

- Einführung
- Für Ihre Sicherheit
- Systembeschreibung
- Art.-Nr. 57140 Kommunikationsmodul RS232
- Serielle Kommunikationsprotokolle
- Allgemeine Daten
- Technische Daten

# Handbuch CUBE20S Erweiterung

## Kommunikationsmodul RS232

**Dieses Dokument gilt für folgende Produkte:**

<b>Name</b>	<b>Art.-Nr.</b>
System CUBE20S	57140
Kommunikationsmodul RS232	
Kommunikationsmodule inkl. Sockel	

**Status**

Handbuchnummer: 57140\_hdb\_de\_10

Sprache: DE

Version: 1.0

Datum: 22.4.14

**Kontakt**

Murrelektronik GmbH

Falkenstraße 3

D-71570 Oppenweiler

Fon +49 (0) 7191 47-0

Fax +49 (0) 7191 47-491000

[info@murrelektronik.de](mailto:info@murrelektronik.de)

# Inhaltsverzeichnis

<b>1</b>	<b>Einführung</b>	<b>5</b>
1.1	Service und Support	5
1.2	Einführung/Zu diesem Dokument	6
1.3	Mitgeltende Unterlagen	6
1.4	Symbolik	6
1.5	Warenzeichen	7
<b>2</b>	<b>Für Ihre Sicherheit</b>	<b>8</b>
2.1	Zielgruppe	8
2.2	Bestimmungsgemäßer Gebrauch	8
2.3	Allgemeine Sicherheitshinweise	8
2.4	Hinweise zu elektrostatisch gefährdeten Baugruppen	9
2.5	EMV-Aufbaurichtlinien	10
2.6	Hinweise zu Ersatzteilen und Zubehör	10
2.7	Umweltgerechte Entsorgung	11
2.8	CE-Konformitätserklärung	11
2.9	Gewährleistung und Haftung	11
<b>3</b>	<b>Systembeschreibung</b>	<b>12</b>
3.1	System	12
3.2	Abmessungen	15
3.3	Montage	16
3.4	Demontage und Modultauch	20
3.5	Verdrahtung	26
3.6	Hilfe zur Fehlersuche - LEDs	32
<b>4</b>	<b>Art.-Nr. 57140 Kommunikationsmodul RS232</b>	<b>34</b>
4.1	Leistungsmerkmale	34
4.2	Aufbau	34
4.3	Schnelleinstieg	39
4.4	Ein-/Ausgabe-Bereich	40
4.5	Prinzip der Rückwandbus-Kommunikation	41
4.6	Rückwandbus-Kommunikation	52
4.7	Diagnosedaten	59
<b>5</b>	<b>Serielle Kommunikationsprotokolle</b>	<b>61</b>
5.1	Übersicht	61
5.2	ASCII	62
5.3	STX/ETX	66
5.4	3964(R)	70
5.5	Modbus	75
5.6	Modbus - Einsatz	78
5.7	Modbus - Funktionscodes	82
5.8	Modbus - Fehlermeldungen	86

---

<b>6</b>	<b>Allgemeine Daten</b>	<b>87</b>
<hr/>		
<b>7</b>	<b>Technische Daten</b>	<b>89</b>
7.1	Protokolle	89
<hr/>		
<b>8</b>	<b>Anhang</b>	<b>91</b>
8.1	Zubehör	91
8.2	Glossar	92
8.3	Rechtliche Hinweise	93

# 1 Einführung

## 1.1 Service und Support

<b>Vertrieb</b>	Unsere Vertriebsmitarbeiter im Innen- und Außendienst sowie unsere Techniker unterstützen Sie jederzeit.
<b>CONNECTIVITY-Systemberater</b>	<p>Unsere Systemberater sind Ihre kompetenten Ansprechpartner für die Entwicklung von CONNECTIVITY-Lösungen. Gemeinsam mit Ihnen ermitteln sie die optimalen Lösungen für Ihre elektrischen Installationen.</p> <p>Die CONNECTIVITY-Berater finden gemeinsam mit Ihnen Wege, die Ihnen dabei helfen, die Wettbewerbsfähigkeit Ihrer Maschinen und Anlagen dauerhaft zu stärken.</p>
<b>Customer Service Center (CSC)</b>	<p>Bei allen Fragen zu Installation und Inbetriebnahme helfen Ihnen die Mitarbeiter unseres Customer Service Center. Sie unterstützen Sie beispielsweise bei Problemen im Zusammenspiel von Produkten unterschiedlicher Hersteller für Hard- und Software.</p> <p>Dabei stehen zahlreiche Support-Tools und Messmöglichkeiten für Feldbus-systeme sowie für EMV-Einflüsse zur Verfügung.</p> <p>Rufen Sie uns unter +49 (0) 7191 47-2050 an oder senden Sie eine E-Mail an <a href="mailto:csc@murrelektronik.de">csc@murrelektronik.de</a>.</p>
<b>Service-Adressen</b>	Sie finden Ihren Ansprechpartner unter <a href="http://www.murrelektronik.com">www.murrelektronik.com</a>

## 1.2 Einführung/Zu diesem Dokument

### Funktion dieses Dokuments

Das Dokument beschreibt die Verwendung des Moduls Kommunikationsmodul RS232 aus dem System CUBE20S der Fa. Murrelektronik GmbH. Beschrieben wird der Aufbau, die Projektierung und die Anwendung.

## 1.3 Mitgeltende Unterlagen

### Mitgeltende Unterlagen

Unterlage	Fundstelle
Betriebsanleitung	Onlineshop der Murrelektronik GmbH

## 1.4 Symbolik

Dieses Dokument enthält Informationen und Hinweise, die Sie zur Wahrung der Sicherheit und zur Vermeidung von Personen- und Sachschäden beachten müssen. Sie sind wie folgt gekennzeichnet:



### **GEFAHR!**

#### Unmittelbare Gefahr

→ Nichtbeachten des Warnhinweises führt unmittelbar zum Tod oder schwerer Körperverletzung.



### **WARNUNG!**

#### Mögliche Gefahr

→ Nichtbeachten des Warnhinweises kann zum Tod oder zu schwerer Körperverletzung führen.



### **VORSICHT!**

#### Gefährdung mit geringem Risiko

→ Nichtbeachten des Warnhinweises führt zu leichten bis mittleren Körperverletzungen.

### **ACHTUNG**

#### Gefahr von Sachschäden

→ Nichtbeachten des Warnhinweises führt zu Sachschäden.



### **HINWEIS**

Weitere technische Informationen, und Hinweise der Murrelektronik GmbH.



### **EMPFEHLUNG**

Hinweise mit diesem Symbol sind Empfehlungen der Murrelektronik GmbH.



### Produkte und Zubehör

Dieses Symbol verweist auf Zubehör oder Produktempfehlungen.

### Handlungsanweisung

- ➔ Ein Pfeil kennzeichnet Handlungsanweisungen.
- ➔ Lesen und befolgen Sie die Handlungsanweisungen.
- 1 | Bei nummerierten Handlungsanweisungen muss die Reihenfolge unbedingt eingehalten werden.
- 2 | Lesen und befolgen Sie die Handlungsanweisungen.

## 1.5 Warenzeichen

In dieser Dokumentation werden die Warenzeichen folgender Firmen verwendet:

<b>Adobe Systems Corp.</b>	Adobe Acrobat Reader
<b>Microsoft Corp.</b>	Microsoft Windows 7, Windows Vista, Windows 2000, Windows XE/XP und Microsoft Internet Explorer
<b>PROFIBUS International (P.I.)</b>	PROFIBUS, PROFIBUS-DP
<b>PROFIBUS / PROFINET International (P.I.)</b>	PROFINET, PROFINET IO
<b>ODVA</b> Open DeviceNet Vendor Association	EtherNet/IP
<b>Beckhoff Automation GmbH</b>	EtherCAT
<b>CAN in AUTOMATION</b> - International Users and Manufacturers Group e.V.	CANopen
<b>Gould Inc. Corporation</b>	Modbus
<b>Siemens AG</b>	S5-200, S5-300, S5-400 S7-200, S7-300, S7-400

## 2 Für Ihre Sicherheit

### 2.1 Zielgruppe

<b>Anwender</b>	Das Handbuch ist geschrieben für Anwender mit Kenntnissen in der Automatisierungstechnik.
<b>Dokumentation</b>	Übergeben Sie das Handbuch allen Mitarbeitern in <ul style="list-style-type: none"><li>■ Projektierung</li><li>■ Installation</li><li>■ Inbetriebnahme</li><li>■ Betrieb</li></ul>

### 2.2 Bestimmungsgemäßer Gebrauch

<b>Bestimmungsgemäße Verwendung</b>	Das System CUBE20S ist konstruiert und gefertigt für: <ul style="list-style-type: none"><li>■ Kommunikation und Prozesskontrolle</li><li>■ Allgemeine Steuerungs- und Automatisierungsaufgaben</li><li>■ den industriellen Einsatz</li><li>■ den Betrieb innerhalb der in den technischen Daten spezifizierten Umgebungsbedingungen</li><li>■ den Einbau in einen Schaltschrank</li></ul>
<b>Vorhersehbarer Fehlgebrauch</b>	Das Gerät ist nicht zugelassen für den Einsatz: <ul style="list-style-type: none"><li>■ in explosionsgefährdeten Umgebungen (EX-Zone),</li><li>■ außerhalb von Schaltschränken.</li></ul>

### 2.3 Allgemeine Sicherheitshinweise

**Beachten Sie:**

- die einschlägigen Sicherheits- und Unfallverhütungsvorschriften;
- die genannten EG-Richtlinien oder sonstige länderspezifische Bestimmungen;
- allgemein anerkannte sicherheitstechnische Regeln;
- den Abschnitt 2.5 "EMV-Aufbaurichtlinien".

**ACHTUNG****Gerätedefekt!**

Durch unsachgemäße Eingriffe in die Hard- und Software kann das Gerät beschädigt werden.

- ➔ Nur Fachpersonal der Firma Murrelektronik GmbH darf in das Gerät eingreifen.
- ➔ Greifen Sie selbst nur so in das Gerät ein, wie es im Handbuch beschrieben ist.

**Vermeiden Sie Unfälle durch elektrische Spannung!**

- ➔ Halten Sie die 5 Sicherheitsregeln der Elektrotechnik ein!
- ➔ Trennen Sie das Gerät vom Spannungsversorgungsnetz.
- ➔ Führen Sie erst dann Arbeiten zur Installation oder Instandhaltung durch.



**Vermeiden Sie Personen- und Materialschäden durch Fehlfunktionen!**

- ➔ Sehen Sie externe Sicherungsschaltungen vor.
- ➔ Das Gerät darf die angegebenen Toleranzen weder über- noch unterschreiten.

**Vermeiden Sie undefinierte Zustände!**

- ➔ Wählen und installieren Sie Anschlussleitungen so, dass kapazitive und induktive Einstreuungen die Anlage nicht beeinträchtigen.
- ➔ Sichern Sie das Gerät gegen missbräuchliche und versehentliche Nutzung.

## 2.4 Hinweise zu elektrostatisch gefährdeten Baugruppen

**ACHTUNG****Überspannungen durch elektrostatische Entladung!**

Die Baugruppen können beschädigt werden.

- ➔ Auf ausreichende Erdung von Mensch und Arbeitsmittel achten!

**Handhabung**

Murrelektronik-Baugruppen enthalten hochintegrierten Bauelementen in MOS-Technik. Diese Bauelemente sind äußerst empfindlich gegenüber Überspannungen, die z.B. bei elektrostatischer Entladung entstehen. Gefährdeten Baugruppen sind mit dem nebenstehenden Symbol gekennzeichnet.

Das Symbol befindet sich auf Baugruppen, Baugruppenträgern oder Verpackungen und weist auf elektrostatisch gefährdete Baugruppen hin. Diese 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. Diese können zur Beschädigung von Bauelementen führen und die Funktionsweise der Baugruppen beeinträchtigen oder die Baugruppe unbrauchbar machen. Auf diese Weise beschädigte Baugruppen werden in den wenigsten Fällen sofort als fehlerhaft erkannt. Der Fehler kann sich erst nach längerem Betrieb einstellen.

Durch statische Entladung beschädigte Bauelemente können bei Temperaturänderungen, Erschütterungen oder Lastwechseln zeitweilige Fehler zeigen.

Nur durch konsequente Anwendung von Schutzeinrichtungen und verantwortungsbewusste Beachtung der Handhabungsregeln vermeiden Sie Funktionsstörungen und Ausfälle an elektrostatisch gefährdeten Baugruppen.

**Versand**

- ➔ Verwenden Sie für den Versand von elektrostatisch gefährdeten Baugruppen **immer** die Originalverpackung.

**Messen**

Beachten Sie bei Messungen an elektrostatisch gefährdeten Baugruppen folgende Punkte:

- ➔ Entladen Sie potenzialfreie Messgeräte kurzzeitig.
- ➔ Erden Sie die verwendeten Messgeräte.

## Ändern

Beachten Sie bei Änderungen an elektrostatisch gefährdeten Baugruppen:

➔ Verwenden Sie einen geerdeten LötKolben.

## 2.5 EMV-Aufbaurichtlinien

### Industrieller Einsatz

Das System CUBE20S ist ein nach dem neuesten Stand der Technik hergestelltes elektronisches Gerät. Sowohl der robuste mechanische Aufbau als auch die Ausführung der Elektronikkomponenten sind für den industriellen Einsatz ausgelegt.

Um einen störungsfreien Betrieb zu gewährleisten, sind beim Aufbau des Geräts in Anlagen Regeln zu beachten. Werden diese nicht beachtet, wird die hohe Stör- und Zerstörfestigkeit des Geräts teilweise wirkungslos.

Die Störfestigkeit der Gesamtanlage hängt maßgeblich vom korrekten Einbau, Aufbauort und der Verdrahtung ab.

- 1 | Prüfen Sie die Aufbauvorschriften des Herstellers der Steuerung für einen gesicherten Betrieb.
- 2 | Bringen Sie diese mit den Empfehlungen zum EMV-gerechten Aufbau in Einklang.
- 3 | Installieren Sie dann das System CUBE20S.

## 2.6 Hinweise zu Ersatzteilen und Zubehör

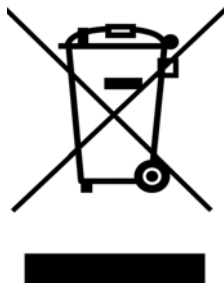
### Ersatzteile

- Verwenden Sie nur Original-Ersatzteile oder Ersatzteile anderer Hersteller, die von der Murrelektronik GmbH freigegeben wurden.
- Überprüfen Sie die Funktion des Geräts, wenn Sie Teile ersetzt haben.

### Zubehör

- Der Einsatz von Zubehör kann die Funktion des Geräts verändern. Verwenden Sie nur Zubehör, das von der Murrelektronik GmbH freigegeben wurde.
- Beachten Sie für die Montage des Zubehörs die diesem beiliegenden Anleitungen.

## 2.7 Umweltgerechte Entsorgung



### Entsorgung

Werfen Sie elektrische Geräte, Batterien oder Akkus nicht in den Hausmüll! Das Produkt kann im Falle einer Entsorgung unentgeltlich an Murrelektronik GmbH zurückgesendet werden. Dies gilt auch für die Originalverpackung und ggf. Batterien oder Akkus.

### Rücksendung

→ Kennzeichnen Sie das Produkt und die Verpackung mit "**Zur Entsorgung**".

→ Verpacken Sie das Produkt.

→ Senden Sie das Paket an:

Murrelektronik GmbH

Falkenstraße 3

D-71570 Oppenweiler

Wir stellen eine Entsorgung nach den deutschen gesetzlichen Vorschriften sicher. Für den Transport zur Rückgabestelle ist der letzte Besitzer bis zum Bestimmungsort verantwortlich.

## 2.8 CE-Konformitätserklärung



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

- 2004/108/EG Elektromagnetische Verträglichkeit
- 2011/65/EU RoHS

## 2.9 Gewährleistung und Haftung

### Gewährleistungs- und Haftungsansprüche

Gewährleistungs- und Haftungsansprüche gehen verloren, wenn

- das Produkt nicht bestimmungsgemäß verwendet wird,
- Schäden darauf zurückzuführen sind, dass Handbuch und Betriebsanleitung nicht beachtet wurden,
- oder das Personal nicht sachkundig war/ist.

## 3 Systembeschreibung

### 3.1 System

#### Übersicht

Das System CUBE20S ist ein modular aufgebautes Automatisierungssystem für die Montage auf einer 35 mm-Profilschiene. Mittels der Erweiterungsmodule in 2-, 4- und 8-Kanalausführung können Sie dieses System passgenau an Ihre Automatisierungsaufgaben anpassen.

Der Verdrahtungsaufwand ist gering, da die 24 V DC-Spannungsversorgung im Rückwandbus integriert ist. Defekte Elektronik-Module können Sie bei stehender Verdrahtung austauschen.

Durch Einsatz der farblich abgesetzten Power-Module können Sie innerhalb des Systems weitere Potenzialbereiche für die 24 V DC-Spannungsversorgung definieren, bzw. die Elektronikversorgung um 2 A erweitern.

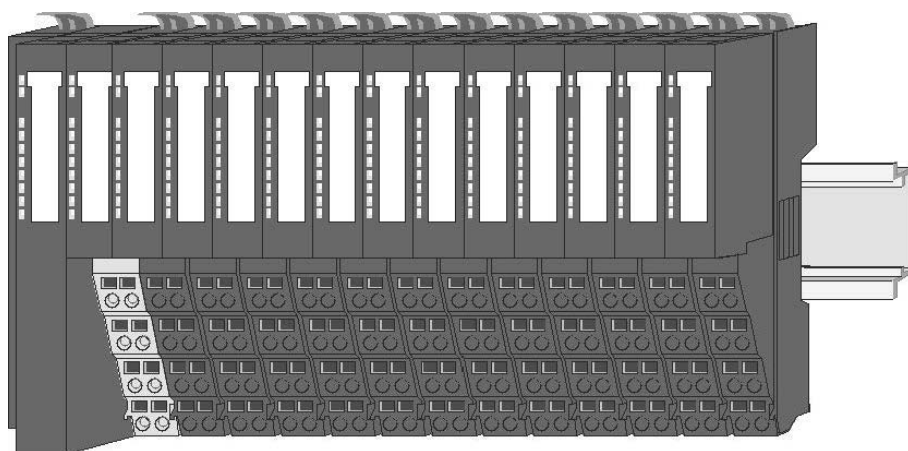


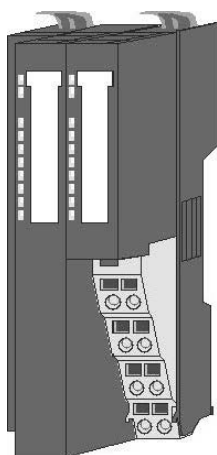
Abb. 3-1: System CUBE20S

## Komponenten

Das System CUBE20S besteht aus folgenden Komponenten:

- Busknoten
- Erweiterungsmodule
- Power-Module
- Zubehör

## Busknoten

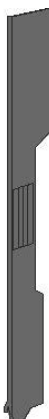


Beim Busknoten sind Bus-Interface und Power-Modul in ein Gehäuse integriert. Das Bus-Interface bietet Anschluss an ein übergeordnetes Bussystem. Über das Power-Modul zur Spannungsversorgung werden sowohl das Bus-Interface als auch die Elektronik der angebunden Erweiterungsmodule versorgt. Die 24 V DC-Spannungsversorgung für die angebunden Erweiterungsmodule erfolgt über einen weiteren Anschluss am Power-Modul.

Durch die Montage von bis zu 64 Erweiterungsmodulen am Busknoten werden diese elektrisch verbunden, d.h.

- sie sind am Rückwandbus eingebunden,
- die Elektronik-Module werden versorgt,
- jedes Erweiterungsmodul ist an die 24 V DC-Spannungsversorgung angeschlossen.

## Busblende

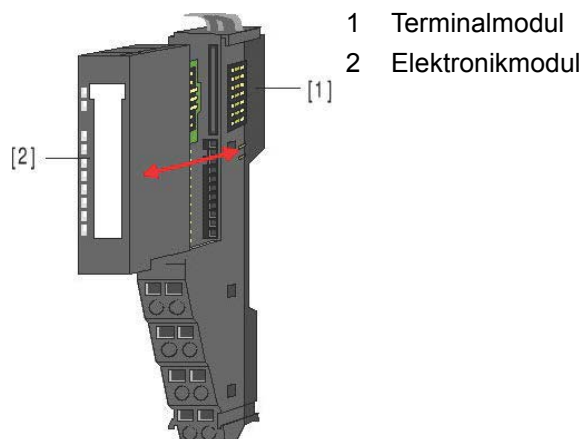


Zu jedem Busknoten gehört zum Schutz der Kontakte eine Busblende.

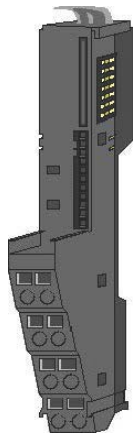
- ➔ Entfernen Sie vor der Montage von CUBE20S-Modulen die Busblende am Busknoten.
- ➔ Montieren Sie zum Schutz der Kontakte die Busblende am äußersten Modul.

## Erweiterungsmodule

Jedes Erweiterungsmodul besteht aus einem Terminal- und einem Elektronikmodul.



### Terminalmodul

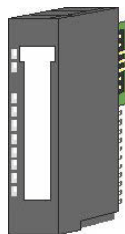


Das Terminalmodul enthält folgende funktionale Elemente:

- einen Schiebemechanismus zur Aufnahme des Elektronikmoduls,
- den Rückwandbus mit Spannungsversorgung für die Elektronik,
- die Anbindung an die 24 V DC-Spannungsversorgung,
- den treppenförmigen Klemmenblock für die Verdrahtung,
- ein sicheres Verriegelungssystem zur Fixierung auf einer Tragschiene.

Mit dieser Verriegelung können Sie Ihr CUBE20S-System außerhalb des Schaltschranks aufbauen und später als Gesamtsystem im Schaltschrank montieren.

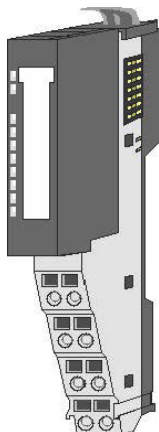
### Elektronikmodul



Über das Elektronikmodul wird die Funktionalität eines Erweiterungsmoduls definiert.

- Im Fehlerfall kann das defekte Elektronikmodul gegen ein funktionsfähiges Modul getauscht werden, hierbei bleibt die Verdrahtung bestehen.
- Auf der Frontseite befinden sich LEDs zur Statusanzeige.
- Zur einfachen Verdrahtung befinden sich bei jedem Elektronikmodul auf der Front und an der Seite entsprechende Anschlussbilder.

### Power-Module



Power-Module versorgen das CUBE20S-System mit Spannung. Die Power-Module sind entweder im Busknoten integriert oder können zwischen die Erweiterungsmodule gesteckt werden.

Je nach Art des Power-Moduls können Potenzialgruppen der 24 V DC-Spannungsversorgung definiert bzw. die Elektronikversorgung um 2 A erweitert werden.

Zur besseren Erkennung sind die Power-Module farblich von den Erweiterungsmodulen abgesetzt.

## 3.2 Abmessungen

### Maße Busknoten

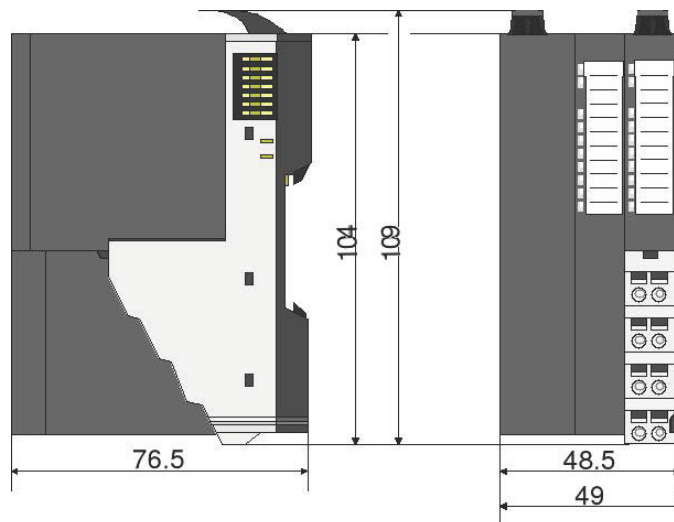


Abb. 3-2: Maße Busknoten

### Maße Erweiterungsmodul

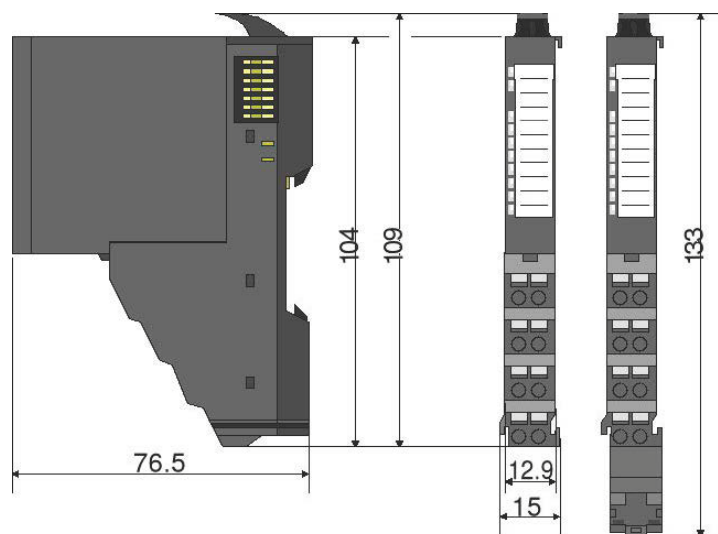


Abb. 3-3: Maße Erweiterungsmodul

### Maße Elektronikmodul

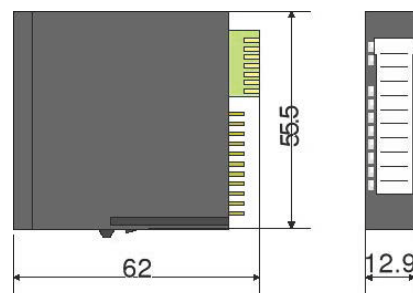


Abb. 3-4: Maße Elektronikmodul

## 3.3 Montage



### HINWEIS

Sie können die Module einzeln oder als Block auf der Profilschiene montieren. Beachten Sie bei der Blockmontage: **Alle** Verriegelungshebel müssen geöffnet sein.

### 3.3.1 Allgemeine Hinweise

Die einzelnen Module werden direkt auf eine Profilschiene montiert. Über die Verbindung mit dem Rückwandbus werden Elektronik- und Spannungsversorgung angebunden.

Bedingungen:

- Max. Anzahl steckbarer Module: 64
- Max. Summenstrom der Elektronikversorgung: 3 A

Ein **Power-Modul Sensor/Aktor/Bus Art.-No. 57131** erweitert den Strom für die Elektronikversorgung um 2 A. Näheres hierzu finden Sie im Abschnitt 3.5 "Verdrahtung".

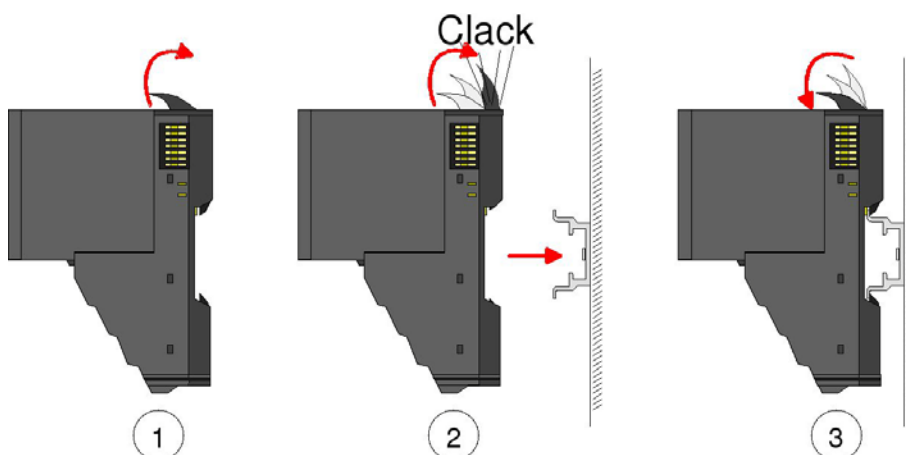


Abb. 3-5: Modul montieren

### 3.3.2 Funktionsprinzip der Verriegelung

#### **Modul einsetzen und verriegeln**

- ✂ Das Terminal-Modul besitzt einen Verriegelungshebel an der Oberseite.
- 1 | Drücken Sie zur Montage und Demontage diesen Hebel nach oben, bis er hörbar einrastet.
- 2 | Stecken Sie das zu montierende Modul an das zuvor gesteckte Modul
- 3 | Schieben Sie das Modul, geführt durch die Führungsleisten an der Ober- und Unterseite, auf die Profilschiene.
- 4 | Klappen Sie den Verriegelungshebel nach unten.

*Das Modul ist auf der Profilschiene fixiert.*



### 3.3.3 Austausch eines Elektronikmoduls

#### Demontage

- ✓ Das Elektronikmodul besitzt an der Unterseite einen Verriegelungshebel.
- 1 | Drücken Sie zur Demontage den Verriegelungshebel nach oben (Press).
- 2 | Ziehen Sie das Elektronikmodul nach vorne ab (Pull).

Das Elektronikmodul ist entfernt.

#### Montage

- ✓ Das Elektronikmodul besitzt an der Unterseite einen Verriegelungshebel.
- ➔ Schieben Sie das Elektronikmodul in der Führungsschiene in das Terminal-Modul.

Das Elektronikmodul rastet an der Unterseite hörbar ein.

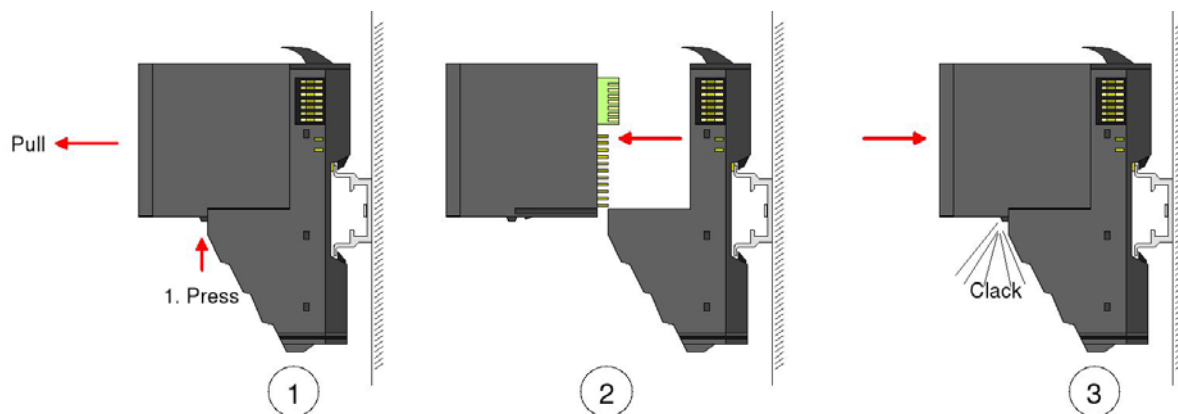


Abb. 3-6: Elektronikmodul demontieren und montieren

### 3.3.4 Montage der Profilschiene

- ➔ Montieren Sie die Profilschiene mit den notwendigen Abständen (siehe Abb. 3-7: "Montageabstände").

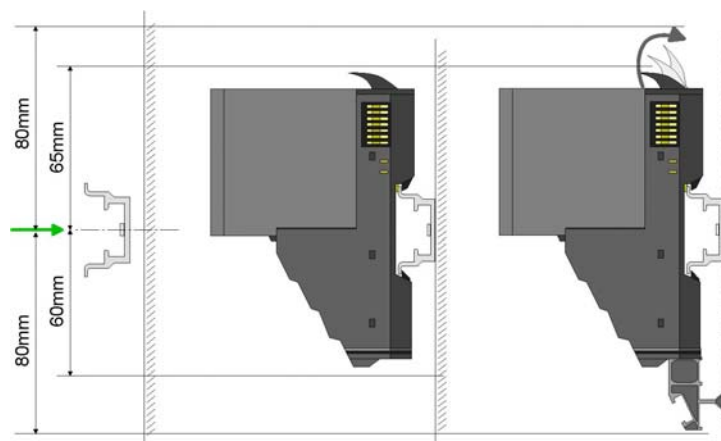


Abb. 3-7: Montageabstände

### 3.3.5 Montage des Busknoten

- ✓ Der Systemaufbau beginnt links mit dem Busknoten.
- 1 | Klappen Sie beide Verriegelungshebel des Busknotens nach oben.
- 2 | Stecken Sie den Busknoten auf die Profilschiene.
- 3 | Klappen Sie beide Verriegelungshebel des Busknotens nach unten.
- 4 | Ziehen Sie die rechte Busblende nach vorn ab.
- 5 | Bewahren Sie die Busblende als Abschluss des Systems auf.

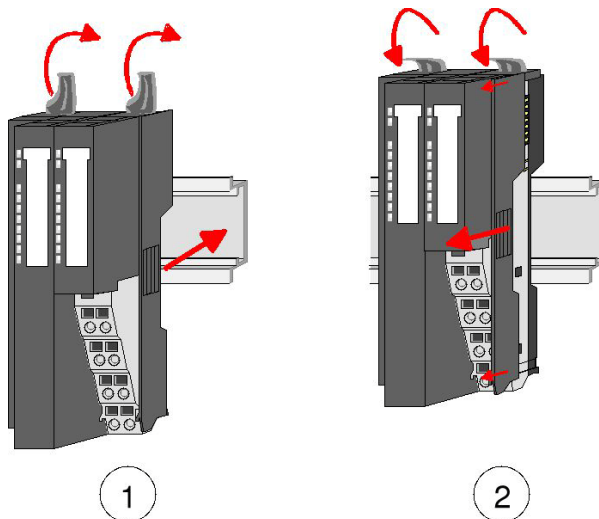


Abb. 3-8: Montage Busknoten

### 3.3.6 Montage der Erweiterungsmodule

- 1 | Klappen Sie den Verriegelungshebel des Erweiterungsmoduls nach oben.
- 2 | Stecken Sie das Erweiterungsmodul auf die Profilschiene.
- 3 | Schieben Sie das Erweiterungsmodul an den Busknoten bzw. an das letzte Erweiterungsmodul.
- 4 | Klappen Sie den Verriegelungshebel des Erweiterungsmoduls nach unten.
- 5 | Montieren Sie alle Erweiterungsmodule wie beschrieben.

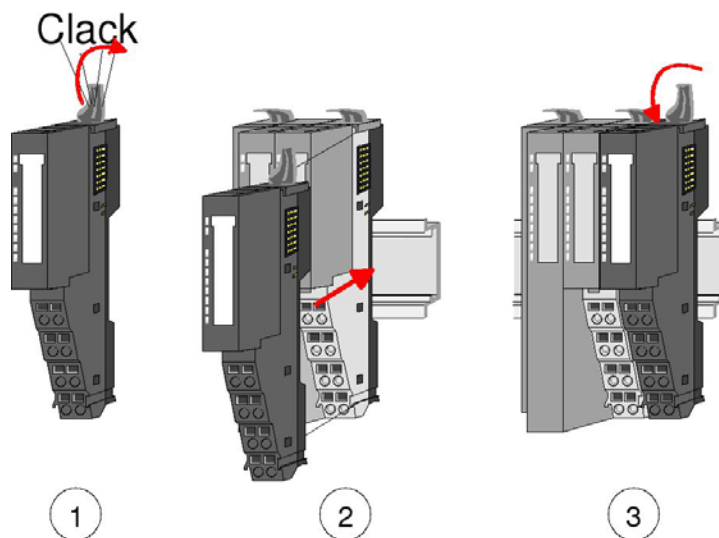


Abb. 3-9: Montage Erweiterungsmodul

### 3.3.7 Montage der Busblende

- ✓ Voraussetzung: Das System ist vollständig montiert.
- ➔ Stecken Sie zum Schutz der Bus-Kontakte die Busblende an das äußerste Modul.

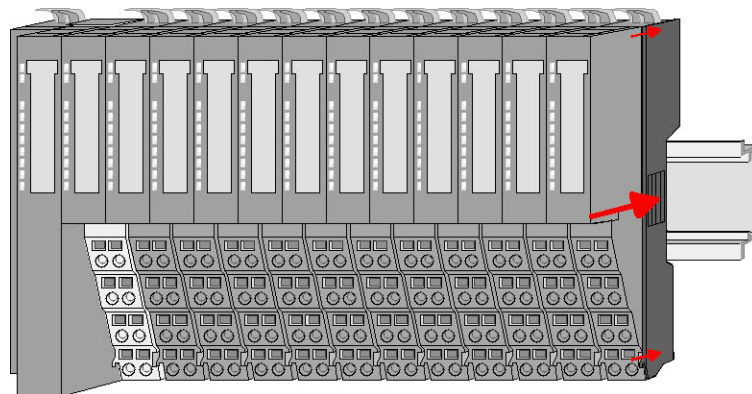


Abb. 3-10: Montage Busblende

## 3.4 Demontage und Modultausch

### 3.4.1 Vorgehensweise

**Beachten Sie bei der Demontage und beim Austausch eines Moduls, einer Modulgruppe:**

- 1 | Entfernen Sie das Elektronikmodul rechts neben dem Modul oder der Modulgruppe.
- 2 | Demontieren/tauschen Sie das Modul oder die Modulgruppe.
- 3 | Stecken Sie das Elektronikmodul ein.

### 3.4.2 Austausch eines Elektronikmoduls

#### **Demontage**

✓ Das Elektronikmodul besitzt an der Unterseite einen Verriegelungshebel.

- 1 | Drücken Sie zur Demontage den Verriegelungshebel nach oben (Press).
- 2 | Ziehen Sie das Elektronikmodul nach vorne ab (Pull).

*Das Elektronikmodul ist entfernt.*

#### **Montage**

✓ Das Elektronikmodul besitzt an der Unterseite einen Verriegelungshebel.

- ➔ Schieben Sie das Elektronikmodul in der Führungsschiene in das Terminal-Modul.

*Das Elektronikmodul rastet an der Unterseite hörbar ein.*

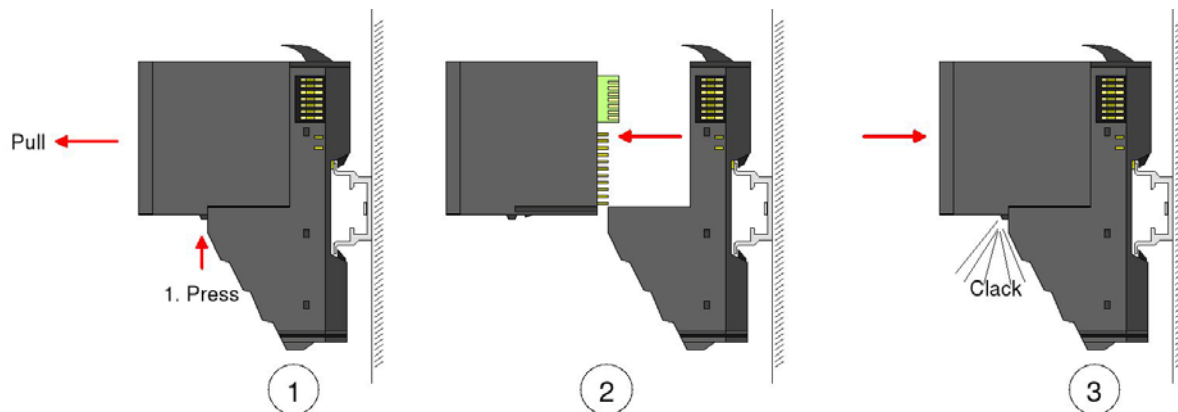


Abb. 3-11: Elektronikmodul demontieren und montieren

### 3.4.3 Austausch eines Moduls

#### Demontage

- 1 | Entfernen Sie - falls vorhanden - die Verdrahtung am Modul. Näheres hierzu finden Sie im Abschnitt **Verdrahtung**.
- 2 | Entriegeln Sie das rechts daneben befindliche Elektronikmodul an der Unterseite.
- 3 | Ziehen Sie das Elektronikmodul nach vorne ab.
- 4 | Klappen Sie den Verriegelungshebel des zu tauschenden Moduls nach oben.
- 5 | Ziehen Sie das Modul nach vorne ab.

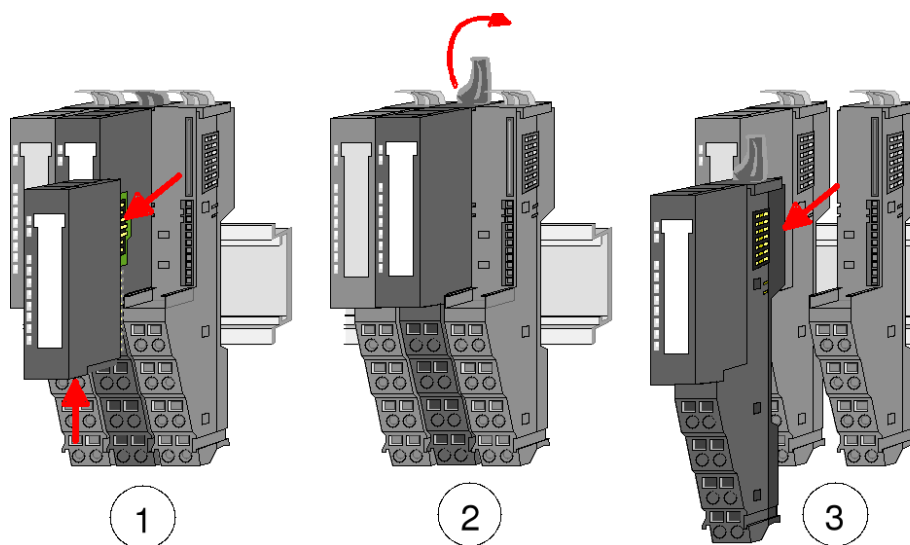


Abb. 3-12: Demontage eines Moduls

### Montage des neuen Moduls

- 1 | Klappen Sie den Verriegelungshebel des Moduls nach oben.
- 2 | Stecken Sie das Modul in die Lücke zwischen den Modulen.
- 3 | Schieben Sie das Modul, geführt durch die Führungsleisten auf beiden Seiten, auf die Profilschiene.
- 4 | Klappen Sie den Verriegelungshebel des Moduls nach unten.
- 5 | Stecken Sie das Elektronikmodul ein.

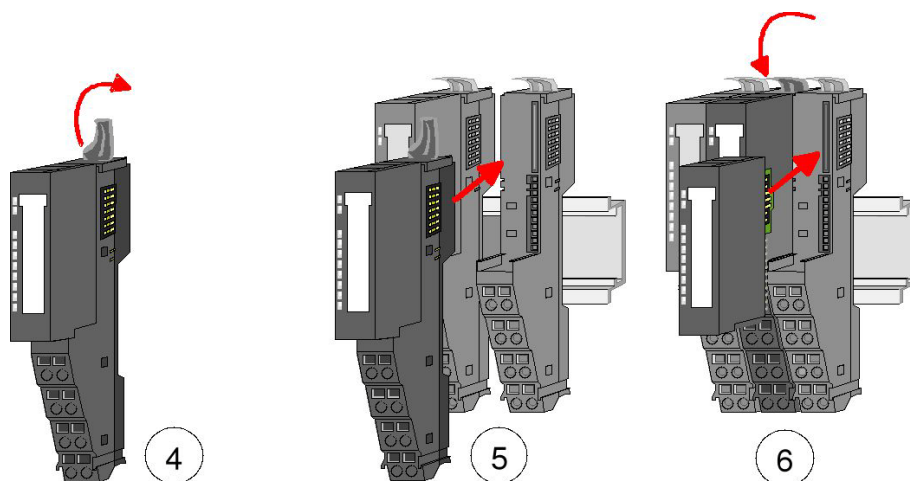


Abb. 3-13: Montage des neuen Moduls

### 3.4.4 Austausch eines Busknoten

#### Demontage

**VORSICHT!**

**Powermodul und Bus-Interface sind eine Einheit!**

Die Module werden durch Trennung zerstört.

→ Powermodul und Bus-Interface nicht voneinander trennen!

- 1 | Entfernen Sie - falls vorhanden - die Verdrahtung am Busknoten. Näheres hierzu finden Sie im Abschnitt **Verdrahtung**.
- 2 | Entriegeln Sie das rechts daneben befindliche Elektronikmodul an der Unterseite.
- 3 | Ziehen Sie das Elektronikmodul nach vorne ab.
- 4 | Klappen Sie die Verriegelungshebel des Busknotens nach oben.
- 5 | Ziehen Sie den Busknoten nach vorne ab.

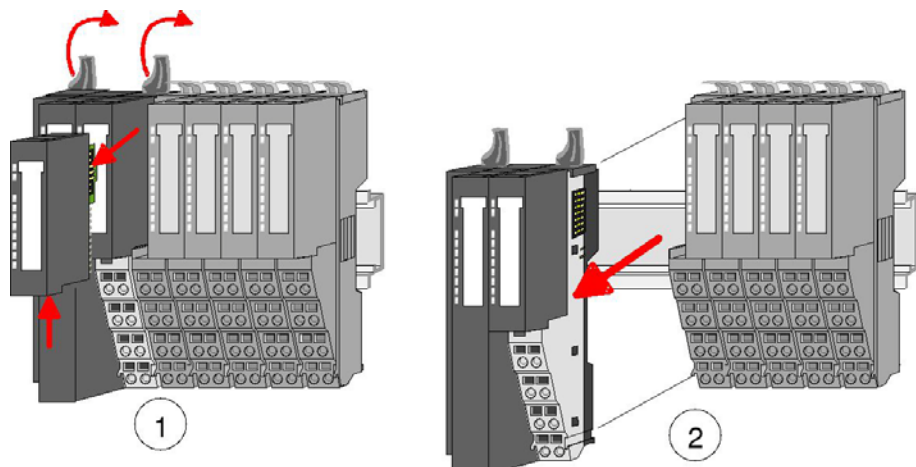


Abb. 3-14: Demontage des Busknoten

### Montage des neuen Busknoten

- 1 | Klappen Sie die Verriegelungshebel des Busknoten nach oben.
- 2 | Stecken Sie den Busknoten an das linke Modul.
- 3 | Schieben Sie den Busknoten, geführt durch die Führungsleisten, auf die Profilschiene.
- 4 | Klappen Sie die Verriegelungshebel nach unten.
- 5 | Stecken Sie das Elektronikmodul ein.

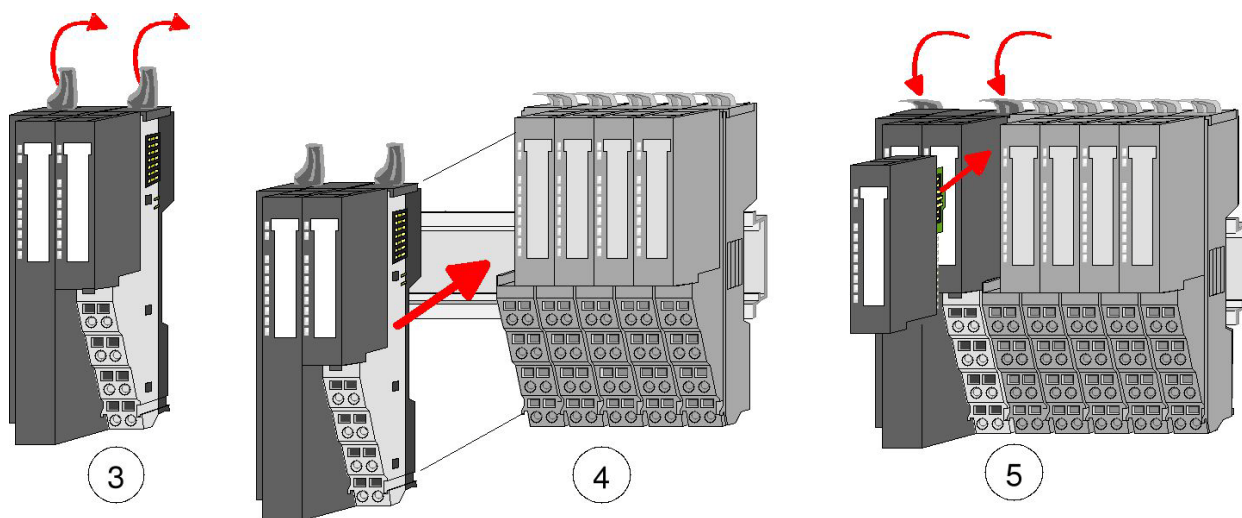


Abb. 3-15: Montage des neuen Busknoten



### 3.4.5 Austausch einer Modulgruppe

#### Demontage

- 1 | Entfernen Sie - falls vorhanden - die Verdrahtung an der Modulgruppe. Näheres hierzu finden Sie im Abschnitt **Verdrahtung**.
- 2 | Entriegeln Sie das rechts daneben befindliche Elektronikmodul an der Unterseite.
- 3 | Ziehen Sie das Elektronikmodul nach vorne ab.
- 4 | Klappen Sie die Verriegelungshebel der Modulgruppe nach oben.
- 5 | Ziehen Sie die Modulgruppe nach vorne ab.

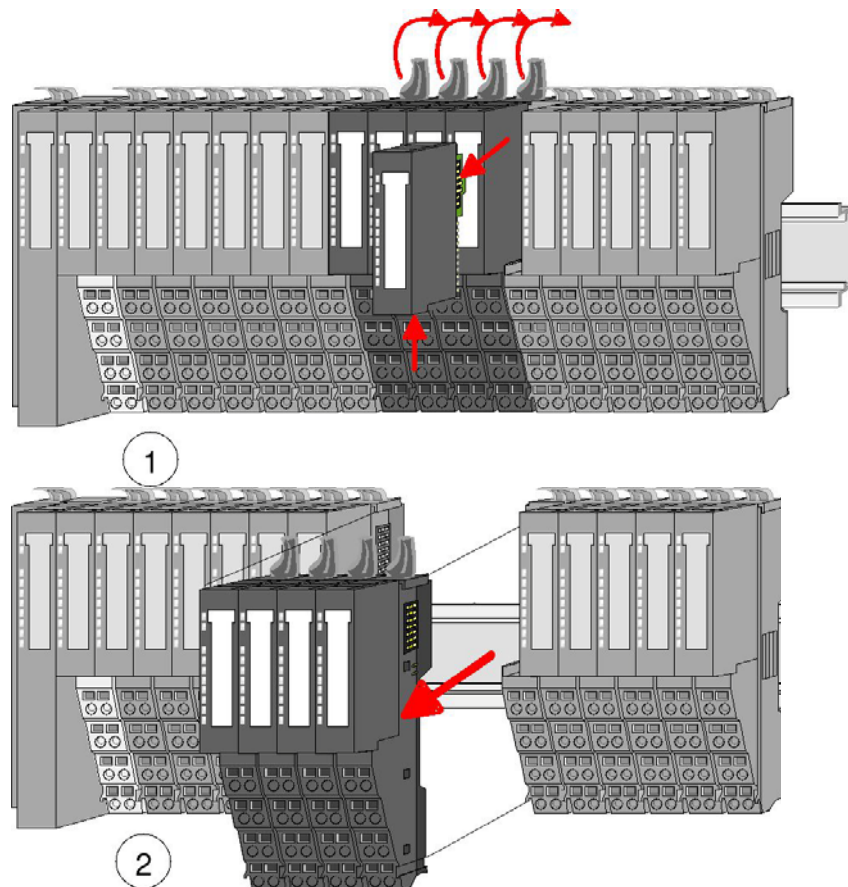


Abb. 3-16: Demontage der Modulgruppe

**Montage der neuen Modulgruppe**

- 1 | Klappen Sie die Verriegelungshebel der Modulgruppe nach oben.
- 2 | Stecken Sie die Modulgruppe in die Lücke zwischen den Modulen.
- 3 | Schieben Sie die Modulgruppe, geführt durch die Führungsleisten auf beiden Seiten, auf die Profilschiene.
- 4 | Klappen Sie die Verriegelungshebel der Modulgruppe nach unten.
- 5 | Stecken Sie das Elektronikmodul ein.

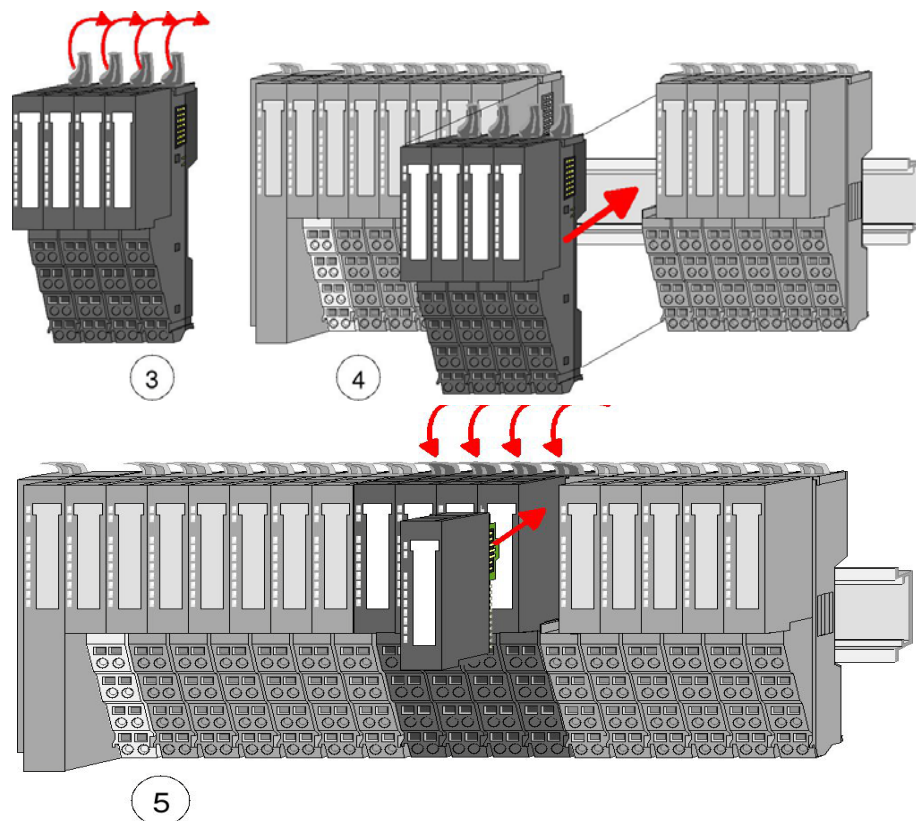


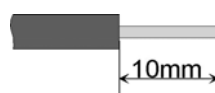
Abb. 3-17: Montage der Modulgruppe

## 3.5 Verdrahtung

### 3.5.1 Federklemmtechnik

**Anschlussklemmen**

Zur Verdrahtung werden Anschlussklemmen mit Federklemmtechnik eingesetzt. Die Federklemmtechnik ermöglicht einen schnellen und einfachen Anschluss der Signal- und Versorgungsleitungen. Diese Verbindungsart ist erschütterungssicher.

**Leistungsdaten**

 $U_{\max}: 240 \text{ V AC} / 30 \text{ V DC}$ 
 $I_{\max}: 10 \text{ A}$ 

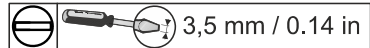
 Querschnitt:  $0,08 \dots 1,5 \text{ mm}^2$  (AWG 28 ... 16)

Abisolierlänge: 10 mm

### 3.5.2 Vorgehensweise

#### Verdrahten

✂ Werkzeug: geeigneter Schraubendreher



✂ Drahtquerschnitt: 0,08 mm<sup>2</sup> ... 1,5 mm<sup>2</sup> (AWG 28 ... 16)

- 1 | Stecken Sie den Schraubendreher leicht schräg in die rechteckige Öffnung (Abb. 3-18: 1).
- 2 | Drücken und halten Sie den Schraubendreher entgegengesetzt zur runden Öffnung. Die Kontaktfeder ist geöffnet (Abb. 3-18: 2).
- 3 | Führen Sie den abisolierten Draht durch die runde Öffnung (Abb. 3-18: 2).
- 4 | Entfernen Sie den Schraubendreher (Abb. 3-18: 3).

*Der Draht ist über einen Federkontakt sicher mit der Anschlussklemme verbunden.*

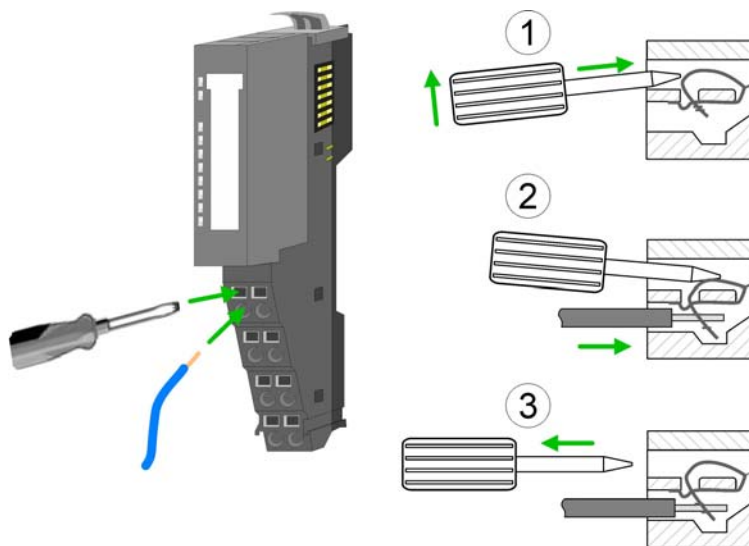


Abb. 3-18: Federklemmtechnik

### 3.5.3 Standardverdrahtung

#### Standardverdrahtung

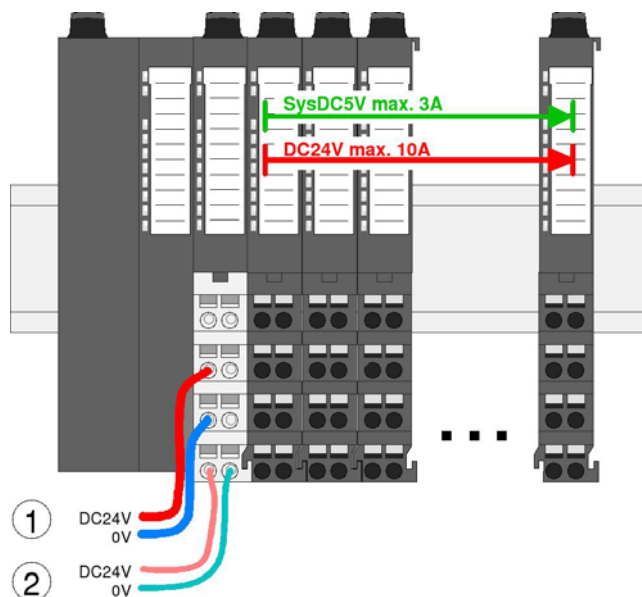


Abb. 3-19: Standardverdrahtung

- 1 24 V DC für Spannungsversorgung I/O-Ebene (max. 10 A)
- 2 24 V DC für Elektronikversorgung Busknoten und I/O-Ebene

### 3.5.4 Absicherung



#### WARNUNG!

**Die Spannungsversorgung ist intern nicht abgesichert.**

Sie kann durch zu hohe Ströme zerstört werden.

→ Die Spannungsversorgung extern mit einer Sicherung oder einem Leitungsschutzschalter absichern!



#### HINWEIS

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!

#### Externe Absicherung

	externe Sicherung	Leistungsschutzschalter (alternativ)	Bemerkung
Spannungsversorgung	10 A (flink)	10 A Charakteristik Z	bis max. Strom 10 A
Elektronikversorgung Busknoten und I/O-Ebene	2 A (flink)	2 A Charakteristik Z	Empfehlung!
Elektronikversorgung I/O-Ebene Powermodul Art.-No. 57131	1 A (flink)	1 A Charakteristik Z	Empfehlung!

Tab. 3-1: Absicherung der Versorgungsspannungen

### 3.5.5 Einsatz von Powermodulen

#### Zustand der Elektronikversorgung

Nach dem Einschalten des CUBE20S-Systems leuchtet an jedem Modul die RUN- bzw. MF-LED.

Ist der Summenstrom für die Elektronikversorgung größer als 3 A, werden die LEDs nicht mehr angesteuert. In diesem Fall muss zwischen die Erweiterungsmodule das Powermodul Art.-No. 57130 eingesetzt werden.



#### HINWEIS

Zur Sicherstellung der Spannungsversorgung dürfen die Powermodule beliebig gemischt eingesetzt werden.

#### Powermodul Art.-No. 57130

Setzen Sie dieses Powermodul ein:

- wenn die 10 A für die Leistungsversorgung nicht mehr ausreichen.
- wenn Sie Potenzialgruppen bilden möchten.

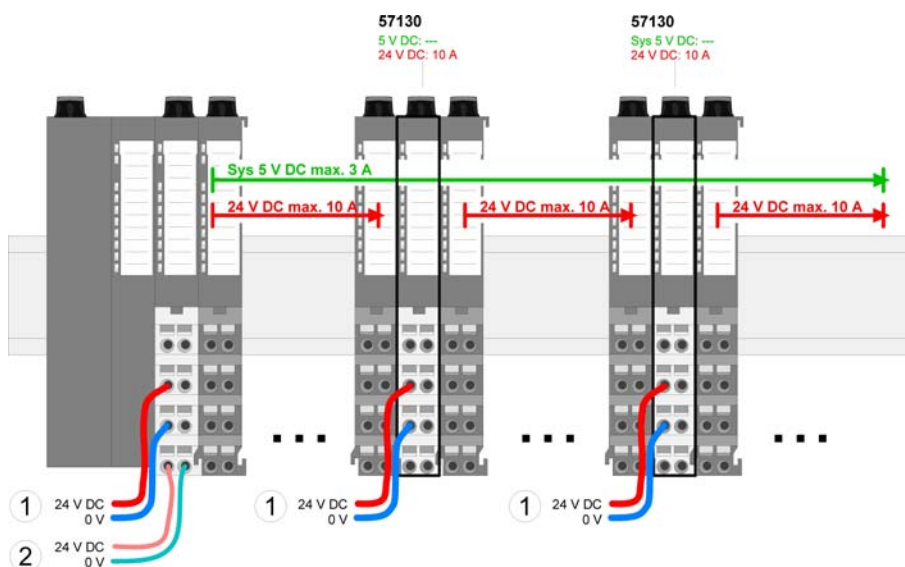


Abb. 3-20: Powermodul Art.-No. 57130

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

**Powermodul Art.-No.  
57131**

Setzen Sie das Powermodul ein, wenn 3 A für die Elektronikversorgung am Rückwandbus nicht ausreichen.

Zusätzlich erhalten Sie eine neue Potenzialgruppe für die 24 V DC-Leistungsversorgung mit max. 4 A.

Durch Einsatz des Powermoduls können Sie am nachfolgenden Rückwandbus Module mit einem maximalen Summenstrom von 2 A stecken. Danach müssen Sie wieder ein Powermodul stecken.

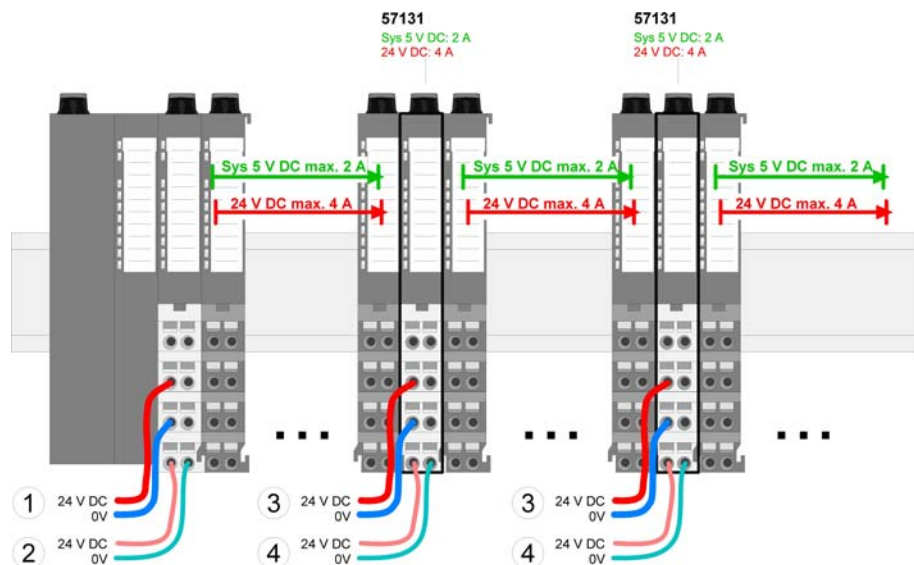


Abb. 3-21: Powermodul Art.-No. 57131

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

### 3.5.6 Schirm auflegen



#### HINWEIS

Zur Schirmauflage ist die Montage von Schirmschienenträgern erforderlich (siehe **Zubehör**).

#### Schirm auflegen

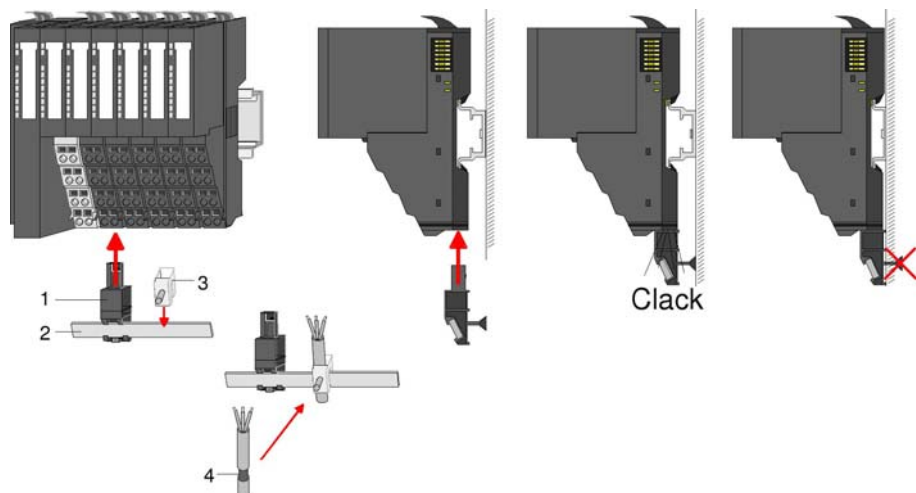


Abb. 3-22: Schirm auflegen

- 1 Schirmschienenträger
- 2 Schirmschiene (10 mm x 3 mm)
- 3 Schirmanschlussklemme
- 4 Leitungsschirm







#### Leitungsschirm auflegen

- ✓ Die Schirmschienenträger und die Schirmschiene sind gesteckt.
- ➔ Legen Sie die Leitungen mit dem entsprechend abisolierten Leitungsschirm auf.
- ➔ Verbinden Sie die Schirmanschlussklemme mit der Schirmschiene.

### 3.6 Hilfe zur Fehlersuche - LEDs

#### Allgemein

Jedes Modul besitzt auf der Frontseite die LEDs **RUN** und **MF**. Mit diesen LEDs ermitteln Sie Fehler in Ihrem System bzw. fehlerhafte Module.

Bezeichnung	Anzeige	LED Zustand
RUN-LED		aus
		grün
		grün blinkend
MF-LED		aus
		rot
		rot blinkend

Tab. 3-2: Zustandsanzeigen der LEDs

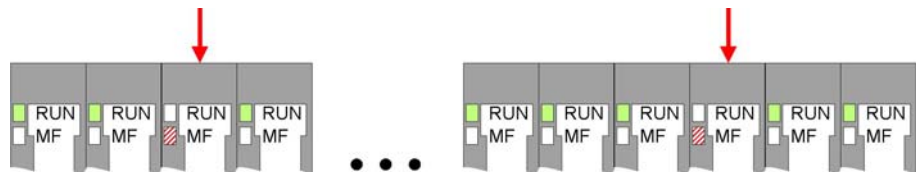
#### Summenstrom der Elektronikversorgung überschritten



Verhalten der LEDs nach dem Einschalten:	Die <b>RUN-LED</b> bleibt an allen Modulen aus. Die <b>MF-LED</b> leuchtet nur an einzelnen Modulen.
Ursache:	Der Summenstrom für die Elektronikversorgung übersteigt den maximalen Strom.
Abhilfe:	Fügen Sie das Power-Modul Art.-No. 57131 ein. Näheres hierzu finden Sie im Abschnitt <b>Verdrahtung</b> .

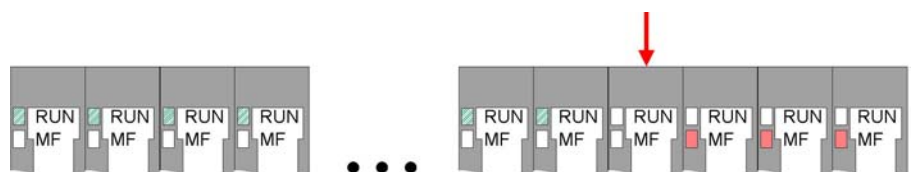


## Konfigurationsfehler



Verhalten der LEDs nach dem Einschalten:	Die <b>RUN-LED</b> ist an einem oder mehreren Modulen aus. Die <b>MF-LED</b> blinkt an diesen Modulen.
Ursache:	Das Modul, an dem die <b>MF-LED</b> blinkt, entspricht nicht der aktuellen Konfiguration.
Abhilfe:	Stimmen Sie Konfiguration und Hardware-Aufbau aufeinander ab.

## Ausfall eines Moduls



Verhalten der LEDs nach dem Einschalten:	Die <b>RUN-LEDs</b> blinken bis zum Modul links vom defekten Modul. Bei den nachfolgenden Modulen ist die <b>RUN-LED</b> aus. Die <b>MF-LEDs</b> sind bis zum Modul links vom defekten Modul aus. Bei den nachfolgenden Modulen leuchtet die <b>MF-LED</b> .
Ursache:	Das Modul rechts von den blinkenden Modulen ist defekt.
Abhilfe:	Ersetzen Sie das defekte Modul.

## 4 Art.-Nr. 57140 Kommunikationsmodul RS232

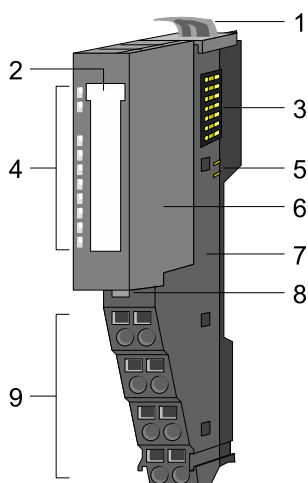
### 4.1 Leistungsmerkmale

#### Eigenschaften

- Serielle RS232-Schnittstelle (potenzialgetrennt zum Rückwandbus)
- Übertragungsrate von 150 Bit/s bis maximal 115,2 kBit/s
- Serielle Kommunikation über RS232
- Protokolle
  - ASCII
  - STX/ETX
  - 3964(R)
  - Modbus (Master/Slave mit ASCII und RTU short & long) mit einer Telegrammlänge von 250 Byte
- Bis zu 250 Telegramme (1024 Byte Empfangs- bzw. Sendepuffer)
- Zeichenverzugszeit im ms-Raster parametrierbar
- Parametrierung über 17 Byte Parameterdaten
- Modem Signals Management DTR-DSR-DCD

### 4.2 Aufbau











#### 57140





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

## Statusanzeige



RUN	MF	Beschreibung
		Buskommunikation ist OK Modulstatus ist OK
		Buskommunikation ist OK Modulstatus meldet Fehler
		Buskommunikation nicht möglich Modulstatus meldet Fehler
		Fehler Versorgungsspannung Bus
		Konfigurationsfehler (siehe 3.6 Hilfe zur Fehlersuche - LEDs, Seite 32)

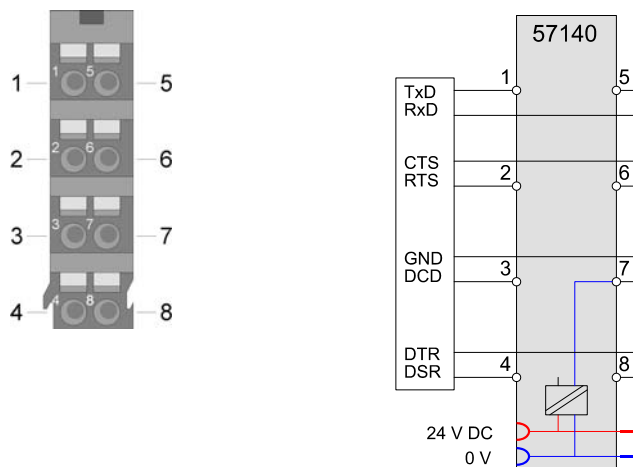
Tab. 4-1: Zustandsanzeigen der RUN und MF-LED

LED	Farbe	Beschreibung
TxD		Daten senden (transmit data)
RxD		Daten empfangen (receive data)
IF		<b>Modbus:</b> interner Fehler <b>andere Protokolle:</b> Leistungsunterbrechung, Überlauf, Paritätsfehler oder Zeichenrahmenfehler

Tab. 4-2: Zustandsanzeige der LEDs TxD, RxD, IF

**Anschlussklemme**

➔ Schließen Sie Drähte mit einem Querschnitt von 0,08 mm<sup>2</sup> bis 1,5 mm<sup>2</sup> an.



Pos.	Funktion	Typ	Beschreibung
1	TxD	Ausgang	Sendedaten
2	RTS	Ausgang	Request to send RTS auf 1: Kommunikationsprozessor ist sendebereit RTS auf 0: Kommunikationsprozessor sendet nicht
3	DCD	Eingang	Data carrier detect Daten können empfangen werden
4	DSR	Eingang	Data set ready Modem signalisiert Betriebsbereitschaft
5	RxD	Eingang	Empfangsdaten
6	CTS	Eingang	Clear to send Kommunikationsprozessor darf Daten senden
7	GND_ISO	Ausgang	Signal-Nullbezugspunkt (isoliert)
8	DTR	Ausgang	Data Terminal Ready Kommunikationsprozessor ist betriebsbereit

Tab. 4-3: Belegung der Anschlussklemmen

**RS232-Schnittstelle****HINWEIS**

**RI (Ring indicator)** - Der Kommunikationsprozessor verwendet das Klingelzeichen RI vom Modem nicht!

- Logische Zustände als Spannungspegel
- Punkt-zu-Punkt-Kopplung mit serieller Vollduplex-Übertragung
- Datenübertragung bis 15 m Entfernung
- Datenübertragungsrate bis 115,2 kBit/s

RS232-  
Verkabelung ohne  
Hardware-  
Handshake

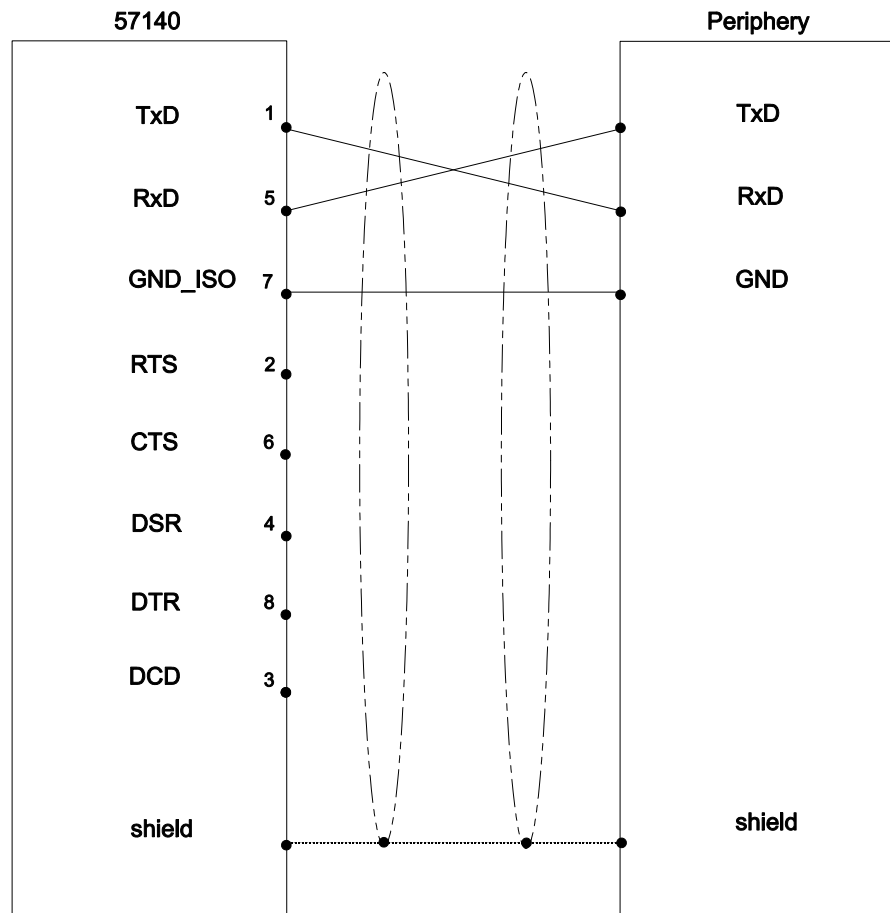


Abb. 4-1: RS232-Verkabelung ohne Hardware-Handshake

RS232-  
Verkabelung mit  
Hardware-  
Handshake

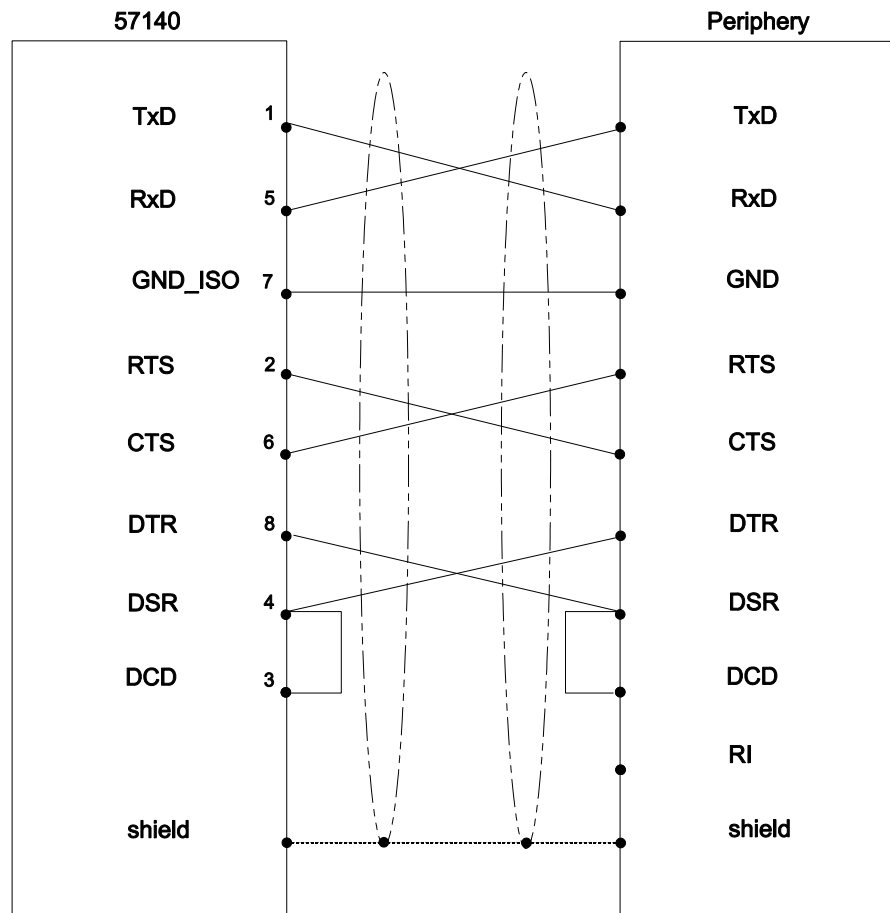


Abb. 4-2: RS232-Verkabelung mit Hardware-Handshake

## 4.3 Schnelleinstieg

<b>Übersicht</b>	Der Kommunikationsprozessor koppelt serielle Prozesse an verschiedene Ziel- oder Quellsysteme. Hierbei wird der Kommunikationsprozessor als Erweiterungsmodul betrieben und über den Rückwandbus mit Betriebsspannung versorgt.
<b>Parameter</b>	<p>Zur Parametrierung übergeben Sie dem Kommunikationsprozessor Parameterdaten. Diese sind je nach gewähltem Protokoll belegt.</p> <p>Nähere Hinweise zur Belegung der Parameter finden Sie im Abschnitt 5 Serielle Kommunikationsprotokolle, Seite 61.</p>
<b>Protokolle</b>	<p>Der Kommunikationsprozessor unterstützt folgende Protokolle:</p> <ul style="list-style-type: none"> <li>■ ASCII</li> <li>■ STX/ETX</li> <li>■ 3964(R)</li> <li>■ Modbus (Master, Slave)</li> </ul>
<b>Kommunikation</b>	<p><b>Senden</b></p> <ol style="list-style-type: none"> <li>1   Das übergeordnete System schreibt Daten über den Rückwandbus in den Ausgabebereich.</li> <li>2   Der Kommunikationsprozessor schreibt die Daten in den Sendepuffer.</li> <li>3   Der Kommunikationsprozessor gibt die Daten von dort über die Schnittstelle aus.</li> </ol> <p><b>Empfangen</b></p> <ol style="list-style-type: none"> <li>1   Der Kommunikationsprozessor empfängt Daten über die Schnittstelle.</li> <li>2   Der Kommunikationsprozessor legt die Daten in einem Ringpuffer ab.</li> <li>3   Über den Rückwandbus trägt der Kommunikationsprozessor die Daten in den Eingabebereich des übergeordneten Systems ein.</li> </ol>



Die Größe des Ein-/Ausgabe-Bereichs und damit auch des Telegramms am Rückwandbus hängt vom übergeordneten System ab.

Den Ein-/Ausgabe-Bereich (E/A-Bereich) und die Kommunikation über den Rückwandbus beschreiben die Abschnitte 4.4 und 4.5.

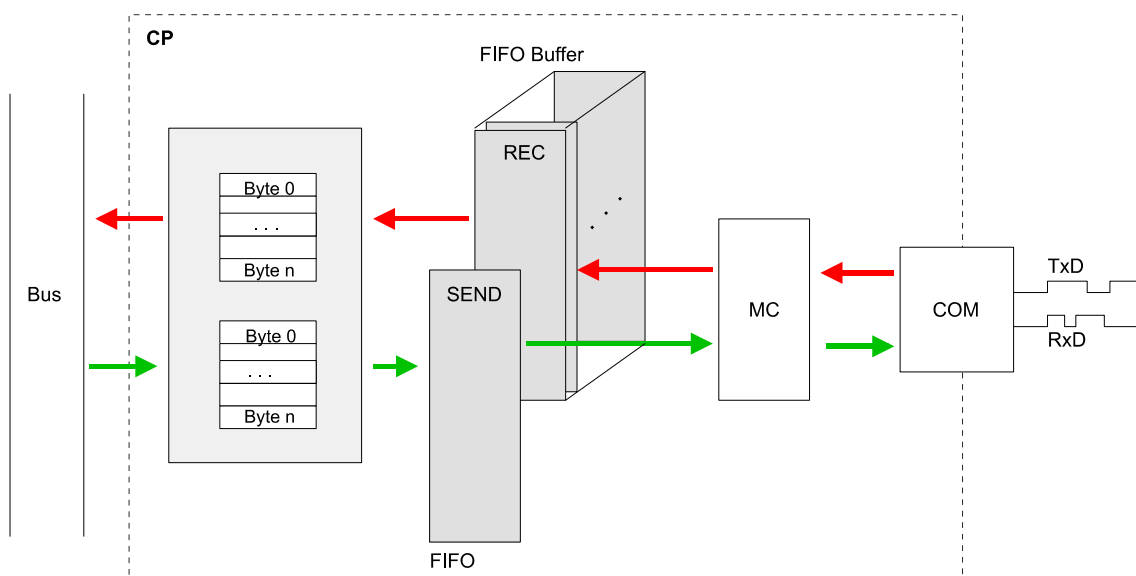


Abb. 4-3: Kommunikation

## 4.4 Ein-/Ausgabe-Bereich

### Übersicht

Abhängig vom übergeordneten System belegt der Kommunikationsprozessor für Ein- und Ausgabe jeweils folgende Anzahl an Bytes im Adressbereich:

System	Adressbereich [Byte]
PROFIBUS	8, 20, 60 (wählbar)
PROFINET	20, 60 (wählbar)
CANopen	8
EtherCAT	60
DeviceNET	60
ModbusTCP	60

Tab. 4-4: Übersicht Adressbereich

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

### Eingabebereich

Adr.	Name	Bytes	Funktion	IX = 0x5450	SX
+0	CP_IN_STS	1	Statusbyte	s = 1	0x01
+1	CP_IN_1	1	Eingabebyte 1	s = 2	0x02
+2	CP_IN_2	1	Eingabebyte 2	s = 3	0x03
...	...	...	...	...	...
+n-1	CP_IN_n-1	1	Eingabebyte n-1	s = m	0xm

Tab. 4-5: Eingabebereich

CP\_IN\_STS

Dieser Parameter enthält 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 = 0x5650	SX
+0	CP_OUT_STS	1	Controlbyte	s = 1	0x01
+1	CP_OUT_1	1	Ausgabebyte 1	s = 2	0x02
+2	CP_OUT_2	1	Ausgabebyte 2	s = 3	0x03
...	...	...	...	...	...
+n-1	CP_OUT_n-1	1	Ausgabebyte n-1	s = m	0xm

Tab. 4-6: Ausgabebereich

CP\_OUT\_CTRL

Hier steuern Sie mit entsprechenden Kommandos die Datenübertragung.

CP\_OUT\_x

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



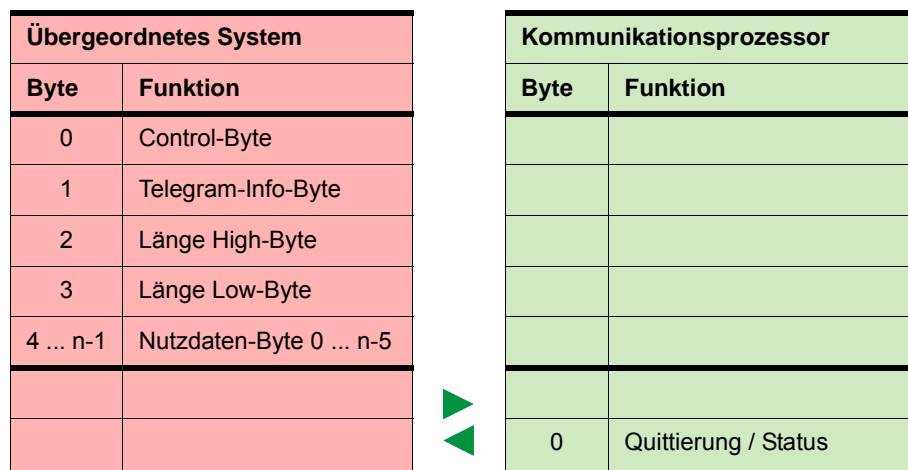
## 4.5 Prinzip der Rückwandbus-Kommunikation

### 4.5.1 Daten senden

#### Übersicht

- Das übergeordneten System trägt die auszugebenden Daten in den Ausgabebereich ein und übergibt sie mit dem **Control-Byte** an den Kommunikationsprozessor.
- Der Kommunikationsprozessor quittiert jedes Telegramm:
  - er kopiert Bit 3 ... 0 von Byte 0 des Ausgabebereichs in Bit 7 ... 4 von Byte 0 des Eingabebereichs **oder**
  - er schickt über dieses Byte eine entsprechende **Statusmeldung** zurück.
- Abhängig von der Länge der zu übertragenden Daten wird das Telegramm in einem Fragment oder mit mehreren Fragmenten an den Kommunikationsprozessor übermittelt.
- Bei der fragmentierten Übertragung quittiert der Kommunikationsprozessor jedes Fragment.

#### Prinzip der Übertragung ohne Fragmentierung



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

#### Control-Byte

Bit	Hex	Funktion
3 ... 0	0x08	Leerlauf - keine Daten vorhanden
	0x0A	Starte Übertragung ohne Fragmentierung
	0x0B	Führe einen Reset auf dem Kommunikationsprozessor aus
7 ... 4		Reserviert für den Empfang

Tab. 4-7: Control-Byte

#### Telegramm-Info-Byte

Beim Sendevorgang 0x00 (fest).

#### Länge

Länge der Nutzdaten für die serielle Kommunikation in Byte.

#### Nutzdaten-Byte

Das Byte enthält die Nutzdaten für die serielle Kommunikation.

## Quittierung / Status

Bit	Hex	Funktion
3 ... 0		Reserviert für den Empfang
7 ... 4	0x08	Quittierung
	0x0A	Leerlauf
	0x0C	Daten ohne Fragmentierung erhalten
	0x0D	Status
	0x0E	Reset auf Kommunikationsprozessor durchgeführt
		Die angegebene Länge ist ungültig
		Fehler in der Kommunikation des Kommunikationsprozessors - Partner antwortet nicht

Tab. 4-8: Quittierung / Status

**Prinzip der Übertragung**  
 mit Fragmentierung

- Das übergeordneten System überträgt mit dem 1. Telegramm (Header) die Anzahl der Nutzdaten und einen Teil der Nutzdaten.
- Danach überträgt das übergeordnete System die Fragment-Telegramme.
- Der Kommunikationsprozessor quittiert jedes Fragment-Telegramm:
  - er kopiert Bit 3 ... 0 von Byte 0 des Ausgabebereichs in Bit 7 ... 4 von Byte 0 des Eingabebereichs **oder**
  - er schickt über dieses Byte eine entsprechende **Statusmeldung** zurück.

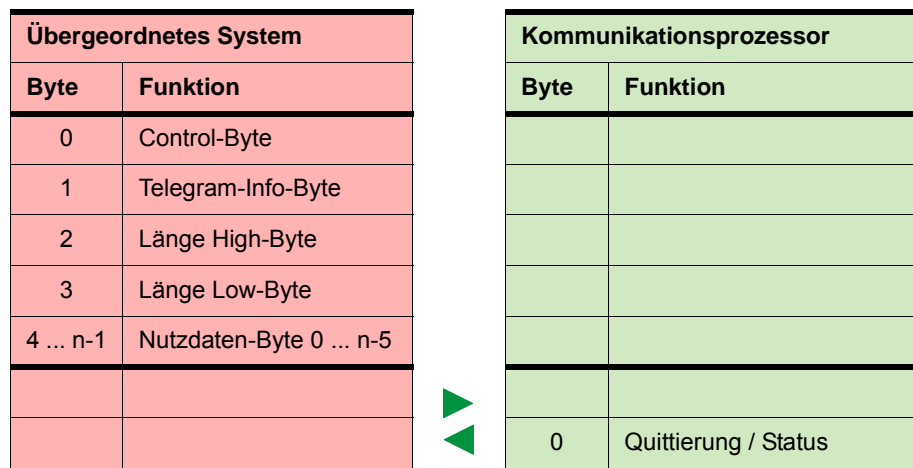
## Ablauf

- 1 | Schreibe 1. Telegramm
- 2 | Schreibe Fragmente
- 3 | Schreibe letztes Fragment

## Berechnung der Fragmentanzahl

$$\text{Fragmentanzahl} = (\text{Länge} + 3) / (\text{IO\_Size} - 1)$$

## Schreibe 1. Telegramm (Header)



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

## Control-Byte

Bit	Hex	Funktion
3 ... 0	0x08	Leerlauf - keine Daten vorhanden
	0x09	Starte fragmentierte Übertragung
	0x0A	Übertrage letztes Fragment
	0x0B	Führe einen Reset auf dem Kommunikationsprozessor aus
7 ... 4		Reserviert für den Empfang

Tab. 4-9: Control-Byte

## Telegramm-Info-Byte

Beim Sendevorgang 0x00 (fest).

Länge Länge der Nutzdaten für die serielle Kommunikation in Byte.

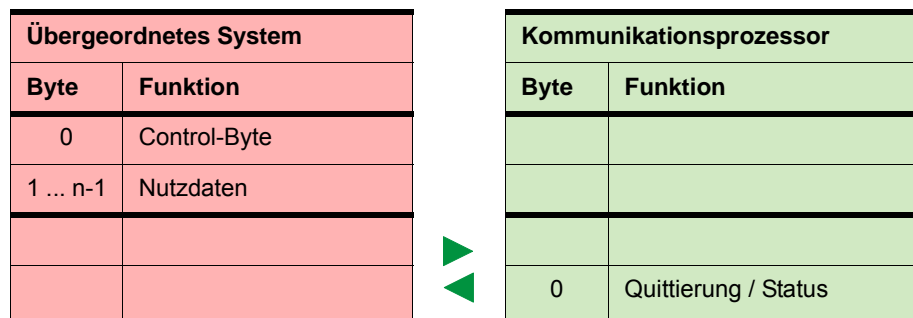
Nutzdaten-Byte Das Byte enthält die Nutzdaten für die serielle Kommunikation.

Quittierung / Status

Bit	Hex	Funktion	
3 ... 0		Reserviert für den Empfang.	
7 ... 4	0x08	Quittierung	Leerlauf
	0x09		Fragmentierte Übertragung gestartet
	0x0A		Daten ohne Fragmentierung erhalten
	0x0C	Status	Reset auf Kommunikationsprozessor durchgeführt
	0x0D		Die angegebene Länge ist ungültig
	0x0E		Fehler Kommunikation des Kommunikationsprozessors - Partner antwortet nicht

Tab. 4-10: Quittierung / Status

Schreibe Fragmente



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

Control-Byte

Bit	Hex	Funktion	
3 ... 0	0x00	Fragment-Nummer	
	...		
	0x07		
	0x08	Leerlauf - keine Daten vorhanden	
	0x0B	Führe einen Reset auf dem Kommunikationsprozessor aus	
7 ... 4		Reserviert für den Empfang	

Tab. 4-11: Control-Byte

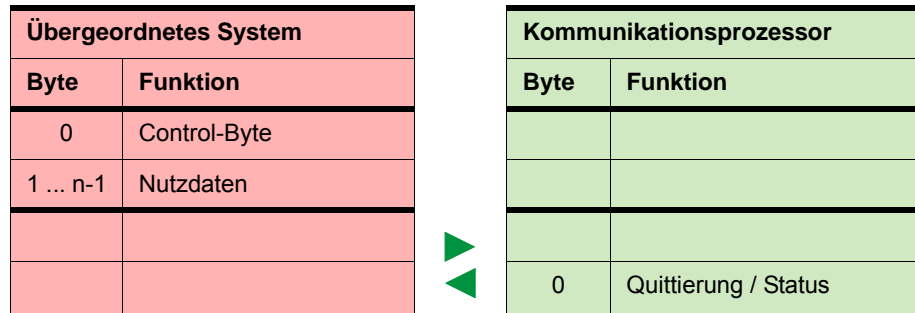
Nutzdaten-Byte Das Byte enthält die Nutzdaten für die serielle Kommunikation.

Quittierung / Status

Bit	Hex	Funktion	
3 ... 0		Reserviert für den Empfang	
7 ... 4	0x00	Quittierung	Fragment-Nummer
	...		
	0x07		
	0x08	Status	Leerlauf
	0x0C		Reset auf Kommunikationsprozessor durchgeführt
	0x0D		Die angegebene Länge ist ungültig
	0x0E	Fehler in der Kommunikation des Kommunikationsprozessors - Partner antwortet nicht	

Tab. 4-12: Quittierung / Status

Schreibe letztes Fragment



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

Control-Byte

Bit	Hex	Funktion
3 ... 0	0x08	Leerlauf - keine Daten vorhanden
	0x0A	Übertrage letztes Fragment
	0x0B	Führe einen Reset auf dem Kommunikationsprozessor aus
7 ... 4		Reserviert für den Empfang

Tab. 4-13: Control-Byte

Nutzdaten-Byte

Das Byte enthält die Nutzdaten für die serielle Kommunikation.

Quittierung / Status

Bit	Hex	Funktion
3 ... 0		Reserviert für den Empfang.
7 ... 4	0x08	Quittierung
	0x0A	Leerlauf
	0x0C	Letztes Fragment erhalten
	0x0D	Status
	0x0E	Reset auf Kommunikationsprozessor durchgeführt
		Die angegebene Länge ist ungültig
		Fehler in der Kommunikation des Kommunikationsprozessors - Partner antwortet nicht

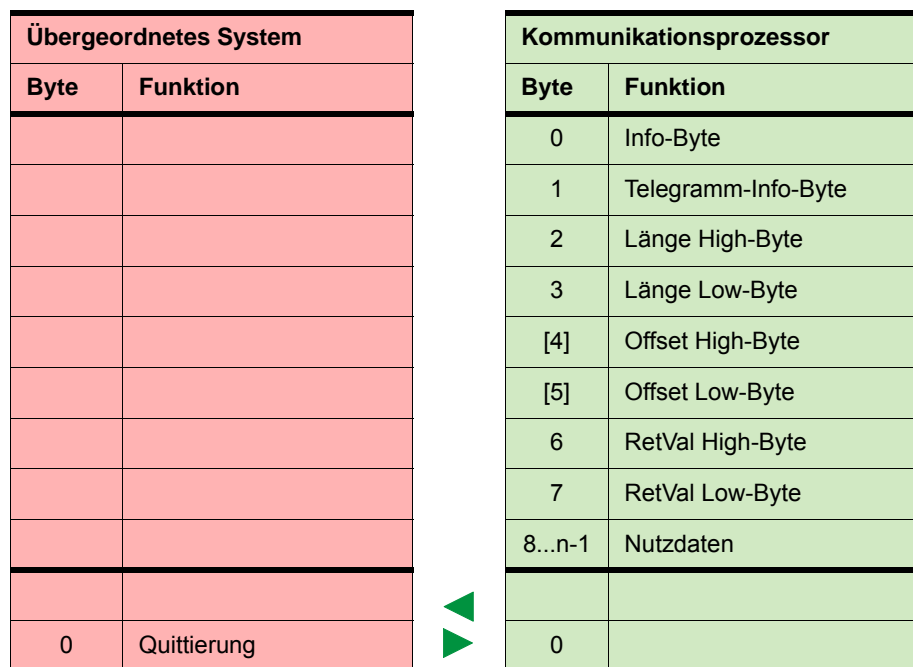
Tab. 4-14: Quittierung / Status

## 4.5.2 Daten empfangen

### Übersicht

- Der Kommunikationsprozessor trägt die empfangenen Daten automatisch im Eingabebereich des übergeordneten Systems ein.
- Abhängig von der Länge der empfangenen Daten übermittelt der Kommunikationsprozessor das Telegramm an das übergeordnete System:
  - in einem Fragment **oder**
  - in mehreren Fragmenten.
- Die fragmentierte Übertragung startet nachdem Bit 3 ... 0 von Byte 0 des Eingabebereichs in Bit 7 ... 4 von Byte 0 des Ausgabebereichs kopiert wurden.
- Übertragungsfehler legt der Kommunikationsprozessor im **RetVal** ab.

### Prinzip der Übertragung ohne Fragmentierung



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

### Info-Byte

Bit	Hex	Funktion
3 ... 0	0x08	Leerlauf - keine Daten vorhanden
	0x09	Daten werden fragmentiert übertragen
	0x0A	Daten werden ohne Fragmentierung übertragen
7 ... 4		Reserviert für Senden

Tab. 4-15: Info-Byte

### Telegramm-Info-Byte

Hex	Funktion
0x00	Das Telegramm enthält keine zusätzlichen Offset-Angaben.
0x04	Das Telegramm enthält zusätzliche Offset-Angaben, die als Wort der <b>Länge</b> nachgestellt sind. Die Offset-Angaben bestimmen die Position der Nutzdaten im Eingabebereich.

Tab. 4-16: Telegramm-Info-Byte

### Länge

Länge der Nutzdaten der seriellen Kommunikation in Byte, zzgl. 2 Byte für **RetVal**.

Offset

Hat das **Telegramm-Info-Byte** den Wert 0x04, wird zusätzlich ein Offset eingetragen. Ansonsten gibt es **Offset** im Telegramm nicht.

RetVal

Hex	Funktion
0x0517	Ungültige Länge (Länge = 0 oder Länge >1024)
0x080A	Kein freier Empfangspuffer vorhanden
0x080C	Fehlerhaftes Zeichen empfangen (Zeichenrahmen- oder Paritätsfehler)

Tab. 4-17: RetVal

## Nutzdaten

Enthält die empfangenen Nutzdaten der seriellen Kommunikation.

## Quittierung

Nachdem das übergeordnete System die Daten verarbeitet hat, muss es dem Kommunikationsprozessor den Empfang quittieren. Erst danach stellt der Kommunikationsprozessor neue Empfangsdaten bereit.

Bit	Hex	Funktion	
3 ... 0		Reserviert für Senden	
7 ... 4	0x08	Quittierung	Leerlauf
	0x0A		Eingabebereich frei für neue Daten
	0x0B	Kommando	Führe einen Reset auf dem Kommunikationsprozessor aus

Tab. 4-18: Quittierung

## Prinzip der Übertragung mit Fragmentierung

Übergeordnetes System	
Byte	Funktion
0	Quittierung

Kommunikationsprozessor	
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
0	

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

Nachdem das übergeordnete System die Daten verarbeitet hat, muss es dem Kommunikationsprozessor den Empfang quittieren.

Dazu kopiert es Bit 3 ... 0 von Byte 0 des Eingabebereichs in Bit 7 ... 4 von Byte 0 des Ausgabebereichs. Erst danach stellt der Kommunikationsprozessor neue Empfangsdaten bereit.

## Berechnung der Fragmentanzahl

$$\text{Fragmentanzahl} = (\text{Länge} + 7) / (\text{IO Size}-1)$$

## Info-Byte

Bit	Hex	Funktion
3 ... 0	0x08	Leerlauf - keine Daten vorhanden
	0x09	Daten werden fragmentiert übertragen
	0x0A	Daten werden ohne Fragmentierung übertragen
7 ... 4		Reserviert für Senden

Tab. 4-19: Info-Byte

## Telegramm-Info-Byte

Hex	Funktion
0x00	Das Telegramm enthält keine zusätzlichen Offset-Angaben.
0x04	Das Telegramm enthält zusätzliche Offset-Angaben, die als Wort der <b>Länge</b> nachgestellt sind. Die Offset-Angaben bestimmen die Position der Nutzdaten im Eingabebereich.

Tab. 4-20: Telegramm-Info-Byte

## Länge

Länge der Nutzdaten in Byte, zzgl. 2 Byte für **RetVal**.

## Offset

Hat das **Telegramm-Info-Byte** den Wert 0x04, wird zusätzlich ein Offset eingetragen. Ansonsten gibt es **Offset** im Telegramm nicht.

Berechnung des Offset bei fragmentierter Übertragung:

$$\text{Daten\_Offset} = (\text{Fragmentzähler} + 1) \times (\text{IO\_SIZE} - 1) - 7 + \text{Offset}$$

mit      Daten\_Offset:      Offset der Daten im Eingabebereich  
           Fragmentzähler:    Absolute Fragmentanzahl  
           IO\_Size:            Anzahl der belegten Bytes im Adressbereich  
           Offset:             Offset-Wert im Telegramm

## Nutzdaten

Enthält die empfangenen Nutzdaten der seriellen Kommunikation.

## Quittierung

Nachdem das übergeordnete System die Daten verarbeitet hat, muss es dem Kommunikationsprozessor den Empfang quittieren. Erst danach stellt der Kommunikationsprozessor neue Empfangsdaten bereit.

Bit	Hex	Funktion
3 ... 0		Reserviert für Senden
7 ... 4	0x08	Quittierung    Leerlauf
	0x0A	Eingabebereich frei für neue Daten
	0x0B	Kommando    Führe einen Reset auf dem Kommunikationsprozessor aus

Tab. 4-21: Quittierung

### 4.5.3 Beispiele

**Daten senden**  
ohne Fragmentierung

IO-Size = 60 Byte, Länge = 40 Byte

Übergeordnetes System	
Byte	Funktion
0	0x0A Kommando
1	0x00 Telegramm-Info
2	0x00 Länge High-Byte
3	0x28 Länge Low-Byte
4 ... 43	Nutzdaten-Byte 0 ... 39
44 ... 59	nicht verwendet

Kommunikationsprozessor	
Byte	Funktion
0	0xA0 Quittierung

**Daten senden**  
mit Fragmentierung

IO-Size = 16 Byte, Länge = 40 Byte

Header

Übergeordnetes System	
Byte	Funktion
0	0x09 Kommando
1	0x00 Telegramm-Info
2	0x00 Länge High-Byte
3	0x28 Länge Low-Byte
4 ... 15	Nutzdaten-Byte 0 ... 11

Kommunikationsprozessor	
Byte	Funktion
0	0x90 Quittierung



## 1. Fragment

Übergeordnetes System	
Byte	Funktion
0	0x00 Fragment
1 ... 15	Nutzdaten-Byte 12 ... 26



Kommunikationsprozessor	
Byte	Funktion
0	0x00 Quittierung

## 2. Fragment

Übergeordnetes System	
Byte	Funktion
0	0x01 Fragment
1 ... 15	Nutzdaten-Byte 27 ... 41



Kommunikationsprozessor	
Byte	Funktion
0	0x10 Quittierung

## Letztes Fragment

Übergeordnetes System	
Byte	Funktion
0	0x0A Kommando
1 ... 8	Nutzdaten-Byte 42 ... 49
11 ... 15	nicht verwendet



Kommunikationsprozessor	
Byte	Funktion
0	0xA0 Quittierung

**Daten empfangen**  
 ohne Fragmentierung

IO-Size = 60 Byte, Länge = 40 Byte

Übergeordnetes System	
Byte	Funktion
0	0xA0 Quittierung



Kommunikationsprozessor	
Byte	Funktion
0	0x0A Fragment-Info
1	0x00 Telegramm-Info-Byte
2	0x00 Länge High-Byte
3	0x2A Länge Low-Byte + 2 Byte
4	0x00 Return Value High-Byte
5	0x00 Return Value Low-Byte
6 ... 45	Nutzdaten-Byte 0 ... 39
46 ... 59	wird nicht verwendet
0	

**Daten empfangen**  
 mit Fragmentierung

IO-Size = 16 Byte, Länge = 40 Byte

Header

Übergeordnetes System	
Byte	Funktion
0	0x90 Quittierung



Kommunikationsprozessor	
Byte	Funktion
0	0x09 Fragment-Info
1	0x00 Telegramm-Info-Byte
2	0x00 Länge High-Byte
3	0x2A Länge Low-Byte + 2Byte
4	0x00 Return Value High-Byte
5	0x00 Return Value Low-Byte
6 ... 15	Nutzdaten-Byte 0 ... 9
0	

## 1. Fragment

Übergeordnetes System	
Byte	Funktion
0	0x00 Quittierung



Kommunikationsprozessor	
Byte	Funktion
0	0x00 Fragment-Info
1 ... 15	Nutzdaten-Byte 10 ... 24
0	

## Letztes Fragment

Übergeordnetes System	
Byte	Funktion
0	0xA0 Quittierung



Kommunikationsprozessor	
Byte	Funktion
0	0x0A Fragment-Info
1 ... 15	Nutzdaten-Byte 25 ... 39
0	

## 4.6 Rückwandbus-Kommunikation

### Übersicht

Für die Verarbeitung der Verbindungsaufträge auf SPS-Seite ist ein Anwenderprogramm in der CPU erforderlich. Hierbei kommen zur Kommunikation zwischen CPU, Kommunikationsprozessor und einem Kommunikationspartner folgende spezifischen Bausteine der Firma Murrelektronik GmbH 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

Tab. 4-22: Bausteine zum Senden und Empfangen

### 4.6.1 Bausteine installieren

#### Bausteine installieren

Die spezifischen Bausteine der Murrelektronik GmbH finden Sie im Service-Bereich auf [www.murrelektronik.com](http://www.murrelektronik.com) unter Downloads > ME LIB als Bibliothek zum Download.

Die Bibliothek liegt als gepackte zip-Datei vor. Sie muss in Ihr Projekt importiert werden.

Folgende Schritte sind hierzu erforderlich:

- Datei FX000011\_Vxxx.zip entpacken
- Bibliothek **dearchivieren**
- Bibliothek öffnen und Bausteine in Projekt übertragen

#### FX000011\_Vxxx.zip

##### **FX000011\_Vxxx.zip entpacken**

- ➔ Starten Sie mit einem Doppelklick auf **FX000011\_Vxxx.zip** Ihr Unzip-Programm.
- ➔ Kopieren Sie die Datei Murrelektronik.ZIP in Ihr Arbeitsverzeichnis.

*Es ist nicht erforderlich diese Datei weiter zu entpacken.*

#### Bibliothek dearchivieren

##### **Bibliothek dearchivieren für die SPEED7-CPU's**

- ➔ Starten Sie den SIMATIC Manager von Siemens.
- ➔ Öffnen Sie über **Datei > Dearchivieren** ein Dialogfenster zur Auswahl des Archivs.
- ➔ Navigieren Sie in Ihr Arbeitsverzeichnis.
- ➔ Klicken Sie auf Murrelektronik.ZIP.
- ➔ Klicken Sie auf **Öffnen**.
- ➔ Geben Sie ein Zielverzeichnis für die Ablage der Bausteine an.
- ➔ Klicken Sie auf **OK**.

*Der Entpackvorgang startet.*

#### Bibliothek öffnen und Bausteine in Projekt übertragen

##### **Bibliothek öffnen**

- ➔ Öffnen Sie die Bibliothek nach dem Entpackvorgang.
- ➔ Öffnen Sie Ihr Projekt.
- ➔ Kopieren Sie die erforderlichen Bausteine aus der Bibliothek in das Verzeichnis **Bausteine** Ihres Projekts.

*Sie haben in Ihrem Anwenderprogramm Zugriff auf die spezifischen Bausteine der Murrelektronik GmbH.*

### Kommunikationsprinzip

#### Telegramm senden

Durch zyklischen Aufruf von FB 60 und FB 61 sendet und empfängt der Kommunikationsprozessor zyklisch Daten. Der Kommunikationsprozessor setzt die Übertragungsprotokolle zum Kommunikationspartner um. Die Übertragungsprotokolle können Sie in der Hardwarekonfiguration parametrieren.

#### Telegramm empfangen

Ein komplettes Telegramm wird, abhängig von der parametrierten IO-Size, in Datenblöcke unterteilt und an den Rückwandbus übergeben. Die CPU setzt die Datenblöcke zusammen.

Der Kommunikationsprozessor empfängt Telegramme von der CPU asynchron über den Rückwandbus.

Ist ein komplettes Telegramm über die serielle Schnittstelle eingetroffen, wird es in einem Ringpuffer abgelegt (Größe des Ringpuffers: 1024 Byte). Aus der Länge des noch freien Ringpuffers ergibt sich die max. Länge eines Telegramms. Je nach Parametrierung können bis zu 250 Telegramme gepuffert werden, deren Gesamtlänge darf jedoch 1024 Byte nicht überschreiten. Ist der Puffer voll, werden neu ankommende Telegramme verworfen.

#### Software-Handshake

Da der Datenaustausch über den Rückwandbus asynchron abläuft, wird ein Software-Handshake zwischen dem Kommunikationsprozessor und der CPU eingesetzt. Hierzu besitzen die Bausteine FB 60 und FB 61 den gemeinsamen Parameter **CONTROL**. Beide Bausteine verwenden für **CONTROL** verschiedene Bits im selben Merker-Byte.

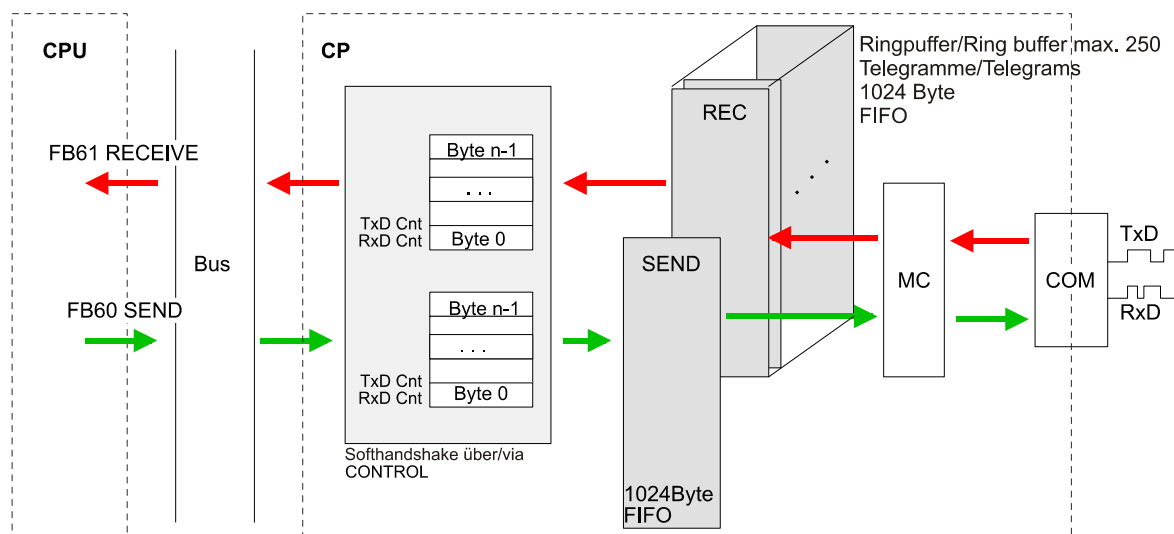


Abb. 4-4: CPU-57140 Kommunikationsprozessor



### HINWEIS

Zum Erkennen eines Signalwechsels ist eine Mindestimpulsdauer erforderlich.

Ausschlaggebend sind:

- ➔ CPU-Zykluszeit,
- ➔ Aktualisierungszeit auf dem Kommunikationsprozessor,
- ➔ Reaktionszeit des Kommunikationspartners.

## 4.6.2 FB 60 - SEND

### FB 60 - SEND Senden an Kommunikationsprozessor

#### Parameter

Der Baustein FB 60 gibt Daten von der CPU an den Kommunikationsprozessor aus. Über die Bezeichner **DB\_NO**, **DBB\_NO** und **LEN** legen Sie den Sendebereich fest. Eine positive Flanke an **REQ** stößt das Senden der Daten an.

Name	Deklaration	Typ	Beschreibung
REQ	IN	BOOL	Sendefreigabe bei positiver Flanke
R	IN	BOOL	Synchron-Reset auslösen
LADDR	IN	INT	Logische Basisadresse des Kommunikationsprozessors
DB_NO	IN	INT	Nummer des Datenbausteins der Sendedaten
DBB_NO	IN	INT	Nummer des Datenbyte - Sendedaten ab Datenbyte
LEN	IN	INT	Länge des Sende-Telegramms in Byte
IO_SIZE	IN	WORD	Parametrierte IO-Größe des Moduls
DONE *	OUT	BOOL	Sende-Auftrag fertig ohne Fehler
ERROR *	OUT	BOOL	Sende-Auftrag fertig mit Fehler <b>STATUS</b> enthält die Fehlerinformation
STATUS *	OUT	WORD	Spezifikation des Fehlers bei ERROR = 1
CONTROL	IN_OUT	BYTE	Geteiltes Byte mit <b>RECEIVE</b> -Baustein: <b>SEND</b> (Bit 0 ... 3) <b>RECEIVE</b> (Bit 4 ... 7)

Tab. 4-23: Parameter



\* Der Parameter steht bis zum nächsten Aufruf von FB 60 zur Verfügung.

REQ

#### Request - Sendefreigabe

Eine positive Flanke am Eingang **REQ** stößt die Übertragung der Daten an. Je nach Datenmenge kann eine Datenübertragung über mehrere Programmzyklen laufen.

R

#### Synchron-Reset

Initialisierung	SEND im Anlauf-OB mit allen Parametern und mit gesetztem <b>R</b> einmalig aufrufen.
Abbruch	Abbruch des laufenden Auftrags jederzeit mit dem Signalzustand <b>1</b> an <b>R</b> . FB wird in den Grundzustand zurückgesetzt. <b>HINWEIS</b> Daten, die der Kommunikationsprozessor bereits erhalten hat, werden noch an den Kommunikationspartner gesendet.
Senden ausgeschaltet	Statischer Signalzustand <b>1</b> an <b>R</b> .

LADDR

#### Peripherieadresse

Mit **LADDR** geben Sie die Adresse des anzusprechenden Kommunikationsprozessor an. Dies ist die Adresse, die Sie über die Hardware-Konfiguration für den Kommunikationsprozessor vergeben haben.

DB\_NO

#### Datenbaustein-Nummer

Nummer des Datenbausteins, der die zu sendenden Daten beinhaltet. Null ist nicht erlaubt.

DBB_NO	<b>Datenbyte-Nummer</b> Nummer des Datenbytes im Datenbaustein, ab dem die Sendedaten abgelegt sind.
LEN	<b>Länge</b> Länge der Nutzdaten, die zu übertragen sind. Es gilt: $1 \leq \text{LEN} \leq 1024$ .
IO_SIZE	<b>Größe E/A-Bereich</b> Geben Sie hier die Größe des E/A-Bereichs an. Abhängig vom übergeordneten System belegt der Kommunikationsprozessor für Ein- und Ausgabe jeweils folgende wählbare Anzahl an Bytes im Adress-Bereich:

System	Adressbereich [Byte]
PROFIBUS	8, 20, 60 (wählbar)
PROFINET	20, 60 (wählbar)
CANopen	8
EtherCAT	60
DeviceNET	60
ModbusTCP	60

Tab. 4-24: IO\_SIZE: Größe E/A-Bereich

DONE	<b>Auftrag fertig ohne Fehler</b> DONE wird gesetzt bei Auftrag fertig ohne Fehler und STATUS = 0x0000.
ERROR	<b>Auftrag fertig mit Fehler</b> ERROR wird gesetzt. STATUS enthält die Fehlerinformation.
STATUS	Bei fehlerfreier Funktion ist STATUS 0x0000. Solange ERROR gesetzt ist, bleibt der Wert in STATUS bestehen.

Fehlercode	Funktion
0x0000	Kein Fehler vorhanden
0x0202	Baustein FB 60 und Kommunikationsprozessor sind nicht synchron (Abhilfe: Synchron-Reset auslösen)
0x0301	DB ist nicht gültig
0x070A	Übertragung fehlgeschlagen, Partner antwortet nicht, oder hat den Auftrag negativ quittiert
0x0816	Ungültiger LEN-Parameter (LEN = 0 oder LEN > 1024)
0x8181	Auftrag läuft (Status und keine Fehlermeldung)

Tab. 4-25: Fehlermeldungen

CONTROL	Die Bausteine SEND und RECEIVE verwenden für den Handshake den gemeinsamen Parameter <b>CONTROL</b> . ➔ Weisen Sie <b>CONTROL</b> ein gemeinsames Merker-Byte zu.
Fehleranzeige	Der Ausgang <b>DONE</b> zeigt <b>Auftragsende ohne Fehler</b> an, <b>STATUS</b> hat den Wert <b>0</b> . Bei <b>ERROR</b> wird die Ereignisnummer in <b>STATUS</b> angezeigt. <b>DONE</b> , <b>ERROR</b> und <b>STATUS</b> werden auch bei Reset des Bausteins ausgegeben. Bei einem Fehler wird das Binärergebnis BIE zurückgesetzt. Wird der Baustein ohne Fehler beendet, hat BIE den Zustand <b>1</b> .



Die Parameter **DONE**, **ERROR** und **STATUS** sind immer nur für einen Baustein aufrufbar.

➔ Kopieren Sie **DONE**, **ERROR** und **STATUS** zur weiteren Auswertung in einen freien Datenbereich.

### 4.6.3 FB 61 - RECEIVE

#### FB 61 - RECEIVE Empfangen von Kommunikationsprozessor

Der Baustein FB 61 unterstützt den Datenempfang der CPU vom Kommunikationsprozessor. Über die Bezeichner **DB\_NO**, **DBB\_NO** und **LEN** legen Sie den Empfangsbereich fest. Die Länge des eingelesenen Telegramms wird in **LEN** abgelegt.

#### Parameter

Name	Deklaration	Typ	Beschreibung
EN_R	IN	BOOL	Freigabe zum Lesen der Daten
R	IN	BOOL	Synchron-Reset auslösen
LADDR	IN	INT	Logische Basisadresse des Kommunikationsprozessors
DB_NO	IN	INT	Nummer des Datenbausteins der Sendedaten
DBB_NO	IN	INT	Nummer des Datenbyte - Empfangsdaten ab Datenbyte
IO_SIZE	IN	WORD	Parametrierte IO-Größe des Moduls
LEN	IN	INT	Länge des Sende-Telegramms in Byte
NDR *	OUT	BOOL	Empfangs-Auftrag fertig ohne Fehler
ERROR *	OUT	BOOL	Empfangs-Auftrag fertig mit Fehler <b>STATUS</b> enthält die Fehlerinformation
STATUS *	OUT	WORD	Spezifikation des Fehlers bei ERROR = 1
CONTROL	IN_OUT	BYTE	Geteiltes Byte mit <b>SEND</b> -Baustein: <b>SEND</b> (Bit 0 ... 3) <b>RECEIVE</b> (Bit 4 ... 7)

Tab. 4-26: Parameter

\* Der Parameter steht bis zum nächsten Aufruf von FB 61 zur Verfügung.



EN\_R

#### Enable Receive - Lesefreigabe

Signalzustand **1** an EN\_R gibt die Überprüfung frei, ob Daten vom Kommunikationsprozessor zu lesen sind. Je nach Datenmenge kann eine Datenübertragung über mehrere Programmzyklen laufen.

Mit dem Signalzustand **0** an EN\_R brechen Sie eine laufende Übertragung ab. Der abgebrochene Empfangsauftrag wird mit einer Fehlermeldung (**STATUS**) beendet. Solange **0** an EN\_R ansteht, ist der Empfang ausgeschaltet.

R

#### Synchron-Reset

Initialisierung	RECEIVE im Anlauf-OB mit allen Parametern und mit gesetztem <b>R</b> einmalig aufrufen.
Abbruch	Abbruch des laufenden Auftrags jederzeit mit dem Signalzustand <b>1</b> an <b>R</b> . FB wird in den Grundzustand zurückgesetzt.
Empfang ausgeschaltet	Statischer Signalzustand <b>1</b> an <b>R</b> .

LADDR

#### Peripherieadresse

Mit **LADDR** geben Sie die Adresse des anzusprechenden Kommunikationsprozessor an. Dies ist die Adresse, die Sie über die Hardware-Konfiguration für den Kommunikationsprozessor vergeben haben.

DB\_NO

#### Datenbaustein-Nummer

Nummer des Datenbausteins, der die zu sendenden Daten beinhaltet. Null ist nicht erlaubt.



DBB_NO	<b>Datenbyte-Nummer</b> Nummer des Datenbytes im Datenbaustein, ab dem die empfangenen Daten abgelegt werden sollen.																		
LEN	<b>Länge</b> Länge der Nutzdaten, die zu übertragen sind. Es gilt: $1 \leq \text{LEN} \leq 1024$ .																		
IO_SIZE	<b>Größe E/A-Bereich</b> Geben Sie hier die Größe des E/A-Bereichs an. Abhängig vom übergeordneten System belegt der Kommunikationsprozessor für Ein- und Ausgabe jeweils folgende wählbare Anzahl an Bytes im Adress-Bereich: <table border="1" data-bbox="528 631 1442 855"> <thead> <tr> <th>System</th><th>Adressbereich [Byte]</th></tr> </thead> <tbody> <tr> <td>PROFIBUS</td><td>8, 20, 60 (wählbar)</td></tr> <tr> <td>PROFINET</td><td>20, 60 (wählbar)</td></tr> <tr> <td>CANopen</td><td>8</td></tr> <tr> <td>EtherCAT</td><td>60</td></tr> <tr> <td>DeviceNET</td><td>60</td></tr> <tr> <td>ModbusTCP</td><td>60</td></tr> </tbody> </table> <p><i>Tab. 4-27: IO_SIZE: Größe E/A-Bereich</i></p>	System	Adressbereich [Byte]	PROFIBUS	8, 20, 60 (wählbar)	PROFINET	20, 60 (wählbar)	CANopen	8	EtherCAT	60	DeviceNET	60	ModbusTCP	60				
System	Adressbereich [Byte]																		
PROFIBUS	8, 20, 60 (wählbar)																		
PROFINET	20, 60 (wählbar)																		
CANopen	8																		
EtherCAT	60																		
DeviceNET	60																		
ModbusTCP	60																		
NDR	<b>New data ready</b> Neue Daten stehen für die CPU im Kommunikationsprozessor bereit.																		
ERROR	<b>Auftrag fertig mit Fehler</b> ERROR wird gesetzt. STATUS enthält die Fehlerinformation.																		
STATUS	Bei fehlerfreier Funktion ist STATUS 0x0000. Solange ERROR gesetzt ist, bleibt der Wert in STATUS bestehen. <table border="1" data-bbox="528 1164 1442 1536"> <thead> <tr> <th>Fehlercode</th><th>Funktion</th></tr> </thead> <tbody> <tr> <td>0x0000</td><td>Kein Fehler vorhanden</td></tr> <tr> <td>0x0202</td><td>Baustein und Kommunikationsprozessor sind nicht synchron (Abhilfe: Synchron Reset auslösen)</td></tr> <tr> <td>0x0301</td><td>DB ist nicht gültig</td></tr> <tr> <td>0x070A</td><td>Übertragung fehlgeschlagen, Partner antwortet nicht, oder hat den Auftrag negativ quittiert</td></tr> <tr> <td>0x0816</td><td>Ungültiger LEN-Parameter (LEN = 0 oder LEN &gt; 1024)</td></tr> <tr> <td>0x080A</td><td>Kein freier Empfangspuffer ist vorhanden</td></tr> <tr> <td>0x080C</td><td>Fehlerhaftes Zeichen empfangen (Zeichenrahmen- oder Paritätsfehler)</td></tr> <tr> <td>0x8181</td><td>Auftrag läuft (Status und keine Fehlermeldung)</td></tr> </tbody> </table> <p><i>Tab. 4-28: Fehlermeldungen</i></p>	Fehlercode	Funktion	0x0000	Kein Fehler vorhanden	0x0202	Baustein und Kommunikationsprozessor sind nicht synchron (Abhilfe: Synchron Reset auslösen)	0x0301	DB ist nicht gültig	0x070A	Übertragung fehlgeschlagen, Partner antwortet nicht, oder hat den Auftrag negativ quittiert	0x0816	Ungültiger LEN-Parameter (LEN = 0 oder LEN > 1024)	0x080A	Kein freier Empfangspuffer ist vorhanden	0x080C	Fehlerhaftes Zeichen empfangen (Zeichenrahmen- oder Paritätsfehler)	0x8181	Auftrag läuft (Status und keine Fehlermeldung)
Fehlercode	Funktion																		
0x0000	Kein Fehler vorhanden																		
0x0202	Baustein und Kommunikationsprozessor sind nicht synchron (Abhilfe: Synchron Reset auslösen)																		
0x0301	DB ist nicht gültig																		
0x070A	Übertragung fehlgeschlagen, Partner antwortet nicht, oder hat den Auftrag negativ quittiert																		
0x0816	Ungültiger LEN-Parameter (LEN = 0 oder LEN > 1024)																		
0x080A	Kein freier Empfangspuffer ist vorhanden																		
0x080C	Fehlerhaftes Zeichen empfangen (Zeichenrahmen- oder Paritätsfehler)																		
0x8181	Auftrag läuft (Status und keine Fehlermeldung)																		
CONTROL	Die Bausteine SEND und RECEIVE verwenden für den Handshake den gemeinsamen Parameter <b>CONTROL</b> . ➔ Weisen Sie <b>CONTROL</b> ein gemeinsames Merker-Byte zu.																		

**Fehleranzeige**

Der Ausgang **NDR** zeigt **Auftrag fertig ohne Fehler/Daten übernommen** an. Bei **ERROR** wird die entsprechende Ereignisnummer in **STATUS** angezeigt. Ist kein Fehler aufgetreten, hat **STATUS** den Wert **0**. **NDR** und **ERROR/STATUS** werden auch bei Reset des FB ausgegeben. Bei einem aufgetretenen Fehler wird das Binärergebnis **BIE** zurückgesetzt. Wird der Baustein ohne Fehler beendet, hat **BIE** den Zustand **1**.



Die Parameter **NDR**, **ERROR** und **STATUS** sind immer nur für einen Bausteinaufruf verfügbar.

➔ Kopieren Sie diese Parameter zur weiteren Auswertung in einen freien Datenbereich.

## 4.7 Diagnosedaten

### Übersicht

Sie können über die Parametrierung einen Diagnosealarm für das Modul aktivieren.

- Mit dem Auslösen eines Diagnosealarms werden vom Modul Diagnosedaten für Diagnose<sub>kommend</sub> bereitgestellt.
- Sobald die Gründe für das Auslösen eines Diagnosealarms nicht mehr gegeben sind, erhalten Sie automatisch einen Diagnosealarm<sub>gehend</sub>.
- Innerhalb dieses Zeitraums (1. Diagnosealarm<sub>kommend</sub> bis letzter Diagnosealarm<sub>gehend</sub>) leuchtet die MF-LED des Moduls.

- DS     Datensatz für Zugriff über CPU, PROFIBUS und PROFINET  
          Zugriff über DS 0x01  
          Zusätzlicher Zugriff über DS 0x00 auf die ersten 4 Byte
- IX     Index für Zugriff über CANopen  
          Zugriff über IX 0x2F01  
          Zusätzlicher Zugriff über IX 0x2F00 auf die ersten 4 Byte
- SX     Subindex für den Zugriff über EtherCAT

Name	Bytes	Funktion	Default	DS	IX	SX
ERR_A	1	Diagnose	0x00	0x01	0x2F01	0x02
MODTYP	1	Modulinformation	0x10			0x03
ERR_C	1	reserviert	0x00			0x04
ERR_D	1	Diagnose	0x00			0x05
CHTYP	1	Kanaltyp	0x60			0x06
NUMBIT	1	Anzahl Diagnosebits pro Kanal	0x08			0x07
NUMCH	1	Anzahl Kanäle des Moduls	0x01			0x08
CHERR	1	reserviert	0x00			0x09
CH0ERR	1	reserviert	0x00			0x0A
CH1ERR ... CH7ERR	7	reserviert	0x00			0x0B ... 0x11
DIAG_US	4	µs-Ticker	0x00			0x12

Tab. 4-29: Diagnosedaten

### ERR\_A Diagnose

Byte	Bit 7 ... 0	Beschreibung
0	0	gesetzt bei <b>Baugruppenstörung</b>
	1	gesetzt bei <b>Fehler intern</b>
	2	reserviert
	3	reserviert
	4	gesetzt bei <b>fehlender externer Versorgungsspannung</b>
	5, 6	reserviert
	7	gesetzt bei <b>Parametrierfehler</b>

Tab. 4-30: Diagnose

### MODTYP Modulinformation

Byte	Bit 7 ... 0	Beschreibung
0	Bit 3 ... 0	Modulkasse
		1111b     Kommunikationsprozessor
	Bit 4	gesetzt bei <b>Kanalinformation vorhanden</b>
	Bit 7 ... 5	reserviert

Tab. 4-31: Modulinformation

ERR\_D  
 Diagnose

Byte	Bit 7 ... 0	Beschreibung
0	Bit 2 ... 0	reserviert
	Bit 3	gesetzt bei <i>internem Diagnosepufferüberlauf</i>
	Bit 4	gesetzt bei <i>internem Kommunikationsfehler</i>
	Bit 7 ... 5	reserviert

Tab. 4-32: Diagnose

 CHTYP  
 Kanaltyp

Byte	Bit 7 ... 0	Beschreibung
0	6 ... 0	Kanaltyp
	0x60	Kommunikationsprozessor
	7	reserviert

Tab. 4-33: Kanaltyp

 NUMBIT  
 Diagnosebits

Byte	Bit 7 ... 0	Beschreibung
0	7 ... 0	Anzahl der Diagnosebits des Moduls pro Kanal (hier 0x08)

Tab. 4-34: Diagnosebits

 NUMCH  
 Kanäle

Byte	Bit 7 ... 0	Beschreibung
0	7 ... 0	Anzahl der Kanäle eines Moduls (hier 0x01)

Tab. 4-35: Kanäle

 CHERR, CH0ERR ...  
 CH7ERRreserviert

Byte	Bit 7 ... 0	Beschreibung
0	7 ... 0	reserviert

Tab. 4-36: CHERR, CH0ERR ... CH7ERR

 DIAG\_US  
 µs-Ticker

Byte	Bit 7 ... 0	Beschreibung
0 ... 3	7 ... 0	Wert des µs-Tickers bei Generierung der Diagnosedaten

Tab. 4-37: µs-Ticker



#### µs-Ticker

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

## 5 Serielle Kommunikationsprotokolle

### 5.1 Übersicht

#### Serielle Übertragung eines Zeichens

Die Punkt-zu-Punkt-Verbindung zwischen zwei Kommunikationspartnern ist die einfachste Form des Informationsaustauschs. Hierbei bildet der Kommunikationsprozessor die Schnittstelle zwischen einem übergeordneten System und einem seriell angebundenen Kommunikationspartner.

Die Daten werden seriell übertragen. 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 stattfinden.

Jedem zu übertragenden Zeichen geht ein Synchronisierimpuls als **Startbit** voraus. Das Ende des Zeichentransfers bildet das **Stoppbit**.

Neben **Startbit** und **Stoppbit** 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 Stoppbits

#### Protokolle

Der Kommunikationsprozessor wickelt die serielle Datenübertragung selbstständig ab. Hierzu ist der Kommunikationsprozessor mit Treibern für die Protokolle ausgestattet.

Nachfolgend sind folgende Protokolle beschrieben:

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

## 5.2 ASCII

### 5.2.1 Funktion

#### Funktionsweise

Die Datenkommunikation via ASCII ist eine einfache Form des Datenaustauschs. Sie kann mit einer Multicast/Broadcast-Funktion verglichen werden.

Über die **Zeichenverzugszeit** (ZVZ) werden die Telegramme logisch getrennt. Innerhalb der Zeichenverzugszeit muss der Sender sein Telegramm an den Empfänger gesendet 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

DS Datensatz für Zugriff über CPU, PROFIBUS und PROFINET  
IX Index für Zugriff über CANopen  
SX Subindex für den Zugriff über EtherCAT

Name	Bytes	Funktion	Default	DS	IX	SX
PII_L	1	Länge Prozessabbild Eingabedaten	Werte hängen vom übergeordneten System ab	0x02	0x3100	0x01
PIQ_L	1	Länge Prozessabbild Ausgabedaten		0x02	0x3101	0x02
DIAG_EN	1	Diagnosealarm	0x00	0x00	0x3102	0x03
BAUD	1	Baudrate	0x00	0x80	0x3103	0x04
PROTOCOL	1	Protokoll	0x01	0x80	0x3104	0x05
OPTION3	1	Zeichenrahmen	0x13	0x80	0x3105	0x06
OPTION4, 5	2	ZNA 0 ... 65535 (ms)	0	0x80	0x3106 ... 0x3107	0x07 ... 0x08
OPTION6, 7	2	ZVZ 0 ... 65535 (ms)	250	0x80	0x3108 ... 0x3109	0x09 ... 0x0A
OPTION8	1	Anz. Receivebuffer	1	0x80	0x310A	0x0B
OPTION9 ... 14	6	reserviert	0x00	0x80	0x310B ... 0x3110	0x0C ... 0x11

Tab. 5-1: Parametrierdaten bei ASCII

### DIAG\_EN Diagnosealarm

➔ Aktivieren bzw. deaktivieren Sie hier die Diagnosefunktion.

Wertebereich 0x00: sperren Default: 0x00  
0x40: freigeben

### BAUD Übertragungsrate

➔ Geben Sie die Geschwindigkeit der Datenübertragung in Bit/s (Baud) an. Sie haben folgende Einstellbereiche; andere Werte sind nicht zulässig:  
Default: 0x00 (9600 Baud)

#### Wertebereich:

Hex	Baud	Hex	Baud	Hex	Baud
0x00	9600	0x06	2400	0x0C	38400
0x01	150	0x07	4800	0x0D	57600
0x02	300	0x08	7200	0x0E	76800
0x03	600	0x09	9600	0x0F	115200
0x04	1200	0x0A	14400	0x10	109700
0x05	1800	0x0B	19200		

Tab. 5-2: Übertragungsrate

**PROTOCOL**

Enthält das gewählte Protokoll. Diese Einstellung beeinflusst den weiteren Aufbau.

➔ Geben Sie für das ASCII-Protokoll den Wert 0x01 an.

**OPTION3****Zeichenrahmen**

Byte	Bit 7 ... 0	Beschreibung	
0	1, 0	<b>binär</b>	<b>Datenbits</b>
		00	5 Datenbits
		01	6 Datenbits
		10	7 Datenbits
		11	8 Datenbits
	3, 2	<b>binär</b>	<b>Parity</b>
		00	none
		01	odd
		10	even
		11	even
	5, 4	<b>binär</b>	<b>Stoppbits</b>
		01	1
		10	1,5
		11	2
	7, 6	<b>binär</b>	<b>Flusskontrolle</b>
		00	keine
		10	Hardware
		11	XON/XOFF

Tab. 5-3: Zeichenrahmen

Default: 0x13 (Datenbits: 8, Parität: keine, Stoppbit: 1, Flusskontrolle: keine)

Datenbits

Anzahl der Datenbits, auf die ein Zeichen abgebildet wird.

Parity

Zur Paritätskontrolle werden die Informationsbits um das Paritätsbit erweitert. Es ergänzt durch seinen Wert (**0** oder **1**) den Wert aller Bits auf einen vereinbarten Zustand.

Ist keine Parität vereinbart, wird das Paritätsbit auf **1** gesetzt, aber nicht ausgewertet.

Stoppbits

Die Stoppbits 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.

**Hardware-Flusskontrolle**

Bei der Hardware-Flusskontrolle werden die Leitungen RTS und CTS verwendet.

➔ Verdrahten Sie RTS und CTS entsprechend.

**Software-Flusskontrolle**

Die Software-Flusskontrolle verwendet zur Steuerung die Steuerzeichen XON=0x11 und XOFF=0x13.



Bei Verwendung der Software-Flusskontrolle dürfen die Daten diese zwei Steuerzeichen nicht enthalten.



#### OPTION4, 5 ZNA

##### **Wartezeit (ZNA) eingeben**

➔ Geben Sie ZNA in ms ein.

*Der nächste Sendeauftrag startet erst nach dieser Wartezeit.*

Option4            ZNA (High-Byte)

Option5            ZNA (Low-Byte)

Wertebereich    0 ... 65535 (ms)

Default: 0

#### OPTION6, 7 ZVZ

Die Zeichenverzugszeit (ZVZ) definiert den maximal zulässigen zeitlichen Abstand zwischen zwei empfangenen Zeichen innerhalb eines Telegramms.

##### **ZVZ in ms vorgeben**

➔ Geben Sie eine ZVZ zwischen 1 ... 65535 ein.

##### **Kommunikationsprozessor berechnet ZVZ**

➔ Geben Sie für ZVZ 0 ein.

*Der Kommunikationsprozessor berechnet anhand der Baudrate die ZVZ selbst (ca. doppelte Zeichenzeit).*

Option6            ZVZ (High-Byte)

Option7            ZVZ (Low-Byte)

Wertebereich    0 ... 65535 (ms)

Default: 250

#### OPTION8 Anzahl Receive- buffer

Legt die Anzahl der Empfangspuffer fest.

##### **1 Empfangspuffer verwenden**

➔ Geben Sie den Wert **1** ein.

*Ist der Eingangspuffer belegt, können keine weiteren Daten empfangen werden.*

##### **2 ... 250 Empfangspuffer verwenden**

➔ Geben Sie einen Wert zwischen 2 ... 250 ein.

*Die empfangenen Daten werden in einen noch freien Empfangspuffer umgeleitet.*

Wertebereich    0 ... 250

Default: 1

## 5.3 STX/ETX

### 5.3.1 Funktion

#### Funktionsweise

STX/ETX ist ein einfaches Protokoll mit Header und Trailer. STX/ETX wird zur Übertragung von ASCII-Zeichen (0x20 ... 0x7F) 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 Kommunikationsprozessor übergeben und von dort, mit STX als Startzeichen und ETX als Schlusszeichen, an den Kommunikationspartner übertragen.

#### Telegrammaufbau

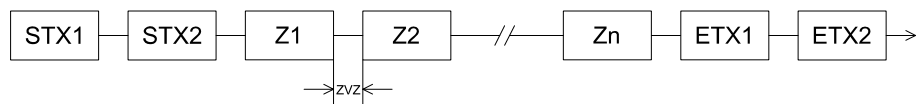


Abb. 5-1: Telegrammaufbau



Sie können bis zu 2 Anfangs- und Endezeichen frei definieren und eine Wartezeit (ZNA) für den Sender vorgeben.

### 5.3.2 Parametrierdaten

- DS     Datensatz für Zugriff über CPU, PROFIBUS und PROFINET  
 IX     Index für Zugriff über CANopen  
 SX     Subindex für den Zugriff über EtherCAT

Name	Bytes	Funktion	Default	DS	IX	SX
PII_L	1	Länge Prozessabbild Eingabedaten	Werte hängen vom übergeordneten System ab.	0x02	0x3100	0x01
PIQ_L	1	Länge Prozessabbild Ausgabedaten		0x02	0x3101	0x02
DIAG_EN	1	Diagnosealarm	0x00	0x00	0x3102	0x03
BAUD	1	Baudrate	0x00	0x80	0x3103	0x04
PROTOCOL	1	Protokoll	0x02	0x80	0x3104	0x05
OPTION3	1	Zeichenrahmen	0x13	0x80	0x3105	0x06
OPTION4, 5	2	ZNA 0 ... 65535 (ms)	0	0x80	0x3106 ... 0x3107	0x07 ... 0x08
OPTION6, 7	2	TMO 0 ... 65535 (ms)	250	0x80	0x3108 ... 0x3109	0x09 ... 0x0A
OPTION8	1	Anzahl Startkennungen	1	0x80	0x310A	0x0B
OPTION9	1	Startkennung 1	3	0x80	0x310B	0x0C
OPTION10	1	Startkennung 2	0	0x80	0x310C	0x0D
OPTION11	1	Anzahl Endekennungen	1	0x80	0x310D	0x0E
OPTION12	1	Endekennung 1	3	0x80	0x310E	0x0F
OPTION13	1	Endekennung 2	0	0x80	0x310F	0x10
OPTION14	1	reserviert	0x00	0x80	0x3110	0x11

Tab. 5-4: Parametrierdaten bei STX/ETX

## DIAG\_EN Diagnosealarm

➔ Aktivieren bzw. deaktivieren Sie hier die Diagnosefunktion.

Wertebereich 0x00: sperren  
0x40: freigeben

Default: 0x00

## BAUD Übertragungsrate

➔ Geben Sie die Geschwindigkeit der Datenübertragung in Bit/s (Baud) an. Sie haben folgende Einstellbereiche; andere Werte sind nicht zulässig:  
Default: 0x00 (9600 Baud)

**Wertebereich:**

Hex	Baud	Hex	Baud	Hex	Baud
0x00	9600	0x06	2400	0x0C	38400
0x01	150	0x07	4800	0x0D	57600
0x02	300	0x08	7200	0x0E	76800
0x03	600	0x09	9600	0x0F	115200
0x04	1200	0x0A	14400	0x10	109700
0x05	1800	0x0B	19200		

Tab. 5-5: Übertragungsrate

## PROTOCOL

Enthält das gewählte Protokoll. Diese Einstellung beeinflusst den weiteren Aufbau.

➔ Geben Sie für das STX/ETX-Protokoll den Wert 0x02 an.

## OPTION3 Zeichenrahmen

Byte	Bit 7 ... 0	Beschreibung	
0	1, 0	binär	<b>Datenbits</b>
		00	5 Datenbits
		01	6 Datenbits
		10	7 Datenbits
		11	8 Datenbits
	3, 2	binär	<b>Parity</b>
		00	none
		01	odd
		10	even
		11	even
	5, 4	binär	<b>Stoppbits</b>
		01	1
		10	1,5
		11	2
	7, 6	binär	<b>Flusskontrolle</b>
		00	keine
		10	Hardware
		11	XON/XOFF

Tab. 5-6: Zeichenrahmen

Default: 0x13 (Datenbits: 8, Parität: keine, Stoppbit: 1, Flusskontrolle: keine)

Datenbits

Anzahl der Datenbits, auf die ein Zeichen abgebildet wird.

Parity

Zur Paritätskontrolle werden die Informationsbits um das Paritätsbit erweitert. Es ergänzt durch seinen Wert (**0** oder **1**) den Wert aller Bits auf einen vereinbarten Zustand.

Ist keine Parität vereinbart, wird das Paritätsbit auf **1** gesetzt, aber nicht ausgewertet.

Stoppbits

Die Stoppbits 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.

**Hardware-Flusskontrolle**

Bei der Hardware-Flusskontrolle werden die Leitungen RTS und CTS verwendet.

➔ Verdrahten Sie RTS und CTS entsprechend.

**Software-Flusskontrolle**

Die Software-Flusskontrolle verwendet zur Steuerung die Steuerzeichen XON=0x11 und XOFF=0x13.



Bei Verwendung der Software-Flusskontrolle dürfen die Daten diese zwei Steuerzeichen nicht enthalten.

**OPTION4, 5  
ZNA****Wartezeit (ZNA) eingeben**

➔ Geben Sie ZNA in ms ein.

*Der nächste Sendeauftrag startet erst nach dieser Wartezeit.*

Option4	ZNA (High-Byte)	
Option5	ZNA (Low-Byte)	
Wertebereich	0 ... 65535 (ms)	Default: 0

**OPTION6, 7  
TMO****Maximal zulässiger zeitlicher Abstand zwischen zwei Telegrammen (TMO)**

➔ Geben Sie TMO in ms ein.

*Das nächste Telegramm muss innerhalb der eingegebenen Zeit starten.*

Option6	TMO (High-Byte)	
Option7	TMO (Low-Byte)	
Wertebereich	0 ... 65535 (ms)	Default: 250

**OPTION8**  
**Anzahl Startkennungen**

**Anzahl der Startkennungen einstellen**

➔ Stellen Sie 1 oder 2 Startkennungen ein.

*Ist 1 als Anzahl der Startkennungen eingestellt, wird der Inhalt des 2. Startkennzeichens ignoriert.*

Bereich 0 ... 2

Default: 1

**OPTION9, 10**  
**Startkennung 1**  
**Startkennung 2**

Die Startkennung wird einem Telegramm voraus gesendet und kennzeichnet den Start einer Übertragung.

Startkennung 1, 2 Bereich: 0 ... 255

Default: 3 (Startkennung 1)  
0 (Startkennung 2)

**ASCII-Wert des Startzeichens eingeben: 1 Startkennung**

✓ Der voreingestellte Wert für Startkennung 2 ist 0.

➔ Geben Sie unter **Startkennung 1** (Option9) einen Wert im Bereich 0 ... 255 ein.

*Vor dem Telegramm wird 1 Startkennung gesendet.*

**ASCII-Wert der Startzeichen eingeben: 2 Startkennungen**

➔ Geben Sie unter **Anzahl Startkennungen** (Option8) eine 2 ein.

➔ Geben Sie unter **Startkennung 1** (Option9) einen Wert im Bereich 0 ... 255 ein.

➔ Geben Sie unter **Startkennung 2** (Option10) einen Wert im Bereich 0 ... 255 ein.

*Vor dem Telegramm werden 2 Startkennungen gesendet.*

**OPTION11**  
**Anzahl Endekennungen**

**Anzahl der Endekennungen einstellen**

➔ Stellen Sie 1 oder 2 Endekennungen ein.

*Ist 1 als Anzahl der Endekennungen eingestellt, wird der Inhalt des 2. Endekennzeichens ignoriert.*

Bereich 0 ... 2

Default: 1

**OPTION12, 13**  
**Endekennung 1**  
**Endekennung 2**

Die Endekennung folgt nach einem Telegramm und kennzeichnet das Ende einer Übertragung.

Endekennung 1, 2 Bereich: 0 ... 255

Default: 3 (Endekennung 1)  
0 (Endekennung 2)

**ASCII-Wert des Endezeichens eingeben: 1 Endekennung**

✓ Der voreingestellte Wert für Endekennung 2 ist 0.

➔ Geben Sie unter **Endekennung 1** (Option12) einen Wert im Bereich 0 ... 255 ein.

*Nach dem Telegramm wird 1 Endekennung gesendet.*

**ASCII-Wert der Endezeichen eingeben: 2 Endekennungen**

➔ Geben Sie unter **Anzahl Endekennungen** (Option11) eine 2 ein.

➔ Geben Sie unter **Endekennung 1** (Option12) einen Wert im Bereich 0 ... 255 ein.

➔ Geben Sie unter **Endekennung 2** (Option13) einen Wert im Bereich 0 ... 255 ein.

*Nach dem Telegramm werden 2 Endekennungen gesendet.*

## 5.4 3964(R)

### 5.4.1 Funktion

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

Das 3964-Protokoll unterscheidet sich vom 3964R-Protokoll nur durch das Fehlen des Cyclic\_Redundancy\_Check (CRC). CRC ermöglicht eine zuverlässigere Übertragung.

Folgende Steuerzeichen werden ausgewertet:

- STX Start of Text
- DLE Data Link Escape
- ETX End of Text
- BCC Block Check Character (nur 3964R, BCC ist die Prüfsumme des 3964R-Protokolls)
- NAK Negative Acknowledge



#### DLE-Verdoppelung

Wird eine DLE als Informationszeichen übertragen, so wird dieses zur Unterscheidung vom Steuerzeichen DLE beim Auf- und Abbau der Verbindung doppelt gesendet. Der Empfänger macht die DLE-Verdoppelung rückgängig.



#### Priorität der Kommunikationspartner

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

#### Ablauf

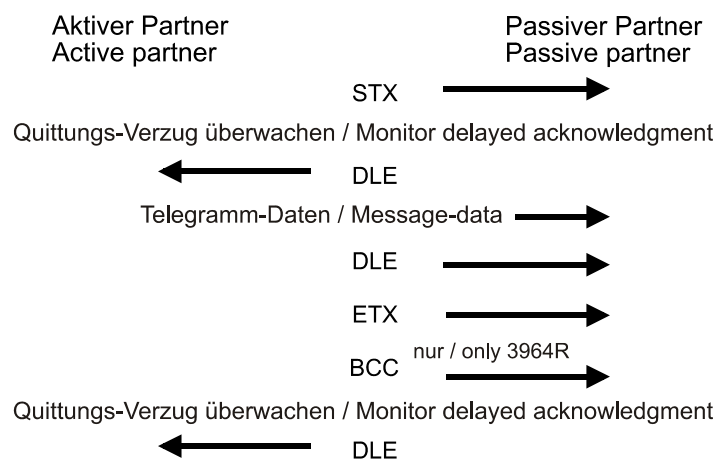


Abb. 5-2: Ablauf



Der Kommunikationsprozessor überträgt pro Telegramm maximal 250 Byte.

## Time-out-Zeiten

### Zeichenverzugszeit (ZVZ)

- ZVZ wird während des gesamten Telegramm-Empfangs überwacht.

### Quittungsverzugszeit (QVZ)

- QVZ wird zwischen STX und DLE sowie zwischen BCC und DLE überwacht.
- Nach STX: 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.
- Nach dem Telegramm: Bei Verstreichen der QVZ nach dem Telegramm (nach BCC-Byte) oder bei Empfang eines Zeichens ungleich DLE werden der Verbindungsaufbau und das Telegramm wiederholt. Nach 5 Versuchen wird ein NAK gesendet und der Verbindungsaufbau abgebrochen.

## 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.

## 3964R: 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 Datend 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 Kommunikationspartner gleichzeitig innerhalb der QVZ einen Verbindungsaufbau, 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 bei 3964R immer die Kombination DLE/ETX/BCC.

## Steuercodes

Befehl	Code
STX	0x02
ETX	0x03
DLE	0x10
NAK	0x15

## 5.4.2 Parametrierdaten

DS Datensatz für Zugriff über CPU, PROFIBUS und PROFINET  
IX Index für Zugriff über CANopen  
SX Subindex für den Zugriff über EtherCAT

Name	Bytes	Funktion	Default	DS	IX	SX
PII_L	1	Länge Prozessabbild Eingabedaten	Werte hängen vom übergeordneten System ab	0x02	0x3100	0x01
PIQ_L	1	Länge Prozessabbild Ausgabedaten		0x02	0x3101	0x02
DIAG_EN	1	Diagnosealarm	0x00	0x00	0x3102	0x03
BAUD	1	Baudrate	0x00	0x80	0x3103	0x04
PROTOCOL	1	Protokoll	0x01	0x80	0x3104	0x05
OPTION3	1	Zeichenrahmen	0x13	0x80	0x3105	0x06
OPTION4	2	ZNA (x 20 ms)	0	0x80	0x3106	0x07
OPTION5	2	ZVZ (x 20 ms)	10	0x80	0x3107	0x08
OPTION6	2	QVZ (x 20 ms)	10	0x80	0x3108	0x09
OPTION7	2	BWZ (x 20 ms)	10	0x80	0x3109	0x0A
OPTION8	1	STX Wiederholungen	5	0x80	0x310A	0x0B
OPTION9	1	DBL	6	0x80	0x310B	0x0C
OPTION10	1	Priorität	0	0x80	0x310C	0x0D
OPTION11 ... 14	4	reserviert	0x00	0x80	0x310D ... 0x3110	0x0E ... 0x11

Tab. 5-7: Parametrierdaten bei 3964(R)

### DIAG\_EN Diagnosealarm

➔ Aktivieren bzw. deaktivieren Sie hier die Diagnosefunktion.

Wertebereich 0x00: sperren  
0x40: freigeben

Default: 0x00

### BAUD Übertragungsrate

➔ Geben Sie die Geschwindigkeit der Datenübertragung in Bit/s (Baud) an.

Sie haben folgende Einstellbereiche; andere Werte sind nicht zulässig:

Default: 0x00 (9600 Baud)

**Wertebereich:**

Hex	Baud	Hex	Baud	Hex	Baud
0x00	9600	0x06	2400	0x0C	38400
0x01	150	0x07	4800	0x0D	57600
0x02	300	0x08	7200	0x0E	76800
0x03	600	0x09	9600	0x0F	115200
0x04	1200	0x0A	14400	0x10	109700
0x05	1800	0x0B	19200		

Tab. 5-8: Übertragungsrate



## PROTOCOL

Enthält das gewählte Protokoll. Diese Einstellung beeinflusst den weiteren Aufbau.

- ➔ Geben Sie für das 3964-Protokoll den Wert 0x03 an.
- ➔ Geben Sie für das 3964R-Protokoll den Wert 0x04 an.

Wertebereich 0x03: 3964  
0x04: 3964R

Default: 0x03

## OPTION3 Zeichenrahmen

Byte	Bit 7 ... 0	Beschreibung	
0	1, 0	binär	<b>Datenbits</b>
		00	5 Datenbits
		01	6 Datenbits
		10	7 Datenbits
		11	8 Datenbits
	3, 2	binär	<b>Parity</b>
		00	none
		01	odd
		10	even
		11	even
	5, 4	binär	<b>Stoppbits</b>
		01	1
		10	1,5
		11	2
	7, 6		<b>reserviert</b>

Tab. 5-9: Zeichenrahmen

Default: 0x13 (Datenbits: 8, Parität: keine, Stoppbit: 1)

Datenbits

Anzahl der Datenbits, auf die ein Zeichen abgebildet wird.

Parity

Zur Paritätskontrolle werden die Informationsbits um das Paritätsbit erweitert. Es ergänzt durch seinen Wert (**0** oder **1**) den Wert aller Bits auf einen vereinbarten Zustand.

Ist keine Parität vereinbart, wird das Paritätsbit auf **1** gesetzt, aber nicht ausgewertet.

Stoppbits

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

## OPTION4 ZNA

### Wartezeit (ZNA) eingeben

- ➔ Geben Sie ZNA als Faktor von 20 ms-Schritten ein.

Der nächste Sendeauftrag startet erst nach dieser Wartezeit.

Wertebereich 0 ... 255

Default: 0

## OPTION5 ZVZ

Die Zeichenverzugszeit (ZVZ) definiert den maximal zulässigen zeitlichen Abstand zwischen zwei empfangenen Zeichen innerhalb eines Telegramms.

### ZVZ vorgeben

- ➔ Geben Sie eine ZVZ zwischen 1 ... 255 als Faktor von 20 ms-Schritten ein.

### Kommunikationsprozessor berechnet ZVZ

- ➔ Geben Sie für ZVZ 0 ein.

Der Kommunikationsprozessor berechnet 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.

**Quittungsverzugszeit (QVZ) eingeben**

➔ Geben Sie QVZ als Faktor von 20 ms-Schritten ein.

Wertebereich 0 ... 255

Default: 10

**OPTION7  
BWZ**

Die Blockwartezeit (BWZ) ist die maximale Zeitdauer zwischen der Bestätigung eines Anforderungstelegrams (DLE) und STX des Reaktionstelegrams.

**Blockwartezeit (BWZ) eingeben**

➔ Geben Sie BWZ als Faktor von 20 ms-Schritten ein.

Wertebereich 0 ... 255

Default: 10

**OPTION8  
STX-Wiederholungen**

Maximale Anzahl der Versuche des Kommunikationsprozessor, eine Verbindung aufzubauen.

**STX-Wiederholungen**

➔ Geben Sie die max. Anzahl der STX-Wiederholungen ein.

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.

**Wiederholungen für das Anforderungstelegramm (DBL) eingeben**

➔ Geben Sie für DBL einen Wert zwischen 0 ... 255 ein.

*Sind die Versuche erfolglos, wird die Übertragung abgebrochen.*

Wertebereich 0 ... 255

Default: 6

**OPTION10  
Priorität**



Ein Kommunikationspartner hat hohe Priorität (high), wenn sein Sendeversuch Vorrang gegenüber dem Sendewunsch des Kommunikationspartner hat.

Bei 3964(R) müssen die Prioritäten der Kommunikationspartner unterschiedlich sein.

**Priorität einstellen**

➔ Geben Sie 0x00 für niedrige Priorität (low) ein.

➔ Geben Sie 0x01 für hohe Priorität (high) ein.

Wertebereich 0x00: low  
0x01: high

Default: 0

## 5.5 Modbus

### 5.5.1 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-Verbindung unter RS232 oder als Mehrpunkt-Verbindung unter RS485.

#### Master-Slave-Kommunikation

Buskonflikte treten nicht auf, da der Master immer nur mit einem Slave kommunizieren kann. Nach einer Anforderung vom Master wartet dieser auf die Antwort des Slaves, bis eine einstellbare Wartezeit abgelaufen ist. Während des Wartens ist eine Kommunikation mit einem anderen Slave nicht möglich.

#### Telegrammaufbau

Die Anforderungstelegramme, die ein Master sendet, und die Antworttelegramme eines Slaves haben den gleichen Aufbau:

Start- zeichen	Slave- Adresse	Funktions- Code	Daten	Fluss- kontrolle	Ende- zeichen
-------------------	-------------------	--------------------	-------	---------------------	------------------

#### Broadcast mit Slave-Adresse = 0

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

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 Anfangs- und Ende-Zeichen gekennzeichnet. Dies macht die Übertragung transparent aber langsam.
- **RTU-Modus:** Jedes Byte wird als ein Zeichen übertragen. Hierdurch wird ein höherer Datendurchsatz als im ASCII-Modus erreicht. Anstelle von Anfangs- und Ende-Zeichen wird eine Zeitüberwachung eingesetzt.

Den Modus wählen Sie bei der Parametrierung.

### 5.5.2 Modbus auf dem Kommunikationsprozessor

#### Modbus-Betriebsarten

Der Kommunikationsprozessor unterstützt folgende Modbus-Betriebsarten:

- Modbus Master
- Modbus Slave short
- Modbus Slave long

#### 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 zurückgreifen.

Im **Modbus Master**-Betrieb überträgt Modbus bis zu 250 Byte Nutzdaten pro Telegramm.

#### Modbus Slave short

Im **Modbus Slave short**-Betrieb kommuniziert der Kommunikationsprozessor mit einem Modbus Master. Abhängig vom Funktionscode empfängt der Kommunikationsprozessor Daten vom Modbus Master oder stellt Daten bereit. Das Datenhandling auf Slave-Seite erfolgt automatisch.

Im **Modbus Slave short**-Betrieb überträgt Modbus bis zu 250 Byte Nutzdaten pro Telegramm.

## Modbus Slave long

Im **Modbus Slave long**-Betrieb wird nur ein geänderter Datenbereich, beginnend bei 0, vom Kommunikationsprozessor an das übergeordnete System übertragen.

Fordert der Modbus Master Daten an, muss mit einem Anwenderprogramm dafür gesorgt werden, dass sich die relevanten Daten im Kommunikationsprozessor befinden.

Schreibende Master-Zugriffe dürfen nicht außerhalb des Empfangsbereichs liegen!



### HINWEIS

Erst nachdem alle Daten im Kommunikationsprozessor vorliegen, sendet er ein Antworttelegramm an den Master!

## 5.5.3 Parametrierdaten

DS Datensatz für Zugriff über CPU, PROFIBUS und PROFINET

IX Index für Zugriff über CANopen

SX Subindex für den Zugriff über EtherCAT

Name	Bytes	Funktion	Default	DS	IX	SX
PII_L	1	Länge Prozessabbild Eingabedaten	Werte hängen vom übergeordneten System ab	0x02	0x3100	0x01
PIQ_L	1	Länge Prozessabbild Ausgabedaten		0x02	0x3101	0x02
DIAG_EN	1	Diagnosealarm	0x00	0x00	0x3102	0x03
BAUD	1	Baudrate	0x00	0x80	0x3103	0x04
PROTOCOL	1	Protokoll	0x0B	0x80	0x3104	0x05
OPTION3	1	Zeichenrahmen	0x13	0x80	0x3105	0x06
OPTION4	1	Slave-Adresse	1	0x80	0x3106	0x07
OPTION5, 6	2	Wartezeit	0	0x80	0x3107 ... 0x3108	0x08 ... 0x09
OPTION7 ... 14	8	reserviert	0x00	0x80	0x3107 ... 0x3110	0x0A ... 0x11

Tab. 5-10: Parametrierdaten

### DIAG\_EN Diagnosealarm

➔ Aktivieren bzw. deaktivieren Sie hier die Diagnosefunktion.

Wertebereich 0x00: sperren

Default: 0x00

0x40: freigeben

### BAUD Übertragungsrate

➔ Geben Sie die Geschwindigkeit der Datenübertragung in Bit/s (Baud) an.

Sie haben folgende Einstellbereiche; andere Werte sind nicht zulässig:

Default: 0x00 (9600 Baud)

#### Wertebereich:

Hex	Baud	Hex	Baud	Hex	Baud
0x00	9600	0x06	2400	0x0C	38400
0x01	150	0x07	4800	0x0D	57600
0x02	300	0x08	7200	0x0E	76800
0x03	600	0x09	9600	0x0F	115200
0x04	1200	0x0A	14400	0x10	109700
0x05	1800	0x0B	19200		

Tab. 5-11: Übertragungsrate

**PROTOCOL**

Enthält das gewählte Protokoll. Diese Einstellung beeinflusst den weiteren Aufbau. Default-Wert: 0x0B (Modbus Master).

➔ Geben Sie für das gewünschte Modbus-Protokoll den Wert aus der Tabelle ein.

ASCII-Modus	Wert	RTU-Modus	Wert
Modbus Master	0x0A	Modbus Master	0x0B
Modbus Slave short	0x0C	Modbus Slave short	0x0D
Modbus Slave long	0x1C	Modbus Slave long	0x1D

Tab. 5-12: PROTOCOL

**OPTION3  
Zeichenrahmen**

Byte	Bit 7 ... 0	Beschreibung	
0	1, 0	binär	<b>Datenbits</b>
		00	5 Datenbits
		01	6 Datenbits
		10	7 Datenbits
		11	8 Datenbits
	3, 2	binär	<b>Parity</b>
		00	none
		01	odd
		10	even
		11	even
	5, 4	binär	<b>Stoppbits</b>
		01	1
		10	1,5
		11	2
	7, 6		<b>reserviert</b>

Tab. 5-13: Zeichenrahmen

Default: 0x13 (Datenbits: 8, Parität: keine, Stoppbit: 1)

Datenbits

Anzahl der Datenbits, auf die ein Zeichen abgebildet wird.

Parity

Zur Paritätskontrolle werden die Informationsbits um das Paritätsbit erweitert. Es ergänzt durch seinen Wert (**0** oder **1**) den Wert aller Bits auf einen vereinbarten Zustand.

Ist keine Parität vereinbart, wird das Paritätsbit auf **1** gesetzt, aber nicht ausgewertet.

Stoppbits

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

**OPTION4  
Slave-Adresse**

Unter Modbus Master wird der Parameter ignoriert.

**Adresse für den Modbus-Slave eingeben**

➔ Geben Sie unter dem Slave-Protokoll eine Adresse für den Modbus Slave an.

Mit dieser Adresse greift Modbus auf die Funktionscodes zu.

Wertebereich 0 ... 255

Default: 1

## OPTION5, 6 Wartezeit

Unter Modbus Slave wird der Parameter ignoriert.

### **Wartezeit für den Modbus-Master vorgeben**

→ Geben Sie für den Modbus Master eine Wartezeit in ms vor. Mit 0 wird die Wartezeit protokollabhängig nach folgender Formel automatisch ermittelt:

Modbus ASCII    50 ms + (2926000 ms / Baudrate) x Bit/s    mit Baudrate in Bit/s

Modbus RTU       50 ms + (5190000 ms / Baudrate) x Bit/s    mit Baudrate in Bit/s

Option5: Wartezeit (High-Byte)

Option6: Wartezeit (Low-Byte)

Wertebereich: 0...60000 (ms)

Default: 0

## 5.6 Modbus - Einsatz

### 5.6.1 Übersicht

#### IO-Size

Die Anzahl der Ein- und Ausgangsdaten ist abhängig von der IO-Size. Sie ist beim Kommunikationsprozessor über GSD parametrierbar.

#### Voraussetzung für den Betrieb

Folgende Komponenten sind zum Einsatz der Modbus-Module erforderlich:

- Master-System bestehend aus dem Cube20S-System mit Kommunikationsprozessor
- Slave-System bestehend aus Cube20S-System mit Kommunikationsprozessor
- Siemens SIMATIC Manager bzw. WinPLC7 von Murrelektronik
- GSD-Datei
- Murrelektronik-Bausteine Fx000011\_Vxxx.zip
- Serielle Verbindung zwischen beiden Kommunikationsprozessoren

#### Parametrierung

Führen Sie für den Modbus-Kommunikationsprozessor immer eine Hardware-Konfiguration durch. Hierfür ist die Einbindung der GSD-Datei von Murrelektronik im Hardware-Katalog erforderlich.

#### **Vorgehensweise zur Parametrierung**

- 1 | Projektieren Sie ein Cube20S-System.
- 2 | Installieren Sie die ausgewählte GSD-Datei im Hardware-Katalog.
- 3 | Starten Sie den Siemens SIMATIC Manager bzw. WinPLC7 von Murrelektronik.
- 4 | Binden Sie einen mit **Modbus** bezeichneten Kommunikationsprozessor an.
- 5 | Parametrieren Sie den Kommunikationsprozessor nach Ihren Vorgaben.
- 6 | Übertragen Sie Ihr Projekt in die SPS.

#### SPS-Programm

Mit Ausnahme von **Modbus Slave short** ist für die Kommunikation ein SPS-Programm erforderlich. Die Kommunikation findet über Bausteinestatt, die Sie in Form der Murrelektronik-Library Fx000011-Vxxx.zip in Ihr Konfigurationsstool einbinden.

Sie finden diese auch unter [www.murrelektronik](http://www.murrelektronik) im Servicebereich.

## 5.6.2 Kommunikation

### Kommunikations- möglichkeiten

Nachfolgend finden Sie die Kommunikationsmöglichkeiten zwischen Modbus Master und Modbus Slave.

Kombinationen:

- Modbus Master ↔ Modbus Slave short
- Modbus Master ↔ Modbus long

### Modbus Master ↔ Modbus Slave short

#### Modbus Master

Im Master-Modus kommuniziert der Kommunikationsprozessor über die Bausteine FB 60 - SEND und FB 61 - RECEIVE. Er überträgt bis zu 250 Byte Nutzdaten pro Telegramm.

#### Modbus Slave short

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

## Vorgehensweise

### Systemaufbau

- ➔ Bauen Sie für Master- und Slave-Seite je ein Cube20S-System auf, das jeweils einen Kommunikationsprozessor enthält.
- ➔ Verbinden Sie beide Systeme über die serielle Schnittstelle.

### Modbus Master projektieren

- ➔ Projektieren Sie die Master-Seite .
- ➔ Parametrieren Sie den Kommunikationsprozessor über die Hardware-Konfiguration.

Für die Kommunikation ist ein SPS-Anwenderprogramm erforderlich, das nach der Struktur in der Tabelle (Tab. 5-14: ) aufgebaut ist.

OB 100	Einmaliger Aufruf der Bausteine <b>FB 60 - SEND</b> und <b>FB 61 - RECEIVE</b> mit allen Parametern und gesetztem <b>R</b> zur Initialisierung.
OB 1	Aufruf des <b>FB 60 - SEND</b> mit Fehlerauswertung. Das SPS-Anwenderprogramm legt das Telegramm gemäß den Modbus-Vorgaben im Sendebaustein an.
	Aufruf des <b>FB 61 - RECEIVE</b> mit Fehlerauswertung. Das SPS-Anwenderprogramm legt die Daten gemäß den Modbus-Vorgaben im Empfangsbaustein ab.

Tab. 5-14: Struktur des SPS-Anwenderprogramms

### Modbus Slave short projektieren

- ➔ Projektieren Sie die Slave-Seite.
- ➔ Parametrieren Sie den Kommunikationsprozessors über die Hardware-Konfiguration.
- ➔ Geben Sie für Ein- und Ausgabe-Bereich die Startadresse an, ab der die Daten liegen (abhängig von der IO-Size).

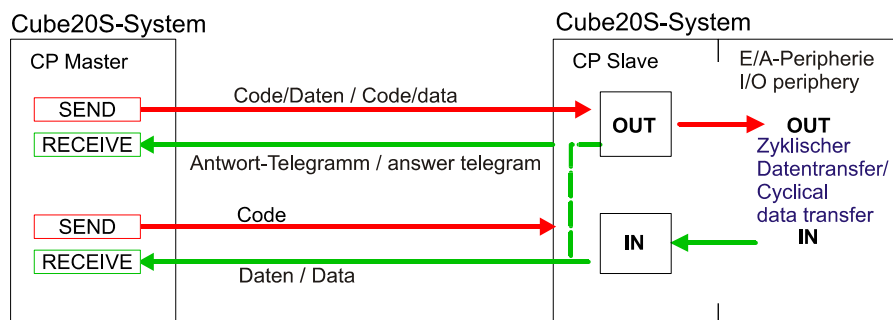


Abb. 5-3: System Cube20S Master <-> Slave short

## Master ↔ Slave long

### Modbus Master

Im Master-Modus kommuniziert der Kommunikationsprozessor über die Bausteine FB 60 - SEND und FB 61 - RECEIVE. Er überträgt bis zu 250 Byte Nutzdaten pro Telegramm.

### Modbus Slave long

Im Modbus Slave long-Modus überträgt der Kommunikationsprozessor mit FB 61 - RECEIVE nur einen geänderter Datenbereich beginnend bei 0 an die CPU. Fordert der Master Daten an, müssen sich die relevanten Daten im Kommunikationsprozessor befinden. Mit einem FB 60 - SEND überträgt die CPU einen gewünschten Datenbereich von 0 beginnend in den Kommunikationsprozessor.



## Vorgehensweise

### Systemaufbau

- ➔ Bauen Sie für Master- und Slave-Seite je ein Cube20S-System auf, das jeweils einen Kommunikationsprozessor enthält.
- ➔ Verbinden Sie beide Systeme über die serielle Schnittstelle.

### Modbus Master projektieren

- ➔ Projektieren Sie die Master-Seite.
- ➔ Parametrieren Sie den Kommunikationsprozessor über die Hardware-Konfiguration.

Für die Kommunikation ist ein SPS-Anwenderprogramm erforderlich, das nach der Struktur in der Tabelle (Tab. 5-15: ) aufgebaut ist.

OB 100	Einmaliger Aufruf der Bausteine <b>FB 60 - SEND</b> und <b>FB 61 - RECEIVE</b> mit allen Parametern und gesetztem <b>R</b> zur Initialisierung.
OB 1	Aufruf des <b>FB 60 - SEND</b> mit Fehlerauswertung. Das SPS-Anwenderprogramm legt das Telegramm gemäß den Modbus-Vorgaben im Sendebaustein an.
	Aufruf des <b>FB 61 - RECEIVE</b> mit Fehlerauswertung. Das SPS-Anwenderprogramm legt die Daten gemäß den Modbus-Vorgaben im Empfangsbaustein ab.

Tab. 5-15: Struktur des SPS-Anwenderprogramms

### Modbus Slave long projektieren

- ➔ Projektieren Sie die Slave-Seite.
- ➔ Parametrieren Sie den Kommunikationsprozessor über die Hardware-Konfiguration.

Für die Kommunikation ist ein SPS-Anwenderprogramm erforderlich, das nach der Struktur in der Tabelle (Tab. 5-16: ) aufgebaut ist.

OB 100	Einmaliger Aufruf der Bausteine <b>FB 60 - SEND</b> und <b>FB 61 - RECEIVE</b> mit allen Parametern und gesetztem <b>R</b> zur Initialisierung.
OB 1	Aufruf des <b>FB 60 - SEND</b> mit Fehlerauswertung. Das SPS-Anwenderprogramm legt einen Bereich von 0 beginnend im Kommunikationsprozessor ab, auf den der Master über Modbus zugreifen kann.
	Aufruf des <b>FB 61 - RECEIVE</b> mit Fehlerauswertung. Der Kommunikationsprozessor überträgt einen Datenbereich in die CPU.
	Bei Änderung der Daten durch den Master werden nur die Daten an die CPU übergeben, in denen geändert wurde.

Tab. 5-16: Struktur des SPS-Anwenderprogramms

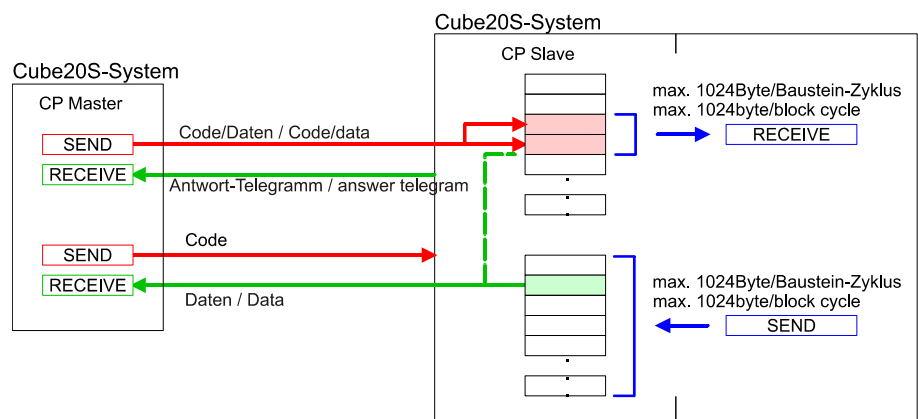


Abb. 5-4: System Cube20S Master ↔ Slave long

## 5.7 Modbus - Funktionscodes

### 5.7.1 Definitionen

#### Namenskonventionen für Modbus

Modbus unterscheidet zwischen Bit- und Wortzugriff

- Bit = **Coil**
- Wort = **Register**
- Bit-Eingang = **Input Status**
- Bit-Ausgang = **Coil Status**
- Wort-Eingang = **Input Register**
- Wort-Ausgang = **Holding Register**

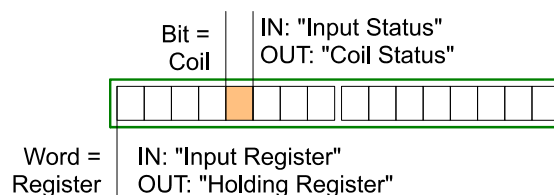


Abb. 5-5: Namenskonventionen

#### Bereichsdefinitionen

Üblicherweise greift Modbus mittels der Bereiche 0x, 1x, 3x und 4x zu.

- 0x und 1x: Zugriff auf digitale Bit-Bereiche
- 3x und 4x: Zugriff auf analoge Wort-Bereiche



#### HINWEIS

Beim Kommunikationsprozessor findet keine Unterscheidung zwischen Digital- und Analogdaten statt, deshalb gilt die Zuordnung in der Tabelle (Tab. 5-17: ).

Bereich	Zugriff über Funktionscode
0x	Bit-Bereich für Ausgabe-Daten des Masters
1x	Bit-Bereich für Eingabe-Daten des Masters
3x	Wort-Bereich für Eingabe-Daten des Masters
4x	Wort-Bereich für Ausgabe-Daten des Masters

Tab. 5-17: Zuordnung der Bit- und Wort-Bereiche

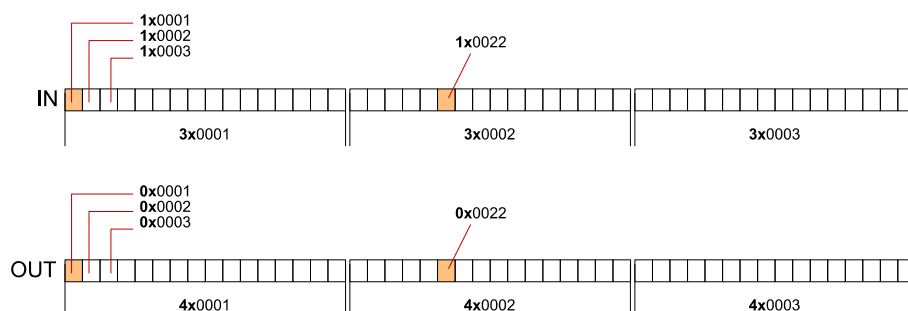


Abb. 5-6: Bereichsdefinitionen

## 5.7.2 Übersicht

### Zugriff

Mit folgenden Funktionscodes können Sie von einem Modbus Master auf einen Slave zugreifen. Die Funktionscodes sind aus Sicht des Masters beschrieben.

Code	Befehl	Beschreibung
0x01	Read n Bits	n Bit lesen von Master-Ausgabe-Bereich 0x
0x02	Read n Bits	n Bit lesen von Master-Eingabe-Bereich 1x
0x03	Read n Words	n Worte lesen von Master-Ausgabe-Bereich 4x
0x04	Read n Words	n Worte lesen von Master-Eingabe-Bereich 3x
0x05	Write 1 Bit	1 Bit schreiben in Master-Ausgabe-Bereich 0x
0x06	Write 1 Word	1 Wort schreiben in Master-Ausgabe-Bereich 4x
0x0F	Write n Bits	n Bit schreiben in Master-Ausgabe-Bereich 0x
0x10	Write n Words	n Worte schreiben in Master-Ausgabe-Bereich 4x

Tab. 5-18: Zugriff

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

Die Funktionscodes sind aus Sicht des Masters beschrieben.

Daten, die der Master an den Slave schickt, werden bis zu ihrem Ziel als **Ausgabe-**Daten (OUT) bezeichnet.

Daten, die der Master vom Slave empfängt, werden als **Eingabe-**Daten (IN) bezeichnet.

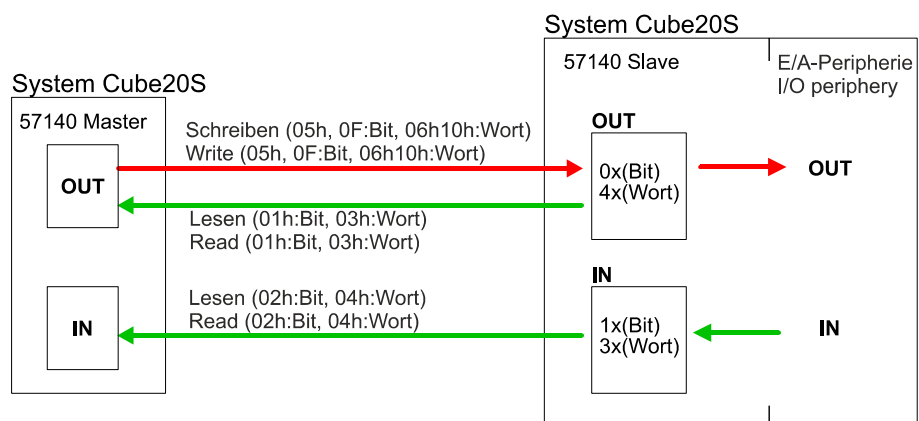


Abb. 5-7: Sichtweise für Ein- und Ausgabedaten

### Antwort des Slaves

Liefert der Slave einen Fehler zurück, wird der Funktionscode mit **OR 0x80** zurückgesendet.

Ist kein Fehler aufgetreten, wird der Funktionscode zurückgeliefert.

Slave-Antwort:	Funktionscode OR 0x80	→	Fehler
	Funktionscode	→	OK



### HINWEIS

Für die Byte-Reihenfolge im Wort gilt immer:

1 Wort	
High-Byte	Low-Byte

**Prüfsumme  
CRC oder LRC**
**CRC im RTU-Modus, LRC im ASCII-Modus**

Die Prüfsummen werden automatisch an jedes Telegramm angehängt. Sie werden nicht im Datenbaustein angezeigt.

**Slave-Adresse**

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

### 5.7.3 Funktionscodes

**Read n Bits 0x01, 0x02**

Code 0x01: n Bit lesen von Master-Ausgabe-Bereich 0x

Code 0x02: n Bit lesen von Master-Eingabe-Bereich 1x

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

Tab. 5-19: Kommando-Telegramm

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

Tab. 5-20: Antwort-Telegramm

**Read n Words  
0x03, 0x04**

0x03: n Worte lesen von Master-Ausgabe-Bereich 4x

0x04: n Worte lesen von Master-Eingabe-Bereich 3x

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

Tab. 5-21: Kommando-Telegramm

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

Tab. 5-22: Antwort-Telegramm

**Write 1 Bit 0x05**

Code 0x05: 1 Bit schreiben in Master-Ausgabe-Bereich 0x

Eine Zustandsänderung erfolgt unter **ZustandBit** mit folgenden Werten:

**Zustand Bit** = 0x0000 → Bit = 0

**Zustand Bit** = 0xFF00 → Bit = 1

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

Tab. 5-23: Kommando-Telegramm

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

Tab. 5-24: Antwort-Telegramm

### Write 1 Word 0x06

Code 0x06: 1 Wort schreiben in Master-Ausgabe-Bereich 4x

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

Tab. 5-25: Kommando-Telegramm

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

Tab. 5-26: Antwort-Telegramm

### Write n Bits 0x0F

Code 0x0F: n Bit schreiben in Master-Ausgabe-Bereich 0x



Die Anzahl der Bits muss zusätzlich in Byte angegeben werden.

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

Tab. 5-27: Kommandotelegramm

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

Tab. 5-28: Antworttelegramm

### Write n Words 0x10

Code 0x10: n Worte schreiben in Master-Ausgabe-Bereich

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

Tab. 5-29: Kommandotelegramm

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

Tab. 5-30: Antworttelegramm

## 5.8 Modbus - Fehlermeldungen

### Übersicht

Bei der Kommunikation unter Modbus gibt es 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 im Klartext ein.

### Fehlermeldungen

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

### Slave antwortet mit einer Fehlermeldung

Liefert der Slave einen Fehler zurück, wird der Funktionscode mit 0x80 **verodert** zurückgesendet.

#### Beispiel:

DB11.DBD 0	DW#16#05900000	<b>Antworttelegramm</b>
	mit 05 →	Slave-Adresse 0x05
	90 →	Funktionscode 0x90 (Fehlermeldung, da 0x10 OR 0x80 = 0x90)
	0000 →	Die Restdaten sind irrelevant, da Fehler rückgemeldet wurde

## 6 Allgemeine Daten

Konformität			
	CE	2004/108/EG	EMV Richtlinie
		2011/65/EU	RoHS
Personenschutz und Geräteschutz			
	Schutzart	EN 60529	IP20
	Potenzialltrennung		
	Zum Feldbus	-	Galvanisch entkoppelt
	Zur Prozessebene	-	Galvanisch entkoppelt
	Isolationsfestigkeit	EN 61131-2	-
	Isolationsspannung gegen Bezugserde		
	Eingänge / Ausgänge	-	50 V AC/DC, bei Prüfspannung 500 V AC
	Schutzmaßnahmen	-	gegen Kurzschluss
Umgebungsbedingungen			
	Klimatisch		
	Lagerung /Transport	EN 60068-2-14	-25 ... +70 °C
	Betrieb		
	Horizontaler Einbau	EN 61131-2	0 ... +60 °C
	Vertikaler Einbau	EN 61131-2	0 ... +60 °C
	Luftfeuchtigkeit	EN 60068-2-30	RH1 (ohne Betauung, relative Feuchte 10 ... 95 %)
	Verschmutzung	EN 61131-2	Verschmutzungsgrad 2
	Mechanisch		
	Schwingung	EN 60068-2-6	1 g, 9 Hz ... 150 Hz
Schock	EN 60068-2-27	15 g, 11 ms	
Montagebedingungen			
	Einbauort	-	Im Schaltschrank
	Einbaulage	-	Horizontal und vertikal
	Befestigung	-	Profilschiene 35 mm
Mechanische Daten		Gehäuse	
	Material	PPE / PPE GF10	
	Abmessungen (B x H x T)	12,9 x 109 x 76,5 mm	
	Gewicht	60 g	
Umgebungsbedingungen			
	Betriebstemperatur	0 °C bis 60 °C	
	Lagertemperatur	-25 °C bis 70 °C	
Zertifizierungen			
	Zertifizierung nach UL 508		ja

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

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



## 7 Technische Daten

Stromaufnahme / Verlustleistung		
	Stromaufnahme aus Rückwandbus	100 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	ja
	Schnittstelle potentialgetrennt	ja
	Schnittstelle RS232	ja
	Schnittstelle RS422	-
	Schnittstelle RS485	-
	Anschluss	Terminalmodul
	Übertragungsgeschwindigkeit, min.	150 bit/s
	Übertragungsgeschwindigkeit, max.	115,2 kbit/s
	Leitungslänge, max.	15 m
Point-to-Point Protokolle		
	Protokoll ASCII	ja
	Protokoll STX/ETX	ja
	Protokoll 3964(R)	ja
	Protokoll RK512	-
	Protokoll USS Master	-
	Protokoll Modbus Master	ja
	Protokoll Modbus Slave	ja
	Spezielle Protokolle	-
Datengrößen		
	Eingangsbytes	8 / 20 / 60
	Ausgangsbytes	8 / 20 / 60
	Parameterbytes	21
	Diagnosebytes	20

### 7.1 Protokolle

ASCII	
Telegrammlänge	max. 1024 Byte
Baudrate	150, 300, 600, 1 200, 1 800, 2 400, 4 800, 7 200, 9 600, 14 400, 19 200, 38 400, 57 600, 76 800, 109 700, 115 200 Baud
Zeichenverzugszeit (ZVZ)	0 ... 65 535 (in ms-Schritten), bei ZVZ = 0 wird 3-fache Zeichenzeit verwendet.
Flusskontrolle	keine Hardware, XON/XOFF
Anzahl pufferbarer Telegramme	max. 250
Enderkennung eines Telegramms	nach Ablauf der Zeichenverzugszeit (ZVZ)

<b>STX/ETX</b>	
Telegrammlänge	max. 1 024 Byte
Baudrate	150, 300, 600, 1 200, 1 800, 2.400, 4 800, 7 200, 9 600, 14 400, 19 200, 38 400, 57 600, 76 800, 109 700, 115 200 Baud
Zeichenverzugszeit (TMO)	0 ... 65 535 (in ms-Schritten), bei TMO = 0 wird 3-fache Zeichenzeit verwendet.
Flusskontrolle	keine Hardware, XON/XOFF
Anzahl pufferbarer Telegramme	max. 250
Enderkennung eines Telegramms	durch parametrisiertes Endezeichen
Anzahl Startzeichen	0 ... 2 (Zeichen parametrierbar)
Anzahl Endzeichen	0 ... 2 (Zeichen parametrierbar)
<b>3964, 3964R</b>	
Telegrammlänge	max. 1 024 Byte
Baudrate	150, 300, 600, 1 200, 1 800, 2.400, 4 800, 7 200, 9 600, 14 400, 19 200, 38 400, 57 600, 76 800, 109 700, 115 200 Baud
Blockprüfzeichen	nur 3964R
Priorität	low/high
Zeichenverzugszeit (ZVZ)	0 ... 255 (in 20 ms-Schritten), bei ZVZ = 0 wird 3-fache Zeichenzeit verwendet.
Quittungsverzugszeit (QVZ)	0 ... 255 (in 20 ms-Schritten), bei QVZ = 0 wird 3-fache Zeichenzeit verwendet.
Anzahl Aufbauversuche	0 ... 255
Anzahl Übertragungsversuche	1 ... 255
<b>Modbus</b>	
Telegrammlänge	max. 258 Byte
Baudrate	150, 300, 600, 1 200, 1 800, 2.400, 4 800, 7 200, 9 600, 14 400, 19 200, 38 400, 57 600, 76 800, 109 700, 115 200 Baud
Modus	Master ASCII, Master RTU, Slave ASCII short, Slave RTU short, Slave ASCII long, Slave, RTU long
Adresse	1 ... 255
Wartezeit	automatisch, 1 ... 60 000 ms

## 8 Anhang

### 8.1 Zubehör

**Busblende**  
**Art.-No. 57190**

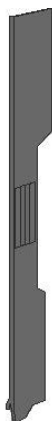


Abb. 8-1: Busblende

**Träger für Schirmschienen**  
**Art.-No. 57191**

Der Träger nimmt Schirmschienen (10mm x 3mm) zum Anschluss von Kabelschirmen auf.



#### HINWEIS

Träger für Schirmschienen, Schirmschienen und Kabelschirmbefestigungen sind nicht im Lieferumfang enthalten.

#### Montage Träger

- ✓ Voraussetzung: Das CUBE20S-System ist vollständig montiert.
- ➔ Brechen Sie bei flacher Profilschiene den Abstandshalter am Träger ab.
- ➔ Stecken Sie den Träger unterhalb des Klemmenblocks in das Terminal-Modul, bis er einrastet.

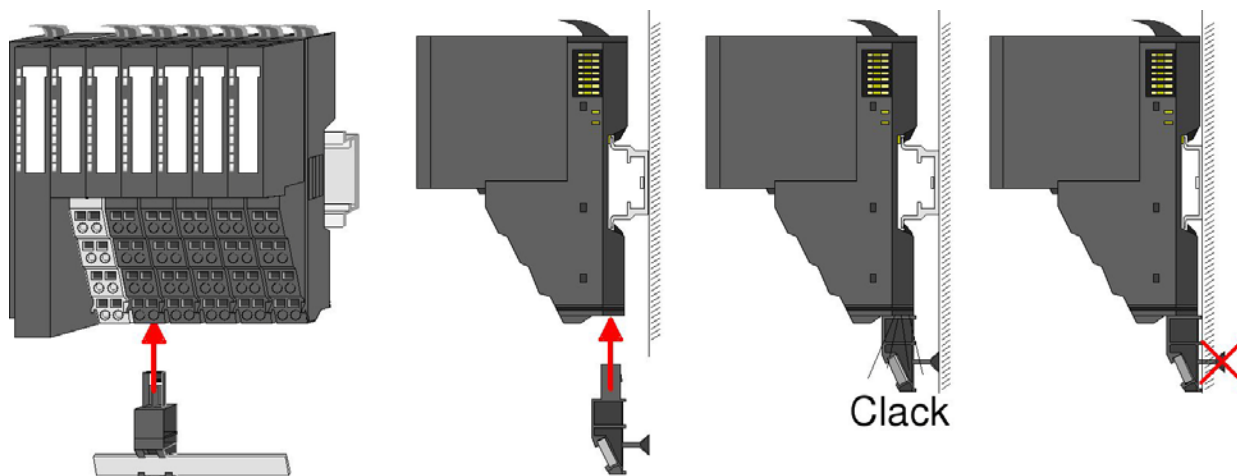


Abb. 8-2: Montage der Träger für Schirmschienen

## 8.2 Glossar

### Allgemeine Einträge:

Begriff	Bedeutung
Bestimmungsgemäßer Gebrauch	Anwendung eines Produkts, eines Verfahrens oder einer Leistung nach den durch den Hersteller gelieferten Spezifikationen, Anweisungen und Angaben.
Bit	Binärziffer
Byte	1 Byte entspricht 8 Bit
DI	Digital Input, Digitale Eingänge
DIN	Deutsches Institut für Normung
E/A	Ein-/Ausgang
Richtlinie 2004/108/EG	EMV-Richtlinie
EMV	Elektromagnetische Verträglichkeit
EN	Europäische Norm
ESD	Elektrostatische Entladungen
FE	Funktionserde
I	Strom
IEC	International Electrotechnical Commission, Internationale Normungsinstitut
IN	Eingang
IP20	Ingress Protection (Eindringenschutz), Schutzart nach DIN EN 60529 1. Kennziffer = Berührungs- und Fremdkörperschutz 2. Kennziffer = Wasserschutz 2: Geschützt gegen: feste Fremdkörper mit Durchmesser ab 12,5 mm und Zugang mit einem Finger 0: kein Schutz
IP67	6: Staubdicht, Schutz gegen den Zugang mit einem Draht 7: Schutz gegen zeitweiliges Untertauchen
ISO	International Standard Organization
LED	Light Emitting Diode
n. c.	nicht belegt (not connected)
OUT	Ausgang
PELV	Protective Extra Low Voltage / Schutzkleinspannung
SELV	Safety Extra Low Voltage / Sicherheitskleinspannung
U	Spannung
U/I	Spannung/Strom

## 8.3 Rechtliche Hinweise

### Haftungsausschluss

Die Fa. Murrelektronik GmbH hat den Inhalt dieser technischen Dokumentation auf Übereinstimmung mit der beschriebenen Hard- bzw. Software überprüft. Abweichungen können im Einzelfall nicht ausgeschlossen werden, weshalb wir die Gewährleistung für die inhaltliche Richtigkeit und die Haftung für Fehler, insbesondere für die vollständige Übereinstimmung ausschließen. Die Beschränkung der Haftung gilt nicht, soweit die Schadensursache auf Vorsatz und/oder grobe Fahrlässigkeit zurückzuführen ist, sowie für sämtliche Ansprüche aus dem Produkthaftungsgesetz. Sofern leicht fahrlässig eine vertragswesentliche Pflicht verletzt wurde, ist die Haftung der Fa. Murrelektronik GmbH auf den typischerweise entstehenden Schaden begrenzt.

Technische und inhaltliche Änderungen bleiben vorbehalten. Wir empfehlen, in regelmäßigen zeitlichen Abständen zu überprüfen, ob diese Dokumentation aktualisiert wurde, da Korrekturen, die beispielsweise durch technische Fortentwicklungen erforderlich werden können, regelmäßig von der Fa. Murrelektronik GmbH eingepflegt werden. Für Verbesserungsvorschläge sind wir jederzeit dankbar.

### Urheberrecht

Die Weitergabe sowie Vervielfältigung der Dokumentation auf Papier oder in digitaler Weise, die Verwertung und Mitteilung ihres Inhaltes ist nicht gestattet, soweit dies nicht durch die Fa. Murrelektronik GmbH ausdrücklich zugestanden wurde oder in Verbindung mit der Erstellung eigener Dokumentationen von Produkten geschieht, die ihrerseits Produkte der Fa. Murrelektronik GmbH enthalten. Zuwiderhandlungen verpflichten zu Schadenersatz. Alle Rechte bleiben vorbehalten, insbesondere für den Fall der Patenterteilung oder GM-Eintragung.

### Nutzungsrechte

Die Fa. Murrelektronik GmbH räumt ihren Kunden an dieser technischen Dokumentation ein jederzeit widerrufliches, nicht ausschließliches und zeitlich unbegrenztes Recht ein, diese zur Erstellung eigener technischer Dokumentationen zu verwenden. Dazu kann die Dokumentation der Fa. Murrelektronik GmbH auszugsweise verändert oder ergänzt sowie vervielfältigt und als Teil der eigenen technischen Dokumentation des Kunden auf Papier oder Datenträgern den Abnehmern des Kunden überlassen werden. Dabei übernimmt der Kunde jedoch die alleinige Verantwortung für die Richtigkeit der Inhalte der von ihm erstellten technischen Dokumentationen.

Wird die technische Dokumentation ganz oder auszugsweise in die technische Dokumentation des Kunden übernommen, muss der Kunde auf die Urheberschaft der Fa. Murrelektronik GmbH hinweisen. Es ist außerdem unbedingt darauf zu achten, dass die sicherheitstechnischen Hinweise erhalten bleiben.

Wenngleich der Kunde auch verpflichtet ist, die Urheberschaft der Fa. Murrelektronik GmbH anzugeben, soweit die technischen Dokumentationen der Fa. Murrelektronik GmbH verwendet werden, so vertreibt bzw. benutzt der Kunde die technischen Dokumentationen in alleiniger Verantwortung. Grund ist, dass wir keinen Einfluss auf die Änderungen bzw. Verwendungen der technischen Dokumentationen hat und bereits geringfügige Veränderungen des Ausgangsproduktes bzw. Abweichungen von den vorgesehenen Verwendungen die in den technischen Dokumentationen festgehaltenen Spezifikationen unrichtig machen können. Aus diesem Grunde ist der Kunde auch verpflichtet, die von der Fa. Murrelektronik GmbH stammenden technischen Dokumentationen zu kennzeichnen, wenn und soweit die Dokumentationen vom Kunden

geändert wurden. Der Kunde verpflichtet sich, die Fa. Murrelektronik von Schadenersatzansprüchen Dritter freizustellen, soweit diese auf ggf. bestehende Mängel der Dokumentation zurückzuführen sind. Dies gilt nicht für vorsätzlich oder grob fahrlässig verursachte Schäden an den Rechten Dritter.

Der Kunde ist zur Nutzung der Firmenmarken der Fa. Murrelektronik GmbH ausschließlich im Rahmen seiner Produktwerbung berechtigt und auch nur soweit, wie Produkte der Fa. Murrelektronik GmbH in die beworbenen Produkte des Kunden integriert wurden. Der Kunde wird bei der Verwendung von Marken der Fa. Murrelektronik GmbH in geeigneter Weise darauf hinweisen, dass es sich um Marken der Fa. Murrelektronik GmbH handelt.

