

VIPA System SLIO

CP | 040-1CA00 | Handbuch

HB300 | CP | 040-1CA00 | de | 18-28

CP 040 - RS422/485



VIPA GmbH
Ohmstr. 4
91074 Herzogenaurach
Telefon: 09132-744-0
Telefax: 09132-744-1864
E-Mail: info@vipa.com
Internet: www.vipa.com

Inhaltsverzeichnis

1	Allgemein	5
	1.1 Copyright © VIPA GmbH	5
	1.2 Über dieses Handbuch.....	6
	1.3 Sicherheitshinweise.....	7
2	Grundlagen und Montage	8
	2.1 Sicherheitshinweis für den Benutzer.....	8
	2.2 Systemvorstellung.....	9
	2.2.1 Übersicht.....	9
	2.2.2 Komponenten.....	10
	2.2.3 Zubehör.....	12
	2.3 Abmessungen.....	13
	2.4 Montage Peripherie-Module.....	16
	2.5 Verdrahtung Peripherie-Module.....	19
	2.6 Verdrahtung Power-Module.....	21
	2.7 Demontage Peripherie-Module.....	25
	2.8 Hilfe zur Fehlersuche - LEDs.....	29
	2.9 Aufbaurichtlinien.....	30
	2.10 Allgemeine Daten.....	32
3	Hardwarebeschreibung	34
	3.1 Leistungsmerkmale.....	34
	3.2 Aufbau.....	35
	3.3 Technische Daten.....	39
	3.3.1 Technische Daten Protokolle.....	40
4	Einsatz	42
	4.1 Schnelleinstieg.....	42
	4.2 Ein-/Ausgabe-Bereich.....	43
	4.3 Prinzip der Rückwandbus-Kommunikation.....	44
	4.3.1 Daten senden.....	44
	4.3.2 Daten empfangen.....	47
	4.3.3 Beispiele.....	50
	4.4 Kommunikation über Hantierungsbausteine.....	53
	4.4.1 Übersicht.....	53
	4.4.2 VIPA Lib.....	54
	4.5 Diagnosedaten.....	55
5	Serielle Kommunikationsprotokolle	58
	5.1 Übersicht.....	58
	5.2 ASCII.....	59
	5.2.1 Grundlagen ASCII.....	59
	5.2.2 Parametrierdaten bei ASCII.....	59
	5.3 STX/ETX.....	64
	5.3.1 Grundlagen STX/ETX.....	64
	5.3.2 Parametrierdaten bei STX/ETX	64
	5.4 3964(R).....	69
	5.4.1 Grundlagen 3964(R).....	69
	5.4.2 Parametrierdaten bei 3964(R).....	70
	5.5 Modbus.....	74
	5.5.1 Grundlagen Modbus.....	74

5.5.2 Modbus auf dem CP von VIPA.....	75
5.5.3 Parametrierdaten bei Modbus.....	76
5.6 Einsatz - Modbus.....	80
5.6.1 Modbus - Übersicht	80
5.6.2 Modbus - Zugriff auf mehrere Slaves.....	82
5.6.3 Modbus - Funktionscodes	83
5.6.4 Modbus - Fehlermeldungen	88

1 Allgemein

1.1 Copyright © VIPA GmbH

All Rights Reserved

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

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

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

Tel.: +49 9132 744 -0

Fax.: +49 9132 744-1864

E-Mail: info@vipa.de

<http://www.vipa.com>



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

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

EG-Konformitätserklärung

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

Informationen zur Konformitätserklärung

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

Warenzeichen

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

SPEED7 ist ein eingetragenes Warenzeichen der profichip GmbH.

SIMATIC, STEP, SINEC, TIA Portal, S7-300, S7-400 und S7-1500 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.

Über dieses Handbuch

- 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
 EMail: 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)
 EMail: support@vipa.de

1.2 Über dieses Handbuch

- Zielsetzung und Inhalt** Das Handbuch beschreibt den CP 040-1CA00 aus dem System SLIO von VIPA. Beschrieben wird Aufbau, Projektierung und Anwendung.

Produkt	Best.-Nr.	ab Stand:	
		HW	FW
CP 040 RS422/485	040-1CA00	01	V1.0.1

- Zielgruppe** Das Handbuch ist geschrieben für Anwender mit Grundkenntnissen in der Automatisierungstechnik.
- Aufbau des Handbuchs** Das Handbuch ist in Kapitel gegliedert. Jedes Kapitel beschreibt eine abgeschlossene Thematik.
- Orientierung im Dokument** Als Orientierungshilfe stehen im Handbuch zur Verfügung:
- Gesamt-Inhaltsverzeichnis am Anfang des Handbuchs
 - Verweise mit Seitenangabe
- Verfügbarkeit** Das Handbuch ist verfügbar in:
- gedruckter Form auf Papier
 - in elektronischer Form als PDF-Datei (Adobe Acrobat Reader)
- Piktogramme Signalwörter** Besonders wichtige Textteile sind mit folgenden Piktogrammen und Signalworten ausgezeichnet:



GEFAHR!

Unmittelbar drohende oder mögliche Gefahr. Personenschäden sind möglich.

**VORSICHT!**

Bei Nichtbefolgen sind Sachschäden möglich.



Zusätzliche Informationen und nützliche Tipps.

1.3 Sicherheitshinweise

Bestimmungsgemäße Verwendung

Das System ist konstruiert und gefertigt für:

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

**GEFAHR!**

Das Gerät ist nicht zugelassen für den Einsatz

- in explosionsgefährdeten Umgebungen (EX-Zone)

Dokumentation

Handbuch zugänglich machen für alle Mitarbeiter in

- Projektierung
- Installation
- Inbetriebnahme
- Betrieb

**VORSICHT!**

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

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

Entsorgung

Zur Entsorgung des Geräts nationale Vorschriften beachten!

2 Grundlagen und Montage

2.1 Sicherheitshinweis für den Benutzer

Handhabung elektrostatisch gefährdeter Baugruppen

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



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

Versenden von Baugruppen

Verwenden Sie für den Versand immer die Originalverpackung.

Messen und Ändern von elektrostatisch gefährdeten Baugruppen

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

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

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



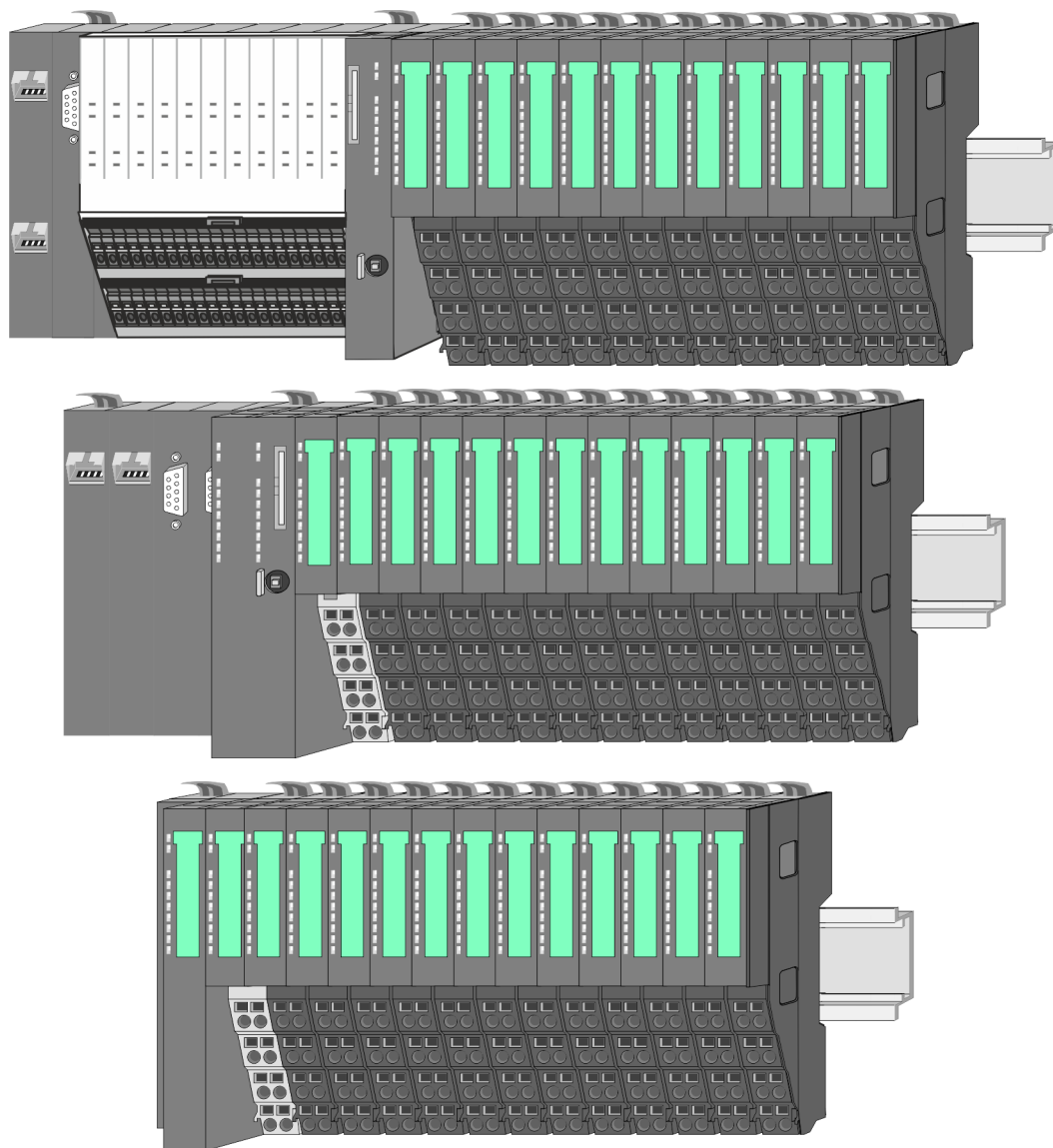
VORSICHT!

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

2.2 Systemvorstellung

2.2.1 Übersicht

Das System SLIO ist ein modular aufgebautes Automatisierungssystem für die Montage auf einer 35mm Tragschiene. Mittels der Peripherie-Module in 2-, 4- und 8-Kanalausführung können Sie dieses System passgenau an Ihre Automatisierungsaufgaben adaptieren. Der Verdrahtungsaufwand ist gering gehalten, da die DC 24V Leistungsversorgung im Rückwandbus integriert ist und defekte Elektronik-Module bei stehender Verdrahtung getauscht werden können. Durch Einsatz der farblich abgesetzten Power-Module können Sie innerhalb des Systems weitere Potenzialbereiche für die DC 24V Leistungsversorgung definieren, bzw. die Elektronikversorgung um 2A erweitern.



2.2.2 Komponenten

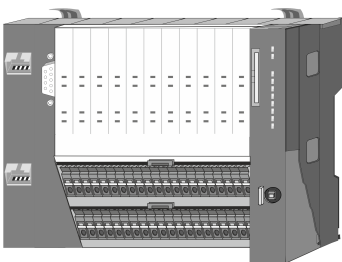
- CPU (Kopf-Modul)
- Bus-Koppler (Kopf-Modul)
- Zeilenanschlusung
- Peripherie-Module
- Zubehör



VORSICHT!

Beim Einsatz dürfen nur Module von VIPA kombiniert werden. Ein Mischbetrieb mit Modulen von Fremdherstellern ist nicht zulässig!

CPU 01xC



Bei der CPU 01xC sind CPU-Elektronik, Ein-/Ausgabe-Komponenten und Spannungsversorgung in ein Gehäuse integriert. Zusätzlich können am Rückwandbus bis zu 64 Peripherie-Module aus dem System SLIO angebunden werden. Als Kopf-Modul werden über die integrierte Spannungsversorgung sowohl die CPU-Elektronik, die Ein-/Ausgabe-Komponenten als auch die Elektronik der über den Rückwandbus angebunden Peripherie-Module versorgt. Zum Anschluss der Spannungsversorgung, der Ein-/Ausgabe-Komponenten und zur DC 24V Leistungsversorgung der über Rückwandbus angebunden Peripherie-Module besitzt die CPU abnehmbare Steckverbinder. Durch Montage von bis zu 64 Peripherie-Modulen am Rückwandbus der CPU werden diese elektrisch verbunden, d.h. sie sind am Rückwandbus eingebunden, die Elektronik-Module werden versorgt und jedes Peripherie-Modul ist an die DC 24V Leistungsversorgung angeschlossen.

CPU 01x



Bei der CPU 01x sind CPU-Elektronik und Power-Modul in ein Gehäuse integriert. Als Kopf-Modul werden über das integrierte Power-Modul zur Spannungsversorgung sowohl die CPU-Elektronik als auch die Elektronik der angebunden Peripherie-Module versorgt. Die DC 24V Leistungsversorgung für die angebunden Peripherie-Module erfolgt über einen weiteren Anschluss am Power-Modul. Durch Montage von bis zu 64 Peripherie-Modulen an der CPU werden diese elektrisch verbunden, d.h. sie sind am Rückwandbus eingebunden, die Elektronik-Module werden versorgt und jedes Peripherie-Modul ist an die DC 24V Leistungsversorgung angeschlossen.

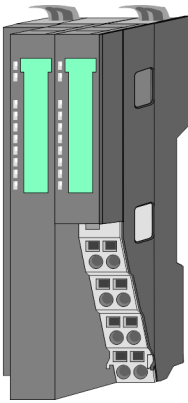


VORSICHT!

CPU-Teil und Power-Modul der CPU dürfen nicht voneinander getrennt werden!

Hier dürfen Sie lediglich das Elektronik-Modul tauschen!

Bus-Koppler



Beim Bus-Koppler sind Bus-Interface und Power-Modul in ein Gehäuse integriert. Das Bus-Interface bietet Anschluss an ein übergeordnetes Bus-System. Als Kopf-Modul werden über das integrierte Power-Modul zur Spannungsversorgung sowohl das Bus-Interface als auch die Elektronik der angebundenen Peripherie-Module versorgt. Die DC 24V Leistungsversorgung für die angebundenen Peripherie-Module erfolgt über einen weiteren Anschluss am Power-Modul. Durch Montage von bis zu 64 Peripherie-Modulen am Bus-Koppler werden diese elektrisch verbunden, d.h. sie sind am Rückwandbus eingebunden, die Elektronik-Module werden versorgt und jedes Peripherie-Modul ist an die DC 24V Leistungsversorgung angeschlossen.

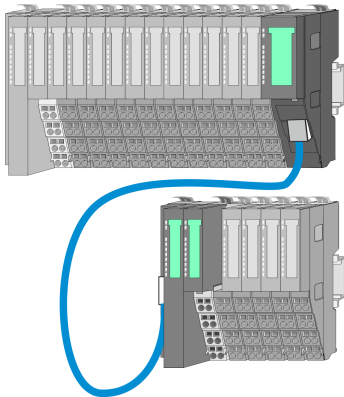


VORSICHT!

Bus-Interface und Power-Modul des Bus-Kopplers dürfen nicht voneinander getrennt werden!

Hier dürfen Sie lediglich das Elektronik-Modul tauschen!

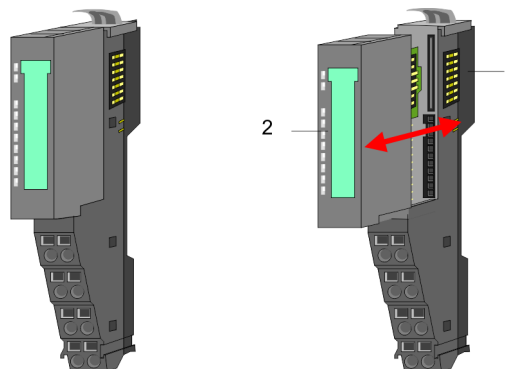
Zeilenanschlutung



Im System SLIO haben Sie die Möglichkeit bis zu 64 Module in einer Zeile zu stecken. Mit dem Einsatz der Zeilenanschlutung können Sie diese Zeile in mehrere Zeilen aufteilen. Hierbei ist am jeweiligen Zeilenende ein Zeilenanschlutung-Master-Modul zu setzen und die nachfolgende Zeile muss mit einem Zeilenanschlutung-Slave-Modul beginnen. Master und Slave sind über ein spezielles Verbindungskabel miteinander zu verbinden. Auf diese Weise können Sie eine Zeile auf bis zu 5 Zeilen aufteilen. Je Zeilenanschlutung vermindert sich die maximal Anzahl steckbarer Module am System SLIO Bus um 1. Für die Verwendung der Zeilenanschlutung ist keine gesonderte Projektierung erforderlich.

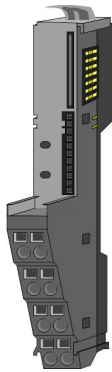
Peripherie-Module

Jedes Peripherie-Modul besteht aus einem *Terminal-* und einem *Elektronik-Modul*.



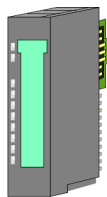
- 1 Terminal-Modul
- 2 Elektronik-Modul

Terminal-Modul



Das *Terminal-Modul* bietet die Aufnahme für das Elektronik-Modul, beinhaltet den Rückwandbus mit Spannungsversorgung für die Elektronik, die Anbindung an die DC 24V Leistungsversorgung und den treppenförmigen Klemmblock für die Verdrahtung. Zusätzlich besitzt das Terminal-Modul ein Verriegelungssystem zur Fixierung auf einer Tragschiene. Mittels dieser Verriegelung können Sie Ihr SLIO-System außerhalb Ihres Schaltschranks aufbauen und später als Gesamtsystem im Schaltschrank montieren.

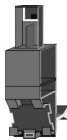
Elektronik-Modul



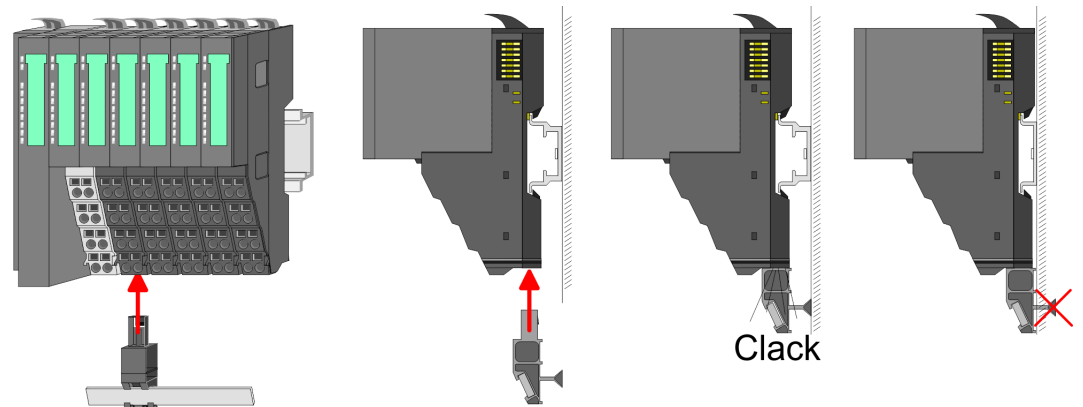
Über das *Elektronik-Modul*, welches durch einen sicheren Schiebemechanismus mit dem Terminal-Modul verbunden ist, wird die Funktionalität eines SLIO-Peripherie-Moduls definiert. Im Fehlerfall können Sie das defekte Elektronik-Modul gegen ein funktionsfähiges Modul tauschen. Hierbei bleibt die Verdrahtung bestehen. Auf der Frontseite befinden sich LEDs zur Statusanzeige. Für die einfache Verdrahtung finden Sie bei jedem Elektronik-Modul auf der Front und an der Seite entsprechende Anschlussbilder.

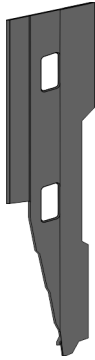
2.2.3 Zubehör

Schirmschienen-Träger



Der Schirmschienen-Träger (Best.-Nr.: 000-0AB00) dient zur Aufnahme von Schirmschienen (10mm x 3mm) für den Anschluss von Kabelschirmen. Schirmschienen-Träger, Schirmschiene und Kabelschirmbefestigungen sind nicht im Lieferumfang enthalten, sondern ausschließlich als Zubehör erhältlich. Der Schirmschienen-Träger wird unterhalb des Klemmblocks in das Terminal-Modul gesteckt. Bei flacher Tragschiene können Sie zur Adaption die Abstandshalter am Schirmschienen-Träger abbrechen.

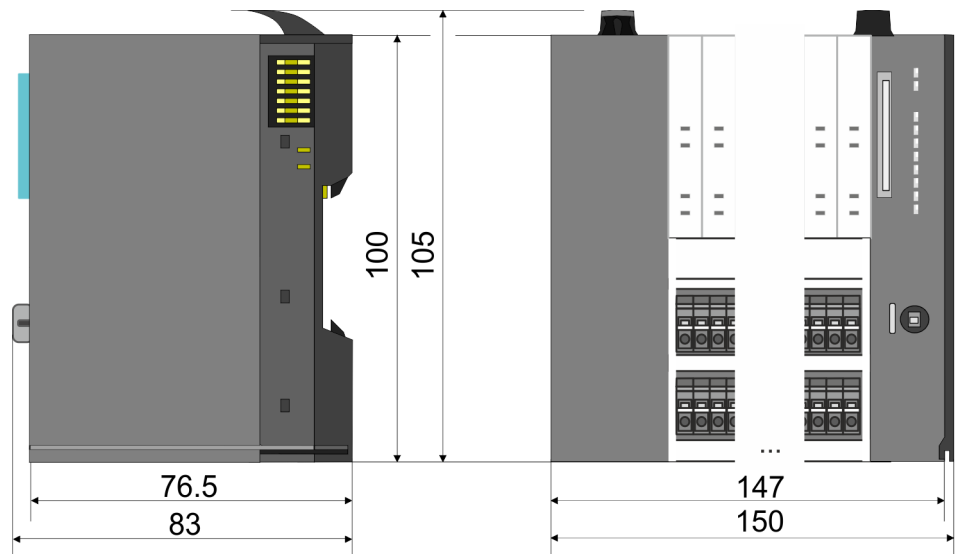


Bus-Blende

Bei jedem Kopf-Modul gehört zum Schutz der Bus-Kontakte eine Bus-Blende zum Lieferumfang. Vor der Montage von System SLIO-Modulen ist die Bus-Blende am Kopf-Modul zu entfernen. Zum Schutz der Bus-Kontakte müssen Sie die Bus-Blende immer am äußersten Modul montieren. Die Bus-Blende hat die Best.-Nr. 000-0AA00.

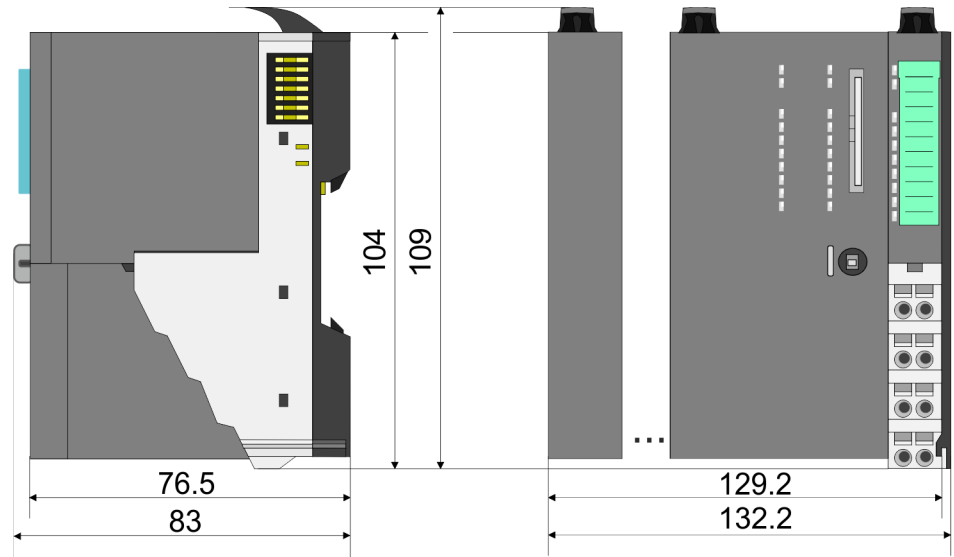
Kodier-Stecker

Sie haben die Möglichkeit die Zuordnung von Terminal- und Elektronik-Modul zu fixieren. Hierbei kommen Kodier-Stecker (Best-Nr.: 000-0AC00) von VIPA zum Einsatz. Die Kodier-Stecker bestehen aus einem Kodierstift-Stift und einer Kodier-Buchse, wobei durch Zusammenfügen von Elektronik- und Terminal-Modul der Kodier-Stift am Terminal-Modul und die Kodier-Buchse im Elektronik-Modul verbleiben. Dies gewährleistet, dass nach Austausch des Elektronik-Moduls nur wieder ein Elektronik-Modul mit der gleichen Kodierung gesteckt werden kann.

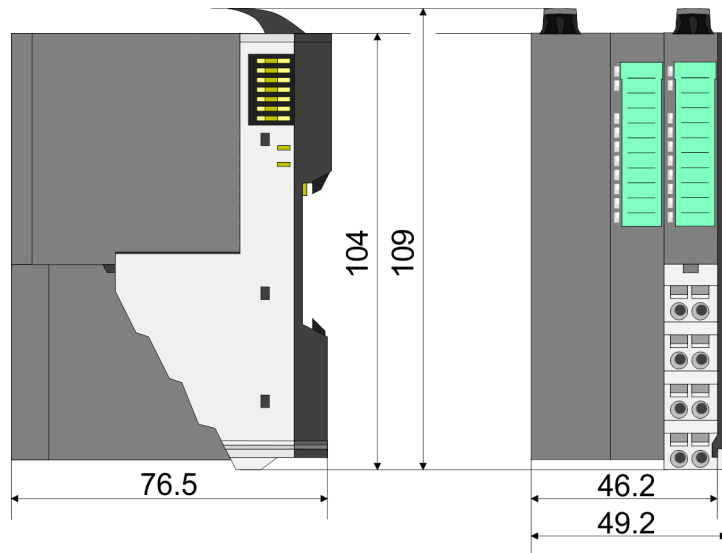
2.3 Abmessungen**Maße CPU 01xC**

Abmessungen

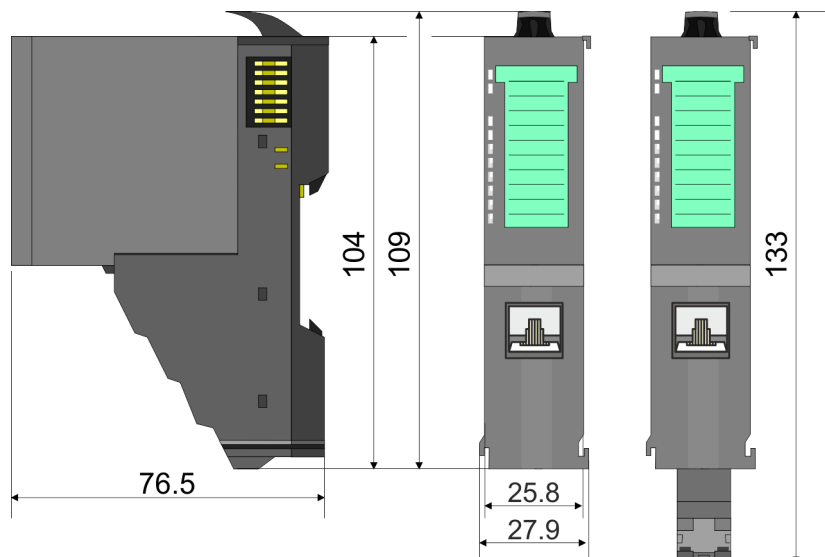
Maße CPU 01x



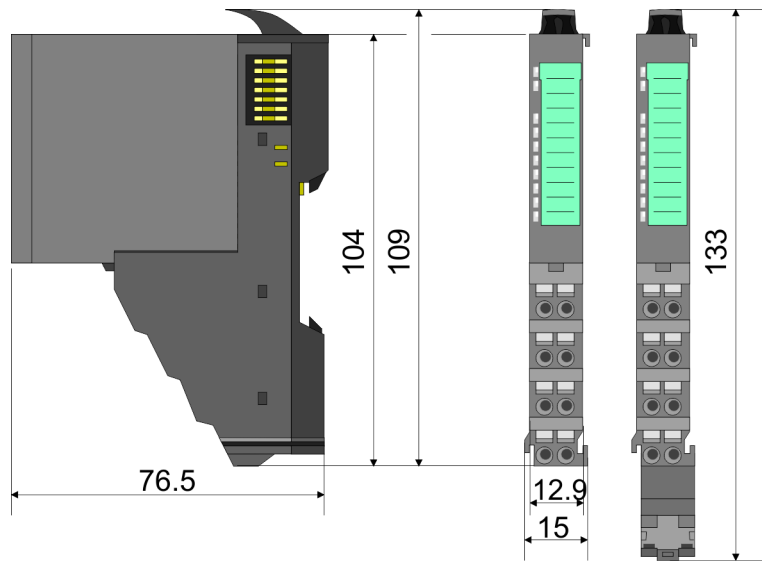
Maße Bus-Koppler und Zeilenanschlusung Slave



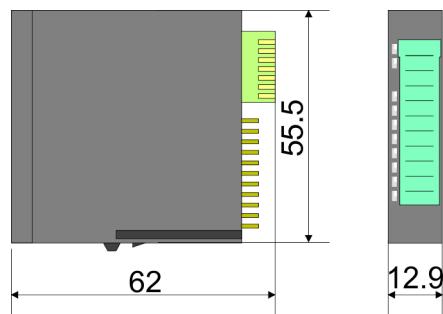
Maße Zeilenanschlusung Master



Maße Peripherie-Modul



Maße Elektronik-Modul



Maße in mm

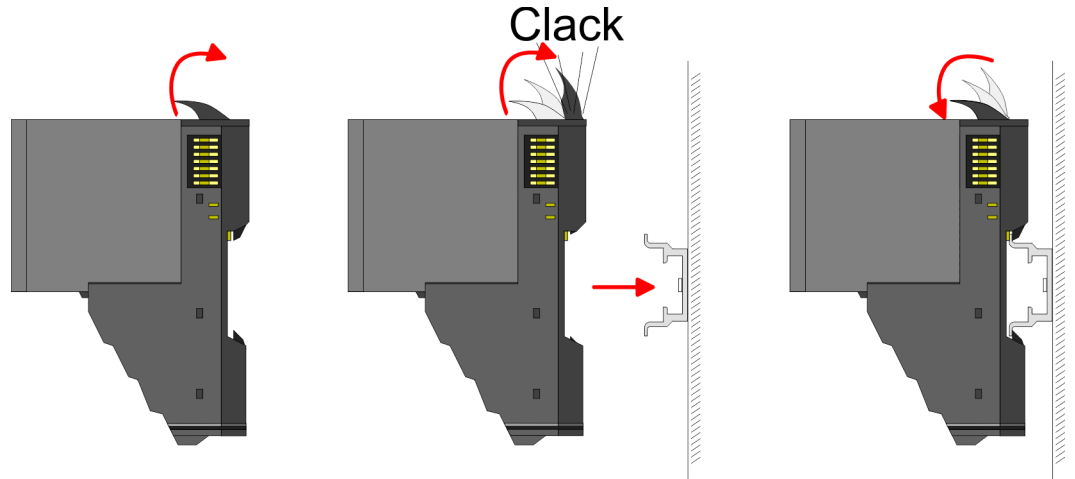
2.4 Montage Peripherie-Module



Voraussetzungen für den UL-konformen Betrieb

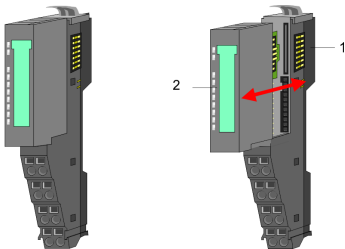
- Verwenden Sie für die Spannungsversorgung ausschließlich SELV/ PELV-Netzteile.
- Das System SLIO darf nur in einem Gehäuse gemäß IEC61010-1 9.3.2 c) eingebaut und betrieben werden.

Das Modul besitzt einen Verriegelungshebel an der Oberseite. Zur Montage und Demontage ist dieser Hebel nach oben zu drücken, bis er einrastet. Stecken Sie das zu montierende Modul an das zuvor gesteckte Modul und schieben Sie das Modul, geführt durch die Führungsleisten an der Ober- und Unterseite, auf die Tragschiene. Durch Klappen des Verriegelungshebels nach unten wird das Modul auf der Tragschiene fixiert. Sie können entweder die Module einzeln auf der Tragschiene montieren oder als Block. Hierbei ist zu beachten, dass jeder Verriegelungshebel geöffnet ist. Die einzelnen Module werden direkt auf eine Tragschiene montiert. Über die Verbindung mit dem Rückwandbus werden Elektronik- und Leistungsversorgung angebinden. Sie können bis zu 64 Module stecken. Bitte beachten Sie hierbei, dass der Summenstrom der Elektronikversorgung den Maximalwert von 3A nicht überschreitet. Durch Einsatz des Power-Moduls 007-1AB10 können Sie den Strom für die Elektronikversorgung entsprechend erweitern.



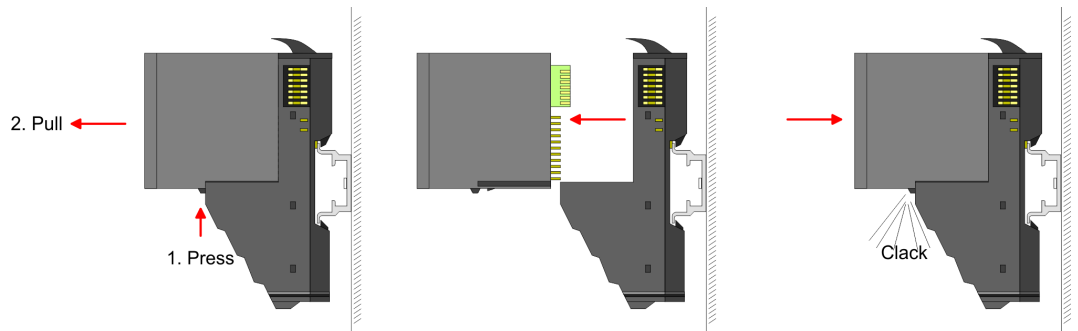
Terminal- und Elektronik-Modul

Jedes Peripherie-Modul besteht aus einem *Terminal-Modul* und einem *Elektronik-Modul*.



- 1 Terminal-Modul
- 2 Elektronik-Modul

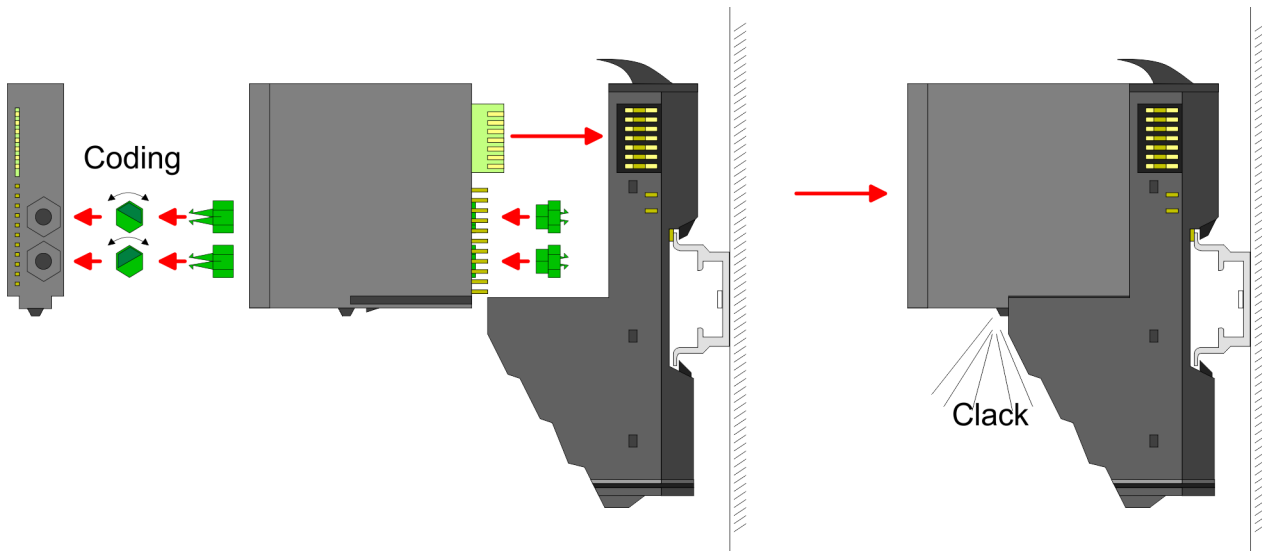
Zum Austausch eines Elektronik-Moduls können Sie das Elektronik-Modul, nach Betätigung der Entriegelung an der Unterseite, nach vorne abziehen. Für die Montage schieben Sie das Elektronik-Modul in die Führungsschiene, bis dieses an der Unterseite hörbar am Terminal-Modul einrastet.



Kodierung



Sie haben die Möglichkeit die Zuordnung von Terminal- und Elektronik-Modul zu fixieren. Hierbei kommen Kodier-Stecker (Best-Nr.: 000-0AC00) von VIPA zum Einsatz. Die Kodier-Stecker bestehen aus einem Kodierstift-Stift und einer Kodier-Buchse, wobei durch Zusammenfügen von Elektronik- und Terminal-Modul der Kodier-Stift am Terminal-Modul und die Kodier-Buchse im Elektronik-Modul verbleiben. Dies gewährleistet, dass nach Austausch des Elektronik-Moduls nur wieder ein Elektronik-Modul mit der gleichen Kodierung gesteckt werden kann.



Jedes Elektronik-Modul besitzt an der Rückseite 2 Kodier-Aufnehmer für Kodier-Buchsen. Durch ihre Ausprägung sind 6 unterschiedliche Positionen pro Kodier-Buchse steckbar. Somit haben sie bei Verwendung beider Kodier-Aufnehmer 36 Kombinationsmöglichkeiten für die Kodierung.

1. ➤ Stecken Sie gemäß Ihrer Kodierung 2 Kodier-Buchsen in die Aufnehmer am Elektronik-Modul, bis diese einrasten.
2. ➤ Stecken Sie nun den entsprechenden Kodier-Stift in die Kodier-Buchse.
3. ➤ Zur Fixierung der Kodierung führen Sie Elektronik- und Terminal-Modul zusammen, bis diese hörbar einrasten.

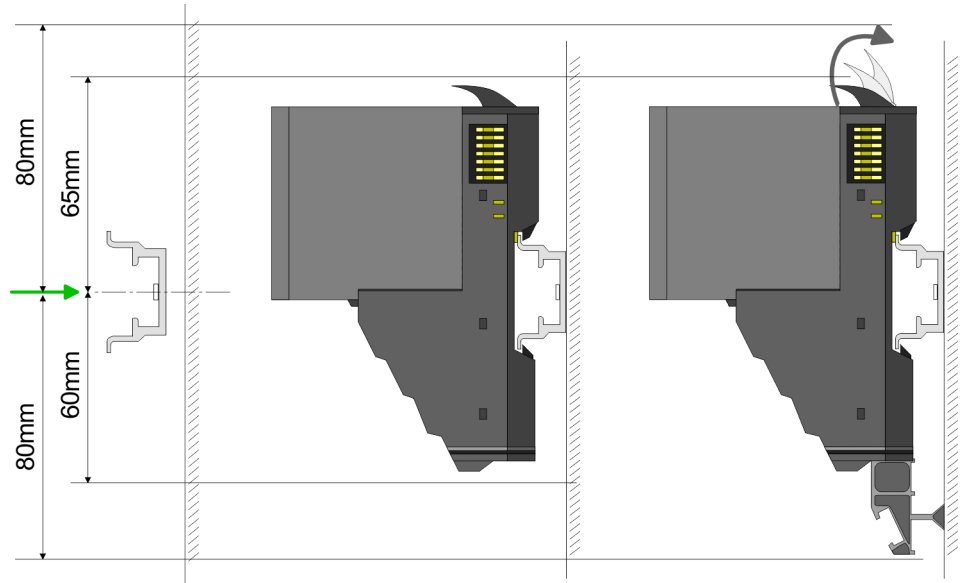


VORSICHT!

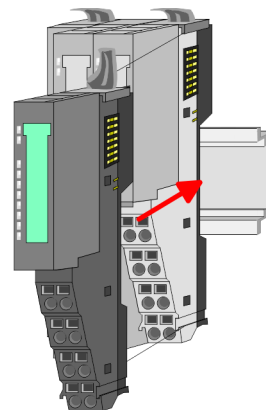
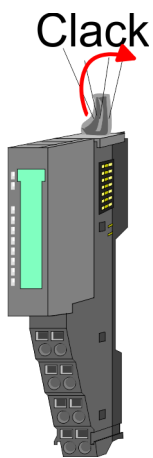
Bitte beachten Sie, dass bei Austausch eines bereits kodierten Elektronik-Moduls dieses immer durch ein Elektronik-Modul mit gleicher Kodierung ersetzt wird.

Auch bei vorhandener Kodierung am Terminal-Modul können Sie ein Elektronik-Modul ohne Kodierung stecken. Die Verantwortung bei der Verwendung von Kodierstiften liegt beim Anwender. VIPA übernimmt keinerlei Haftung für falsch gesteckte Elektronik-Module oder für Schäden, welche aufgrund fehlerhafter Kodierung entstehen!

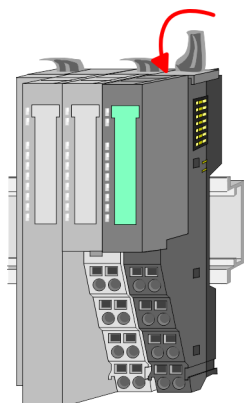
Montage Peripherie-Modul



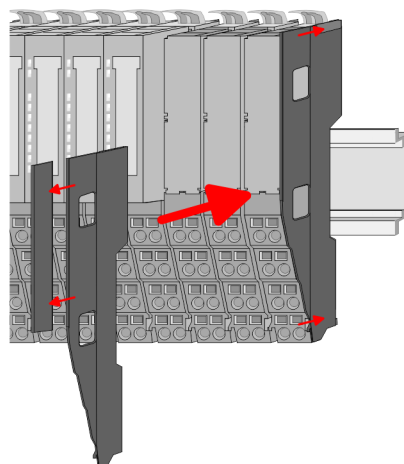
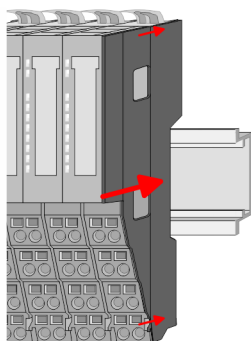
1. ➤ Montieren Sie die Tragschiene! Bitte beachten Sie, dass Sie von der Mitte der Tragschiene nach oben einen Montageabstand von mindestens 80mm und nach unten von 60mm bzw. 80mm bei Verwendung von Schirmschienen-Trägern einhalten.
2. ➤ Montieren Sie Ihr Kopfmodul wie z.B. CPU oder Feldbus-Koppler.
3. ➤ Entfernen Sie vor der Montage der Peripherie-Module die Bus-Blende auf der rechten Seite des Kopf-Moduls, indem Sie diese nach vorn abziehen. Bewahren Sie die Blende für spätere Montage auf.



4. ➤ Klappen Sie zur Montage den Verriegelungshebel des Peripherie-Moduls nach oben, bis dieser einrastet.
5. ➤ Stecken Sie das zu montierende Modul an das zuvor gesteckte Modul und schieben Sie das Modul, geführt durch die Führungsleisten an der Ober- und Unterseite, auf die Tragschiene.



6. ➔ Klappen Sie den Verriegelungshebel des Peripherie-Moduls wieder nach unten.



7. ➔ Nachdem Sie Ihr Gesamt-System montiert haben, müssen Sie zum Schutz der Bus-Kontakte die Bus-Blende am äußersten Modul wieder stecken. Handelt es sich bei dem äußersten Modul um ein Klemmen-Modul, so ist zur Adaption der obere Teil der Bus-Blende abzubrechen.

2.5 Verdrahtung Peripherie-Module

Terminal-Modul Anschlussklemmen



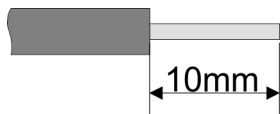
VORSICHT!

Keine gefährliche Spannungen anschließen!

Sofern dies nicht ausdrücklich bei der entsprechenden Modulbeschreibung vermerkt ist, dürfen Sie an dem entsprechenden Terminal-Modul keine gefährlichen Spannungen anschließen!

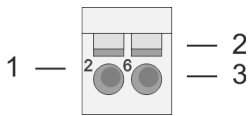
Bei der Verdrahtung von Terminal-Modulen kommen Anschlussklemmen mit Federklemmtechnik zum Einsatz. Die Verdrahtung mit Federklemmtechnik ermöglicht einen schnellen und einfachen Anschluss Ihrer Signal- und Versorgungsleitungen. Im Gegensatz zur Schraubverbindung ist diese Verbindungsart erschütterungssicher.

Daten

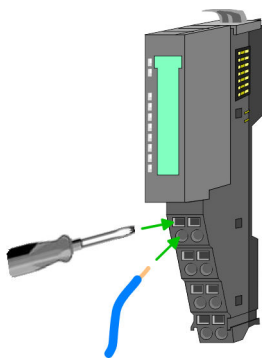
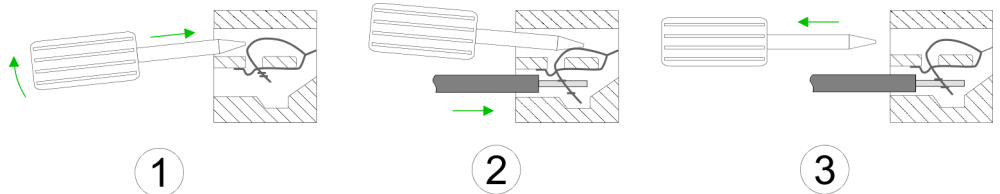


U_{\max}	240V AC / 30V DC
I_{\max}	10A
Querschnitt	0,08 ... 1,5mm ² (AWG 28 ... 16)
Abisolierlänge	10mm

Verdrahtung Vorgehensweise

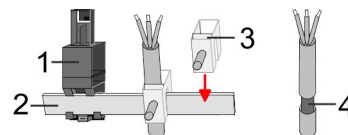


- 1 Pin-Nr. am Steckverbinder
- 2 Entriegelung für Schraubendreher
- 3 Anschlussöffnung für Draht



- 1. 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.
- 2. Führen Sie durch die runde Öffnung Ihren abisolierten Draht ein. Sie können Drähte mit einem Querschnitt von 0,08mm² bis 1,5mm² anschließen.
- 3. Durch Entfernen des Schraubendrehers wird der Draht über einen Federkontakt sicher mit der Anschlussklemme verbunden.

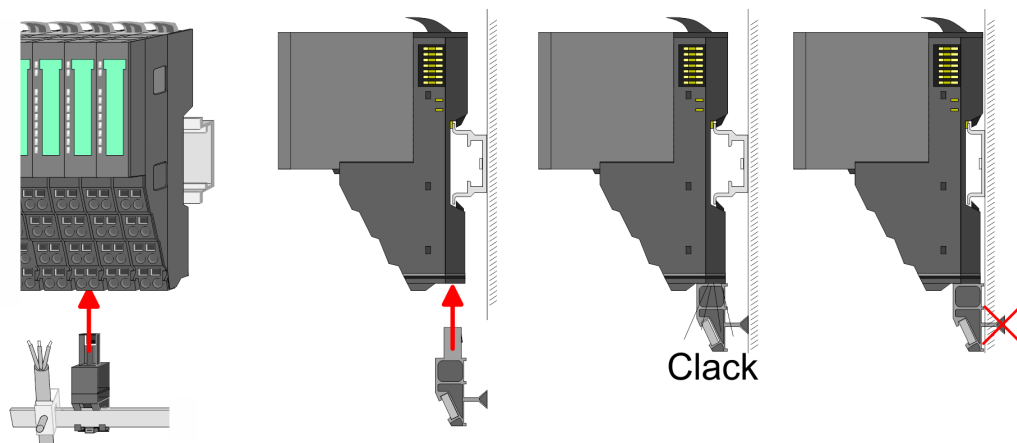
Schirm auflegen



- 1 Schirmschienen-Träger
- 2 Schirmschiene (10mm x 3mm)
- 3 Schirmanschlussklemme
- 4 Kabelschirm

Zur Schirmauflage ist die Montage von Schirmschienen-Trägern erforderlich. Der Schirmschienen-Träger (als Zubehör erhältlich) dient zur Aufnahme der Schirmschiene für den Anschluss von Kabelschirmen.

- 1. Jedes System SLIO-Modul besitzt an der Unterseite Aufnehmer für Schirmschienen-Träger. Stecken Sie Ihre Schirmschienen-Träger, bis diese am Modul einrasten. Bei flacher Tragschiene können Sie zur Adaption den Abstandshalter am Schirmschienen-Träger abbrechen.
- 2. Legen Sie Ihre Schirmschiene in den Schirmschienen-Träger ein.



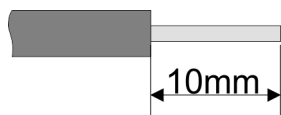
3. Legen Sie ihre Kabel mit dem entsprechend abisolierten Kabelschirm auf und verbinden Sie diese über die Schirmanschlussklemme mit der Schirmschiene.

2.6 Verdrahtung Power-Module

Terminal-Modul Anschlussklemmen

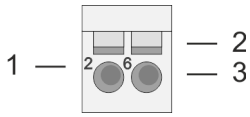
Power-Module sind entweder im Kopf-Modul integriert oder können zwischen die Peripherie-Module gesteckt werden. Bei der Verdrahtung von Power-Modulen kommen Anschlussklemmen mit Federklemmtechnik zum Einsatz. Die Verdrahtung mit Federklemmtechnik ermöglicht einen schnellen und einfachen Anschluss Ihrer Signal- und Versorgungsleitungen. Im Gegensatz zur Schraubverbindung ist diese Verbindungsart erschütterungssicher.

Daten

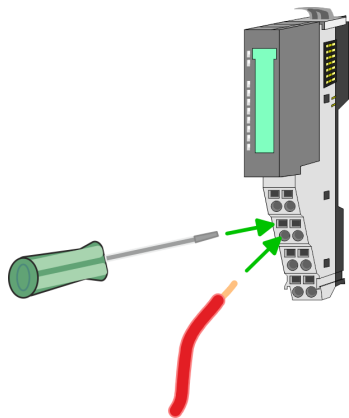
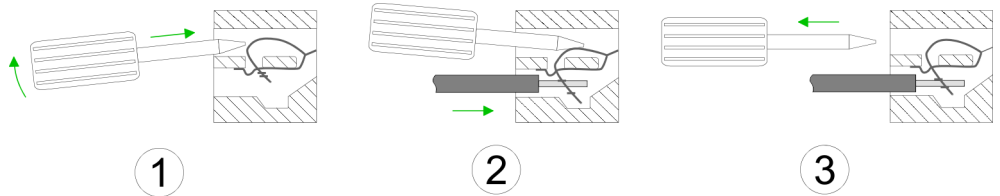


U_{\max}	30V DC
I_{\max}	10A
Querschnitt	0,08 ... 1,5mm ² (AWG 28 ... 16)
Abisolierlänge	10mm

Verdrahtung Vorgehensweise

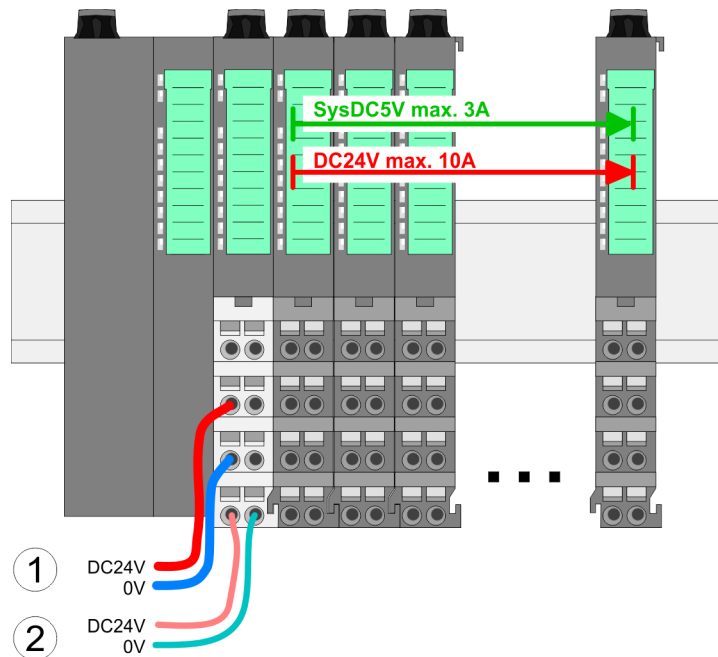


- 1 Pin-Nr. am Steckverbinder
- 2 Entriegelung für Schraubendreher
- 3 Anschlussöffnung für Draht

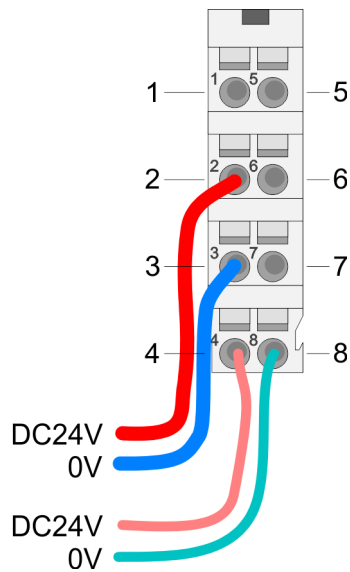


1. Zum Verdrahten stecken Sie, wie in der Abbildung gezeigt, einen passenden Schraubendreher leicht schräg in die rechteckige Öffnung. Zum Öffnen der Kontaktfeder müssen Sie den Schraubendreher in die entgegengesetzte Richtung drücken und halten.
2. Führen Sie durch die runde Öffnung Ihren abisolierten Draht ein. Sie können Drähte mit einem Querschnitt von 0,08mm² bis 1,5mm² anschließen.
3. Durch Entfernen des Schraubendrehers wird der Draht über einen Federkontakt sicher mit der Anschlussklemme verbunden.

Standard-Verdrahtung



- (1) DC 24V für Leistungsversorgung I/O-Ebene (max. 10A)
- (2) DC 24V für Elektronikversorgung Bus-Koppler und I/O-Ebene

PM - Power Modul

Für Drähte mit einem Querschnitt von 0,08mm² bis 1,5mm².

Pos.	Funktion	Typ	Beschreibung
1	---	---	nicht belegt
2	DC 24V	E	DC 24V für Leistungsversorgung
3	0V	E	GND für Leistungsversorgung
4	Sys DC 24V	E	DC 24V für Elektronikversorgung
5	---	---	nicht belegt
6	DC 24V	E	DC 24V für Leistungsversorgung
7	0V	E	GND für Leistungsversorgung
8	Sys 0V	E	GND für Elektronikversorgung

E: Eingang

**VORSICHT!**

Da die Leistungsversorgung keine interne Absicherung besitzt, ist diese extern mit einer Sicherung entsprechend dem Maximalstrom abzusichern, d.h. max. 10A mit einer 10A-Sicherung (flink) bzw. einem Leitungsschutzschalter 10A Charakteristik Z!



Die Elektronikversorgung ist intern gegen zu hohe Spannung durch eine Sicherung geschützt. Die Sicherung befindet sich innerhalb des Power-Moduls. Wenn die Sicherung ausgelöst hat, muss das Elektronik-Modul getauscht werden!

Absicherung

- Die Leistungsversorgung ist extern mit einer Sicherung entsprechend dem Maximalstrom abzusichern, d.h. max. 10A mit einer 10A-Sicherung (flink) bzw. einem Leitungsschutzschalter 10A Charakteristik Z.
- Es wird empfohlen die Elektronikversorgung für Kopf-Modul und I/O-Ebene extern mit einer 2A-Sicherung (flink) bzw. einem Leitungsschutzschalter 2A Charakteristik Z abzusichern.
- Die Elektronikversorgung für die I/O-Ebene des Power-Moduls 007-1AB10 sollte ebenfalls extern mit einer 1A-Sicherung (flink) bzw. einem Leitungsschutzschalter 1A Charakteristik Z abgesichert werden.

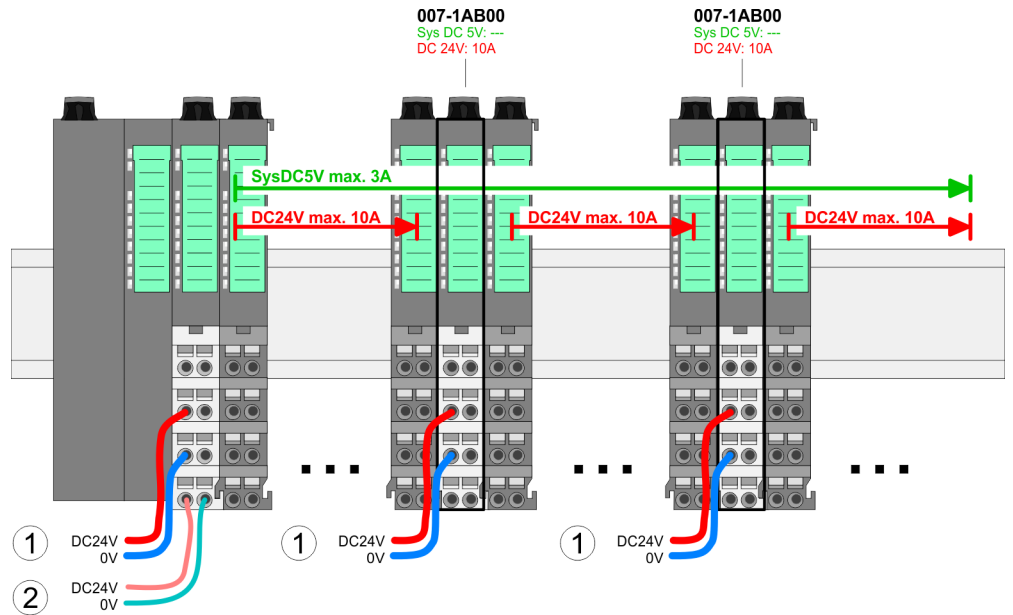
Zustand der Elektronikversorgung über LEDs

Nach PowerON des System SLIO leuchtet an jedem Modul die RUN- bzw. MF-LED, sofern der Summenstrom für die Elektronikversorgung 3A nicht übersteigt. Ist der Summenstrom größer als 3A, werden die LEDs nicht mehr angesteuert. Hier müssen Sie zwischen Ihre Peripherie-Module das Power-Modul mit der Best.-Nr. 007-1AB10 platzieren.

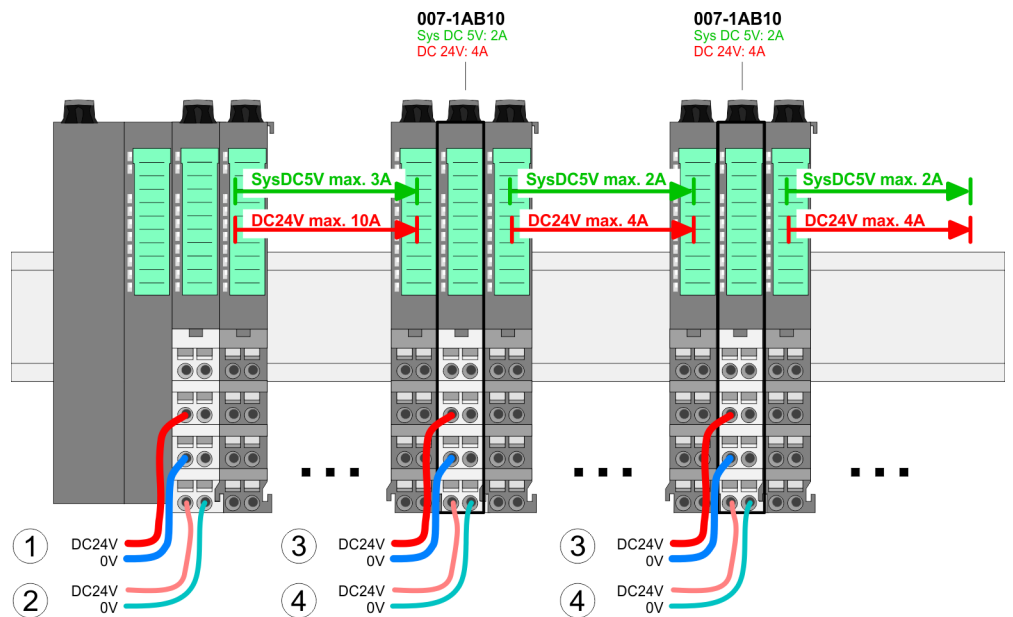
Einsatz von Power-Modulen

- Das Power-Modul mit der Best.-Nr. 007-1AB00 setzen Sie ein, wenn die 10A für die Leistungsversorgung nicht mehr ausreichen. Sie haben so auch die Möglichkeit, Potenzialgruppen zu bilden.
- Das Power-Modul mit der Best.-Nr. 007-1AB10 setzen Sie ein, wenn die 3A für die Elektronikversorgung am Rückwandbus nicht mehr ausreichen. Zusätzlich erhalten Sie eine neue Potenzialgruppe für die DC 24V Leistungsversorgung mit max. 4A.
- Durch Stecken des Power-Moduls 007-1AB10 können am nachfolgenden Rückwandbus Module gesteckt werden mit einem maximalen Summenstrom von 2A. Danach ist wieder ein Power-Modul zu stecken. Zur Sicherstellung der Spannungsversorgung dürfen die Power-Module beliebig gemischt eingesetzt werden.

Power-Modul 007-1AB00

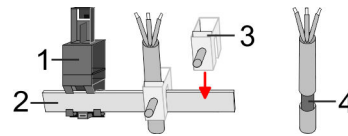


Power-Modul 007-1AB10



- (1) DC 24V für Leistungsversorgung I/O-Ebene (max. 10A)
- (2) DC 24V für Elektronikversorgung Bus-Koppler und I/O-Ebene
- (3) DC 24V für Leistungsversorgung I/O-Ebene (max. 4A)
- (4) DC 24V für Elektronikversorgung I/O-Ebene

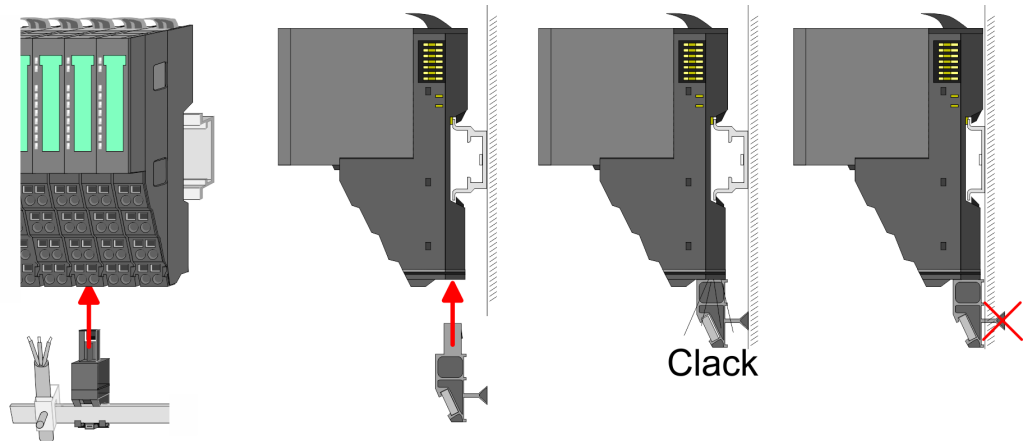
Schirm auflegen



- 1 Schirmschienen-Träger
- 2 Schirmschiene (10mm x 3mm)
- 3 Schirmanschlussklemme
- 4 Kabelschirm

Zur Schirmauflage ist die Montage von Schirmschienen-Trägern erforderlich. Der Schirmschienen-Träger (als Zubehör erhältlich) dient zur Aufnahme der Schirmschiene für den Anschluss von Kabelschirmen.

- 1. ➤ Jedes System SLIO-Modul besitzt an der Unterseite Aufnehmer für Schirmschienen-Träger. Stecken Sie Ihre Schirmschienen-Träger, bis diese am Modul einrasten. Bei flacher Tragschiene können Sie zur Adaption den Abstandshalter am Schirmschienen-Träger abbrechen.
- 2. ➤ Legen Sie Ihre Schirmschiene in den Schirmschienen-Träger ein.



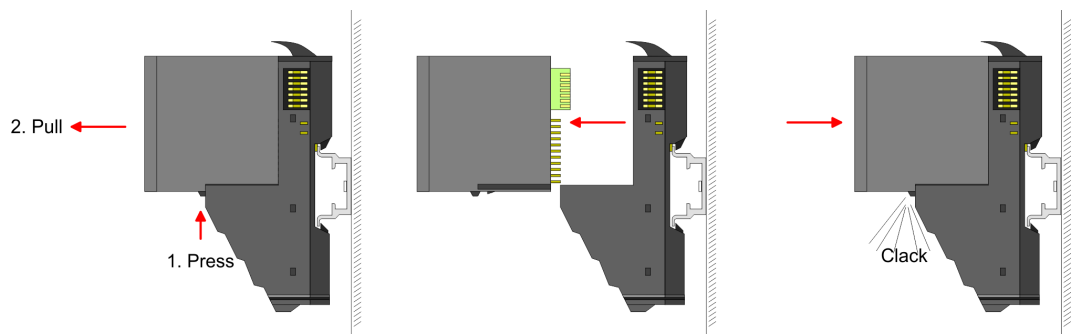
- 3. ➤ Legen Sie ihre Kabel mit dem entsprechend abisolierten Kabelschirm auf und verbinden Sie diese über die Schirmanschlussklemme mit der Schirmschiene.

2.7 Demontage Peripherie-Module

Vorgehensweise

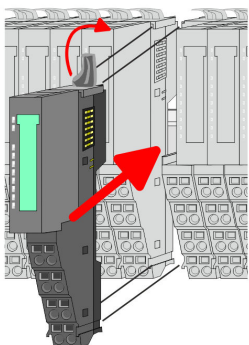
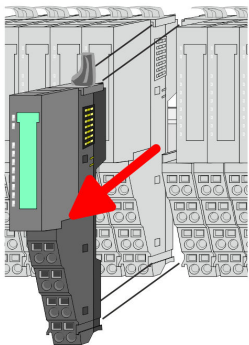
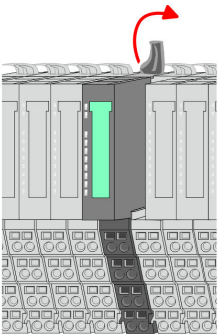
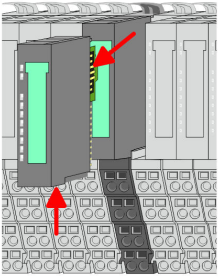
Austausch eines Elektronik-Moduls

- 1. ➤ Machen Sie Ihr System stromlos.



- 2. ➤ Zum Austausch eines Elektronik-Moduls können Sie das Elektronik-Modul, nach Betätigung der Entriegelung an der Unterseite, nach vorne abziehen.
- 3. ➤ Für die Montage schieben Sie das neue Elektronik-Modul in die Führungsschiene, bis dieses an der Unterseite am Terminal-Modul einrastet.
 - ⇒ Jetzt können Sie Ihr System wieder in Betrieb nehmen.

Austausch eines Peripherie-Moduls



1. ➤ Machen Sie Ihr System stromlos.
2. ➤ Entfernen Sie falls vorhanden die Verdrahtung am Modul.
3. ➤

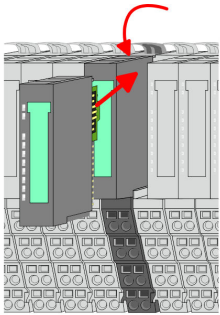
i *Bei der Demontage und beim Austausch eines (Kopf)-Moduls oder einer Modulgruppe müssen Sie aus montage-technischen Gründen immer das rechts daneben befindliche Elektronik-Modul entfernen! Nach der Montage kann es wieder gesteckt werden.*

Betätigen Sie die Entriegelung an der Unterseite des rechts daneben befindlichen Elektronik-Moduls und ziehen Sie dieses nach vorne ab.

4. ➤ Klappen Sie den Verriegelungshebel des zu tauschenden Moduls nach oben.

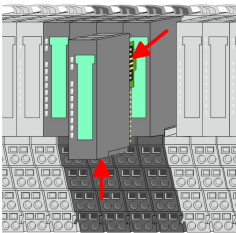
5. ➤ Ziehen Sie das Modul nach vorne ab.
6. ➤ Zur Montage klappen Sie den Verriegelungshebel des zu montierenden Moduls nach oben.

7. ➤ Stecken Sie das zu montierende Modul in die Lücke zwischen die beiden Module und schieben Sie das Modul, geführt durch die Führungsleisten auf beiden Seiten, auf die Tragschiene.
8. ➤ Klappen Sie den Verriegelungshebel wieder nach unten.



9. ➤ Stecken Sie wieder das zuvor entnommene Elektronik-Modul.
10. ➤ Verdrahten Sie Ihr Modul.
⇒ Jetzt können Sie Ihr System wieder in Betrieb nehmen.

Austausch einer Modulgruppe



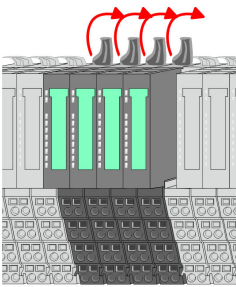
1. ➤ Machen Sie Ihr System stromlos.
2. ➤ Entfernen Sie falls vorhanden die Verdrahtung an der Modulgruppe.
3. ➤



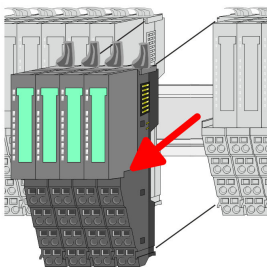
Bei der Demontage und beim Austausch eines (Kopf)-Moduls oder einer Modulgruppe müssen Sie aus montage-technischen Gründen immer das rechts daneben befindliche Elektronik-Modul entfernen! Nach der Montage kann es wieder gesteckt werden.

Betätigen Sie die Entriegelung an der Unterseite des rechts neben der Modulgruppe befindlichen Elektronik-Moduls und ziehen Sie dieses nach vorne ab.

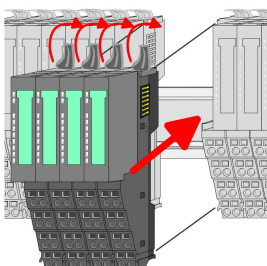
4. ➤ Klappen Sie alle Verriegelungshebel der zu tauschenden Modulgruppe nach oben.



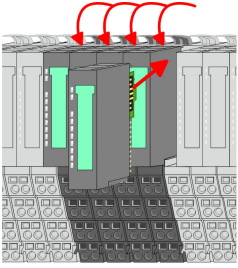
5. ➤ Ziehen Sie die Modulgruppe nach vorne ab.
6. ➤ Zur Montage klappen Sie alle Verriegelungshebel der zu montierenden Modulgruppe nach oben.



7. ➤ Stecken Sie die zu montierende Modulgruppe in die Lücke zwischen die beiden Module und schieben Sie die Modulgruppe, geführt durch die Führungsleisten auf beiden Seiten, auf die Tragschiene.
8. ➤ Klappen Sie alle Verriegelungshebel wieder nach unten.



Demontage Peripherie-Module




- 9.** ➤ Stecken Sie wieder das zuvor entnommene Elektronik-Modul.
- 10.** ➤ Verdrahten Sie Ihre Modulgruppe.
 - ⇒ Jetzt können Sie Ihr System wieder in Betrieb nehmen.

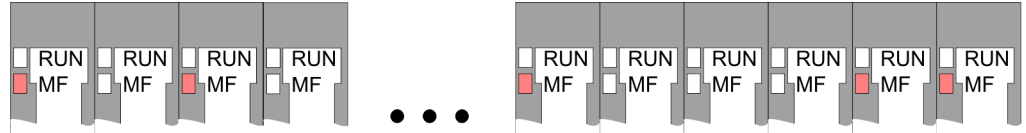
2.8 Hilfe zur Fehlersuche - LEDs

Allgemein

Jedes Modul besitzt auf der Frontseite die LEDs RUN und MF. Mittels dieser LEDs können Sie Fehler in Ihrem System bzw. fehlerhafte Module ermitteln.

In den nachfolgenden Abbildungen werden blinkende LEDs mit  gekennzeichnet.

Summenstrom der Elektronik-Versorgung überschritten

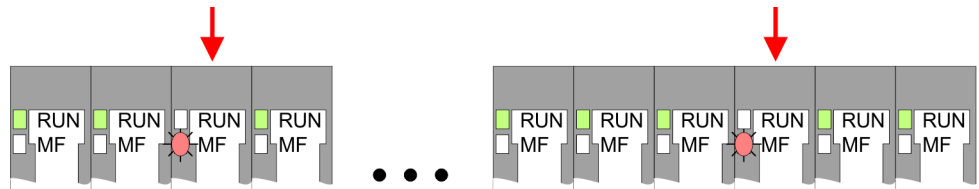


Verhalten: Nach dem Einschalten bleibt an jedem Modul die RUN-LED aus und es leuchtet sporadisch die MF-LED.

Ursache: Der maximale Strom für die Elektronikversorgung ist überschritten.

Abhilfe: Platzieren Sie immer, sobald der Summenstrom für die Elektronikversorgung den maximalen Strom übersteigt, das Power-Modul 007-1AB10. [↪ Kapitel 2.6 "Verdrahtung Power-Module" auf Seite 21](#)

Konfigurationsfehler

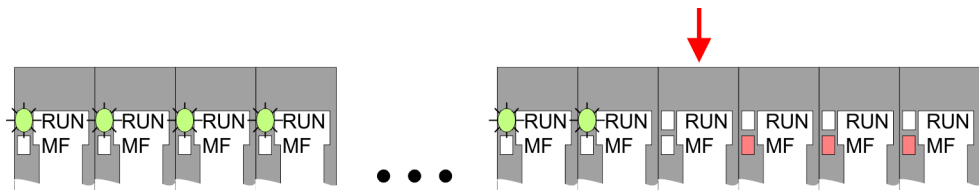


Verhalten: Nach dem Einschalten blinkt an einem Modul bzw. an mehreren Modulen die MF-LED. Die RUN-LED bleibt ausgeschaltet.

Ursache: An dieser Stelle ist ein Modul gesteckt, welches nicht dem aktuell konfigurierten Modul entspricht.

Abhilfe: Stimmen Sie Konfiguration und Hardware-Aufbau aufeinander ab.

Modul-Ausfall



Verhalten: Nach dem Einschalten blinken alle RUN-LEDs bis zum fehlerhaften Modul. Bei allen nachfolgenden Modulen leuchtet die MF LED und die RUN-LED ist aus.

Ursache: Das Modul rechts der blinkenden Module ist defekt.

Abhilfe: Ersetzen Sie das defekte Modul.

2.9 Aufbaurichtlinien

Allgemeines

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

Was bedeutet EMV?

Unter Elektromagnetischer Verträglichkeit (EMV) versteht man die Fähigkeit eines elektrischen Gerätes, in einer vorgegebenen elektromagnetischen Umgebung fehlerfrei zu funktionieren, ohne vom Umfeld beeinflusst zu werden bzw. das Umfeld in unzulässiger Weise zu beeinflussen.

Die Komponenten von VIPA sind für den Einsatz in Industrieumgebungen entwickelt und erfüllen hohe Anforderungen an die EMV. Trotzdem sollten Sie vor der Installation der Komponenten eine EMV-Planung durchführen und mögliche Störquellen in die Betrachtung einbeziehen.

Mögliche Störeinträge

Elektromagnetische Störungen können sich auf unterschiedlichen Pfaden in Ihre Steuerung einkoppeln:

- Elektromagnetische Felder (HF-Einkopplung)
- Magnetische Felder mit energietechnischer Frequenz
- Bus-System
- Stromversorgung
- Schutzleiter

Je nach Ausbreitungsmedium (leitungsgebunden oder -ungebunden) und Entfernung zur Störquelle gelangen Störungen über unterschiedliche Kopplungsmechanismen in Ihre Steuerung.

Man unterscheidet:

- galvanische Kopplung
- kapazitive Kopplung
- induktive Kopplung
- Strahlungskopplung

Grundregeln zur Sicherstellung der EMV

Häufig genügt zur Sicherstellung der EMV das Einhalten einiger elementarer Regeln. Beachten Sie beim Aufbau der Steuerung deshalb die folgenden Grundregeln.

- Achten Sie bei der Montage Ihrer Komponenten auf eine gut ausgeführte flächenhafte Massung der inaktiven Metallteile.
 - Stellen Sie eine zentrale Verbindung zwischen der Masse und dem Erde/Schutzleitersystem her.
 - Verbinden Sie alle inaktiven Metallteile großflächig und impedanzarm.
 - Verwenden Sie nach Möglichkeit keine Aluminiumteile. Aluminium oxidiert leicht und ist für die Massung deshalb weniger gut geeignet.
- Achten Sie bei der Verdrahtung auf eine ordnungsgemäße Leitungsführung.
 - Teilen Sie die Verkabelung in Leitungsgruppen ein. (Starkstrom, Stromversorgungs-, Signal- und Datenleitungen).
 - Verlegen Sie Starkstromleitungen und Signal- bzw. Datenleitungen immer in getrennten Kanälen oder Bündeln.
 - Führen Sie Signal- und Datenleitungen möglichst eng an Masseflächen (z.B. Tragholme, Metallschienen, Schrankbleche).

- Achten Sie auf die einwandfreie Befestigung der Leitungsschirme.
 - Datenleitungen sind geschirmt zu verlegen.
 - Analogleitungen sind geschirmt zu verlegen. Bei der Übertragung von Signalen mit kleinen Amplituden kann das einseitige Auflegen des Schirms vorteilhaft sein.
 - Legen Sie die Leitungsschirme direkt nach dem Schrankeintritt großflächig auf eine Schirm-/Schutzleiterschiene auf, und befestigen Sie die Schirme mit Kabelschellen.
 - Achten Sie darauf, dass die Schirm-/Schutzleiterschiene impedanzarm mit dem Schrank verbunden ist.
 - Verwenden Sie für geschirmte Datenleitungen metallische oder metallisierte Steckergehäuse.
- Setzen Sie in besonderen Anwendungsfällen spezielle EMV-Maßnahmen ein.
 - Erwägen Sie bei Induktivitäten den Einsatz von Löschiern.
 - Beachten Sie, dass bei Einsatz von Leuchtstofflampen sich diese negativ auf Signalleitungen auswirken können.
- Schaffen Sie ein einheitliches Bezugspotenzial und erden Sie nach Möglichkeit alle elektrischen Betriebsmittel.
 - Achten Sie auf den gezielten Einsatz der Erdungsmaßnahmen. Das Erden der Steuerung dient als Schutz- und Funktionsmaßnahme.
 - Verbinden Sie Anlagenteile und Schränke mit Ihrer SPS sternförmig mit dem Erde/Schutzleitersystem. Sie vermeiden so die Bildung von Erdschleifen.
 - Verlegen Sie bei 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. μA) übertragen werden.
 - Folienschirme (statische Schirme) verwendet werden.
- Benutzen Sie bei Datenleitungen für serielle Kopplungen immer metallische oder metallisierte Stecker. Befestigen Sie den Schirm der Datenleitung am Steckergehäuse. Schirm nicht auf den PIN 1 der Steckerleiste auflegen!
- Bei stationärem Betrieb ist es empfehlenswert, das geschirmte Kabel unterbrechungsfrei abzuisolieren und auf die Schirm-/Schutzleiterschiene aufzulegen.
- Benutzen Sie zur Befestigung der Schirmgeflechte Kabelschellen aus Metall. Die Schellen müssen den Schirm großflächig umschließen und guten Kontakt ausüben.
- Legen Sie den Schirm direkt nach Eintritt der Leitung in den Schrank auf eine Schirmschiene auf. Führen Sie den Schirm bis zu Ihrer SPS weiter, legen Sie ihn dort jedoch nicht erneut auf!



VORSICHT!

Bitte bei der Montage beachten!

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

Abhilfe: Potenzialausgleichsleitung.

2.10 Allgemeine Daten

Konformität und Approbation

Konformität		
CE	2014/35/EU	Niederspannungsrichtlinie
	2014/30/EU	EMV-Richtlinie
Approbation		
UL	-	Siehe Technische Daten
Sonstiges		
RoHS	2011/65/EU	Richtlinie zur Beschränkung der Verwendung bestimmter gefährlicher Stoffe in Elektro- und Elektronikgeräten

Personenschutz und Geräteschutz

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

Umgebungsbedingungen gemäß EN 61131-2

Klimatisch		
Lagerung /Transport	EN 60068-2-14	-25...+70°C
Betrieb		
Horizontaler Einbau hängend	EN 61131-2	0...+60°C
Horizontaler Einbau liegend	EN 61131-2	0...+55°C
Vertikaler Einbau	EN 61131-2	0...+50°C
Luftfeuchtigkeit	EN 60068-2-30	RH1 (ohne Betauung, relative Feuchte 10 ... 95%)
Verschmutzung	EN 61131-2	Verschmutzungsgrad 2

Umgebungsbedingungen gemäß EN 61131-2

Aufstellhöhe max.	-	2000m
Mechanisch		
Schwingung	EN 60068-2-6	1g, 9Hz ... 150Hz
Schock	EN 60068-2-27	15g, 11ms

Montagebedingungen

Einbauort	-	Im Schaltschrank
Einbaulage	-	Horizontal und vertikal

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

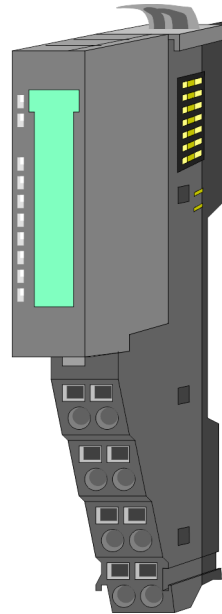
*) Aufgrund der energiereichen Einzelimpulse ist bei Surge eine angemessene externe Beschaltung mit Blitzschutzelementen wie z.B. Blitzstromableitern und Überspannungsableitern erforderlich.

3 Hardwarebeschreibung

3.1 Leistungsmerkmale

Eigenschaften

- RS422/485-Schnittstelle (potenzialgetrennt zum Rückwandbus)
- Übertragungsrate von 150Bit/s bis maximal 115,2kBit/s
- Serielle Busverbindung
 - Vollduplex (RS422 Vierdraht-Betrieb)
 - Halbduplex (RS485 Zweidraht-Betrieb)
- Protokolle
 - ASCII
 - STX/ETX
 - 3964(R)
 - Modbus (Master/Slave mit ASCII und RTU short & long) mit einer Telegrammlänge von 250Byte
- Bis zu 250 Telegramme (1024Byte Empfangs- bzw. Sendepuffer)
- Zeichenverzugszeit im ms Raster parametrierbar
- Parametrierung über Parameterdaten

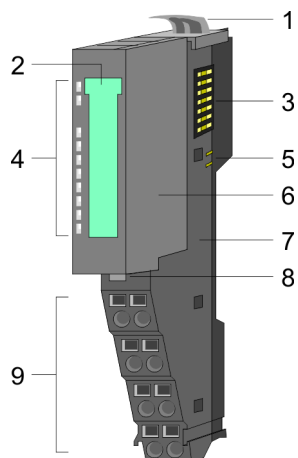


Bestelldaten

Typ	Bestellnummer	Beschreibung
CP 040 RS422/485	040-1CA00	Kommunikationsprozessor RS422/485, potentialgetrennt, ASCII, STX/ETX, 3964(R), Modbus-Master/Slave short/long

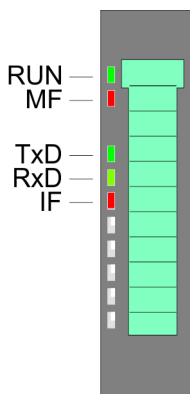
3.2 Aufbau

040-1CA00



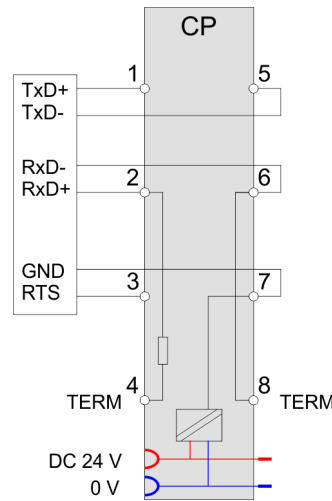
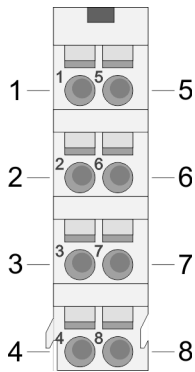
- 1 Verriegelungshebel Terminal-Modul
- 2 Beschriftungsstreifen
- 3 Rückwandbus
- 4 LED-Statusanzeige
- 5 DC 24V Leistungsversorgung
- 6 Elektronik-Modul
- 7 Terminal-Modul
- 8 Verriegelungshebel Elektronik-Modul
- 9 Anschlussklemmen

Statusanzeige



LED		Beschreibung
RUN	MF	
■ grün	■ rot	
■	□	Bus-Kommunikation ist OK Modul-Status ist OK
■	■	Bus-Kommunikation ist OK Modul-Status meldet Fehler
□	■	Bus-Kommunikation nicht möglich Modul-Status meldet Fehler
□	□	Fehler Busversorgungsspannung
X	▣ 2Hz	Konfigurationsfehler <i>☞ Kapitel 2.8 "Hilfe zur Fehlersuche - LEDs" auf Seite 29</i>
TxD	■ grün	Daten senden (transmit data)
RxD	■ grün	Daten empfangen (receive data)
IF	▣ 2Hz	Modbus: interner Fehler Andere Protokolle: Leitungsunterbrechung, Überlauf, Paritätsfehler oder Zeichenrahmenfehler
nicht relevant: X		

Anschlüsse

Für Drähte mit einem Querschnitt von 0,08mm² bis 1,5mm².

Pos.	Funktion	Typ	Beschreibung
1	TxD-P (B)	A	Sendedaten (RS422)
2	RxD-P (B) TxD/RxD-P (B)	E A/E	Empfangsdaten (RS422) Sende-/Empfangsdaten (RS485)
3	RTS	A	Request to send (RS485) RTS auf "1": CP sendebereit RTS auf "0": CP sendet nicht
4	TERM	E	Abschlusswiderstand *
5	TxD-N (A)	A	Sendedaten (RS422)
6	RxD-N (A) TxD/RxD-N (A)	E A/E	Empfangsdaten (RS422) Sende-/Empfangsdaten (RS485)
7	GND_ISO	A	Signal Nullbezugspunkt (isoliert)
8	TERM	E	Abschlusswiderstand *

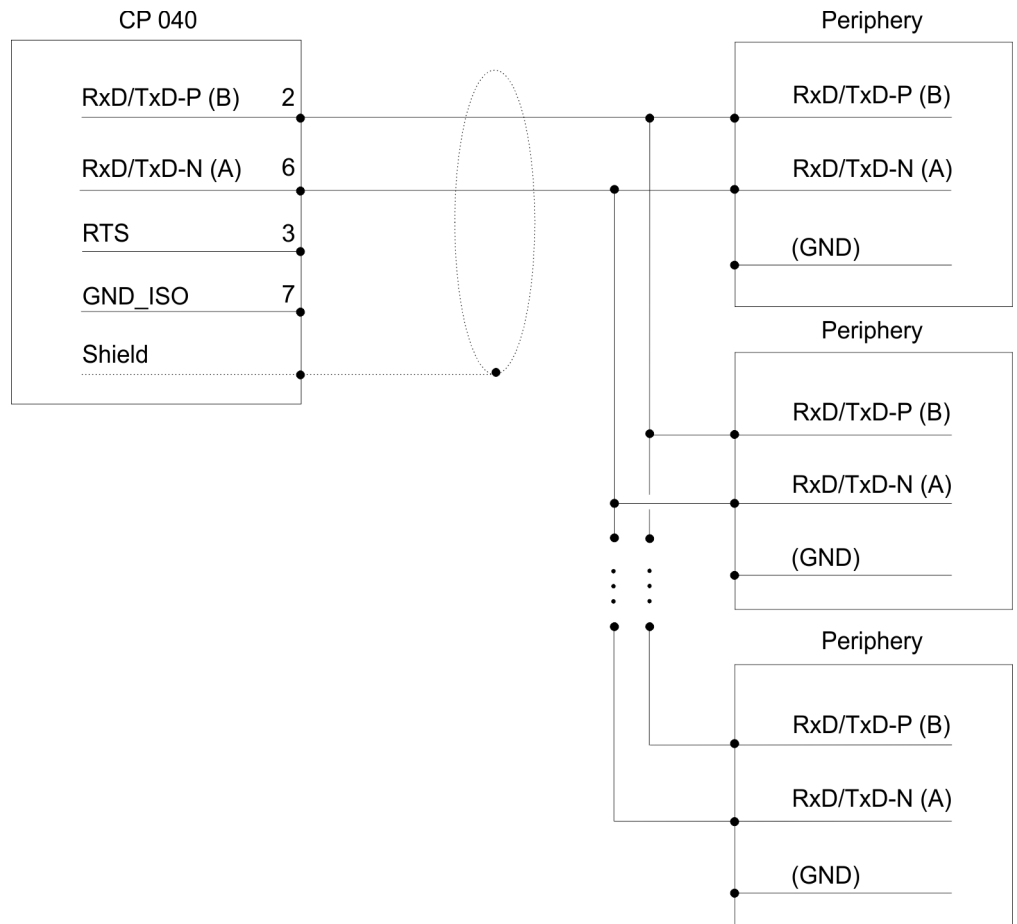
*) Eine Brücke zwischen den zwei TERM Eingängen aktiviert einen Abschlusswiderstand von 120Ω auf der Empfängerseite zwischen RxD-P (Pin 2) und RxD-N (Pin 6).

E: Eingang, A: Ausgang

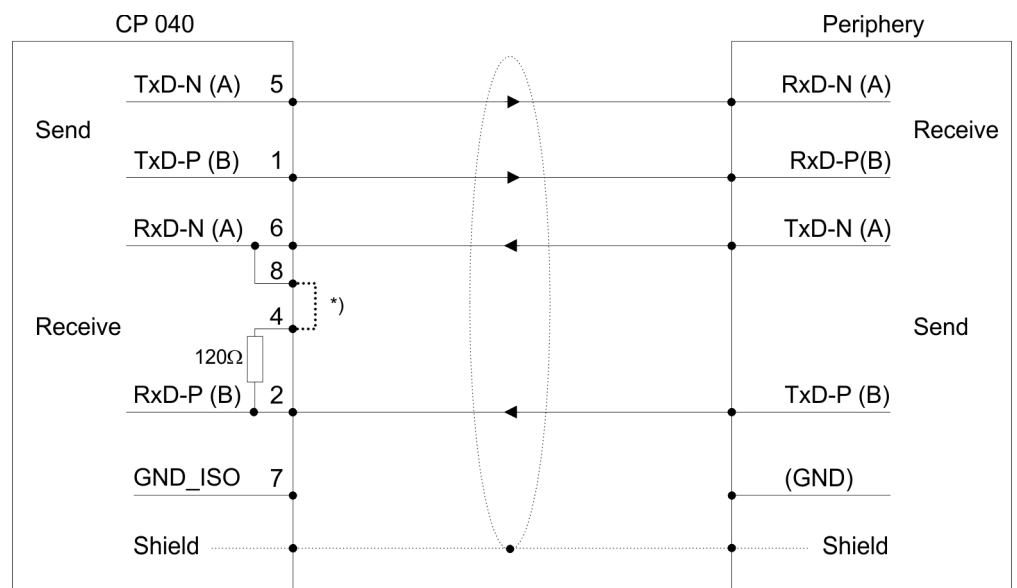
RS422/485-Schnittstelle

- Logische Zustände als Spannungsdifferenz zwischen 2 verdrehten Adern
- Serielle Busverbindung
 - Vollduplex (RS422 Vierdraht-Betrieb)
 - Halbduplex (RS485 Zweidraht-Betrieb)
- Leitungslänge: 250m bei 115,2kBit/s ... 1200m bei 19,2kBit/s
- Datenübertragungsrate: max. 115,2kBit/s

RS485- Verkabelung



RS422- Verkabelung

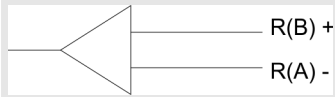
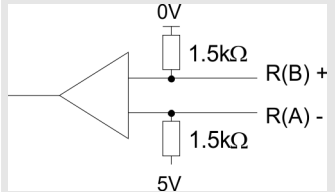
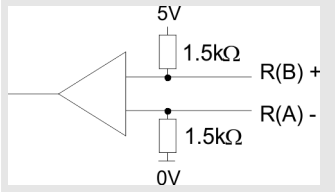


*) Eine Brücke zwischen den zwei TERM Eingängen aktiviert einen Abschlusswiderstand von 120Ω auf der Empfängerseite zwischen RxD-P (Pin 2) und RxD-N (Pin 6).

Definierte Ruhepegel über Parameter

Für einen reflexionsarmen Anschluss und die Drahtbruchererkennung im RS422/485-Betrieb können die Leitungen über Parameter mit definiertem Ruhepegel vorbelegt werden.

An der CP-Schnittstelle ist die Beschaltung des Empfängers folgendermaßen realisiert:

Parameter	Beschreibung	Beschaltung Empfänger
Keine (00h)	Keine Vorbelegung der Empfangsleitung. Diese Einstellung ist nur sinnvoll für busfähige Sondertreiber.	
Signal R(A) 5V (Breakerkennung) Signal R(B) 0V (01h)	Bei dieser Vorbelegung ist bei Vollduplex-Betrieb (RS422) Drahtbruchererkennung möglich.	
Signal R(A) 0V Signal R(B) 5V (02h)	Diese Vorbelegung entspricht dem Ruhezustand (kein Sender aktiv) bei Halbduplex-Betrieb unter RS485. Hier ist aber keine Drahtbruchererkennung möglich.	

3.3 Technische Daten

Artikelnr.	040-1CA00
Bezeichnung	CP 040 - Kommunikationsprozessor
Modulkennung	0E41 1700
Stromaufnahme/Verlustleistung	
Stromaufnahme aus Rückwandbus	125 mA
Stromaufnahme aus Lastspannung L+ (ohne Last)	10 mA
Verlustleistung	1 W
Status, Alarm, Diagnosen	
Statusanzeige	ja
Alarme	ja, parametrierbar
Prozessalarm	nein
Diagnosealarm	ja, parametrierbar
Diagnosefunktion	ja, parametrierbar
Diagnoseinformation auslesbar	möglich
Versorgungsspannungsanzeige	grüne LED
Sammelfehleranzeige	rote LED
Kanalfehleranzeige	rote LED
Point-to-Point Kommunikation	
PtP-Kommunikation	✓
Schnittstelle potentialgetrennt	✓
Schnittstelle RS232	-
Schnittstelle RS422	✓
Schnittstelle RS485	✓
Anschluss	Terminal-Modul
Übertragungsgeschwindigkeit, min.	150 bit/s
Übertragungsgeschwindigkeit, max.	115,2 kbit/s
Leitungslänge, max.	1200 m
Point-to-Point Protokolle	
Protokoll ASCII	✓
Protokoll STX/ETX	✓
Protokoll 3964(R)	✓
Protokoll RK512	-
Protokoll USS Master	-
Protokoll Modbus Master	✓
Protokoll Modbus Slave	✓
Spezielle Protokolle	-

Artikelnr.	040-1CA00
Datengrößen	
Eingangsbytes	8 / 20 / 60
Ausgangsbytes	8 / 20 / 60
Parameterbytes	23
Diagnosebytes	20
Gehäuse	
Material	PPE / PPE GF10
Befestigung	Profilschiene 35mm
Mechanische Daten	
Abmessungen (BxHxT)	12,9 mm x 109 mm x 76,5 mm
Gewicht Netto	59 g
Gewicht inklusive Zubehör	59 g
Gewicht Brutto	74 g
Umgebungsbedingungen	
Betriebstemperatur	0 °C bis 60 °C
Lagertemperatur	-25 °C bis 70 °C
Zertifizierungen	
Zertifizierung nach UL	ja
Zertifizierung nach KC	ja

3.3.1 Technische Daten Protokolle

ASCII	
Telegrammlänge	max. 1024 Byte
Baudrate	150, 300, 600, 1200, 1800, 2400, 4800, 7200, 9600, 14400, 19200, 38400, 57600, 76800, 109700, 115200 Baud
Zeichenverzugszeit ZVZ	0 ... 65535 in ms-Schritten (bei 0 wird 3-fache Zeichenzeit verwendet)
Flusskontrolle	keine, Hardware, XON/XOFF
Anzahl pufferbarer Telegramme	max. 250
Enderkennung eines Telegramms	nach Ablauf der Zeichenverzugszeit ZVZ
STX/ETX	
Telegrammlänge	max. 1024 Byte
Baudrate	150, 300, 600, 1200, 1800, 2400, 4800, 7200, 9600, 14400, 19200, 38400, 57600, 76800, 109700, 115200 Baud

Zeichenverzugszeit TMO	0 ... 65535 in ms-Schritten (bei 0 wird 3-fache Zeichenzeit verwendet)
Flusskontrolle	keine, Hardware, XON/XOFF
Anzahl pufferbarer Telegramme	max. 250
Endeerkennung eines Telegramms	durch parametriertes Endezeichen
Anzahl Startzeichen	0 ... 2 (Zeichen parametrierbar)
Anzahl Endezeichen	0 ... 2 (Zeichen parametrierbar)
3964, 3964R	
Telegrammlänge	max. 1024Byte
Baudrate	150, 300, 600, 1200, 1800, 2400, 4800, 7200, 9600, 14400, 19200, 38400, 57600, 76800, 109700, 115200 Baud
Blockprüfzeichen	nur 3964R
Priorität	low/high
Zeichenverzugszeit ZVZ	0 ... 255 in 20ms-Schritten (bei 0 wird 3-fache Zeichenzeit verwendet)
Quittungsverzugszeit QVZ	0 ... 255 in 20ms-Schritten (bei 0 wird 3-fache Zeichenzeit verwendet)
Anzahl Aufbauversuche	0 ... 255
Anzahl Übertragungsversuche	1 ... 255
Modbus	
Telegrammlänge	max. 258 Byte
Adressierbarer Bereich	je 1024 Byte
Baudrate	150, 300, 600, 1200, 1800, 2400, 4800, 7200, 9600, 14400, 19200, 38400, 57600, 76800, 109700, 115200 Baud
Modus	Master ASCII, Master RTU, Slave ASCII short, Slave RTU short, Slave ASCII long, Slave, RTU long
Adresse	1 ... 255
Wartezeit	automatisch, 1 ... 60000 ms

4 Einsatz

4.1 Schnelleinstieg

Übersicht

Der Kommunikationsprozessor 040-1CA00 für das System SLIO ermöglicht die serielle Prozessankopplung zu verschiedenen Ziel- oder Quellsystemen. Hierbei wird der CP als Peripherie-Modul betrieben und über den Rückwandbus mit Betriebsspannung versorgt.

Parameter

Zur Parametrierung können dem CP Parameterdaten übergeben werden, die je nach gewähltem Protokoll entsprechend belegt sind. Näheres zur Belegung der Parameter finden Sie im Teil "Serielle Kommunikationsprotokolle". ↪ *Kapitel 5.1 "Übersicht" auf Seite 58*

Protokolle

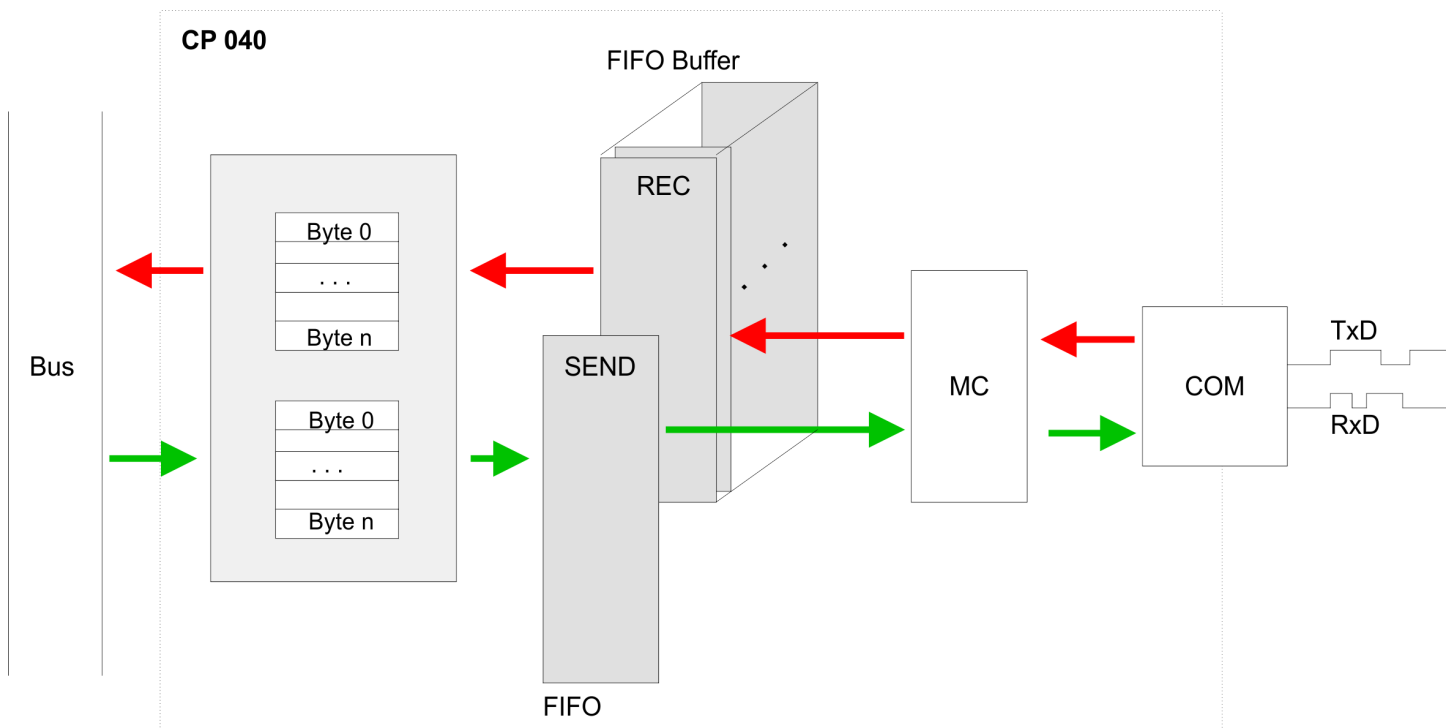
- ASCII
- STX/ETX
- 3964(R)
- Modbus (Master, Slave)

Kommunikation

Beim Senden werden Daten, welche von einem übergeordneten System über den Rückwandbus in den entsprechenden Ausgabe-Bereich geschrieben werden, in den Sendepuffer geschrieben und von dort über die Schnittstelle ausgegeben.

Empfängt der Kommunikationsprozessor Daten über die Schnittstelle, werden diese Daten in einem Ringpuffer abgelegt und über den Rückwandbus in den Eingabebereich des übergeordneten Systems eingetragen.

Bitte beachten Sie, dass die Größe des E/A-Bereichs und damit auch des Telegramms am Rückwandbus vom übergeordneten System abhängt. Auf den nachfolgenden Seiten sind der E/A-Bereich und die Kommunikation über den Rückwandbus näher beschrieben.



4.2 Ein-/Ausgabe-Bereich

Übersicht

Abhängig vom übergeordneten System belegt der CP für Ein- und Ausgabe jeweils folgende Anzahl an Bytes im Adress-Bereich.

-
- PROFIBUS: 8Byte, 20Byte oder 60Byte wählbar
- PROFINET: 20Byte oder 60Byte wählbar
- CANopen: 8Byte
- EtherCAT: 60Byte
- DeviceNET: 60Byte
- ModbusTCP: 60Byte

Bei CPU, PROFIBUS und PROFINET wird der Ein- bzw. Ausgabebereich im entsprechenden Adressbereich eingeblendet mit $n = 8, 20$ oder 60 .

IX - Index für Zugriff über CANopen. Mit $s =$ Subindex adressieren Sie das entsprechende Byte.

SX - Subindex für Zugriff über EtherCAT mit Index $6000h/7000h +$ EtherCAT-Slot

Näheres hierzu finden Sie im Handbuch zu Ihrem Bus-Koppler.

Eingabebereich

Adr.	Name	Bytes	Funktion	IX = 5450h	SX
+0	CP_IN_STS	1	Status-Byte	$s = 1$	01h
+1	CP_IN_1	1	Eingabe-Byte 1	$s = 2$	02h
+2	CP_IN_2	1	Eingabe-Byte 2	$s = 3$	03h
...
+n-1	CP_IN_n-1	1	Eingabe-Byte n-1	$s = m$	mh

CP_IN_STS

Dieser Parameter beinhaltet Informationen über die Fragmentierung der Daten im Empfangspuffer.

CP_IN_x

Der Inhalt dieser Daten richtet sich nach dem Aufbau der Daten im Empfangspuffer. Nähere Informationen hierzu finden Sie auf den Folgeseiten.

Ausgabebereich

Adr.	Name	Bytes	Funktion	IX = 5650h	SX
+0	CP_OUT_CTRL	1	Control-Byte	$s = 1$	01h
+1	CP_OUT_1	1	Ausgabe-Byte 1	$s = 2$	02h
+2	CP_OUT_2	1	Ausgabe-Byte 2	$s = 3$	03h
...
+n-1	CP_OUT_n-1	1	Ausgabe-Byte n-1	$s = m$	mh

CP_OUT_CTRL

Hier können Sie mittels entsprechender Kommandos die Datenübertragung steuern.

CP_OUT_x

Der Inhalt dieser Daten richtet sich nach dem Aufbau der Daten im Sendepuffer. Nähere Informationen hierzu finden Sie auf den Folgeseiten.

4.3 Prinzip der Rückwandbus-Kommunikation

4.3.1 Daten senden

Beim Senden sind vom übergeordneten System die auszugebenden Daten in den Ausgabebereich einzutragen und mit dem *Control-Byte* an den CP zu übergeben.

Der CP reagiert bei jedem Telegramm mit einer Quittierung, indem er Bit 3...0 von Byte 0 des Ausgabebereichs in Bit 7...4 von Byte 0 des Eingabebereichs kopiert oder über dieses Byte eine entsprechende *Statusmeldung* zurückschickt.

Abhängig von der Länge der zu übertragenden Daten ist das Telegramm in einem Fragment oder mit mehreren Fragmenten an den CP zu übermitteln. Bei der fragmentierten Übertragung wird jedes Fragment vom CP quittiert.

Prinzip der Übertragung ohne Fragmentierung

Übergeordnetes System			CP	
Byte	Funktion		Byte	Funktion
0	Control-Byte	▶		
1	Telegramm-Info-Byte			
2	Länge High-Byte			
3	Länge Low-Byte			
4...n-1	Nutzdaten Byte 0...n-5			
		◀	0	Quittierung / Status

mit n = Anzahl der belegten Bytes im Adressbereich (IO-Size)

Control-Byte

Bit 3...0	<ul style="list-style-type: none"> ■ 8h: Leerlauf - keine Daten vorhanden ■ Ah: Starte Übertragung ohne Fragmentierung ■ Bh: Führe einen Reset auf dem CP aus
Bit 7...4	Reserviert für den Empfang

Telegramm-Info-Byte

Beim Sendevorgang 00h (fix).

Länge

Länge der Nutzdaten für die serielle Kommunikation in Byte.

Nutzdaten

Geben Sie hier die Nutzdaten für die serielle Kommunikation an.

Quittierung Status

Bit 3...0	Reserviert für den Empfang
Bit 7...4	<ul style="list-style-type: none"> ■ 8h: Quittierung: Leerlauf ■ Ah: Quittierung: Daten ohne Fragmentierung erhalten ■ Ch: Status: Reset wurde auf CP durchgeführt ■ Dh: Status: Die angegebene Länge ist ungültig ■ Eh: Status: Fehler CP-Kommunikation <ul style="list-style-type: none"> – Partner antwortet nicht

Prinzip der Übertragung mit Fragmentierung

Bei der fragmentierten Übertragung werden mit dem 1. Telegramm (Header) die Anzahl der Nutzdaten und schon ein Teil der Nutzdaten übermittelt. Danach folgen die Fragment-Telegramme. Der CP reagiert bei jedem Telegramm mit einer Quittierung, indem er Bit 3...0 von Byte 0 des Ausgabebereichs in Bit 7...4 von Byte 0 des Eingabebereichs kopiert oder über dieses Byte eine entsprechende *Statusmeldung* zurückschickt.

Ablauf

- Schreibe 1. Telegramm
- Schreibe Fragmente
- Schreibe letztes Fragment

Berechnung der Fragmentanzahl

$$Fragmentanzahl = \frac{Länge + 3}{IO_Size - 1}$$

Schreibe 1. Telegramm (Header)

Übergeordnetes System			CP	
Byte	Funktion		Byte	Funktion
0	Control-Byte	▶		
1	Telegramm-Info-Byte			
2	Länge High-Byte			
3	Länge Low-Byte			
4...n-1	Nutzdaten Byte 0...n-5			
		◀	0	Quittierung / Status

mit n = Anzahl der belegten Bytes im Adressbereich (IO-Size)

Control-Byte

Bit 3...0	<ul style="list-style-type: none"> ■ 8h: Leerlauf - keine Daten vorhanden ■ 9h: Starte fragmentierte Übertragung ■ Ah: Übertrage letztes Fragment ■ Bh: Führe einen Reset auf dem CP aus
Bit 7...4	Reserviert für den Empfang

Telegramm-Info-Byte

Beim Sendevorgang 00h (fix).

Länge Länge der Nutzdaten für die serielle Kommunikation in Byte.

Nutzdaten Geben Sie hier die Nutzdaten für die serielle Kommunikation an.

Quittierung Status

Bit 3...0	Reserviert für den Empfang
Bit 7...4	<ul style="list-style-type: none"> ■ 8h: Quittierung: Leerlauf ■ 9h: Quittierung: Fragmentierte Übertragung gestartet ■ Ah: Quittierung: Daten ohne Fragmentierung erhalten ■ Ch: Status: Reset wurde auf CP durchgeführt ■ Dh: Status: Die angegebene Länge ist ungültig ■ Eh: Status: Fehler CP-Kommunikation <ul style="list-style-type: none"> – Partner antwortet nicht

Schreibe Fragmente

Übergeordnetes System			CP	
Byte	Funktion		Byte	Funktion
0	Control-Byte	▶		
1...n-1	Nutzdaten			
		◀	0	Quittierung / Status

mit n = Anzahl der belegten Bytes im Adressbereich (IO-Size)

Control-Byte

Bit 3...0	<ul style="list-style-type: none"> ■ 0h...7h: Fragment-Nummer ■ 8h: Leerlauf - keine Daten vorhanden ■ Bh: Führe einen Reset auf dem CP aus
Bit 7...4	Reserviert für den Empfang

Nutzdaten Geben Sie hier die Nutzdaten für die serielle Kommunikation an.

Quittierung Status

Bit 3...0	Reserviert für den Empfang
Bit 7...4	<ul style="list-style-type: none"> ■ 0h...7h: Quittierung: Fragment-Nummer ■ 8h: Quittierung: Leerlauf ■ Ch: Status: Reset wurde auf CP durchgeführt ■ Dh: Status: Die angegebene Länge ist ungültig ■ Eh: Status: Fehler CP-Kommunikation <ul style="list-style-type: none"> – Partner antwortet nicht

Schreibe letztes Fragment

Übergeordnetes System			CP	
Byte	Funktion		Byte	Funktion
0	Control-Byte	▶		
1...n-1	Nutzdaten			

Übergeordnetes System			CP	
Byte	Funktion		Byte	Funktion
		◀	0	Quittierung / Status

mit n = Anzahl der belegten Bytes im Adressbereich (IO-Size)

Control-Byte

Bit 3...0	<ul style="list-style-type: none"> ■ 8h: Leerlauf - keine Daten vorhanden ■ Ah: Übertrage letztes Fragment ■ Bh: Führe einen Reset auf dem CP aus
Bit 7...4	Reserviert für den Empfang

Nutzdaten

Geben Sie hier die Nutzdaten für die serielle Kommunikation an.

Quittierung Status

Bit 3...0	Reserviert für den Empfang
Bit 7...4	<ul style="list-style-type: none"> ■ 8h: Quittierung: Leerlauf ■ Ah: Quittierung: Letztes Fragment erhalten ■ Ch: Status: Reset wurde auf CP durchgeführt ■ Dh: Status: Die angegebene Länge ist ungültig ■ Eh: Status: Fehler CP-Kommunikation <ul style="list-style-type: none"> – Partner antwortet nicht

4.3.2 Daten empfangen

Beim Empfangen werden die empfangenen Daten vom CP automatisch im Eingabebereich des übergeordneten Systems eingetragen.

Abhängig von der Länge der empfangenen Daten wird das Telegramm in einem Fragment oder mit mehreren Fragmenten an das übergeordnete System übermittelt.

Die fragmentierte Übertragung starten Sie, indem Sie Bit 3...0 von Byte 0 des Eingabebereichs in Bit 7...4 von Byte 0 des Ausgabebereichs kopieren. Eventuelle Fehler bei der Übertragung finden Sie in RetVal.

Prinzip der Übertragung ohne Fragmentierung

Übergeordnetes System			CP	
Byte	Funktion		Byte	Funktion
		▶	0	Info-Byte
			1	Telegramm-Info-Byte
			2	Länge High-Byte
			3	Länge Low-Byte
			[4]	Offset High-Byte
			[5]	Offset Low-Byte
			6	RetVal High-Byte
			7	RetVal Low-Byte

Übergeordnetes System			CP	
Byte	Funktion		Byte	Funktion
			8...n-1	Nutzdaten
0	Quittierung	◀	0	

mit n = Anzahl der belegten Bytes im Adressbereich (IO-Size)

Info-Byte

Bit 3...0	<ul style="list-style-type: none"> ■ 8h: Leerlauf - keine Daten vorhanden ■ 9h: Daten werden fragmentiert übertragen ■ Ah: Daten wurden ohne Fragmentierung übertragen
Bit 7...4	Reserviert für Senden

Telegramm-Info-Byte

00h:	Das Telegramm beinhaltet keine zusätzlichen Offset-Angaben.
04h:	Das Telegramm beinhaltet zusätzliche Offset-Angaben, welche als Wort der <i>Länge</i> nachgestellt sind. Über die Offset-Angaben wird die Position der Nutzdaten im Eingabebereich bestimmt.

Länge

Länge der Nutzdaten der seriellen Kommunikation in Byte zusätzlich 2Byte für RetVal.
Länge 2Byte: nur RetVal ohne Nutzdaten.

Offset

Sofern das *Telegramm-Info-Byte* den Wert 04h hat, wird zusätzlich ein *Offset* eingetragen. Ansonsten gibt es Offset im Telegramm nicht.

RetVal

0517h:	Ungültige <i>Länge</i> (<i>Länge</i> = 0 oder <i>Länge</i> > 1024)
080Ah:	Ein freier Empfangspuffer ist nicht vorhanden.
080Ch:	Fehlerhaftes Zeichen empfangen (Zeichenrahmen- oder Paritätsfehler)

Nutzdaten

Hier finden Sie die empfangenen Nutzdaten der seriellen Kommunikation.

Quittierung

Nachdem Sie die Daten in Ihrem übergeordneten System entsprechend verarbeitet haben, müssen Sie dem CP den Empfang quittieren (auch RetVal-Telegramme ohne Nutzdaten). Erst dann kann dieser neue Empfangsdaten bereitstellen.

Bit 3...0	Reserviert für Senden
Bit 7...4	8h: Quittierung: Leerlauf Ah: Quittierung: Eingabebereich frei für neue Daten Bh: Kommando: Führe einen Reset auf dem CP aus

Prinzip der Übertragung mit Fragmentierung

Übergeordnetes System			CP	
Byte	Funktion		Byte	Funktion
		◀	0	Info-Byte
			1	Telegramm-Info-Byte
			2	Länge High-Byte
			3	Länge Low-Byte
			[4]	Offset High-Byte
			[5]	Offset Low-Byte
			6...n-1	Nutzdaten

mit n = Anzahl der belegten Bytes im Adressbereich (IO-Size)

Nachdem Sie die Daten in Ihrem übergeordneten System entsprechend verarbeitet haben, müssen Sie dem CP den Empfang quittieren, indem Sie Bit 3...0 von Byte 0 des Eingabebereichs in Bit 7...4 von Byte 0 des Ausgabebereichs kopieren. Erst dann kann der CP das nächste Fragment bereitstellen.



Berechnung der Fragmentanzahl

$$Fragmentanzahl = \frac{Länge + 7}{IO_Size - 1}$$

Info-Byte

Bit 3...0	8h: Leerlauf - keine Daten vorhanden 9h: Daten wurden fragmentiert übertragen Ah: Daten wurden ohne Fragmentierung übermittelt
Bit 7...4	Reserviert für Senden

Telegramm-Info-Byte

00h:	Das Telegramm beinhaltet keine zusätzlichen Offset-Angaben
04h:	Das Telegramm beinhaltet zusätzliche Offset-Angaben, welche als Wort der <i>Länge</i> nachgestellt sind. Über die Offset-Angaben wird die Position der Nutzdaten im Eingabebereich bestimmt.

Länge

Länge der Nutzdaten in Byte zusätzlich 2 Byte für RetVal.

Offset

Sofern das *Telegramm-Info-Byte* den Wert 04h hat, wird zusätzlich ein Offset eingetragen. Ansonsten befindet sich hier RetVal. Berechnung des Offset bei fragmentierter Übertragung:

$$\text{Daten_Offset} = (\text{Fragmentzähler} + 1) \times (\text{IO_Size} - 1) - 7 + \text{Offset}$$

- Daten_Offset:
 - Offset der Daten im Eingabebereich
- Fragmentzähler:
 - Absolute Fragmentanzahl
- IO_Size:
 - Anzahl der belegten Bytes im Adressbereich
- Offset:
 - Offset-Wert im Telegramm

Nutzdaten

Hier finden Sie die empfangenen Nutzdaten der seriellen Kommunikation.

Quittierung

Bit 3...0	Reserviert für Senden
Bit 7...4	<ul style="list-style-type: none"> ■ 8h: Quittierung: Leerlauf ■ Ah: Quittierung: Eingabebereich frei für neue Daten ■ Bh: Kommando: Führe einen Reset auf dem CP aus

4.3.3 Beispiele**Daten senden ohne Fragmentierung**

IO-Size = 60Byte, Länge = 40Byte

Übergeordnetes System			CP	
Byte	Funktion		Byte	Funktion
0	0Ah Kommando	▶		
1	00h Telegramm-Info			
2	00h Länge High-Byte			
3	28h Länge Low-Byte			
4...43	Nutzdaten Byte 0...39			
44...59	wird nicht verwendet			
		◀	0	A0h Quittierung

Daten senden mit Fragmentierung

IO-Size = 16Byte, Länge = 50Byte

Header Übergeordnetes System			CP	
Byte	Funktion		Byte	Funktion
0	09h Kommando	▶		
1	00h Telegramm-Info			
2	00h Länge High-Byte			

Header Übergeordnetes System			CP	
Byte	Funktion		Byte	Funktion
3	28h Länge Low-Byte			
4...15	Nutzdaten Byte 0...11			
		◀	0	90h Quittierung

1. Fragment Übergeordnetes System			CP	
Byte	Funktion		Byte	Funktion
0	00h Fragment	▶		
1...15	Nutzdaten Byte 12...26			
		◀	0	00h Quittierung

2. Fragment Übergeordnetes System			CP	
Byte	Funktion		Byte	Funktion
0	01h Fragment	▶		
1...15	Nutzdaten Byte 27...41			
		◀	0	10h Quittierung

Letztes Fragment Übergeordnetes System			CP	
Byte	Funktion		Byte	Funktion
0	0Ah Kommando	▶		
1...8	Nutzdaten Byte 42...49			
11...15	wird nicht verwendet			
		◀	0	A0h Quittierung

Daten empfangen *ohne Fragmentierung*

IO-Size = 60Byte, Länge = 40Byte

Übergeordnetes System			CP	
Byte	Funktion		Byte	Funktion
		◀	0	0Ah Fragment-Info
			1	00h Telegramm-Info-Byte

Übergeordnetes System			CP	
Byte	Funktion		Byte	Funktion
			2	00h Länge High-Byte
			3	2Ah Länge Low-Byte + 2Byte
			4	00h Return Value High-Byte
			5	00h Return Value Low-Byte
			6...45	Nutzdaten Byte 0...39
			46...59	wird nicht verwendet
0	A0h Quittierung	▶	0	

**Daten empfangen mit
Fragmentierung**

IO-Size = 16Byte, Länge = 40Byte

Header Übergeordnetes System			CP	
Byte	Funktion		Byte	Funktion
		◀	0	09h Fragment-Info
			1	00h Telegramm-Info-Byte
			2	00h Länge High-Byte
			3	2Ah Länge Low-Byte + 2Byte
			4	00h Return Value High-Byte
			5	00h Return Value Low-Byte
			6...15	Nutzdaten Byte 0...9
0	90h Quittierung	▶	0	

1. Fragment Übergeordnetes System			CP	
Byte	Funktion		Byte	Funktion
		◀	0	00h Fragment
			1...15	Nutzdaten Byte 10...24
0	00h Quittierung	▶	0	

Letztes Fragment Übergeordnetes System			CP	
Byte	Funktion		Byte	Funktion
		◀	0	0Ah Fragment-Info
			1...15	Nutzdaten Byte 25...39
0	A0h Quittierung	▶	0	

4.4 Kommunikation über Hantierungsbausteine

Kommunikation

Für die Verarbeitung der Verbindungsaufträge auf SPS-Seite ist ein Anwenderprogramm in der CPU erforderlich.



Beim Einsatz einer System SLIO CPU verwenden Sie zur Kommunikation FB 65 SEND_RECV.

Zur Kommunikation zwischen CPU, CP und einem Kommunikationspartner kommen folgende VIPA-spezifischen Bausteine zum Einsatz:

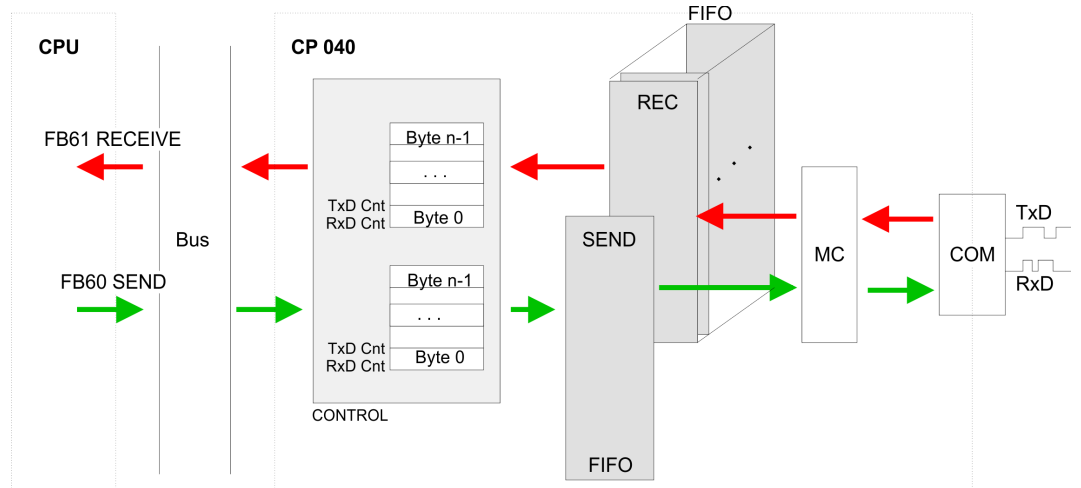
Baustein	Symbol	Kommentar
FB 60	SEND	Baustein für das Senden von Daten an einen Kommunikationspartner.
FB 61	RECEIVE	Baustein für den Empfang von Daten von einem Kommunikationspartner.
FB 65	SEND_RECV	Baustein für Senden und Empfangen von Daten zu einem Kommunikationspartner (z.B. System SLIO CPU).

4.4.1 Übersicht

Kommunikationsprinzip

- Durch zyklischen Aufruf von FB 60 SEND und FB 61 RECEIVE bzw. FB 65 CP040_COM können Sie mit dem CP zyklisch Daten senden und empfangen.
- Auf dem CP erfolgt die Umsetzung der Übertragungsprotokolle zum Kommunikationspartner, welche Sie mittels der Hardwarekonfiguration parametrieren können.
- Ein zu sendendes Telegramm wird in der CPU, abhängig von der IO-Size, in Blöcke unterteilt und über den Datenkanal an den CP übergeben. Im CP werden diese Blöcke im Sendepuffer zusammengesetzt und bei Vollständigkeit des Telegramms über die serielle Schnittstelle gesendet.
- Der Austausch von empfangenen Telegrammen über den Rückwandbus erfolgt asynchron.
- Ist ein komplettes Telegramm über die serielle Schnittstelle eingetroffen, so wird dies in einem 1024Byte großen Ringpuffer abgelegt. Aus der Länge des noch freien Ringpuffers ergibt sich die max. Länge eines Telegramms.
- Je nach Parametrierung können bis zu 250 Telegramme gepuffert werden, wobei deren Gesamtlänge 1024 nicht überschreiten darf.
- Ist der Puffer voll, werden neu ankommende Telegramme verworfen.
- Ein komplettes Telegramm wird in Blöcke, abhängig von der parametrierten IO-Size unterteilt und an den Rückwandbus übergeben.

- Das Zusammensetzen der Datenblöcke hat in der CPU zu erfolgen.
- Da der Datenaustausch über den Rückwandbus asynchron abläuft, wird ein Software-Handshake zwischen dem CP und der CPU eingesetzt. Hierzu besitzen beide Handierungsbausteine den gemeinsamen Parameter CONTROL. Für diesen Parameter ist das selbe Merker-Byte zu verwenden.



FIFO Ringpuffer max. 250 Telegramme 1024Byte
 CONTROL Software-Handshake über CONTROL-Baustein



Zum Erkennen eines Signalwechsels ist eine Mindestimpulsdauer erforderlich. Ausschlaggebend sind die CPU-Zykluszeit, die Aktualisierungszeit auf dem CP und die Reaktionszeit des Kommunikationspartners.

4.4.2 VIPA Lib



Näheres zum Einsatz dieser Bausteine finden Sie im Handbuch "Serial Communication - SW90GS0MA" auf www.vipa.com im "Service/Support"-Bereich unter "Handbücher → VIPA Lib".

4.5 Diagnosedaten

Übersicht

Sie haben die Möglichkeit, einen Diagnosealarm für den CP zu aktivieren. Mit dem Auslösen eines Diagnosealarms werden vom Modul Diagnosedaten für Diagnose_{kommend} bereitgestellt. Sobald die Gründe für das Auslösen eines Diagnosealarms nicht mehr gegeben sind, erhalten Sie automatisch einen Diagnosealarm_{gehend}.

Innerhalb dieses Zeitraums (1. Diagnosealarm_{kommend} bis letzter Diagnosealarm_{gehend}) leuchtet die MF-LED des Moduls.

DS - Datensatz für Zugriff über CPU, PROFIBUS und PROFINET. Der Zugriff erfolgt über DS 01h. Zusätzlich können Sie über DS 00h auf die ersten 4 Byte zugreifen.

IX - Index für Zugriff über CANopen. Der Zugriff erfolgt über IX 2F01h. Zusätzlich können Sie über IX 2F00h auf die ersten 4 Byte zugreifen.

SX - Subindex für Zugriff über EtherCAT mit Index 5005h.

Näheres hierzu finden Sie im Handbuch zu Ihrem Bus-Koppler.

Name	Bytes	Funktion	Default	DS	IX	SX
ERR_A	1	Diagnose	00h	01h	2F01h	02h
MODTYP	1	Modulinformation	1Ch			03h
ERR_C	1	reserviert	00h			04h
ERR_D	1	Diagnose	00h			05h
CHTYP	1	Kanaltyp	60h			06h
NUMBIT	1	Anzahl Diagnosebits pro Kanal	08h			07h
NUMCH	1	Anzahl Kanäle des Moduls	01h			08h
CHERR	1	Kanalfehler	01h			09h
CH0ERR	1	Kanalspezifischer Fehler	01h			0Ah
CH1ERR...CH7ERR	7	reserviert	00h			0Bh ... 11h
DIAG_US	4	µs-Ticker	00h			13h

ERR_A Diagnose

Byte	Bit 7 ... 0
0	<ul style="list-style-type: none"> ■ Bit 0: gesetzt bei Baugruppenstörung ■ Bit 1: gesetzt bei Fehler intern ■ Bit 2: gesetzt bei Fehler extern (Kabelbruch nur bei RS422) ■ Bit 3: reserviert ■ Bit 4: gesetzt bei fehlender externer Versorgungsspannung ■ Bit 5, 6: reserviert ■ Bit 7: gesetzt bei Parametrierfehler

Diagnosedaten

MODTYP Modulinformation

Byte	Bit 7 ... 0
0	<ul style="list-style-type: none"> ■ Bit 3 ... 0: Modulklasse <ul style="list-style-type: none"> – 1100b: CP ■ Bit 4: gesetzt bei Kanalinformation vorhanden ■ Bit 7 ... 5: reserviert

ERR_D Diagnose

Byte	Bit 7 ... 0
0	<ul style="list-style-type: none"> ■ Bit 2 ... 0: reserviert ■ Bit 3: gesetzt bei internem Diagnosepufferüberlauf ■ Bit 4: gesetzt bei internem Kommunikationsfehler ■ Bit 7 ... 5: reserviert

CHTYP Kanaltyp

Byte	Bit 7 ... 0
0	<ul style="list-style-type: none"> ■ Bit 6 ... 0: Kanaltyp <ul style="list-style-type: none"> – 60h: Kommunikationsprozessor ■ Bit 7: reserviert

NUMBIT Diagnosebits

Byte	Bit 7 ... 0
0	Anzahl der Diagnosebits des Moduls pro Kanal (hier 08h)

NUMCH Kanäle

Byte	Bit 7 ... 0
0	Anzahl der Kanäle eines Moduls (hier 01h)

CHERR Kanalfehler

Byte	Bit 7 ... 0
0	<ul style="list-style-type: none"> ■ Bit 0: gesetzt bei Fehler Kanalgruppe 0 ■ Bit 7 ... 1: reserviert

CH0ERR

Byte	Bit 7 ... 0
0	<ul style="list-style-type: none"> ■ Bit 3 ... 0: reserviert ■ Bit 4: Drahtbruch (nur bei RS422 möglich) ■ Bit 7 ... 5: reserviert

CH1ERR ... CH7ERR

Byte	Bit 7 ... 0
0	Bit 7 ... 0: reserviert

DIAG_US μ s-Ticker

Byte	Bit 7 ... 0
0...3	Wert des μ s-Ticker bei Auftreten der Diagnose

 μ s-Ticker

Im SLIO-Modul befindet sich ein 32-Bit Timer (μ s-Ticker), welcher mit NetzEIN gestartet wird und nach $2^{32}-1\mu$ s wieder bei 0 beginnt.

5 Serielle Kommunikationsprotokolle

5.1 Übersicht

Serielle Übertragung eines Zeichens

Die Punkt-zu-Punkt-Kopplung zwischen zwei Kommunikationspartnern ist die einfachste Form des Informationsaustauschs. Hierbei bildet der CP die Schnittstelle zwischen einem übergeordneten System und einem seriell angebundenen Kommunikationspartner. Die Datenübertragung erfolgt seriell. Bei der seriellen Datenübertragung werden die einzelnen Bits eines Bytes einer zu übertragenden Information in einer festgelegten Reihenfolge nacheinander übertragen.

Zeichenrahmen

Beim bidirektionalen Datenverkehr wird zwischen *Halbduplex*- und *Vollduplex*-Betrieb unterschieden. Im *Halbduplex*-Betrieb werden zu einem Zeitpunkt Daten entweder gesendet oder empfangen. Ein gleichzeitiger Datenaustausch kann nur im *Vollduplex*-Betrieb erfolgen. Jedem zu übertragenden Zeichen geht ein Synchronisierimpuls als *Startbit* voraus. Das Ende des Zeichentransfers bildet das *Stopbit*. Neben Start- und Stopbit sind weitere parametrierbare Vereinbarungen zwischen den Kommunikationspartnern für eine serielle Datenübertragung erforderlich.

Dieser Zeichenrahmen besteht aus folgenden Elementen:

- Übertragungsgeschwindigkeit (Baudrate)
- Zeichen- und Quittungsverzugszeit
- Parität
- Anzahl Datenbits
- Anzahl Stopbits

Protokolle

Der CP wickelt die serielle Datenübertragung selbständig ab. Hierzu ist der CP mit Treiber für die entsprechenden Protokolle ausgestattet.

Nachfolgend sind folgende Protokolle beschrieben:

- ASCII
- STX/ETX
- 3964(R)
- Modbus (Master, Slave)

5.2 ASCII

5.2.1 Grundlagen ASCII

Funktionsweise

Die Datenkommunikation via ASCII ist eine einfache Form des Datenaustauschs und kann mit einer Multicast/Broadcast-Funktion verglichen werden. Die logische Trennung der Telegramme erfolgt über die Zeichenverzugszeit (ZVZ). Innerhalb dieser Zeit muss der Sender sein Telegramm an den Empfänger geschickt haben. Ein Telegramm wird erst dann an das übergeordnete System weitergereicht, wenn dieses vollständig empfangen wurde. Solange "Zeit nach Auftrag" (ZNA) nicht abgelaufen ist, wird kein neuer Sendeauftrag angenommen. Mit diesen beiden Zeitangaben kann eine einfache serielle Kommunikation aufgebaut werden. Da bei ASCII-Übertragung neben der Verwendung des Paritätsbit keine weiteren Maßnahmen zur Datensicherung erfolgen, ist der Datentransfer zwar sehr effizient, aber nicht gesichert. Mit der Parität wird das Kippen eines Bits in einem Zeichen abgesichert. Kippen mehrere Bits eines Zeichens, kann dieser Fehler nicht mehr erkannt werden.

5.2.2 Parametrierdaten bei ASCII

Parameter

DS - Datensatz für Zugriff über CPU, PROFIBUS und PROFINET

IX - Index für Zugriff über CANopen

SX - Subindex für Zugriff über EtherCAT mit Index 3100h + EtherCAT-Slot

Näheres hierzu finden Sie im Handbuch zu Ihrem Bus-Koppler.

Name	Bytes	Funktion	Default	DS	IX	SX
PII_L	1	Länge Prozessabbild Eingabedaten ¹⁾	2)	02h	3100h	01h
PIQ_L	1	Länge Prozessabbild Ausgabedaten ¹⁾	2)	02h	3101h	02h
DIAG_EN	1	Diagnosealarm ¹⁾	00h	00h	3102h	03h
BAUD	1	Baudrate	00h	80h	3103h	04h
PROTOCOL	1	Protokoll	01h	80h	3104h	05h
OPTION3	1	Zeichenrahmen	13h	80h	3105h	06h
OPTION4, 5	2	ZNA 0 ... 65535 (in ms)	0	80h	3106h ... 3107h	07h ... 08h
OPTION6, 7	2	ZVZ 0 ... 65535 (in ms)	250	80h	3108h ... 3109h	09h ... 0Ah
OPTION8	1	Anz. Receivebuffer	1	80h	310Ah	0Bh
OPTION9...14	6	reserviert	00h	80h	310Bh ... 3110h	0Ch ... 11h
OPTION15	1	Betriebsart	00h	80h	3111h	12h
OPTION16	1	Leistungsbelegung	00h	80h	3112h	13h

1) Diesen Datensatz dürfen Sie ausschließlich im STOP-Zustand übertragen.

2) Wert hängt vom übergeordneten System ab.

DIAG_EN: Diagnosealarm Hier aktivieren bzw. deaktivieren Sie die Diagnosefunktion.

Byte	Bit 7 ... 0
0	<ul style="list-style-type: none"> ■ Diagnosealarm <ul style="list-style-type: none"> – 00h: sperren – 40h: freigeben

■ Default: 00h

BAUD: Übertragungsrate Geschwindigkeit der Datenübertragung in Bit/s (Baud). Sie haben folgende Einstellbereiche; andere Werte sind nicht zulässig.

Wertebereich:

Hex	Baud	Hex	Baud	Hex	Baud
00	9600	06	2400	0C	38400
01	150	07	4800	0D	57600
02	300	08	7200	0F	76800
03	600	09	9600	0E	115200
04	1200	0A	14400	10	109700
05	1800	0B	19200		

■ Default: 00h (9600Baud)

PROTOCOL Das Protokoll, das verwendet werden soll. Diese Einstellung beeinflusst den weiteren Aufbau. Geben Sie für das ASCII-Protokoll den Wert 01h an.

OPTION3: Zeichenrahmen

Byte	Bit 7 ... 0
0	<ul style="list-style-type: none"> ■ Bit 1, 0: Datenbits <ul style="list-style-type: none"> – 00b = 5 Datenbits – 01b = 6 Datenbits – 10b = 7 Datenbits – 11b = 8 Datenbits ■ Bit 3, 2: Parity <ul style="list-style-type: none"> – 00b = none – 01b = odd – 10b = even – 11b = even ■ Bit 5, 4: Stopbits <ul style="list-style-type: none"> – 01b = 1 – 10b = 1,5 – 11b = 2 ■ Bit 7, 6: Flusskontrolle <ul style="list-style-type: none"> – 00b = keine – 10b = Hardware – 11b = XON/XOFF

- Default: 13h
 - (Datenbits: 8, Parität: keine, Stopbit: 1, Flusskontrolle: keine)

Datenbits

Anzahl der Datenbits, auf die ein Zeichen abgebildet wird.

Parity

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.

Stopbits

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

Flusskontrolle

Mechanismus, der den Datentransfer synchronisiert, wenn der Sender schneller Daten schickt, als der Empfänger verarbeiten kann. Die Flusskontrolle kann hardware- oder softwaremäßig (XON/XOFF) erfolgen. Bei der Hardware-Flusskontrolle werden die Leitungen RTS und CTS verwendet, die dann entsprechend zu verdrahten sind. Die Software-Flusskontrolle verwendet zur Steuerung die Steuerzeichen XON=11h und XOFF=13h. Bitte beachten Sie, dass dann Ihre Daten diese zwei Steuerzeichen nicht beinhalten dürfen.

OPTION4, 5: ZNA

Wartezeit, die eingehalten wird, bis der nächste Sendeauftrag ausgeführt wird. Die ZNA ist in ms anzugeben.

Option4: ZNA (High-Byte)

Option5: ZNA (Low-Byte)

Wertebereich: 0 ... 65535

- Default: 0

OPTION6, 7: ZVZ

Die Zeichenverzugszeit definiert den maximal zulässigen zeitlichen Abstand zwischen zwei empfangenen Zeichen innerhalb eines Telegramms. Die ZVZ wird in ms angegeben.

Bei ZVZ=0 berechnet sich der CP anhand der Baudrate die ZVZ selbst (ca. doppelte Zeichenzeit).

Option6: ZVZ (High-Byte)

Option7: ZVZ (Low-Byte)

Wertebereich: 0 ... 65535

- Default: 250

OPTION8: Anzahl Receive-buffer

Legt die Anzahl der Empfangspuffer fest. Solange nur 1 Empfangspuffer verwendet wird und dieser belegt ist, können keine weiteren Daten empfangen werden. Durch Aneinanderreihung von bis zu 250 Empfangspuffern können die empfangenen Daten in einen noch freien Empfangspuffer umgeleitet werden.

Wertebereich: 1 ... 250

- Default: 1

OPTION15: Betriebsart

Über die Betriebsart bestimmen Sie, ob die Schnittstelle halbduplex (RS485) oder voll duplex (RS422) betrieben werden soll.



Bei der Parametrierung halbduplex unter RS485 ist keine Software-Datenflusskontrolle möglich.

Wert	Beschreibung
00h	Halbduplex - Zweidraht-Betrieb (RS485) Die Daten werden zwischen den Kommunikationspartnern abwechselnd in beide Richtungen übertragen. Halbduplex-Betrieb bedeutet, dass zu einem Zeitpunkt entweder gesendet oder empfangen wird.
01h	Vollduplex - Vierdraht-Betrieb (RS422) Die Daten werden zwischen den Kommunikationspartnern gleichzeitig ausgetauscht, es kann zu einem Zeitpunkt sowohl gesendet als auch empfangen werden. Jeder Kommunikationspartner muss simultan eine Empfangsleitung betreiben.

Wertebereich:

00h: halbduplex


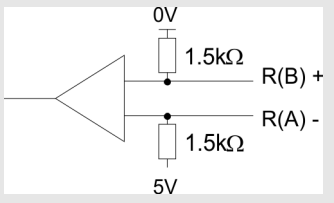
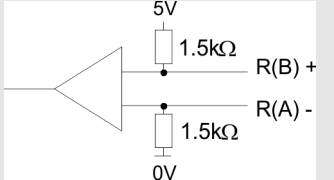
01h: voll duplex

- Default: 00h

OPTION16: Leitungsbelegung

Für einen reflexionsarmen Anschluss und die Drahtbruchererkennung (Breakerkennung) im RS422/485-Betrieb können die Leitungen über Parameter mit definiertem Ruhepegel vorgelegt werden.

An der CP-Schnittstelle ist die Beschaltung des Empfängers folgendermaßen realisiert:

Parameter	Beschreibung	Beschaltung Empfänger
00h (Default)	keine Keine Vorbelegung der Empfangsleitung. Diese Einstellung ist nur sinnvoll für busfähige Sondertreiber.	
01h	Signal R(A) 5V (Breakerkennung) Signal R(B) 0V Hier ist bei Vollduplex-Betrieb unter RS422 Drahtbrucherkennung möglich.	
02h	Signal R(A) 0V Signal R(B) 5V Diese Vorbelegung entspricht dem Ruhezustand (kein Sender aktiv) bei Halbduplex-Betrieb unter RS485. Es ist keine Drahtbrucherkennung möglich.	

Wertebereich:

00h: keine

01h: R(A) 5Volt R(B) 0Volt

02h: R(A) 0Volt R(B) 5Volt

■ Default: 00h

5.3 STX/ETX

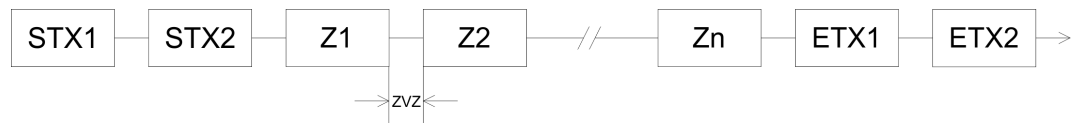
5.3.1 Grundlagen STX/ETX

Funktionsweise

STX/ETX ist ein einfaches Protokoll mit Header und Trailer. STX/ETX wird zur Übertragung von ASCII-Zeichen (20h...7Fh) eingesetzt. Dies erfolgt ohne Blockprüfung (BCC). Sollen Daten von der Peripherie eingelesen werden, muss als Startzeichen STX (Start of Text) vorhanden sein, anschließend folgen die zu übertragenden Zeichen. Als Schlusszeichen muss ETX (End of Text) vorliegen. Die Nutzdaten, d.h. alle Zeichen zwischen STX und ETX, werden nach Empfang des Schlusszeichens ETX an das übergeordnete System übergeben. Beim Senden der Daten werden die Nutzdaten an den CP übergeben und von dort, mit STX als Startzeichen und ETX als Schlusszeichen, an den Kommunikationspartner übertragen.

Telegrammaufbau

Sie können bis zu 2 Anfangs- und Endezeichen frei definieren. Auch hier kann eine ZNA für den Sender vorgegeben werden.



5.3.2 Parametrierdaten bei STX/ETX

Parameter

- DS - Datensatz für Zugriff über CPU, PROFIBUS und PROFINET
 - IX - Index für Zugriff über CANopen
 - SX - Subindex für Zugriff über EtherCAT mit Index 3100h + EtherCAT-Slot
- Näheres hierzu finden Sie im Handbuch zu Ihrem Bus-Koppler.

Name	Bytes	Funktion	Default	DS	IX	SX
PII_L	1	Länge Prozessabbild Eingabedaten ¹⁾	²⁾	02h	3100h	01h
PIQ_L	1	Länge Prozessabbild Ausgabedaten ¹⁾	²⁾	02h	3101h	02h
DIAG_EN	1	Diagnosealarm ¹⁾	00h	00h	3102h	03h
BAUD	1	Baudrate	00h	80h	3103h	04h
PROTOCOL	1	Protokoll	02h	80h	3104h	05h
OPTION3	1	Zeichenrahmen	13h	80h	3105h	06h
OPTION4, 5	2	ZNA 0 ... 65535 (in ms)	0	80h	3106h ... 3107h	07h ... 08h
OPTION6, 7	2	TMO 0 ... 65535 (in ms)	250	80h	3108h ... 3109h	09h ... 0Ah
OPTION8	1	Anzahl Startkennungen	1	80h	310Ah	0Bh
OPTION9	1	Startkennung 1	2	80h	310Bh	0Ch
OPTION10	1	Startkennung 2	0	80h	310Ch	0Dh
OPTION11	1	Anzahl Endekennungen	1	80h	310Dh	0Eh
OPTION12	1	Endekennung 1	3	80h	310Eh	0Fh

Name	Bytes	Funktion	Default	DS	IX	SX
OPTION13	1	Endekennung 2	0	80h	310Fh	10h
OPTION14	1	reserviert	00h	80h	3110h	11h
OPTION15	1	Betriebsart	00h	80h	3111h	12h
OPTION16	1	Leitungsbelegung	00h	80h	3112h	13h

1) Diesen Datensatz dürfen Sie ausschließlich im STOP-Zustand übertragen.

2) Wert hängt vom übergeordneten System ab.

DIAG_EN: Diagnosealarm

Hier aktivieren bzw. deaktivieren Sie die Diagnosefunktion.

Byte	Bit 7 ... 0
0	<ul style="list-style-type: none"> ■ Diagnosealarm <ul style="list-style-type: none"> – 00h: sperren – 40h: freigeben

■ Default: 00h

BAUD: Übertragungsrage

Geschwindigkeit der Datenübertragung in Bit/s (Baud). Sie haben folgende Einstellbereiche; andere Werte sind nicht zulässig.

Wertebereich:

Hex	Baud	Hex	Baud	Hex	Baud
00	9600	06	2400	0C	38400
01	150	07	4800	0D	57600
02	300	08	7200	0F	76800
03	600	09	9600	0E	115200
04	1200	0A	14400	10	109700
05	1800	0B	19200		

■ Default: 00h (9600Baud)

PROTOCOL

Das Protokoll, das verwendet werden soll. Diese Einstellung beeinflusst den weiteren Aufbau. Geben Sie für STX/ETX den Wert 02h an.

OPTION3: Zeichenrahmen

Byte	Bit 7 ... 0
0	<ul style="list-style-type: none"> ■ Bit 1, 0: Datenbits <ul style="list-style-type: none"> – 00b = 5 Datenbits – 01b = 6 Datenbits – 10b = 7 Datenbits – 11b = 8 Datenbits ■ Bit 3, 2: Parity <ul style="list-style-type: none"> – 00b = none – 01b = odd – 10b = even – 11b = even ■ Bit 5, 4: Stopbits <ul style="list-style-type: none"> – 01b = 1 – 10b = 1,5 – 11b = 2 ■ Bit 7, 6: Flusskontrolle <ul style="list-style-type: none"> – 00b = keine – 10b = Hardware – 11b = XON/XOFF

- Default: 13h
 - (Datenbits: 8, Parität: keine, Stopbit: 1, Flusskontrolle: keine)

Datenbits

Anzahl der Datenbits, auf die ein Zeichen abgebildet wird.

Parity

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.

Stopbits

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

Flusskontrolle

Mechanismus, der den Datentransfer synchronisiert, wenn der Sender schneller Daten schickt, als der Empfänger verarbeiten kann. Die Flusskontrolle kann hardware- oder softwaremäßig (XON/XOFF) erfolgen. Bei der Hardware-Flusskontrolle werden die Leitungen RTS und CTS verwendet, die dann entsprechend zu verdrahten sind. Die Software-Flusskontrolle verwendet zur Steuerung die Steuerzeichen XON=11h und XOFF=13h. Bitte beachten Sie, dass dann Ihre Daten diese zwei Steuerzeichen nicht beinhalten dürfen.

OPTION4, 5: ZNA

Wartezeit, die eingehalten wird, bis der nächste Sendeauftrag ausgeführt wird. Die ZNA ist in ms anzugeben.

Option4: ZNA (High-Byte)

Option5: ZNA (Low-Byte)

Wertebereich: 0 ... 65535

- Default: 0

OPTION6, 7: TMO

Mit TMO definieren Sie den maximal zulässigen zeitlichen Abstand zwischen zwei Telegrammen. TMO ist in ms anzugeben.

Option6: TMO (High-Byte)

Option7: TMO (Low-Byte)

Wertebereich: 0 ... 65535

- Default: 250

OPTION8: Anzahl Startkennungen

Hier können Sie 1 oder 2 Startkennungen einstellen. Ist "1" als Anzahl der Startkennungen eingestellt, wird der Inhalt des 2. Startkennzeichens ignoriert.

Bereich: 0 ... 2

- Default: 1

OPTION9, 10: Startkennung 1, 2

ASCII-Wert des Startzeichens, das einem Telegramm vorausgeschickt wird und den Start einer Übertragung kennzeichnet. Sie können 1 oder 2 Startzeichen verwenden. Bei Einsatz von 2 Startzeichen müssen Sie unter "Anzahl Startkennungen" eine 2 eintragen.

Startkennung 1, 2: Bereich: 0 ... 255

- Startkennung 1: Default: 3
- Startkennung 2: Default: 0

OPTION11: Anzahl Endekennungen

Hier können Sie 1 oder 2 Endekennungen einstellen. Ist "1" als Anzahl der Endekennungen eingestellt, wird der Inhalt des 2. Endekennzeichens ignoriert.

Bereich: 0 ... 2

- Default: 1

OPTION12, 13: Endekennung 1, 2

ASCII-Wert des Endezeichens, das nach einem Telegramm folgt und das Ende einer Übertragung kennzeichnet. Sie können 1 oder 2 Endezeichen verwenden. Bei Einsatz von 2 Endezeichen müssen Sie unter "Anzahl Endekennungen" eine 2 eintragen.

Endekennung 1, 2: Bereich: 0 ... 255

- Endekennung 1: Default: 3
- Endekennung 2: Default: 0

OPTION15: Betriebsart

Über die Betriebsart bestimmen Sie, ob die Schnittstelle halbduplex (RS485) oder voll duplex (RS422) betrieben werden soll.



Bei der Parametrierung halbduplex unter RS485 ist keine Software-Datenflusskontrolle möglich.

Wert	Beschreibung
00h	Halbduplex - Zweidraht-Betrieb (RS485) Die Daten werden zwischen den Kommunikationspartnern abwechselnd in beide Richtungen übertragen. Halbduplex-Betrieb bedeutet, dass zu einem Zeitpunkt entweder gesendet oder empfangen wird.
01h	Vollduplex - Vierdraht-Betrieb (RS422) Die Daten werden zwischen den Kommunikationspartnern gleichzeitig ausgetauscht, es kann zu einem Zeitpunkt sowohl gesendet als auch empfangen werden. Jeder Kommunikationspartner muss simultan eine Empfangsleitung betreiben.

Wertebereich:

00h: halbduplex

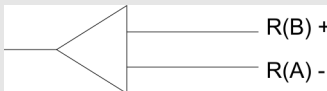
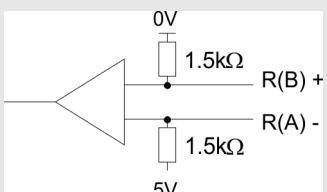
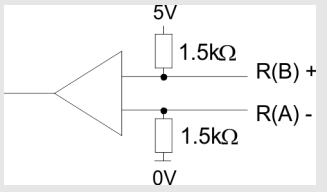
01h: vollduplex

■ Default: 00h

OPTION16: Leitungsbelegung

Für einen reflexionsarmen Anschluss und die Drahtbruchererkennung (Breakerkennung) im RS422/485-Betrieb können die Leitungen über Parameter mit definiertem Ruhepegel vorgelegt werden.

An der CP-Schnittstelle ist die Beschaltung des Empfängers folgendermaßen realisiert:

Parameter	Beschreibung	Beschaltung Empfänger
00h (Default)	keine Keine Vorbelegung der Empfangsleitung. Diese Einstellung ist nur sinnvoll für busfähige Sondertreiber.	
01h	Signal R(A) 5V (Breakerkennung) Signal R(B) 0V Hier ist bei Vollduplex-Betrieb unter RS422 Drahtbruchererkennung möglich.	
02h	Signal R(A) 0V Signal R(B) 5V Diese Vorbelegung entspricht dem Ruhezustand (kein Sender aktiv) bei Halbduplex-Betrieb unter RS485. Es ist keine Drahtbruchererkennung möglich.	

Wertebereich:

00h: keine

01h: R(A) 5Volt R(B) 0Volt

02h: R(A) 0Volt R(B) 5Volt

■ Default: 00h

5.4 3964(R)

5.4.1 Grundlagen 3964(R)

Funktionsweise

3964(R) steuert die Datenübertragung bei einer Punkt-zu-Punkt-Kopplung zwischen dem CP und einem Kommunikationspartner. Hier werden bei der Datenübertragung den Nutzdaten Steuerzeichen hinzugefügt. Durch diese Steuerzeichen kann der Kommunikationspartner kontrollieren, ob die Daten vollständig und fehlerfrei bei ihm angekommen sind.

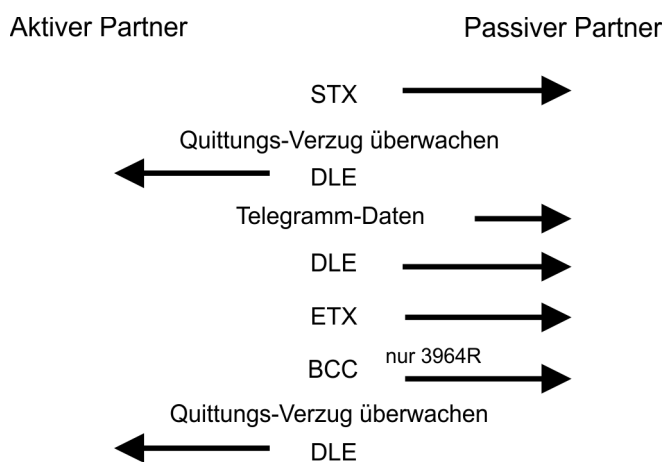
Folgende Steuerzeichen werden ausgewertet:

- STX Start of Text
- DLE Data Link Escape
- ETX End of Text
- BCC Block Check Character (nur bei 3964R)
- NAK Negative Acknowledge



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 3964(R) muss dem Kommunikationspartner eine niedrigere Priorität zugeordnet sein. Wenn beide Kommunikationspartner gleichzeitig einen Sendeauftrag erteilen, dann stellt der Partner mit niedrigerer Priorität seinen Sendeauftrag zurück.

Ablauf



Sie können pro Telegramm maximal 250Byte übertragen.

Time-out-Zeiten

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 5 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 5 Versuche unternommen, danach ein NAK gesendet und die Übertragung abgebrochen.

3964(R) > Parametrierdaten bei 3964(R)

- Passivbetrieb** Wenn der Treiber auf den Verbindungsaufbau wartet und ein Zeichen ungleich STX empfängt, sendet er NAK. Bei Empfang eines Zeichens NAK sendet der Treiber keine Antwort. Wird beim Empfang die ZVZ überschritten, wird ein NAK gesendet und auf erneuten Verbindungsaufbau gewartet. Wenn der Treiber beim Empfang des STX noch nicht bereit ist, sendet er ein NAK.
- Block-Check-Character (BCC-Byte)** Zur weiteren Datensicherung wird bei 3964R am Ende des Telegramms ein Block-Check-Charakter 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.
- Initialisierungskonflikt** Versuchen beide Partner gleichzeitig innerhalb der QVZ einen Verbindungsaufbau, so sendet der Partner mit der niedrigeren Priorität das DLE und geht auf Empfang.
- Data Link Escape (DLE-Zeichen)** Das DLE-Zeichen in einem Telegramm wird vom Treiber 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

5.4.2 Parametrierdaten bei 3964(R)

Parameter

- DS - Datensatz für Zugriff über CPU, PROFIBUS und PROFINET
 IX - Index für Zugriff über CANopen
 SX - Subindex für Zugriff über EtherCAT mit Index 3100h + EtherCAT-Slot
- Näheres hierzu finden Sie im Handbuch zu Ihrem Bus-Koppler.

Name	Bytes	Funktion	Default	DS	IX	SX
PII_L	1	Länge Prozessabbild Eingabedaten ¹⁾	2)	02h	3100h	01h
PIQ_L	1	Länge Prozessabbild Ausgabedaten ¹⁾	2)	02h	3101h	02h
DIAG_EN	1	Diagnosealarm ¹⁾	00h	00h	3102h	03h
BAUD	1	Baudrate	00h	80h	3103h	04h
PROTOCOL	1	Protokoll	03h	80h	3104h	05h
OPTION3	1	Zeichenrahmen	13h	80h	3105h	06h
OPTION4	1	ZNA (x 20ms)	0	80h	3106h	07h
OPTION5	1	ZVZ (x 20ms)	10	80h	3107h	08h
OPTION6	1	QVZ (x 20ms)	10	80h	3108h	09h
OPTION7	1	BWZ (x 20ms)	10	80h	3109h	0Ah

Name	Bytes	Funktion	Default	DS	IX	SX
OPTION8	1	STX Wiederholungen	5	80h	310Ah	0Bh
OPTION9	1	DBL	6	80h	310Bh	0Ch
OPTION10	1	Priorität	0	80h	310Ch	0Dh
OPTION11...14	4	reserviert	00h	80h	310Dh 3110h	0Eh ... 11h
OPTION15	1	Betriebsart	00h	80h	3111h	12h
OPTION16	1	Leitungsbelegung	00h	80h	3112h	13h

1) Diesen Datensatz dürfen Sie ausschließlich im STOP-Zustand übertragen.

2) Wert hängt vom übergeordneten System ab.

DIAG_EN: Diagnosealarm

Hier aktivieren bzw. deaktivieren Sie die Diagnosefunktion.

Byte	Bit 7 ... 0
0	<ul style="list-style-type: none"> <input type="checkbox"/> Diagnosealarm – 00h: sperren – 40h: freigeben

■ Default: 00h

BAUD: Übertragungsrate

Geschwindigkeit der Datenübertragung in Bit/s (Baud). Sie haben folgende Einstellbereiche; andere Werte sind nicht zulässig.

Wertebereich:

Hex	Baud	Hex	Baud	Hex	Baud
00	9600	06	2400	0C	38400
01	150	07	4800	0D	57600
02	300	08	7200	0F	76800
03	600	09	9600	0E	115200
04	1200	0A	14400	10	109700
05	1800	0B	19200		

■ Default: 00h (9600Baud)

PROTOCOL

Das Protokoll, das verwendet werden soll. Diese Einstellung beeinflusst den weiteren Aufbau.

Wertebereich: 03h: 3964

Wertebereich: 04h: 3964R

■ Default: 03h

OPTION3: Zeichenrahmen

Byte	Bit 7 ... 0
0	<ul style="list-style-type: none"> ■ Bit 1, 0: Datenbits <ul style="list-style-type: none"> – 00b = 5 Datenbits – 01b = 6 Datenbits – 10b = 7 Datenbits – 11b = 8 Datenbits ■ Bit 3, 2: Parity <ul style="list-style-type: none"> – 00b = none – 01b = odd – 10b = even – 11b = even ■ Bit 5, 4: Stopbits <ul style="list-style-type: none"> – 01b = 1 – 10b = 1,5 – 11b = 2 ■ Bit 7, 6: reserviert

- Default: 13h
 - (Datenbits: 8, Parität: keine, Stopbit: 1)

Datenbits

Anzahl der Datenbits, auf die ein Zeichen abgebildet wird.

Parity

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.

Stopbits

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

OPTION4: ZNA

Wartezeit, die eingehalten wird, bis der nächste Sendeauftrag ausgeführt wird. Die ZNA wird als Faktor von 20ms angegeben.

Wertebereich: 0 ... 255

- Default: 0

OPTION5: ZVZ

Die Zeichenverzugszeit (ZVZ) definiert den maximal zulässigen zeitlichen Abstand zwischen zwei empfangenen Zeichen innerhalb eines Telegramms. Die ZVZ wird als Faktor von 20ms-Schritten angegeben.

Bei ZVZ=0 berechnet sich der CP anhand der Baudrate die ZVZ selbst (ca. doppelte Zeichenzeit).

Wertebereich: 0 ... 255

- Default: 10

OPTION6: QVZ

Die Quittungsverzugszeit (QVZ) definiert den maximal zulässigen zeitlichen Abstand bis zur Quittung des Partners bei Verbindungsauf- und -abbau. Die QVZ wird als Faktor von 20ms-Schritten angegeben.

Wertebereich: 0 ... 255

- Default: 10

- OPTION7: BWZ** Die Blockwartezeit (BWZ) ist die maximale Zeitdauer zwischen der Bestätigung eines Anforderungstelegrams (DLE) und STX des Reaktionstelegramms. Die BWZ wird als Faktor von 20ms-Schritten angegeben.
Wertebereich: 0 ... 255
 ■ Default: 10
- OPTION8: STX-Wiederholungen** Maximale Anzahl der Versuche des CP, eine Verbindung aufzubauen.
Wertebereich: 0 ... 255
 ■ Default: 5
- OPTION9: DBL** Bei Überschreiten der Blockwartezeit (BWZ) können Sie über den Parameter DBL die maximale Anzahl Wiederholungen für das Anforderungstelegramm vorgeben. Sind diese Versuche erfolglos, wird die Übertragung abgebrochen.
Wertebereich: 0 ... 255
 ■ Default: 6
- OPTION10: Priorität** Ein Kommunikationspartner hat hohe Priorität, wenn sein Sendeversuch Vorrang gegenüber dem Sendewunsch des Partners hat. Bei niedriger Priorität muss dieser hinter dem Sendewunsch des Partners zurückstehen. Bei 3964(R) müssen die Prioritäten beider Partner unterschiedlich sein. Sie haben folgende Einstellmöglichkeiten:
Wertebereich: 00h: low
Wertebereich: 01h: high
 ■ Default: 0
- OPTION15: Betriebsart** Über die Betriebsart bestimmen Sie, ob die Schnittstelle halbduplex (RS485) oder voll duplex (RS422) betrieben werden soll.



Bei der Parametrierung halbduplex unter RS485 ist keine Software-Datenflusskontrolle möglich.

Wert	Beschreibung
00h	Halbduplex - Zweidraht-Betrieb (RS485) Die Daten werden zwischen den Kommunikationspartnern abwechselnd in beide Richtungen übertragen. Halbduplex-Betrieb bedeutet, dass zu einem Zeitpunkt entweder gesendet oder empfangen wird.
01h	Vollduplex - Vierdraht-Betrieb (RS422) Die Daten werden zwischen den Kommunikationspartnern gleichzeitig ausgetauscht, es kann zu einem Zeitpunkt sowohl gesendet als auch empfangen werden. Jeder Kommunikationspartner muss simultan eine Empfangsleitung betreiben.

- Wertebereich:**
 00h: halbduplex
 01h: vollduplex
 ■ Default: 00h

OPTION16: Leitungsbelegung

Für einen reflexionsarmen Anschluss und die Drahtbruchererkennung (Breakerkennung) im RS422/485-Betrieb können die Leitungen über Parameter mit definiertem Ruhepegel vorbelegt werden.

An der CP-Schnittstelle ist die Beschaltung des Empfängers folgendermaßen realisiert:

Parameter	Beschreibung	Beschaltung Empfänger
00h (Default)	keine Keine Vorbelegung der Empfangsleitung. Diese Einstellung ist nur sinnvoll für busfähige Sondertreiber.	
01h	Signal R(A) 5V (Breakerkennung) Signal R(B) 0V Hier ist bei Vollduplex-Betrieb unter RS422 Drahtbruchererkennung möglich.	
02h	Signal R(A) 0V Signal R(B) 5V Diese Vorbelegung entspricht dem Ruhezustand (kein Sender aktiv) bei Halbduplex-Betrieb unter RS485. Es ist keine Drahtbruchererkennung möglich.	

Wertebereich:

- 00h: keine
- 01h: R(A) 5Volt R(B) 0Volt
- 02h: R(A) 0Volt R(B) 5Volt
- Default: 00h

5.5 Modbus

5.5.1 Grundlagen Modbus

Übersicht

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 als Punkt-zu-Punkt- unter RS232 oder als Mehrpunkt-Verbindung unter RS485.

Master-Slave-Kommunikation

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.

Telegramm-Aufbau

Die Anforderungs-Telegramme, die ein Master sendet, und die Antwort-Telegramme eines Slaves haben den gleichen Aufbau:

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

Broadcast mit Slave-Adresse = 0

Eine Anforderung kann an einen bestimmten Slave gerichtet sein oder als Broadcast-Nachricht an alle Slaves gehen. Zur Kennzeichnung einer Broadcast-Nachricht wird die Slave-Adresse 0 eingetragen.

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 erreichen Sie einen höheren Datendurchsatz als im ASCII-Modus. Anstelle von Anfang- und Ende-Zeichen wird eine Zeitüberwachung eingesetzt.

Die Modus-Wahl erfolgt bei der Parametrierung.

5.5.2 Modbus auf dem CP von VIPA

Der CP unterstützt verschiedene Modbus-Betriebsarten, die nachfolgend beschrieben sind:

Modbus Master

Im *Modbus Master* Betrieb steuern Sie die Modbus-Kommunikation in Ihrem übergeordneten System über Ihr SPS-Anwenderprogramm. Mittels der Modbus Funktionscodes können Sie über den Modbus Master schreibend bzw. lesend auf Ihre Modbus Slaves zugreifen. Sie haben hier die Möglichkeit bis zu 250Byte Nutzdaten pro Telegramm zu übertragen.

Modbus Slave short

Im *Modbus Slave short* Betrieb kommuniziert der CP mit einem Modbus Master. Abhängig vom Funktionscode empfängt der CP Daten vom Modbus Master oder stellt ihm Daten bereit. Das Datenhandling auf Slave-Seite erfolgt automatisch. Diese Betriebsart eignet sich besonders zur schnellen Datenübertragung kleiner Datenmengen über Modbus.

Modbus Slave long

Im *Modbus Slave long* Betrieb wird nur ein geänderter Datenbereich, beginnend bei 0, vom CP an das übergeordnete System übertragen. Fordert der Modbus Master Daten an, so ist mit einem Anwenderprogramm dafür zu sorgen, dass sich die relevanten Daten im CP befinden. Schreibende Master-Zugriffe dürfen nicht außerhalb des Empfangsbereichs liegen!



Erst nachdem alle Daten im CP vorliegen, sendet der CP ein Antworttelegramm an den Master.

5.5.3 Parametrierdaten bei Modbus

Parameter

DS - Datensatz für Zugriff über CPU, PROFIBUS und PROFINET

IX - Index für Zugriff über CANopen

SX - Subindex für Zugriff über EtherCAT mit Index 3100h + EtherCAT-Slot

Näheres hierzu finden Sie im Handbuch zu Ihrem Bus-Koppler.

Name	Bytes	Funktion	Default	DS	IX	SX
PII_L	1	Länge Prozessabbild Eingabedaten ¹⁾	²⁾	02h	3100h	01h
PIQ_L	1	Länge Prozessabbild Ausgabedaten ¹⁾	²⁾	02h	3101h	02h
DIAG_EN	1	Diagnosealarm ¹⁾	00h	00h	3102h	03h
BAUD	1	Baudrate	00h	80h	3103h	04h
PROTOCOL	1	Protokoll	0Bh	80h	3104h	05h
OPTION3	1	Zeichenrahmen	13h	80h	3105h	06h
OPTION4	1	Slave-Adresse	1	80h	3106h	07h
OPTION5, 6	2	Wartezeit	0	80h	3107h ... 3108h	08h ... 09h
OPTION7...14	8	reserviert	00h	80h	3109h ... 3110h	0Ah ... 11h
OPTION15	1	Betriebsart	00h	80h	3111h	12h
OPTION16	1	Leitungsbelegung	00h	80h	3112h	13h

1) Diesen Datensatz dürfen Sie ausschließlich im STOP-Zustand übertragen.

2) Wert hängt vom übergeordneten System ab.

DIAG_EN: Diagnosealarm

Hier aktivieren bzw. deaktivieren Sie die Diagnosefunktion.

Byte	Bit 7 ... 0
0	<ul style="list-style-type: none"> ■ Diagnosealarm – 00h: sperren – 40h: freigeben

■ Default: 00h

BAUD: Übertragungsrage

Geschwindigkeit der Datenübertragung in Bit/s (Baud). Sie haben folgende Einstellbereiche; andere Werte sind nicht zulässig.

Wertebereich:

Hex	Baud	Hex	Baud	Hex	Baud
00	9600	06	2400	0C	38400
01	150	07	4800	0D	57600
02	300	08	7200	0F	76800
03	600	09	9600	0E	115200
04	1200	0A	14400	10	109700
05	1800	0B	19200		

- Default: 00h (9600Baud)

PROTOCOL

Geben Sie hier das Protokoll an, das verwendet werden soll. Diese Einstellung beeinflusst den weiteren Aufbau.

Wertebereich unter Modbus:

0Ah:	Modbus Master ASCII
0Bh:	Modbus RTU
0Ch:	Modbus Slave ASCII short
0Dh:	Modbus Slave RTU short
1Ch:	Modbus Slave ASCII long
1Dh:	Modbus Slave RTU long

- Default: 0Bh

OPTION3: Zeichenrahmen

Byte	Bit 7 ... 0
0	<ul style="list-style-type: none"> ■ Bit 1, 0: Datenbits <ul style="list-style-type: none"> – 00b = 5 Datenbits – 01b = 6 Datenbits – 10b = 7 Datenbits – 11b = 8 Datenbits ■ Bit 3, 2: Parity <ul style="list-style-type: none"> – 00b = none – 01b = odd – 10b = even – 11b = even ■ Bit 5, 4: Stopbits <ul style="list-style-type: none"> – 01b = 1 – 10b = 1,5 – 11b = 2 ■ Bit 7, 6: reserviert

- Default: 13h
 - (Datenbits: 8, Parität: keine, Stopbit: 1)

Datenbits

Anzahl der Datenbits, auf die ein Zeichen abgebildet wird.

- Parity** 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.
- Stopbits** Die Stopbits werden jedem zu übertragenden Zeichen nachgesetzt und kennzeichnen das Ende eines Zeichens.
- OPTION4: Slave-Adresse** Geben Sie unter dem Modbus Slave Protokoll eine Adresse für den Modbus Slave an. Mittels dieser Adresse erfolgt der Zugriff über die Modbus-Funktionscodes.
Unter Modbus Master wird dieser Parameter ignoriert.
Bereich: 1 ... 255
■ Default: 1
- OPTION5, 6: Wartezeit** Hier ist für den Modbus Master eine Wartezeit in ms vorzugeben. Mit 0 wird die Wartezeit protokollabhängig nach folgender Formel automatisch ermittelt:
Modbus ASCII:
- $$50ms + \frac{2926000ms}{\text{Baudrate}} \times \text{Bit/s}$$
- mit Baudrate in Bit/s
Modbus RTU:
- $$50ms + \frac{5190000ms}{\text{Baudrate}} \times \text{Bit/s}$$
- mit Baudrate in Bit/s
Unter Modbus Slave wird dieser Parameter ignoriert.
Option5: Wartezeit (High-Byte)
Option6: Wartezeit (Low-Byte)
Wertebereich: 0 ... 60000 in ms
■ Default: 0
- OPTION15: Betriebsart** Über die Betriebsart bestimmen Sie, ob die Schnittstelle halbduplex (RS485) oder voll-duplex (RS422) betrieben werden soll.



Bei der Parametrierung halbduplex unter RS485 ist keine Software-Datenflusskontrolle möglich.

Wert	Beschreibung
00h	Halbduplex - Zweidraht-Betrieb (RS485) Die Daten werden zwischen den Kommunikationspartnern abwechselnd in beide Richtungen übertragen. Halbduplex-Betrieb bedeutet, dass zu einem Zeitpunkt entweder gesendet oder empfangen wird.
01h	Vollduplex - Vierdraht-Betrieb (RS422) Die Daten werden zwischen den Kommunikationspartnern gleichzeitig ausgetauscht, es kann zu einem Zeitpunkt sowohl gesendet als auch empfangen werden. Jeder Kommunikationspartner muss simultan eine Empfangsleitung betreiben.

Wertebereich:

00h: halbduplex

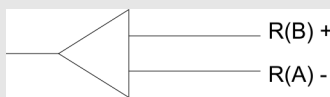
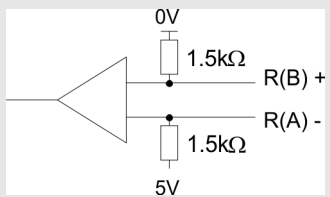
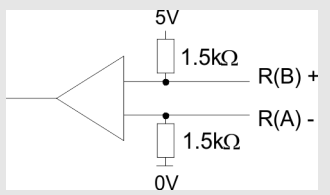
01h: vollduplex

■ Default: 00h

OPTION16: Leitungsbelegung

Für einen reflexionsarmen Anschluss und die Drahtbruchererkennung (Breakerkennung) im RS422/485-Betrieb können die Leitungen über Parameter mit definiertem Ruhepegel vorgelegt werden.

An der CP-Schnittstelle ist die Beschaltung des Empfängers folgendermaßen realisiert:

Parameter	Beschreibung	Beschaltung Empfänger
00h (Default)	keine Keine Vorbelegung der Empfangsleitung. Diese Einstellung ist nur sinnvoll für busfähige Sondertreiber.	
01h	Signal R(A) 5V (Breakerkennung) Signal R(B) 0V Hier ist bei Vollduplex-Betrieb unter RS422 Drahtbruchererkennung möglich.	
02h	Signal R(A) 0V Signal R(B) 5V Diese Vorbelegung entspricht dem Ruhezustand (kein Sender aktiv) bei Halbduplex-Betrieb unter RS485. Es ist keine Drahtbruchererkennung möglich.	

Wertebereich:

00h: keine

01h: R(A) 5Volt R(B) 0Volt

02h: R(A) 0Volt R(B) 5Volt

■ Default: 00h

5.6 Einsatz - Modbus

5.6.1 Modbus - Übersicht

Die Anzahl der Ein- und Ausgangs-Daten, abhängig von der IO-Size, ist beim CP 040-1CA00 über GSD parametrierbar. Für den Einsatz unter Modbus ist immer eine Hardware-Konfiguration durchzuführen.

Voraussetzung für den Betrieb

Folgende Komponenten sind zum Einsatz der SLIO Modbus-Module erforderlich:

- Master-System bestehend aus System SLIO mit CP 040
- Slave-System bestehend aus System SLIO mit CP 040
- Siemens SIMATIC Manager bzw. WinPLC7 von VIPA
- GSD-Datei
- VIPA Hantierungsbausteine Fx000011_Vxxx.zip
- Serielle Verbindung zwischen beiden CPs

Parametrierung

Für den CP 040 ist immer eine Hardware-Konfiguration durchzuführen. Hierfür ist die Einbindung der GSD-Datei von VIPA im Hardware-Katalog erforderlich. Die Parametrierung erfolgt nach folgender Vorgehensweise:

- Starten Sie den Siemens SIMATIC Manager bzw. WinPLC7 von VIPA.
- Installieren Sie die ausgewählte GSD-Datei im Hardware-Katalog.
- Projektieren Sie ein SLIO System.
- Binden Sie einen mit "Modbus" bezeichneten CP 040 an.
- Parametrieren Sie den CP 040 nach Ihren Vorgaben.
- Übertragen Sie Ihr Projekt in die SPS.

SPS-Programm

Mit Ausnahme bei "Modbus Slave short" ist immer für die Kommunikation zusätzlich ein SPS-Programm erforderlich. Hierbei erfolgt die Kommunikation über Hantierungsbausteine, die Sie in Form der VIPA-Library Fx000011_Vxxx.zip in Ihr Konfigurationstool einbinden können. Sie finden diese auch unter www.vipa.com im Servicebereich.

5.6.1.1 Kommunikationsmöglichkeiten

Nachfolgend sollen die Kommunikationsmöglichkeiten zwischen Modbus Master und Modbus Slave an folgenden Kombinationsmöglichkeiten gezeigt werden:

- CP 040 Modbus Master ↔ CP 040 Modbus Slave short
- CP 040 Modbus Master ↔ CP 040 Modbus Slave long

Master ↔ Slave short

Modbus Master

Die Kommunikation im Master-Modus erfolgt über Datenbausteine unter Einsatz der Hantierungsbausteine FB 60 - SEND und FB 61 - RECEIVE (oder FB 65 SEND_RECV). Hier können Sie bis zu 250Byte Nutzdaten pro Telegramm übertragen.

Modbus Slave short

Im Modbus Slave short Modus ist die Anzahl der Nutzdaten für Ein- und Ausgabe auf IO-Size begrenzt. Hierbei ist für den Einsatz auf Slave-Seite lediglich eine Hardware-Konfiguration durchzuführen.

Vorgehensweise

1. ➤ Bauen Sie für Master- und Slave-Seite je ein System SLIO auf, welches jeweils einen CP 040 beinhaltet.
2. ➤ Verbinden Sie beide Systeme über die serielle Schnittstelle.
3. ➤ Projektieren Sie die Master-Seite.

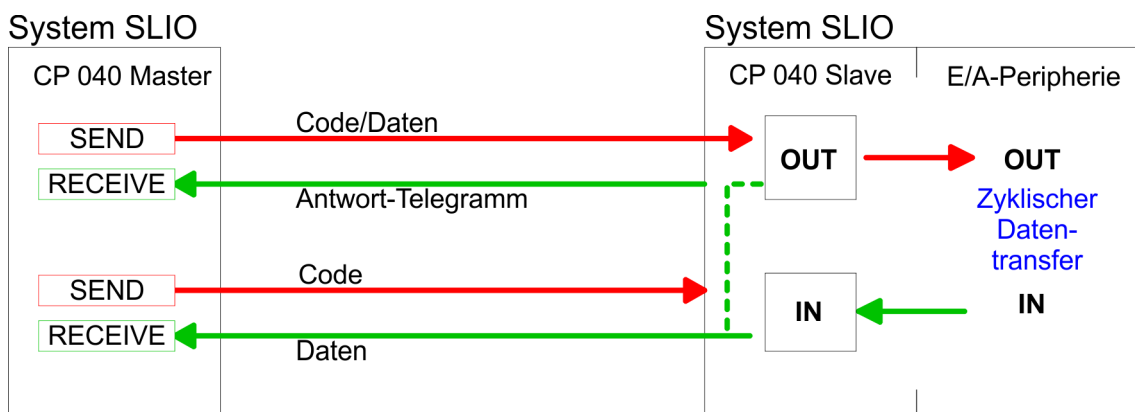
Die Parametrierung des CP 040 als Modbus Master erfolgt über die Hardware-Konfiguration. Zusätzlich ist für die Kommunikation ein SPS-Anwenderprogramm erforderlich, das nach folgender Struktur aufgebaut sein sollte:

OB 100: Einmaliger Aufruf der Hantierungsbausteine FB 60 - SEND und FB 61 - RECEIVE (oder FB 65 SEND_RECV) mit allen Parametern und gesetztem R zur Initialisierung.

OB 1: Aufruf des FB 60 - SEND (oder FB 65 SEND_RECV) mit Fehlerauswertung. Hierbei ist das Telegramm gemäß den Modbus-Vorgaben im Sendebaustein abzulegen. Aufruf des FB 61 - RECEIVE mit Fehlerauswertung. Gemäß den Modbus-Vorgaben werden die Daten im Empfangsbaustein abgelegt.

4. ➤ Projektieren Sie die Slave-Seite

Die Parametrierung des CP 040 erfolgt über die Hardware-Konfiguration. Geben Sie hier für Ein- und Ausgabe-Bereich die Startadresse an ab welcher, abhängig von der IO-Size, für Ein- und Ausgabe die Daten abliegen.



Master ↔ Slave long

Modbus Master

Die Kommunikation im Master-Modus erfolgt über Datenbausteine unter Einsatz der Hantierungsbausteine FB 60 - SEND und FB 61 - RECEIVE (oder FB 65 SEND_RECV). Hier können Sie bis zu 250Byte Nutzdaten pro Telegramm übertragen.

Modbus Slave long

Im Modbus Slave long Modus wird nur ein geänderter Datenbereich beginnend bei 0 mit FB 61 - RECEIVE an die CPU übertragen. Fordert der Master Daten an, so ist dafür Sorge zu tragen, dass sich die relevanten Daten im CP befinden. Mit einem FB 60 - SEND wird ein gewünschter Datenbereich von 0 beginnend in den CP übertragen.

Vorgehensweise

1. ➤ Bauen Sie für Master- und Slave-Seite je ein System SLIO auf, welches jeweils einen CP 040 beinhaltet.
2. ➤ Verbinden Sie beide Systeme über die serielle Schnittstelle.

3. ▶ Projektieren Sie die Master-Seite.

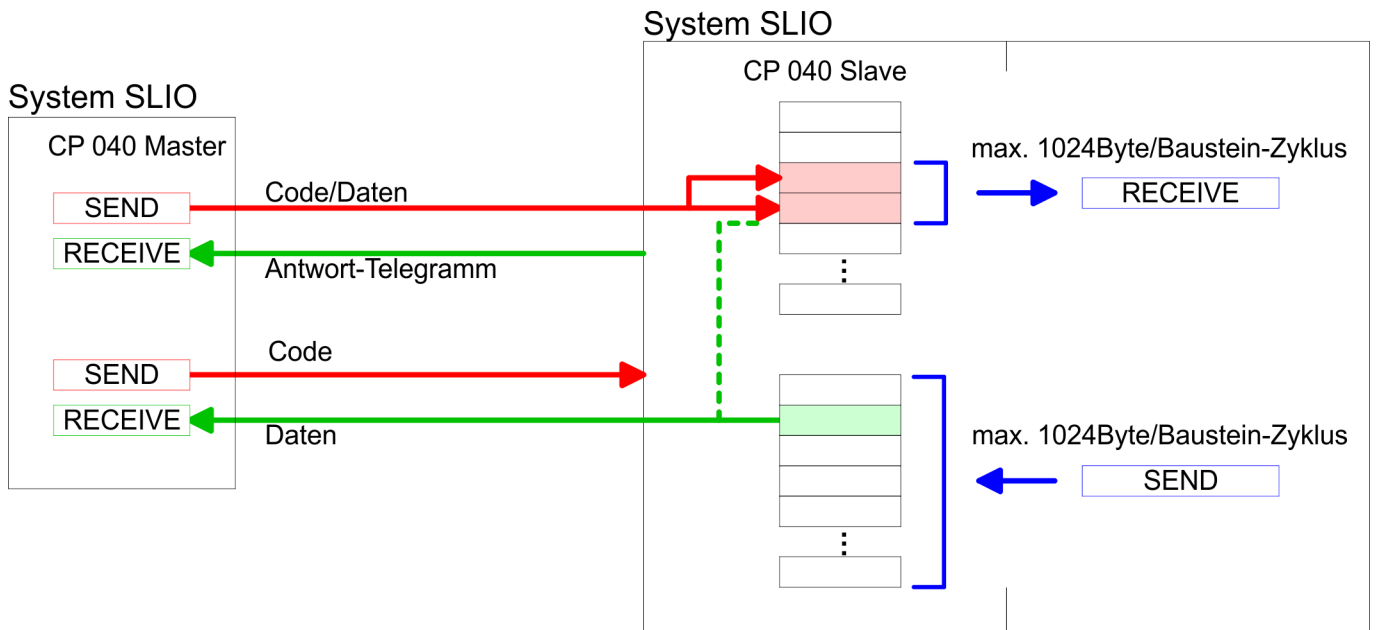
Die Projektierung auf der Master-Seite erfolgt auf die gleiche Weise, wie im Beispiel weiter oben beschrieben.

4. ▶ Projektieren Sie die Slave-Seite.

Die Parametrierung des CP 040 als Modbus Slave erfolgt über die Hardware-Konfiguration. Zusätzlich ist für die Kommunikation ein SPS-Anwenderprogramm erforderlich, das nach folgender Struktur aufgebaut sein sollte:

OB 100: Einmaliger Aufruf der Hantierungsbausteine FB 60 - SEND und FB 61 - RECEIVE (oder FB 65 SEND_RECV) mit allen Parametern und gesetztem R zur Initialisierung.

OB 1: Aufruf des FB 60 - SEND (oder FB 65 SEND_RECV) mit Fehlerauswertung. Hierbei wird ein Bereich von 0 beginnend im CP 040 abgelegt, auf den vom Master über Modbus zugegriffen werden kann. Mit dem FB 61 - RECEIVE mit Fehlerauswertung können Sie einen Datenbereich in die CPU übertragen. Bei Datenänderung durch den Master werden nur die Daten an die CPU übergeben, in denen geändert wurde.



5.6.2 Modbus - Zugriff auf mehrere Slaves

Bei Einsatz mehrerer Slaves können unter RS485 keine Buskonflikte auftreten, da der Master immer nur mit einem Slave kommunizieren kann. Der Master schickt an den über die Adresse spezifizierten Slave ein Kommandotelegramm und wartet eine gewisse Zeit, in der der Slave sein Antworttelegramm senden kann. Während des Wartens ist eine Kommunikation mit einem anderen Slave nicht möglich. Zur Kommunikation mit mehreren Slaves ist für jeden Slave ein SEND Datenbaustein für das Kommandotelegramm und ein RECEIVE Datenbaustein für das Antworttelegramm erforderlich. Eine Applikation mit mehreren Slaves würde aus entsprechend vielen Datenbausteinen mit Kommandos bestehen.

Diese werden der Reihe nach abgearbeitet:

1. Slave:

1. ➔ Sende Kommandotelegramm an Slave-Adresse 1. Slave
2. ➔ Empfange Antworttelegramm von Slave-Adresse 1. Slave
3. ➔ Werte Antworttelegramm aus

2. Slave:

1. ➔ Sende Kommandotelegramm an Slave-Adresse 2. Slave
2. ➔ Empfange Antworttelegramm von Slave-Adresse 2. Slave
3. ➔ Werte Antworttelegramm aus

3. Slave:

➔ ...

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.

Nur Schreibaufträge dürfen als Broadcast gesendet werden.

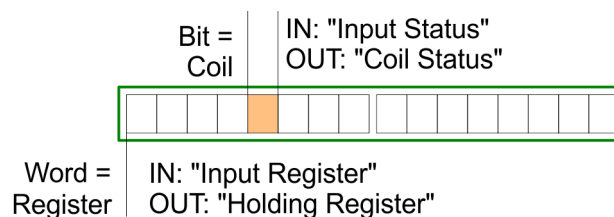


Nach einem Broadcast wartet der Master nicht auf ein Antworttelegramm.

5.6.3 Modbus - Funktionscodes

Namenskonventionen

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



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

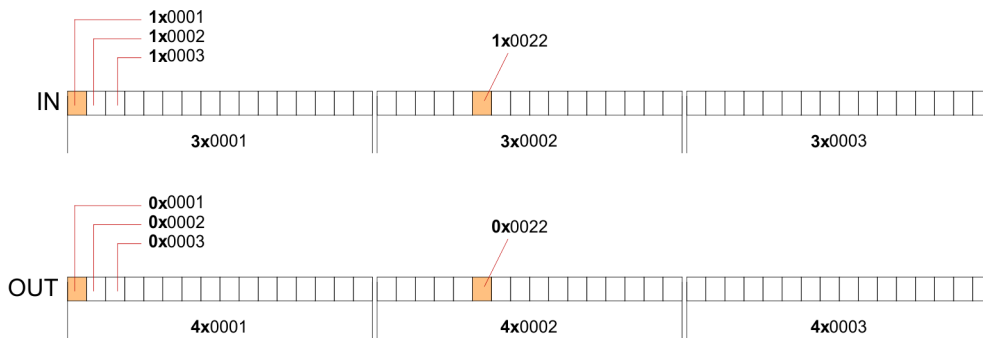
Bereichsdefinitionen

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

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

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

- 0x - Bit-Bereich für Ausgabe-Daten des Masters
Zugriff über Funktions-Code 01h, 05h, 0Fh
- 1x - Bit-Bereich für Eingabe-Daten des Masters
Zugriff über Funktions-Code 02h
- 3x - Wort-Bereich für Eingabe-Daten des Masters
Zugriff über Funktions-Code 04h
- 4x - Wort-Bereich für Ausgabe-Daten des Masters
Zugriff über Funktions-Code 03h, 06h, 10h



Eine Beschreibung der Funktions-Codes finden Sie auf den Folgeseiten.

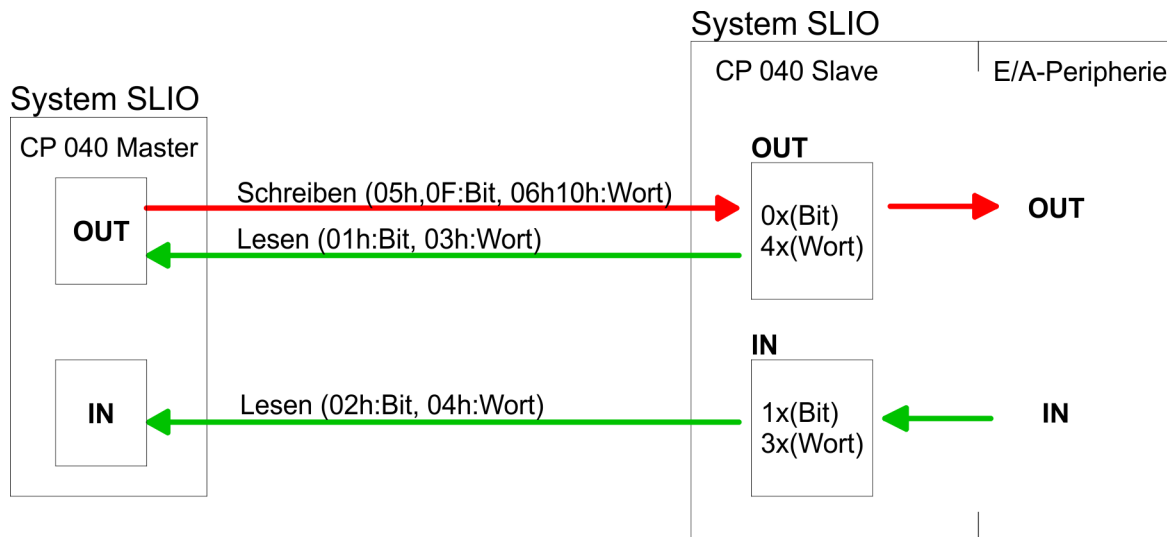
Übersicht

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
0Fh	Write n Bits	n Bit schreiben in Master-Ausgabe-Bereich 0x
10h	Write n Words	n Worte schreiben in Master-Ausgabe-Bereich 4x

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

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



Antwort des Slaves

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

Ist kein Fehler aufgetreten, wird der Funktionscode zurückgeliefert.

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

Byte-Reihenfolge im Wort

1 Wort	
High-Byte	Low-Byte

Prüfsumme CRC, RTU, LRC

Die aufgezeigten Prüfsummen CRC bei RTU- und LRC bei ASCII-Modus werden auto-matisch an jedes Telegramm angehängt. Sie werden nicht im Datenbaustein angezeigt.

Slave-Adresse

Die Slave-Adresse muss identisch sein mit der parametrisierten Slave- Adresse (OPTION4).

Read n Bits 01h, 02h

Code 01h: n Bit lesen von Master-Ausgabe-Bereich 0x

Code 02h: n Bit lesen von Master-Eingabe-Bereich 1x

Kommandotelegramm

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

Antworttelegramm

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

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

Kommandotelegramm

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

Antworttelegramm

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

Write 1 Bit 05h Code 05h: 1 Bit schreiben in Master-Ausgabe-Bereich 0x
 Eine Zustandsänderung erfolgt unter "Zustand Bit" mit folgenden Werten:
 "Zustand Bit" = 0000h → Bit = 0
 "Zustand Bit" = FF00h → Bit = 1

Kommandotelegramm

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

Antworttelegramm

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

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

Kommandotelegramm

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

Antworttelegramm

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

Write n Bits 0Fh

Code 0Fh: n Bit schreiben in Master-Ausgabe-Bereich 0x

Bitte beachten Sie, dass die Anzahl der Bits zusätzlich in Byte anzugeben sind.

Kommandotelegramm

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

Antworttelegramm

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

Write n Words 10h

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

Kommandotelegramm

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

Antworttelegramm

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

5.6.4 Modbus - Fehlermeldungen

Übersicht

Bei der Kommunikation unter Modbus gibt es folgende 2 Fehlerarten:

- Master bekommt keine gültigen Daten
- Slave antwortet mit einer Fehlermeldung

Master bekommt keine gültigen Daten

Antwortet der Slave nicht innerhalb der vorgegebenen Wartezeit oder ist ein Telegramm fehlerbehaftet, trägt der Master im Empfangs-Baustein eine Fehlermeldung in Klartext ein.

Folgende Fehlermeldungen sind möglich:

ERROR01 NO DATA	<i>Error no data</i> Innerhalb der Wartezeit wurde kein Telegramm empfangen.
ERROR02 D LOST	<i>Error data lost</i> Es liegen keine Daten vor, da entweder der Empfangspuffer voll ist oder ein Fehler im Empfangsteil aufgetreten ist.
ERROR03 F OVERF	<i>Error frame overflow</i> Das Telegrammende wurde nicht erkannt und die maximale Telegrammlänge überschritten.
ERROR04 F INCOM	<i>Error frame incomplete</i> Es wurde nur ein Teiltelegramm empfangen.
ERROR05 F FAULT	<i>Error frame fault</i> Die Checksumme innerhalb eines Telegramms ist nicht korrekt.
ERROR06 F START	<i>Error frame start</i> Das Startzeichen ist falsch. Dieser Fehler kann ausschließlich unter Modbus-ASCII auftreten.

Slave antwortet mit einer Fehlermeldung

Liefert der Slave einen Fehler zurück, so wird der Funktionscode, wie nachfolgend gezeigt, mit 80h "verodert" zurückgesendet:

DB11.DBD 0	DW#16#05900000 mit 05 →	Antworttelegramm Slave-Adresse 05h
	90 →	Funktionscode 90h (Fehlermeldung, da 10h "oder" 80h = 90h)
	0000 →	Die Restdaten sind irrelevant, da Fehler rückgemeldet wurde.