für Master- und Slave-Seite dargestellt.

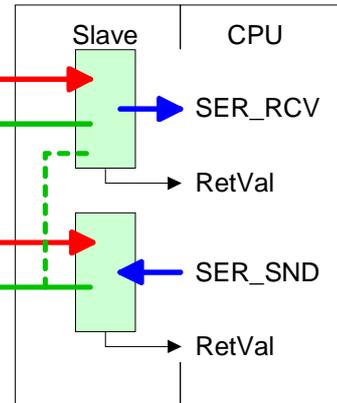
Master

Slave

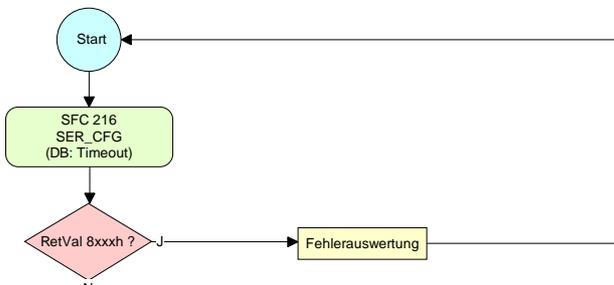
CPU 21xSER-1



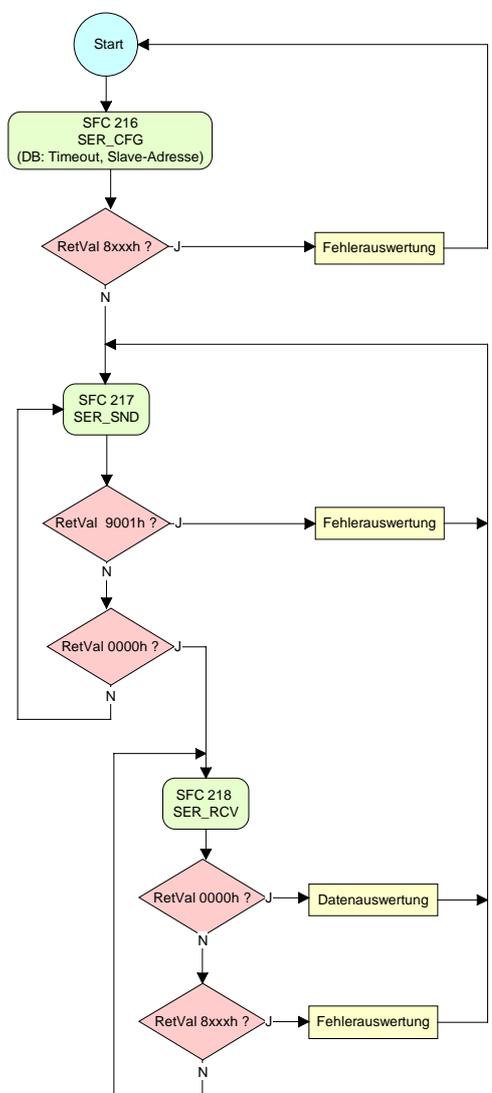
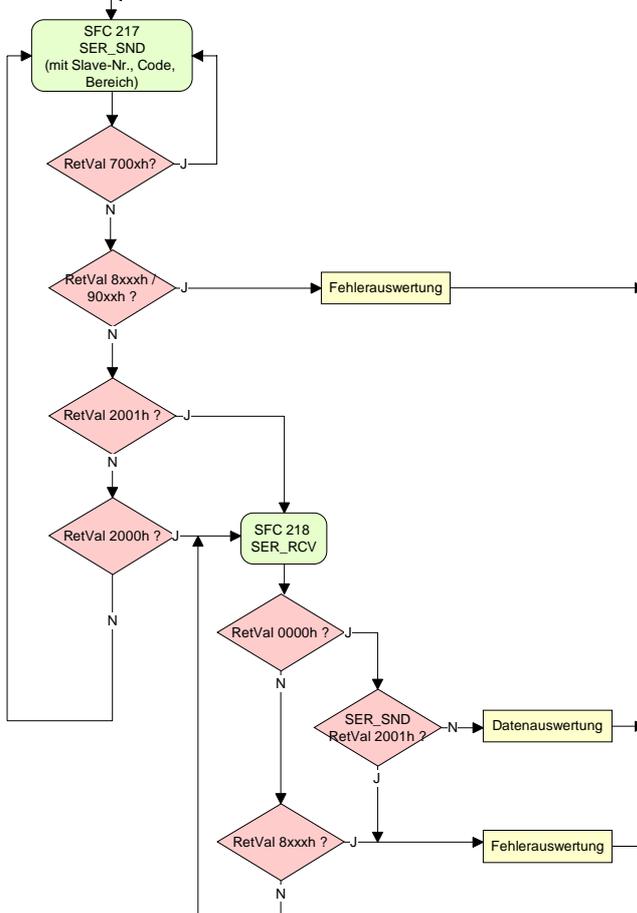
CPU 21xSER-1



OB100:



OB1:



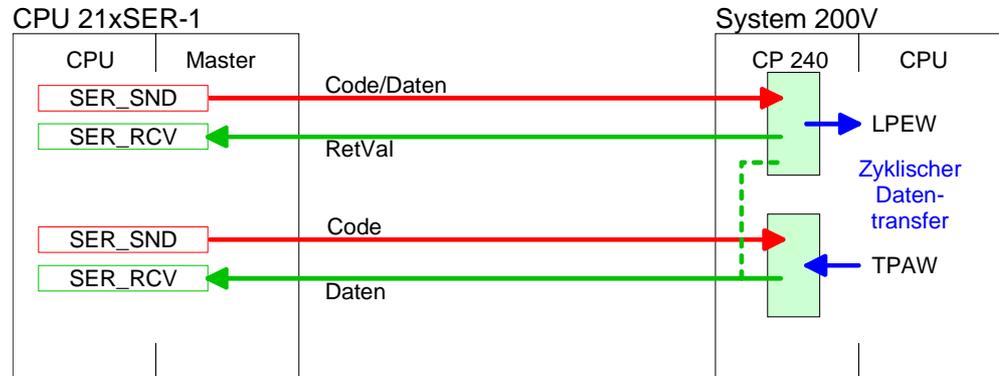
**M: CPU 21xSER-1**  
**S: CP 240**

Folgende Komponenten sind für das Beispiel erforderlich:

- 1 CPU 21xSER-1 als Modbus RTU-Master
- 1 System 200V mit CP 240 als Modbus RTU-Slave
- Siemens SIMATIC Manager und Möglichkeit für Projekttransfer
- Modbus-Kabel-Verbindung

**Vorgehensweise**

- Projektieren Sie die Master-Seite!  
Die Master-Seite ist auf die gleiche Weise zu projektieren, wie die der CPU 21xSER-1 (Struktur siehe oben).
- Projektieren Sie den CP 240-Slave!  
Die Parametrierung des CP 240 erfolgt über die Hardware-Konfiguration. Geben Sie hier für Ein- und Ausgabe-Bereich die Startadresse an, ab welcher die fixe Anzahl von 16Byte für Ein- und Ausgabe im Peripherie-Bereich der CPU abliegen.  
Für den Datentransfer über Modbus ist kein SPS-Programm erforderlich. Sie müssen lediglich dafür sorgen, dass Sie Daten, die vom Master empfangen wurden, in der CPU ausgewertet werden und Daten, die an den Master zu übertragen sind, im Ausgabe-Bereich stehen. Dies erreichen Sie, indem Sie das entsprechende Wort des Prozessabbilds zyklisch transferieren.  
Die nachfolgende Abbildung soll dies verdeutlichen:



**Ablauf einer Kommunikation**

Eine Kommunikation zwischen Master und Slave verläuft auf folgende Weise:

**Sende-Baustein des Masters als DB10**

Der Master sendet mit diesem Sende-DB 16Byte Nutzdaten an den Slave mit Adresse 5:

DB10.DBD 0	DW#16#05100000 mit 05 → 10 → 0000 →	<b>Kommandotelegramm</b> Slave-Adresse 05h Funktionscode 10h (write n words) Offset 0000h
DB10.DBD 4	DW#16#000810A0 mit 0008 → 10 → A0 →	<b>Kommandotelegramm + 1 Datenbyte</b> Wordcount 0008h Bytecount 10h Beginn 16Byte Daten mit A0h
DB10.DBD 8	DW#16#A1A2A3A4	<b>Daten Byte 2 ... 5</b>
DB10.DBD 12	DW#16#A5A6A7A8	<b>Daten Byte 6 ... 9</b>
DB10.DBD 16	DW#16#A9AAABAC	<b>Daten Byte 10 ... 13</b>
DB10.DBD 20	DW#16#ADAEEAF00 mit ADAEEAF → 00 →	<b>Daten Byte 14 ... 16 + 1 Byte nicht benützt</b> Daten Byte 14 ... 16 vom Modul nicht mehr belegt

**Empfangs-Baustein des Masters als DB11**

Sofern kein Fehler aufgetreten ist, sendet der Slave folgende Daten an den Master zurück:

DB11.DBD 0	DW#16#05100000 mit 05 → 10 → 0000 →	<b>Antworttelegramm</b> Slave-Adresse 05h Funktionscode 10h (kein Fehler) Offset 0000h
DB11.DBD 4	DW#16#000810A0 mit 0008 → 10 → 00 →	<b>Antworttelegramm + 1 Datenbyte</b> Wordcount 0008h Bytecount 10h Beginn 16Byte Daten mit 00h (bei Schreibbefehl irrelevant)
DB11.DBD 8	DW#16#00000000	<b>Daten Byte 2 ... 5</b>
DB11.DBD 12	DW#16#00000000	<b>Daten Byte 6 ... 9</b>
DB11.DBD 16	DW#16#00000000	<b>Daten Byte 10 ... 13</b>
DB11.DBD 20	DW#16#00000000 mit ADAEEAF → 00 →	<b>Daten Byte 14 ... 16 + 1 Byte nicht benützt</b> Daten Byte 14 ... 16 vom Modul nicht mehr belegt

**Empfangs-Baustein  
mit Fehlerrück-  
meldung**

Bei der Kommunikation unter Modbus gibt es 2 Fehlerarten:

- Slave antwortet nicht auf Kommando von Master

Antwortet der Slave nicht innerhalb der vorgegebenen Time-out-Zeit, generiert der Master im Empfangs-Baustein folgende Fehlermeldung: ERROR01 NO DATA.

In der Hex-Darstellung werden folgende Werte eingetragen:

DB11.DBD 0	<b>DW#16#452524F</b> mit 45 → 52 → 52 → 4F →	<b>Antworttelegramm</b> E R R O
DB11.DBD 4	<b>DW#16#52000120</b> mit 52 → 0001 → 20 →	<b>Antworttelegramm</b> R 0001h:1 (als Wort) " "
DB11.DBD 8	<b>DW#16#4E4F2044</b> mit 4E → 4F → 20 → 44 →	<b>Antworttelegramm</b> N O " " D
DB11.DBD 12	<b>DW#16#41544100</b> mit 41 → 54 → 41 → 00 →	<b>Antworttelegramm</b> A T A 00h: (Nullterminierung)

- Slave antwortet mit einer Fehlermeldung

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

DB11.DBD 0	<b>DW#16#05900000</b> mit 05 → 90 → 0000 →	<b>Antworttelegramm</b> Slave-Adresse 05h Funktionscode 90h (Fehlermeldung da 10h OR 80h = 90h) Die Restdaten sind irrelevant da Fehler rückgemeldet wurde.
------------	---	--



## Teil 11 Einsatz CPU 21xSER-2

### Überblick

Inhalt dieses Kapitels ist der Einsatz der CPU 21x-2BS02 mit zwei RS232-Schnittstellen.

Sie erhalten hier alle Informationen zum Einsatz, zu den Prozeduren und Übertragungsprotokollen der CPU 21x-2BS02.

### Inhalt

Thema	Seite
<b>Teil 11 Einsatz CPU 21xSER-2</b> .....	<b>11-1</b>
Schnelleinstieg .....	11-2
Protokolle und Prozeduren .....	11-3
RS232-Schnittstelle .....	11-7
Kommunikationsprinzip .....	11-8
Schnittstellen initialisieren .....	11-9
Schnittstellen-Parameter .....	11-11
Schnittstellenkommunikation .....	11-14

## Schnelleinstieg

**Allgemein** Die CPU 21x-2BS02 ermöglicht die serielle Prozessankopplung zu verschiedenen Ziel- oder Quellsystemen. Zur seriellen Kommunikation besitzt die CPU zwei RS232-Schnittstellen.  
Die Kommunikation erfolgt mittels Hantierungsbausteinen, die sich als Bibliothek in der CPU befinden.

**Protokolle** Unterstützt werden die Protokolle bzw. Prozeduren ASCII, STX/ETX, 3964(R) und RK512.

**Parametrierung** Die Parameterübertragung an den Kommunikationsprozessor (CP) erfolgt zur Laufzeit mit einem SEND (SFC 230) mit Auftrags-Nr. 201. Hierbei werden die Parameter in einem DB angelegt, dessen Aufbau sich nach dem gewünschten Protokoll richtet.  
Zur Aktivierung der Parameter ist nach dem SEND ein RESET (SFC 234) mit Auftrags-Nr. 0 auszuführen.



**Hinweis!**

Bitte beachten Sie, dass den Aufträgen SEND, RECEIVE, FETCH und RESET immer ein "VKE"=1 voranzustellen ist, da diese ansonsten nicht ausgeführt werden.

**Kommunikation** Der interne Kommunikationsprozessor der CPU 21x-2BS02 ist über ein Dual-Port-RAM, auch "Kachel" genannt, direkt mit dem CPU-Teil verbunden. Diese Kachel steht auf der CPU-Seite als Standard-CP-Interface zur Verfügung. Der Datenaustausch findet über die Standard Hantierungsbausteine (SEND, RECEIVE und FETCH) statt.  
Die Kommunikation über die entsprechenden Protokolle regeln Verbindungsaufträge, die im Anwenderprogramm zu programmieren sind.  
Hierbei kommen folgende SFCs zum Einsatz:

SFC	Bezeichnung	Beschreibung
SFC 230	Send	Send über Kachel (Kachelkommunikation)
SFC 231	Receive	Receive über Kachel (Kachelkommunikation)
SFC 232	Fetch	Fetch über Kachel (Kachelkommunikation)
SFC 233	Control	Control für Kachelkommunikation
SFC 234	Reset	Reset für Kachelkommunikation
SFC 235	Synchron	Synchron für Kachelkommunikation
SFC 236	Send_All	Send_All über Kachel (Kachelkommunikation)
SFC 237	Recv_All	Receive_All über Kachel (Kachelkommunikation)

## Protokolle und Prozeduren

### Übersicht

Die CPU 21x-2BS02 unterstützt folgende Protokolle und Prozeduren:

- ASCII-Übertragung
- STX/ETX
- 3964(R) mit RK512

### ASCII

Die Datenkommunikation via ASCII ist die einfachste Form der Kommunikation. Die Zeichen werden 1 zu 1 übergeben. Zur logischen Trennung der Telegramme ist es erforderlich, dass die **Zeichenverzugszeit** (ZVZ) des Empfängers vom Sender eingehalten wird. Hierbei ist die ZVZ in Millisekunden (ms) anzugeben und muss größer gleich 2ms sein.

Was beim Empfänger die Zeichenverzugszeit ist, ist beim Sender die "**Zeit nach Auftrag**" (ZNA). Mit diesen beiden Zeitangaben können Sie eine einfach serielle SPS-Kommunikation aufbauen. Ein Sendeauftrag wird erst dann mit "**Auftrag ohne Fehler**" (AFOF) gekennzeichnet, wenn die Daten gesendet wurden. Ein neuer Auftrag wird erst dann gestartet, wenn die ZNA abgelaufen ist. Wird ZNA auf 0 gesetzt, ist die Sendefolge über das Anwenderprogramm zu steuern.

### STX/ETX

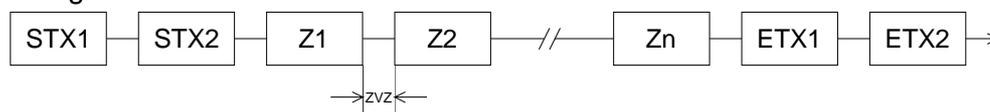
STX/ETX ist ein einfaches Protokoll mit Start- und Ende-Kennung. Hierbei stehen STX für **Start of Text** und ETX für **End of Text**.

Die Prozedur STX/ETX wird zur Übertragung von ASCII-Zeichen (20h...7Fh) 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.

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 die CPU 21x-2BS02 übergeben und von dort mit angefügten Start- und Endezeichen über die serielle Schnittstelle an den Kommunikationspartner übertragen.

Telegrammaufbau:



Sie können bis zu 2 Anfangs- und Endezeichen frei definieren.

Es kann mit 1, 2 oder keiner Start- und mit 1, 2 oder keiner Ende-Kennung gearbeitet werden. Als Start- bzw. Ende-Kennung sind alle Hex-Werte von 01h 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). Wird kein Ende-Zeichen definiert, so werden alle gelesenen Zeichen nach Ablauf einer parametrierbaren Zeichenverzugszeit (Timeout) an die CPU übergeben.

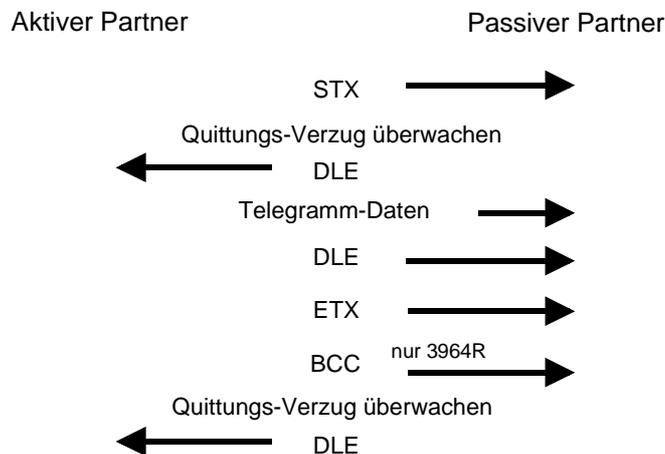
**3964(R)**

Die Prozedur 3964(R) steuert die Datenübertragung bei einer Punkt-zu-Punkt-Kopplung zwischen der CPU 21x-2BS02 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 (nur bei 3964R)
- NAK           **N**egative **A**cknowledge

**Prozedurablauf**



Sie können pro Telegramm maximal 255Byte übertragen.



**Hinweis!**

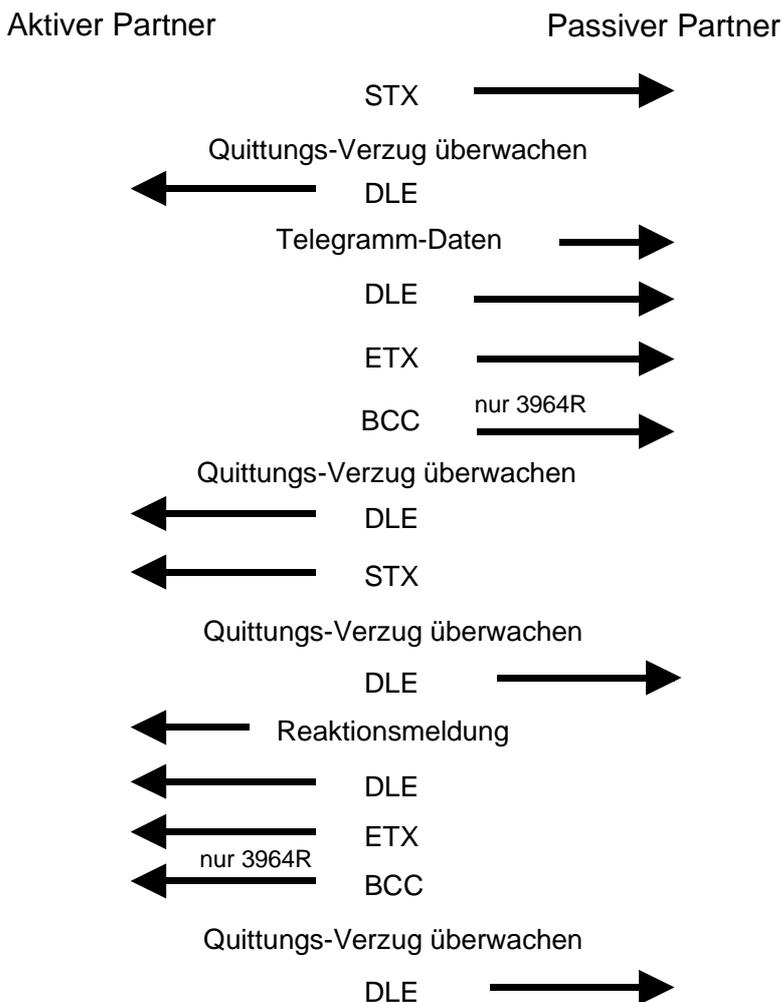
Wird ein "DLE" als Informationszeichen übertragen, so wird dieses zur Unterscheidung vom Steuerzeichen "DLE" beim Verbindungsauf- und -abbau auf der Sendeleitung doppelt gesendet (DLE-Verdoppelung).

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.

**3964(R) mit RK512** Das RK512 ist ein erweitertes 3964(R). Es wird lediglich vor der Übertragung der Nutzdaten ein Telegrammkopf gesendet. Der Telegrammkopf enthält für den Kommunikationspartner Infos über Größe, Art und Länge der Nutzdaten.

**Prozedurablauf** Im folgenden wird der Aufbau der Prozedur und der Telegramme beschrieben:



**Koordinierungsmerker** Der Koordinierungsmerker wird im Aktiv-Betrieb im Partner-AG bei Empfang eines Telegramms gesetzt. Dies geschieht sowohl bei Eingabe- als auch bei Ausgabe-Befehlen. Ist der Koordinierungsmerker gesetzt und wird ein Telegramm mit diesem Merker empfangen, so werden die Daten nicht übernommen (bzw. übergeben), sondern es wird eine Fehler-Reaktionsmeldung gesendet (Fehlercode 32h). In diesem Fall muss der Koordinierungsmerker vom Anwender im Partner-AG zurückgesetzt werden.

Soll ein Telegramm keinen Koordinierungsmerker enthalten, muss FFFFh angegeben werden.

**Time-Out-Zeiten**

Als Time-Out gelten folgende Zeiten :

Quittungs-Verzugs-Zeit: (QVZ) = 2000 ms

Zeichen-Verzugs-Zeit: (ZVZ) = 220 ms

QVZ wird überwacht zwischen STX und DLE sowie zwischen BCC und DLE. ZVZ wird während des gesamten Telegramm-Empfangs überwacht.

Bei Verstreichen der QVZ nach STX wird erneut STX gesendet, nach 3\*) Versuchen wird ein NAK gesendet und der Verbindungsaufbau abgebrochen. Dasselbe geschieht, wenn nach einem STX ein NAK oder ein beliebiges Zeichen empfangen wird.

Bei Verstreichen der QVZ nach dem Telegramm (nach BCC-Byte) oder bei Empfang eines Zeichens ungleich DLE werden der Verbindungsaufbau und das Telegramm wiederholt. Auch hier werden 3\*) Versuche unternommen, danach ein NAK gesendet und die Übertragung abgebrochen.

\*) einstellbar als Parameter

**Passivbetrieb**

Wenn der Prozedurtreiber auf den Verbindungsaufbau wartet und ein Zeichen ungleich STX empfängt, sendet er NAK. Bei Empfang eines Zeichens NAK sendet der Prozedurtreiber keine Antwort.

Wird beim Empfang die ZVZ überschritten, wird ein NAK gesendet und auf erneuten Verbindungsaufbau gewartet.

Wenn der Prozedurtreiber beim Empfang des STX noch nicht bereit ist, sendet er ein NAK.

**Block-Check-Character (BCC-Byte)**

Zur weiteren Datensicherung wird bei der Prozedur 3964R am Ende des Telegramms ein BlockCheck-Character angehängt. Das BCC-Byte wird durch eine XOR-Verknüpfung über die Daten des gesamten Telegramms einschließlich DLE/ETX gebildet.

Beim Empfang eines BCC-Bytes, das vom selbst ermittelten abweicht, wird anstatt des DLEs ein NAK gesendet.

**Initialisierungs-konflikt**

Versuchen beide Partner gleichzeitig innerhalb der QVZ einen Verbindungsaufbau, so sendet der Partner mit der niedrigeren Priorität das DLE und geht auf Empfang.

**DLE**

Das DLE-Zeichen in einem Telegramm wird vom Prozedurtreiber verdoppelt, d.h. es wird DLE/DLE gesendet. Beim Empfang werden doppelte DLEs als ein DLE im Puffer abgelegt. Als Ende des Telegramms gilt immer die Kombination DLE/ETX/BCC (nur bei 3964R).

Die Steuercodes :           02h = STX

                                  03h = ETX

                                  10h = DLE

                                  15h = NAK

## RS232-Schnittstelle

### Eigenschaften

- Schnittstelle kompatibel zur COM Schnittstelle eines PCs
- Unterstützt werden die Protokolle ASCII, STX/ETX, 3964(R) und RK512
- Empfangs- und Sendepuffer haben jeweils eine Größe von 1024Byte
- Die maximale Telegrammlänge beträgt 1024Byte

### Merkmale

#### RS232-Schnittstelle

- Logische Zustände als Spannungspegel
- Punkt-zu-Punkt-Kopplung mit serieller Vollduplex-Übertragung (nur bei RS232) in 3-Draht-Technik bis zu einer Entfernung von 15m
- Datenübertragungsrate bis 57,6kBaud

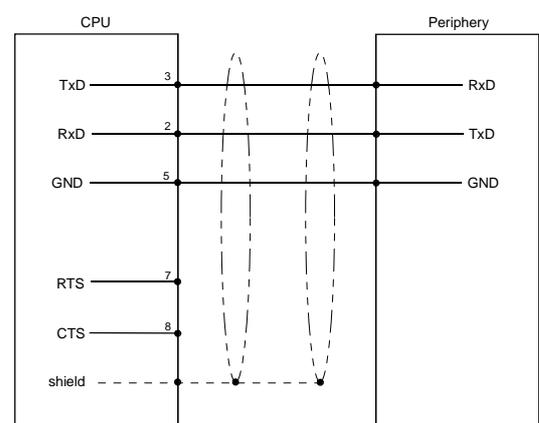
### Anschluss Schnittstelle

Über den 9poligen Stecker können Sie eine serielle Punkt-zu-Punkt-Verbindung herstellen.

#### 9poliger Stecker

Pin	Belegung
1	CD-
2	RxD
3	TxD
4	DTR-
5	GND
6	DSR-
7	RTS-
8	CTS-
9	RI-

#### Anschluss RS232



Folgende RS232-Signale werden von der CPU 21x-2BS02 unterstützt:

#### TxD Transmit Data

Die Sendedaten werden über die TxD-Leitung übertragen. Die Sendeleitung wird von der CPU 21x-2BS02 im Ruhezustand auf logisch "1" gehalten.

#### RxD Receive Data

Die Empfangsdaten kommen über die RxD-Leitung an. Die Empfangsleitung muss vom Sendepartner im Ruhezustand auf logisch "1" gehalten werden.

## Kommunikationsprinzip

### Übersicht

Die Kommunikation wird mittels der integrierten Hantierungsbausteine (siehe unten) abgewickelt.

Die CPU entscheidet aufgrund der Parametrierung, wie die Datenübergabe stattfindet. In den Standard-Modi wird durch die SEND/RECEIVE-Bausteine der Auftrags-Anstoß und durch die "ALL"-Bausteine die Nutzdatenkommunikation durchgeführt.

Bei der Kommunikation kommen folgende Bausteine zum Einsatz:

Nr.	Bezeichnung	
SFC 235	SYNCHRON	Synchronisation zwischen CPU und CP und Vorgabe der Blockgröße
SFC 230	SEND	Auslösen eines Sendeauftrags
SFC 236	SEND-ALL	Nutzdaten senden
SFC 231	RECEIVE	Auslösen eines Empfang-Auftrags
SFC 237	RECEIVE-ALL	Nutzdaten empfangen
SFC 232	FETCH	Auslösen eines Holauftrages
SFC 233	CONTROL	Baustein zur Kommunikationskontrolle
SFC 234	RESET	Löscht alle Aufträge und aktiviert neue Parameter

### Programmierung

In kurzen Schritten soll gezeigt werden, wie Sie bei der Programmierung vorzugehen haben:

#### *Anlauf-OB100:*

- Rufen Sie mit SFC 235 SYNCHRON auf und geben Sie die gewünschte Blockgröße an (Kachelbasisadresse=0, Blockgröße, PAFE)
- Parametrieren Sie die Schnittstellen mit SEND (SFC 230) mit Auftrags-Nr. 201 und Parameter-DB
- Zur Übernahme der Parameter rufen Sie RESET (SFC 234) mit Auftrags-Nr. 0 auf

#### *Zyklus-OB1:*

- SEND- und RECEIVE-Aufträge für Sende- und Empfangsanstoß generieren
- SEND ALL und RECEIVE ALL-Aufträge für Nutzdatenübertragung generieren

Nachfolgend finden Sie hierzu eine nähere Beschreibung.

## Schnittstellen initialisieren

### Übersicht

Die Initialisierung der Schnittstellen findet im OB 100 statt und sollte nach folgender Vorgehensweise durchgeführt werden:

- Rufen Sie mit SFC 235 SYNCHRON auf und geben Sie die gewünschte Blockgröße an (Kachelbasisadresse=0, Blockgröße, PAFE).
- Parametrieren Sie die Schnittstellen mit SEND (SFC 230) mit Auftrags-Nr. 201 und Parameter-DB.
- Zur Übernahme der Parameter rufen Sie RESET (SFC 234) mit Auftrags-Nr. 0 auf.

### SFC 235 SYNCHRON

Der Baustein stellt im CPU-Anlauf die Synchronisation zwischen CPU und CP her und ist daher im Anlauf-OB OB 100 aufzurufen. Gleichzeitig wird der Übergabebereich der Schnittstelle gelöscht und voreingestellt, sowie die Blockgröße zwischen CP und CPU ausgehandelt.

#### Parameter

Name	Deklaration	Typ	Beschreibung
SSNR	IN	INT	Interface number, number of logical interface
BLGR	IN	INT	Blocksize
PAFE	OUT	BYTE	Error indicator for configuration errors

#### SSNR

Nummer der logischen Schnittstelle (Kacheladresse) auf die sich der betreffende Auftrag bezieht. SSNR muss 0 sein!

#### Blockgröße

Zur Vermeidung von langen Zykluszeiten ist es sinnvoll große Datenmengen in kleinen Blöcken zwischen CPU und CP zu übertragen. Die Größe dieser Blöcke stellen Sie über die Blockgröße ein.

Hierbei bedeutet große Blockgröße = hoher Datendurchsatz aber auch lange Laufzeit und damit hohe Zykluszeitbelastung.

Kleine Blockgröße = kleiner Datendurchsatz aber auch kleine Laufzeiten der Bausteine.

Als Blockgröße haben Sie folgende Einstellmöglichkeiten:

Wert	Blockgröße	Wert	Blockgröße
0	Default (64Byte)	4	128Byte
1	16Byte	5	256Byte
2	32Byte	6	512Byte
3	64Byte	255	512Byte

Parameterart : Integer

Möglicher Bereich : 0 ... 255

#### Beispiel

```
CALL SFC 235
SSNR:=0
BLGR:=6
PAFE:=MB199
```

**SFC 230 - SEND mit ANR=201 und Parameter-DB**

Mit einem SEND (SFC 230), ANR=201 und DB können Sie Parameter an den CP übergeben.

Bitte beachten Sie, dass ein Sende-Auftrag nur dann ausgeführt wird, wenn folgende Bedingungen erfüllt sind:

- dem SEND wurde ein VKE "1" übergeben
- im Anzeigenwort ist das Bit "Auftrag läuft" zurückgesetzt

**Parameter**

Name	Deklaration	Typ	Beschreibung
SSNR	IN	INT	Interface number, number of logical interface
ANR	IN	INT	Job must be initiated at the interface, start trans.
IND	IN	INT	Mode of addressing (direct/indirect)
QANF	IN	ANY	Pointer to data source
PAFE	OUT	BYTE	Error indicator for configuration errors
ANZW	IN_OUT	DWORD	Indicatorword (progress of started jobs displayed)

**SFC 234 - RESET**

Mit einem RESET mit Auftragsnummer 0 werden alle Aufträge abgebrochen und die zuvor geladenen Parameter aktiviert.

Wie bei SEND müssen Sie diesem Baustein ein VKE=1 vorschalten.

**Parameter**

Name	Deklaration	Typ	Beschreibung
SSNR	IN	INT	Interface number, number of logical interface
ANR	IN	INT	Job must be initiated at the interface, start trans.
PAFE	OUT	BYTE	Error indicator for configuration errors

**Beispiel OB100**

```

O      M      0.0
ON     M      0.0          VKE=1
CALL  SFC    230          SEND
SSNR:=0
ANR  :=201                Kennung für Parametrierung
IND  :=0
QANF:=P#DB9.DBX0.0 BYTE 20  Pointer auf Parameter für COM1
PAFE:=MB198
ANZW:=MD200

O      M      0.0
ON     M      0.0          VKE=1
CALL  SFC    230          SEND
SSNR:=0
ANR  :=201                Kennung für Parametrierung
IND  :=0
QANF:=P#DB9.DBX20.0 BYTE 20  Pointer auf Parameter für COM2
PAFE:=MB198
ANZW:=MD200

O      M      0.0
ON     M      0.0          VKE=1
CALL  SFC    234          RESET
SSNR:=0
ANR  :=0                  Kennung für RESET
PAFE:=MB197
    
```

## Schnittstellen-Parameter

**Aufbau** Die Parameterübertragung an den Kommunikationsprozessor erfolgt zur Laufzeit unter Einsatz des SFC 230 mit Auftrags-Nr. 201. Hierbei sind die Parameter für die Protokolle in einem DB abzulegen.

**Parameter-DB** Zur Aktivierung der Parameter ist nach der Übertragung mit SFC 234 ein RESET auszuführen.

*Allgemein für jeden benutzten Kanal:*

Datenbyte	Typ	Bezeichner	Werte	Default	
0	BYTE	Kanal	COM 1 COM 2	1 2	
1	BYTE	Modi	MODI_NONE (deaktiviert) MODI_1 (Parameter nachfolgend)	0 81h	0
2	BYTE	Baudrate	BAUDRATE_DEF BAUDRATE_150 BAUDRATE_300 BAUDRATE_600 BAUDRATE_1K2 BAUDRATE_1K8 BAUDRATE_2K4 BAUDRATE_4K8 BAUDRATE_7K2 BAUDRATE_9K6 BAUDRATE_14K4 BAUDRATE_19K2 BAUDRATE_38K4 BAUDRATE_57K6	00h 01h 02h 03h 04h 05h 06h 07h 08h 09h 0Ah 0Bh 0Ch 0Dh	09h
3	BYTE	DataBits	DATABIT_5 DATABIT_6 DATABIT_7 DATABIT_8	0 1 2 3	3
4	BYTE	Parity	PARITY_NONE PARITY_ODD PARITY_EVEN	0 1 3	0
5	BYTE	StopBits	STOPBIT_1 STOPBIT_1_5 STOPBIT_2	1 2 3	1
6	BYTE	FlowControl	FLOW_NONE FLOW_HARDWARE FLOW_XON_XOFF	0 1 2	1
7	BYTE	Protocol	PROTOCOL_ASCII PROTOCOL_STXETX_HTB PROTOCOL_3964 PROTOCOL_3964R PROTOCOL_3964_RK512 PROTOCOL_3964R_RK512	01h 02h 03h 04h 05h 06h	01h

Zusätzliche protokollabhängige Parameter      Abhängig von dem eingestellten Protokoll sind noch folgende Parameter im DB anzugeben:

wenn *PROTOCOL\_ASCII*:

Datenbyte	Typ	Bezeichner	Werte	Default
Sendekanal				
8, 9	WORD	BufAnz	1..n	1
10, 11	WORD	BufSize	16..1024	256
12, 13	WORD	ZNA, Zeit nach Auftrag	0..n	500
Empfangskanal				
14, 15	WORD	BufAnz	1..n	1
16, 17	WORD	BufSize	16..1024	256
18, 19	WORD	ZVZ, Zeichenverzugszeit	2..n	200

wenn *PROTOCOL\_STXETX*:

Datenbyte	Typ	Bezeichner	Werte	Defaults
Sendekanal				
8, 9	WORD	BufAnz	1..n	1
10, 11	WORD	BufSize	16..1024	256
12, 13	WORD	ZNA, Zeit nach Auftrag	0..n	0
Startkennung				
14, 15	WORD	Anzahl	1, 2	1
16	BYTE	Kennung1	0..255	STX
17	BYTE	Kennung2	0..255	STX
Endekennung				
18, 19	WORD	Anzahl	1, 2	1
20	BYTE	Kennung1	0..255	ETX
21	BYTE	Kennung2	0..255	ETX
Empfangskanal				
22, 23	WORD	BufAnz	1..n	1
24, 25	WORD	BufSize	16..1024	256
26, 27	WORD	TMO, Timeout	2..n	200
Startkennung				
28, 29	WORD	Anzahl	1, 2	1
30	BYTE	Kennung1	0..255	STX
31	BYTE	Kennung2	0..255	STX
Endekennung				
32, 33	WORD	Anzahl	1, 2	1
34	BYTE	Kennung1	0..255	ETX
35	BYTE	Kennung2	0..255	ETX

wenn *PROTOCOL\_3964(R)*:

Datenbyte	Typ	Bezeichner	Werte	Defaults
Sende-/ Empfangskanal				
8, 9	WORD	BufAnz	1..n	1
10, 11	WORD	BufSize	16..1024	128
12, 13	WORD	ZNA, Zeit nach Auftrag	0..n	0
14, 15	WORD	ZVZ Zeichenverzugszeit	1..n	200
16, 17	WORD	QVZ Quittungsverzugszeit	1..n	500
18, 19	WORD	BWZ Blockwartezeit	1..n	10000
20, 21	WORD	STX Anzahl Wiederholungen Verbindungsaufbau	1..n	3
22, 23	WORD	DBL Anzahl Wiederholungen Datenblöcke	1..n	6
24, 25	WORD	Priorität 0==Low, >0==High	0, 1	1

wenn *PROTOCOL\_3964(R)\_RK512*:

Datenbyte	Typ	Bezeichner	Werte	Defaults
Sende-/ Empfangskanal				
8, 9	WORD	BufAnz	1..n	1
10, 11	WORD	BufSize	16..1024	128
12, 13	WORD	ZNA, Zeit nach Auftrag	0..n	0
14, 15	WORD	ZVZ Zeichenverzugszeit	1..n	200
16, 17	WORD	QVZ Quittungsverzugszeit	1..n	500
18, 19	WORD	BWZ Blockwartezeit	1..n	10000
20, 21	WORD	STX Anzahl Wiederholungen Verbindungsaufbau	1..n	3
22, 23	WORD	DBL Anzahl Wiederholungen Datenblöcke	1..n	6
24, 25	WORD	Priorität 0==Low, >0==High	0, 1	1
26, 27	WORD	QVZ für Anwenderquittung	1..n	5000

### Fest definierte Parameter

Folgende Parameter sind fest definiert und können nicht geändert werden:

Parameter	Einstellung
Kachelbasisadresse	0
Kachelanzahl	1
Auftragsnummer	1: SEND für COM 1 2: RECEIVE für COM 1 3: SEND für COM 2 4: RECEIVE für COM 2 201: für SEND zur Parametrierung
Auftragspriorität	2

## Schnittstellenkommunikation

### Übersicht

Die Kommunikation erfolgt über folgende Hantierungsbausteine im OB1:

Nr.	Bezeichnung	
SFC 230	SEND	Auslösen eines Sendeauftrags
SFC 236	SEND-ALL	Nutzdaten senden
SFC 231	RECEIVE	Auslösen eines Empfang-Auftrags
SFC 237	RECEIVE-ALL	Nutzdaten empfangen
SFC 232	FETCH	Auslösen eines Holauftrages
SFC 233	CONTROL	Baustein zur Kommunikationskontrolle

#### Zyklus-OB1:

- SEND- und RECEIVE-Aufträge für Sende- und Empfangsanstoß generieren
- SEND ALL und RECEIVE ALL-Aufträge für Nutzdatenübertragung generieren

Nachfolgend finden Sie eine Zusammenfassung dieser Bausteine.

### SFC 230 - SEND

Der SEND-Baustein dient zum Auslösen eines Sende-Auftrags zu einem CP.

Bitte beachten Sie, dass ein Sende-Auftrag nur dann ausgeführt wird, wenn folgende Bedingungen erfüllt sind:

- dem SEND wurde ein VKE "1" übergeben
- im Anzeigenwort ist das Bit "Auftrag läuft" zurückgesetzt

### Parameter

Name	Deklaration	Typ	Beschreibung
SSNR	IN	INT	Interface number, number of logical interface
ANR	IN	INT	Job must be initiated at the interface, start trans.
IND	IN	INT	Mode of addressing (direct/indirect)
QANF	IN	ANY	Pointer to data source
PAFE	OUT	BYTE	Error indicator for configuration errors
ANZW	IN_OUT	DWORD	Indicatorword (progress of started jobs displayed)

**SFC 236 -  
SEND\_ALL**

Kann der CP die Daten direkt übernehmen, überträgt der SEND-Baustein die angeforderten Daten in einem Zug zum CP. Signalisiert der CP jedoch, dass er nur die Parameter des Auftrages wünscht oder ist die Anzahl der zu übergebenden Daten zu groß, werden dem CP nur die Sende-Parameter bzw. die Parameter mit dem ersten Datenblock übergeben. Die Daten oder der Folgeblock zu diesen Aufträgen fordert der CP über SEND\_ALL bei der CPU an. Hierzu ist es jedoch erforderlich, dass mindestens einmal im Zyklus der Baustein SEND\_ALL aufgerufen wird.

**SFC 231 - RECEIVE**

Der RECEIVE-Baustein dient zum Empfangen von Daten von einem CP. Der RECEIVE-Baustein ist im zyklischen Teil des Anwenderprogramms aufzurufen.

Bitte beachten Sie, dass ein Empfangs-Auftrag nur dann ausgeführt wird, wenn folgende Bedingungen erfüllt sind:

- dem RECEIVE wurde ein VKE "1" übergeben
- der CP hat den Auftrag freigegeben (Bit "Handshake sinnvoll" = 1)

**Parameter**

Name	Deklaration	Typ	Beschreibung
SSNR	IN	INT	Interface number, number of logical interface
ANR	IN	INT	Job must be initiated at the interface, start trans.
IND	IN	INT	Mode of addressing (direct/indirect)
ZANF	IN	ANY	Pointer to data destination
PAFE	OUT	BYTE	Error indicator for configuration errors
ANZW	IN_OUT	DWORD	Indicatorword (progress of started jobs displayed)

**SFC 237 -  
RECEIVE\_ALL**

Mit dem RECEIVE\_ALL-Baustein werden die Daten, die vom CP empfangen werden, vom CP an die CPU unter Verwendung der eingestellten Blockgröße übermittelt.

Die Lage und Größe des Datenbereichs, der mit RECEIVE\_ALL zu übermitteln ist, muss zuvor über einen RECEIVE-Aufruf definiert werden.

Im Anzeigenwort, das dem zu bearbeitenden Auftrag zugeordnet ist, werden die Bits "Enable/Disable", "Datenübernahme erfolgt" sowie "Datenübernahme-/übergabe läuft" ausgewertet oder beeinflusst und im Folgewort die "Empfangslänge" angezeigt.

**Parameter**

Name	Declaration	Type	Comment
SSNR	IN	INT	Interface number, number of logical interface
PAFE	OUT	BYTE	Error indicator for configuration errors
ANZW	IN_OUT	DWORD	Indicatorword (progress of started jobs displayed)

**Hinweis!**

In folgenden Fällen ist mindestens einmal RECEIVE\_ALL im Zyklus-Baustein OB1 aufzurufen:

- wenn der CP selbständig Daten an die CPU senden soll.
- wenn ein CP-Auftrag mit RECEIVE angestoßen wird, der CP die Daten zu diesem Auftrag jedoch erst über die "Hintergrundkommunikation" an die CPU weitergeben kann.
- wenn die Anzahl der Daten, die mit einem RECEIVE an die CPU übergeben werden sollen, größer als die eingestellte Blockgröße ist.

**SFC 232 - FETCH**

Der FETCH-Baustein dient dem Auslösen eines "Holauftrags" auf einer Gegenstation.

Mit dem FETCH-Auftrag werden Daten-Quelle und -Ziel definiert und die Datenquelle an die Gegenstation übertragen.

Bei der CPU von VIPA erfolgt die Angabe von Quelle und Ziel über einen Zeiger-Parameter.

Die Gegenstation stellt die Daten aus der *Quelle* bereit und schickt diese über SEND\_ALL an die anfordernde Station zurück. Über RECEIVE\_ALL werden die Daten empfangen und in *Ziel* abgelegt.

Die Aktualisierung des Anzeigenworts erfolgt über FETCH bzw. CONTROL.

Bitte beachten Sie, dass ein Hol-Auftrag nur dann ausgeführt wird, wenn folgende Bedingungen erfüllt sind:

- dem FETCH wurde ein VKE "1" übergeben
- im Anzeigenwort ist das Bit "Auftrag läuft" zurückgesetzt

**Parameter**

Name	Deklaration	Typ	Beschreibung
SSNR	IN	INT	Interface number, number of logical interface
ANR	IN	INT	Job must be initiated at the interface, start trans.
IND	IN	INT	Mode of addressing (direct/indirect)
ZANF	IN	ANY	Pointer to data destination
PAFE	OUT	BYTE	Error indicator for configuration errors
ANZW	IN_OUT	DWORD	Indicatorword (progress of started jobs displayed)

**SFC 233 - CONTROL**

Der CONTROL-Baustein hat folgende Aufgaben:

- Aktualisierung des Anzeigenworts
- Abfrage, ob ein bestimmter Auftrag des CP zur Zeit "tätig" ist, z.B. Nachfrage nach einem Empfangstelegramm
- Abfrage des CP, welcher Auftrag z.Z. bearbeitet wird

Der Baustein ist nicht VKE abhängig und sollte im zyklischen Teil des Programms aufgerufen werden.

**Parameter**

Name	Deklaration	Typ	Beschreibung
SSNR	IN	INT	Interface number, number of logical interface
ANR	IN	INT	Job must be initiated at the interface, start trans.
PAFE	OUT	BYTE	Error indicator for configuration errors
ANZW	IN_OUT	DWORD	Indicatorword (progress of started jobs displayed)