

# SIEMENS

## SIMATIC

### ET 200S Serielle Schnittstellenbaugruppen

Betriebsanleitung

Vorwort

1

Serielle  
Schnittstellenbaugruppe

2


Modbus/USS


3


## Rechtliche Hinweise

### Warnhinweiskonzept

Dieses Handbuch enthält Hinweise, die Sie zu Ihrer persönlichen Sicherheit sowie zur Vermeidung von Sachschäden beachten müssen. Die Hinweise zu Ihrer persönlichen Sicherheit sind durch ein Warndreieck hervorgehoben, Hinweise zu alleinigen Sachschäden stehen ohne Warndreieck. Je nach Gefährdungsstufe werden die Warnhinweise in abnehmender Reihenfolge wie folgt dargestellt.

 <b>GEFAHR</b>
bedeutet, dass Tod oder schwere Körperverletzung eintreten <b>wird</b> , wenn die entsprechenden Vorsichtsmaßnahmen nicht getroffen werden.

 <b>WARNUNG</b>
bedeutet, dass Tod oder schwere Körperverletzung eintreten <b>kann</b> , wenn die entsprechenden Vorsichtsmaßnahmen nicht getroffen werden.

 <b>VORSICHT</b>
mit Warndreieck bedeutet, dass eine leichte Körperverletzung eintreten kann, wenn die entsprechenden Vorsichtsmaßnahmen nicht getroffen werden.

<b>VORSICHT</b>
ohne Warndreieck bedeutet, dass Sachschaden eintreten kann, wenn die entsprechenden Vorsichtsmaßnahmen nicht getroffen werden.

<b>ACHTUNG</b>
bedeutet, dass ein unerwünschtes Ergebnis oder Zustand eintreten kann, wenn der entsprechende Hinweis nicht beachtet wird.


Beim Auftreten mehrerer Gefährdungsstufen wird immer der Warnhinweis zur jeweils höchsten Stufe verwendet. Wenn in einem Warnhinweis mit dem Warndreieck vor Personenschäden gewarnt wird, dann kann im selben Warnhinweis zusätzlich eine Warnung vor Sachschäden angefügt sein.

### Qualifiziertes Personal

Das zugehörige Gerät/System darf nur in Verbindung mit dieser Dokumentation eingerichtet und betrieben werden. Inbetriebsetzung und Betrieb eines Gerätes/Systems dürfen nur von **qualifiziertem Personal** vorgenommen werden. Qualifiziertes Personal im Sinne der sicherheitstechnischen Hinweise dieser Dokumentation sind Personen, die die Berechtigung haben, Geräte, Systeme und Stromkreise gemäß den Standards der Sicherheitstechnik in Betrieb zu nehmen, zu erden und zu kennzeichnen.

### Bestimmungsgemäßer Gebrauch von Siemens-Produkten

Beachten Sie Folgendes:

 <b>WARNUNG</b>
Siemens-Produkte dürfen nur für die im Katalog und in der zugehörigen technischen Dokumentation vorgesehenen Einsatzfälle verwendet werden. Falls Fremdprodukte und -komponenten zum Einsatz kommen, müssen diese von Siemens empfohlen bzw. zugelassen sein. Der einwandfreie und sichere Betrieb der Produkte setzt sachgemäßen Transport, sachgemäße Lagerung, Aufstellung, Montage, Installation, Inbetriebnahme, Bedienung und Instandhaltung voraus. Die zulässigen Umgebungsbedingungen müssen eingehalten werden. Hinweise in den zugehörigen Dokumentationen müssen beachtet werden.

### Marken

Alle mit dem Schutzrechtsvermerk ® gekennzeichneten Bezeichnungen sind eingetragene Marken der Siemens AG. Die übrigen Bezeichnungen in dieser Schrift können Marken sein, deren Benutzung durch Dritte für deren Zwecke die Rechte der Inhaber verletzen kann.

### Haftungsausschluss

Wir haben den Inhalt der Druckschrift auf Übereinstimmung mit der beschriebenen Hard- und Software geprüft. Dennoch können Abweichungen nicht ausgeschlossen werden, so dass wir für die vollständige Übereinstimmung keine Gewähr übernehmen. Die Angaben in dieser Druckschrift werden regelmäßig überprüft, notwendige Korrekturen sind in den nachfolgenden Auflagen enthalten.

# Inhaltsverzeichnis

<b>1</b>	<b>Vorwort</b> .....	<b>7</b>
<b>2</b>	<b>Serielle Schnittstellenbaugruppe</b> .....	<b>9</b>
2.1	Produktübersicht .....	9
2.2	Kurzanleitung zur Inbetriebnahme der seriellen Schnittstellenbaugruppe .....	12
2.3	Schaltbild mit Anschlussbelegung .....	16
2.4	RS-232C-Schnittstelle .....	22
2.5	RS-422/485-Schnittstelle .....	23
2.6	Grundlagen der seriellen Datenübertragung .....	24
2.6.1	Serielle Datenübertragung .....	24
2.6.2	Zeichenrahmen .....	26
2.6.3	Übertragungsverfahren bei der Punkt-zu-Punkt-Kopplung .....	29
2.6.4	Übertragungssicherheit .....	31
2.7	Datenübertragung mit der Prozedur 3964(R) .....	33
2.7.1	Grundlagen der Datenübertragung mit der Prozedur 3964(R) .....	33
2.7.2	Daten senden mit der Prozedur 3964(R) .....	35
2.7.3	Daten empfangen mit der Prozedur 3964(R) .....	37
2.7.4	Fehlerbehandlung bei der Prozedur 3964(R) .....	39
2.8	Datenübertragung mit dem ASCII-Treiber .....	42
2.8.1	Grundlagen zur Datenübertragung mit ASCII-Treiber .....	42
2.8.2	Daten senden mit dem ASCII-Treiber .....	43
2.8.3	Daten empfangen mit dem ASCII-Treiber .....	44
2.8.4	Endekriterien für die Datenübertragung mit dem ASCII-Treiber .....	46
2.8.5	RS-232C-Begleitsignale für die Datenübertragung mit dem ASCII-Treiber .....	49
2.9	Konfigurieren und Parametrieren der seriellen Schnittstellenbaugruppe .....	53
2.9.1	Konfigurieren der seriellen Schnittstellenbaugruppe .....	53
2.9.2	Parametrieren des ASCII-Treibers .....	53
2.9.3	Parametrieren der Treiber für das Protokoll 3964(R) .....	57
2.9.4	Identifikationsdaten .....	61
2.9.5	Nachladen von Firmware-Updates .....	63
2.10	Kommunikation über Funktionsbausteine .....	65
2.10.1	Grundlagen zur Kommunikation über Funktionsbausteine .....	65
2.10.2	Funktionsbaustein FB3 S_SEND .....	67
2.10.3	Funktionsbaustein FB2 S_RCV .....	71
2.10.4	Funktionen für die Parametrierung von Optionen für die Datenflusskontrolle .....	75
2.10.5	Lesen und Steuern von RS-232C-Begleitsignalen .....	81
2.11	Anlaufeigenschaften und Betriebszustände .....	85
2.12	Referenzdaten für andere Master als S7-PROFIBUS .....	87
2.12.1	Grundlagen zu Referenzdaten .....	87
2.12.2	Beispielablauf beim Senden von Daten von der CPU zur Baugruppe .....	91
2.12.3	Beispielablauf beim Empfangen von Daten der Baugruppe in der CPU .....	93
2.12.4	Beispielablauf beim Lesen des V.24-Signalstatus .....	94
2.12.5	Beispielablauf beim Schreiben von V.24-Signalen .....	95

2.12.6	Parameter für die Datenflusskontrolle.....	96
2.12.7	Fehlerbehandlung .....	99
2.13	Diagnose .....	101
2.14	Technische Daten .....	108
<b>3</b>	<b>Modbus/USS .....</b>	<b>111</b>
3.1	Produktübersicht .....	111
3.2	Kurzanleitung zur Inbetriebnahme der seriellen Schnittstellenbaugruppe .....	114
3.3	Schaltbilder mit Anschlussbelegung .....	120
3.3.1	Anschlussbelegung .....	120
3.3.2	RS-232C-Schnittstelle.....	126
3.3.3	RS-422/485-Schnittstelle .....	128
3.4	Modbus-Übertragungsprotokoll.....	129
3.4.1	Eigenschaften und Telegrammaufbau .....	129
3.4.2	Slave-Adresse .....	130
3.4.3	Master- und Slave-Funktionscodes .....	130
3.4.4	Datenfeld DATA .....	131
3.4.5	Telegrammende und CRC-Prüfung .....	131
3.4.6	Ausnahmeantworten .....	132
3.5	Modbus-Master-Treiber.....	133
3.5.1	Einsatz des Modbus-Master-Treibers .....	133
3.5.2	Datenübertragung beim Modbus-Master ET 200S .....	133
3.5.3	Konfigurieren und Parametrieren des Modbus-Masters .....	143
3.5.4	Vom Modbus-Master verwendete Funktionscodes.....	147
3.5.5	Funktionscode 01 – Read Output Status .....	148
3.5.6	Funktionscode 02 – Read Input Status .....	149
3.5.7	Funktionscode 03 – Read Output Registers .....	150
3.5.8	Funktionscode 04 – Read Input Registers .....	151
3.5.9	Funktionscode 05 – Force Single Coil .....	152
3.5.10	Funktionscode 06 – Preset Single Register .....	153
3.5.11	Funktionscode 07 – Read Exception Status .....	154
3.5.12	Funktionscode 08 – Loop Back Diagnostic Test.....	155
3.5.13	Funktionscode 11 – Fetch Communications Event Counter.....	156
3.5.14	Funktionscode 12 – Fetch Communications Event Log.....	157
3.5.15	Funktionscode 15 – Force Multiple Coils .....	158
3.5.16	Funktionscode 16 – Preset Multiple Registers.....	159
3.6	Modbus-Slave-Treiber.....	160
3.6.1	Bestandteile der Modbus-Slave-Kopplung.....	160
3.6.2	Datenübertragung beim Modbus-Slave ET 200S .....	161
3.6.3	Datenbereiche in der SIMATIC CPU .....	163
3.6.4	Konfigurieren der Parameter für die Kopplung .....	164
3.6.5	Slave-Funktionscodes.....	169
3.6.6	Funktionscode 01 – Read Coil (Output) Status.....	170
3.6.7	Funktionscode 02 – Read Input Status .....	173
3.6.8	Funktionscode 03 – Read Output Registers .....	176
3.6.9	Funktionscode 04 – Read Input Registers .....	179
3.6.10	Funktionscode 05 – Force Single Coil .....	182
3.6.11	Funktionscode 06 – Preset Single Register .....	185
3.6.12	Funktionscode 08 – Loop Back Diagnostic Test.....	188
3.6.13	Funktionscode 15 – Force Multiple Coils .....	189
3.6.14	Funktionscode 16 – Preset Multiple Registers.....	192
3.6.15	Bitorientierte Funktionscode-Umsetzung .....	195

3.6.16	Registerorientierte Funktionscode-Umsetzung.....	195
3.6.17	Freigeben/Sperren von Schreibzugriffen .....	197
3.6.18	Umsetzung der Modbus-Adressen für Bitfunktionen .....	198
3.6.19	Umsetzung der Modbus-Adressen für Registerfunktionen.....	203
3.6.20	Grenzen für Schreibfunktionen .....	205
3.7	Diagnose .....	207
3.7.1	Möglichkeiten der Diagnose.....	207
3.7.2	Diagnoseinformationen der Status-LEDs .....	207
3.7.3	Diagnosemeldungen der Funktionsbausteine .....	208
3.7.4	PROFIBUS-Slave-Diagnose .....	216
3.7.5	Diagnosefunktionen Modbus-Slave .....	217
3.7.6	Fehler .....	218
3.8	USS-Master.....	220
3.8.1	Was ist der USS-Master? .....	220
3.8.2	USS-Protokoll.....	221
3.8.3	Konfiguration und Parametrierung .....	223
3.8.4	Funktionsübersicht.....	224
3.8.5	FC17 S_USST: Daten an einen Slave senden.....	226
3.8.6	FC18 S_USSR: Daten von einem Slave empfangen.....	229
3.8.7	FC19 S_USSI: Initialisierung.....	232
3.8.8	Netzdaten-DB.....	235
3.8.9	Parametrierungs-DB .....	240
3.8.10	Kommunikationsprozessor-DB .....	242
3.9	Anlaufeigenschaften und Betriebszustände der seriellen Schnittstellenbaugruppe ET 200S Modbus/USS.....	244
3.9.1	Laden der Konfigurations- und Parametrierungsdaten.....	244
3.9.2	Betriebszustände der seriellen Schnittstellenbaugruppe ET 200S Modbus/USS .....	245
3.9.3	Anlaufeigenschaften der seriellen Schnittstellenbaugruppe ET 200S Modbus/USS .....	245
3.9.4	Verhalten der seriellen Schnittstellenbaugruppe ET 200S Modbus/USS bei Betriebszustandsübergängen der CPU .....	246
3.10	Technische Daten .....	248
<b>Index.....</b>		<b>253</b>



# Vorwort

## Wie das Handbuch strukturiert ist...

Dieses Handbuch ist ein ergänzender Band zum Handbuch *Dezentrales Peripheriesystem ET 200S*.

Im Handbuch *Dezentrales Peripheriesystem ET 200S* finden Sie ausführliche Informationen zu Hardware-Konfiguration, Aufbau, Verdrahtung, Inbetriebnahme, Diagnose und den technischen Daten des dezentralen Peripheriesystems ET 200S

Im vorliegenden Handbuch finden Sie die Funktionsbeschreibungen und die technischen Daten der seriellen Schnittstellenbaugruppen ET 200S 1SI und ET 200S Modbus/US\$.

## Wie Sie sich darin zurechtfinden...

Am Anfang jedes Kapitels finden Sie eine **Produktübersicht**, in der die Eigenschaften des beschriebenen Moduls und damit auch die Einsatzmöglichkeiten aufgelistet sind. Dort finden Sie auch eine Referenz auf die Bestellnummer des beschriebenen Moduls und auf Namen und Ausgabestand der Software. Die aktuelle GSD-Datei finden Sie im Internet unter:

<http://support.automation.siemens.com>

Daran anschließend finden Sie in jedem Kapitel eine **Kurzanleitung zur Inbetriebnahme**. Innerhalb dieser Kurzanleitung erfahren Sie in kurzen Arbeitsschritten, wie Sie das jeweilige Modul montieren, projektieren, in Ihr Anwenderprogramm einbinden und testen.

## Normen und Zulassungen

Angaben zu den Normen und Zulassungen finden Sie im Handbuch *Dezentrales Peripheriesystem ET 200S* im Kapitel "Allgemeine Technische Daten". Dieses Handbuch finden Sie hier:

<http://www.siemens.de/simatic-tech-doku-portal>

## Recycling und Entsorgung

Die seriellen Schnittstellenbaugruppen ET 200S 1SI 3964/ASCII und ET 200S 1SI Modbus/US\$ sind wegen ihrer schadstoffarmen Ausrüstung recyclingfähig. Für ein umweltverträgliches Recycling und die Entsorgung Ihres Altgeräts wenden Sie sich an einen zertifizierten Entsorgungsbetrieb für Elektronikschrott.

## Index

Der Index enthält Stichworte zu allen Kapiteln.

## Technical Support

Sie erreichen den Technical Support für alle Industry Automation Produkte über das Web-Formular für den Support Request.

<http://www.siemens.de/automation/support-request>

Weitere Informationen zu unserem Technical Support finden Sie im Internet unter

<http://www.siemens.de/automation/service>

## Service & Support im Internet

Zusätzlich zu unserem Dokumentations-Angebot bieten wir Ihnen im Internet unser Know-how an.

<http://www.siemens.com/automation/service&support>

Dort finden Sie:

- den Newsletter, der Sie ständig mit den aktuellsten Informationen zu Ihren Produkten versorgt.
- die für Sie richtigen Dokumente über unsere Suche im Produkt Support.
- ein Forum, in welchem Anwender und Spezialisten weltweit Erfahrungen austauschen.
- Ihren Ansprechpartner für Industry vor Ort.
- Informationen über Reparaturen, Ersatzteile und Consulting.

## Weitere Unterstützung

Bei Fragen zur Nutzung der im Handbuch beschriebenen Produkte, die Sie hier nicht beantwortet finden, wenden Sie sich bitte an Ihren Siemens-Ansprechpartner in den für Sie zuständigen Vertretungen und Geschäftsstellen.

Ihren Ansprechpartner finden Sie unter:

<http://www.siemens.com/automation/partner>

Den Wegweiser zum Angebot an technischen Dokumentationen für die einzelnen SIMATIC Produkte und Systeme finden Sie unter:

<http://www.siemens.de/simatic-tech-doku-portal>

Den Online-Katalog und das Online-Bestellsystem finden Sie unter:

<http://mall.automation.siemens.com>

## Trainingscenter

Um Ihnen den Einstieg in das Automatisierungssystem SIMATIC S7 zu erleichtern, bieten wir entsprechende Kurse an. Wenden Sie sich bitte an Ihr regionales Trainingscenter oder an das zentrale Trainingscenter in D 90327 Nürnberg.

<http://www.sitrain.com>



# Serielle Schnittstellenbaugruppe

## 2.1 Produktübersicht

### Bestellnummer

6ES7 138-4DF01-0AB0

### Produktbeschreibung

Die serielle Schnittstellenbaugruppe ET 200S 1SI ist eine steckbare Baugruppe der Produktreihe ET 200S. Sie bietet Zugang zur seriellen Kommunikation über drei Hardwareschnittstellen (RS-232C, RS-422 und RS-485) und zwei Softwareprotokolle (ASCII und 3964(R)).

Mit der Schnittstellenbaugruppe ET 200S 1SI können Sie über eine Punkt-zu-Punkt-Kopplung Daten zwischen Automatisierungssystemen oder Computern austauschen. Sämtliche Kommunikation läuft über serielle asynchrone Übertragungen ab.

Sie wählen die Kommunikationsart, wenn Sie die Baugruppe in der Hardware-Konfiguration von STEP 7 bzw. in einer anderen Konfigurationsanwendung parametrieren. Die Baugruppe wird im Hardware-Katalog in folgenden sechs Ausführungen angezeigt:

- ASCII (4B)
- ASCII (8B)
- ASCII (32B)
- 3964R (4B)
- 3964R (8B)
- 3964R (32B)

8 bzw. 32-Byte-Datenübertragungen erhöhen den Durchsatz, doch benötigen mehr E/A-Speicher auf dem ET 200S Baugruppenträger, während 4-Byte-Datenübertragungen weniger E/A-Speicher auf dem ET 200S Baugruppenträger benötigen, jedoch einen geringeren Durchsatz bieten. Die Baugruppenvariante richtet sich nach Ihren Anwendungsanforderungen.

### Funktionalität der seriellen Schnittstellenbaugruppe ET 200S 1SI

Die serielle Schnittstellenbaugruppe ET 200S 1SI bietet folgende Funktionen:

- Integrierte Schnittstelle nach RS-232C, RS-422 oder RS-485
- Übertragungsgeschwindigkeit bis 115,2 Kbaud, halbduplex
- Integration der folgenden Übertragungsprotokolle in die Firmware der Baugruppe:
  - Prozedur 3964(R)
  - ASCII-Treiber

Die Parametrierung der Baugruppe bestimmt die Funktionalität der Treiber.

Die folgende Tabelle führt die Funktionen der einzelnen Treiberschnittstellen auf.

Tabelle 2- 1 Funktionen der Baugruppentreiber der seriellen Schnittstellenbaugruppe ET 200S 1SI

Funktion	RS-232C	RS-422	RS-485
<b>ASCII-Treiber</b>	<b>Ja</b>	<b>Ja</b>	<b>Ja</b>
Verwendung von RS-232C-Begleitsignalen	Ja	Nein	Nein
Steuern/Lesen von RS-232C-Begleitsignalen mit FBs	Ja	Nein	Nein
Flussskontrolle mit RTS/CTS	Ja	Nein	Nein
Flussskontrolle mit XON/XOFF	Ja	Ja	Nein
<b>Prozedur 3964(R)</b>	<b>Ja</b>	<b>Ja</b>	<b>Nein</b>

## Kommunikation

Die serielle Schnittstellenbaugruppe ET 200S 1SI ermöglicht eine Punkt-zu-Punkt-Kopplung mit unterschiedlichen Siemens-Baugruppen und Fremdprodukten, einschließlich:

- SIMATIC S5 über den 3964(R)-Treiber mit entsprechender Schnittstellenbaugruppe auf S5-Seite
- Siemens BDE-Terminals ES 2-Familie über 3964(R)-Treiber
- MOBY I (ASM 420/421, SIM), MOBY L (ASM 520) und Erfassungsstation ES 030K über 3964(R)-Treiber
- SIMOVERT und SIMOREG (USS-Protokoll) über den ASCII-Treiber (ET 200S SI RS 422/485) mit entsprechender Protokollanpassung mit einem STEP 7-Programm
- PCs über Prozedur 3964(R) (hierfür existieren Entwicklungstools zur Programmierung auf PC: PRODAVE DOS 64R (6ES5 897-2UD11) für MS-DOS, PRODAVE WIN 64R (6ES5 897-2VD01) für Windows oder ASCII-Treiber)
- Barcodeleser über 3964(R)-Treiber oder ASCII-Treiber
- SPS von anderen Herstellern über den 3964(R)-Treiber oder ASCII-Treiber
- Weitere Geräte mit einfachen Protokollstrukturen über entsprechende Protokollanpassung mit dem ASCII-Treiber
- Weitere Geräte, die ebenfalls über 3964(R)-Treiber verfügen

---

### Hinweis

Hinter den externen Kommunikations-CPs CP 342-5 (Profibus DP) und CP 343-1 (Profinet IO) ist das Modul ET 200S 1SI mit den normalen Standard-FBs nicht betreibbar!

Für den Betrieb des Moduls hinter den Kommunikations-CPs CP 342-5 (Profibus DP) oder CP 343-1 (Profinet IO) stehen auf den Internet-Seiten des Customer Supports entsprechende Spezial-FBs zur Verfügung:

Siehe <http://support.automation.siemens.com/WWW/view/de/26263724>

---

## LED-Anzeigen

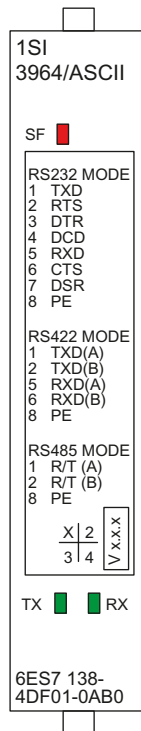
Folgende Status-LEDs befinden sich auf der Frontplatte der Schnittstellenbaugruppe:

LED	Farbe	Beschreibung
SF	rot	Sammelfehleranzeige
TX	grün	Die Schnittstelle sendet.
RX	grün	Die Schnittstelle empfängt.

Die Betriebszustände und Fehler, die von diesen LEDs angezeigt werden, sind im Abschnitt Diagnose (Seite 101) beschrieben.

## Frontplatte

Das folgende Bild zeigt die Beschriftung der Frontplatte der seriellen Schnittstellenbaugruppe ET 200S 1SI.



## 2.2 Kurzanleitung zur Inbetriebnahme der seriellen Schnittstellenbaugruppe

### Einleitung

In dieser Kurzanleitung wird anhand eines Beispiels zum Senden und Empfangen von Daten zwischen seriellen Schnittstellenbaugruppen erläutert, wie Sie eine funktionierende Anwendung einrichten, wie die grundlegenden Operationen der seriellen Schnittstellenbaugruppe (Hardware und Software) funktionieren und wie Sie Hard- und Software prüfen.

In diesem Beispiel betreiben wir zwei serielle Schnittstellenbaugruppen ET 200S 1SI 3964(R)/ASCII im RS-232C ASCII-Modus.

### Voraussetzungen

Folgende Voraussetzungen müssen erfüllt sein:

- Sie nehmen eine ET 200S-Station an einer S7-Station mit DP-Master in Betrieb.
- Sie benötigen folgende Komponenten:
  - Zwei Terminalmodule TM-E15S24-01
  - Zwei serielle Schnittstellenbaugruppen ET 200S 1SI 3964(R)/ASCII
  - Das erforderliche Verdrahtungsmaterial

### Montieren, Verdrahten und Bestücken

Montieren und verdrahten Sie die zwei Terminalmodule TM-E15S24-01 (siehe folgendes Bild). Verbinden Sie die zwei seriellen Schnittstellenbaugruppen ET 200S 1SI 3964(R)/ASCII mit den Terminalmodulen. (Eine ausführliche Anleitung hierzu finden Sie im Handbuch *Dezentrales Peripheriegerät ET 200S*).

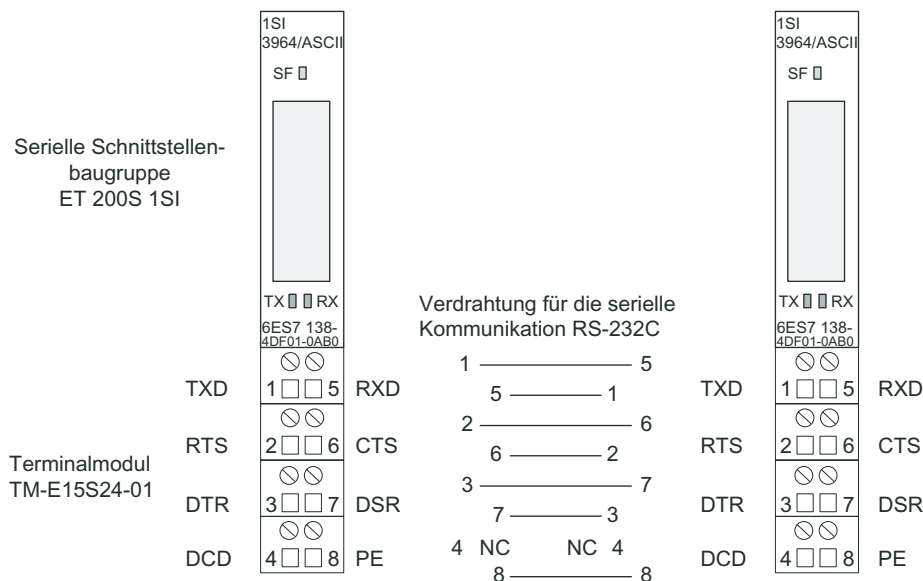


Bild 2-1 Anschlussbelegung für das Beispiel

## Verwendete Konfiguration

In der folgenden Tabelle finden Sie die für das Beispielprogramm verwendete Konfiguration.

Tabelle 2- 2 Parametrierung für die Beispielanwendung

Parameter	Wert
Sammeldiagnose	sperren
Schnittstelle	RS232-C
Vorbelegte Empfangsleitung	Bei RS232 nicht relevant
Datenflusskontrolle (Vorbelegung)	keine
Baudrate	9600
Datenbits	8
Stoppbits	1
Parität	gerade
Empfang Endekennung	Ablauf der Zeichenverzugszeit
Zeichenverzugszeit (ms)	4
Endezeichen 1	Bei RS232 nicht relevant
Endezeichen 2	Bei RS232 nicht relevant
Empfang Zeichenanzahl	Bei RS232 nicht relevant
Dynamischer Telegrammpuffer	ja
Telegrammpuffer nicht überschreiben	ja
Empfangspuffer bei Anlauf löschen	ja

## Verwendete Bausteine

In der folgenden Tabelle finden Sie die für das Beispielprogramm verwendeten Bausteine.

Baustein	Symbol	Kommentar
OB 1	CYCLE	Zyklische Programmbearbeitung
OB 100	RESTART	Anlaufbearbeitung Neustart
DB 21	SEND_IDB_SI_0	Instanz-DB für FB S_SEND_SI
DB 22	RECV_IDB_SI_1	Instanz-DB für FB S_RECV_SI
DB 40	SEND_WORK_DB_SI_0	Arbeits-DB für den Standard-FB 3
DB 41	RECV_WORK_DB_SI_1	Arbeits-DB für den Standard-FB 2
DB 42	SEND_SRC_DB_SI_0	Sende-Datenbaustein
DB 43	RECV_DST_DB_SI_1	Empfangs-Datenbaustein
FB 2	S_RECV_SI	Standard-FB für Daten empfangen
FB 3	S_SEND_SI	Standard-FB für Daten senden
FC 21	SEND_SI_0	Daten senden
FC 22	RECV_SI_1	Daten empfangen

## Lieferform und Installation

Das Programmbeispiel der ET 200S 1SI liegt zusammen mit den Funktionsbausteinen im Internet unter

<http://support.automation.siemens.com/WW/view/de/10805265/133100>

Nach der Installation befindet sich das Programmbeispiel im Projekt zXX21\_10\_1SI\_ASCII.

Das Projekt öffnen Sie im SIMATIC-Manager von STEP 7 mit dem Menübefehl "Datei > Öffnen > Beispielprojekte".

Das Programmbeispiel liegt in kompilierter Form und als ASCII-Quelldatei vor. Ebenfalls dabei ist eine Symbolliste der im Beispiel verwendeten Symbole.

Wenn Ihnen keine zweite ET 200S 1SI als Kommunikationspartner zur Verfügung steht, müssen Sie in HW-Konfig die zweite ET 200S 1SI mit "Bearbeiten > Löschen" entfernen. Zusätzlich muss im OB 1 der Aufruf des FC 22 (FC für Receive) auskommentiert werden.

## Laden in die CPU

Die Hardware für das Beispiel ist komplett aufgebaut, das Programmiergerät angeschlossen.

Nach dem Umräumen der CPU (Betriebsart STOP) übertragen Sie das Beispiel komplett in den Anwenderspeicher. Danach schalten Sie den Betriebsartenschalter von STOP auf RUN.

## Fehlerverhalten

Ist im Anlauf ein Fehler aufgetreten, werden die zyklisch bearbeiteten Bausteinaufrufe nicht ausgeführt, es wird die Fehleranzeige gesetzt.

Bei einer Fehlermeldung wird der Parameterausgang ERROR der Bausteine gesetzt. Eine genauere Fehlerbeschreibung ist dann im Parameter STATUS der Bausteine hinterlegt. Steht eine der Fehlermeldungen 16#1E0E oder 16#1E0F im STATUS, dann ist die genaue Fehlerbeschreibung in der Variablen SFCERR im Instanz-DB abgelegt.

## Einschalten, Anlaufprogramm

Das Anlaufprogramm steht im OB 100.

Im Anlauf werden die Steuerbits und die Zähler rückgesetzt

## Zyklisches Programm

Das zyklische Programm steht im OB 1.

Im Beispiel arbeiten die Funktionsbausteine FB 2 S\_RECV\_SI und FB 3 S\_SEND\_SI mit den Funktionen FC 21 und FC 22 zusammen, sowie mit den Datenbausteinen DB 21 und DB 22 als Instanz-DBs und DB 42 und DB 43 als Sende- bzw. Empfangs-DB.

Die Parametrierung der Funktionsbausteine erfolgt im Beispiel teils durch Konstanten und teils durch symbolisch adressierte Aktualoperanden.

## Beschreibung

Die Datenübertragung findet statt vom ET 200S 1SI auf Steckplatz 2 zum ET 200S 1SI auf Steckplatz 3. Wenn Sie mit einem anderen Kommunikationspartner arbeiten, entfällt der Aufruf des FC 22 (RECEIVE).

## Beschreibung FC 21 (SEND)

Programmteil "Generate edge S\_SEND\_SI\_REQ":

Der S\_SEND\_SI wird am Anfang einmal mit S\_SEND\_SI\_REQ=0 durchlaufen. Danach wird S\_SEND\_SI\_REQ auf 1 gesetzt. Wenn am Steuerparameter S\_SEND\_SI\_REQ ein Signalzustandswechsel von 0 nach 1 erkannt wird, wird der S\_SEND\_SI Auftrag gestartet.

Mit S\_SEND\_SI\_DONE=1 oder S\_SEND\_SI\_ERROR=1 wird S\_SEND\_SI\_REQ wieder auf 0 gesetzt.

Programmteil "S\_SEND\_SI\_DONE=1":

Bei einem erfolgreichen Transfer wird am Parameterausgang des S\_SEND\_SI der Parameter S\_SEND\_SI\_DONE auf 1 gesetzt.

Um zeitlich nacheinander ablaufende Transfers zu unterscheiden, wird im Datenwort 0 des Quellbausteins DB 42 ein Sendezähler S\_SEND\_SI\_COUNTER\_OK mitgeführt.

Programmteil "S\_SEND\_SI\_ERROR=1":

Wird der S\_SEND\_SI mit S\_SEND\_SI\_ERROR=1 durchlaufen, wird im Datenwort 2 der Fehlerzähler S\_SEND\_SI\_COUNTER\_ERR hochgezählt. Außerdem wird der S\_SEND\_SI\_WORK\_STAT umkopiert, da er im nächsten Durchlauf mit 0 überschrieben wird und dann nicht mehr ausgelesen werden könnte.

## Beschreibung FC 22 (RECEIVE)

Programmteil "Enable Receive Data":

Um Daten empfangen zu können, muss die Empfangsfreigabe S\_RECV\_SI\_EN\_R am Baustein S\_RECV\_SI auf 1 gesetzt sein.

Programmteil "S\_RECV\_SI\_NDR=1":

Wenn S\_RECV\_SI\_NDR gesetzt ist, sind neue Daten empfangen worden und es wird der Empfangszähler S\_RECV\_SI\_WORK\_CNT\_OK hochgezählt.

Programmteil "S\_RECV\_SI\_ERROR=1":

Bei fehlerhaftem Ablauf, d.h. wenn das Errorbit am Parameterausgang des S\_RECV\_SI gesetzt ist, wird der Fehlerzähler S\_RECV\_SI\_WORK\_CNT\_ERR hochgezählt. Außerdem wird der S\_RECV\_SI\_WORK\_STAT umkopiert, da er im nächsten Durchlauf mit 0 überschrieben wird und dann nicht mehr ausgelesen werden könnte.

Alle relevanten Werte können zum Test in der VAT beobachtet werden.

## 2.3 Schaltbild mit Anschlussbelegung

### Verdrahtungsregeln

Die Kabel (Klemmen 1 bis 8) müssen geschirmt sein. Der Schirm muss beidseitig aufgelegt werden. Verwenden Sie hierzu Schirm-Kontaktelemente (siehe Handbuch *Dezentrales Peripheriegerät ET 200S 1SI*).

### Anschlussbelegung für die RS-232-C-Kommunikation

Die folgende Tabelle zeigt die Anschlussbelegung für die serielle Schnittstellenbaugruppe ET 200S 1SI bei eingestelltem RS-232C-Kommunikationsprotokoll.

Tabelle 2-3 Anschlussbelegung für die serielle Schnittstellenbaugruppe ET 200S 1SI für die RS-232C-Kommunikation

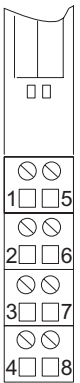
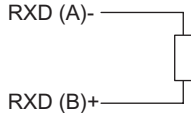
Ansicht		Bemerkungen		
		Modus: halb- und vollduplex		
		Klemmen		
		1	TXD	Gesendete Daten
		5	RXD	Empfangene Daten
		2	RTS	Sendeauftrag
		6	CTS	Sendebereit
		3	DTR	Datenterminal bereit
		7	DSR	Datensatz bereit
4	DCD	Datenträgererkennung		
8	PE	Erde		



### Anschlussbelegung für die RS-422-Kommunikation

Die folgende Tabelle zeigt die Anschlussbelegung für die serielle Schnittstellenbaugruppe ET 200S 1SI bei eingestelltem RS-422-Kommunikationsprotokoll.

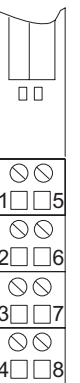
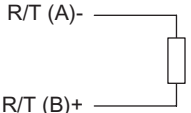
Tabelle 2- 4 Anschlussbelegung für die serielle Schnittstellenbaugruppe ET 200S 1SI für die RS-422-Kommunikation

Ansicht	Anschlussbelegung	Bemerkungen										
 <p style="margin-left: 20px;">TXD (A)- 1 □ □ 5</p> <p style="margin-left: 20px;">TXD (B)+ 2 □ □ 6</p> <p style="margin-left: 20px;">3 □ □ 7</p> <p style="margin-left: 20px;">4 □ □ 8 PE</p>	<p>Hinweis: Bei Kabeln über 50 m ergänzen Sie einen Abschlusswiderstand von ca. 330 Ω, damit problemloser Datenverkehr gewährleistet wird.</p> 	<p>Modus: voll duplex Klemmen</p> <table border="1" style="width: 100%;"> <tr> <td style="width: 10%;">1</td> <td>TXD (A)-</td> </tr> <tr> <td>5</td> <td>RXD (A)-</td> </tr> <tr> <td>2</td> <td>TXD (B)+</td> </tr> <tr> <td>6</td> <td>RXD (B)+</td> </tr> <tr> <td>8</td> <td>PE Erde</td> </tr> </table>	1	TXD (A)-	5	RXD (A)-	2	TXD (B)+	6	RXD (B)+	8	PE Erde
		1	TXD (A)-									
		5	RXD (A)-									
		2	TXD (B)+									
		6	RXD (B)+									
8	PE Erde											

### Anschlussbelegung für die RS-485-Kommunikation

Die folgende Tabelle zeigt die Anschlussbelegung für die serielle Schnittstellenbaugruppe ET 200S 1SI bei eingestelltem RS-485-Kommunikationsprotokoll.

Tabelle 2- 5 Anschlussbelegung für die serielle Schnittstellenbaugruppe ET 200S 1SI für die RS-485-Kommunikation

Ansicht	Anschlussbelegung	Bemerkungen						
 <p style="margin-left: 20px;">R/T (A)- 1 □ □ 5</p> <p style="margin-left: 20px;">R/T (B)+ 2 □ □ 6</p> <p style="margin-left: 20px;">3 □ □ 7</p> <p style="margin-left: 20px;">4 □ □ 8 PE</p>	<p>Hinweis: Bei Kabeln über 50 m ergänzen Sie einen Abschlusswiderstand von ca. 330 Ω, damit problemloser Datenverkehr gewährleistet wird.</p> 	<p>Modus: halbduplex Klemmen</p> <table border="1" style="width: 100%;"> <tr> <td style="width: 10%;">1</td> <td>R/T (A)-</td> </tr> <tr> <td>2</td> <td>R/T (B)+</td> </tr> <tr> <td>8</td> <td>PE Erde</td> </tr> </table>	1	R/T (A)-	2	R/T (B)+	8	PE Erde
		1	R/T (A)-					
		2	R/T (B)+					
		8	PE Erde					

**Anschlussbelegung des RS-232C-Anschlusskabels für 9-poligen Anschlussstecker**

Das folgende Bild zeigt die Leitungsanschlüsse für die Punkt-zu-Punkt-Kommunikation nach RS-232C zwischen der seriellen Schnittstellenbaugruppe ET 200S 1SI und einem Kommunikationspartner mit einer 9-poligen D-Anschlussbuchse.

- Auf der ET 200S 1SI-Seite werden die Signaldrähte an die entsprechend nummerierten Klemmen angeschlossen.
- Am Kommunikationspartner verwenden Sie eine 9-polige Sub-D-Anschlussbuchse.

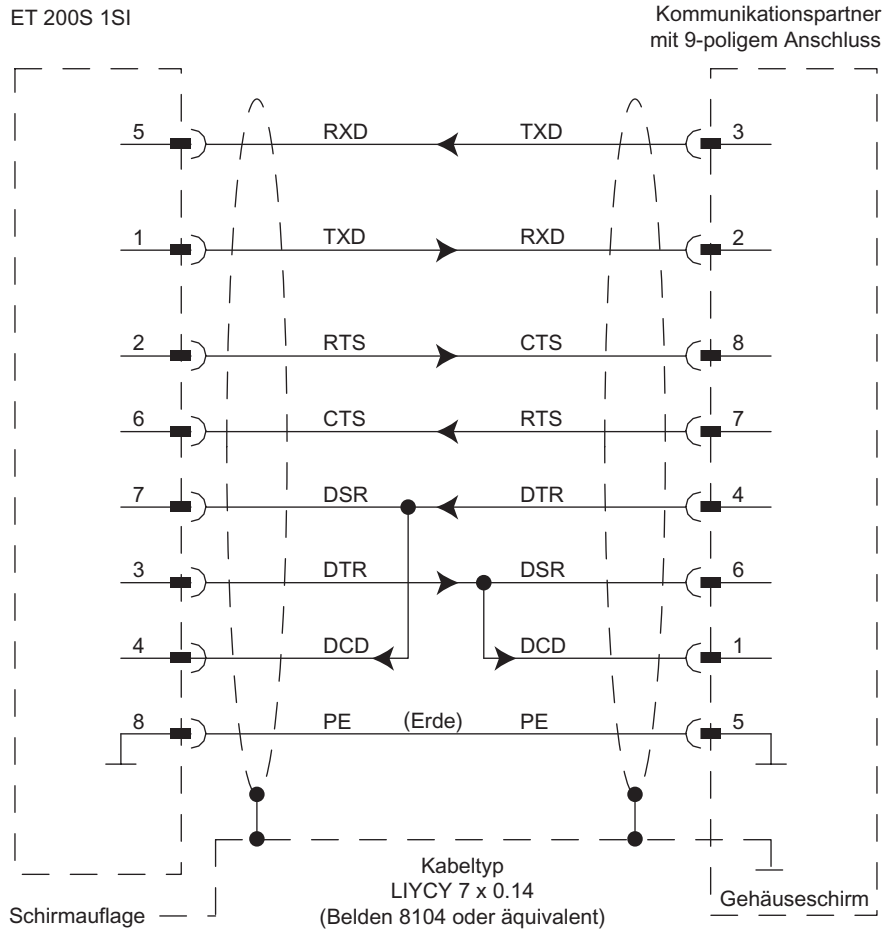


Bild 2-2 Anschlussbelegung des RS-232C-Anschlusskabels für 9-poligen Anschlussstecker

### Anschlussbelegung des RS-232C-Anschlusskabels für 25-poligen Anschlussstecker

Das folgende Bild zeigt die Leitungsanschlüsse für die Punkt-zu-Punkt-Kommunikation nach RS-232C zwischen der seriellen Schnittstellenbaugruppe ET 200S 1SI und einem Kommunikationspartner mit einem 25-poligen D-Anschlussstecker.

- Auf der ET 200S 1SI-Seite werden die Signaldrähte an die entsprechend nummerierten Klemmen angeschlossen.
- Am Kommunikationspartner verwenden Sie einen 25-poligen Sub-D-Stecker.

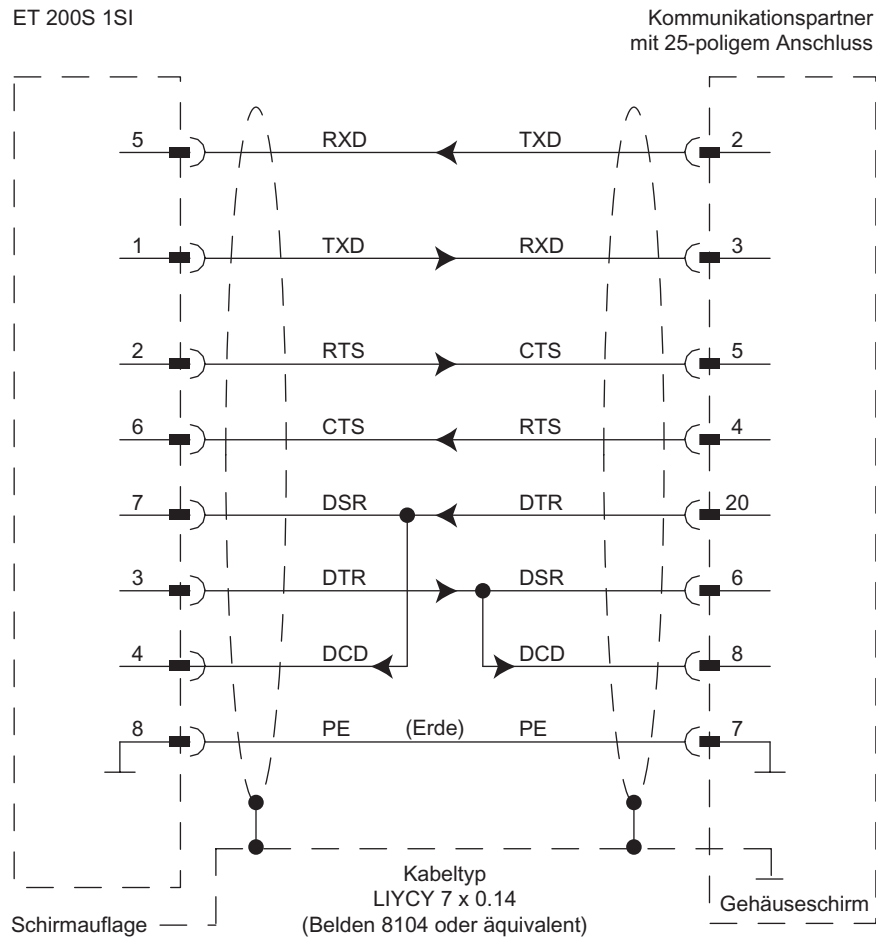


Bild 2-3 Anschlussbelegung des RS-232C-Anschlusskabels für 25-poligen Anschlussstecker

### Anschlussbelegung des RS-422-Anschlusskabels für 15-poligen Anschlussstecker

Das folgende Bild zeigt die Leitungsanschlüsse für die Punkt-zu-Punkt-Kommunikation über RS-422 zwischen der seriellen Schnittstellenbaugruppe ET 200S 1SI und einem Kommunikationspartner mit einem 15-poligen D-Anschlussstecker.

- Auf der ET 200S 1SI-Seite werden die Signaldrähte an die entsprechend nummerierten Klemmen angeschlossen.
- Am Kommunikationspartner verwenden Sie einen 15-poligen Sub-D-Stecker.

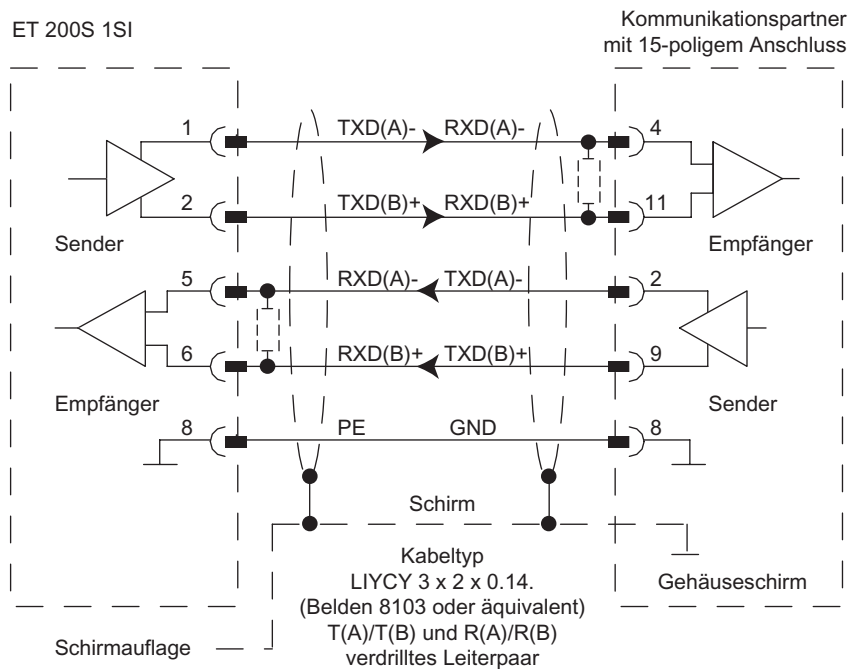


Bild 2-4 Anschlussbelegung des RS-422-Anschlusskabels für 15-poligen Anschlussstecker

#### Hinweis

Bei Kabeln über 50 m ergänzen Sie einen Abschlusswiderstand von ca. 330 Ω (siehe obiges Bild), damit problemloser Datenverkehr gewährleistet ist.

Bei dem verwendeten Kabeltyp sind für das ET 200S 1SI-Modul als Kommunikationspartner folgende Längen möglich:

- max. 1200 m bei 19.200 Baud
- max. 500 m bei 38.400 Baud
- max. 250 m bei 76.800 Baud

### Anschlussbelegung des RS-485-Anschlusskabels für 15-poligen Anschlussstecker

Das folgende Bild zeigt die Leitungsanschlüsse für die Punkt-zu-Punkt-Kommunikation über RS-485 zwischen der seriellen Schnittstellenbaugruppe ET 200S 1SI und einem Kommunikationspartner mit einem 15-poligen D-Anschlussstecker.

- Auf der ET 200S 1SI-Seite werden die Signaldrähte an die entsprechend nummerierten Klemmen angeschlossen.
- Am Kommunikationspartner verwenden Sie einen 15-poligen Sub-D-Stecker.

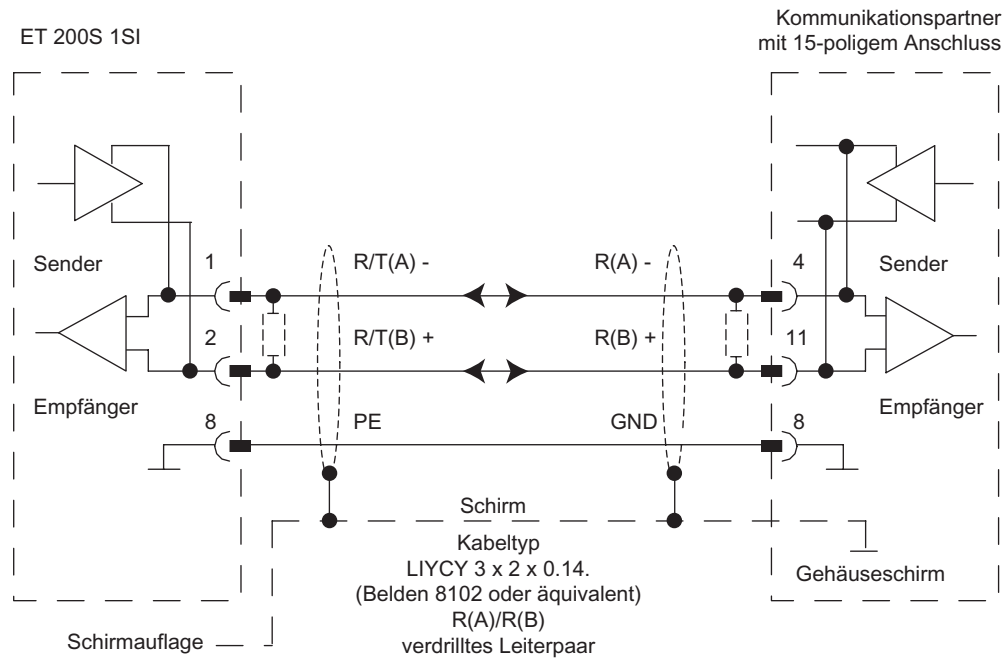


Bild 2-5 Anschlussbelegung des RS-485-Anschlusskabels für 15-poligen Anschlussstecker

#### Hinweis

Bei Kabeln über 50 m ergänzen Sie einen Abschlusswiderstand von ca. 330  $\Omega$  (siehe obiges Bild), damit problemloser Datenverkehr gewährleistet ist.

Bei dem verwendeten Kabeltyp sind für das ET 200S 1SI-Modul als Kommunikationspartner folgende Längen möglich:

- max. 1200 m bei 19.200 Baud
- max. 500 m bei 38.400 Baud
- max. 250 m bei 76.800 Baud
- max. 200 m bei 115.200 Baud

## 2.4 RS-232C-Schnittstelle

### Definition

Die RS-232C-Schnittstelle ist eine Spannungsschnittstelle und dient zur seriellen Datenübertragung nach der Norm RS-232C.

### Eigenschaften

Die RS-232C-Schnittstelle verfügt über folgende Eigenschaften:

Art:	Spannungsschnittstelle
Frontstecker:	8-poliger Standard-Klemmenstecker der ET 200S
RS-232C-Signale:	TXD, RXD, RTS, CTS, DTR, DSR, DCD, GND
Übertragungsgeschwindigkeit:	maximal 115,2 KBaud (Prozedur 3964(R)) maximal 115,2 KBaud (ASCII-Treiber)
Kabellänge:	maximal 15 m, Kabeltyp LIYCY 7 x 0.14
Normen:	DIN 66020, DIN 66259, EIA RS-232C, CCITT V.24/V.28
Schutzart:	IP20

### RS-232C-Signale

Nachfolgende Tabelle beschreibt die RS-232C-Signale.

Signal	Bezeichnung	Bedeutung
TXD	Transmitted Data	Sendedaten; Sendeleitung wird im Ruhezustand logisch auf "1" gehalten.
RXD	Received Data	Empfangsdaten; Empfangsleitung muss vom Kommunikationspartner auf logisch "1" gehalten werden.
RTS	Request To Send	ON: ET 200S 1SI ist sendebereit. OFF: ET 200S 1SI sendet nicht.
CTS	Clear To Send	Kommunikationspartner kann Daten vom ET 200S empfangen. Die Schnittstellenbaugruppe erwartet dies als Antwort auf RTS gleich ON.
DTR	Data Terminal Ready	ON: ET 200S SI ist eingeschaltet und betriebsbereit. OFF: ET 200S SI ist nicht eingeschaltet und nicht betriebsbereit.
DSR	Data Set Ready	ON: Kommunikationspartner ist eingeschaltet und betriebsbereit. OFF: Kommunikationspartner ist nicht eingeschaltet und nicht betriebsbereit.
DCD	Data Carrier Detect	Trägersignal bei Anschluss eines Modems.

## 2.5 RS-422/485-Schnittstelle

### Definition

Die RS-422/485-Schnittstelle ist eine Spannungsdifferenz-Schnittstelle und dient zur seriellen Datenübertragung nach der Norm RS-422/485.

### Eigenschaften

Die RS-422/485-Schnittstelle verfügt über folgende Eigenschaften:

Art:	Spannungsdifferenz-Schnittstelle
Frontstecker:	8-poliger Standard-Klemmenstecker der ET 200S
RS-422-Signale:	TXD (A)-, RXD (A)-, TXD (B)+, RXD (B)+, GND
RS-485-Signale:	R/T (A)-, R/T (B)+, GND
Übertragungsgeschwindigkeit:	maximal 115,2 KBaud (Prozedur 3964(R)) maximal 115,2 KBaud (ASCII-Treiber)
Kabellänge:	maximal 1200 m, Kabeltyp LIYCY 7 x 0.14
Normen:	EIA RS-422/485, CCITT V.11/V.27
Schutzart:	IP20

## 2.6 Grundlagen der seriellen Datenübertragung

### 2.6.1 Serielle Datenübertragung

#### Punkt-zu-Punkt-Kopplung

Zum Austausch von Daten zwischen zwei oder mehr Kommunikationspartnern stehen unterschiedliche Möglichkeiten der Vernetzung zur Verfügung. Die Punkt-zu-Punkt-Kopplung zwischen zwei Kommunikationspartnern ist der einfachste Fall des Informationsaustauschs.

Bei der Punkt-zu-Punkt-Kopplung bildet die serielle Schnittstellenbaugruppe die Schnittstelle zwischen einer speicherprogrammierbaren Steuerung und einem Kommunikationspartner. Die Übertragung der Daten erfolgt bei der Punkt-zu-Punkt-Kopplung mit der seriellen Schnittstellenbaugruppe ET 200S 1SI seriell.

#### Serielle Datenübertragung

Bei der seriellen Datenübertragung werden die einzelnen Bits eines Byte einer zu übertragenden Information in einer festgelegten Reihenfolge nacheinander übertragen.

Die Datenübertragung mit dem Kommunikationspartner wird über die serielle Schnittstelle von der Schnittstellenbaugruppe ET 200S 1SI selbständig abgewickelt. Die Baugruppe ist hierfür mit zwei verschiedenen Treibern für den bidirektionalen Datenverkehr ausgestattet.

- ASCII-Treiber
- Prozedur 3964(R)

#### Bidirektionaler Datenverkehr - Betriebsarten

Beim bidirektionalen Datenverkehr unterscheidet man beim ET 200S 1SI zwei Betriebsarten:

- Halbduplexbetrieb (Prozedur 3964(R), ASCII-Treiber)

Die Daten werden zwischen den Kommunikationspartnern abwechselnd in beide Richtungen übertragen. Halbduplexbetrieb bedeutet, dass zu einem Zeitpunkt entweder gesendet oder empfangen wird. Die Ausnahme hiervon können einzelne Steuerzeichen zur Datenflusskontrolle (z.B. XON/XOFF) bilden, die auch während eines Sende-/Empfangsbetriebs empfangen/gesendet werden können.

- Vollduplexbetrieb (ASCII-Treiber)

Die Daten werden zwischen den Kommunikationspartnern in beide Richtungen gleichzeitig ausgetauscht. Vollduplexbetrieb bedeutet, dass gleichzeitig gesendet und empfangen werden kann. Jeder Kommunikationspartner muss simultan eine Sende- und Empfangseinrichtung betreiben können.



Die folgende Tabelle führt die Betriebsarten des Datenverkehrs für die Schnittstellenarten mit ASCII-Treibern auf.

Tabelle 2- 6 Betriebsarten des Datenverkehrs für die serielle Schnittstellenbaugruppe ET 200S 1SI

Datenverkehr	RS-232C	RS-422	RS-485
Halbduplex	Ja	Ja	Ja
Vollduplex	Ja	Ja	Nicht möglich

## Vereinbarungen

Für die serielle Datenübertragung sind zwischen den beiden Kommunikationspartnern Vereinbarungen nötig. Darunter fallen:

- Übertragungsgeschwindigkeit (Baudrate)
- Zeichen- und Quittungsverzugszeit
- Parität
- Anzahl Datenbits
- Anzahl Stoppbits
- Anzahl Aufbau- und Übertragungsversuche

Abschnitt Grundlagen der Datenübertragung mit der Prozedur 3964(R) (Seite 33) und Grundlagen zur Datenübertragung mit ASCII-Treiber (Seite 42) beschreiben, welche Rolle die Absprachen bei den verschiedenen Übertragungsverfahren spielen und wie sie parametrieren werden.

## 2.6.2 Zeichenrahmen

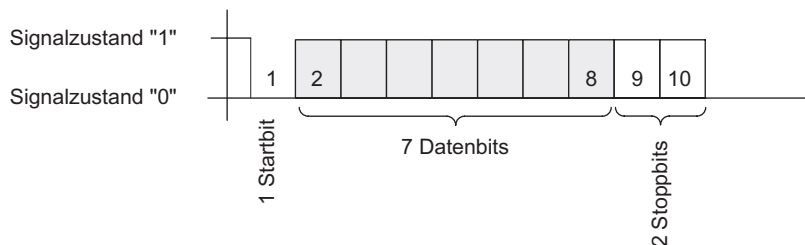
### Prinzip

Die Daten zwischen der seriellen Schnittstellenbaugruppe ET 200S 1SI und einem Kommunikationspartner werden über die serielle Schnittstelle in einem 10-Bit- bzw. 11-Bit-Zeichenrahmen übertragen. Für jeden Zeichenrahmen stehen drei Datenformate zur Verfügung. Sie können das benötigte Format in STEP 7 parametrieren.

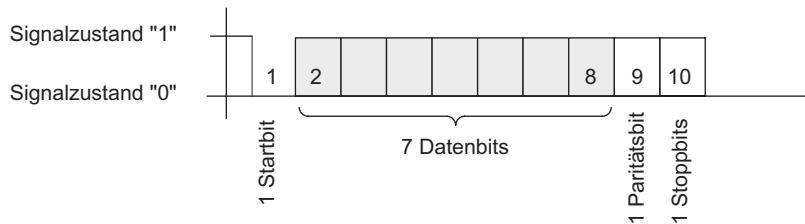
### 10-Bit-Zeichenrahmen

Das folgende Bild zeigt die drei Datenformate des 10-Bit-Zeichenrahmens.

7 Datenbits: 1 Startbit, 7 Datenbits, 2 Stoppbits



7 Datenbits: 1 Startbit, 7 Datenbits, 1 Paritätsbit, 1 Stoppbit



8 Datenbits: 1 Startbit, 8 Datenbits, 1 Stoppbit

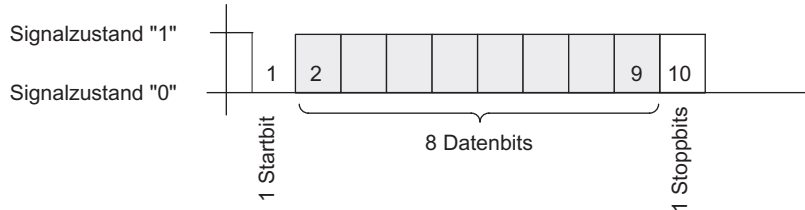
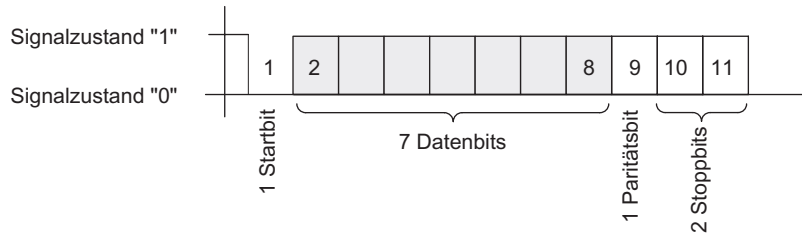


Bild 2-6 10-Bit-Zeichenrahmen

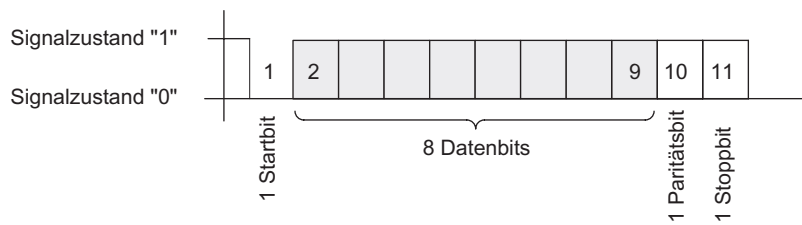
## 11-Bit-Zeichenrahmen

Das folgende Bild zeigt die drei Datenformate des 11-Bit-Zeichenrahmens.

7 Datenbits: 1 Startbit, 7 Datenbits, 1 Paritätsbit, 2 Stoppbits



8 Datenbits: 1 Startbit, 8 Datenbits, 1 Paritätsbit, 1 Stoppbit



8 Datenbits: 1 Startbit, 8 Datenbits, 2 Stoppbits

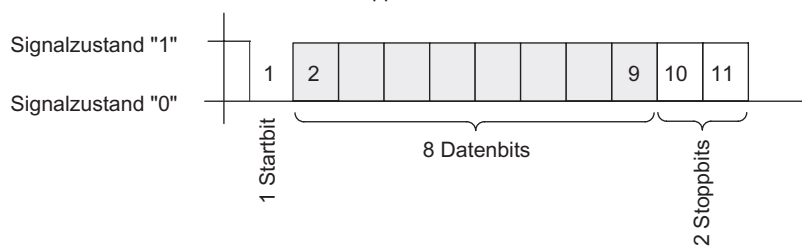


Bild 2-7 11-Bit-Zeichenrahmen

### Zeichenverzugszeit

Das folgende Bild zeigt den maximal zulässigen zeitlichen Abstand zwischen zwei empfangenen Zeichen innerhalb eines Telegramms. Dieser Abstand wird Zeichenverzugszeit genannt.

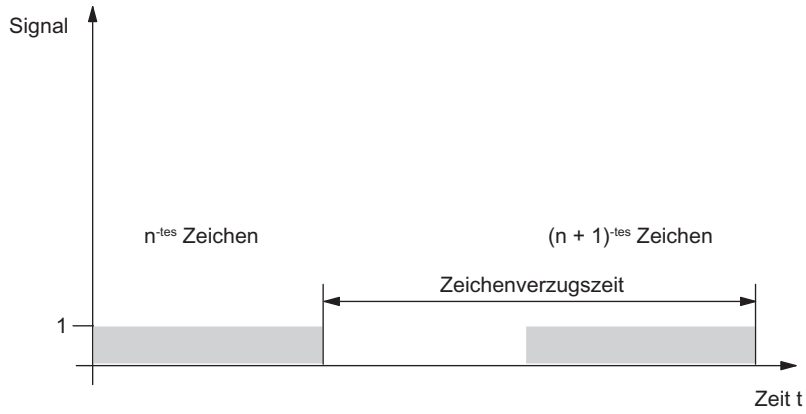


Bild 2-8 Zeichenverzugszeit

### 2.6.3 Übertragungsverfahren bei der Punkt-zu-Punkt-Kopplung

Bei einer Datenübertragung müssen sich alle Kommunikationspartner an feste Regeln für die Abwicklung und Durchführung des Datenverkehrs halten. Die ISO hat ein 7-Schichtenmodell festgeschrieben, das als Basis einer weltweiten Normung von Übertragungsprotokollen anerkannt ist.

#### Protokoll

Alle Kommunikationspartner einer Datenübertragung müssen sich an feste Regeln für die Abwicklung und Durchführung des Datenverkehrs halten. Diese Regeln werden Protokolle genannt.

Ein Protokoll legt im Einzelnen fest:

- **die Betriebsart**  
Halbduplex- oder Vollduplexbetrieb
- **die Initiative**  
Vereinbarungen, welcher Kommunikationspartner und unter welchen Bedingungen der Kommunikationspartner die Initiative zur Datenübertragung ergreifen darf.
- **die Steuerzeichen**  
Festlegung der zur Datenübertragung verwendeten Steuerzeichen
- **den Zeichenrahmen**  
Festlegung, welcher Zeichenrahmen zur Datenübertragung benutzt wird.
- **die Datensicherung**  
Festlegung des Datensicherungsverfahrens
- **die Zeichenverzugszeit**  
Festlegung des Zeitraums, wann ein zu empfangendes Zeichen ankommen muss.
- **die Übertragungsgeschwindigkeit**  
Festlegung der Baudrate in Bit/s

#### Prozedur

Der Ablauf einer Datenübertragung nach einem bestimmten Verfahren wird dabei als Prozedur bezeichnet.

### ISO-7-Schichten-Referenzmodell

Das Referenzmodell definiert das externe Verhalten der Kommunikationspartner. Jede Protokollschicht ist in die nächstniedrigere Schicht eingebettet, mit Ausnahme der untersten Schicht.

Die einzelnen Schichten sind wie folgt festgelegt:

**1. Bitübertragungsschicht**

- Physikalische Voraussetzungen für die Datenübertragung, z. B. Übertragungsmedium, Baudrate

**2. Sicherungsschicht**

- Sicherungsverfahren der Datenübertragung
- Zugriffsverfahren

**3. Vermittlungsschicht**

- Festlegung der Kommunikationswege
- und der Adressierung für die Datenübertragung zwischen zwei Kommunikationspartnern

**4. Transportschicht**

- Fehlererkennungsverfahren
- Korrekturmaßnahmen
- Handshakeverfahren

**5. Kommunikationssteuerungsschicht**

- Aufbau der Datenübertragung
- Durchführung
- Abbau der Datenübertragung

**6. Darstellungsschicht**

- Umsetzung der normierten Darstellungsart des Kommunikationssystems in eine gerätespezifische Form (Interpretationsvorschriften der Daten)

**7. Verarbeitungsschicht**

- Festlegung der Kommunikationsaufgabe und der dafür notwendigen Funktionen

### Abarbeitung der Protokolle

Der sendende Kommunikationspartner durchläuft die Protokolle von der obersten Schicht (Nr. 7, anwendungsorientiert) zur untersten Schicht (Nr. 1, physikalische Festlegungen), während der empfangende Kommunikationspartner die Protokolle von Schicht 1 aufwärts abarbeitet.

Nicht jedes Protokoll muss alle 7 Schichten berücksichtigen. Sprechen sendender und empfangender Kommunikationspartner dieselbe Sprache, ist die Schicht 6 hinfällig.

## 2.6.4 Übertragungssicherheit

### Prinzip

Die Übertragungssicherheit spielt bei der Übertragung von Daten und bei der Wahl des Übertragungsverfahrens eine wichtige Rolle. Allgemein gilt, je mehr Schichten des Referenzmodells durchlaufen werden, desto höher ist die Übertragungssicherheit.

### Unterstützte Protokolle

Das folgende Bild zeigt, wie die von der Schnittstellenbaugruppe ET 200S 1SI unterstützten Protokolle ASCII und 3964(R) in das ISO-Referenzmodell einzuordnen sind.

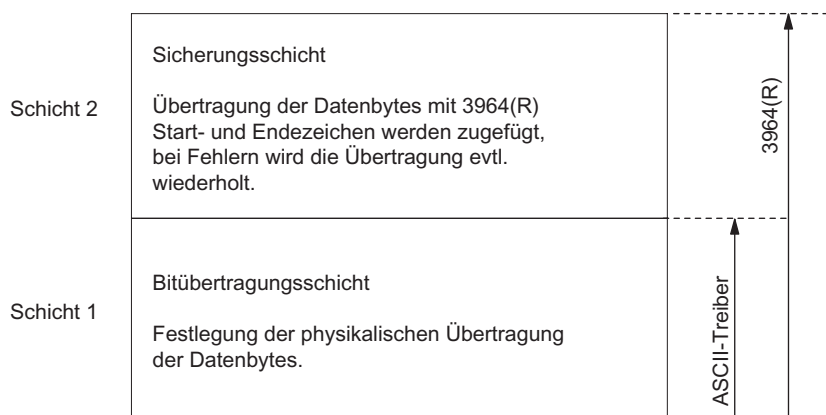


Bild 2-9 Einordnung der unterstützten Protokolle in das Referenzmodell

### Übertragungssicherheit beim ASCII-Treiber

Richten Sie sich nach den folgenden Richtlinien, um die Datensicherheit bei Verwendung des ASCII-Treibers zu erhöhen:

- Bei einem Datentransport mit dem ASCII-Treiber gibt es neben der Verwendung des Paritätsbits (kann auch abhängig von der Einstellung des Zeichenrahmens abgewählt werden) keine weiteren Maßnahmen zur Datensicherung. Deshalb ist eine Datenübertragung mit dem ASCII-Treiber zwar sehr effizient, was den Datendurchsatz anbelangt, ein abgesicherter Datentransport ist jedoch nicht gegeben.
- Durch die Verwendung des Paritätsbits wird das Kippen eines Bits in einem zu übertragenden Zeichen abgesichert. Kippen zwei oder mehr Bits eines Zeichens, so kann dieser Fehler nicht mehr erkannt werden.
- Soll die Übertragungssicherheit erhöht werden, so kann dies durch Einführung einer Prüfsumme und Längenangabe eines Telegramms erfolgen. Diese Maßnahmen müssen durch den Anwender ergriffen werden.
- Eine weitere Steigerung der Datensicherheit kann durch die Einführung von Quittungstelegrammen auf Sende- oder Empfangstelegramme erfolgen. Dies ist ebenfalls bei hochwertigen Protokollen zur Datenkommunikation der Fall (siehe ISO-7-Schichten-Referenzmodell).

### Übertragungssicherheit bei 3964(R)

Die Prozedur 3964(R) bietet erhöhte Datensicherheit:

- Die Hamming-Distanz bei 3964(R) beträgt 3. Die Hamming-Distanz ist ein Maß für die Sicherheit einer Datenübertragung.
- Durch die Prozedur 3964(R) wird eine hohe Übertragungssicherheit in der Übertragungsleitung gewährleistet. Die hohe Übertragungssicherheit wird durch einen festgelegten Telegrammauf- und -abbau sowie die Mitführung des Blockprüfzeichens (BCC) erreicht.

Je nachdem, ob Sie mit oder ohne Blockprüfzeichen Daten übertragen wollen, unterscheidet man zwischen

- Datenübertragung ohne Blockprüfzeichen: **3964**
- Datenübertragung mit Blockprüfzeichen: **3964(R)**

Bei Beschreibungen und Hinweisen, die sich auf beide Datenübertragungsarten beziehen, verwenden wir in diesem Handbuch die Bezeichnung **3964(R)**.

### Grenzen der Leistungsfähigkeit bei 3964(R)

- Nicht sichergestellt wird die programmtechnische Weiterverarbeitung der Sende-/Empfangsdaten im Kommunikationspartner. Dies können Sie nur durch einen zu programmierenden Quittungsmechanismus sicherstellen.
- Durch die Blockprüfung der Prozedur 3964R (EXOR-Verknüpfung) kann das Fehlen von Nullen (als ganzes Zeichen) nicht erkannt werden, da eine Null bei der EXOR-Verknüpfung nichts zur Veränderung des Rechenergebnisses beiträgt.

Der Verlust eines ganzen Zeichens (dieses Zeichen muss dann gerade eine Null sein) ist zwar sehr unwahrscheinlich, kann aber z.B. bei sehr schlechten Übertragungsbedingungen vorkommen.

Gegen solche Fehler können Sie eine Übertragung absichern, indem Sie zusätzlich zu den zu sendenden Daten die Länge des Datentelegramms mitsenden und im Kommunikationspartner auswerten.



## 2.7 Datenübertragung mit der Prozedur 3964(R)

### 2.7.1 Grundlagen der Datenübertragung mit der Prozedur 3964(R)

#### Prinzip

Die Prozedur 3964(R) steuert die Datenübertragung bei einer Punkt-zu-Punkt-Kopplung zwischen der Baugruppe ET 200S und einem Kommunikationspartner. Die Prozedur 3964(R) beinhaltet neben der Bitübertragungsschicht (Schicht 1) die Sicherungsschicht (Schicht 2).

#### Steuerzeichen

Die Prozedur 3964(R) fügt bei der Datenübertragung den Nutzdaten Steuerzeichen hinzu (Sicherungsschicht). Durch diese Steuerzeichen kann der Kommunikationspartner kontrollieren, ob die Daten vollständig und fehlerfrei bei ihm angekommen sind.

Die Prozedur 3964(R) wertet die folgenden Steuerzeichen aus:

- **STX**: Start of Text;  
Anfang der zu übertragenden Zeichenfolge
- **DLE**: Data Link Escape;  
Datenübertragungsumschaltung
- **ETX**: End of Text;  
Ende der zu übertragenden Zeichenfolge
- **BCC**: Block Check Character (nur bei 3964R);  
Blockprüfzeichen
- **NAK**: Negative Acknowledge;  
Negative Rückmeldung

---

#### Hinweis

Wird als Informationszeichen das Zeichen DLE übertragen, so wird dieses zur Unterscheidung vom Steuerzeichen DLE beim Verbindungsaufbau und -abbau auf der Sendeleitung doppelt gesendet (DLE-Verdopplung). Der Empfänger macht die DLE-Verdopplung wieder rückgängig.

---

#### Priorität

Bei der Prozedur 3964(R) muss dem einen Kommunikationspartner eine höhere und dem anderen Kommunikationspartner eine niedrigere Priorität zugeordnet sein. Wenn beide Kommunikationspartner gleichzeitig einen Sendeauftrag erteilen, dann stellt der Partner mit niedriger Priorität seinen Sendeauftrag zurück.

**Blockprüfsumme**

Beim Übertragungsprotokoll 3964R wird die Datensicherheit durch ein zusätzlich gesendetes Blockprüfzeichen (BCC = Block Check Character) erhöht (siehe folgendes Bild).

Telegramm:

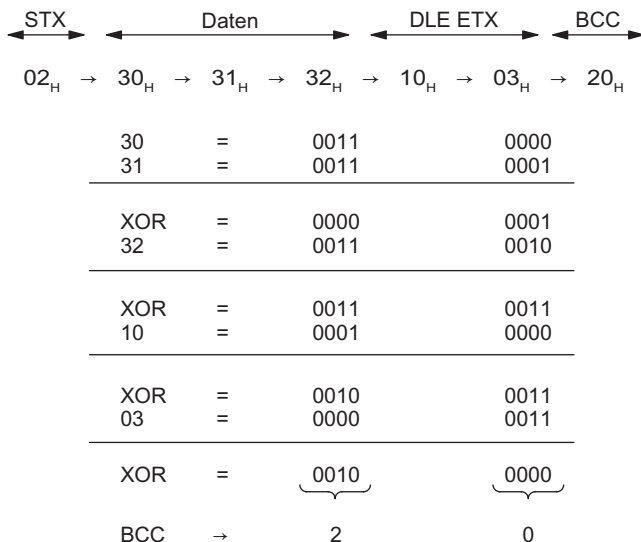


Bild 2-10 Blockprüfsumme

Die Blockprüfsumme ist die gerade Längsparität (EXOR-Verknüpfung aller Datenbytes) eines gesendeten bzw. empfangenen Blocks. Die Bildung beginnt mit dem ersten Nutzdatenbyte (1. Byte des Telegramms) nach dem Verbindungsaufbau und endet nach dem Zeichen DLE ETX beim Verbindungsabbau.

**Hinweis**

Bei einer DLE-Verdopplung wird das Zeichen DLE zweimal in die BCC-Bildung einbezogen.

## 2.7.2 Daten senden mit der Prozedur 3964(R)

### Daten senden mit 3964(R)

Das folgende Bild zeigt den Ablauf der Datenübertragung beim Senden mit der Prozedur 3964(R).

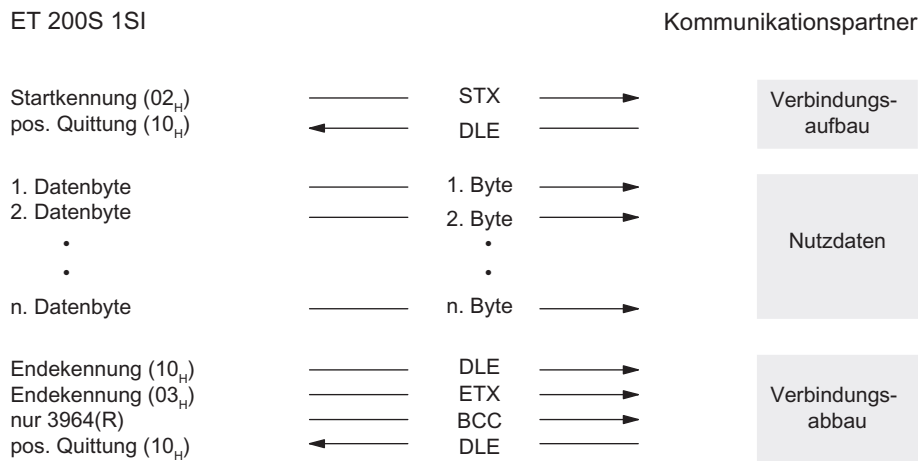


Bild 2-11 Datenverkehr beim Senden mit der Prozedur 3964(R)

### Verbindungsaufbau beim Senden

Zum Aufbau der Verbindung sendet die Prozedur 3964(R) das Steuerzeichen STX. Antwortet der Kommunikationspartner vor Ablauf der Quittungsverzugszeit (QVZ) mit dem Zeichen DLE, geht die Prozedur in den Sendebetrieb über.

Antwortet der Kommunikationspartner mit NAK, einem beliebigen anderen Zeichen (außer DLE) oder verstreicht die Quittungsverzugszeit ohne Reaktion, wiederholt die Prozedur den Verbindungsaufbau. Nach der parametrisierten Anzahl vergeblicher Aufbauversuche bricht die Prozedur den Verbindungsaufbau ab und sendet das Zeichen NAK an den Kommunikationspartner. Das Systemprogramm meldet den Fehler an den Funktionsbaustein S\_SEND (Ausgangsparameter STATUS).

### Daten senden

Gelingt der Verbindungsaufbau, werden die im Ausgabepuffer der Baugruppe ET 200S enthaltenen Nutzdaten mit den gewählten Übertragungsparametern an den Kommunikationspartner gesendet. Dieser überwacht den zeitlichen Abstand der ankommenden Zeichen. Der Abstand zwischen zwei Zeichen darf nicht mehr als die Zeichenverzugszeit (ZVZ) betragen.

### Verbindungsabbau beim Senden

Sendet der Kommunikationspartner während einer laufenden Sendung das Zeichen NAK, bricht die Prozedur den Block ab und wiederholt ihn in der oben beschriebenen Weise. Bei einem anderen Zeichen wartet die Prozedur zunächst auf den Ablauf der Zeichenverzugszeit und sendet anschließend NAK, um den Kommunikationspartner in den Ruhezustand zu bringen. Danach beginnt die Prozedur das Senden erneut mit dem Verbindungsaufbau STX.

Nach dem Senden des Pufferinhalts fügt die Prozedur die Zeichen DLE, ETX und **nur bei 3964(R)** die Blockprüfsumme BCC als Endekennung an und wartet auf ein Quittungszeichen. Sendet der Kommunikationspartner innerhalb der Quittungsverzugszeit das Zeichen DLE, wurde der Datenblock fehlerfrei übernommen. Antwortet der Kommunikationspartner mit NAK, einem beliebigen anderen Zeichen (außer DLE), einem gestörten Zeichen oder verstreicht die Quittungsverzugszeit ohne Reaktion, beginnt die Prozedur das Senden erneut mit dem Verbindungsaufbau STX.

Nach Ablauf der parametrisierten Anzahl von Übertragungsversuchen, den Datenblock zu senden, bricht die Prozedur das Verfahren ab und sendet NAK an den Kommunikationspartner. Das Systemprogramm meldet den Fehler an den Funktionsbaustein S\_SEND (Ausgangsparameter STATUS).

### 2.7.3 Daten empfangen mit der Prozedur 3964(R)

#### Daten empfangen mit 3964(R)

Das folgende Bild zeigt den Ablauf der Datenübertragung beim Empfangen mit der Prozedur 3964(R).

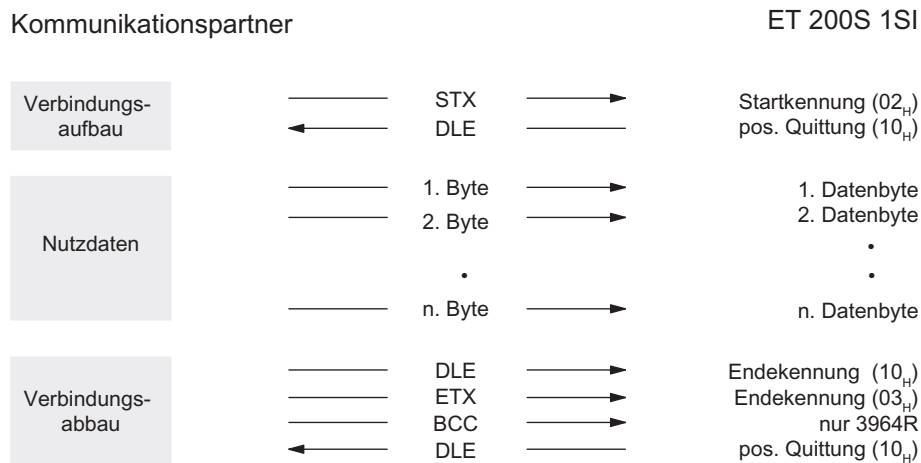


Bild 2-12 Datenverkehr beim Empfangen mit der Prozedur 3964(R)

#### Verbindungsaufbau beim Empfangen

Im Ruhezustand, wenn kein Sendeauftrag zu bearbeiten ist, wartet die Prozedur auf den Aufbau der Verbindung durch den Kommunikationspartner.

Empfängt die Prozedur im Ruhezustand ein beliebiges Zeichen (außer STX oder NAK), wartet Sie auf den Ablauf der Zeichenverzugszeit (ZVZ) und sendet dann das Zeichen NAK.

#### Daten empfangen

Empfängt die Prozedur das Zeichen STX und steht ihr ein leerer Empfangspuffer zur Verfügung, antwortet sie mit DLE. Ankommende Empfangszeichen werden nun im Empfangspuffer abgelegt. Werden zwei aufeinanderfolgende Zeichen DLE empfangen, wird nur ein Zeichen DLE in den Empfangspuffer übernommen.

Nach jedem Empfangszeichen wird während der Zeichenverzugszeit auf das nächste Zeichen gewartet. Verstreicht die Zeichenverzugszeit ohne Empfang, wird das Zeichen NAK an den Kommunikationspartner gesendet. Das Systemprogramm meldet den Fehler an den Funktionsbaustein S\_RCV (Ausgangsparameter STATUS).

Steht beim Verbindungsaufbau mit STX kein leerer Empfangspuffer zur Verfügung, wird eine Wartezeit von 400 ms gestartet. Liegt nach dieser Zeit noch kein leerer Empfangspuffer vor, meldet das Systemprogramm den Fehler (Fehlermeldung am Ausgang STATUS des FB). Die Prozedur sendet ein Zeichen NAK und geht wieder in den Ruhezustand zurück. Andernfalls sendet die Prozedur das Zeichen DLE und empfängt die Daten in der oben beschriebenen Weise.

### Verbindungsabbau beim Empfangen

Treten während des Empfangens Übertragungsfehler auf (verlorenes Zeichen, Rahmenfehler, Paritätsfehler usw.), so wird bis zum Verbindungsabbau weiterempfangen und dann NAK an den Kommunikationspartner gesendet. Anschließend wird eine Wiederholung erwartet. Kann der Block auch nach der bei der Parametrierung angegebenen Anzahl von Wiederholversuchen nicht fehlerfrei empfangen werden, oder wird die Wiederholung vom Kommunikationspartner nicht innerhalb einer Blockwartezeit von 4 s gestartet, bricht die Prozedur den Empfang ab. Das Systemprogramm meldet den Fehler an den Funktionsbaustein S\_RCV (Ausgangsparameter STATUS).

Erkennt die Prozedur 3964(R) die Zeichenfolge DLE ETX, beendet sie den Empfang und sendet DLE für einen fehlerfrei empfangenen Block an den Kommunikationspartner. Bei einem Empfangsfehler wird NAK an den Kommunikationspartner gesendet. Anschließend wird eine Wiederholung erwartet.

Erkennt die Prozedur 3964(R) die Zeichenfolge DLE ETX BCC, beendet sie den Empfang. Sie vergleicht das empfangene Blockprüfzeichen BCC mit der intern gebildeten Längsparität. Ist das Blockprüfzeichen korrekt und kein anderer Empfangsfehler aufgetreten, sendet die Prozedur 3964(R) DLE und kehrt zurück in den Ruhezustand. Bei fehlerhaftem BCC oder einem anderen Empfangsfehler wird NAK an den Kommunikationspartner gesendet. Anschließend wird eine Wiederholung erwartet.

---

#### Hinweis

Die Prozedur 3964(R) sendet, sobald sie betriebsbereit ist, einmal das Zeichen NAK an den Partner, um den Kommunikationspartner in den Ruhezustand zu bringen.

---

### Prozedur-Parameter

Wählen Sie bei beiden Koppelpartnern einer 3964(R)-Kommunikationsstrecke die folgenden Prozedur-Parameter identisch:

- Zeichenverzugszeit
- Quittungsverzugszeit
- Aufbauversuche
- Übertragungsversuche

#### Ausnahme:

Wenn Sie das Modul ET 200S 1SI mit niedriger Priorität betreiben, dann müssen Sie bei diesem Modul die Anzahl der Aufbauversuche um mindestens "1" niedriger parametrieren als beim Koppelpartner, damit mögliche Initialisierungskonflikte schneller aufgelöst werden (siehe Abschnitt Initialisierungskonflikt im Kapitel Fehlerbehandlung bei der Prozedur 3964(R) (Seite 39)).

## 2.7.4 Fehlerbehandlung bei der Prozedur 3964(R)

### Behandlung fehlerbehafteter Daten

Das folgende Bild zeigt den Ablauf bei der Behandlung fehlerbehafteter Daten mit der Prozedur 3964(R).

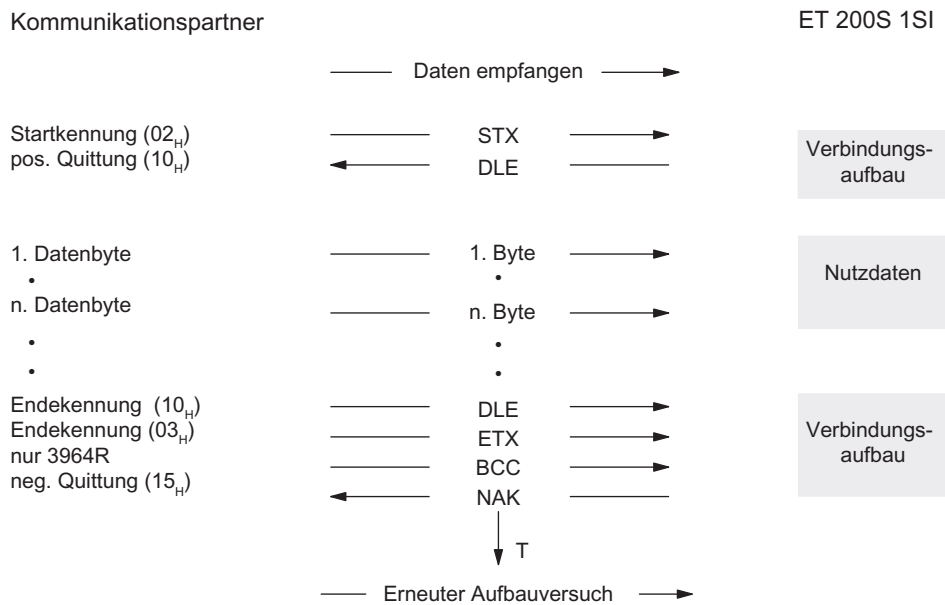


Bild 2-13 Datenverkehr beim Empfangen fehlerbehafteter Daten

Nach dem Empfang von DLE, ETC, BCC vergleicht die Baugruppe ET 200S 1SI den BCC des Kommunikationspartners mit dem eigenen intern gebildeten Wert. Ist der BCC korrekt und kein anderer Empfangsfehler aufgetreten, antwortet die Baugruppe ET 200S 1SI mit DLE.

Andernfalls antwortet die Baugruppe mit NAK und wartet die Blockwartezeit (T) von 4 s auf einen erneuten Versuch. Kann der Block nach der parametrisierten Anzahl von Übertragungsversuchen nicht empfangen werden oder wird kein weiterer Versuch in der Blockwartezeit unternommen, bricht die Baugruppe ET 200S 1SI den Empfang ab.

### Initialisierungskonflikt

Das folgende Bild zeigt den Ablauf der Datenübertragung bei einem Initialisierungskonflikt.

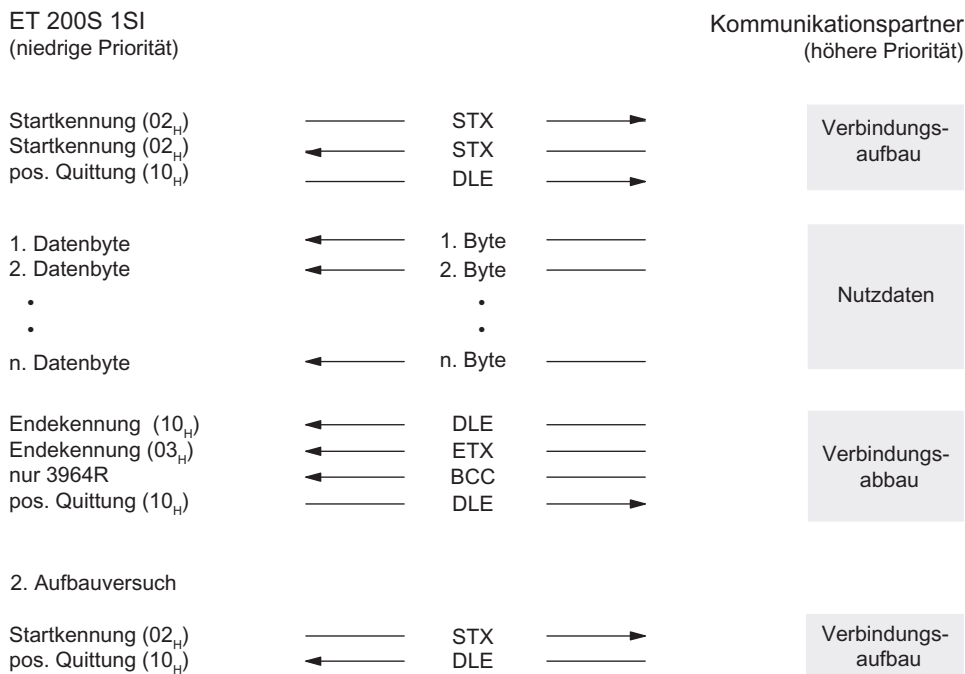


Bild 2-14 Datenverkehr bei Initialisierungskonflikt

Antwortet ein Gerät auf den Sendewunsch (Zeichen STX) des Kommunikationspartners innerhalb der Quittungsverzugszeit (QVZ) nicht mit der Quittung DLE oder NAK, sondern mit dem Zeichen STX, liegt ein Initialisierungskonflikt vor. Beide Geräte möchten einen vorliegenden Sendeauftrag ausführen. Das Gerät mit der niedrigeren Priorität stellt seinen Sendeauftrag zurück und antwortet mit dem Zeichen DLE. Das Gerät mit der höheren Priorität sendet seine Daten in der zuvor beschriebenen Weise. Nach dem Verbindungsabbau kann das Gerät mit der niedrigeren Priorität seinen Sendeauftrag ausführen.

Um den Initialisierungskonflikt aufzulösen, müssen Sie die Kommunikationspartner mit unterschiedlichen Prioritäten parametrieren.



## Prozedurfehler

Die Prozedur erkennt sowohl Fehler, die durch ein fehlerhaftes Verhalten des Kommunikationspartners ausgelöst werden, als auch Fehler, die durch Störungen in der Leitung verursacht werden.

In beiden Fällen wird zunächst versucht, beim Wiederholen den Datenblock richtig zu senden/zu empfangen. Kann der Datenblock bis zur Maximalanzahl der Wiederholungen nicht fehlerfrei gesendet oder empfangen werden (oder ergibt sich ein neuer Fehlerzustand), bricht die Prozedur das Senden bzw. Empfangen ab. Sie meldet die Fehlernummer für den ersten erkannten Fehler und begibt sich in den Ruhezustand. Diese Fehlermeldungen werden am Ausgang STATUS des FB angezeigt.

Falls des Öfteren eine Fehlernummer am Ausgang STATUS des FB für Sende- und Empfangswiederholungen anliegt, lässt dies auf gelegentliche Störungen des Datenverkehrs schließen. Die Vielzahl der Wiederholungen gleicht dies jedoch aus. In diesem Fall empfehlen wir Ihnen, die Übertragungstrecke auf Störeinflüsse zu untersuchen, da die Nutzdatenrate und Sicherheit der Übertragung bei vielen Wiederholungen sinkt. Die Ursache der Störung kann jedoch auch in einem fehlerhaften Verhalten des Kommunikationspartners liegen.

Bei BREAK in der Empfangsleitung (Empfangsleitung unterbrochen) wird ein BREAK-Zustand (BREAK-Anzeige über Diagnosealarm der Baugruppe ET 200S gemeldet (siehe Abschnitt Diagnose (Seite 101)). Es wird keine Wiederholung gestartet. Der BREAK-Zustand wird automatisch zurückgesetzt, sobald die Verbindung in der Leitung wiederhergestellt ist.

Für alle erkannten Übertragungsfehler (verlorenes Zeichen, Rahmen-/Paritätsfehler) wird eine einheitliche Nummer gemeldet, gleichgültig, ob der Fehler beim Senden oder Empfangen eines Datenblocks erkannt wurde. Der Fehler wird aber nur gemeldet, wenn die Wiederholungen zuvor erfolglos verliefen.

## **2.8 Datenübertragung mit dem ASCII-Treiber**

### **2.8.1 Grundlagen zur Datenübertragung mit ASCII-Treiber**

#### **Einleitung**

Der ASCII-Treiber steuert die Datenübertragung bei einer Punkt-zu-Punkt-Kopplung zwischen der Baugruppe ET 200S 1SI und einem Kommunikationspartner. Der ASCII-Treiber beinhaltet die Bitübertragungsschicht (Schicht 1).

Der Aufbau der Telegramme wird dadurch offengehalten, dass der S7-Anwender das komplette Sendetelegramm an die Baugruppe ET 200S 1SI übergibt. Für die Empfangsrichtung ist das Endekriterium eines Telegramms zu parametrieren. Der Aufbau der Sendetelegramme kann sich vom Aufbau der Empfangstelegramme unterscheiden.

Mit dem ASCII-Treiber können Daten mit beliebigem Aufbau (alle abdruckbaren ASCII-Zeichen ebenso wie alle anderen Zeichen von 00 bis FF<sub>H</sub> (bei Zeichenrahmen mit 8 Datenbits) bzw. von 00 bis 7F<sub>H</sub> (bei Zeichenrahmen mit 7 Datenbits)) gesendet und empfangen werden.

#### **Siehe auch**

Grundlagen zur Kommunikation über Funktionsbausteine (Seite 65)

Anlaufeigenschaften und Betriebszustände (Seite 85)

## 2.8.2 Daten senden mit dem ASCII-Treiber

### Daten senden mit dem ASCII-Treiber

Beim Senden geben Sie die Anzahl der zu übertragenden Nutzdaten-Bytes beim Aufruf des Funktionsbausteins S\_SEND als Parameter LEN an. In den Nutzdaten müssen eventuell benötigte Start- und Endezeichen enthalten sein.

Wenn Sie beim Empfangen mit dem Endekriterium "Ablauf der Zeichenverzugszeit" arbeiten, hält der ASCII-Treiber auch beim Senden eine Pause zwischen zwei Telegrammen ein. Sie können den FB S\_SEND jederzeit aufrufen, der ASCII-Treiber beginnt mit der Ausgabe aber erst, wenn seit dem letzten gesendeten Telegramm eine Zeit größer als die parametrisierte Zeichenverzugszeit vergangen ist.

#### Hinweis

Bei der Parametrierung der Flusskontrolle XON/XOFF dürfen die Nutzdaten keines der parametrisierten XON- oder XOFF-Zeichen enthalten. Die Voreinstellungen sind DC1 = 11<sub>H</sub> für XON und DC3 = 13<sub>H</sub> für XOFF.

### Daten senden

Das folgende Bild zeigt die Abläufe beim Senden.

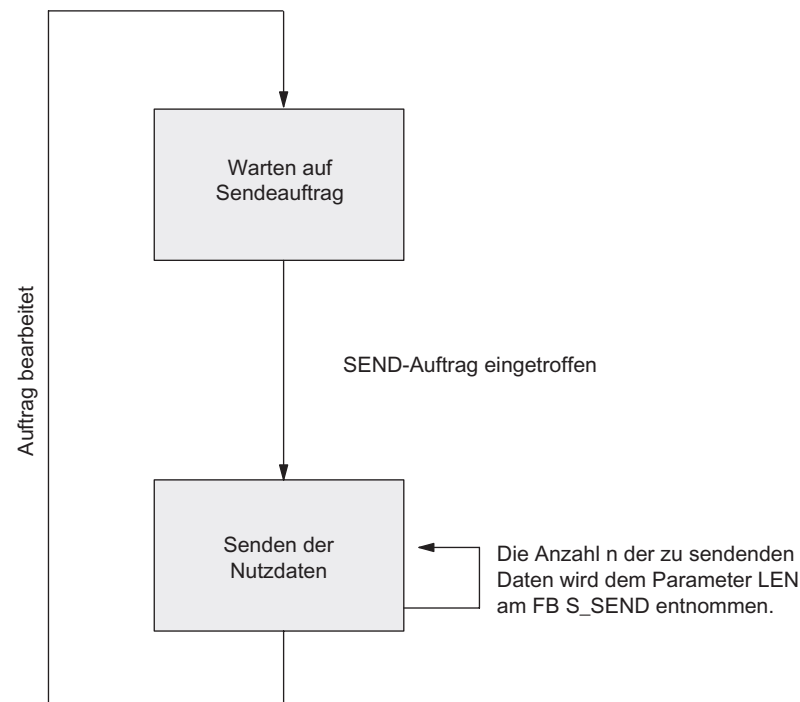


Bild 2-15 Ablaufschema beim Senden

### 2.8.3 Daten empfangen mit dem ASCII-Treiber

#### Daten empfangen mit dem ASCII-Treiber

Bei der Datenübertragung mit dem ASCII-Treiber können Sie zwischen drei verschiedenen Endekriterien wählen. Das Endekriterium legt fest, wann ein Telegramm vollständig empfangen wurde. Die einstellbaren Endekriterien sind:

- **Ablauf der Zeichenverzugszeit**  
Das Telegramm hat weder eine feste Länge noch definierte Endezeichen, das Ende eines Telegramms ist durch eine Pause in der Leitung (Ablauf der Zeichenverzugszeit) festgelegt. Die Mindestwerte der einzelnen Baudraten finden Sie weiter unten.
- **Empfang der/des Endezeichen(s)**  
Am Ende des Telegramms stehen ein oder zwei definierte Endezeichen.
- **Empfang einer festen Zeichenanzahl**  
Die Länge der Empfangstelegramme ist immer gleich.

#### Codetransparenz

Die Codetransparenz der Prozedur hängt von der Wahl des parametrisierten Endekriteriums und der Flusskontrolle ab:

- Mit ein oder zwei Endezeichen
  - nicht codetransparent
- Endekriterium Zeichenverzugszeit oder feste Telegrammlänge
  - codetransparent
- Bei Verwendung der Flusskontrolle XON/XOFF ist kein codetransparenter Betrieb möglich.

Codetransparent bedeutet, dass in den Nutzdaten alle beliebigen Zeichenkombinationen vorkommen dürfen, ohne dass das Endekriterium erkannt wird.

#### Mindestzeichenverzugszeit je nach Baudrate

Der Mindestwert für die Zeichenverzugszeit richtet sich nach der Baudrate. Die folgende Tabelle listet die Mindestzeichenverzugszeit in ms für die einzelnen Baudraten auf.

Tabelle 2- 7 Mindestzeichenverzugszeit

Baudrate	Mindestzeichenverzugszeit
115	365 ms
300	130 ms
600	65 ms
1.200	32 ms
2.400	16 ms
4.800	8 ms
9.600	4 ms

Baudrate	Mindestzeichenverzugszeit
19.200	2 ms
38.400	1 ms
57.600	1 ms
76.800	1 ms
115.200	1 ms

### Empfangspuffer der Baugruppe ET 200S

Der Empfangspuffer der Schnittstellenbaugruppe ET 200S 1SI umfasst 4096 Bytes. Bei der Parametrierung können Sie angeben, ob der Empfangspuffer im Anlauf gelöscht werden soll und ob ein Überschreiben von Daten im Empfangspuffer verhindert werden soll. Zusätzlich können Sie das Puffern der empfangenen Telegramme aktivieren oder sperren.

Beim Empfangspuffer der seriellen Schnittstellenbaugruppe ET 200S 1SI handelt es sich um einen Ringpuffer:

- Werden mehrere Telegramme in den Empfangspuffer der Baugruppe ET 200S 1SI eingetragen, gilt: es wird immer das älteste Telegramm der Baugruppe ET 200S 1SI an die CPU übertragen.
- Wenn Sie immer nur das neueste Telegramm zur CPU übertragen wollen, müssen Sie Dynamische Telegramme sperren **und** den Überschreibschutz ausschalten.

---

#### Hinweis

Wird das ständige Auslesen der Empfangsdaten im Anwenderprogramm für eine Zeit lang ausgesetzt, kann es beim erneuten Anfordern der Empfangsdaten dazu kommen, dass von der Baugruppe 200S 1SI erst ein altes und dann erst das neueste Telegramm von der CPU empfangen wird.

Das alte Telegramm ist das Telegramm, das bei der Unterbrechung auf dem Weg zwischen der ET 200S 1SI und CPU unterwegs war, bzw. vom FB bereits empfangen wurde.

---

### 2.8.4 Endekriterien für die Datenübertragung mit dem ASCII-Treiber

#### Endekriterium "Ablauf der Zeichenverzugszeit"

Beim Empfang von Daten wird das Telegrammende erkannt, wenn die Zeichenverzugszeit abgelaufen ist. Die empfangenen Daten werden mit dem Funktionsbaustein S\_RCV von der CPU übernommen.

Die Zeichenverzugszeit muss in diesem Fall so eingestellt werden, dass sie zwischen zwei aufeinanderfolgenden Telegrammen sicher abläuft. Sie sollte aber so groß sein, dass bei Sendepausen des Koppelpartners innerhalb eines Telegramms nicht fälschlicherweise das Telegrammende erkannt wird.

Das folgende Bild zeigt die Abläufe beim Empfangen mit dem Endekriterium "Ablauf der Zeichenverzugszeit".

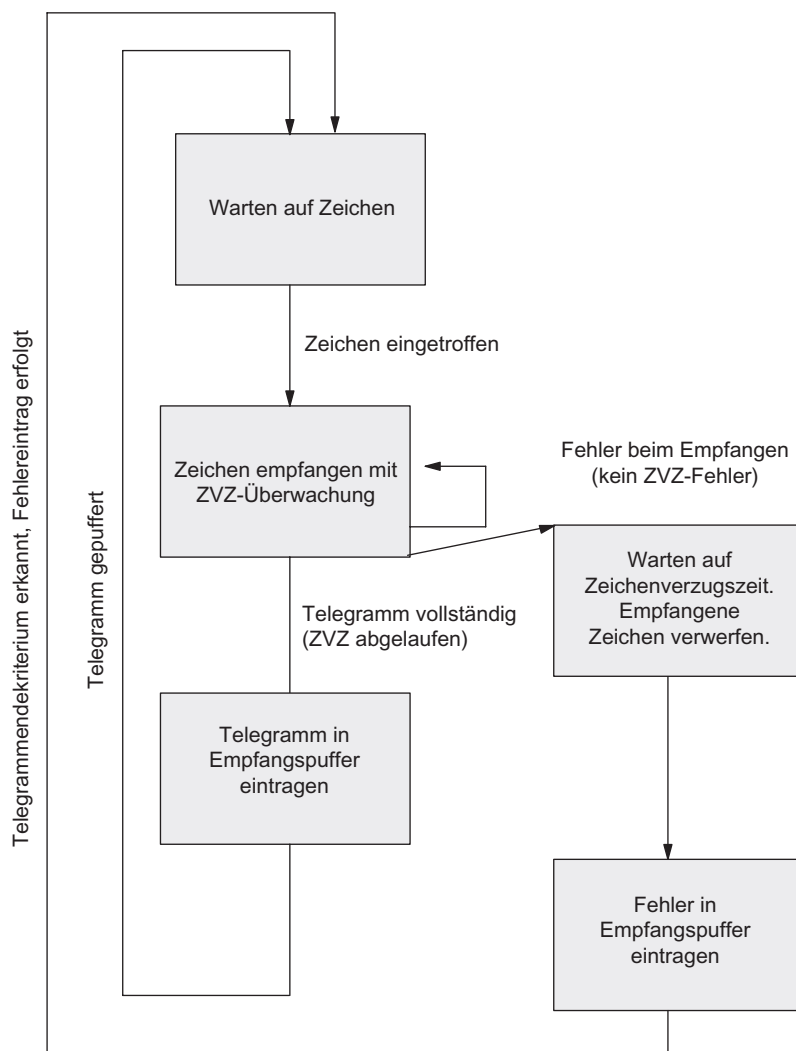


Bild 2-16 Ablaufschema beim Empfangen mit Endekriterium "Ablauf der Zeichenverzugszeit"

### Endekriterium "Endezeichen"

Beim Empfang von Daten wird das Telegrammende erkannt, wenn das/die parametrisierten Endezeichen empfangen werden. Die empfangenen Daten werden inklusive Endezeichen mit dem Funktionsbaustein S\_RCV von der CPU übernommen.

Der Ablauf der Zeichenverzugszeit während des Empfangs führt zur Beendigung des Empfangs. Es erfolgt eine Fehlermeldung, und das Telegrammfragment wird verworfen.

Wenn mit Endezeichen gearbeitet wird, ist die Übertragung nicht codetransparent, und es muss ausgeschlossen werden, dass die Endekennung/en in den Nutzdaten vorkommen.

Das folgende Bild zeigt die Abläufe beim Empfangen mit dem Endekriterium "Endezeichen".

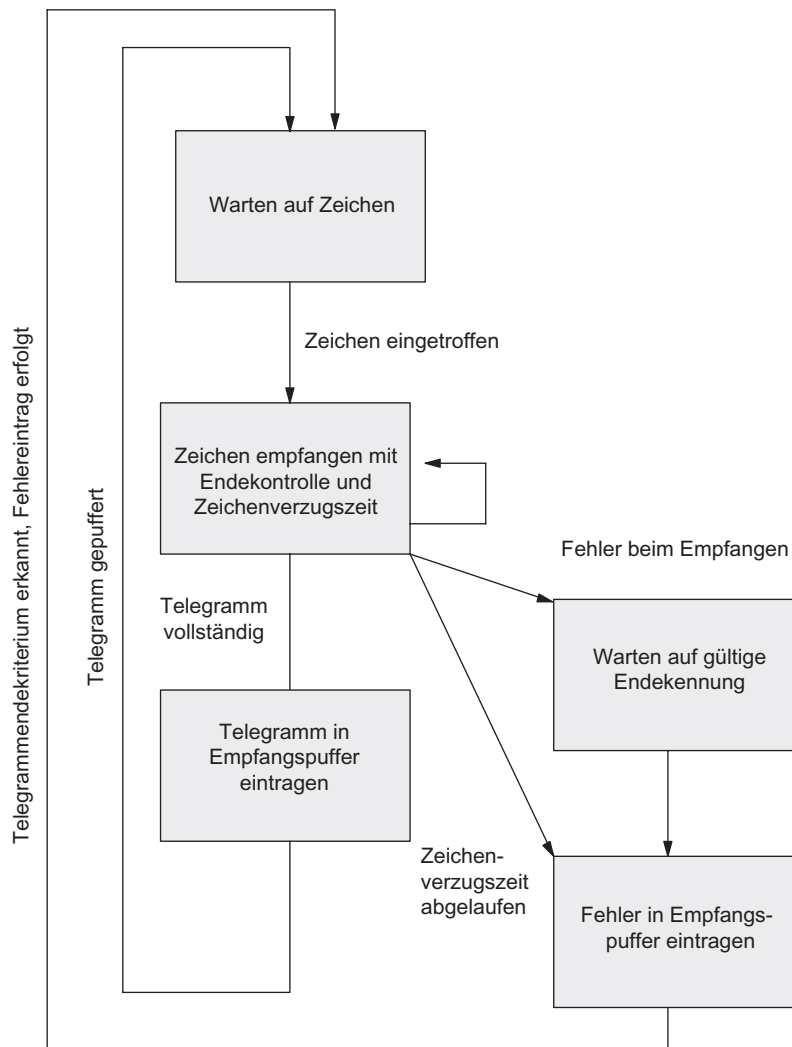


Bild 2-17 Ablaufschema beim Empfangen mit Endekriterium "Endezeichen"

### Endekriterium "Feste Telegrammlänge"

Beim Empfang von Daten wird das Telegrammende erkannt, wenn die parametrisierte Anzahl Zeichen empfangen wurde. Die empfangenen Daten werden mit dem Funktionsbaustein S\_RCV von der CPU übernommen.

Der Ablauf der Zeichenverzugszeit vor Erreichen der parametrisierten Zeichenanzahl führt zur Beendigung des Empfangs. Es erfolgt eine Fehlermeldung, und das Telegrammfragment wird verworfen.

Das folgende Bild zeigt die Abläufe beim Empfangen mit dem Endekriterium "feste Telegrammlänge".

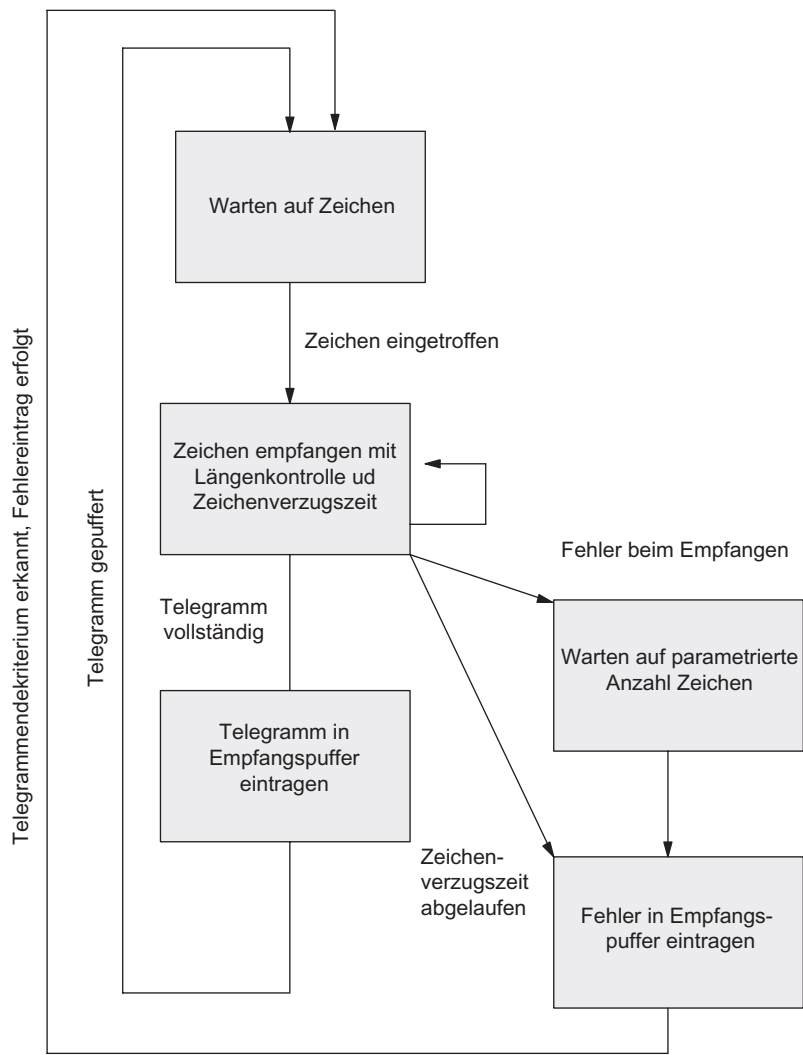


Bild 2-18 Ablaufschema beim Empfangen mit Endekriterium "feste Telegrammlänge"



## 2.8.5 RS-232C-Begleitsignale für die Datenübertragung mit dem ASCII-Treiber

### RS-232C-Begleitsignale

Die Baugruppe ET 200S 1SI unterstützt die folgenden RS-232C-Begleitsignale:

- DCD (Eingang) Data carrier detect; Datenträger erkannt
- DTR (Ausgang) Data terminal ready; ET 200S 1SI ist betriebsbereit
- DSR (Eingang) Data set ready; Kommunikationspartner betriebsbereit
- RTS (Ausgang) Request to send; ET 200S 1SI ist sendebereit
- CTS (Eingang) Clear to send; Kommunikationspartner kann Daten von der Baugruppe ET 200S 1SI empfangen (Antwort auf RTS = ON der ET 200S 1SI)

Nach dem Einschalten der Baugruppe ET 200S 1SI befinden sich die Ausgangssignale im Zustand OFF (inaktiv).

Die Bedienung der Steuersignale DTR/DSR und RTS/CTS können Sie mit der Parametrieroberfläche parametrieren oder über Funktionen (FCs) im Anwenderprogramm steuern.

### RS-232C-Begleitsignale bedienen

Die RS 232C-Begleitsignale können bedient werden:

- bei parametrierter automatischer Bedienung aller RS 232C-Begleitsignale
- bei parametrierter Datenflusskontrolle (RTS/CTS)
- über die Funktionsbausteine (FBs) S\_VSTAT und S\_VSET

---

#### **Hinweis**

Bei der Parametrierung einer automatischen Bedienung der RS 232C-Begleitsignale ist weder eine Datenflusskontrolle mit RTS/CTS noch ein Steuern von RTS und DTR über den FB S\_VSET möglich.

Bei der Parametrierung einer Datenflusskontrolle mit RTS/CTS ist ein Steuern von RTS über den FB S\_VSET nicht möglich.

Das Lesen aller RS 232C-Begleitsignale über den FB S\_VSTAT ist hingegen immer möglich.

---

Die folgenden Abschnitte erläutern das prinzipielle Handling bei der Steuerung und Auswertung der RS-232C-Begleitsignale.

### Automatische Bedienung der RS-232C-Begleitsignale

Die automatische Bedienung der RS-232C-Begleitsignale in der Baugruppe ET 200S 1SI ist folgendermaßen implementiert:

- Sobald die Baugruppe ET 200S 1SI durch Parametrierung in eine Betriebsart mit automatischer Bedienung der RS-232C-Begleitsignale gebracht wurde, setzt sie die Leitungen RTS auf OFF und DTR auf ON (ET 200S 1SI betriebsbereit).

Das Senden und Empfangen von Telegrammen ist erst möglich, nachdem die Leitung DTR auf ON gesetzt wurde. Solange DTR auf OFF gesetzt bleibt, werden keine Daten über die RS-232C-Schnittstelle empfangen. Ein Sendeauftrag wird mit einer entsprechender Fehlermeldung abgebrochen.

- Steht ein **Sendeauftrag** an, wird RTS auf ON gesetzt und die parametrierte Datenausgabewartezeit gestartet. Nach Ablauf der Datenausgabezeit und CTS = ON werden die Daten über die RS 232C-Schnittstelle gesendet.
- Wird beim Senden innerhalb der Datenausgabewartezeit die Leitung CTS nicht auf ON gesetzt, oder erfolgt innerhalb des Sendevorgangs ein Wechsel von CTS auf OFF, wird der Sendeauftrag abgebrochen und eine entsprechende Fehlermeldung generiert.
- Nach dem Senden der Daten wird nach Ablauf der parametrierten RTS-Wegnahmezeit die Leitung RTS auf OFF gesetzt. Die Baugruppe ET 200S 1SI wartet nicht auf den Wechsel von CTS auf OFF.
- Ein **Empfang** von Daten über die RS-232C-Schnittstelle ist möglich, sobald die Leitung DSR auf ON gesetzt ist. Droht der Empfangspuffer der Baugruppe ET 200S 1SI überzulaufen, erfolgt keine Reaktion der Baugruppe ET 200S SI.
- Bei einem Wechsel von DSR = ON auf OFF wird sowohl ein laufender Sendeauftrag als auch das Empfangen von Daten mit einer Fehlermeldung abgebrochen.

---

#### Hinweis

Bei der Parametrierung einer automatischen Bedienung der RS 232C-Begleitsignale ist weder eine Datenflusskontrolle mit RTS/CTS noch ein Steuern von RTS und DTR über den FB S\_VSET möglich.

---

## Zeitdiagramm

Das folgende Bild zeigt den zeitlichen Ablauf eines Sendeauftrags.

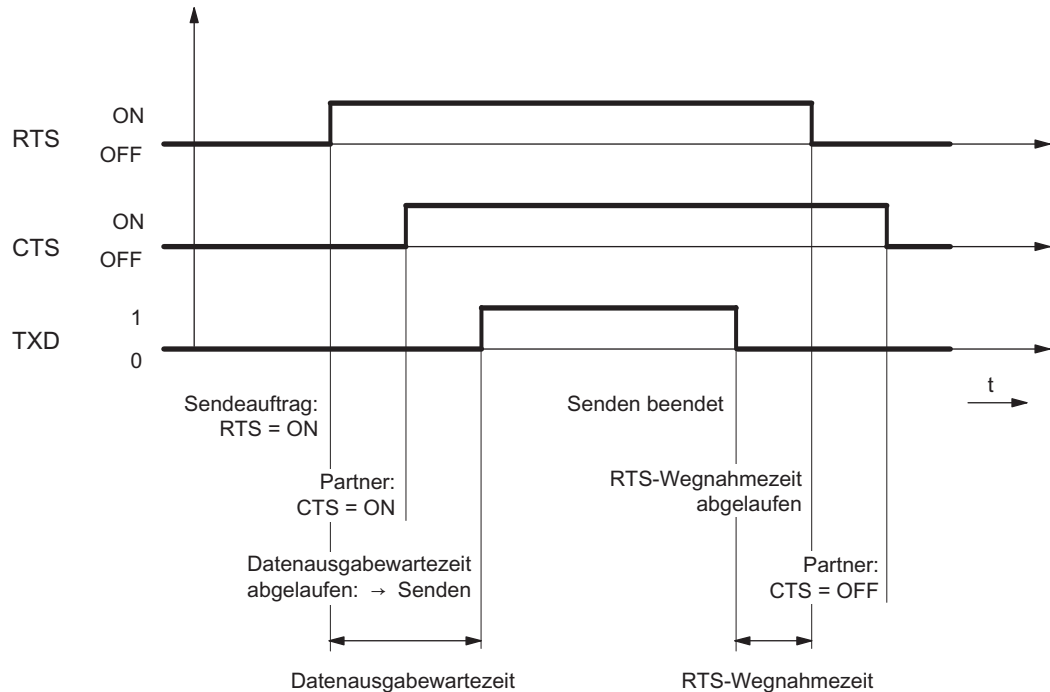


Bild 2-19 Zeitdiagramm bei automatischer Bedienung der RS-232C-Begleitsignale

## Datenflusskontrolle/Handshakeverfahren

Handshakeverfahren steuern den Datenfluss zwischen zwei Kommunikationspartnern. Durch die Verwendung von Handshakeverfahren wird vermieden, dass bei unterschiedlich schnell arbeitenden Geräten Daten bei der Übertragung verloren gehen. Grundsätzlich lassen sich die folgenden zwei Verfahren unterscheiden:

- Softwarehandshake (z.B. XON/XOFF)
- Hardwarehandshake (z.B. RTS/CTS)

Die Datenflusskontrolle der Baugruppe ET 200S 1SI ist folgendermaßen implementiert:

- Sobald die Baugruppe ET 200S 1SI durch Parametrierung in eine Betriebsart mit Flusskontrolle gebracht wurde, sendet sie das Zeichen XON bzw. setzt die Leitung RTS auf ON.
- Bei Erreichen der parametrierten Telegrammanzahl bzw. 50 Zeichen bevor der Empfangspuffer überläuft (Größe des Empfangspuffers: 4096 Byte), sendet die Baugruppe ET 200S 1SI das Zeichen XOFF bzw. setzt die Leitung RTS auf OFF. Sendet der Kommunikationspartner trotzdem weiter, wird bei Überlauf des Empfangspuffers eine Fehlermeldung generiert. Die empfangenen Daten des letzten Telegramms werden verworfen.

- Sobald ein Telegramm durch die S7-CPU abgeholt wurde und der Empfangspuffer aufnahmebereit ist, sendet die Baugruppe ET 200S 1SI das Zeichen XON bzw. setzt die Leitung RTS auf ON.
- Empfängt die Baugruppe ET 200S SI das Zeichen XOFF bzw. wird das Steuersignal CTS auf OFF gesetzt, unterbricht die Baugruppe ET 200S1 SI den Sendevorgang. Wird nach einer bestimmten parametrierbaren Zeit kein XON empfangen bzw. CTS nicht auf ON gesetzt, wird der Sendevorgang abgebrochen und eine entsprechende Fehlermeldung (0708<sub>H</sub>) am Ausgang STATUS der Funktionsbausteine generiert.

### **Lesen/Steuern über FB S\_VSTAT und FB S\_VSET**

Mittels des Funktionsbausteins S\_VSTAT kann der Zustand von jedem RS-232C-Begleitsignal ermittelt werden. Mit dem Funktionsbaustein S\_VSET ist ein Steuern der Ausgangssignale DTR und RTS möglich. Im Abschnitt Grundlagen zur Kommunikation über Funktionsbausteine (Seite 65) finden Sie Informationen dazu, wie die Funktionsbausteine als Schnittstelle zwischen der CPU und der Baugruppe ET 200S 1SI einsetzen.

## 2.9 Konfigurieren und Parametrieren der seriellen Schnittstellenbaugruppe

### 2.9.1 Konfigurieren der seriellen Schnittstellenbaugruppe

#### Prinzip

Wenn Sie mit einem S7-Master mit der Schnittstellenbaugruppe ET 200S 1SI über ein PROFIBUS-Netzwerk kommunizieren, dann arbeiten Sie in der Hardware-Konfiguration in STEP 7, um die Baugruppe im PROFIBUS-Netzwerk einzurichten und die Kommunikationsparameter der Baugruppe einzustellen.

Wenn Sie die Baugruppe ET 200S 1SI im Hardware-Katalog wählen und sie in die ET 200S Basis in der Konfiguration des Netzwerks einfügen, werden die Bestellnummer der Baugruppe, die Nummer des Steckplatzes sowie die Adressen von Ein- und Ausgängen automatisch in die Konfigurationstabelle aufgenommen. Sie können dann den Eigenschaftsdialog der Baugruppe ET 200S 1SI aufrufen und dort die Kommunikationsart und andere Parameter einstellen.

### 2.9.2 Parametrieren des ASCII-Treibers

#### Prinzip

Die folgende Tabelle führt die Parameter auf, die für den ASCII-Treiber der seriellen Schnittstellenbaugruppe eingestellt werden können.

Tabelle 2- 8 Parameter für den ASCII-Treiber

Parameter	Beschreibung	Wertebereich	Defaultwert
Diagnosealarm	Geben Sie an, ob die Baugruppe einen Diagnosealarm erzeugt, wenn ein schwerer Fehler auftritt.	<ul style="list-style-type: none"> <li>• Nein</li> <li>• Ja</li> </ul>	Nein
BREAK Erkennung aktivieren	Bei Leitungsbruch bzw. wenn kein Schnittstellenkabel angeschlossen ist erzeugt die Baugruppe die Fehlermeldung "Break".	<ul style="list-style-type: none"> <li>• Nein</li> <li>• Ja</li> </ul>	Nein
Art der Schnittstelle	Geben Sie die zu verwendende elektrische Schnittstelle an (siehe Abschnitt RS-232C-Schnittstelle (Seite 22) und RS-422/485-Schnittstelle (Seite 23) .	<ul style="list-style-type: none"> <li>• RS-232C</li> <li>• RS-422 (voll duplex)</li> <li>• RS-485 (halbduplex)</li> </ul>	RS-232C
Halb- und voll duplex Vorbelegung der Empfangsleitung	Geben Sie die Vorbelegung der Empfangsleitung in den Betriebsarten RS-422 und RS-485 an. Nicht in der Betriebsart RS-232C.  Die Einstellung "Invertierte Pegel" wird nur im Ersatzteilfall zur Sicherstellung der Kompatibilität benötigt.	RS422: R(A) 5V / R(B) 0V (BREAK) R(A) 0V / R(B) 5V Invertierte Pegel  RS485: Keine R(A) 0V / R(B) 5V	RS422: R(A) 5V / R(B) 0V (BREAK)  RS 485: R(A) 0V / R(B) 5V

Parameter	Beschreibung	Wertebereich	Defaultwert
Datenflusskontrolle (mit voreingestellten Parametern; voreingestellte Werte im Anwenderprogramm ändern)	Sie können Daten mit Datenflusskontrolle senden und empfangen. Durch die Datenflusskontrolle wird die Datenübertragung synchronisiert, wenn ein Kommunikationspartner schneller arbeitet als der andere. Wählen Sie die Art der Datenflusskontrolle und stellen Sie die zugehörigen Parameter ein (siehe Abschnitt Grundlagen zur Datenübertragung mit ASCII-Treiber (Seite 42) ).  Hinweis: Bei der RS 485-Schnittstelle ist keine Datenflusskontrolle möglich. Datenflusskontrolle mit "RTS/CTS" und "Automatische Bedienung der V24-Signale" ist nur bei der RS-232C-Schnittstelle möglich.	<ul style="list-style-type: none"> <li>• Keine</li> <li>• XON/OFF</li> <li>• RTS/CTS</li> <li>• Automatische Bedienung der V.24-Signale</li> </ul>	Keine
Baudrate	Wählen Sie die Geschwindigkeit der Datenübertragung in Bits pro Sekunde.	<ul style="list-style-type: none"> <li>• 110</li> <li>• 300</li> <li>• 600</li> <li>• 1200</li> <li>• 2400</li> <li>• 4800</li> <li>• 9600</li> <li>• 19200</li> <li>• 38400</li> <li>• 57600</li> <li>• 76800</li> <li>• 115200</li> </ul>	9600
Datenbits	Wählen Sie die Anzahl Bits, auf die ein Zeichen abgebildet wird.	<ul style="list-style-type: none"> <li>• 7</li> <li>• 8</li> </ul>	8
Stoppbits	Wählen Sie die Anzahl Stoppbits, die bei der Übertragung jedem Zeichen nachgesetzt werden und das Ende eines Zeichens kennzeichnen.	<ul style="list-style-type: none"> <li>• 1</li> <li>• 2</li> </ul>	1

2.9 Konfigurieren und Parametrieren der seriellen Schnittstellenbaugruppe

Parameter	Beschreibung	Wertebereich	Defaultwert
Parität	<p>Die Reihenfolge der Datenbits kann um ein Zeichen erweitert werden, so dass das Paritätsbit aufgenommen wird. Der zusätzliche Wert (0 oder 1) versetzt den Wert aller Bits (Datenbits und Paritätsbit) in einen definierten Zustand.</p> <p><b>Keine:</b> Daten werden ohne Paritätsbit gesendet.</p> <p><b>Ungerade:</b> Das Paritätsbit wird gesetzt, so dass die Gesamtzahl der Datenbits (einschließlich Paritätsbit) mit dem Signalzustand "1" ungerade ist.</p> <p><b>Gerade:</b> Das Paritätsbit wird gesetzt, so dass die Gesamtzahl der Datenbits (einschließlich Paritätsbit) mit dem Signalzustand "1" gerade ist.</p> <p><b>Beliebige:</b> Der Signalzustand des Paritätsbits hat keine Bedeutung. Die Parität wird beim Empfangen von Daten nicht geprüft, doch wird immer auf "0" gesetzt, wenn Daten gesendet werden.</p>	<ul style="list-style-type: none"> <li>• Keine</li> <li>• Ungerade</li> <li>• Gerade</li> <li>• Beliebig</li> </ul>	Gerade
Anzeige für Ende des Empfangstelegramms	<p>Wenn Daten mittels ASCII-Treiber übertragen werden, kann das Ende des Empfangstelegramms auf drei verschiedene Arten erkannt werden. Hier können Sie eine der drei Übertragungsarten wählen und die spezifischen Parameter angeben.</p> <p><b>Hinweis:</b> Läuft die Zeichenverzugszeit ab, während Daten empfangen werden, wird der Empfang in allen drei Betriebsarten vorzeitig abgebrochen. Das Telegramm wird außer in der Betriebsart "Ablauf der Zeichenverzugszeit" verworfen.</p> <ul style="list-style-type: none"> <li>• Ablauf der Zeichenverzugszeit: Das Telegrammende wird erkannt, wenn die parametrierte Zeichenverzugszeit abgelaufen ist.</li> <li>• Empfang der/des Endezeichen(s): Das Telegrammende wird erkannt, wenn das/die definierten Endezeichen empfangen werden.</li> <li>• Empfang einer festen Zeichenanzahl: Das Telegrammende wird anhand der parametrierten Telegrammlänge erkannt. Alle Telegramme, die empfangen werden sollen, haben die gleiche Länge.</li> </ul>	<ul style="list-style-type: none"> <li>• Ablauf der Zeichenverzugszeit</li> <li>• Empfang der/des Endezeichen(s)</li> <li>• Nach Empfang einer festen Zeichenanzahl</li> </ul>	Ablauf der Zeichenverzugszeit
Ablauf der Zeichenverzugszeit, ms	Die maximale Zeitspanne, die zwischen dem Empfang von zwei Zeichen vergehen darf. 1	1 bis 65535 ms	4 ms

Parameter	Beschreibung	Wertebereich	Defaultwert
Endezeichen 1 <sup>2</sup>	Zum Empfangen von Daten mit Endezeichen können Sie maximal zwei Endezeichen definieren. Die gewählten Endezeichen begrenzen die Länge des Telegramms.	<ul style="list-style-type: none"> <li>• Bei 7 Datenbits:<sup>3</sup> 1 bis 7FH</li> <li>• Bei 8 Datenbits:<sup>3</sup> 1 bis FFH</li> </ul>	3
Endezeichen 2 <sup>2</sup>	Zum Empfangen von Daten mit Endezeichen können Sie maximal zwei Endezeichen definieren. Die gewählten Endezeichen begrenzen die Länge des Telegramms. Code der zweiten Endekennung, sofern angewählt.	<ul style="list-style-type: none"> <li>• Bei 7 Datenbits:<sup>3</sup> 0 bis 7FH</li> <li>• Bei 8 Datenbits:<sup>3</sup> 0 bis FFH</li> </ul>	0
Telegrammlänge bei Empfang <sup>4</sup>	Geben Sie die Telegrammlänge an, wenn Daten mit einer festen Anzahl Zeichen empfangen werden sollen. Die Telegrammlänge muss genau der Anzahl Datenbytes entsprechen, die vom Kommunikationspartner empfangen werden sollen.	1 bis 224 Bytes	100
Dynamische Telegramme	Zum Empfangen von Meldungen können Sie angeben, ob nur eine Meldung gepuffert werden soll, oder ob die Meldungen dynamisch gepuffert werden sollen. Durch Aktivieren von dynamischen Telegrammen kann die Baugruppe mehrere Meldungen mit unterschiedlichen Längen puffern. Bei dem Puffer handelt es sich um einen Ringpuffer. Ist der Puffer voll, wird die älteste Meldung überschrieben, es sei denn, der Parameter "Überschreiben des Puffers verhindern" ist aktiviert. In diesem Fall wird die neueste Meldung verworfen. In beiden Fällen zeigt ein Diagnosealarm an, dass Daten verloren gegangen sind.	<ul style="list-style-type: none"> <li>• Aktiviert</li> <li>• Gesperrt</li> </ul>	Aktiviert
Überschreiben des Puffers verhindern	Durch diesen Parameter verhindern Sie, dass gepufferte Telegramme überschrieben werden, wenn die Baugruppe ein neues Telegramm empfängt, doch der Empfangspuffer noch nicht gelöscht ist. Es wird auf diese Weise verhindert, dass alte empfangene Telegramme verloren gehen.	<ul style="list-style-type: none"> <li>• Nein</li> <li>• Ja</li> </ul>	Ja



Parameter	Beschreibung	Wertebereich	Defaultwert
ET 200S 1SI- Empfangspuffer beim Anlauf löschen	Geben Sie an, ob der Empfangspuffer der Baugruppe automatisch gelöscht werden soll, wenn die CPU vom Betriebszustand STOP nach RUN wechselt (CPU-Anlauf). Sie können auf diese Weise sicherstellen, dass der Empfangspuffer der Baugruppe nur Telegramme enthält, die nach dem Anlauf der CPU empfangen wurden.	<ul style="list-style-type: none"> <li>• Nein</li> <li>• Ja</li> </ul>	Ja
<p><sup>1</sup> Die kürzeste Zeichenverzugszeit richtet sich nach der Baudrate.  <sup>2</sup> Nur bei Endkriterium Endezeichen einstellbar.  <sup>3</sup> Je nachdem, ob Sie für den Zeichenrahmen 7 oder 8 Datenbits parametrieren.  <sup>4</sup> Nur bei Endkriterium feste Telegrammlänge einstellbar.</p>			

### 2.9.3 Parametrieren der Treiber für das Protokoll 3964(R)

#### Prinzip

Die folgende Tabelle führt die Parameter auf, die für das Protokoll 3964(R) der seriellen Schnittstellenbaugruppe eingestellt werden können.

Tabelle 2- 9 Treiberparameter für das Protokoll 3964(R)

Parameter	Beschreibung	Wertebereich	Defaultwert
Diagnosealarm	Geben Sie an, ob die Baugruppe einen Diagnosealarm erzeugt, wenn ein schwerer Fehler auftritt.	<ul style="list-style-type: none"> <li>• Nein</li> <li>• Ja</li> </ul>	Nein
BREAK Erkennung aktivieren	Bei Leitungsbruch bzw. wenn kein Schnittstellenkabel angeschlossen ist erzeugt die Baugruppe die Fehlermeldung "Break".	<ul style="list-style-type: none"> <li>• Nein</li> <li>• Ja</li> </ul>	Nein
Art der Schnittstelle	Geben Sie die zu verwendende elektrische Schnittstelle an.	<ul style="list-style-type: none"> <li>• RS-232C</li> <li>• RS-422</li> </ul>	RS-232C
Vorbelegung der Empfangsleitung	Geben Sie die Vorbelegung der Empfangsleitung in der Betriebsart RS-422 an. Nicht in der Betriebsart RS-232C. Die Einstellung "Invertierte Pegel" wird nur im Ersatzteillfall zur Sicherstellung der Kompatibilität benötigt.	R(A) 5V / R(B) 0V (BREAK) R(A) 0V / R(B) 5V Invertierte Pegel	R(A) 5V / R(B) 0V (BREAK)

Parameter	Beschreibung	Wertebereich	Defaultwert
Betriebsart des Protokolls	<p>Geben Sie an, ob die Daten mit Blockprüfzeichen (BCC) gesendet werden sollen, um die Datensicherheit zu erhöhen.</p> <p>Das Blockprüfzeichen ist die gerade Längsparität (EXOR-Verknüpfung aller Datenbytes) eines gesendeten bzw. empfangenen Blocks. Erkennt ein Kommunikationspartner ein Blockprüfzeichen beim Empfangen der Daten, vergleicht er das BCC mit der intern berechneten Längsparität. Ist das Blockprüfzeichen fehlerhaft, wird 4 Sekunden gewartet (Blockwartezeit) und dann die Datenübertragung wiederholt.</p> <p>Kann der Datenblock nach der parametrierten Anzahl von Übertragungsversuchen nicht empfangen werden oder wird kein weiterer Versuch während der Blockwartezeit unternommen, wird der Empfang abgebrochen.</p>	<ul style="list-style-type: none"> <li>• keine Blockprüfung</li> <li>• Blockprüfung</li> </ul>	Blockprüfung
Baudrate	Wählen Sie die Geschwindigkeit der Datenübertragung in Bits pro Sekunde.	<ul style="list-style-type: none"> <li>• 110</li> <li>• 300</li> <li>• 600</li> <li>• 1200</li> <li>• 2400</li> <li>• 4800</li> <li>• 9600</li> <li>• 19200</li> <li>• 38400</li> <li>• 57600</li> <li>• 76800</li> <li>• 115200</li> </ul>	9600
Datenbits	Wählen Sie die Anzahl Bits, auf die ein Zeichen abgebildet wird.	<ul style="list-style-type: none"> <li>• 7</li> <li>• 8</li> </ul>	8
Stoppbits	Wählen Sie die Anzahl Stoppbits, die bei der Übertragung jedem Zeichen nachgesetzt werden und das Ende eines Zeichens kennzeichnen.	<ul style="list-style-type: none"> <li>• 1</li> <li>• 2</li> </ul>	1

2.9 Konfigurieren und Parametrieren der seriellen Schnittstellenbaugruppe

Parameter	Beschreibung	Wertebereich	Defaultwert
Parität	<p>Die Reihenfolge der Datenbits kann um ein Zeichen erweitert werden, so dass das Paritätsbit aufgenommen wird. Der zusätzliche Wert (0 oder 1) versetzt den Wert aller Bits (Datenbits und Paritätsbit) in einen definierten Zustand.</p> <ul style="list-style-type: none"> <li>Keine: Daten werden ohne Paritätsbit gesendet.</li> <li>Ungerade: Das Paritätsbit wird gesetzt, so dass die Gesamtzahl der Datenbits (einschließlich Paritätsbit) mit dem Signalzustand "1" ungerade ist.</li> <li>Gerade: Das Paritätsbit wird gesetzt, so dass die Gesamtzahl der Datenbits (einschließlich Paritätsbit) mit dem Signalzustand "1" gerade ist.</li> <li>Beliebig: Der Signalzustand des Paritätsbits hat keine Bedeutung. Die Parität wird beim Empfangen von Daten nicht geprüft, doch wird immer auf "0" gesetzt, wenn Daten gesendet werden.</li> </ul>	<ul style="list-style-type: none"> <li>keine</li> <li>ungerade</li> <li>gerade</li> <li>beliebig</li> </ul>	gerade
Zeichenverzugszeit (ms)	<p>Die maximale Zeitspanne, die zwischen dem Empfang von zwei Zeichen vergehen darf. Stellen Sie die für Ihre Anwendung kürzeste Zeichenverzugszeit ein. Beachten Sie, dass die Zeichenverzugszeit je nach Baudrate einen bestimmten Mindestwert haben muss.</p>	20 bis 655350 ms in Schritten von 10 ms	220 ms
Quittungsverzugszeit (ms)	<p>Geben Sie die maximale Zeit ein, die ablaufen darf, bevor beim Auf- und Abbau der Verbindung eine Quittung vom Kommunikationspartner empfangen wird. Beachten Sie, dass die Quittungsverzugszeit je nach Baudrate einen bestimmten Mindestwert haben muss.</p>	10 bis 655350 ms in Schritten von 10 ms	2000 ms (550 ms ohne Blockprüfung)
Aufbauversuche	<p>Geben Sie die Anzahl (<math>n</math>) Versuche für den Aufbau einer Verbindung an. (Nach <math>n</math> fehlgeschlagenen Versuchen wird die Funktion abgebrochen und der Fehler im Ausgang STATUS des Funktionsbausteins S_SEND angezeigt.)</p>	1 bis 255	6
Übertragungsversuche	<p>Geben Sie die Anzahl (<math>n</math>) Versuche für die Übertragung eines Telegramms an. (Nach <math>n</math> fehlgeschlagenen Versuchen, das Telegramm fehlerfrei zu senden, wird die Funktion abgebrochen und der Fehler wird im Ausgang STATUS des Funktionsbausteins S_SEND angezeigt.)</p> <p>Mögliche Ursachen für den Abbruch:</p> <ul style="list-style-type: none"> <li>Paritätsfehler</li> <li>BBC-Fehler; Paritätsfehler</li> <li>Unterschiedliche Parametrierung der Kommunikationspartner (z.B. Baudrate, Parität, Zeichenrahmen, Blockprüfzeichen, unterschiedliche Protokolle)</li> </ul>	1 bis 255	6

Parameter	Beschreibung	Wertebereich	Defaultwert
Priorität	Wenn beide Kommunikationspartner gleichzeitig einen Sendeauftrag erteilen, dann stellt der Partner mit niedriger Priorität zunächst seinen Sendeauftrag zurück. Für die Datenübertragung müssen Sie dem einen Kommunikationspartner eine höhere und dem anderen Kommunikationspartner eine niedrigere Priorität zuordnen.	<ul style="list-style-type: none"> <li>• Hoch</li> <li>• Niedrig</li> </ul>	Niedrig
ET 200S 1SI- Empfangspuffer beim Anlauf löschen	Geben Sie an, ob der Empfangspuffer der Baugruppe automatisch gelöscht werden soll, wenn die CPU vom Betriebszustand STOP nach RUN wechselt (CPU-Anlauf). Sie können auf diese Weise sicherstellen, dass der Empfangspuffer der Baugruppe nur Telegramme enthält, die nach dem Anlauf der CPU empfangen wurden.	<ul style="list-style-type: none"> <li>• Nein</li> <li>• Ja</li> </ul>	Ja

## 2.9.4 Identifikationsdaten

### Definition

Identifikationsdaten sind in einer Baugruppe gespeicherte Informationen, die Sie unterstützen beim

- Beheben von Fehlern in einer Anlage
- Überprüfen der Anlagenkonfiguration
- Auffinden von Hardware-Änderungen einer Anlage.

Mit den Identifikationsdaten können Baugruppen online eindeutig identifiziert werden. Ab der MLFB-Nr. 6ES7 138-4DFx1-0AE0 sind diese Daten auf den ET 200S 1SI Modulen verfügbar.

Sie können die Identifikationsdaten über Zielsystem > Baugruppenzustand oder– wie im folgenden beschrieben – mit “Datensatz lesen”, anzeigen.

### Lesen der Identifikationsdaten

Über Datensatz lesen kann der Anwender gezielt auf bestimmte Identifikationsdaten zugreifen.

Unter der zugehörigen Datensatznummer ist der dem jeweiligen Index zugeordnete Teil der Identifikationsdaten zu finden.

- Alle Datensätze mit Identifikationsdaten haben eine Länge von 64 Byte.
- Die Datensätze sind nach dem in folgender Tabelle dargestellten Prinzip aufgebaut.

Tabelle 2- 10 Prinzipaufbau der Datensätze mit Identifikationsdaten

Inhalt	Länge (Byte)	Codierung (hex)
<b>Kopfinformation</b>		
SZL-ID	2	F1 11
Index	2	00 0x
Länge der Identifikationsdaten	2	00 38
Anzahl der Blöcke mit Identifikationsdaten	2	00 01
<b>Identifikationsdaten</b>		
Index	2	00 0x
Identifikationsdaten zum jeweiligen Index (siehe folgende Tabelle)	54	

**Identifikationsdaten der Baugruppe ET 200S 1SI**

Tabelle 2- 11 Identifikationsdaten der Baugruppe ET 200S 1SI

Identifikationsdaten	Zugriff	Voreinstellung	Erläuterung
<b>Index 1 (Datensatz 231/ nur Lesen)</b>			
Hersteller	lesen (2 Byte)	00 2A hex (=42 dez)	Hier ist der Name des Herstellers gespeichert. (42 dez = Siemens AG)
Gerätebezeichnung	lesen (20 Byte)	6ES7 138-4DFx1-0AB0	Bestellnummer der Baugruppe x = 0 (ASCII/3964@), 1 (MODBUS/USS)
Geräte-Seriennummer	lesen (16 Byte)	Hier ist die Seriennummer der Baugruppe gespeichert. Damit ist eine eindeutige Identifikation der Baugruppe möglich.	
Hardware-Revision	lesen (2 Byte)	Gibt Auskunft über den Erzeugnisstand der Baugruppe.	
Software-Revision	lesen (4 Byte)	Gibt Auskunft über die Firmware-Version der Baugruppe.	
Statistische Revisions-Nr.	lesen (2 Byte)	–	Wird nicht unterstützt
Profile_ID	lesen (2 Byte)	F6 00 hex	Interner Parameter (gemäß PROFIBUS DP)
Profile-specific type	lesen (2 Byte)	00 04 hex (= 4 dez)	Interner Parameter (Communication Module, gemäß PROFIBUS DP)
I&M Version	lesen (2 Byte)	00 00 hex (= 0 dez)	Interner Parameter (gemäß PROFIBUS DP)
I&M supported	lesen (2 Byte)	00 01 hex (= 1 dez)	Interner Parameter (I&M0 und I&M1, gemäß PROFIBUS DP)
<b>Index 2 (Datensatz 232/ Lesen und Schreiben)</b>			
AKZ	lesen/schreiben (max. 32 Zeichen)	–	Anlagenkennzeichen der Baugruppe
OKZ	lesen/schreiben (max. 22 Zeichen)	–	Ortskennzeichen der Baugruppe

## 2.9.5 Nachladen von Firmware-Updates

### Beschreibung

Zur Funktionserweiterung und Fehlerbehebung ist es möglich, Firmware-Updates in den Betriebssystemspeicher des ET 200S 1SI zu laden.

Das Nachladen von Firmware-Updates erfolgt über Hardware Konfig.

### Basis-Firmware

Das ET 200S 1SI wird mit einer Basis-Firmware ausgeliefert.

### Voraussetzung

Für das Nachladen von Firmware-Updates müssen folgende Voraussetzungen erfüllt sein:

- Das ET 200S 1SI muss vom PG/PC aus online erreichbar sein.
- Die Dateien mit der neuen Version der Firmware müssen im Dateisystem Ihres PG/PC zur Verfügung stehen.

### Firmware laden

Gehen Sie wie folgt vor für ein Firmware-Update (nur nutzbar, wenn die IM 151 diese Funktion unterstützt):

1. Öffnen Sie HW Konfig und markieren Sie die gewünschte ET 200S 1SI-Baugruppe.
2. Wählen Sie den Menübefehl Zielsystem > Firmware aktualisieren.

Der weitere Ablauf ist in der Online-Hilfe zu STEP 7 beschrieben.

---

#### Hinweis

Zum Laden der Firmware-Datei für die Baugruppe ET 200S 1SI müssen Sie die CPU in STOP schalten.

---

Ist das Update erfolgreich, erscheint eine Bestätigungsmeldung und die neue Firmware ist sofort aktiviert.

Nach erfolgreichem Update ist der bisherige Stand der Firmware des ET 200S 1SI durch einen Aufkleber mit dem aktualisierten Stand der Firmware zu überkleben.

### Update nicht erfolgreich

Wenn das Update misslingt, dann blinkt die rote SF-LED auf der Baugruppe. Wiederholen Sie das Update. Lässt sich das Update nicht erfolgreich durchführen, so wenden Sie sich bitte an Ihren zuständigen Siemens-Ansprechpartner.

### LED-Anzeigen

Tabelle 2- 12 LED-Anzeigen während des Ladens eines Firmware-Updates

Zustand	SF	TXD	RXD	Bemerkung	Abhilfe
Firmware-Update läuft	an	an	an	-	-
Firmware-Update fertig	an	aus	aus	-	-
ET 200S 1SI ohne Baugruppen-Firmware	blinkt (2Hz)	aus	aus	Baugruppen-Firmware gelöscht, Firmware-Update wurde abgebrochen, Firmware-Update weiterhin möglich	Firmware neu laden
Hardware-Fehler beim Firmware-Update	blinkt (2Hz)	blinkt (2Hz)	blinkt (2Hz)	Löschen/Schreiben fehlgeschlagen	Versorgungsspannung der Baugruppe aus- und wieder einschalten und Firmware neu laden. Prüfen, ob Baugruppe defekt ist.

### Hardware- und Firmwareausgabestand anzeigen

Der aktuelle Hardware- und Firmwareausgabestand auf dem ET 200S 1SI wird Ihnen in **STEP 7** im Registerdialog "Baugruppenzustand" angezeigt. Sie gelangen zu diesem Dialog über:

Im SIMATIC-Manager: **Datei > Öffnen > Projekt > HW-Konfig öffnen > Station > Online öffnen**> und Doppelklick auf die Baugruppe 1 SI.



## 2.10 Kommunikation über Funktionsbausteine

### 2.10.1 Grundlagen zur Kommunikation über Funktionsbausteine

#### Übersicht

Die Kommunikation zwischen CPU, ET 200S 1SI und einem Kommunikationspartner erfolgt über die Funktionsbausteine und die Protokolle der Baugruppe ET 200S 1SI. (Informationen zur Kommunikation mit Fremd-CPU's (nicht S7) finden Sie im Abschnitt Grundlagen zu Referenzdaten (Seite 87).

Die Funktionsbausteine bilden die Softwareschnittstelle zwischen der CPU und der seriellen Schnittstellenbaugruppe ET 200S 1SI. Sie müssen zyklisch aus dem Anwenderprogramm aufgerufen werden.

#### Aufbauen der Kommunikation mit der CPU

Mit jedem Anlauf der CPU wird die Baugruppe ET 200S 1SI per Systemdienst der CPU mit den aktuellen Parametern versorgt. Nachdem die Verbindung zwischen der CPU und der Baugruppe ET 200S 1SI aufgebaut ist, muss die Baugruppe ET 200S 1SI initialisiert werden.

Jeder Funktionsbaustein hat einen eigenen Anlaufmechanismus. Bevor aktiv Aufträge abgewickelt werden können, muss der zugehörige Anlaufmechanismus abgeschlossen sein.

Die Baugruppe ET 200S 1SI kann in der CPU einen Diagnosealarm auslösen. Hierbei stellt das Betriebssystem dem Anwender 2 Byte Alarminformationen zur Verfügung. Das Auswerten der Alarminformationen muss vom Anwender programmiert werden (OB82). Ein Aufruf der Funktionsbausteine im Prozess- oder Diagnosealarmprogramm ist nicht erlaubt. In den Funktionsbausteinen werden die Alarme nicht gesperrt.

Die Protokollumwandlung findet in der Baugruppe ET 200S 1SI statt. Entsprechend dem gewählten Protokoll (Prozedur 3964(R) oder ASCII-Treiber) wird die Schnittstelle der Baugruppe ET 200S 1SI an die Schnittstelle des Kommunikationspartners angepasst.

### Funktionsbausteine der Baugruppe ET 200S 1SI

Das Automatisierungssystem S7-300 stellt Ihnen eine Reihe von Funktionsbausteinen zur Verfügung, die im Anwenderprogramm die Kommunikation zwischen CPU und der seriellen Schnittstellenbaugruppe ET 200S 1SI anstoßen und steuern. Die folgende Tabelle führt die von der Baugruppe ET 200S 1SI verwendeten FBs auf.

Tabelle 2- 13 Funktionsbausteine der Baugruppe ET 200S 1SI

<b>FB</b>	<b>Name</b>	<b>Bedeutung</b>
FB2	S_RCV	Der Funktionsbaustein S_RCV ermöglicht Ihnen, Daten von einem Kommunikationspartner zu empfangen und in einem Datenbaustein zu hinterlegen.
FB3	S_SEND	Der Funktionsbaustein S_SEND ermöglicht Ihnen, einen gesamten Bereich oder einen Teilbereich eines Datenbausteins an einen Kommunikationspartner zu senden.
FB4	S_VSTAT	Der Funktionsbaustein S_VSTAT ermöglicht Ihnen, die Signalzustände an der RS-232C-Schnittstelle der Baugruppe ET 200S 1SI zu lesen.
FB5	S_VSET	Der Funktionsbaustein S_VSET ermöglicht Ihnen, die Ausgänge der RS-232C-Schnittstelle der Baugruppe ET 200S 1SI zu setzen/rückzusetzen.
FB6	S_XON	Mit dem Funktionsbaustein S_XON können Sie zusätzliche Parameter einstellen, wenn die Baugruppe für die Flusskontrolle XON/XOFF parametrierung wurde.
FB7	S_RTS	Mit dem Funktionsbaustein S_RTS können Sie zusätzliche Parameter einstellen, wenn die Baugruppe für die Flusskontrolle RTS/CTS parametrierung wurde.
FB8	S_V24	Mit dem Funktionsbaustein S_V24 können Sie zusätzliche Parameter einstellen, wenn die Baugruppe für die automatische Bedienung der V.24-Signale parametrierung wurde.

---

#### Hinweis

Diese Instanzdatenbausteine dürfen nicht während einer aktiven Kommunikation des SEND/RECEIVE-Bausteins auf die CPU geladen werden.

---

#### Siehe auch

Technische Daten (Seite 108)

## 2.10.2 Funktionsbaustein FB3 S\_SEND

### FB3 S\_SEND: Daten an einen Kommunikationspartner senden

Der FB S\_SEND überträgt einen Datenblock von einem Datenbaustein, spezifiziert durch die Parameter DB\_NO, DBB\_NO und LEN, zur Baugruppe ET 200S 1SI. Der FB S\_SEND wird zur Datenübertragung im Zyklus statisch (ohne Bedingungen) oder **alternativ** in einem zeitgesteuerten Programm aufgerufen.

Mit einer positiven Flanke am Eingang REQ wird die Übertragung der Daten angestoßen. Je nach Datenmenge kann eine Datenübertragung über mehrere Aufrufe (Programmzyklen) laufen.

Der Funktionsbaustein FB S\_SEND kann mit Signalzustand "1" am Parametereingang R im Zyklus aufgerufen werden. Es wird damit die Übertragung zur Baugruppe ET 200S 1SI abgebrochen und der FB S\_SEND in den Grundzustand versetzt. Daten, die die Baugruppe ET 200S 1SI bereits erhalten hat, werden noch an den Kommunikationspartner gesendet. Steht statisch Signalzustand "1" am Eingang R an, so ist das Senden ausgeschaltet.

An dem Parameter LADDR wird die Adresse der anzusprechenden Baugruppe ET 200S 1SI angegeben.

Der Ausgang DONE zeigt "Auftragsende ohne Fehler" an. ERROR zeigt einen aufgetretenen Fehler an. In STATUS wird bei einem Fehler die entsprechende Ereignisnummer (siehe Kapitel Diagnose (Seite 101)) angezeigt. Ist kein Fehler aufgetreten, hat STATUS den Wert 0. DONE und ERROR/STATUS werden auch bei RESET des FB S\_SEND ausgegeben (siehe Zeitablaufdiagramm). Bei einem aufgetretenen Fehler wird das Binärergebnis BIE rückgesetzt. Wird der Baustein ohne Fehler beendet, hat das Binärergebnis den Zustand "1".

### Anlauf

Der Parameter COM\_RST des FB S\_SEND dient dazu, dem FB einen Anlauf mitzuteilen.

Setzen Sie den Parameter COM\_RST im Anlauf-OB auf 1.

Rufen Sie den FB im zyklischen Betrieb auf, ohne den Parameter COM\_RST zu setzen oder rückzusetzen.

Wenn der Parameter COM\_RST gesetzt ist,

- ermittelt der FB Informationen über das Modul ET 200S 1SI (Anzahl Byte im Peripheriebereich, in dezentraler Peripherie oder nicht).
- setzt sich der FB zurück und beendet dabei evtl. einen zuvor (vor dem letzten Stopp-Übergang der CPU) begonnenen Auftrag.

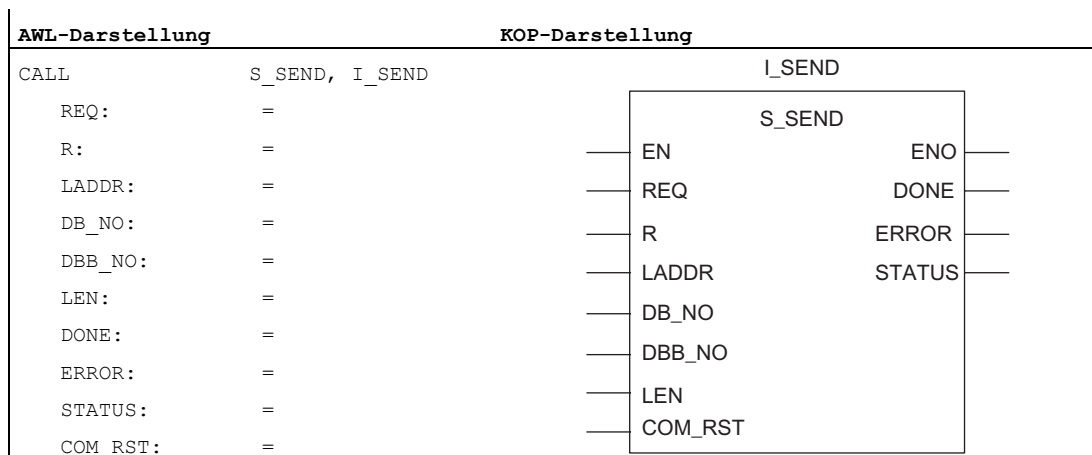
Wenn der FB die Information über das Modul ET 200S 1SI ermittelt hat, setzt er den Parameter COM\_RST selbst zurück.

**Hinweis**

Der Funktionsbaustein S\_SEND hat keine Parameterprüfung, bei falscher Parametrierung kann die CPU in den Zustand STOP verzweigen.

Bevor ein angestoßener Auftrag nach einem Zustandsübergang der CPU von STOP nach RUN von der Baugruppe ET 200S 1SI bearbeitet werden kann, muss der CPU-Anlaufmechanismus der Baugruppe ET 200S für den FB S\_SEND abgeschlossen sein (siehe oben). Ein in der Zwischenzeit angestoßener Auftrag geht nicht verloren. Er wird nach Abschluss der Anlaufkoordinierung zur Baugruppe ET 200S 1SI übertragen.

**Aufruf FB3**



**Hinweis**

Die Parameter EN und ENO sind nur in der grafischen Darstellung vorhanden (bei KOP oder FUP). Für die Bearbeitung dieser Parameter wird vom Compiler das Binärergebnis BIE verwendet.

Das Binärergebnis BIE wird auf Signalzustand "1" gesetzt, wenn der Baustein fehlerfrei beendet wurde. Liegt ein Fehler vor, wird das Binärergebnis BIE auf "0" gesetzt.

### Belegung im Datenbereich

Der FB S\_SEND arbeitet mit einem Instanz-DB I\_SEND zusammen. Die DB-Nummer wird beim Aufruf mitgegeben. Ein Zugriff auf die Daten im Instanz-DB ist nicht zulässig.

#### Hinweis

Ausnahme: Im Fehlerfall, STATUS == W#16#1Exx, können Sie einen genaueren Fehlerhinweis aus der Variablen SFCERR entnehmen (siehe Kapitel Diagnose (Seite 101)). Diese Fehlervariable kann nur über einen symbolischen Zugriff auf den Instanz-DB geladen werden.

### Parameter FB3 S\_SEND

Die folgende Tabelle führt die Parameter von S\_SEND (FB3) auf.

Tabelle 2- 14 FB3: Parameter S\_SEND

Name	Typ	Datentyp	Beschreibung	Zulässige Belegung, Bemerkung
REQ	INPUT	BOOL	Auftragsanstoß bei positiver Flanke	
R	INPUT	BOOL	Auftragsabbruch	Laufender Auftrag wird abgebrochen. Senden gesperrt.
LADDR	INPUT	INT	Basisadresse der Baugruppe ET 200S 1SI	Die Basisadresse wird aus STEP 7 entnommen.
DB_NO	INPUT	INT	Datenbausteinnummer	Sende-DB-Nr.: CPU-spezifisch (Null ist nicht zulässig)
DBB_NO	INPUT	INT	Datenbytenummer	0 ≤ DBB_NO ≤ 8190 Sendedaten ab Datenwort
LEN	INPUT	INT	Datenlänge	1 ≤ LEN ≤ 224 Angabe in Anzahl Byte
DONE <sup>1</sup>	OUTPUT	BOOL	Auftrag fertig ohne Fehler	Parameter STATUS == 16#00
ERROR <sup>1</sup>	OUTPUT	BOOL	Auftrag fertig mit Fehler	Parameter STATUS enthält die Fehlerinformation.
STATUS <sup>1</sup>	OUTPUT	WORD	Spezifikation des Fehlers	Bei ERROR == 1 steht im Parameter STATUS die Fehlerinformation.
COM_RST	IN_OUT	BOOL	Neustart des FB	

<sup>1</sup> Die Parameter DONE, ERROR und STATUS stehen nach korrektem Sendeauftrag **einen** CPU-Zyklus lang zur Verfügung!

### Zeitablaufdiagramm FB3 S\_SEND

Das folgende Bild zeigt das Verhalten der Parameter DONE und ERROR je nachdem, wie die Eingänge REQ und R verdrahtet sind.

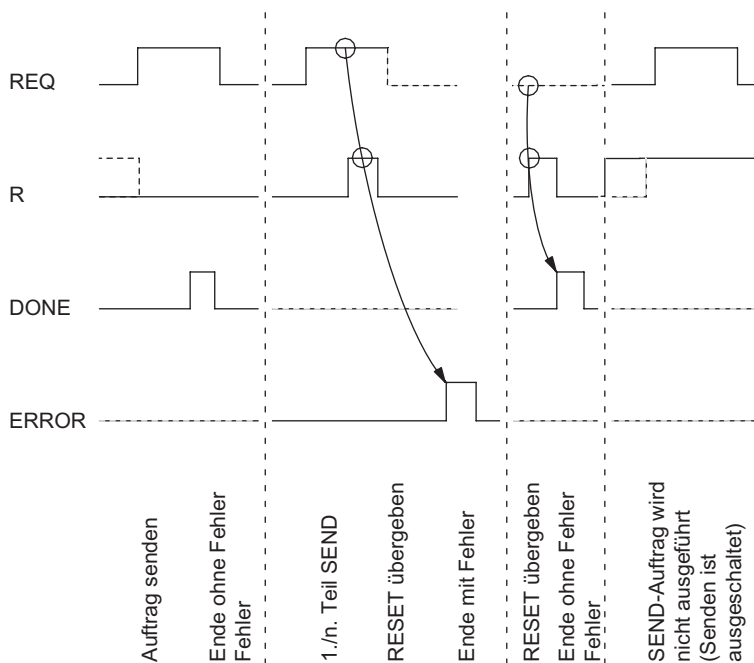


Bild 2-20 Zeitablaufdiagramm FB3 S\_SEND

#### Hinweis

Der Eingang REQ ist flankengetriggert. Es genügt am Eingang REQ eine positive Flanke. Es muss nicht während der gesamten Übertragung das VKE (Verknüpfungsergebnis) auf "1" sein.

### 2.10.3 Funktionsbaustein FB2 S\_RCV

#### FB S\_RCV: Daten von einem Kommunikationspartner empfangen

Der FB S\_RCV überträgt Daten von der Baugruppe ET 200S 1SI zu einem S7-Datenbereich, spezifiziert durch die Parameter DB\_NO und DBB\_NO. Der FB S\_RCV wird zur Datenübertragung im Zyklus oder alternativ in einem zeitgesteuerten Programm statisch (ohne Bedingungen) aufgerufen.

Mit (statisch) Signalzustand "1" am Parameter EN\_R wird die Überprüfung, ob Daten von der Baugruppe ET 200S 1SI zu lesen sind, freigegeben. Eine laufende Übertragung kann mit Signalzustand "0" am Parameter EN\_R abgebrochen werden. Der abgebrochene Empfangsauftrag wird mit einer Fehlermeldung (STATUS-Ausgang) beendet. Der Empfang ist ausgeschaltet, solange Signalzustand "0" am Parameter EN\_R ansteht. Je nach Datenmenge kann eine Datenübertragung über mehrere Aufrufe (Programmzyklen) laufen.

Erkennt der Funktionsbaustein Signalzustand "1" am Parameter R, dann wird der momentane Übertragungsauftrag abgebrochen und der FB S\_RCV in den Grundzustand versetzt. Der Empfang ist ausgeschaltet, solange Signalzustand "1" am Parameter R ansteht. Bei erneutem Signalzustand "0" wird das abgebrochene Telegramm erneut, von Anfang an, empfangen.

An dem Parameter LADDR wird die Adresse der anzusprechenden Baugruppe ET 200S 1SI angegeben.

Der Ausgang NDR zeigt "Auftrag fertig ohne Fehler/Daten übernommen" an (alle Daten gelesen). ERROR zeigt einen aufgetretenen Fehler an. In STATUS wird bei einem Fehler die entsprechende Fehlernummer angezeigt. Ist der Empfangspuffer mehr als 2/3 voll, enthält STATUS nach jedem Aufruf von S\_RCV eine Warnung. Liegen keine Fehler oder Warnungen vor, hat STATUS den Wert "0".

NDR und ERROR/STATUS werden auch dann ausgegeben, wenn der FB S\_RCV zurückgesetzt wird (Parameter LEN == 16#00) (siehe Zeitablaufdiagramm). Bei einem aufgetretenen Fehler wird das Binärergebnis BIE rückgesetzt. Wird der Baustein ohne Fehler beendet, hat das Binärergebnis den Zustand "1".

#### Anlauf

Der Parameter COM\_RST des FB S\_RCV dient dazu, dem FB einen Anlauf mitzuteilen.

Setzen Sie den Parameter COM\_RST im Anlauf-OB auf 1.

Rufen Sie den FB im zyklischen Betrieb auf, ohne den Parameter COM\_RST zu setzen oder rückzusetzen.

Wenn der Parameter COM\_RST gesetzt ist,

- ermittelt der FB Informationen über das Modul ET 200S 1SI (Anzahl Byte im Peripheriebereich, in dezentraler Peripherie oder nicht).
- setzt sich der FB zurück und beendet dabei evtl. einen zuvor (vor dem letzten Stopp-Übergang der CPU) begonnenen Auftrag.

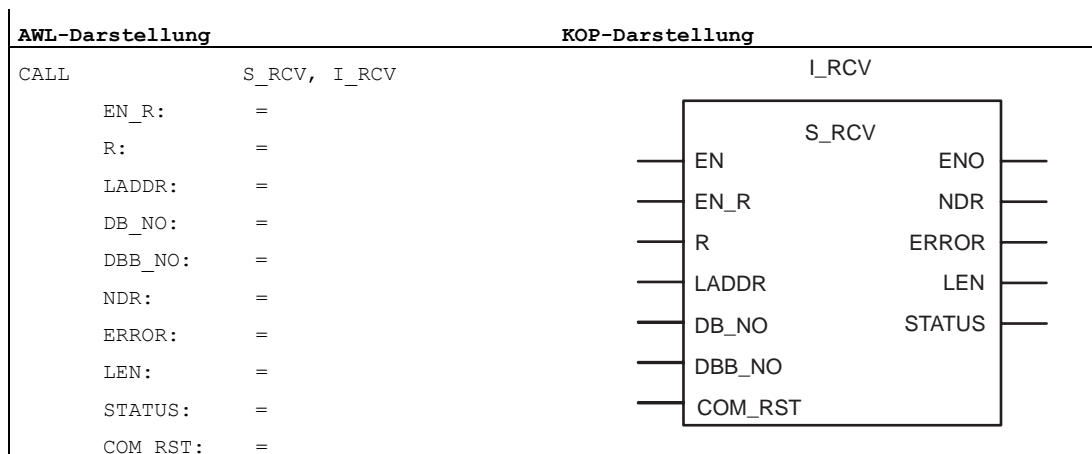
Wenn der FB die Information über das Modul ET 200S 1SI ermittelt hat, setzt er den Parameter COM\_RST selbst zurück.

**Hinweis**

Der Funktionsbaustein S\_RCV hat keine Parameterprüfung, bei falscher Parametrierung kann die CPU in den Zustand STOP verzweigen.

Bevor ein angestoßener Auftrag nach einem Zustandsübergang der CPU von STOP nach RUN von der Baugruppe ET 200S 1SI empfangen werden kann, muss der CPU-Anlaufmechanismus der Baugruppe ET 200S für den FB S\_RCV abgeschlossen sein.

**Aufruf FB 2**



**Hinweis**

Die Parameter EN und ENO sind nur in der grafischen Darstellung vorhanden (bei KOP oder FUP). Für die Bearbeitung dieser Parameter wird vom Compiler das Binärergebnis BIE verwendet.

Das Binärergebnis BIE wird auf Signalzustand "1" gesetzt, wenn der Baustein fehlerfrei beendet wurde. Liegt ein Fehler vor, wird das Binärergebnis BIE auf "0" gesetzt.

**Belegung im Datenbereich**

Der FB S\_RCV arbeitet mit einem Instanz-DB I\_RCV zusammen. Die DB-Nummer wird beim Aufruf mitgegeben. Ein Zugriff auf die Daten im Instanz-DB ist nicht zulässig.

**Hinweis**

Ausnahme: Im Fehlerfall, STATUS == W#16#1Exx, können Sie einen genaueren Fehlerhinweis aus der Variablen SFCERR entnehmen. Diese Fehlervariable kann nur über einen symbolischen Zugriff auf den Instanz-DB geladen werden.



## Parameter FB2 S\_RCV

Die folgende Tabelle führt die Parameter von S\_RCV (FBs) auf.

Tabelle 2- 15 FB2: Parameter S\_RCV

Name	Typ	Datentyp	Beschreibung	Zulässige Belegung, Bemerkung
EN_R	INPUT	BOOL	Freigabe für Daten lesen	
R	INPUT	BOOL	Auftragsabbruch	Laufender Auftrag wird abgebrochen. Empfang gesperrt.
LADDR	INPUT	INT	Basisadresse der Baugruppe ET 200S 1SI	Die Basisadresse wird aus STEP 7 entnommen.
DB_NO	INPUT	INT	Datenbausteinnummer	Empfangs-DB-Nr.: CPU-spezifisch, Null ist nicht erlaubt
DBB_NO	INPUT	INT	Datenbytenummer	0 ≤ DBB_NO ≤ 8190 Empfangsdaten ab Datenwort
NDR <sup>1</sup>	OUTPUT	BOOL	Auftrag fertig ohne Fehler, Daten übernommen	Parameter STATUS == 16#00
ERROR <sup>1</sup>	OUTPUT	BOOL	Auftrag fertig mit Fehler	Parameter STATUS enthält die Fehlerinformation.
LEN <sup>1</sup>	OUTPUT	INT	Länge des empfangenen Telegramms	1 ≤ LEN ≤ 224 Angabe in Anzahl Byte
STATUS <sup>1</sup>	OUTPUT	WORD	Spezifikation des Fehlers	Bei ERROR == 1 steht im Parameter STATUS die Fehlerinformation.
COM_RST	IN_OUT	BOOL	Neustart des FB	
<sup>1</sup> Die Parameter NDR; ERROR, LEN und STATUS stehen nach korrektem Empfangsauftrag <b>einen</b> CPU-Zyklus lang zur Verfügung!				

### Zeitablaufdiagramm FB2 S\_RCV

Das folgende Bild zeigt das Verhalten der Parameter NDR, LEN und ERROR je nachdem, wie die Eingänge EN\_R und R verdrahtet sind.

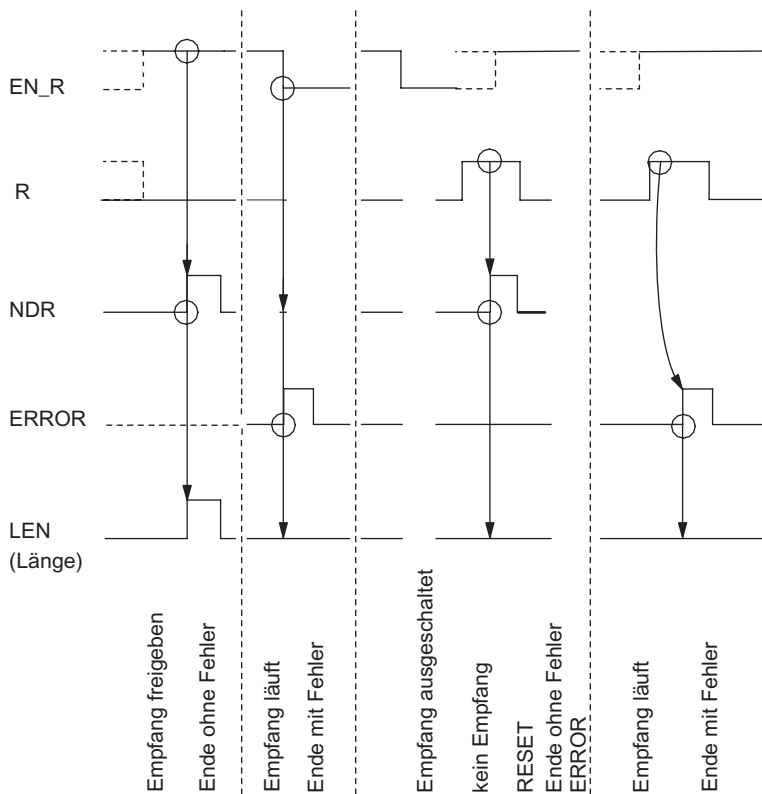


Bild 2-21 Zeitablaufdiagramm FB2 S\_RCV

#### Hinweis

Der Eingang EN\_R ist statisch auf "1" zu legen. Während des gesamten Empfangsauftrages muss der Parameter EN\_R mit dem VKE "1" (Verknüpfungsergebnis) versorgt werden.

## 2.10.4 Funktionen für die Parametrierung von Optionen für die Datenflusskontrolle

### Prinzip

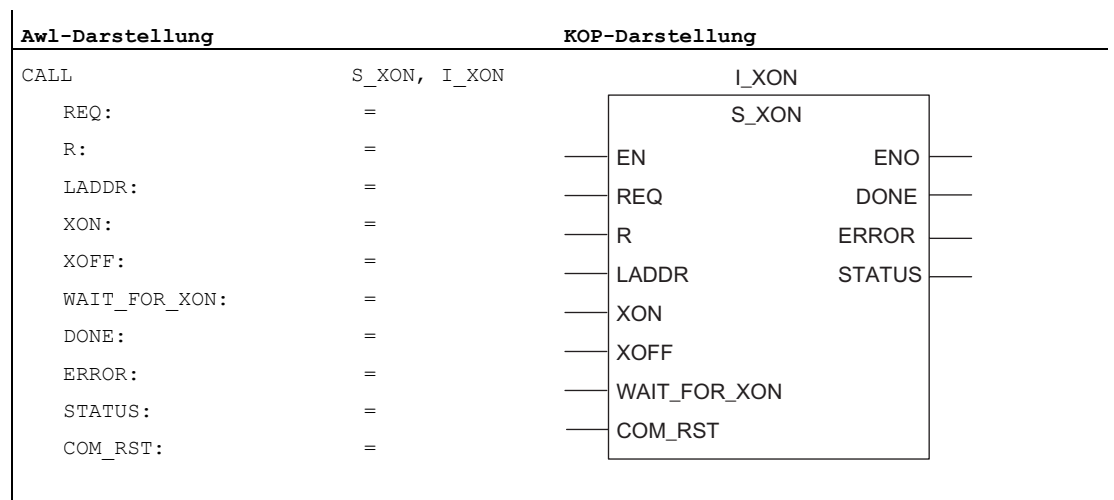
Wenn Sie die serielle Schnittstellenbaugruppe ET 200S 1SI mit einer S7-CPU einsetzen und die Baugruppe mit der Hardware-Konfiguration von STEP 7 konfigurieren, können Sie unter den folgenden Optionen eine Methode für die Datenflusskontrolle auswählen:

- keine
- XON/XOFF
- RTS/CTS
- Automatische Bedienung der V.24-Signale

Für jede dieser Optionen können zusätzliche Parameter eingestellt werden. Diese zusätzlichen Parameter nehmen voreingestellte Werte an, bei denen es sich um typische Werte handelt, die für die meisten Anwendungen angemessen sind. Sie können diese Parameter jedoch über das Anwenderprogramm und die folgenden Funktionsbausteine ändern.

### FB6 S\_XON: Einstellen der Zeichen für XON/XOFF

Mit dem Funktionsbaustein S\_XON können Sie zusätzliche Parameter einstellen (siehe Parameter FB6), wenn die Baugruppe für die Flusskontrolle XON/XOFF parametrierung wurde.



### Belegung im Datenbereich

Der FB S\_XON arbeitet mit einem Instanz-DB I\_XON zusammen. Die DB-Nummer wird beim Aufruf mitgegeben. Ein Zugriff auf die Daten im Instanz-DB ist nicht zulässig.

#### Hinweis

Ausnahme: Im Fehlerfall, STATUS == W#16#1Exx, können Sie einen genaueren Fehlerhinweis aus der Variablen SFCERR entnehmen. Diese Fehlervariable kann nur über einen symbolischen Zugriff auf den Instanz-DB geladen werden.

### Parameter FB6

Die folgende Tabelle führt die Parameter für FB6 auf.

Tabelle 2- 16 FB6: Parameter S\_XON

Name	Typ	Datentyp	Beschreibung	Zulässige Belegung, Bemerkung	Voreinstellung
REQ	INPUT	BOOL	Auftragsanstoß bei positiver Flanke		
R	INPUT	BOOL	Auftragsabbruch	Laufender Auftrag wird abgebrochen. Senden gesperrt.	
LADDR	INPUT	INT	Basisadresse der Baugruppe ET 200S 1SI	Die Basisadresse wird aus STEP 7 entnommen.	
XON	INPUT	BYTE	XON-Zeichen	0 bis 7FH (7 Datenbits) 0 bis FFH (8 Datenbits)	11 (DC1)
XOFF	INPUT	BYTE	XOFF-Zeichen	0 bis 7FH (7 Datenbits) 0 bis FFH (8 Datenbits)	13 (DC3)
WAIT_FOR_XON	INPUT	TIME	Wartezeit für XON nach XOFF	20 ms bis 10 min 55 s 350 ms	2 s
DONE <sup>1</sup>	OUTPUT	BOOL	Auftrag fertig ohne Fehler	Parameter STATUS == 16#00	
ERROR <sup>1</sup>	OUTPUT	BOOL	Auftrag fertig mit Fehler	Parameter STATUS enthält die Fehlerinformation.	
STATUS <sup>1</sup>	OUTPUT	WORD	Spezifikation des Fehlers	Bei ERROR == 1 steht im Parameter STATUS die Fehlerinformation.	
COM_RST	IN_OUT	BOOL	Neustart des FB		

<sup>1</sup> Die Parameter DONE, ERROR und STATUS stehen nach korrektem Auftrag **einen** CPU-Zyklus lang zur Verfügung!

## Anlauf

Der Parameter COM\_RST des FB S\_XON dient dazu, dem FB einen Anlauf mitzuteilen.

Setzen Sie den Parameter COM\_RST im Anlauf-OB auf 1.

Rufen Sie den FB im zyklischen Betrieb auf, ohne den Parameter COM\_RST zu setzen oder rückzusetzen.

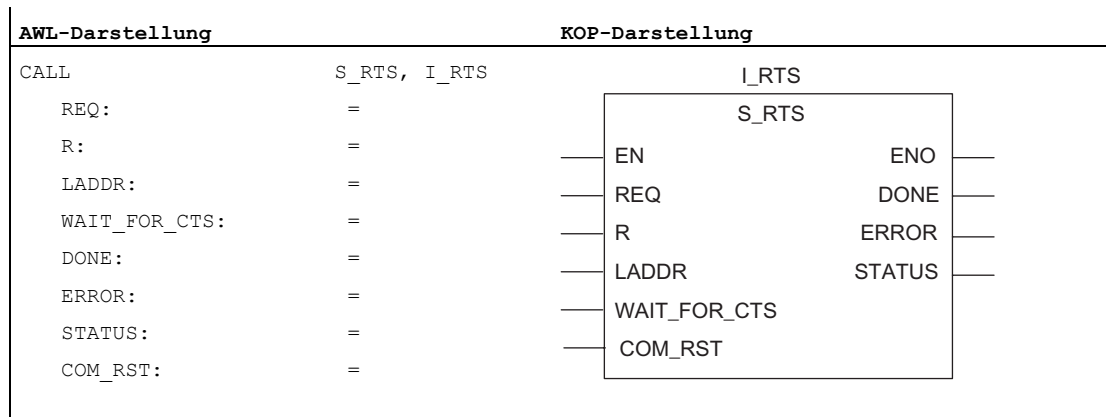
Wenn der Parameter COM\_RST gesetzt ist,

- ermittelt der FB Informationen über das Modul ET 200S 1SI (Anzahl Byte im Peripheriebereich, in dezentraler Peripherie oder nicht).
- setzt sich der FB zurück und beendet dabei evtl. einen zuvor (vor dem letzten Stopp-Übergang der CPU) begonnenen Auftrag.

Wenn der FB die Information über das Modul ET 200S 1SI ermittelt hat, setzt er den Parameter COM\_RST selbst zurück.

## FB7 S\_RTS: Einstellen der Parameter für RTS/CTS

Mit dem Funktionsbaustein S\_RTS können Sie zusätzliche Parameter einstellen (siehe Parameter FB7), wenn die Baugruppe für die Flusskontrolle RTS/CTS parametrierung wurde.



## Belegung im Datenbereich

Der FB S\_RTS arbeitet mit einem Instanz-DB I\_RTS zusammen. Die DB-Nummer wird beim Aufruf mitgegeben. Ein Zugriff auf die Daten im Instanz-DB ist nicht zulässig.

### Hinweis

Ausnahme: Im Fehlerfall, STATUS == W#16#1Exx, können Sie einen genaueren Fehlerhinweis aus der Variablen SFCERR entnehmen. Diese Fehlervariable kann nur über einen symbolischen Zugriff auf den Instanz-DB geladen werden.

**Parameter FB7**

Die folgende Tabelle führt die Parameter für FB7 auf.

Tabelle 2- 17 FB7: Parameter S\_RTS

Name	Typ	Datentyp	Beschreibung	Zulässige Belegung, Bemerkung	Voreinstellung
REQ	INPUT	BOOL	Auftragsanstoß bei positiver Flanke		
R	INPUT	BOOL	Auftragsabbruch	Laufender Auftrag wird abgebrochen. Senden gesperrt.	
LADDR	INPUT	INT	Basisadresse der Baugruppe ET 200S SI	Die Basisadresse wird aus STEP 7 entnommen.	
WAIT_FOR_CTS	INPUT	TIME	Wartezeit für CTS = ON	20 ms bis 10 min 55 s 350 ms	2 s
DONE <sup>1</sup>	OUTPUT	BOOL	Auftrag fertig ohne Fehler	Parameter STATUS == 16#00	
ERROR <sup>1</sup>	OUTPUT	BOOL	Auftrag fertig mit Fehler	Parameter STATUS enthält die Fehlerinformation.	
STATUS <sup>1</sup>	OUTPUT	WORD	Spezifikation des Fehlers	Bei ERROR == 1 steht im Parameter STATUS die Fehlerinformation.	
COM_RST	IN_OUT	BOOL	Neustart des FB		

<sup>1</sup> Die Parameter DONE, ERROR und STATUS stehen nach korrektem Auftrag **einen** CPU-Zyklus lang zur Verfügung!

**Anlauf**

Der Parameter COM\_RST des FB S\_RST dient dazu, dem FB einen Anlauf mitzuteilen.

Setzen Sie den Parameter COM\_RST im Anlauf-OB auf 1.

Rufen Sie den FB im zyklischen Betrieb auf, ohne den Parameter COM\_RST zu setzen oder rückzusetzen.

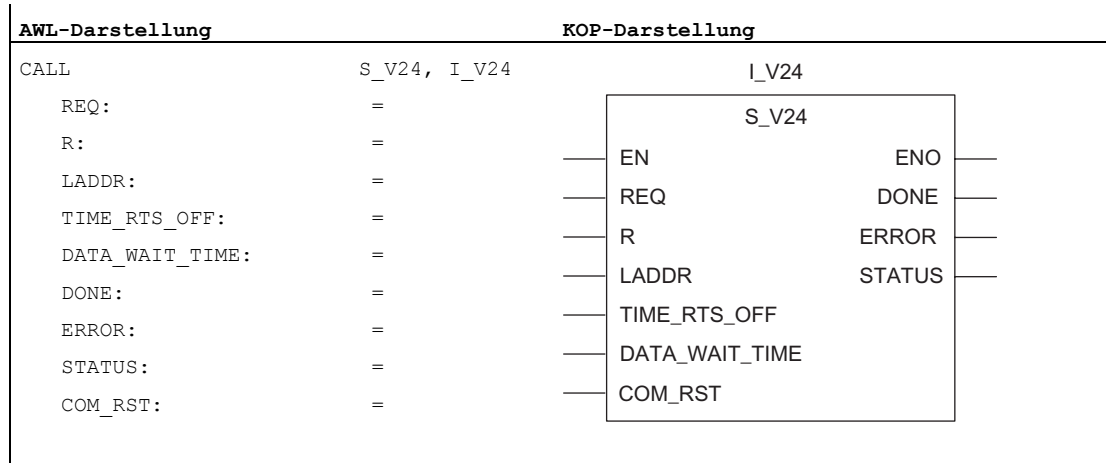
Wenn der Parameter COM\_RST gesetzt ist,

- ermittelt der FB Informationen über das Modul ET 200S 1SI (Anzahl Byte im Peripheriebereich, in dezentraler Peripherie oder nicht).
- setzt sich der FB zurück und beendet dabei evtl. einen zuvor (vor dem letzten Stopp-Übergang der CPU) begonnenen Auftrag.

Wenn der FB die Information über das Modul ET 200S 1SI ermittelt hat, setzt er den Parameter COM\_RST selbst zurück.

### FB8 S\_V24: Einstellen der Parameter für automatische Bedienung der RS-232C-Begleitsignale

Mit dem Funktionsbaustein S\_V24 können Sie zusätzliche Parameter einstellen (siehe Parameter FB8), wenn die Baugruppe für automatische Bedienung der RS-232C-Begleitsignale parametrierbar ist.



### Belegung im Datenbereich

Der FB P\_V24 arbeitet mit einem Instanz-DB I\_V24 zusammen. Die DB-Nummer wird beim Aufruf mitgegeben. Ein Zugriff auf die Daten im Instanz-DB ist nicht zulässig.

#### Hinweis

Ausnahme: Im Fehlerfall, STATUS == W#16#1Exx, können Sie einen genaueren Fehlerhinweis aus der Variablen SFCERR entnehmen. Diese Fehlervariable kann nur über einen symbolischen Zugriff auf den Instanz-DB geladen werden.

### Parameter FB8

Die folgende Tabelle führt die Parameter für FB8 auf.

Tabelle 2- 18 FB8: Parameter S\_V24

Name	Typ	Datentyp	Beschreibung	Zulässige Belegung, Bemerkung	Voreinstellung
REQ	INPUT	BOOL	Auftragsanstoß bei positiver Flanke		
R	INPUT	BOOL	Auftragsabbruch	Laufender Auftrag wird abgebrochen. Senden gesperrt.	
LADDR	INPUT	INT	Basisadresse der Baugruppe ET 200S 1SI	Die Basisadresse wird aus STEP 7 entnommen.	
TIME_RTS_OFF	INPUT	TIME	Zeit, die nach der Übertragung ablaufen muss, bevor RTS ausgeschaltet wird.	0 ms bis 10 min 55 s 350 ms	10 ms
DATA_WAIT_TIME	INPUT	TIME	Zeit, die darauf gewartet wird, dass der Partner CTS = ON setzt, nachdem RTS gesetzt wurde.	0 ms bis 10 min 55 s 350 ms	10 ms
DONE <sup>1</sup>	OUTPUT	BOOL	Auftrag fertig ohne Fehler	Parameter STATUS == 16#00	
ERROR <sup>1</sup>	OUTPUT	BOOL	Auftrag fertig mit Fehler	Parameter STATUS enthält die Fehlerinformation.	
STATUS <sup>1</sup>	OUTPUT	WORD	Spezifikation des Fehlers	Bei ERROR == 1 steht im Parameter STATUS die Fehlerinformation.	
COM_RST	IN_OUT	BOOL	Neustart des FB		
<sup>1</sup> Die Parameter DONE, ERROR und STATUS stehen nach korrektem Auftrag <b>einen</b> CPU-Zyklus lang zur Verfügung!					

### Anlauf

Der Parameter COM\_RST des FB S\_V24 dient dazu, dem FB einen Anlauf mitzuteilen.

Setzen Sie den Parameter COM\_RST im Anlauf-OB auf 1.

Rufen Sie den FB im zyklischen Betrieb auf, ohne den Parameter COM\_RST zu setzen oder rückzusetzen.

Wenn der Parameter COM\_RST gesetzt ist,

- ermittelt der FB Informationen über das Modul ET 200S 1SI (Anzahl Byte im Peripheriebereich, in dezentraler Peripherie oder nicht).
- setzt sich der FB zurück und beendet dabei evtl. einen zuvor (vor dem letzten Stopp-Übergang der CPU) begonnenen Auftrag.

Wenn der FB die Information über das Modul ET 200S 1SI ermittelt hat, setzt er den Parameter COM\_RST selbst zurück.



## 2.10.5 Lesen und Steuern von RS-232C-Begleitsignalen

### Prinzip

Zum Lesen und Steuern der RS-232C-Begleitsignale stehen Ihnen die Funktionsbausteine FB4 S\_VSTAT zum Prüfen der Schnittstellenzustände und FB S\_VSET zum Setzen/Rücksetzen der Schnittstellenausgänge zur Verfügung.

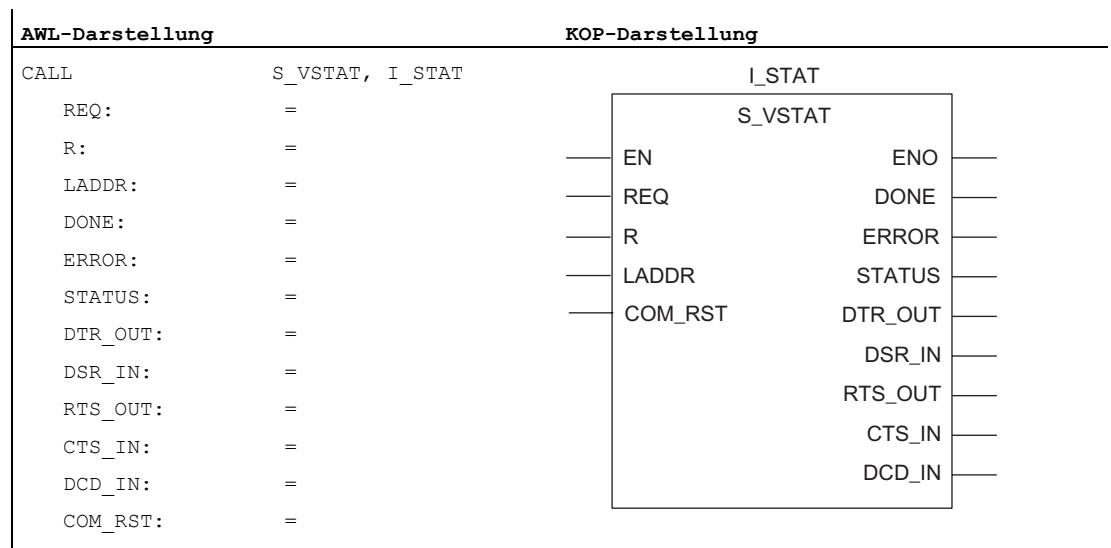
### FB4 S\_VSTAT: Prüfen des Schnittstellenzustands der Baugruppe ET 200S 1SI

Der FB S\_VSTAT liest die RS-232C-Begleitsignale der Baugruppe ET 200S 1SI und stellt sie dem Anwender in den Bausteinparametern zur Verfügung. Die FB S\_VSTAT wird statisch (ohne Bedingungen) für die Datenübertragung im Zyklus oder alternativ in einem zeitgesteuerten Programm aufgerufen.

Die RS-232C-Begleitsignale werden mit jedem Aufruf der Funktion aktualisiert (zyklisches Pollen).

An dem Parameter LADDR wird die Adresse der anzusprechenden Baugruppe ET 200S 1SI angegeben.

### Aufruf FB4



### Hinweis

Die Parameter EN und ENO sind nur in der grafischen Darstellung vorhanden (bei KOP oder FUP). Für die Bearbeitung dieser Parameter wird vom Compiler das Binärergebnis BIE verwendet.

Das Binärergebnis BIE wird auf Signalzustand "1" gesetzt, wenn der Baustein fehlerfrei beendet wurde. Liegt ein Fehler vor, wird das Binärergebnis BIE auf "0" gesetzt.

### Belegung im Datenbereich

Der FB S\_VSTAT arbeitet mit einem Instanz-DB I\_STAT zusammen. Die DB-Nummer wird beim Aufruf mitgegeben. Ein Zugriff auf die Daten im Instanz-DB ist nicht zulässig.

#### Hinweis

Zum Erkennen eines Signalwechsels ist eine Mindestimpulsdauer notwendig. Ausschlaggebende Größen sind die CPU-Zykluszeit, die Aktualisierungszeit auf der Baugruppe ET 200S 1SI und die Reaktionszeit des Kommunikationspartners.

### Parameter FB4 V24\_STAT

Die folgende Tabelle führt die Parameter des Funktionsbausteins S\_VSTAT (FB4) auf.

Tabelle 2- 19 FB4: Parameter V24\_STAT

Name	Typ	Datentyp	Beschreibung	Zulässige Belegung, Bemerkung
REQ	INPUT	BOOL	Auftragsanstoß bei positiver Flanke	
R	INPUT	BOOL	Auftragsabbruch	Laufender Auftrag wird abgebrochen. Senden gesperrt.
LADDR	INPUT	INT	Basisadresse der Baugruppe ET 200S 1SI	Die Basisadresse wird aus STEP 7 entnommen.
DONE <sup>1</sup>	OUTPUT	BOOL	Zeigt an, dass der FB beendet ist	(Ausgang ET 200S 1SI)
ERROR <sup>1</sup>	OUTPUT	BOOL	Auftrag fertig mit Fehler	Parameter STATUS enthält die Fehlerinformation.
STATUS <sup>1</sup>	OUTPUT	WORD	Spezifikation des Fehlers	Bei ERROR == 1 steht im Parameter STATUS die Fehlerinformation.
DTR_OUT <sup>1</sup>	OUTPUT	BOOL	Data terminal ready, ET 200S 1SI ist betriebsbereit.	(Ausgang ET 200S 1SI)
DSR_IN <sup>1</sup>	OUTPUT	BOOL	Data set ready, Kommunikationspartner ist betriebsbereit.	(Eingang ET 200S 1SI)
RTS_OUT <sup>1</sup>	OUTPUT	BOOL	Request to send, ET 200S 1SI ist sendebereit.	(Ausgang ET 200S 1SI)
CTS_IN <sup>1</sup>	OUTPUT	BOOL	Clear to send, Kommunikationspartner kann Daten von der Baugruppe ET 200S 1SI empfangen (Antwort auf RTS = ON der ET 200S 1SI)	(Eingang ET 200S 1SI)
DCD_IN <sup>1</sup>	OUTPUT	BOOL	Data carrier detect, Empfangssignalpegel	(Eingang ET 200S 1SI)
COM_RST	IN_OUT	BOOL	Neustart des FB	

<sup>1</sup> Diese Parameter stehen nach korrektem Auftrag **einen** CPU-Zyklus lang zur Verfügung!

## Anlauf

Der Parameter COM\_RST des FB S\_VSTAT dient dazu, dem FB einen Anlauf mitzuteilen.

Setzen Sie den Parameter COM\_RST im Anlauf-OB auf 1.

Rufen Sie den FB im zyklischen Betrieb auf, ohne den Parameter COM\_RST zu setzen oder rückzusetzen.

Wenn der Parameter COM\_RST gesetzt ist,

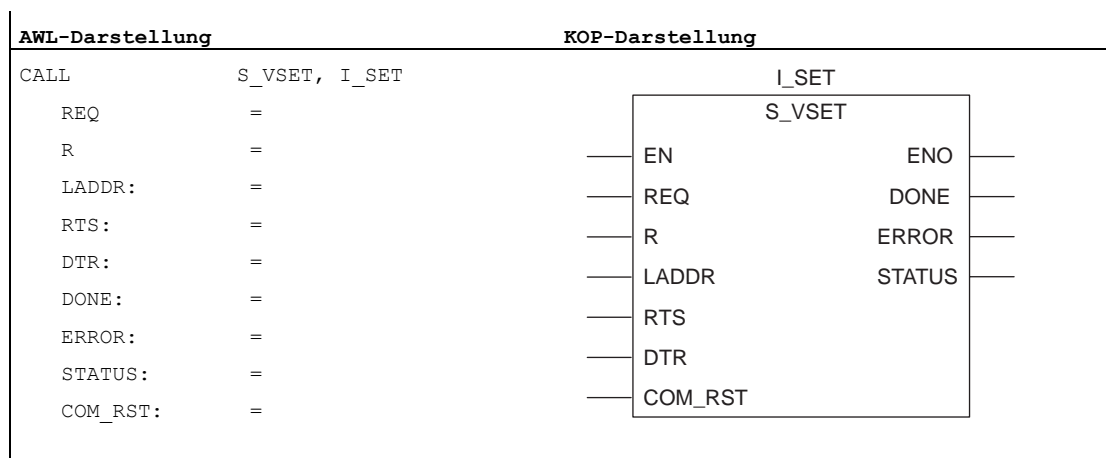
- ermittelt der FB Informationen über das Modul ET 200S 1SI (Anzahl Byte im Peripheriebereich, in dezentraler Peripherie oder nicht).
- setzt sich der FB zurück und beendet dabei evtl. einen zuvor (vor dem letzten Stopp-Übergang der CPU) begonnenen Auftrag.

Wenn der FB die Information über das Modul ET 200S 1SI ermittelt hat, setzt er den Parameter COM\_RST selbst zurück.

## FB5 S\_VSET: Setzen/Rücksetzen der Schnittstellenausgänge der Baugruppe ET 200S 1SI

Sie können die Schnittstellenausgänge über die entsprechenden Parametereingänge des FB S\_VSET setzen und rücksetzen. Der Funktionsbaustein FB S\_VSET wird im Zyklus oder alternativ in einem zeitgesteuerten Programm statisch (ohne Bedingungen) aufgerufen.

An dem Parameter LADDR wird die Adresse der anzusprechenden Baugruppe ET 200S 1SI angegeben.



### Hinweis

Die Parameter EN und ENO sind nur in der grafischen Darstellung vorhanden (bei KOP oder FUP). Für die Bearbeitung dieser Parameter wird vom Compiler das Binärergebnis BIE verwendet.

Das Binärergebnis BIE wird auf Signalzustand "1" gesetzt, wenn der Baustein fehlerfrei beendet wurde. Liegt ein Fehler vor, wird das Binärergebnis BIE auf "0" gesetzt.

### Belegung im Datenbereich

Der FB S\_VSET arbeitet mit einem Instanz-DB I\_SET zusammen. Die DB-Nummer wird beim Aufruf mitgegeben. Ein Zugriff auf die Daten im Instanz-DB ist nicht zulässig.

### Parameter FB5 S\_VSET

Die folgende Tabelle führt die Parameter des Funktionsbausteins S\_VSET (FB5) auf.

Tabelle 2- 20 FB5: Parameter S\_VSET

Name	Typ	Datentyp	Beschreibung	Zulässige Belegung, Bemerkung
REQ	INPUT	BOOL	Auftragsanstoß bei positiver Flanke	
R	INPUT	BOOL	Auftragsabbruch	Laufender Auftrag wird abgebrochen. Senden gesperrt.
LADDR	INPUT	INT	Basisadresse der Baugruppe ET 200S 1SI	Die Basisadresse wird aus STEP 7 entnommen.
RTS	INPUT	BOOL	Request to send, ET 200S 1SI ist sendebereit.	(Ausgang ET 200S 1SI steuern)
DTR	INPUT	BOOL	Data terminal ready, ET 200S 1SI ist betriebsbereit.	(Ausgang ET 200S 1SI steuern)
DONE <sup>1</sup>	OUTPUT	BOOL	Zeigt an, dass der FB beendet ist	(Ausgang ET 200S 1SI)
ERROR <sup>1</sup>	OUTPUT	BOOL	Auftrag fertig mit Fehler	Parameter STATUS enthält die Fehlerinformation.
STATUS <sup>1</sup>	OUTPUT	WORD	Spezifikation des Fehlers	Bei ERROR == 1 steht im Parameter STATUS die Fehlerinformation.
COM_RST	IN_OUT	BOOL	Neustart des FB	

<sup>1</sup> Diese Parameter stehen nach korrektem Auftrag **einen** CPU-Zyklus lang zur Verfügung!

### Anlauf

Der Parameter COM\_RST des FB S\_VSET dient dazu, dem FB einen Anlauf mitzuteilen.

Setzen Sie den Parameter COM\_RST im Anlauf-OB auf 1.

Rufen Sie den FB im zyklischen Betrieb auf, ohne den Parameter COM\_RST zu setzen oder rückzusetzen.

Wenn der Parameter COM\_RST gesetzt ist,

- ermittelt der FB Informationen über das Modul ET 200S 1SI (Anzahl Byte im Peripheriebereich, in dezentraler Peripherie oder nicht).
- setzt sich der FB zurück und beendet dabei evtl. einen zuvor (vor dem letzten Stopp-Übergang der CPU) begonnenen Auftrag.

Wenn der FB die Information über das Modul ET 200S 1SI ermittelt hat, setzt er den Parameter COM\_RST selbst zurück.

## 2.11 Anlaufeigenschaften und Betriebszustände

### Betriebszustände der seriellen Schnittstellenbaugruppe ET 200S 1SI

Die Baugruppe ET 200S 1SI verfügt über folgende Betriebszustände:

- **STOP:** Im Zustand STOP der Baugruppe ET 200S 1SI ist kein Protokoll-Treiber aktiv, alle Sende- und Empfangsaufträge von der CPU werden negativ quittiert. Die Baugruppe ET 200S 1SI bleibt im Betriebszustand STOP, bis die Ursache für den STOP beseitigt ist (z.B. Drahtbruch oder ungültiger Parameter).

- **Neuparametrierung:** Wenn Sie die Baugruppe ET 200S 1SI neu parametrieren, wird der Protokolltreiber initialisiert. Während der Neuparametrierung ist die SF-LED an.

Es ist kein Sende- und Empfangsbetrieb möglich, in der Baugruppe ET 200S 1SI gespeicherte Sende- und Empfangstelegramme gehen wegen des Treiber-Neustarts verloren. Die Kommunikation zwischen der Baugruppe ET 200S 1SI und der CPU wird neu gestartet (laufende Telegramme werden abgebrochen).

Zum Abschluss der Neuparametrierung befindet sich die Baugruppe ET 200S 1SI im Betriebszustand RUN und ist sende- und empfangsbereit.

- **RUN:** Die Baugruppe ET 200S 1SI bearbeitet die Sendeaufträge der CPU. Die vom Kommunikationspartner empfangenen Telegramme werden zur Abholung durch die CPU bereitgestellt.

### Anlaufeigenschaften der Baugruppe ET 200S 1SI

Der Anlauf besteht aus zwei Phasen:

- **Initialisierung:** Sobald die Baugruppe ET 200S 1SI an Spannung liegt, wird die serielle Schnittstelle initialisiert und wartet auf Parametrierungsdaten von der CPU.
- **Parametrierung:** Bei der Parametrierung empfängt die Baugruppe ET 200S 1SI die Baugruppenparameter, die dem aktuellen Steckplatz mit STEP 7 zugeordnet wurden.

### Verhalten der Baugruppe ET 200S 1SI bei Betriebszustandsübergängen der CPU

Nach Anlauf der Baugruppe ET 200S 1SI werden alle Daten zwischen CPU und ET 200S 1SI über die Funktionsbausteine ausgetauscht.

- **CPU-STOP:** Im Betriebszustand CPU-STOP ist die Kommunikation über PROFIBUS nicht möglich. Eine laufende Datenübertragung zwischen der Baugruppe und der CPU, sowohl Sende- als auch Empfangsauftrag, wird abgebrochen und ein Neustart der Verbindung eingeleitet.

Der Datenverkehr an der RS-232C-Schnittstelle der Baugruppe ET 200S 1SI wird beim ASCII-Treiber bei Parametrierung ohne Flusskontrolle fortgesetzt, d.h. der laufende Sendeauftrag wird noch beendet. Empfangstelegramme werden beim ASCII-Treiber solange empfangen, bis der Empfangspuffer voll ist.

- **CPU-Anlauf:** Beim Anlauf überträgt die CPU Parameter an die Baugruppe ET 200S 1SI.

Durch entsprechende Parametrierung können Sie den Empfangspuffer der ET 200 S 1SI im CPU-Anlauf automatisch löschen.

- **CPU-RUN:** Im Zustand RUN der CPU ist ein uneingeschränkter Sende- und Empfangsbetrieb möglich. In den ersten FB-Durchläufen nach CPU-Neustart werden die Baugruppe ET 200S 1SI und die jeweiligen FBs synchronisiert. Erst danach wird ein neuer FB S\_SEND oder S\_RCV ausgeführt.

### Besonderheiten beim Senden von Telegrammen

Telegramme können nur im Betriebszustand CPU-RUN gesendet werden.

Schaltet die CPU während der Datenübertragung von der CPU zur Baugruppe in den Betriebszustand STOP, meldet der FB S\_SEND den Fehler (05) 02<sub>H</sub> nach dem Wiederanlauf. Zum Verhindern dieses Verhaltens kann das Anwenderprogramm den FB S\_SEND mit dem Eingang RESET aus dem Anlauf-OB aufrufen.

---

#### Hinweis

Die Baugruppe ET 200S 1SI sendet Daten erst dann zum Kommunikationspartner, wenn sie alle Daten von der CPU erhalten hat.

---

### Besonderheiten beim Empfangen von Telegrammen

Mit STEP 7 können Sie "Baugruppen-Empfangspuffer im Anlauf löschen = ja/nein". parametrieren.

- Haben Sie "ja" parametriert, wird der Empfangspuffer der Baugruppe ET 200S 1SI beim Übergang von STOP nach RUN der CPU automatisch gelöscht.
- Haben Sie "nein" parametriert, werden so viele Telegramme im Empfangspuffer der Baugruppe ET 200S 1SI gepuffert, wie Sie parametriert haben.

Schaltet die CPU während der Datenübertragung von der CPU zur Baugruppe ET 200S 1SI in den Betriebszustand STOP, meldet der FB den Fehler (05) 02<sub>H</sub> nach dem Wiederanlauf. Zum Verhindern dieses Verhaltens kann das Anwenderprogramm den FB S\_SEND mit dem Eingang RESET aus dem Anlauf-OB aufrufen. Bei "ET 200S 1SI-Empfangspuffer im Anlauf löschen = nein" wird das Telegramm erneut von der Baugruppe an die CPU übertragen.

### Dynamischer Telegrammpuffer

Wählen Sie, ob nur ein Telegramm gepuffert wird oder ob Telegramme dynamisch gepuffert werden. Wenn Sie das Kontrollkästchen "aktivieren" wählen, kann die Baugruppe verschiedene Telegramme unterschiedlicher Länge puffern. Der Puffer ist ein Ringpuffer. Wenn der Puffer voll ist, wird die älteste Meldung überschrieben, es sei denn, Sie haben "Telegrammpuffer nicht überschreiben" aktiviert. In diesem Fall wird die neueste Meldung verworfen. Wenn eine Meldung überschrieben wird, führt das zu einem Diagnosealarm, der den Datenverlust anzeigt.

## 2.12 Referenzdaten für andere Master als S7-PROFIBUS

### 2.12.1 Grundlagen zu Referenzdaten

#### Datenaustausch zwischen dem Master und der Baugruppe ET 200S 1SI

Die Baugruppe ET 200S 1SI ist für die Datenübertragung von 4, 8 oder 32 Byte, Eingang oder Ausgang, mit Konsistenz auf der gesamten Länge konfiguriert. Die Baugruppe ET 200S 1SI nutzt den 4-, 8- bzw. 32-Byte-Eingangs-/Ausgangsspeicher für die Datenübertragung an die und aus der CPU über PROFIBUS-DP-Übertragungsmedien.

Die CPU kann jederzeit auf folgende Art Daten in die Ein- und Ausgänge schreiben und auch Daten aus den Ein- und Ausgängen lesen:

- Die CPU setzt im ersten Byte des Ausgangsspeichers der Baugruppe einen Auftrag an die Baugruppe ET 200S 1SI ab.
- Die Baugruppe ET 200S 1SI nimmt den Auftrag an, indem es den Auftragscode in den Eingangsspeicher überträgt.
- Die CPU tauscht Daten über Segmente aus 3, 7 oder 31Byte aus (so viele Segmente wie entsprechend der E/A-Größe erforderlich sind), bis alle Daten des Auftrags übertragen sind.

Das erste Byte des Segments ist ein Koordinationsbyte, das dazu dient, die Übertragung des jeweiligen Segments zwischen der CPU und der Baugruppe ET 200S 1SI zu synchronisieren (siehe folgendes Bild). Die übrigen Bytes des E/A-Speichers enthalten die Daten des Auftrags.

Die CPU überträgt Daten an die Baugruppe ET 200S SI folgendermaßen:

Byte	Inhalt
0	Koordinationsbyte
1	Datenbyte 0
2	Datenbyte 1
•	•
•	•
•	•
•	•
N	Datenbyte n



Die Baugruppe ET 200S SI überträgt Daten an die CPU folgendermaßen:

Byte	Inhalt
0	Koordinationsbyte
1	Datenbyte 0
2	Datenbyte 1
•	•
•	•
•	•
•	•
N	Datenbyte n



n = 3, 7 oder 31, je nach der in der Konfiguration ausgewählten Baugruppenvariante

Bild 2-22 Datenaustausch zwischen der CPU und der Baugruppe ET200S 1SI

### Beschreibung des Koordinationsbyte

Die folgende Tabelle beschreibt den Inhalt des Koordinationsbyte (Byte 0), das die Datenübertragung zwischen der CPU und der seriellen Schnittstellenbaugruppe ET 200S 1SI synchronisiert.

Tabelle 2- 21 Inhalt des Koordinationsbyte 0 für die Datenübertragung

Bytesegment	Beschreibung																
<b>Von CPU geschriebenes Auftragsbyte</b>	<table border="1" style="width: 100%; text-align: center;"> <tr> <td>Bit 7</td> <td>6</td> <td>5</td> <td>4</td> <td>3</td> <td>2</td> <td>1</td> <td>0</td> </tr> <tr> <td>Res.</td> <td colspan="3">Auftragscode</td> <td>Fehler</td> <td colspan="3">Ablaufnummer</td> </tr> </table>	Bit 7	6	5	4	3	2	1	0	Res.	Auftragscode			Fehler	Ablaufnummer		
Bit 7	6	5	4	3	2	1	0										
Res.	Auftragscode			Fehler	Ablaufnummer												
Bit 7	Reserviert für Spezialanwendungen des FB S_SEND. Für Auswertungen des Koordinations-Bytes müssen Sie dieses Bit ausblenden.																
Auftragscode	Von CPU gesetzt, um einen Auftrag zu initiieren.																
Ablaufnummer	<p><b>Auftrag senden:</b> Wird von der CPU um 1 erhöht, wenn die CPU ein weiteres Segment an die Baugruppe ET 200S 1SI sendet... <i>oder</i></p> <p><b>Auftrag empfangen:</b> Wird jedesmal vom Eingangsbyte 0 der CPU übernommen, wenn die CPU von der Schnittstellenbaugruppe ein neues Segment in der richtigen Reihenfolge empfängt. Zeigt die letzte gültige Ablaufnummer an, wenn das Fehlerbit gesetzt wird. (Wert geht von 1 auf 7).</p>																
Fehler	Wird von der CPU gesetzt, um anzuzeigen, dass ein Segment nicht in der richtigen Reihenfolge empfangen wurde. Das Feld Ablaufnummer zeigt die letzte gültige Ablaufnummer an.																
<b>Von der Baugruppe ET 200S 1SI geschriebenes Auftragsbyte</b>	<table border="1" style="width: 100%; text-align: center;"> <tr> <td>Bit 7</td> <td>6</td> <td>5</td> <td>4</td> <td>3</td> <td>2</td> <td>1</td> <td>0</td> </tr> <tr> <td>Res.</td> <td colspan="3">Auftragscode</td> <td>Fehler</td> <td colspan="3">Ablaufnummer</td> </tr> </table>	Bit 7	6	5	4	3	2	1	0	Res.	Auftragscode			Fehler	Ablaufnummer		
Bit 7	6	5	4	3	2	1	0										
Res.	Auftragscode			Fehler	Ablaufnummer												
Bit 7	Reserviert für Spezialanwendungen des FB S_SEND. Für Auswertungen des Koordinations-Bytes müssen Sie dieses Bit ausblenden.																
Auftragscode	Wird von der Baugruppe ET 200S 1SI übernommen, um zu quittieren, dass der Auftrag angenommen wurde.																
Ablaufnummer	<p><b>Auftrag senden:</b> Wird jedesmal vom Ausgangsbyte 0 der Baugruppe übernommen, wenn die Baugruppe von der CPU ein neues Segment in der richtigen Reihenfolge empfängt. Zeigt die letzte gültige Ablaufnummer an, wenn das Fehlerbit gesetzt wird.</p> <p><b>Auftrag empfangen:</b> Wird von der Baugruppe um 1 erhöht, wenn die Baugruppe ein weiteres Segment an die CPU sendet. (Wert geht von 1 auf 7).</p>																
Fehler	<p>Der Sender überwacht das Fehlerbit des Empfängers auf eine segmentierte Transaktion. Falls das Fehlerbit gesetzt ist:</p> <ul style="list-style-type: none"> <li><b>Sender CPU (Sendeauftrag):</b> sendet die CPU die Segmente erneut, wobei mit dem nächsten Segment nach der vom Empfänger gemeldeten Nummer begonnen wird.</li> <li><b>Sender Modul (Empfangsauftrag):</b> Das 1SI Modul bricht die weitere Übertragung des Rx-Telegramms zum Anwender mit der Fehlermeldung 0x0551 im Statuswort ab. Das Modul wartet auf die Quittierung dieser Fehlermeldung (Idle). Nach Beendigung dieser laufenden Fehlersequenz wird das abgebrochene Rx-Telegramm dem Anwender wieder gemeldet bzw. zur Abholung zur Verfügung gestellt.</li> </ul>																



### Definitionen der Auftragscodes

Die folgende Tabelle führt die Aufträge entsprechend der Zuordnung der Bits 4 bis 6 im Koordinationsbyte 0 auf.

Tabelle 2- 22 Auftragscodes

Bits 6 5 4	Hex.wert	Definition
0 0 0	0 <sub>H</sub>	Ruhezustand
0 0 1	1 <sub>H</sub>	Senden
0 1 0	2 <sub>H</sub>	Empfangen
0 1 1	3 <sub>H</sub>	V.24-Signalstatus lesen
1 0 0	4 <sub>H</sub>	V.24-Signale schreiben
1 0 1	5 <sub>H</sub>	Parameter übertragen: Mit diesem Auftrag können Sie zusätzliche Parameter einstellen, die nicht in der GSD-Datei angegeben sind.
1 1 0	6 <sub>H</sub>	Reserviert
1 1 1	7 <sub>H</sub>	Auftragsendequittung

### Regeln zum Schreiben von Auftragscodes

Folgende Regeln gelten zum Schreiben von Auftragscodes im Koordinationsbyte, wodurch die CPU und die Baugruppe ET 200S 1SI die Datenübertragungen synchronisieren können:

- Bevor das Anwenderprogramm der CPU einen Auftragscode in das Ausgangs-Koordinationsbyte schreiben kann, muss es einen Ruhecode aus dem Eingangs-Koordinationsbyte der Baugruppe ET 200S 1SI sehen.
- Bevor das Anwenderprogramm der CPU dann das erste Segment in das Ausgangsbyte 1 ... n schreiben kann, muss es den Auftragsquittungscode (d.h. den angenommenen Auftragscode) im Eingangs-Koordinationsbyte der Baugruppe sehen.
- Sieht das Anwenderprogramm andere Auftragsquittungscodes als den vom Programm gesendeten, dann darf es nicht in das Ausgangsbyte 0 .. n schreiben, bis es erneut einen Ruhecode vom Eingangs-Koordinationsbyte der Baugruppe ET 200S 1SI gesehen hat.

Diese Situation kann beispielsweise auftreten, wenn zwei getrennte Aufträge im selben Zyklus ausgeführt werden, beide Aufträge sehen den Ruhecode und beide schreiben einen anderen Auftragscode in das Ausgangsbyte. Aufgrund des asynchronen Zyklus zwischen dem CPU-Zyklus und dem PROFIBUS-DP-Zyklus ist nicht sichergestellt, dass der Auftrag die Baugruppe zuerst erreicht. Deshalb muss jeder Auftrag auf das Ende des anderen Auftrags warten können, bevor er selbst bearbeitet wird.

### Empfangsstatus des 1SI Moduls

Das 1SI Modul zeigt seinen Empfangsstatus immer dann an, wenn es im Ruhezustand ist (Auftragsquittungs-Byte 0 = 00H). Der Empfangsstatus ist in den Bytes 1 und 2 hinterlegt.

Status	Bedeutung
0000 <sub>H</sub>	Keine empfangene Meldung verfügbar
0001 <sub>H</sub>	Empfangene Meldung bzw. Empfangstelegramm verfügbar
0B01 <sub>H</sub>	Der Empfangspuffer ist zu mehr als 2/3 gefüllt.

### Definitionen der Statusworte

In den Beispielen für Datenübertragungen auf den folgenden Seiten verwendet die Baugruppe ET 200S 1SI in einigen Antworten an die CPU die Bytes 1 und 2 für die Statusmeldung. Die Tabelle "Diagnosemeldungen im Parameter STATUS" listet die Statusworte samt Definitionen auf.

### Reihenfolge der Bytes im Wort

Bei den Datenübertragungen zwischen der CPU und der Baugruppe ET 200S 1SI wird bei allen 16-Bit-Worten (z.B. Status und Länge) das höchstwertige Byte zuerst gesendet.

### Empfangsstatus des 1SI Moduls

Der Zustand des Empfangspuffers des 1SI Moduls wird dem Anwender immer dann angezeigt, wenn sich das Modul im Ruhezustand (Auftragsquittung Byte 0 = 00H befindet. Der Status ist dann in den Bytes 1+2 hinterlegt.

Status	Bedeutung
0000H	Keine empfangene Meldung verfügbar
0001H	Empfangene Meldung bzw. Empfangstelegramm verfügbar
0B01H	Empfangspuffer ist mehr als 2/3 voll.

## 2.12.2 Beispielablauf beim Senden von Daten von der CPU zur Baugruppe

### Beispielablauf

Die folgende Tabelle zeigt ein Beispiel für eine CPU, die eine Meldung mit den ersten 22 Zeichen des Alphabets sendet. Der E/A-Speicher beträgt 8 Byte. Der DP-Zyklus ist ungefähr gleich dem CPU-Zyklus, so dass es zu einer Latenzzeit von einem Zyklus kommt, wenn die Baugruppe mit der Ablaufnummer antwortet.

Tabelle 2- 23 Beispielablauf beim Senden

CPU-Zyklus	CPU schreibt in ET 200S 1SI	CPU liest ET 200S 1SI																											
1.	Anwenderprogramm sieht den folgenden Ruhecode der Baugruppe: <table border="1" style="margin-left: auto; margin-right: auto;"> <tr> <td>Byte</td> <td>0</td> <td>1</td> <td>2</td> <td>3</td> <td>4</td> <td>5</td> <td>6</td> <td>7</td> </tr> <tr> <td></td> <td>00<sub>H</sub></td> <td>nnnn<sub>H</sub></td> <td>xx<sub>H</sub></td> <td>xx<sub>H</sub></td> <td>xx<sub>H</sub></td> <td>xx<sub>H</sub></td> <td>xx<sub>H</sub></td> <td>xx<sub>H</sub></td> </tr> <tr> <td>←</td> <td>Auft.quitt.</td> <td>Status</td> <td colspan="6">Irrelevant</td> </tr> </table>	Byte	0	1	2	3	4	5	6	7		00 <sub>H</sub>	nnnn <sub>H</sub>	xx <sub>H</sub>	xx <sub>H</sub>	xx <sub>H</sub>	xx <sub>H</sub>	xx <sub>H</sub>	xx <sub>H</sub>	←	Auft.quitt.	Status	Irrelevant						
	Byte	0	1	2	3	4	5	6	7																				
	00 <sub>H</sub>	nnnn <sub>H</sub>	xx <sub>H</sub>	xx <sub>H</sub>	xx <sub>H</sub>	xx <sub>H</sub>	xx <sub>H</sub>	xx <sub>H</sub>																					
←	Auft.quitt.	Status	Irrelevant																										
	CPU schreibt Auftrag zum Senden: <table border="1" style="margin-left: auto; margin-right: auto;"> <tr> <td>Byte</td> <td>0</td> <td>1</td> <td>2</td> <td>3</td> <td>4</td> <td>5</td> <td>6</td> <td>7</td> </tr> <tr> <td></td> <td>10<sub>H</sub></td> <td>xx<sub>H</sub></td> <td>xx<sub>H</sub></td> <td>xx<sub>H</sub></td> <td>xx<sub>H</sub></td> <td>xx<sub>H</sub></td> <td>xx<sub>H</sub></td> <td>xx<sub>H</sub></td> </tr> <tr> <td></td> <td>Auftrag</td> <td colspan="7">Irrelevant</td> </tr> </table>	Byte	0	1	2	3	4	5	6	7		10 <sub>H</sub>	xx <sub>H</sub>	xx <sub>H</sub>	xx <sub>H</sub>	xx <sub>H</sub>	xx <sub>H</sub>	xx <sub>H</sub>	xx <sub>H</sub>		Auftrag	Irrelevant							
Byte	0	1	2	3	4	5	6	7																					
	10 <sub>H</sub>	xx <sub>H</sub>	xx <sub>H</sub>	xx <sub>H</sub>	xx <sub>H</sub>	xx <sub>H</sub>	xx <sub>H</sub>	xx <sub>H</sub>																					
	Auftrag	Irrelevant																											
2.	Anwenderprogramm liest immer noch den Ruhecode der Baugruppe: <table border="1" style="margin-left: auto; margin-right: auto;"> <tr> <td>Byte</td> <td>0</td> <td>1</td> <td>2</td> <td>3</td> <td>4</td> <td>5</td> <td>6</td> <td>7</td> </tr> <tr> <td></td> <td>00<sub>H</sub></td> <td>nnnn<sub>H</sub></td> <td>xx<sub>H</sub></td> <td>xx<sub>H</sub></td> <td>xx<sub>H</sub></td> <td>xx<sub>H</sub></td> <td>xx<sub>H</sub></td> <td>xx<sub>H</sub></td> </tr> <tr> <td>←</td> <td>Auft.quitt.</td> <td>Status</td> <td colspan="6">Irrelevant</td> </tr> </table>	Byte	0	1	2	3	4	5	6	7		00 <sub>H</sub>	nnnn <sub>H</sub>	xx <sub>H</sub>	xx <sub>H</sub>	xx <sub>H</sub>	xx <sub>H</sub>	xx <sub>H</sub>	xx <sub>H</sub>	←	Auft.quitt.	Status	Irrelevant						
	Byte	0	1	2	3	4	5	6	7																				
	00 <sub>H</sub>	nnnn <sub>H</sub>	xx <sub>H</sub>	xx <sub>H</sub>	xx <sub>H</sub>	xx <sub>H</sub>	xx <sub>H</sub>	xx <sub>H</sub>																					
←	Auft.quitt.	Status	Irrelevant																										
	CPU wiederholt Auftrag zum Senden: <table border="1" style="margin-left: auto; margin-right: auto;"> <tr> <td>Byte</td> <td>0</td> <td>1</td> <td>2</td> <td>3</td> <td>4</td> <td>5</td> <td>6</td> <td>7</td> </tr> <tr> <td></td> <td>10<sub>H</sub></td> <td>xx<sub>H</sub></td> <td>xx<sub>H</sub></td> <td>xx<sub>H</sub></td> <td>xx<sub>H</sub></td> <td>xx<sub>H</sub></td> <td>xx<sub>H</sub></td> <td>xx<sub>H</sub></td> </tr> <tr> <td></td> <td>Auftrag</td> <td colspan="7">Irrelevant</td> </tr> </table>	Byte	0	1	2	3	4	5	6	7		10 <sub>H</sub>	xx <sub>H</sub>	xx <sub>H</sub>	xx <sub>H</sub>	xx <sub>H</sub>	xx <sub>H</sub>	xx <sub>H</sub>	xx <sub>H</sub>		Auftrag	Irrelevant							
Byte	0	1	2	3	4	5	6	7																					
	10 <sub>H</sub>	xx <sub>H</sub>	xx <sub>H</sub>	xx <sub>H</sub>	xx <sub>H</sub>	xx <sub>H</sub>	xx <sub>H</sub>	xx <sub>H</sub>																					
	Auftrag	Irrelevant																											
3.	Anwenderprogramm liest folgende Antwort der Baugruppe: <table border="1" style="margin-left: auto; margin-right: auto;"> <tr> <td>Byte</td> <td>0</td> <td>1</td> <td>2</td> <td>3</td> <td>4</td> <td>5</td> <td>6</td> <td>7</td> </tr> <tr> <td></td> <td>10<sub>H</sub></td> <td>nnnn<sub>H</sub></td> <td>xx<sub>H</sub></td> <td>xx<sub>H</sub></td> <td>xx<sub>H</sub></td> <td>xx<sub>H</sub></td> <td>xx<sub>H</sub></td> <td>xx<sub>H</sub></td> </tr> <tr> <td>←</td> <td>Auft.quitt.</td> <td>Status</td> <td colspan="6">Irrelevant</td> </tr> </table>	Byte	0	1	2	3	4	5	6	7		10 <sub>H</sub>	nnnn <sub>H</sub>	xx <sub>H</sub>	xx <sub>H</sub>	xx <sub>H</sub>	xx <sub>H</sub>	xx <sub>H</sub>	xx <sub>H</sub>	←	Auft.quitt.	Status	Irrelevant						
	Byte	0	1	2	3	4	5	6	7																				
	10 <sub>H</sub>	nnnn <sub>H</sub>	xx <sub>H</sub>	xx <sub>H</sub>	xx <sub>H</sub>	xx <sub>H</sub>	xx <sub>H</sub>	xx <sub>H</sub>																					
←	Auft.quitt.	Status	Irrelevant																										
	CPU sendet 1. Segment: <table border="1" style="margin-left: auto; margin-right: auto;"> <tr> <td>Byte</td> <td>0</td> <td>1</td> <td>2</td> <td>3</td> <td>4</td> <td>5</td> <td>6</td> <td>7</td> </tr> <tr> <td></td> <td>11<sub>H</sub></td> <td>0016<sub>H</sub></td> <td>'a'</td> <td>'b'</td> <td>'c'</td> <td>'d'</td> <td>'e'</td> <td></td> </tr> <tr> <td></td> <td>Auftrag</td> <td>Sendelänge</td> <td colspan="6">Daten</td> </tr> </table>	Byte	0	1	2	3	4	5	6	7		11 <sub>H</sub>	0016 <sub>H</sub>	'a'	'b'	'c'	'd'	'e'			Auftrag	Sendelänge	Daten						
Byte	0	1	2	3	4	5	6	7																					
	11 <sub>H</sub>	0016 <sub>H</sub>	'a'	'b'	'c'	'd'	'e'																						
	Auftrag	Sendelänge	Daten																										
4.	Anwenderprogramm liest folgende Antwort der Baugruppe: <table border="1" style="margin-left: auto; margin-right: auto;"> <tr> <td>Byte</td> <td>0</td> <td>1</td> <td>2</td> <td>3</td> <td>4</td> <td>5</td> <td>6</td> <td>7</td> </tr> <tr> <td></td> <td>10<sub>H</sub></td> <td>xx<sub>H</sub></td> <td>xx<sub>H</sub></td> <td>xx<sub>H</sub></td> <td>xx<sub>H</sub></td> <td>xx<sub>H</sub></td> <td>xx<sub>H</sub></td> <td>xx<sub>H</sub></td> </tr> <tr> <td>←</td> <td>Auft.quitt.</td> <td colspan="7">Irrelevant</td> </tr> </table>	Byte	0	1	2	3	4	5	6	7		10 <sub>H</sub>	xx <sub>H</sub>	xx <sub>H</sub>	xx <sub>H</sub>	xx <sub>H</sub>	xx <sub>H</sub>	xx <sub>H</sub>	xx <sub>H</sub>	←	Auft.quitt.	Irrelevant							
	Byte	0	1	2	3	4	5	6	7																				
	10 <sub>H</sub>	xx <sub>H</sub>	xx <sub>H</sub>	xx <sub>H</sub>	xx <sub>H</sub>	xx <sub>H</sub>	xx <sub>H</sub>	xx <sub>H</sub>																					
←	Auft.quitt.	Irrelevant																											
	CPU wiederholt 1. Segment: <table border="1" style="margin-left: auto; margin-right: auto;"> <tr> <td>Byte</td> <td>0</td> <td>1</td> <td>2</td> <td>3</td> <td>4</td> <td>5</td> <td>6</td> <td>7</td> </tr> <tr> <td></td> <td>11<sub>H</sub></td> <td>'f'</td> <td>'g'</td> <td>'h'</td> <td>'i'</td> <td>'j'</td> <td>'k'</td> <td>'l'</td> </tr> <tr> <td></td> <td>Auftrag</td> <td colspan="7">Daten</td> </tr> </table>	Byte	0	1	2	3	4	5	6	7		11 <sub>H</sub>	'f'	'g'	'h'	'i'	'j'	'k'	'l'		Auftrag	Daten							
Byte	0	1	2	3	4	5	6	7																					
	11 <sub>H</sub>	'f'	'g'	'h'	'i'	'j'	'k'	'l'																					
	Auftrag	Daten																											

CPU-Zyklus	CPU schreibt in ET 200S 1S1	CPU liest ET 200S 1S1																
5.	Anwenderprogramm liest folgende Antwort der Baugruppe:																	
	←	<table border="1"> <tr> <td>11<sub>H</sub></td> <td>xx<sub>H</sub></td> <td>xx<sub>H</sub></td> <td>xx<sub>H</sub></td> <td>xx<sub>H</sub></td> <td>xx<sub>H</sub></td> <td>xx<sub>H</sub></td> <td>xx<sub>H</sub></td> </tr> <tr> <td>Auft.quitt.</td> <td colspan="7">Irrelevant</td> </tr> </table>	11 <sub>H</sub>	xx <sub>H</sub>	xx <sub>H</sub>	xx <sub>H</sub>	xx <sub>H</sub>	xx <sub>H</sub>	xx <sub>H</sub>	xx <sub>H</sub>	Auft.quitt.	Irrelevant						
11 <sub>H</sub>	xx <sub>H</sub>	xx <sub>H</sub>	xx <sub>H</sub>	xx <sub>H</sub>	xx <sub>H</sub>	xx <sub>H</sub>	xx <sub>H</sub>											
Auft.quitt.	Irrelevant																	
CPU sendet 2. Segment, da kein Fehler angezeigt wurde und der Ablauf in Ordnung ist:																		
<table border="1"> <tr> <td>12<sub>H</sub></td> <td>'m'</td> <td>'n'</td> <td>'o'</td> <td>'p'</td> <td>'q'</td> <td>'r'</td> <td>'s'</td> </tr> <tr> <td>Auftrag</td> <td colspan="7">Daten</td> </tr> </table> <span style="float: right;">→</span>			12 <sub>H</sub>	'm'	'n'	'o'	'p'	'q'	'r'	's'	Auftrag	Daten						
12 <sub>H</sub>	'm'	'n'	'o'	'p'	'q'	'r'	's'											
Auftrag	Daten																	
6.	Anwenderprogramm liest folgende Antwort der Baugruppe:																	
	←	<table border="1"> <tr> <td>12<sub>H</sub></td> <td>xx<sub>H</sub></td> <td>xx<sub>H</sub></td> <td>xx<sub>H</sub></td> <td>xx<sub>H</sub></td> <td>xx<sub>H</sub></td> <td>xx<sub>H</sub></td> <td>xx<sub>H</sub></td> </tr> <tr> <td>Auft.quitt.</td> <td colspan="7">Irrelevant</td> </tr> </table>	12 <sub>H</sub>	xx <sub>H</sub>	xx <sub>H</sub>	xx <sub>H</sub>	xx <sub>H</sub>	xx <sub>H</sub>	xx <sub>H</sub>	xx <sub>H</sub>	Auft.quitt.	Irrelevant						
12 <sub>H</sub>	xx <sub>H</sub>	xx <sub>H</sub>	xx <sub>H</sub>	xx <sub>H</sub>	xx <sub>H</sub>	xx <sub>H</sub>	xx <sub>H</sub>											
Auft.quitt.	Irrelevant																	
CPU sendet 3. Segment, da kein Fehler angezeigt wurde und der Ablauf in Ordnung ist:																		
<table border="1"> <tr> <td>13<sub>H</sub></td> <td>'t'</td> <td>'u'</td> <td>'v'</td> <td>xx<sub>H</sub></td> <td>xx<sub>H</sub></td> <td>xx<sub>H</sub></td> <td>xx<sub>H</sub></td> </tr> <tr> <td>Auftrag</td> <td colspan="2">Daten</td> <td colspan="5">Irrelevant</td> </tr> </table> <span style="float: right;">→</span>			13 <sub>H</sub>	't'	'u'	'v'	xx <sub>H</sub>	xx <sub>H</sub>	xx <sub>H</sub>	xx <sub>H</sub>	Auftrag	Daten		Irrelevant				
13 <sub>H</sub>	't'	'u'	'v'	xx <sub>H</sub>	xx <sub>H</sub>	xx <sub>H</sub>	xx <sub>H</sub>											
Auftrag	Daten		Irrelevant															
7.	Anwenderprogramm liest folgende Antwort der Baugruppe:																	
	←	<table border="1"> <tr> <td>13<sub>H</sub></td> <td>xx<sub>H</sub></td> <td>xx<sub>H</sub></td> <td>xx<sub>H</sub></td> <td>xx<sub>H</sub></td> <td>xx<sub>H</sub></td> <td>xx<sub>H</sub></td> <td>xx<sub>H</sub></td> </tr> <tr> <td>Auft.quitt.</td> <td colspan="7">Irrelevant</td> </tr> </table>	13 <sub>H</sub>	xx <sub>H</sub>	xx <sub>H</sub>	xx <sub>H</sub>	xx <sub>H</sub>	xx <sub>H</sub>	xx <sub>H</sub>	xx <sub>H</sub>	Auft.quitt.	Irrelevant						
13 <sub>H</sub>	xx <sub>H</sub>	xx <sub>H</sub>	xx <sub>H</sub>	xx <sub>H</sub>	xx <sub>H</sub>	xx <sub>H</sub>	xx <sub>H</sub>											
Auft.quitt.	Irrelevant																	
CPU sendet 4. Segment, da kein Fehler angezeigt wurde und der Ablauf in Ordnung ist:																		
<table border="1"> <tr> <td>14<sub>H</sub></td> <td>'t'</td> <td>'u'</td> <td>'v'</td> <td>xx<sub>H</sub></td> <td>xx<sub>H</sub></td> <td>xx<sub>H</sub></td> <td>xx<sub>H</sub></td> </tr> <tr> <td>Auftrag</td> <td colspan="2">Daten</td> <td colspan="5">Irrelevant</td> </tr> </table> <span style="float: right;">→</span>			14 <sub>H</sub>	't'	'u'	'v'	xx <sub>H</sub>	xx <sub>H</sub>	xx <sub>H</sub>	xx <sub>H</sub>	Auftrag	Daten		Irrelevant				
14 <sub>H</sub>	't'	'u'	'v'	xx <sub>H</sub>	xx <sub>H</sub>	xx <sub>H</sub>	xx <sub>H</sub>											
Auftrag	Daten		Irrelevant															
8.	Anwenderprogramm liest folgende Antwort der Baugruppe:																	
	←	<table border="1"> <tr> <td>13<sub>H</sub></td> <td>xx<sub>H</sub></td> <td>xx<sub>H</sub></td> <td>xx<sub>H</sub></td> <td>xx<sub>H</sub></td> <td>xx<sub>H</sub></td> <td>xx<sub>H</sub></td> <td>xx<sub>H</sub></td> </tr> <tr> <td>Auft.quitt.</td> <td colspan="7">Irrelevant</td> </tr> </table>	13 <sub>H</sub>	xx <sub>H</sub>	xx <sub>H</sub>	xx <sub>H</sub>	xx <sub>H</sub>	xx <sub>H</sub>	xx <sub>H</sub>	xx <sub>H</sub>	Auft.quitt.	Irrelevant						
13 <sub>H</sub>	xx <sub>H</sub>	xx <sub>H</sub>	xx <sub>H</sub>	xx <sub>H</sub>	xx <sub>H</sub>	xx <sub>H</sub>	xx <sub>H</sub>											
Auft.quitt.	Irrelevant																	
CPU wartet beim 4. Segment auf die Quittung:																		
<table border="1"> <tr> <td>14<sub>H</sub></td> <td>'t'</td> <td>'u'</td> <td>'v'</td> <td>xx<sub>H</sub></td> <td>xx<sub>H</sub></td> <td>xx<sub>H</sub></td> <td>xx<sub>H</sub></td> </tr> <tr> <td>Auftrag</td> <td colspan="2">Daten</td> <td colspan="5">Irrelevant</td> </tr> </table> <span style="float: right;">→</span>			14 <sub>H</sub>	't'	'u'	'v'	xx <sub>H</sub>	xx <sub>H</sub>	xx <sub>H</sub>	xx <sub>H</sub>	Auftrag	Daten		Irrelevant				
14 <sub>H</sub>	't'	'u'	'v'	xx <sub>H</sub>	xx <sub>H</sub>	xx <sub>H</sub>	xx <sub>H</sub>											
Auftrag	Daten		Irrelevant															
9.	Anwenderprogramm liest folgende Antwort der Baugruppe:																	
	←	<table border="1"> <tr> <td>14<sub>H</sub></td> <td>xx<sub>H</sub></td> <td>xx<sub>H</sub></td> <td>xx<sub>H</sub></td> <td>xx<sub>H</sub></td> <td>xx<sub>H</sub></td> <td>xx<sub>H</sub></td> <td>xx<sub>H</sub></td> </tr> <tr> <td>Auft.quitt.</td> <td colspan="7">Irrelevant</td> </tr> </table>	14 <sub>H</sub>	xx <sub>H</sub>	xx <sub>H</sub>	xx <sub>H</sub>	xx <sub>H</sub>	xx <sub>H</sub>	xx <sub>H</sub>	xx <sub>H</sub>	Auft.quitt.	Irrelevant						
14 <sub>H</sub>	xx <sub>H</sub>	xx <sub>H</sub>	xx <sub>H</sub>	xx <sub>H</sub>	xx <sub>H</sub>	xx <sub>H</sub>	xx <sub>H</sub>											
Auft.quitt.	Irrelevant																	
CPU sendet nichts Neues (Ausgänge bleiben gleich) und wartet auf letzte Quittung der Baugruppe; es wird angezeigt, dass die Meldung an den Kommunikationspartner gesendet wurde:																		
<table border="1"> <tr> <td>14<sub>H</sub></td> <td>'t'</td> <td>'u'</td> <td>'v'</td> <td>xx<sub>H</sub></td> <td>xx<sub>H</sub></td> <td>xx<sub>H</sub></td> <td>xx<sub>H</sub></td> </tr> <tr> <td>Auftrag</td> <td colspan="2">Daten</td> <td colspan="5">Irrelevant</td> </tr> </table> <span style="float: right;">→</span>			14 <sub>H</sub>	't'	'u'	'v'	xx <sub>H</sub>	xx <sub>H</sub>	xx <sub>H</sub>	xx <sub>H</sub>	Auftrag	Daten		Irrelevant				
14 <sub>H</sub>	't'	'u'	'v'	xx <sub>H</sub>	xx <sub>H</sub>	xx <sub>H</sub>	xx <sub>H</sub>											
Auftrag	Daten		Irrelevant															
n.	Einige CPU-Zyklen später sieht das Anwenderprogramm folgende Antwort der Baugruppe:																	
	←	<table border="1"> <tr> <td>74<sub>H</sub></td> <td>nnnn<sub>H</sub></td> <td>xx<sub>H</sub></td> <td>xx<sub>H</sub></td> <td>xx<sub>H</sub></td> <td>xx<sub>H</sub></td> <td>xx<sub>H</sub></td> <td>xx<sub>H</sub></td> </tr> <tr> <td>Auft.quitt.</td> <td>Status</td> <td colspan="6">Irrelevant</td> </tr> </table>	74 <sub>H</sub>	nnnn <sub>H</sub>	xx <sub>H</sub>	xx <sub>H</sub>	xx <sub>H</sub>	xx <sub>H</sub>	xx <sub>H</sub>	xx <sub>H</sub>	Auft.quitt.	Status	Irrelevant					
74 <sub>H</sub>	nnnn <sub>H</sub>	xx <sub>H</sub>	xx <sub>H</sub>	xx <sub>H</sub>	xx <sub>H</sub>	xx <sub>H</sub>	xx <sub>H</sub>											
Auft.quitt.	Status	Irrelevant																
—	CPU schreibt Ruhecode in den Auftrag und beendet den Auftrag.																	

### 2.12.3 Beispielablauf beim Empfangen von Daten der Baugruppe in der CPU

#### Beispielablauf

Die folgende Tabelle zeigt ein Beispiel dafür, wie die CPU eine Meldung von der seriellen Schnittstellenbaugruppe empfängt. Der E/A-Speicher beträgt 8 Byte. Der DP-Zyklus ist kleiner als der CPU-Zyklus, so dass keine Latenzzeit in der Baugruppe entsteht.

Tabelle 2- 24 Beispielablauf beim Empfangen

CPU-Zyklus	CPU schreibt in ET 200S 1SI	CPU liest ET 200S 1SI																																																															
<b>n</b>	<p>In mehreren Zyklen liest das Anwenderprogramm den Ruhecode der Baugruppe, bis der Status anzeigt, dass eine empfangene Meldung zur Verfügung steht:</p> <table border="1"> <tr> <td>Byte</td> <td>0</td> <td>1</td> <td>2</td> <td>3</td> <td>4</td> <td>5</td> <td>6</td> <td>7</td> </tr> <tr> <td></td> <td>00<sub>H</sub></td> <td>nnnn<sub>H</sub></td> <td>xx<sub>H</sub></td> <td>xx<sub>H</sub></td> <td>xx<sub>H</sub></td> <td>xx<sub>H</sub></td> <td>xx<sub>H</sub></td> <td>xx<sub>H</sub></td> </tr> <tr> <td>←</td> <td>Auft. quitt.</td> <td>Status</td> <td colspan="6">Irrelevant</td> </tr> </table> <p>Status:                      0000<sub>H</sub> = Keine empfangene Meldung verfügbar.                      0001<sub>H</sub> = Empfangene Meldung verfügbar.                      0B01<sub>H</sub> = Empfangspuffer ist mehr als 2/3 voll.</p> <p>CPU schreibt Auftrag zum Empfangen:</p> <table border="1"> <tr> <td>Byte</td> <td>0</td> <td>1</td> <td>2</td> <td>3</td> <td>4</td> <td>5</td> <td>6</td> <td>7</td> </tr> <tr> <td></td> <td>20<sub>H</sub></td> <td>xx<sub>H</sub></td> <td>xx<sub>H</sub></td> <td>xx<sub>H</sub></td> <td>xx<sub>H</sub></td> <td>xx<sub>H</sub></td> <td>xx<sub>H</sub></td> <td>xx<sub>H</sub></td> </tr> <tr> <td></td> <td>Auftrag</td> <td colspan="7">Irrelevant</td> </tr> <tr> <td></td> <td></td> <td colspan="7">→</td> </tr> </table>	Byte	0	1	2	3	4	5	6	7		00 <sub>H</sub>	nnnn <sub>H</sub>	xx <sub>H</sub>	xx <sub>H</sub>	xx <sub>H</sub>	xx <sub>H</sub>	xx <sub>H</sub>	xx <sub>H</sub>	←	Auft. quitt.	Status	Irrelevant						Byte	0	1	2	3	4	5	6	7		20 <sub>H</sub>	xx <sub>H</sub>	xx <sub>H</sub>	xx <sub>H</sub>	xx <sub>H</sub>	xx <sub>H</sub>	xx <sub>H</sub>	xx <sub>H</sub>		Auftrag	Irrelevant									→							
Byte	0	1	2	3	4	5	6	7																																																									
	00 <sub>H</sub>	nnnn <sub>H</sub>	xx <sub>H</sub>	xx <sub>H</sub>	xx <sub>H</sub>	xx <sub>H</sub>	xx <sub>H</sub>	xx <sub>H</sub>																																																									
←	Auft. quitt.	Status	Irrelevant																																																														
Byte	0	1	2	3	4	5	6	7																																																									
	20 <sub>H</sub>	xx <sub>H</sub>	xx <sub>H</sub>	xx <sub>H</sub>	xx <sub>H</sub>	xx <sub>H</sub>	xx <sub>H</sub>	xx <sub>H</sub>																																																									
	Auftrag	Irrelevant																																																															
		→																																																															
<b>Nächster Zyklus (n + 1)</b>	<p>Das Anwenderprogramm liest folgende Antwort der Baugruppe (Baugruppe quittiert Empfang, antwortet mit dem ersten Segment und erhöht die Ablaufnummer):</p> <table border="1"> <tr> <td>←</td> <td>21<sub>H</sub></td> <td>0006<sub>H</sub></td> <td>'a'</td> <td>'b'</td> <td>'c'</td> <td>'d'</td> <td>'e'</td> </tr> <tr> <td></td> <td>Auft. quitt.</td> <td>Länge</td> <td colspan="5">Daten</td> </tr> </table> <p>CPU schreibt Auftrag, um das 1. Segment zu quittieren:</p> <table border="1"> <tr> <td>21<sub>H</sub></td> <td>xx<sub>H</sub></td> <td>xx<sub>H</sub></td> <td>xx<sub>H</sub></td> <td>xx<sub>H</sub></td> <td>xx<sub>H</sub></td> <td>xx<sub>H</sub></td> <td>xx<sub>H</sub></td> </tr> <tr> <td>Auftrag</td> <td colspan="7">Irrelevant</td> </tr> <tr> <td></td> <td colspan="7">→</td> </tr> </table>	←	21 <sub>H</sub>	0006 <sub>H</sub>	'a'	'b'	'c'	'd'	'e'		Auft. quitt.	Länge	Daten					21 <sub>H</sub>	xx <sub>H</sub>	xx <sub>H</sub>	xx <sub>H</sub>	xx <sub>H</sub>	xx <sub>H</sub>	xx <sub>H</sub>	xx <sub>H</sub>	Auftrag	Irrelevant								→																														
←	21 <sub>H</sub>	0006 <sub>H</sub>	'a'	'b'	'c'	'd'	'e'																																																										
	Auft. quitt.	Länge	Daten																																																														
21 <sub>H</sub>	xx <sub>H</sub>	xx <sub>H</sub>	xx <sub>H</sub>	xx <sub>H</sub>	xx <sub>H</sub>	xx <sub>H</sub>	xx <sub>H</sub>																																																										
Auftrag	Irrelevant																																																																
	→																																																																
<b>Nächster Zyklus (n + 2)</b>	<p>Anwenderprogramm liest das 2. Segment der Baugruppe:</p> <table border="1"> <tr> <td>←</td> <td>22<sub>H</sub></td> <td>'f'</td> <td>xx<sub>H</sub></td> <td>xx<sub>H</sub></td> <td>xx<sub>H</sub></td> <td>xx<sub>H</sub></td> <td>xx<sub>H</sub></td> <td>xx<sub>H</sub></td> </tr> <tr> <td></td> <td>Auft. quitt.</td> <td>Daten</td> <td colspan="5">Irrelevant</td> </tr> </table> <p>CPU schreibt Auftrag, um das 2. Segment zu quittieren:</p> <table border="1"> <tr> <td>22<sub>H</sub></td> <td>xx<sub>H</sub></td> <td>xx<sub>H</sub></td> <td>xx<sub>H</sub></td> <td>xx<sub>H</sub></td> <td>xx<sub>H</sub></td> <td>xx<sub>H</sub></td> <td>xx<sub>H</sub></td> </tr> <tr> <td>Auftrag</td> <td colspan="7">Irrelevant</td> </tr> <tr> <td></td> <td colspan="7">→</td> </tr> </table>	←	22 <sub>H</sub>	'f'	xx <sub>H</sub>	xx <sub>H</sub>	xx <sub>H</sub>	xx <sub>H</sub>	xx <sub>H</sub>	xx <sub>H</sub>		Auft. quitt.	Daten	Irrelevant					22 <sub>H</sub>	xx <sub>H</sub>	xx <sub>H</sub>	xx <sub>H</sub>	xx <sub>H</sub>	xx <sub>H</sub>	xx <sub>H</sub>	xx <sub>H</sub>	Auftrag	Irrelevant								→																													
←	22 <sub>H</sub>	'f'	xx <sub>H</sub>	xx <sub>H</sub>	xx <sub>H</sub>	xx <sub>H</sub>	xx <sub>H</sub>	xx <sub>H</sub>																																																									
	Auft. quitt.	Daten	Irrelevant																																																														
22 <sub>H</sub>	xx <sub>H</sub>	xx <sub>H</sub>	xx <sub>H</sub>	xx <sub>H</sub>	xx <sub>H</sub>	xx <sub>H</sub>	xx <sub>H</sub>																																																										
Auftrag	Irrelevant																																																																
	→																																																																
<b>Nächster Zyklus (n + 3)</b>	<p>Baugruppe kehrt in Ruhezustand zurück, nachdem die erste Empfangstransaktion beendet ist.</p> <table border="1"> <tr> <td>←</td> <td>00<sub>H</sub></td> <td>nnnn<sub>H</sub></td> <td>xx<sub>H</sub></td> <td>xx<sub>H</sub></td> <td>xx<sub>H</sub></td> <td>xx<sub>H</sub></td> <td>xx<sub>H</sub></td> </tr> <tr> <td></td> <td>Auft. quitt.</td> <td>StatusUS</td> <td colspan="5">Irrelevant</td> </tr> </table>	←	00 <sub>H</sub>	nnnn <sub>H</sub>	xx <sub>H</sub>	xx <sub>H</sub>	xx <sub>H</sub>	xx <sub>H</sub>	xx <sub>H</sub>		Auft. quitt.	StatusUS	Irrelevant																																																				
←	00 <sub>H</sub>	nnnn <sub>H</sub>	xx <sub>H</sub>	xx <sub>H</sub>	xx <sub>H</sub>	xx <sub>H</sub>	xx <sub>H</sub>																																																										
	Auft. quitt.	StatusUS	Irrelevant																																																														
—	CPU beendet den Auftrag.																																																																

### 2.12.4 Beispielablauf beim Lesen des V.24-Signalstatus

#### Beispielablauf

Die folgende Tabelle zeigt ein Beispiel dafür, wie die CPU den Status der V.24-Signale aus der seriellen Schnittstellenbaugruppe liest. Der E/A-Speicher beträgt 8 Byte.

Tabelle 2- 25 Beispielablauf beim Lesen des V.24-Signalstatus

CPU-Zyklus	CPU schreibt in ET 200S 1SI	CPU liest ET 200S 1SI																																									
1.	Anwenderprogramm liest den Ruhecode der Baugruppe: Byte 0 1 2 3 4 5 6 7 ← <table border="1" style="display: inline-table; vertical-align: middle;"> <tr> <td>00<sub>H</sub></td> <td>nnnn<sub>H</sub></td> <td>xx<sub>H</sub></td> <td>xx<sub>H</sub></td> <td>xx<sub>H</sub></td> <td>xx<sub>H</sub></td> <td>xx<sub>H</sub></td> <td>xx<sub>H</sub></td> </tr> <tr> <td>Auft.quitt.</td> <td>Status</td> <td colspan="5">Irrelevant</td> </tr> </table>	00 <sub>H</sub>	nnnn <sub>H</sub>	xx <sub>H</sub>	xx <sub>H</sub>	xx <sub>H</sub>	xx <sub>H</sub>	xx <sub>H</sub>	xx <sub>H</sub>	Auft.quitt.	Status	Irrelevant																															
	00 <sub>H</sub>	nnnn <sub>H</sub>	xx <sub>H</sub>	xx <sub>H</sub>	xx <sub>H</sub>	xx <sub>H</sub>	xx <sub>H</sub>	xx <sub>H</sub>																																			
Auft.quitt.	Status	Irrelevant																																									
	CPU schreibt den Auftrag zum Lesen des V.24-Signalstatus: Byte 0 1 2 3 4 5 6 7 <table border="1" style="display: inline-table; vertical-align: middle;"> <tr> <td>30<sub>H</sub></td> <td>xx<sub>H</sub></td> <td>xx<sub>H</sub></td> <td>xx<sub>H</sub></td> <td>xx<sub>H</sub></td> <td>xx<sub>H</sub></td> <td>xx<sub>H</sub></td> <td>xx<sub>H</sub></td> </tr> <tr> <td>Auftrag</td> <td colspan="6">Irrelevant</td> </tr> </table> →	30 <sub>H</sub>	xx <sub>H</sub>	xx <sub>H</sub>	xx <sub>H</sub>	xx <sub>H</sub>	xx <sub>H</sub>	xx <sub>H</sub>	xx <sub>H</sub>	Auftrag	Irrelevant																																
30 <sub>H</sub>	xx <sub>H</sub>	xx <sub>H</sub>	xx <sub>H</sub>	xx <sub>H</sub>	xx <sub>H</sub>	xx <sub>H</sub>	xx <sub>H</sub>																																				
Auftrag	Irrelevant																																										
2.	Anwenderprogramm liest folgende Antwort der Baugruppe: ← <table border="1" style="display: inline-table; vertical-align: middle;"> <tr> <td>31<sub>H</sub></td> <td>nnnn<sub>H</sub></td> <td>xx<sub>H</sub></td> <td>xx<sub>H</sub></td> <td>xx<sub>H</sub></td> <td>xx<sub>H</sub></td> <td>xx<sub>H</sub></td> <td>xx<sub>H</sub></td> </tr> <tr> <td>Auft.quitt.</td> <td>Signale</td> <td colspan="5">Irrelevant</td> </tr> </table> <table border="1" style="display: inline-table; vertical-align: middle; margin-left: 100px;"> <tr> <td colspan="3" style="text-align: center;">MSB</td> <td colspan="5" style="text-align: center;">LSB</td> </tr> <tr> <td>00</td> <td>0</td> <td>0</td> <td>0</td> <td>DCD</td> <td>CTS</td> <td>RTS</td> <td>DSR</td> <td>DTR</td> </tr> <tr> <td></td> <td>7</td> <td>6</td> <td>5</td> <td>4</td> <td>3</td> <td>2</td> <td>1</td> <td>0</td> </tr> </table>	31 <sub>H</sub>	nnnn <sub>H</sub>	xx <sub>H</sub>	xx <sub>H</sub>	xx <sub>H</sub>	xx <sub>H</sub>	xx <sub>H</sub>	xx <sub>H</sub>	Auft.quitt.	Signale	Irrelevant					MSB			LSB					00	0	0	0	DCD	CTS	RTS	DSR	DTR		7	6	5	4	3	2	1	0	
	31 <sub>H</sub>	nnnn <sub>H</sub>	xx <sub>H</sub>	xx <sub>H</sub>	xx <sub>H</sub>	xx <sub>H</sub>	xx <sub>H</sub>	xx <sub>H</sub>																																			
Auft.quitt.	Signale	Irrelevant																																									
MSB			LSB																																								
00	0	0	0	DCD	CTS	RTS	DSR	DTR																																			
	7	6	5	4	3	2	1	0																																			
	CPU schreibt die Quittung und nimmt die Ablaufnummer an. <table border="1" style="display: inline-table; vertical-align: middle;"> <tr> <td>31<sub>H</sub></td> <td>xx<sub>H</sub></td> <td>xx<sub>H</sub></td> <td>xx<sub>H</sub></td> <td>xx<sub>H</sub></td> <td>xx<sub>H</sub></td> <td>xx<sub>H</sub></td> <td>xx<sub>H</sub></td> </tr> <tr> <td>Auftrag</td> <td colspan="6">Irrelevant</td> </tr> </table> →	31 <sub>H</sub>	xx <sub>H</sub>	xx <sub>H</sub>	xx <sub>H</sub>	xx <sub>H</sub>	xx <sub>H</sub>	xx <sub>H</sub>	xx <sub>H</sub>	Auftrag	Irrelevant																																
31 <sub>H</sub>	xx <sub>H</sub>	xx <sub>H</sub>	xx <sub>H</sub>	xx <sub>H</sub>	xx <sub>H</sub>	xx <sub>H</sub>	xx <sub>H</sub>																																				
Auftrag	Irrelevant																																										
3.	Baugruppe kehrt in Ruhezustand zurück, nachdem die erste Transaktion beendet ist. ← <table border="1" style="display: inline-table; vertical-align: middle;"> <tr> <td>00<sub>H</sub></td> <td>nnnn<sub>H</sub></td> <td>xx<sub>H</sub></td> <td>xx<sub>H</sub></td> <td>xx<sub>H</sub></td> <td>xx<sub>H</sub></td> <td>xx<sub>H</sub></td> <td>xx<sub>H</sub></td> </tr> <tr> <td>Auft.quitt.</td> <td>Status</td> <td colspan="5">Irrelevant</td> </tr> </table>	00 <sub>H</sub>	nnnn <sub>H</sub>	xx <sub>H</sub>	xx <sub>H</sub>	xx <sub>H</sub>	xx <sub>H</sub>	xx <sub>H</sub>	xx <sub>H</sub>	Auft.quitt.	Status	Irrelevant																															
00 <sub>H</sub>	nnnn <sub>H</sub>	xx <sub>H</sub>	xx <sub>H</sub>	xx <sub>H</sub>	xx <sub>H</sub>	xx <sub>H</sub>	xx <sub>H</sub>																																				
Auft.quitt.	Status	Irrelevant																																									
—	CPU beendet den Auftrag.																																										



## 2.12.6 Parameter für die Datenflusskontrolle

### Parameter für die Datenflusskontrolle

Der Auftragscode für die Parameterübertragung mit dem ASCII-Treiber ermöglicht es Ihnen, zusätzliche Parameter einzustellen.

Dies richtet sich danach, welche Art der Datenflusskontrolle in der GSD-Datei gewählt ist. Die drei Arten der Datenflusskontrolle werden in der folgenden Tabelle beschrieben.

Tabelle 2- 27 Parameter für die Datenflusskontrolle

<b>Parameterrahmen für die Datenflusskontrolle mit XON/XOFF</b>			
<b>Byte</b>	<b>Beschreibung</b>	<b>Wertebereich</b>	<b>Defaultwert</b>
1	Parameterblocknummer	20 <sub>H</sub>	
2 und 3	Länge	0004 <sub>H</sub>	0004 <sub>H</sub>
4	XON-Zeichen	0 bis 127 (7 Datenbits) 0 bis 255 (8 Datenbits)	11 (DC1)
5	XOFF-Zeichen	0 bis 127 (7 Datenbits) 0 bis 255 (8 Datenbits)	13 (DC3)
6 und 7	Wartezeit für XON nach XOFF	20 bis 655350 in Schritten von 10 ms	200 (2000 ms)
<b>Parameterrahmen für die Datenflusskontrolle mit RTS/CTS</b>			
<b>Byte</b>	<b>Beschreibung</b>	<b>Wertebereich</b>	<b>Defaultwert</b>
1	Parameterblocknummer	21 <sub>H</sub>	
2 und 3	Länge	0002 <sub>H</sub>	0002 <sub>H</sub>
4 und 5	Wartezeit für CTS = ON	20 bis 655350 in Schritten von 10 ms	200 (2000 ms)
<b>Parameterrahmen für automatische Bedienung der RS-232C-Begleitsignale</b>			
<b>Byte</b>	<b>Beschreibung</b>	<b>Wertebereich</b>	<b>Defaultwert</b>
1	Parameterblocknummer	22 <sub>H</sub>	
2 und 3	Länge	0004 <sub>H</sub>	0004 <sub>H</sub>
4 und 5	Zeit für RTS = OFF nach der Übertragung	0 bis 655350 in Schritten von 10 ms	1 (10 ms)
6 und 7	Wartezeit für CTS = ON nach RTS = ON	0 bis 655350 in Schritten von 10 ms	1 (10 ms)



**Beispielablauf für XON/XOFF**

Die folgende Tabelle zeigt ein Beispiel dafür, wie die CPU die Parameter XON/XOFF einstellt. Der E/A-Speicher beträgt 4 Byte.

Tabelle 2- 28 Beispielablauf für XON/XOFF

CPU-Zyklus	CPU schreibt in ET 200S 1SI	CPU liest ET 200S 1SI												
1.	Anwenderprogramm sieht den folgenden Ruhecode der Baugruppe: →	<table border="1"> <tr> <td>Byte 0</td> <td>1</td> <td>2</td> <td>3</td> </tr> <tr> <td>00<sub>H</sub></td> <td>nnnn<sub>H</sub></td> <td colspan="2">xx<sub>H</sub></td> </tr> <tr> <td>Auft.quitt.</td> <td>Status</td> <td colspan="2">Irrelev.</td> </tr> </table>	Byte 0	1	2	3	00 <sub>H</sub>	nnnn <sub>H</sub>	xx <sub>H</sub>		Auft.quitt.	Status	Irrelev.	
	Byte 0	1	2	3										
00 <sub>H</sub>	nnnn <sub>H</sub>	xx <sub>H</sub>												
Auft.quitt.	Status	Irrelev.												
	<table border="1"> <tr> <td>Byte 0</td> <td>1</td> <td>2</td> <td>3</td> </tr> <tr> <td>50<sub>H</sub></td> <td>xx<sub>H</sub></td> <td>xx<sub>H</sub></td> <td>xx<sub>H</sub></td> </tr> <tr> <td>Auftrag</td> <td colspan="3">Irrelevant</td> </tr> </table>	Byte 0	1	2	3	50 <sub>H</sub>	xx <sub>H</sub>	xx <sub>H</sub>	xx <sub>H</sub>	Auftrag	Irrelevant			← Auftrag: Parametercode senden (1 0 1 oder 5 <sub>H</sub> ) plus Ablaufnummer 0
Byte 0	1	2	3											
50 <sub>H</sub>	xx <sub>H</sub>	xx <sub>H</sub>	xx <sub>H</sub>											
Auftrag	Irrelevant													
2.	Anwenderprogramm sieht folgende Antwort der Baugruppe: →	<table border="1"> <tr> <td>50<sub>H</sub></td> <td>xx<sub>H</sub></td> <td>xx<sub>H</sub></td> <td>xx<sub>H</sub></td> </tr> <tr> <td>Auft.quitt.</td> <td colspan="3">Irrelevant</td> </tr> </table>	50 <sub>H</sub>	xx <sub>H</sub>	xx <sub>H</sub>	xx <sub>H</sub>	Auft.quitt.	Irrelevant						
	50 <sub>H</sub>	xx <sub>H</sub>	xx <sub>H</sub>	xx <sub>H</sub>										
Auft.quitt.	Irrelevant													
	CPU sendet 1. Segment, da der Auftrag angenommen wurde.													
	<table border="1"> <tr> <td>51<sub>H</sub></td> <td>20<sub>H</sub></td> <td colspan="2">0004<sub>H</sub></td> </tr> <tr> <td>Auftrag</td> <td>Datenfluss</td> <td colspan="2">Sendelänge</td> </tr> </table>	51 <sub>H</sub>	20 <sub>H</sub>	0004 <sub>H</sub>		Auftrag	Datenfluss	Sendelänge		← Auftrag: Parameter fortsetzen und Ablaufnummer erhöhen ← Datenfluss: Code für Datenflussparameter				
51 <sub>H</sub>	20 <sub>H</sub>	0004 <sub>H</sub>												
Auftrag	Datenfluss	Sendelänge												
3.	Anwenderprogramm sieht folgende Antwort der Baugruppe: →	<table border="1"> <tr> <td>51<sub>H</sub></td> <td>xx<sub>H</sub></td> <td>xx<sub>H</sub></td> <td>xx<sub>H</sub></td> </tr> <tr> <td>Auft.quitt.</td> <td colspan="3">Irrelevant</td> </tr> </table>	51 <sub>H</sub>	xx <sub>H</sub>	xx <sub>H</sub>	xx <sub>H</sub>	Auft.quitt.	Irrelevant						
	51 <sub>H</sub>	xx <sub>H</sub>	xx <sub>H</sub>	xx <sub>H</sub>										
Auft.quitt.	Irrelevant													
	CPU sendet 2. Segment, da kein Fehler angezeigt wurde:													
	<table border="1"> <tr> <td>52<sub>H</sub></td> <td>0B<sub>H</sub></td> <td>0D<sub>H</sub></td> <td>00<sub>H</sub></td> </tr> <tr> <td>Auftrag</td> <td>DC1</td> <td>DC3</td> <td>Wartezeit für XON nach XOFF - msB</td> </tr> </table>	52 <sub>H</sub>	0B <sub>H</sub>	0D <sub>H</sub>	00 <sub>H</sub>	Auftrag	DC1	DC3	Wartezeit für XON nach XOFF - msB					
52 <sub>H</sub>	0B <sub>H</sub>	0D <sub>H</sub>	00 <sub>H</sub>											
Auftrag	DC1	DC3	Wartezeit für XON nach XOFF - msB											
4.	Anwenderprogramm sieht folgende Antwort der Baugruppe: →	<table border="1"> <tr> <td>52<sub>H</sub></td> <td>xx<sub>H</sub></td> <td>xx<sub>H</sub></td> <td>xx<sub>H</sub></td> </tr> <tr> <td>Auft.quitt.</td> <td colspan="3">Irrelevant</td> </tr> </table>	52 <sub>H</sub>	xx <sub>H</sub>	xx <sub>H</sub>	xx <sub>H</sub>	Auft.quitt.	Irrelevant						
	52 <sub>H</sub>	xx <sub>H</sub>	xx <sub>H</sub>	xx <sub>H</sub>										
Auft.quitt.	Irrelevant													
	CPU sendet 3. Segment, da kein Fehler angezeigt wurde:													
	<table border="1"> <tr> <td>53<sub>H</sub></td> <td>C8<sub>H</sub></td> <td>xx<sub>H</sub></td> <td>xx<sub>H</sub></td> </tr> <tr> <td>Auftrag</td> <td>Wartezeit für XON nach XOFF - LSB</td> <td colspan="2">Irrelevant</td> </tr> </table>	53 <sub>H</sub>	C8 <sub>H</sub>	xx <sub>H</sub>	xx <sub>H</sub>	Auftrag	Wartezeit für XON nach XOFF - LSB	Irrelevant						
53 <sub>H</sub>	C8 <sub>H</sub>	xx <sub>H</sub>	xx <sub>H</sub>											
Auftrag	Wartezeit für XON nach XOFF - LSB	Irrelevant												

CPU-Zyklus	CPU schreibt in ET 200S 1SI	CPU liest ET 200S 1SI								
5.	Anwenderprogramm sieht folgende Antwort der Baugruppe: →	<table border="1"> <tr> <td data-bbox="863 376 986 421">53<sub>H</sub></td> <td data-bbox="991 376 1098 421">xx<sub>H</sub></td> <td data-bbox="1102 376 1209 421">xx<sub>H</sub></td> <td data-bbox="1214 376 1305 421">xx<sub>H</sub></td> </tr> <tr> <td data-bbox="863 427 986 465">Auftr.quitt.</td> <td colspan="3" data-bbox="991 427 1305 465">Irrelevant</td> </tr> </table>	53 <sub>H</sub>	xx <sub>H</sub>	xx <sub>H</sub>	xx <sub>H</sub>	Auftr.quitt.	Irrelevant		
	53 <sub>H</sub>	xx <sub>H</sub>	xx <sub>H</sub>	xx <sub>H</sub>						
Auftr.quitt.	Irrelevant									
CPU wiederholt 3. Segment und wartet auf Auftragsendequittung.										
<table border="1"> <tr> <td data-bbox="261 533 363 577">53<sub>H</sub></td> <td data-bbox="368 533 539 577">C8<sub>H</sub></td> <td data-bbox="544 533 651 577">xx<sub>H</sub></td> <td data-bbox="655 533 715 577">xx<sub>H</sub></td> </tr> <tr> <td data-bbox="261 584 363 674">Auftrag</td> <td data-bbox="368 584 539 674">Wartezeit für XON nach XOFF - LSB</td> <td colspan="2" data-bbox="544 584 715 674">Irrelevant</td> </tr> </table>	53 <sub>H</sub>	C8 <sub>H</sub>	xx <sub>H</sub>	xx <sub>H</sub>	Auftrag	Wartezeit für XON nach XOFF - LSB	Irrelevant			
53 <sub>H</sub>	C8 <sub>H</sub>	xx <sub>H</sub>	xx <sub>H</sub>							
Auftrag	Wartezeit für XON nach XOFF - LSB	Irrelevant								
6.	Anwenderprogramm sieht folgende Antwort der Baugruppe: →	<table border="1"> <tr> <td data-bbox="863 696 986 741">73<sub>H</sub></td> <td data-bbox="991 696 1177 741">nnnn<sub>H</sub></td> <td data-bbox="1182 696 1289 741">xx<sub>H</sub></td> </tr> <tr> <td data-bbox="863 748 986 786">Auftr.quitt.</td> <td data-bbox="991 748 1177 786">Status</td> <td data-bbox="1182 748 1289 786">Irrelev.</td> </tr> </table>	73 <sub>H</sub>	nnnn <sub>H</sub>	xx <sub>H</sub>	Auftr.quitt.	Status	Irrelev.		
	73 <sub>H</sub>	nnnn <sub>H</sub>	xx <sub>H</sub>							
Auftr.quitt.	Status	Irrelev.								
CPU schreibt Ruhecode in den Auftrag und beendet den Auftrag.										
<table border="1"> <tr> <td data-bbox="261 853 363 898">00<sub>H</sub></td> <td data-bbox="368 853 475 898">xx<sub>H</sub></td> <td data-bbox="480 853 587 898">xx<sub>H</sub></td> <td data-bbox="592 853 683 898">xx<sub>H</sub></td> </tr> <tr> <td data-bbox="261 904 363 969">Auftrag</td> <td colspan="3" data-bbox="368 904 683 969">Irrelevant</td> </tr> </table>	00 <sub>H</sub>	xx <sub>H</sub>	xx <sub>H</sub>	xx <sub>H</sub>	Auftrag	Irrelevant				
00 <sub>H</sub>	xx <sub>H</sub>	xx <sub>H</sub>	xx <sub>H</sub>							
Auftrag	Irrelevant									

## 2.12.7 Fehlerbehandlung

### Fehlerbedingungen

Die serielle Schnittstellenbaugruppe gibt einen Fehler als Reaktion auf folgende Bedingungen aus:

- Ist der Sendeauftrag länger als 224 Byte, antwortet die Baugruppe mit einer Auftragsendequittung und das Statuswort enthält den Fehlercode. Die CPU schreibt dann einen Ruhecode in den Auftrag und beendet den Auftrag.
- Wurde ein Empfangsauftrag an die Baugruppe gesendet und die empfangene Meldung enthält einen Fehler, übernimmt die Baugruppe den Empfangsauftragscode mit der Ablaufnummer Null und das Statuswort enthält den Fehlercode. Die CPU schreibt dann einen Ruhecode in den Auftrag und beendet den Auftrag.
- Wurde ein Empfangsauftrag an die Baugruppe gesendet und es ist keine empfangene Meldung verfügbar, übernimmt die Baugruppe den Empfangsauftragscode mit der Ablaufnummer Null und das Statuswort enthält den Wert 0101<sub>H</sub>. Dies ist keine Fehlerbedingung, doch es verhindert, dass die Baugruppe im Empfangsauftragsmodus gesperrt wird und auf eine empfangene Meldung wartet, so dass Sendeaufträge ausgeführt werden können. Die CPU schreibt einen Ruhecode in den Auftrag und beendet den Auftrag.

### Ausnahmen

Wie bereits gesagt, darf eine bestimmte Operation (z.B. ein Sendeauftrag) im Anwenderprogramm nicht initiiert werden, bevor die Baugruppe im Ruhezustand ist. Nach dem Senden eines Auftrags muss die Operation darauf warten, dass die Baugruppe den Auftragscode annimmt, bevor die bestimmte Operation ausgeführt wird. Bei Operationen mit Segmentierung im Ablauf können folgende Ausnahmen auftreten:

---

#### Hinweis

In den folgenden Beschreibungen einer Operation zum Senden oder Parametrieren bezieht sich der Sender auf die CPU und der Empfänger auf die serielle Schnittstellenbaugruppe. Bei einer Operation zum Empfangen bezieht sich der Sender auf die serielle Schnittstellenbaugruppe und der Empfänger auf die CPU.

---

- **Fehler:** Der Sender überwacht das Fehlerbit des Empfängers auf eine segmentierte Transaktion. Bei gesetztem Fehlerbit geschieht folgendes:
  - **Die CPU ist Sender** (Sendeauftrag): Die CPU sendet die Segmente erneut und beginnt dabei mit dem nächsten Segment nach der vom Empfänger gemeldeten Nummer.
  - **Das Modul ist Sender** (Empfangsauftrag): Das 1SI Modul bricht die weitere Übertragung des Rx-Telegramms zum Anwender mit der Fehlermeldung 0x0551 im Statuswort ab. Das Modul wartet auf die Quittierung dieser Fehlermeldung. Nach Beendigung dieser laufenden Empfangssequenz wird das abgebrochene Rx-Telegramm dem Anwender wieder gemeldet bzw. zur Abholung zur Verfügung gestellt.

- **Ablaufnummer nicht in der richtigen Reihenfolge:** Wenn der Empfänger während einer segmentierten Operation ein Segment mit einer Ablaufnummer empfängt, die nicht wie die vorherige Ablaufnummer + 1 ist, muss er einen Fehler und die letzte empfangene Ablaufnummer in der Antwort melden.
  - **Die CPU ist Empfänger** (Empfangsauftrag): Wenn die CPU ein Segment mit gesetztem Fehlerbit und Fehlermeldung 0x0551 im Statusword empfängt, dann muss sie den Empfangsauftrag abbrechen und die zuvor übernommenen Daten verwerfen.
- **Geänderter Auftragscode:**
  - Empfängt der Empfänger ein Segment mit einem Auftragscode, der sich von dem Code unterscheidet, mit der die segmentierte Operation begonnen wurde und wobei es sich nicht um 000 oder 111 handelt, ignoriert der Empfänger den anderen Code und verwirft die zugehörigen Daten.
  - Empfängt der Empfänger ein Segment mit dem Auftragscode des Ruhezustands während einer segmentierten Operation, wird die Operation abgebrochen und der Ruhezustand eingenommen, ohne dass ein Fehlerbit gesetzt wird.
  - Empfängt der Empfänger ein Segment mit dem Auftragscode der Auftragsendequittung während einer segmentierten Operation, wird die Operation abgebrochen und der Ruhezustand eingenommen, ohne dass ein Fehlerbit gesetzt wird.
  - Empfängt der Sender während einer segmentierten Operation eine Antwort, die einen anderen Auftragscode hat, muss die Meldung abgebrochen werden. Daraufhin wird der Ruhecode erneut gesendet, die Baugruppe muss in den Ruhezustand gehen und die Operation erneut ausführen.

## 2.13 Diagnose

### Übersicht

Die Diagnosefunktionen der Baugruppe ET 200S 1SI erlauben Ihnen eine schnelle Lokalisierung evtl. während des Betriebs aufgetretener Fehler. Folgende Diagnosemöglichkeiten stehen Ihnen zur Verfügung:

- Diagnose über die Status-LEDs auf der Frontplatte der Baugruppe ET 200S 1SI
- Diagnose über den Ausgang STATUS der Funktionsbausteine
- Diagnose über PROFIBUS-Slave-Diagnose

### Diagnoseinformationen über Status-LEDs

Die folgenden Status-LEDs befinden sich auf der Frontplatte der Schnittstellenbaugruppe ET 200S 1SI:

- **TX** (grün): Leuchtet auf, wenn die Baugruppe Daten über die Schnittstelle sendet.
- **RX** (grün): Leuchtet auf, wenn die Baugruppe Daten über die Schnittstelle empfängt.
- **SF** (rot): Zeigt einen der möglichen Fehler an:
  - Hardwarefehler
  - Parametrierungsfehler
  - Drahtbruch oder lockere Kabel zwischen der Baugruppe und dem Kommunikationspartner:  
werden nur bei RS-422-Schnittstellenverbindungen mit dem Parameter Vorbelegung der Empfangsleitung = R(A) 5V / R(B) 0V erkannt.
  - Kommunikationsfehler (Parität, Rahmenfehler, Pufferüberlauf)

### Aufbau der Diagnosemeldungen der Funktionsbausteine

Für eine Fehlerdiagnose besitzt jeder Funktionsbaustein einen Parameter STATUS. Jede STATUS-Meldungsnummer hat unabhängig vom verwendeten Funktionsbaustein die gleiche Bedeutung. Das folgende Bild zeigt den Aufbau des Parameters STATUS.

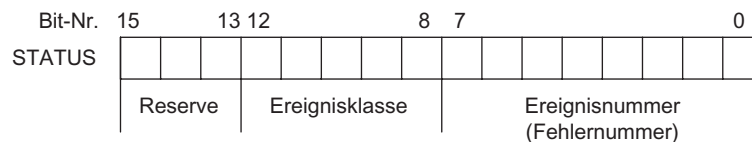


Bild 2-23 Aufbau des Parameters STATUS

Beispiel: Das folgende Bild zeigt den Inhalt des Parameters STATUS für das Ereignis "Auftragsabbruch wegen Neustart, Wiederanlauf oder Reset" (Ereignisklasse 1E<sub>H</sub>, Ereignisnummer 0D<sub>H</sub>).

Ereignis: "Auftragsabbruch wegen Neustart, Wiederanlauf oder Reset"

STATUS	x	x	x	2 <sup>4</sup>	1	1	1	1	2 <sup>0</sup>	0	0	0	0	2 <sup>7</sup>	0	0	0	0	2 <sup>0</sup>	1	1	0	1
	Reserve			Ereignisklasse: 1E <sub>H</sub>					Ereignisnummer: 0D <sub>H</sub>														

Bild 2-24 Beispiel: Parameter STATUS für Ereignisklasse 1EH, Ereignis 0DH

### Diagnosemeldungen der Funktionsbausteine

Die folgende Tabelle beschreibt die Ereignisklassen, die Definitionen der Ereignisnummern und die empfohlene Abhilfe zu jeder Fehlerbedingung.

Tabelle 2- 29 Diagnosemeldungen im Parameter STATUS

Ereignisnummer	Ereignis	Abhilfe
<b>Ereignisklasse 2 (0x02<sub>H</sub>): "Fehler bei der Initialisierung"</b>		
(02) 01 <sub>H</sub>	Keine (gültige) Parametrierung vorhanden.	Versorgen Sie die Baugruppe mit korrekten Parametern. Überprüfen Sie ggf. die ordnungsgemäße Installation der Anlage.
<b>Ereignisklasse 5 (05<sub>H</sub>): "Fehler bei Bearbeitung eines CPU-Auftrags"</b>		
(05) 02 <sub>H</sub>	Auftrag ist in diesem Betriebszustand der Baugruppe ET 200S 1SI nicht erlaubt (z. B. Geräteschnittstelle nicht parametrierbar).	Das Sendetelegramm ist länger als 224 Byte. Der Sendeauftrag wurde von der Baugruppe ET 200S 1SI abgebrochen. Wählen Sie eine kleinere Telegrammlänge.
(05) 0E <sub>H</sub>	Ungültige Telegrammlänge	Das Sendetelegramm ist länger als 224 Byte. Der Sendeauftrag wurde von der Baugruppe ET 200S 1SI abgebrochen. Wählen Sie eine kleinere Telegrammlänge.
(05) 50 <sub>H</sub>	Parameteraktualisierungsauftrag ungültig für aktuelle Datenflusskontrolle der Baugruppe ET 200S 1SI.	Entweder ändern Sie die Parameter des Funktionsbausteins (FB6 S_XON, FB7 S_RTS, FB8 S_V24) im AS-Programm oder Sie ändern die Datenflusskontrolle der Baugruppe ET 200S 1SI in der Hardware-Konfiguration, so dass sich beide entsprechen.
(05) 51 <sub>H</sub>	Rahmenablauffehler bei der Kommunikation zwischen der Baugruppe ET 200S 1SI und dem Automatisierungssystem. Der Fehler ist beim Übertragen eines empfangenen Telegramms der Baugruppe ET 200S 1SI im Automatisierungssystem aufgetreten.	Die Baugruppe und das Automatisierungssystem haben die Übertragung abgebrochen. Wiederholen Sie den Empfangsauftrag; die Baugruppe ET 200S 1SI sendet die empfangene Meldung erneut.
<b>Ereignisklasse 7 (07<sub>H</sub>): "Sendefehler"</b>		
(07) 02 <sub>H</sub>	Nur bei 3964(R): Fehler beim Verbindungsaufbau: Nachdem STX gesendet wurde, wurde NAK oder ein beliebiges Zeichen (außer DLE oder STX) empfangen.	Fehlverhalten des Partnergeräts ggf. mit Schnittstellentestgerät (FOXPG) untersuchen, das in die Übertragungsleitung eingeschaltet wird.

Ereignisnummer	Ereignis	Abhilfe
(07) 03 <sub>H</sub>	Nur bei 3964(R): Quittungsverzugszeit (QVZ) überschritten: Nach Senden von STX kam keine Antwort vom Partner innerhalb der Quittungsverzugszeit.	Partnergerät ist zu langsam oder nicht empfangsbereit, oder es liegt z. B. ein Bruch der Sendeleitung vor. Fehlverhalten des Partnergeräts ggf. mit Schnittstellentestgerät (FOXPG) untersuchen, das in die Übertragungsleitung eingeschaltet wird.
(07) 04 <sub>H</sub>	Nur bei 3964(R): Abbruch durch Partner: Während des laufenden Sendebetriebs wurden vom Partner ein oder mehrere Zeichen empfangen.	Prüfen Sie, ob der Partner ebenfalls Fehler anzeigt, da evtl. nicht alle Sendedaten angekommen sind (z. B. Bruch in der Sendeleitung) oder schwere Störungen vorliegen, oder es liegt ein Fehlverhalten des Partnergeräts vor. Fehlverhalten des Partnergeräts ggf. mit Schnittstellentestgerät (FOXPG) untersuchen, das in die Übertragungsleitung eingeschaltet wird.
(07) 05 <sub>H</sub>	Nur bei 3964(R): Negative Quittung beim Senden	Prüfen Sie, ob der Partner ebenfalls Fehler anzeigt, da evtl. nicht alle Sendedaten angekommen sind (z.B. Bruch in der Sendeleitung) oder schwere Störungen vorliegen, oder es liegt ein Fehlverhalten des Partnergeräts vor. Fehlverhalten des Partnergeräts ggf. mit Schnittstellentestgerät (FOXPG) untersuchen, das in die Übertragungsleitung eingeschaltet wird.
(07) 06 <sub>H</sub>	Nur bei 3964(R): Fehler bei Verbindungsende: <ul style="list-style-type: none"> <li>• Das Telegramm wurde vom Partner am Ende mit NAK oder einem beliebigen Zeichen (außer DLE) abgelehnt oder</li> <li>• das Quittungszeichen (DLE) wurde zu früh empfangen.</li> </ul>	Prüfen Sie, ob der Partner ebenfalls Fehler anzeigt, da evtl. nicht alle Sendedaten angekommen sind (z.B. Bruch in der Sendeleitung) oder schwere Störungen vorliegen, oder es liegt ein Fehlverhalten des Partnergeräts vor. Fehlverhalten des Partnergeräts ggf. mit Schnittstellentestgerät (FOXPG) untersuchen, das in die Übertragungsleitung eingeschaltet wird.
(07) 07 <sub>H</sub>	Nur bei 3964(R): Quittungsverzugszeit am Verbindungsende/Antwortüberwachungszeit nach Sendetelegramm überschritten: Nach Verbindungsabbau mit DLE ETX kam innerhalb der QVZ keine Antwort vom Partner.	Partnergerät ist zu langsam oder gestört. Dies ggf. mit Schnittstellentestgerät nachweisen, das in die Übertragungsleitung eingeschaltet wird.
(07) 08 <sub>H</sub>	Nur bei ASCII-Treiber: Die Wartezeit auf XON bzw. CTS = ON ist abgelaufen.	Der Kommunikationspartner ist gestört, zu langsam oder offline geschaltet. Überprüfen Sie den Kommunikationspartner oder ändern Sie ggf. die Parametrierung.
(07) 0B <sub>H</sub>	Nur bei 3964(R): Initialisierungskonflikt ist nicht lösbar, weil beide Partner hochprior eingestellt sind.	Ändern Sie die Parametrierung.
(07) 0C <sub>H</sub>	Nur bei 3964(R): Initialisierungskonflikt ist nicht lösbar, weil beide Partner niederprior eingestellt sind.	Ändern Sie die Parametrierung.

Ereignisnummer	Ereignis	Abhilfe
<b>Ereignisklasse 8 (08<sub>H</sub>): "Empfangsfehler"</b>		
(08) 02 <sub>H</sub>	<p>Nur bei 3964(R):</p> <p>Fehler beim Verbindungsaufbau:</p> <ul style="list-style-type: none"> <li>In Ruhestellung wurden ein oder mehrere beliebige Zeichen (außer NAK oder STX) empfangen oder</li> <li>Nach einem empfangenen STX wurden vom Partner weitere Zeichen gesendet, ohne die Antwort DLE abzuwarten.</li> </ul> <p>Nach Netz-EIN des Partners:</p> <ul style="list-style-type: none"> <li>Während der Partner eingeschaltet wird, empfängt die Baugruppe ein undefiniertes Zeichen.</li> </ul>	<p>Fehlverhalten des Partnergeräts ggf. mit Schnittstellentestgerät (FOXPG) untersuchen, das in die Übertragungsleitung eingeschaltet wird.</p>
(08) 05 <sub>H</sub>	<p>Nur bei 3964(R):</p> <p>Logischer Fehler während des Empfangs:</p> <p>Nach Empfang von DLE wurde ein weiteres beliebiges Zeichen empfangen (außer DLE, ETX).</p>	<p>Prüfen Sie, ob der Partner DLE im Telegrammkopf und im Datenstring immer verdoppelt bzw. der Verbindungsabbau mit DLE ETX vorgenommen wird.</p> <p>Fehlverhalten des Partnergeräts ggf. mit Schnittstellentestgerät (FOXPG) untersuchen, das in die Übertragungsleitung eingeschaltet wird.</p>
(08) 06 <sub>H</sub>	<p>Zeichenverzugszeit (ZVZ) überschritten:</p> <ul style="list-style-type: none"> <li>Zwei aufeinanderfolgende Zeichen wurden nicht innerhalb der ZVZ empfangen oder</li> </ul> <p>Nur bei 3964(R):</p> <ul style="list-style-type: none"> <li>1. Zeichen nach Senden von DLE beim Verbindungsaufbau wurde nicht innerhalb der ZVZ empfangen.</li> </ul>	<p>Partnergerät ist zu langsam oder gestört.</p> <p>Fehlverhalten des Partnergeräts ggf. mit Schnittstellentestgerät (FOXPG) untersuchen, das in die Übertragungsleitung eingeschaltet wird.</p>
(08) 07 <sub>H</sub>	<p>Nur bei 3964(R):</p> <p>Telegrammlänge unzulässig:</p> <p>Es wurde ein Telegramm mit der Länge 0 empfangen.</p>	<p>Der Empfang eines Telegramms mit Länge 0 ist kein Fehler.</p> <p>Überprüfen Sie, warum der Kommunikationspartner Telegramme ohne Nutzdaten sendet.</p>
(08) 08 <sub>H</sub>	<p>Nur bei 3964(R):</p> <p>Fehler beim Blockprüfzeichen BCC:</p> <p>Der intern gebildete Wert des BCC stimmt nicht mit dem vom Partner am Verbindungs-ende empfangenen BCC überein.</p>	<p>Prüfen Sie, ob die Verbindung stark gestört ist, in diesem Fall werden auch gelegentlich Fehlercodes zu beobachten sein.</p> <p>Fehlverhalten des Partnergeräts ggf. mit Schnittstellentestgerät (FOXPG) untersuchen, das in die Übertragungsleitung eingeschaltet wird.</p>
(08) 09 <sub>H</sub>	<p>Nur bei 3964(R):</p> <p>Die Wiederholanzahl muss gleich eingestellt sein.</p>	<p>Parametrieren Sie beim Kommunikationspartner die gleiche Blockwartezeit wie an der Baugruppe.</p> <p>Fehlverhalten des Kommunikationspartners ggf. mit Schnittstellengerät nachweisen, das in die Übertragungsleitung eingeschaltet wird.</p>
(08) 0A <sub>H</sub>	<p>Ein freier Empfangspuffer ist nicht vorhanden:</p> <p>Beim Empfang stand kein leerer Empfangspuffer zur Verfügung.</p>	<p>Der FB S_RCV muss häufiger aufgerufen werden.</p>



Ereignisnummer	Ereignis	Abhilfe
<b>Ereignisklasse 8 (08<sub>H</sub>): "Empfangsfehler"</b>		
(08) 0C <sub>H</sub>	<p>Übertragungsfehler:</p> <ul style="list-style-type: none"> <li>Ein Übertragungsfehler (Paritätsfehler, Stoppbitfehler, Überlauffehler) wurde erkannt.</li> </ul> <p>Nur bei 3964(R):</p> <ul style="list-style-type: none"> <li>Falls dies während des Sende- oder Empfangsbetriebs auftritt, werden Wiederholungen gestartet.</li> <li>Wird in Ruhestellung ein gestörtes Zeichen empfangen, wird der Fehler sofort gemeldet, damit Störeinflüsse in der Übertragungsleitung frühzeitig erkannt werden können.</li> <li>Ist die SF-LED (rot) eingeschaltet, liegt eine Unterbrechung der Verbindungsleitung der beiden Kommunikationspartner vor.</li> </ul>	<p>Störungen in der Übertragungsleitung verursachen Telegrammwiederholungen und erniedrigen dadurch den Nutzdatendurchsatz. Die Gefahr eines nicht erkannten Fehlers steigt. Ändern Sie Ihren Systemaufbau bzw. die Leitungsverlegung. Überprüfen Sie die Verbindungsleitung der Kommunikationspartner bzw. überprüfen Sie, ob bei beiden Geräten Baudrate, Parität und Stoppbitanzahl gleich eingestellt sind.</p>
(08) 0D <sub>H</sub>	BREAK: Empfangsleitung zum Partner ist unterbrochen.	Stellen Sie die Verbindung wieder her oder schalten Sie den Partner ein.
(08) 10 <sub>H</sub>	<p>Nur bei ASCII-Treiber:</p> <p>Paritätsfehler:</p> <ul style="list-style-type: none"> <li>Ist die SF-LED (rot) eingeschaltet, liegt eine Unterbrechung der Verbindungsleitung der beiden Kommunikationspartner vor.</li> </ul>	<p>Überprüfen Sie die Verbindungsleitung der Kommunikationspartner bzw. überprüfen Sie, ob bei beiden Geräten Baudrate, Parität und Stoppbitanzahl gleich eingestellt sind.</p> <p>Ändern Sie Ihren Systemaufbau bzw. die Leitungsverlegung.</p>
(08) 11 <sub>H</sub>	<p>Nur bei ASCII-Treiber:</p> <p>Zeichenrahmenfehler:</p> <ul style="list-style-type: none"> <li>Ist die SF-LED (rot) eingeschaltet, liegt eine Unterbrechung der Verbindungsleitung der beiden Kommunikationspartner vor.</li> </ul>	<p>Überprüfen Sie die Verbindungsleitung der Kommunikationspartner bzw. überprüfen Sie, ob bei beiden Geräten Baudrate, Parität und Stoppbitanzahl gleich eingestellt sind.</p> <p>Ändern Sie Ihren Systemaufbau bzw. die Leitungsverlegung.</p>
(08) 12 <sub>H</sub>	<p>Nur bei ASCII-Treiber:</p> <p>Nachdem die Baugruppe XOFF gesendet hat oder CTS auf OFF gestellt hat, wurden weitere Zeichen empfangen.</p>	Parametrieren Sie den Kommunikationspartner neu oder lesen Sie die Daten der Baugruppe schneller.
(08) 18 <sub>H</sub>	<p>Nur bei ASCII-Treiber:</p> <p>DSR = OFF bzw. CTS = OFF</p>	<p>Vor oder während eines Sendevorgangs sind die Signale DSR bzw. CTS vom Partner auf "OFF" geschaltet worden.</p> <p>Überprüfen Sie die Steuerung der RS-232C-Begleitsignale beim Partner.</p>
(08) 50 <sub>H</sub>	Länge des Empfangstelegramms größer als 224 Byte bzw. größer als die parametrisierte Telegrammlänge	Telegrammlänge des Partners anpassen
<b>Ereignisklasse 11 (0B<sub>H</sub>): Warnung</b>		
(0B) 01 <sub>H</sub>	Empfangspuffer ist mehr als 2/3 voll	

Ereignisnummer	Ereignis	Abhilfe
<b>Ereignisklasse 30 (1E<sub>H</sub>): "Fehler bei Kommunikation zwischen Baugruppe und CPU"</b>		
(1E) 0D <sub>H</sub>	"Auftragsabbruch wegen Neustart, Wiederanlauf oder Reset"	
(1E) 0E <sub>H</sub>	Statischer Fehler bei Aufruf der SFC DPRD_DAT. Der Returnwert RET_VAL der SFC wird Ihnen in der Variablen SFCERR im Instanz-DB zur Auswertung zur Verfügung gestellt.	Laden Sie die Variable SFCERR aus dem Instanz-DB.
(1E) 0F <sub>H</sub>	Statischer Fehler bei Aufruf der SFC DPWR_DAT. Der Returnwert RET_VAL der SFC wird Ihnen in der Variablen SFCERR im Instanz-DB zur Auswertung zur Verfügung gestellt.	Laden Sie die Variable SFCERR aus dem Instanz-DB.
(1E) 10 <sub>H</sub>	Statischer Fehler bei Aufruf der SFC RD_LGADR. Der Returnwert RET_VAL der SFC wird Ihnen in der Variablen SFCERR im Instanz-DB zur Auswertung zur Verfügung gestellt.	Laden Sie die Variable SFCERR aus dem Instanz-DB.
(1E) 11 <sub>H</sub>	Statischer Fehler bei Aufruf der SFC RDSYSST. Der Returnwert RET_VAL der SFC wird Ihnen in der Variablen SFCERR im Instanz-DB zur Auswertung zur Verfügung gestellt.	Laden Sie die Variable SFCERR aus dem Instanz-DB.
(1E) 20 <sub>H</sub>	Parameter außerhalb des Bereichs.	Ändern Sie den Eingang des Funktionsbausteins, so dass er im gültigen Bereich liegt.
(1E) 41 <sub>H</sub>	Anzahl der am Parameter LEN der FBs angegebenen Bytes unzulässig	Halten Sie den Wertebereich von 1 bis 224 Byte ein.

**Auswerten der Variable SFCERR**

Nähere Informationen zu den aufgetretenen Fehlern (1E) 0E<sub>H</sub>, (1E) 0F<sub>H</sub>, (1E) 10<sub>H</sub> und (1E) 11<sub>H</sub>, der Ereignisklasse 30 erhalten Sie über die Variable SFCERR.

Die Variable SFCERR können Sie aus dem Instanz-DB des entsprechenden Funktionsbausteins laden.

Die Fehlermeldungen, die in der Variablen SFCERR eingetragen werden, finden Sie bei den Systemfunktionen "DPRD\_DAT" und SFC15 "DPWR\_DAT" im Referenzhandbuch *Systemsoftware für S7 300/400, System- und Standardfunktionen*.

**PROFIBUS-Slave-Diagnose**

Die Slave-Diagnose verhält sich nach der Norm EN 50170, Volume 2, PROFIBUS. Sie kann in Abhängigkeit vom DP-Master für alle DP-Slaves, die sich nach Norm verhalten, mit STEP 5 oder STEP 7 ausgelesen werden.

Die PROFIBUS-Slave-Diagnose umfasst die Baugruppendiagnose, den Baugruppenstatus und die kanalbezogene Diagnose. Ausführliche Informationen zur DP-Slave-Diagnose finden Sie im Handbuch *Dezentrales Peripheriegerät ET 200S*.

**Kanalbezogene Diagnose:** Die kanalbezogene Diagnose gibt Auskunft über Kanalfehler von Baugruppen und beginnt nach dem Baugruppenstatus. Die folgende Tabelle führt die kanalbezogenen Fehlerarten auf.

Tabelle 2- 30 Arten von Kanalfehlern bei der seriellen Schnittstellenbaugruppe ET 200S 1SI

<b>Fehlertyp</b>	<b>Bedeutung</b>	<b>Abhilfe</b>
00110: Drahtbruch	Draht gebrochen oder gelöst.	Überprüfen Sie die Verdrahtung zu den Klemmen. Überprüfen Sie das Kabel zum Partner.
00111: Überlauf	Pufferüberlauf; Meldungslängenüberlauf	Der FB S_RCV muss häufiger aufgerufen werden.
01000: Unterlauf	Nur 3964(R): Meldung mit Länge 0 gesendet	Überprüfen Sie, warum der Kommunikationspartner Telegramme ohne Nutzdaten sendet.
01001: Fehler	Interner Baugruppenfehler aufgetreten.	Tauschen Sie die Baugruppe aus.
10000: Parametrierungsfehler	Baugruppe ist nicht parametriert.	Korrigieren Sie die Parametrierung.
10110: Meldungsfehler	Rahmenfehler; Paritätsfehler	Prüfen Sie die Kommunikationseinstellungen.

## 2.14 Technische Daten

### Allgemeine technische Daten

Für die serielle Schnittstellenbaugruppe ET 200S 1SI 3964/ASCII gelten die allgemeinen technischen Daten, wie sie im Handbuch *Dezentrales Peripheriesystem ET 200S* im Kapitel "Allgemeine Technische Daten" beschrieben sind. Dieses Handbuch finden Sie hier:

<http://www.siemens.de/simatic-tech-doku-portal>

### Technische Daten der Protokolle und Schnittstelle

Tabelle 2- 31 Allgemeine technische Daten der Baugruppe ET 200S 1SI

<b>Allgemeine technische Daten</b>	
Anzeigeelemente	<ul style="list-style-type: none"> <li>• LED (grün): TX (Senden)</li> <li>• LED (grün): RX (Empfangen)</li> <li>• LED (rot): SF (Sammelfehler)</li> </ul>
Mitgelieferte Protokolltreiber	Treiber 3964(R) ASCII-Treiber
Baudraten beim Protokoll 3964(R) Baudraten bei ASCII-Treibern	110, 300, 600, 1200, 2400, 4800, 9600, 19.200, 38.400. 57.66, 76.88, 115.200
Zeichenrahmen (10 Bit oder 11 Bit)	Anzahl Bit pro Zeichen: 7 oder 8 Anzahl Start-/Stopbits: 1 oder 2 Parität: keine, gerade, ungerade, beliebig
Speicherplatzbedarf der Standardbausteine (FBs)	Senden und Empfangen: ca. 4300 Byte
<b>Technische Daten der RS-232C-Schnittstelle</b>	
Schnittstelle	RS-232C, 8 Klemmen
RS-232C-Signale	TXD, RXD, RTS, CTS, DTR, DSR, DCD, PE Alle elektrisch von der internen Spannungsversorgung der Baugruppe ET 200S 1SI getrennt.
Maximaler Übertragungsweg	15 m
<b>Technische Daten der RS-422/485-Schnittstelle</b>	
Schnittstelle	<ul style="list-style-type: none"> <li>• RS-422, 5 Klemmen</li> <li>• RS-485, 3 Klemmen</li> </ul>
RS-422-Signale RS-485-Signale	TXD (A)-, RXD (A)-, TXD (B)+, RXD (B)+, PE R/T (A)-, R/T (B)+, PE Alle elektrisch von der internen Spannungsversorgung der Baugruppe ET 200S 1SI getrennt.
Maximaler Übertragungsweg	1200 m

## Technische Daten

<b>Maße und Gewicht</b>	
Abmessungen B x H x T (in mm)	15 x 81 x 52
Gewicht	ca. 50 g
<b>Baugruppenspezifische Daten</b>	
RS-232C	
• Anzahl der Eingänge	4
• Anzahl Ausgänge	3
RS-422	
• Anzahl Eingangspaare	1
• Anzahl Ausgangspaare	1
RS-485	
Anzahl E/A-Paare	1
Kabellänge	
• Geschirmt (RS-232C)	max. 15 m
• Geschirmt (RS-422/485)	max. 1200 m
Schutzart <sup>1</sup>	IEC 801-5
<b>Spannungen, Ströme, Potenziale</b>	
Versorgungsnennspannung der Elektronik (L+)	24V DC
• Verpolschutz	ja
Potenzialtrennung	
• zwischen Kanälen und Rückwandbus	ja
• zwischen Kanälen und Spannungsversorgung der Elektronik	ja
• zwischen Kanälen	nein
• zwischen Kanälen und PROFIBUS-DP	ja
Isolation geprüft mit	
• Kanäle gegen Rückwandbus und Lastspannung L+	DC 500 V
• Lastspannung L+ gegen Rückwandbus	AC 500 V
Stromquelle	
• aus Rückwandbus	max. 10 mA
• aus Spannungsversorgung L+	max. 120 mA, typ. 50 mA
Verlustleistung der Baugruppe	typ. 1,2 W
<b>Status, Alarme, Diagnose</b>	
Statusanzeige	<ul style="list-style-type: none"> <li>• grüne LED (TX)</li> <li>• grüne LED (RX)</li> </ul>
Diagnosefunktionen	
• Sammelfehleranzeige	rote LED (SF)
• Diagnoseinformationen können angezeigt werden	möglich

Maße und Gewicht	
<b>Ausgänge</b>	
Ausgang, RS-232C-Bereich <ul style="list-style-type: none"> <li>• für kapazitive Last</li> <li>• Kurzschlussschutz</li> <li>• Kurzschlussstrom</li> <li>• Spannung an den Ausgängen oder Eingängen zu PE (Erde)</li> </ul>	± max. 10 V max. 2500 pF ja ca. 60 mA max. 25 V
Ausgang, RS-422/485 <ul style="list-style-type: none"> <li>• Bürdenwiderstand</li> <li>• Kurzschlussschutz</li> <li>• Kurzschlussstrom</li> </ul>	min. 50 kΩ ja ca. 60 mA
<sup>1</sup> Externe Schutzgeräte erforderlich in den Eingangsleitungen der Anwenderspannung: <ul style="list-style-type: none"> <li>• Blitzductor Standard-Hutschienen-Adapter</li> <li>• Blitzductor Schutzmodultyp KT AD-24V</li> </ul>	

# Modbus/USS

## 3.1 Produktübersicht

### Bestellnummer

6ES7 138-4DF11-0AB0

### Produktbeschreibung

Bei der seriellen Schnittstellenbaugruppe ET 200S Modbus/USS handelt es sich um eine steckbare Baugruppe der Produktreihe ET 200S, die mit Hilfe von drei Hardware-Schnittstellen (RS-232C, RS-422 und RS-485) und zwei Software-Protokollen Zugang zur seriellen Kommunikation bietet:

- Modbus
- USS-Master

Mit der seriellen Schnittstellenbaugruppe ET 200S Modbus/USS können Sie über eine Punkt-zu-Punkt-Kopplung Daten zwischen Automatisierungssystemen oder Computern austauschen. Sämtliche Kommunikation läuft über serielle asynchrone Übertragungen ab.

Sie wählen die Kommunikationsart, wenn Sie die Baugruppe in der Hardware-Konfiguration von STEP 7 bzw. in einer anderen Konfigurationsanwendung parametrieren. Die Baugruppe wird im Hardware-Katalog in folgenden neun Ausführungen angezeigt:

- Modbus-Master (4 Byte)
- Modbus-Master (8 Byte)
- Modbus-Master (32 Byte)
- Modbus-Slave (4 Byte)
- Modbus-Slave (8 Byte)
- Modbus-Slave (32 Byte)
- USS-Master (4 Byte)
- USS-Master (8 Byte)
- USS-Master (32 Byte)

Durch 8- bzw. 32-Byte-Datenübertragung wird die Durchsatzeffizienz erhöht, doch es wird mehr E/A-Platz im Baugruppenträger ET 200S eingenommen. Die 4-Byte-Datenübertragung nimmt weniger E/A-Platz im Baugruppenträger ET 200S ein, doch bietet sie weniger Durchsatzeffizienz. Die Baugruppenvariante richtet sich nach Ihren Anwendungsanforderungen.

**Funktionalität der seriellen Schnittstellenbaugruppe ET 200S Modbus/USS**

Die serielle Schnittstellenbaugruppe ET 200S Modbus/USS bietet folgende Funktionen:

- Integrierte Schnittstelle nach RS-232C, RS-422 oder RS-485
- Übertragungsgeschwindigkeit bis 115,2 Kbaud, halbduplex
- Integration der folgenden Übertragungsprotokolle in die Firmware der Baugruppe:
  - Modbus-Master-Treiber
  - Modbus-Slave-Treiber
  - USS-Master-Treiber

Die Parametrierung der Baugruppe bestimmt die Funktionalität der Treiber.

Die folgende Tabelle führt die Funktionen der einzelnen Treiberschnittstellen auf.

Tabelle 3- 1 Funktionen der Baugruppentreiber der seriellen Schnittstellenbaugruppe ET 200S Modbus/USS

Funktion	RS-232C	RS-422	RS-485
<b>Modbus-Treiber</b>	ja	ja	ja
Automatische Bedienung der RS-232C-Signale	ja	nein	nein
<b>USS-Master-Treiber</b>	ja	nein	ja

**Hinweis**

Hinter den externen Kommunikations-CPs CP 342-5 (Profibus DP) und CP 343-1 (Profinet IO) ist das Modul ET 200S Modbus/USS mit den normalen Standard-FBs nicht betreibbar!

Für den Betrieb des Moduls hinter den Kommunikations-CPs CP 342-5 (Profibus DP) oder CP 343-1 (Profinet IO) stehen auf den Seiten des Customer Supports entsprechende Spezial-FBs zur Verfügung:

Siehe <http://support.automation.siemens.com/WW/view/de/26263724>

**LED-Anzeigen**

Folgenden Status-LEDs befinden sich auf der Frontplatte der Schnittstellenbaugruppe:

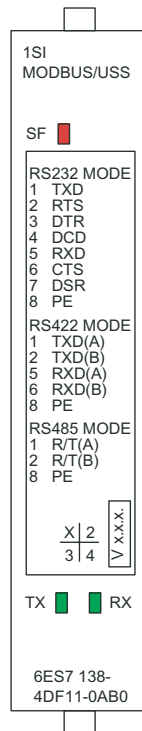
LED	Farbe	Beschreibung
SF	rot	Sammelfehleranzeige
TX	grün	Die Schnittstelle sendet.
RX	grün	Die Schnittstelle empfängt.

Die Betriebszustände und Fehler, die von diesen LEDs angezeigt werden, sind im Abschnitt Diagnoseinformationen der Status-LEDs (Seite 207) beschrieben.



## Frontplatte

Das folgende Bild zeigt die Beschriftung der Frontplatte der seriellen Schnittstellenbaugruppe ET 200S Modbus/USS.



### 3.2 Kurzanleitung zur Inbetriebnahme der seriellen Schnittstellenbaugruppe

#### Aufgabe

In dieser Kurzerklärung wird anhand eines Beispiels zum Senden und Empfangen von Daten zwischen seriellen Schnittstellenbaugruppen erläutert, wie Sie eine funktionierende Anwendung einrichten, wie die grundlegenden Operationen der seriellen Schnittstellenbaugruppe (Hardware und Software) funktionieren und wie Sie Hard- und Software prüfen.

In diesem Beispiel betreiben wir zwei serielle Schnittstellenbaugruppen ET 200S 1SI Modbus/USS als RS-232C Modbus-Master <-> Modbus-Slave Kopplung.

#### Voraussetzungen

Folgende Voraussetzungen müssen erfüllt sein

- Sie nehmen eine ET 200S-Station an einer S7-Station mit DP-Master in Betrieb.
- Sie benötigen folgende Komponenten:
  - Zwei Terminalmodule TM-E15S24-01
  - Zwei serielle Schnittstellenbaugruppen ET 200S 1SI Modbus/USS
  - Das erforderliche Verdrahtungsmaterial

#### Montieren, Verdrahten und Bestücken

Montieren und verdrahten Sie die zwei Terminalmodule TM-E15S24-01 (siehe folgendes Bild). Verbinden Sie die zwei seriellen Schnittstellenbaugruppen ET 200S 1SI Modbus/USS mit den Terminalmodulen. (Eine ausführliche Anleitung hierzu finden Sie im Handbuch *Dezentrale Peripherie*).

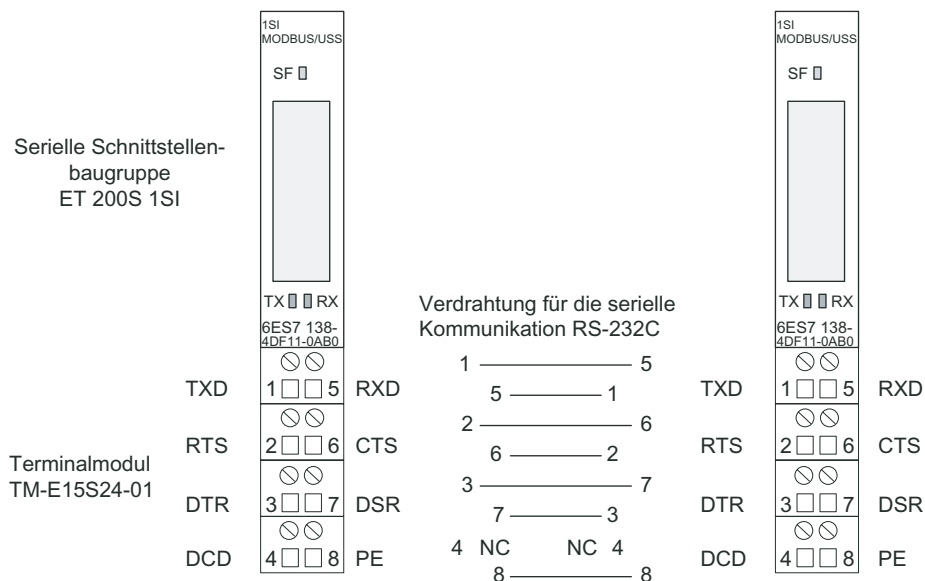


Bild 3-1 Anschlussbelegung für das Beispiel

## Verwendete Konfiguration

In der folgenden Tabelle finden Sie die für das Beispielprogramm verwendete Konfiguration.

Tabelle 3-2 Parametrierung für die Beispielanwendung

Parameter	Wert
Sammeldiagnose	sperren
Schnittstelle	RS232-C
<b>Vorbelegte Empfangsleitung</b>	
Betriebsart	Normal Operation
Slaveadresse <sup>1</sup>	1
Datenflusskontrolle (Vorbelegung)	keine
Baudrate	9600
Stopbits	1
Parität	gerade
Vielfaches der Ablaufzeit	1
Antwortzeit (ms) <sup>2</sup>	2000
<b>Zeit für RTS = aus (ms)</b>	
<b>Wartezeit für das Auswerten von Daten (ms)</b>	
Empfangspuffer bei Anlauf löschen	ja
<sup>1</sup> nur bei Modbus-Slave	
<sup>2</sup> nur bei Modbus-Master	

## Verwendete Bausteine

In der folgenden Tabelle finden Sie die für das Beispielprogramm verwendeten Bausteine.

Bausteine	Symbol	Kommentar
OB 1	CYCLE	Zyklische Programmbearbeitung
OB 100	RESTART	Anlaufbearbeitung Neustart
DB 21	SEND_IDB_SI_0	Instanz-DB für FB S_SEND_SI
DB 22	RECV_IDB_SI_1	Instanz-DB für FB S_RECV_SI
DB 40	SEND_WORK_DB_SI_0	Arbeits-DB für den Standard-FB 3
DB 41	RECV_WORK_DB_SI_1	Arbeits-DB für den Standard-FB 2
DB 42	SEND_SRC_DB_SI_0	Sende-Datenbaustein
DB 43	RECV_DST_DB_SI_0	Empfangs-Datenbaustein
DB 81	MODSL_IDB_SI_1	Instanz DB für FB S_MODB
DB 100	CONVERSION_DB	Konvertierungs_DB für FB S_MODB
FB 2	S_RECV_SI	Standard-FB für Daten empfangen
FB 3	S_SEND_SI	Standard-FB für Daten senden
FB 81	S_MODB	Standard-FB für Modbus-Slave Kommunikation
FC 10	Initiation	Datenbausteine initialisieren
FC 21	SEND_SI_0	Daten senden
FC 22	RECV_SI_1	Daten empfangen

## Lieferform und Installation

Das Programmbeispiel des Moduls ET 200S 1SI Modbus/USS liegt zusammen mit den Funktionsbausteinen im Internet unter

<http://support.automation.siemens.com/WW/view/de/10805265/133100>

Nach der Installation befindet sich das Programmbeispiel im Projekt zXX21\_11\_1SI\_MODBUS.

Das Projekt öffnen Sie im SIMATIC-Manager von STEP 7 mit dem Menübefehl "Datei > Öffnen > Beispielprojekte".

Das Programmbeispiel liegt in kompilierter Form und als ASCII-Quelldatei vor. Ebenfalls dabei ist eine Symbolliste der im Beispiel verwendeten Symbole.

Wenn Ihnen kein zweites Modul ET 200S 1SI Modbus/USS als Kommunikationspartner zur Verfügung steht, müssen Sie in HW-Konfig das zweite ET 200S 1SI Modbus/USS mit "Bearbeiten > Löschen" entfernen. Zusätzlich muss im OB 1 der Aufruf des FB 81 (Modbus Slave FB) auskommentiert werden.

## Laden in die CPU

Die Hardware für das Beispiel ist komplett aufgebaut, das Programmiergerät angeschlossen.

Nach dem Urtlöschen der CPU (Betriebsart STOP) übertragen Sie das Beispiel komplett in den Anwenderspeicher. Danach schalten Sie den Betriebsartenschalter von STOP auf RUN.

## Fehlerverhalten

Ist im Anlauf ein Fehler aufgetreten, werden die zyklisch bearbeiteten Bausteinaufrufe nicht ausgeführt, es wird die Fehleranzeige gesetzt.

Bei einer Fehlermeldung wird der Parameterausgang ERROR der Bausteine gesetzt. Eine genauere Fehlerbeschreibung ist dann im Parameter STATUS der Bausteine hinterlegt. Steht eine der Fehlermeldungen 16#1E0E oder 16#1E0F im STATUS, dann ist die genaue Fehlerbeschreibung in der Variablen SFCERR im Instanz-DB abgelegt.

## Einschalten, Anlaufprogramm

Das Anlaufprogramm steht im OB 100.

Im Anlauf werden die Steuerbits und die Zähler rückgesetzt

## Zyklisches Programm

Das zyklische Programm steht im OB 1.

Im Beispiel arbeiten für den Modbus Master die Funktionsbausteine FB 2 S\_RECV\_SI und FB 3 S\_SEND\_SI mit den Funktionen FC 21 und FC 22 zusammen, sowie mit den Datenbausteinen DB 21 und DB 22 als Instanz-DBs und DB 42 und DB 43 als Sende- bzw. Empfangs-DB.

Für den Modbus Slave arbeitet der FB 81 S\_MODB zusammen mit dem DB 81 als Instanz-DB und dem DB 100 als Konvertierungs-DB.

Die Parametrierung der Funktionsbausteine erfolgt im Beispiel teils durch Konstanten und teils durch symbolisch adressierte Aktualoperanden.

## Beschreibung

Bei der Datenübertragung "holt" die ET 200S 1SI Modbus/USS auf Steckplatz 2 (Modbus Master) Daten von der ET 200S 1SI Modbus/USS auf Steckplatz 3 (Modbus Slave). Wenn Sie mit einem anderen Kommunikationspartner arbeiten, entfällt der Aufruf des FB 81 (S\_MODB).

## Beschreibung FC 21 (SEND)

Programmteil "Generate edge S\_SEND\_SI\_REQ":

Der S\_SEND\_SI wird am Anfang einmal mit S\_SEND\_SI\_REQ=0 durchlaufen. Danach wird S\_SEND\_SI\_REQ auf 1 gesetzt. Wenn am Steuerparameter S\_SEND\_SI\_REQ ein Signalzustandswechsel von 0 nach 1 erkannt wird, wird der S\_SEND\_SI Auftrag gestartet.

Mit S\_SEND\_SI\_DONE=1 oder S\_SEND\_SI\_ERROR=1 wird S\_SEND\_SI\_REQ wieder auf 0 gesetzt.

Programmteil "S\_SEND\_SI\_DONE=1":

Bei einem erfolgreichen Transfer wird am Parameterausgang des S\_SEND\_SI der Parameter S\_SEND\_SI\_DONE auf 1 gesetzt.

Um zeitlich nacheinander ablaufende Transfers zu unterscheiden, wird im Datenwort 18 des Arbeitsbaustein DB 40 ein Sendezähler S\_SEND\_SI\_WORK\_CNT\_OK mitgeführt.

Programmteil "S\_SEND\_SI\_ERROR=1":

Wird der S\_SEND\_SI mit S\_SEND\_SI\_ERROR=1 durchlaufen, wird im Datenwort 20 der Fehlerzähler S\_SEND\_SI\_WORK\_CNT\_ERR hochgezählt. Außerdem wird der S\_SEND\_SI\_WORK\_STAT umkopiert, da er im nächsten Durchlauf mit 0 überschrieben wird und dann nicht mehr ausgelesen werden könnte.

## Beschreibung FC 22 (RECEIVE)

Programmteil "Enable Receive Data":

Um Daten empfangen zu können, muss die Empfangsfreigabe S\_RECV\_SI\_EN\_R am Baustein S\_RECV\_SI auf 1 gesetzt sein.

Programmteil "S\_RECV\_SI\_NDR=1":

Wenn S\_RECV\_SI\_NDR gesetzt ist, sind neue Daten empfangen worden und es wird der Empfangszähler S\_RECV\_SI\_WORK\_CNT\_OK hochgezählt.

Programmteil "S\_RECV\_SI\_ERROR=1":

Bei fehlerhaftem Ablauf, d.h. wenn das Errorbit am Parameterausgang des S\_RECV\_SI gesetzt ist, wird der Fehlerzähler S\_RECV\_SI\_WORK\_CNT\_ERR hochgezählt. Außerdem wird der S\_RECV\_SI\_WORK\_STAT umkopiert, da er im nächsten Durchlauf mit 0 überschrieben wird und dann nicht mehr ausgelesen werden könnte.

Alle relevanten Werte können zum Test in der VAT beobachtet werden.

**Beschreibung DB 42**

Mit dem im vorliegenden Beispiel projektierten Auftrag Funktion Code 1 (Read Coil Status) sollen vom Modbus Slave mit der Adresse "1" ab der Startadresse "0" 16 Bit gelesen werden. Die gelesenen 16 Bit werden über den FC 22 (RECV) im Empfangs-DB (DB43) ab Offset-Adresse 0 abgelegt.

Die Parameter des Modbus Master-Auftrags (FC 21 (SEND)) sind im Sende-DB (DB 24) hinterlegt. Siehe folgende Tabelle:

Adresse	Name	Typ	Anfangswert	Kommentar
0.0		STRUC		
+ 0.0	slave_adress	BYTE	B#16#01	von Modbus Slave "1"
+ 1.0	function_code	BYTE	B#16#01	mit FC 1 (Read Coil Status)
+ 2.0	bit_start_adr	WORD	W#16#0000	ab Modbus Startadresse 0
+ 4.0	bit_count	INT	16	16 Bit (1 Wort) lesen
+ 6.0	a	ARRAY [1...1194]		
* 1.0		BYTE		
= 1200.0		END_STRUCT		

**Modbus Master-Auftrag anstoßen**

Sie stoßen den Modbus Master-Auftrag an, indem Sie in der VAT den Merker M 120.7 TRUE setzen.

**Beschreibung DB 100**

Auf der Modbus Slave-Seite werden die angeforderten Daten über den Aufruf des FB 81 (S\_MODB) bereitgestellt.

Die im Modbus Master-Telegramm verwendeten Adressen werden im SIMATIC-Datenbereich im projektierten Konvertierungs-DB (DB 100), wie folgt, abgelegt:

Adresse	Name	Typ	Anfangswert	Kommentar
0.0		STRUCT		
+0.0	FC01_MOD_STRT_ADR_1	WORD	W#16#0	Abbildung der Modbus-Adressen 0 bis 255 auf SIMATIC-Merkerbereich ab 0
+2.0	FC01_MOD_END_ADR_1	WORD	W#16#0FF	
+4.0	FC01_CNV_TO_FLAG_A	WORD	W#16#0	
+6.0	FC01_MOD_STRT_ADR_2	WORD	W#16#100	
+8.0	FC01_MOD_END_ADR_2	WORD	W#16#1FF	
+10.0	FC01_CNV_TO_OUTPUT	WORD	W#16#0	
+12.0	FC01_MOD_STRT_ADR_3	WORD	W#16#200	
+14.0	FC01_MOD_END_ADR_3	WORD	W#16#2FF	
+16.0	FC01_CNV_TO_TIMER	WORD	W#16#0	
+18.0	FC01_MOD_STRT_ADR_4	WORD	W#16#300	
+20.0	FC01_MOD_END_ADR_4	WORD	W#16#3FF	

## 3.2 Kurzanleitung zur Inbetriebnahme der seriellen Schnittstellenbaugruppe

Adresse	Name	Typ	Anfangswert	Kommentar
+22.0	FC01_CNV_TO_COUNTER	WORD	W#16#0	
+24.0	FC02_MOD_STRT_ADR_5	WORD	W#16#0	
+26.0	FC02_MOD_END_ADR_5	WORD	W#16#0FF	
+28.0	FC02_CNV_TO_FLAG_B	WORD	W#16#0	
+30.0	FC02_MOD_STRT_ADR_6	WORD	W#16#100	
+32.0	FC02_MOD_END_ADR_6	WORD	W#16#2FF	
+34.0	FC02_CNV_TO_INPUT	WORD	W#16#0	
+36.0	FC03_06_16_DB_NO	WORD	W#16#02A	
+38.0	FC04_DB_NO	WORD	W#16#02A	
+40.0	DB_MIN	WORD	W#16#02A	
+42.0	DB_MAX	WORD	W#16#02A	
+44.0	FLAG_MIN	WORD	W#16#0	Merkerbereich 0 bis 255 freigeben
+46.0	FLAG_MAX	WORD	W#16#0FF	
+48.0	OUTPUT_MIN	WORD	W#16#0	
+50.0	OUTPUT_MAX	WORD	W#16#0FF	
=52.0		END_STRUCTURE		

Im konkreten Beispiel werden über die Adressen 0 bis 4 des DB 100, die mit einem FC 1 angeforderten Modbus-Adressen 0 bis 255 auf den SIMATIC-Merkerbereich ab 0 abgebildet.

Nach den DB 100-Adresse 44 und 46 ist für Aufträge des Modbus Master der SIMATIC-Merkerbereich 0 bis 255 freigegeben.

### 3.3 Schaltbilder mit Anschlussbelegung

#### 3.3.1 Anschlussbelegung

##### Verdrahtungsrichtlinien

Die Kabel (Klemmen 1 bis 8) müssen geschirmt sein und der Schirm muss beidseitig aufgelegt werden. Verwenden Sie hierzu Schirmkontaktelemente. Informationen zu diesen Elementen finden Sie im Handbuch *Dezentrales Peripheriesystem ET 200S* im Abschnitt *Zubehör*.

##### Anschlussbelegung für die RS-232C-Kommunikation

Mit einem Slave-System können Sie eine Punkt-zu-Punkt-Verbindung aufbauen. Hilfskanäle der RS-232C-Schnittstelle werden nicht unterstützt.

Die Tabelle zeigt die Anschlussbelegung der seriellen Schnittstellenbaugruppe ET 200S Modbus/USS bei eingestelltem RS-232C-Kommunikationsprotokoll.

Tabelle 3-3 Anschlussbelegung für die RS-232C-Kommunikation

Ansicht		Bemerkungen			
<p>TXD 1 □ □ 5 RXD</p> <p>RTS 2 □ □ 6 CTS</p> <p>DTR 3 □ □ 7 DSR</p> <p>DCD 4 □ □ 8 PE</p>		Modus: voll duplex			
		Klemmen			
		1	TXD	Gesendete Daten	
		5	RXD	Empfangene Daten	
		2	RTS	Sendeauftrag	
		6	CTS	Sendebereit	
		3	DTR	Datenterminal bereit	
		7	DSR	Datensatz bereit	
		4	DCD	Datenträgererkennung	
		8	PE	Erde	



### Anschlussbelegung für die RS-422-Kommunikation

Mit einem Slave-System können Sie eine Punkt-zu-Punkt-Verbindung aufbauen.

Die Tabelle zeigt die Anschlussbelegung der seriellen Schnittstellenbaugruppe ET 200S Modbus/USS bei eingestelltem RS-422-Kommunikationsprotokoll.

Tabelle 3- 4 Anschlussbelegung für die RS-422-Kommunikation

Ansicht		Anschlussbelegung		Bemerkungen	
		<p>Hinweis: Bei Kabeln über 50 m ergänzen Sie einen Abschlusswiderstand von ca. 330 Ω, damit problemloser Datenverkehr gewährleistet wird.</p>		Modus: voll duplex	
				Klemmen	
				1	TXD (A) -
				5	RXD (A) -
				2	TXD (B) +
6	RXD (B) +				
8	PE Erde				

### Anschlussbelegung für die RS-485-Kommunikation

Mit einem Master-System können Sie eine Mehrpunktverbindung (Netzwerk) mit bis zu 32 Slaves aufbauen. Der Treiber der Baugruppe schaltet die 2-Draht-Empfangsleitung zwischen Senden und Empfangen um.

Die Tabelle zeigt die Anschlussbelegung der seriellen Schnittstellenbaugruppe ET 200S Modbus/USS bei eingestelltem RS-485-Kommunikationsprotokoll.

Tabelle 3- 5 Anschlussbelegung für die RS-485-Kommunikation

Ansicht		Anschlussbelegung		Bemerkungen	
		<p>Hinweis: Bei Kabeln über 50 m ergänzen Sie einen Abschlusswiderstand von ca. 330 Ω, damit problemloser Datenverkehr gewährleistet wird.</p>		Modus: voll duplex	
				Klemmen	
				1	R/T (A) -
				2	R/T (B) +
				8	PE Erde

### Anschlussbelegung des RS-232C-Anschlusskabels für 9-poligen Anschlussstecker

Das folgende Bild zeigt die Leitungsanschlüsse für die Punkt-zu-Punkt-Kommunikation nach RS-232C zwischen der Baugruppe und einem Kommunikationslave mit einer 9-poligen D-Anschlussbuchse.

- Auf der ET 200S-Seite werden die Signaldrähte an die entsprechend nummerierten Klemmen angeschlossen.
- Am Kommunikationslave verwenden Sie eine 9-polige Sub-D-Anschlussbuchse.

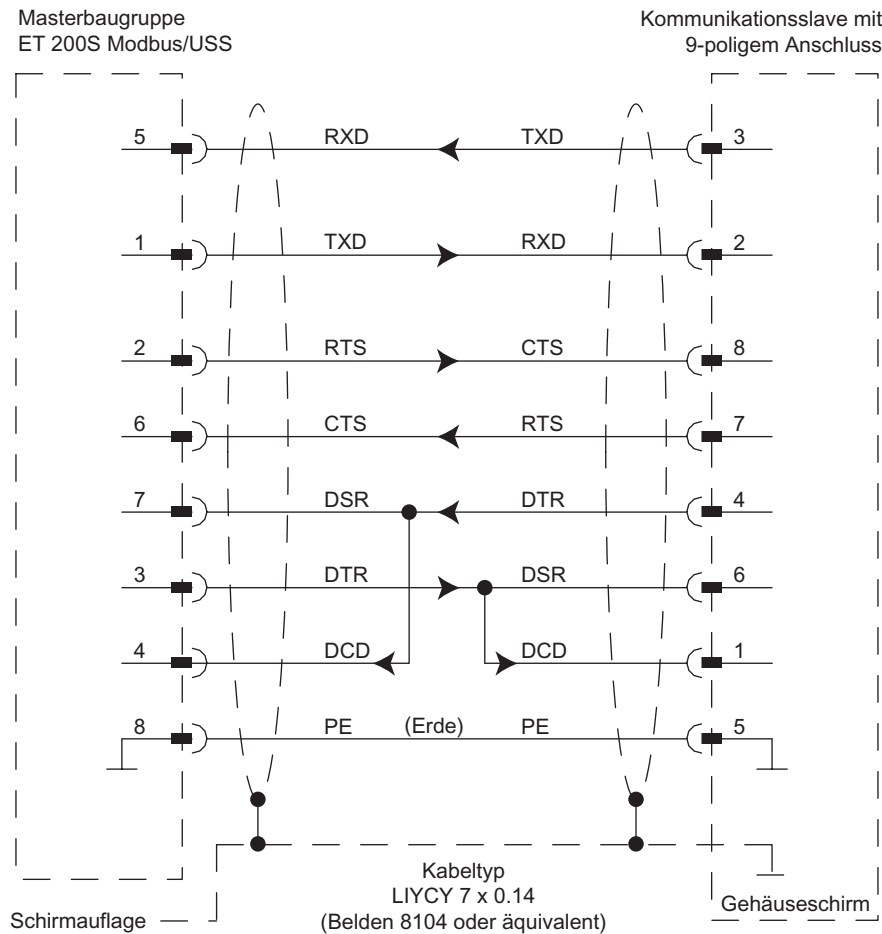


Bild 3-2 RS-232C-Anschlusskabel für 9-poligen Anschlussstecker (System 1 Master, 1 Slave)

### Anschlussbelegung des RS-232C-Anschlusskabels für 25-poligen Anschlussstecker

Das folgende Bild zeigt die Leitungsanschlüsse für die Punkt-zu-Punkt-Kommunikation nach RS-232C zwischen der Baugruppe und einem Kommunikationslave mit einem 25-poligen D-Anschlussstecker.

- Auf der ET 200S-Seite werden die Signaldrähte an die entsprechend nummerierten Klemmen angeschlossen.
- Am Kommunikationslave verwenden Sie einen 25-poligen Sub-D-Anschlussstecker.

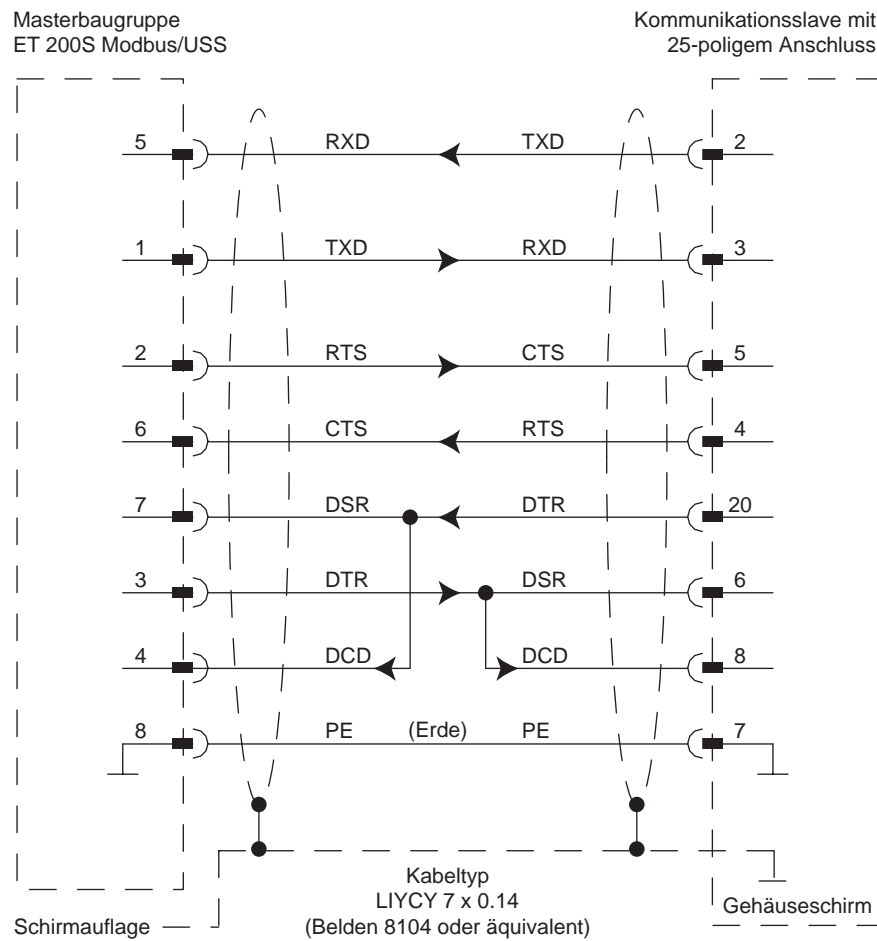


Bild 3-3 RS-232C-Anschlusskabel für 25-poligen Anschlussstecker (System 1 Master, 1 Slave)

### Anschlussbelegung des RS-422-Anschlusskabels für 15-poligen Anschlussstecker

Das folgende Bild zeigt die Leitungsanschlüsse für die Kommunikation nach RS-422 zwischen der Baugruppe und einem Kommunikationslave mit einem 15-poligen D-Anschlussstecker.

- Auf der ET 200S-Seite werden die Signaldrähte an die entsprechend nummerierten Klemmen angeschlossen.
- Am Kommunikationslave verwenden Sie einen 15-poligen Sub-D-Anschlussstecker.

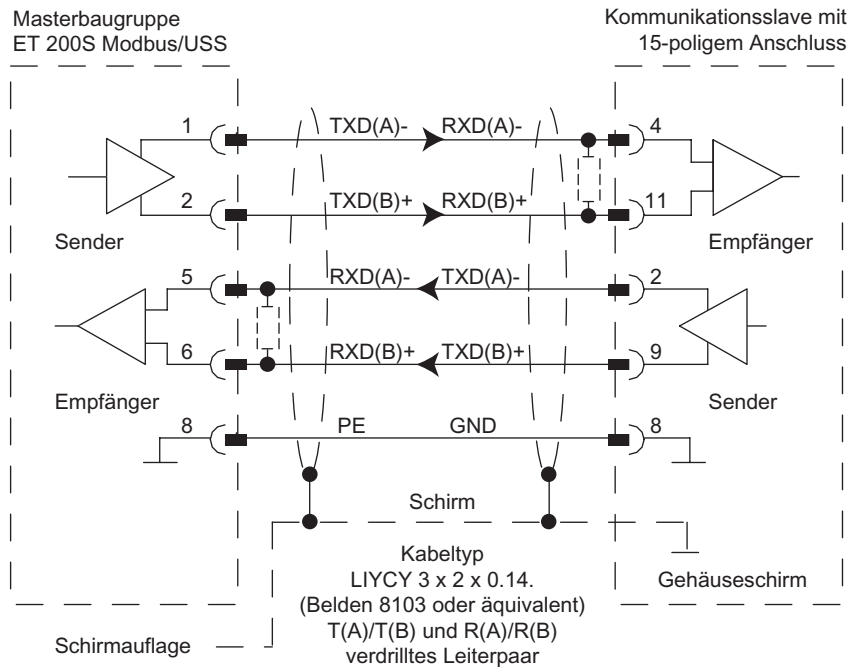


Bild 3-4 RS-422-Anschlusskabel für 15-poligen Anschlussstecker (System 1 Master, 1 Slave)

#### Hinweis

Bei Kabeln über 50 m ergänzen Sie einen Abschlusswiderstand von ca. 330 Ω , wie im vorherigen Bild dargestellt, damit problemloser Datenverkehr gewährleistet ist.

Maximal darf dieser Kabeltyp bei 38.400 Baud eine Länge von 1200 m haben.

- max. 1200 m bei 19.200 Baud
- max. 500 m bei 38.400 Baud
- max. 250 m bei 76.800 Baud

### Anschlussbelegung des RS-485-Anschlusskabels für 15-poligen Anschlussstecker

Das folgende Bild zeigt die Leitungsanschlüsse für die Kommunikation nach RS-485 zwischen der Baugruppe und einem Kommunikationslave mit einem 15-poligen D-Anschlussstecker.

- Auf der ET 200S-Seite werden die Signaldrähte an die entsprechend nummerierten Klemmen angeschlossen.
- Am Kommunikationslave verwenden Sie einen 15-poligen Sub-D-Anschlussstecker.

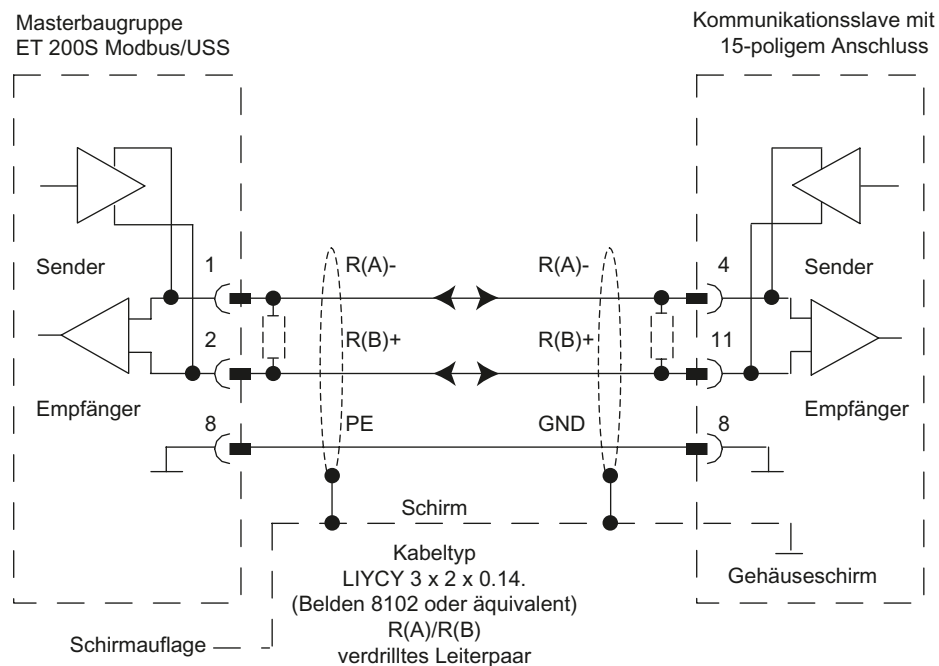


Bild 3-5 RS-485-Anschlusskabel für 15-poligen Anschlussstecker (System 1 Master, 1 Slave)

#### Hinweis

Bei Kabeln über 50 m ergänzen Sie einen Abschlusswiderstand von ca. 330  $\Omega$ , wie im vorherigen Bild argestellt, damit problemloser Datenverkehr gewährleistet ist.

Maximal darf dieser Kabeltyp bei 38.400 Baud eine Länge von 1200 m haben.

- max. 1200 m bei 19.200 Baud
- max. 500 m bei 38.400 Baud
- max. 250 m bei 76.800 Baud
- max. 200 m bei 115.200 Baud

### 3.3.2 RS-232C-Schnittstelle

#### Eigenschaften der RS-232C-Schnittstelle

Die RS-232C-Schnittstelle ist eine Spannungsschnittstelle und dient zur seriellen Datenübertragung nach der Norm RS-232C. Die Tabelle zeigt die Eigenschaften für RS-232C.

Tabelle 3- 6 Signale der RS-232C-Schnittstelle

Eigenschaft	Beschreibung
Art	Spannungsschnittstelle
Frontstecker	8-poliger Standardklemmenstecker der ET 200S
RS-232C-Signale	TXD, RXD, RTS, CTS, DTR, DSR, DCD, GND
Übertragungsgeschwindigkeit	maximal 115,2 Kbaud
Leitungslänge	maximal 15 m, Kabeltyp LIYCY 7 x 0.14
Normen	DIN 66020, DIN 66259, EIA RS-232C, CCITT V.24/V.28
Schutzart	IP20

#### RS-232C-Signale

Die Modbus/USS-Baugruppe unterstützt die RS-232C-Signale.

Tabelle 3- 7 Signale der RS-232C-Schnittstelle

Signal	Bezeichnung	Bedeutung
TXD	Gesendete Daten	Sendedaten; Sendeleitung wird im Ruhezustand logisch auf "1" gehalten.
RXD	Empfangene Daten	Empfangsdaten; Empfangsleitung muss vom Kommunikationspartner auf logisch "1" gehalten werden.
RTS	Sendeauftrag	ON: Baugruppe ist sendebereit. OFF: Baugruppe sendet nicht.
CTS	Sendebereit	Kommunikationspartner kann Daten vom ET 200S empfangen. Die serielle Schnittstellenbaugruppe erwartet dies als Antwort auf RTS gleich ON.
DTR	Datenterminal bereit	ON: Die Baugruppe ist eingeschaltet und betriebsbereit. OFF: Die Baugruppe ist nicht eingeschaltet und nicht betriebsbereit.
DSR	Datensatz bereit	ON: Kommunikationspartner ist eingeschaltet und betriebsbereit. OFF: Kommunikationspartner ist nicht eingeschaltet und nicht betriebsbereit.
DCD	Datenträgererkennung	Trägersignal bei Anschluss eines Modems.

## Automatische Bedienung der Begleitsignale

Die automatische Bedienung der RS-232C-Begleitsignale in der Baugruppe ist folgendermaßen implementiert:

- Sobald die Baugruppe durch Parametrierung in eine Betriebsart mit automatischer Bedienung der RS 232C-Begleitsignale gebracht wurde, setzt sie die Leitungen RTS auf OFF und DTR auf ON (Baugruppe betriebsbereit).

Das Senden und Empfangen von Telegrammen ist erst möglich, nachdem die Leitung DTR auf ON gesetzt wurde. Solange DTR auf OFF gesetzt bleibt, werden keine Daten über die RS-232C-Schnittstelle empfangen. Ein Sendeauftrag wird mit einer entsprechender Fehlermeldung abgebrochen.

- Steht ein Sendeauftrag an, wird RTS auf ON gesetzt und die parametrierte Datenausgabewartezeit gestartet. Nach Ablauf der Datenausgabezeit und CTS = ON werden die Daten über die RS 232C-Schnittstelle gesendet.
- Wird beim Senden innerhalb der Datenausgabewartezeit die Leitung CTS nicht auf ON gesetzt, oder erfolgt innerhalb des Sendevorgangs ein Wechsel von CTS auf OFF wird der Sendeauftrag abgebrochen und eine entsprechende Fehlermeldung generiert.
- Nach dem Senden der Daten wird nach Ablauf der parametrierten RTS-Wegnahmezeit die Leitung RTS auf OFF gesetzt. Die ET 200S wartet nicht auf den Wechsel von CTS auf OFF.
- Ein Empfang von Daten über die RS-232C-Schnittstelle ist möglich, sobald die Leitung DSR auf ON gesetzt ist. Droht der Empfangspuffer der Baugruppe überzulaufen, erfolgt keine Reaktion der Baugruppe.
- Bei einem Wechsel von DSR = ON auf OFF wird sowohl ein laufender Sendeauftrag als auch das Empfangen von Daten mit einer Fehlermeldung abgebrochen.

---

### Hinweis

Die automatische Bedienung der RS-232C-Begleitsignale ist nur im Halbduplexbetrieb möglich.

---

### Hinweis

"Zeit bis RTS OFF" muss in der Parametrierungsschnittstelle eingestellt werden, so dass der Kommunikationspartner die letzten Zeichen des Telegramms vollständig empfangen kann, bevor RTS, und damit der Sendeauftrag, weggenommen wird. Die "Datenausgabewartezeit" muss eingestellt sein, damit der Kommunikationspartner empfangsbereit sein kann, bevor die Zeit abläuft.

---

### Zeitdiagramm für Begleitsignale

Das folgende Bild zeigt den zeitlichen Ablauf eines Sendeauftrags:

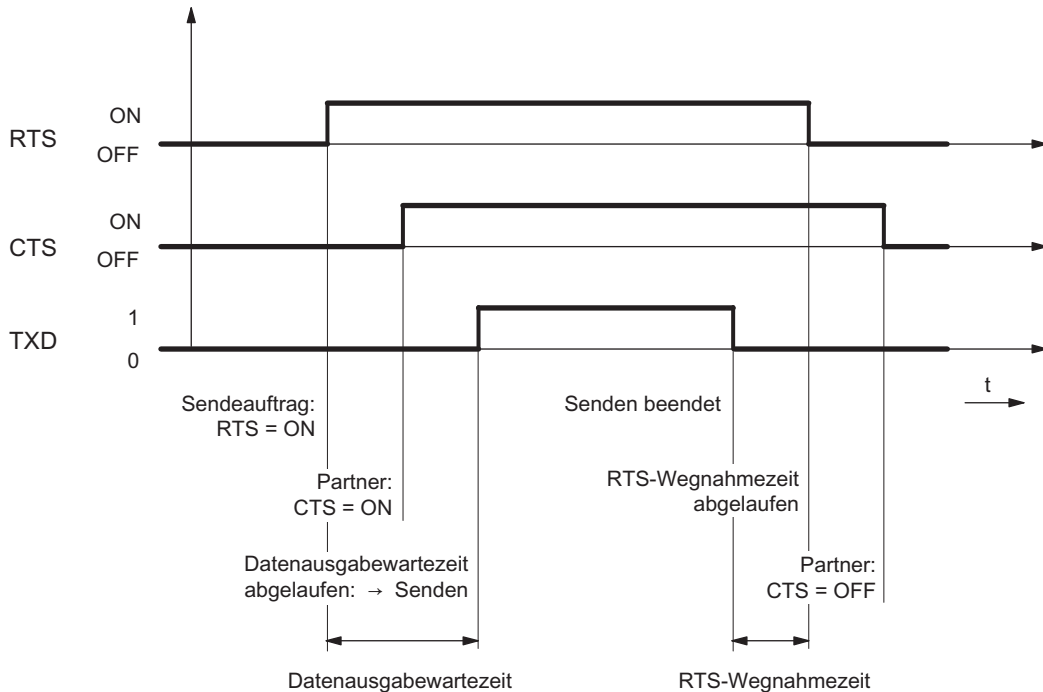


Bild 3-6 Zeitdiagramm bei automatischer Bedienung der RS-232C-Begleitsignale

### 3.3.3 RS-422/485-Schnittstelle

#### Eigenschaften der RS-422/458-Schnittstelle

Die RS-422/485-Schnittstelle ist eine Spannungsdifferenzschnittstelle und dient zur seriellen Datenübertragung nach der Norm RS-422/485. Die Tabelle zeigt die Eigenschaften der RS-422/485-Schnittstelle.

Tabelle 3-8 Eigenschaften der RS-422/485-Schnittstelle

Eigenschaft	Beschreibung
Art	Spannungsdifferenzschnittstelle
Frontstecker	8-poliger Standardklemmenstecker der ET 200S
RS-422-Signale	TXD (A)-, RXD (A)-, TXD (B)+, RXD (B)+, GND
RS-485-Signale	R/T (A)-, R/T (B)+, GND
Übertragungsgeschwindigkeit	maximal 115,2 KBAud
Leitungslänge	maximal 1200 m, Kabeltyp LIYCY 7 x 0.14
Normen	EIA RS-422/485, CCITT V.11/V.27
Schutzart	IP20



## 3.4 Modbus-Übertragungsprotokoll

### 3.4.1 Eigenschaften und Telegrammaufbau

#### Eigenschaften

Bei der für die Modbus-Übertragung verwendeten Prozedur handelt es sich um eine codetransparente, asynchrone Halbduplexprozedur. Die Datenübertragung erfolgt ohne Handshake.

Die Baugruppe stößt die Übertragung (als Master) an. Nach Ausgabe des Auftragstelegramms wartet die Baugruppe während der Antwortüberwachungszeit auf ein Antworttelegramm vom Slave.

#### Telegrammaufbau

Der Datenverkehr "Master-Slave" bzw. "Slave-Master" beginnt mit der Slave-Adresse, gefolgt vom Funktionscode. Dann werden die Daten übertragen. Der Datenaustausch "Master-Slave" bzw. "Slave-Master" verfügt über folgende Elemente:

SLAVE-ADRESSE	Modbus-Slave-Adresse
FUNKTIONSCODE	Modbus-Funktionscode
Daten	Telegrammdaten: Byte_Count, Coil_Number, Data
CRC-PRÜFUNG	Telegramm-Prüfsumme

Der Aufbau des Datenfelds ist abhängig vom verwendeten Funktionscode. Am Ende des Telegramms wird die CRC-Prüfung übertragen. Die Tabelle zeigt die Komponenten des Telegrammaufbaus.

Tabelle 3- 9 Telegrammaufbau

Adresse	Funktion	Daten	CRC-PRÜFUNG
Byte	Byte	n Byte	2 Byte

### 3.4.2 Slave-Adresse

#### Beschreibung

Die Slave-Adresse kann im Bereich von 1 bis 247 liegen. Mit der Adresse wird ein definierter Slave am Bus angesprochen.

#### Übertragungstelegramm

Mit der Slave-Adresse Null spricht der Master alle Slaves am Bus an.

---

#### Hinweis

Übertragungstelegramme sind nur in Verbindung mit den Funktionscodes 05, 06, 15 bzw. 16 zulässig.

---

Bei einem Übertragungstelegramm wird vom Slave kein Antworttelegramm gesendet.

### 3.4.3 Master- und Slave-Funktionscodes

#### Funktionscodes für Master und Slave

Der Funktionscode definiert die Bedeutung und den Aufbau des Telegramms. Die Tabelle führt die Funktionscodes sowie deren Verfügbarkeit für Master und Slaves auf.

Tabelle 3- 10 Master- und Slave-Funktionscodes

Funktionscode	Beschreibung	Master	Slave
01	Read Coil Status	√	√
02	Read Input Status	√	√
03	Read Holding Registers	√	√
04	Read Input Registers	√	√
05	Force Single Coil	√	√
06	Preset Single Register	√	√
07	Read Exception Status	√	-
08	Loop Back Test	√	√
11	Fetch Communications Event Counter	√	-
12	Fetch Communications Event Log	√	-
15	Force Multiple Coils	√	√
16	Preset Multiple Registers	√	√

### 3.4.4 Datenfeld DATA

#### Beschreibung

Im Datenfeld DATA werden die funktionscode-spezifischen Daten übertragen:

- Byte count
- Coil Start Address
- Register Start Address
- Number of Coils
- Number of Registers

### 3.4.5 Telegrammende und CRC-Prüfung

#### Beschreibung

Den Telegrammabschluss bildet die zwei Byte lange CRC-16-Prüfsumme. Sie wird nach folgendem Polynom berechnet:

$$x^{16} + x^{15} + x^2 + 1$$

Zuerst wird das Low-Byte, dann das High-Byte übertragen.

#### Telegrammende-Erkennung

Die Modbus/USS-Baugruppe erkennt das Telegrammende dann, wenn während der Zeit, die dreieinhalb Zeichen zur Übermittlung benötigen (3,5-fache Zeichenverzugszeit), keine Übertragung stattfindet.

Das Telegrammende-Timeout richtet sich nach der Baudrate.

Bei Ablauf des Telegrammende-Timeouts wird das vom Slave empfangene Antworttelegramm ausgewertet und formal geprüft.

Tabelle 3- 11 Telegrammende

Übertragungsgeschwindigkeit	Timeout
115.200 bps	1 ms
76.800 bps	1 ms
57.600 bps	1 ms
38.400 bps	1 ms
19.200 bps	2 ms
9.600 bps	4 ms
4.800 bps	8 ms
2.400 bps	16 ms
1.200 bps	32 ms
600 bps	65 ms
300 bps	130 ms
115 bps	364 ms

### 3.4.6 Ausnahmeantworten

#### Antworttelegramm im Fehlerfall

Erkennt der Slave einen Fehler im Auftragstelegramm des Masters (z.B. unzulässige Registeradresse), dann führt der Slave folgende Aktionen aus:

- Der Slave setzt das höchstwertige Bit im Funktionscode des Antworttelegramms.
- Der Slave sendet ein Byte Fehlercode (Ausnahmecode), um die Fehlerursache zu beschreiben.

#### Beispiel: Ausnahmecodetelegramm

Das Fehlercode-Antworttelegramm vom Slave hat beispielsweise folgenden Aufbau: Slave-Adresse 5, Funktionscode 5, Ausnahmecode 2.

<b>Antworttelegramm vom Slave EXCEPTION_CODE_xx</b>	05H	Slave-Adresse
	85H	Funktionscode
	02H	Ausnahmecode (1 bis 7)
	xxH	CRC-Prüfcode "Low"
	xxH	CRC-Prüfcode "High"

Bei Empfang eines Fehlercode-Antworttelegramms vom Treiber wird der aktuelle Auftrag mit Fehler beendet.

Außerdem wird eine Fehlernummer, die dem empfangenen Fehlercode entspricht (Ausnahmecode 1-7), in den Bereich SYSTAT eingetragen.

In einen Zieldatenbaustein S\_RCV wird kein Eintrag vorgenommen.

#### Tabelle der Fehlercodes

Die Tabelle führt die Fehlercodes auf, die von der Baugruppe gesendet werden.

Tabelle 3- 12 Fehlercodes

Ausnahmecode	Beschreibung	Mögliche Ursache
01	Unzulässige Funktion	Unzulässiger Funktionscode empfangen.
02	Unzulässige Datenadresse	Zugriff auf einen SIMATIC-Bereich, der nicht freigegeben ist (siehe Modbus -- Datenumsetzungstabelle)
03	Unzulässiger Datenwert	Länge größer als 2040 Bit oder 127 Register, Datenfeld nicht FF00 oder 0000 für FC05, Diagnose-Subcode <> 0000 für FC08.
04	Ausfall in zugehörigem Gerät	Initialisierung durch Modbus-Kommunikations-FB noch nicht durchgeführt oder FB meldet Fehler. Fehler bei der Datenübertragung Baugruppe – CPU (Beispiel: DB nicht vorhanden, maximal übertragbare Datenlänge überschritten (Blockgröße CPU <-> Baugruppe).

## 3.5 Modbus-Master-Treiber

### 3.5.1 Einsatz des Modbus-Master-Treibers

#### Einsatzzweck

Der Modbus-Treiber ET 200S ist in den S7-Automatisierungssystemen einsetzbar und kann serielle Kommunikationsverbindungen zu Partnersystemen herstellen.

Mit diesem Treiber wird eine Kommunikationsverbindung zwischen dem Modbus-Master-Treiber ET 200S und modbusfähigen Steuerungssystemen ermöglicht.

#### Ablauf der Übertragung

Zur Übertragung wird das Modbus-Protokoll im RTU-Format verwendet. Die Datenübertragung wird nach dem Master-Slave-Prinzip abgewickelt.

Der Master stößt die Übertragung an.

Die Funktionscodes 01, 02, 03, 04, 05, 06, 07, 08, 11, 12, 15 und 16 können vom Modbus-Master verwendet werden.

#### Einsetzbare Schnittstellen und Protokolle

Sie können die Schnittstellen RS-232 oder RS-422/485 (X27) für die Baugruppe verwenden.

Die RS422/485-Schnittstelle kann bei diesem Treiber sowohl im 2-Draht- als auch im 4-Draht-Betrieb eingesetzt werden. Im 2-Draht-Betrieb ist es möglich, bis zu 32 Slaves an einen Master im Halbduplex-Betrieb anzuschließen. So wird eine Mehrpunktverbindung (Netzwerk) gebildet. Im 4-Draht-Betrieb (RS-422) sind nur 1 Master und 1 Slave im Halbduplex-Betrieb möglich.

### 3.5.2 Datenübertragung beim Modbus-Master ET 200S

#### Einleitung

Die Datenübergabe zwischen Baugruppe und CPU erfolgt mittels der FBs S\_SEND und S\_RCV. Der FB S\_SEND wird mit einer Flanke am Eingang REQ aktiviert, wenn Daten ausgegeben werden sollen. Der FB S\_RCV wird mit EN\_R=1 empfangsbereit gemacht. Ein S\_RCV ist bei allen lesenden Funktionscodes notwendig.

#### FB3 S\_SEND: Daten an einen Kommunikationspartner senden

Für die Ausführung eines Modbus-Master-Auftrags müssen die FBs S\_SEND und S\_RCV aktiviert werden. Der FB S\_SEND wird mit einer Flanke am Eingang REQ aktiviert, wenn Daten an die Baugruppe ausgegeben werden sollen. Der FB S\_RCV wird mit EN\_R=1 für den Empfang von Daten von der Baugruppe vorbereitet. Ein S\_RCV ist bei allen lesenden

Funktionscodes notwendig. Das folgende Bild zeigt das Gesamtverhalten der Parameter S\_SEND und S\_RCV, wenn ein Modbus-Auftrag ausgeführt wird.

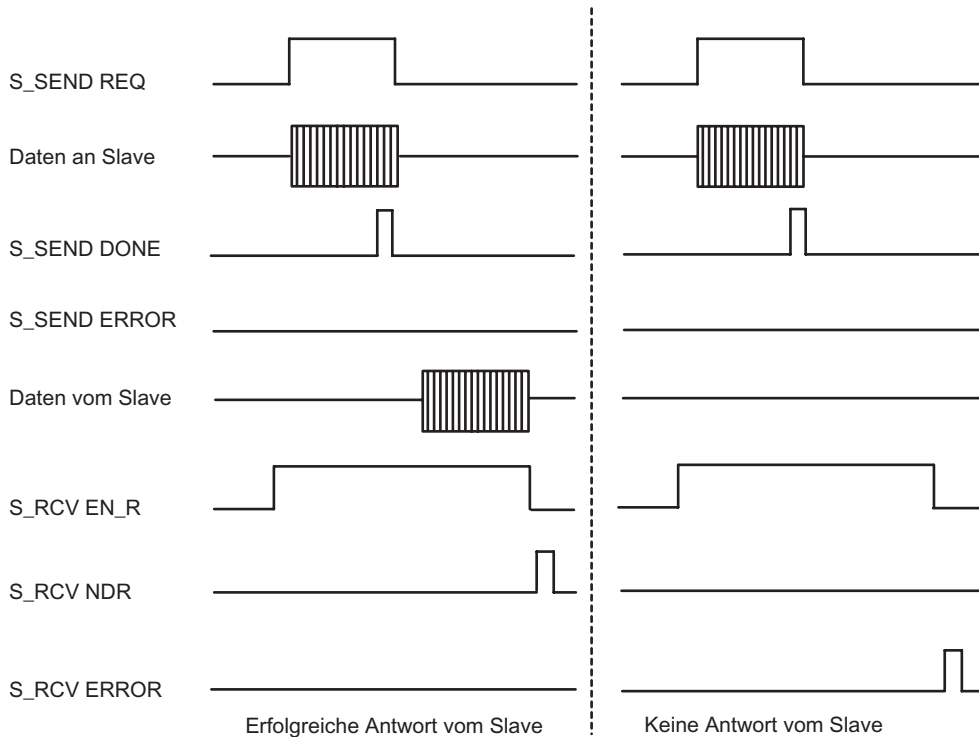


Bild 3-7 Zeitdiagramm eines Modbus-Auftrags

Mit einer positiven Flanke am Eingang REQ wird die Übertragung der Daten angestoßen. Je nach Datenmenge kann eine Datenübertragung über mehrere Aufrufe (Programmzyklen) laufen.

Der Funktionsbaustein FB S\_SEND kann mit Signalzustand "1" am Parametereingang R im Zyklus aufgerufen werden. Es wird damit die Übertragung zur Baugruppe abgebrochen und der FB S\_SEND in den Grundzustand versetzt. Daten, die die Baugruppe bereits erhalten hat, werden noch an den Kommunikationspartner gesendet. Steht statisch Signalzustand "1" am Eingang R an, so ist das Senden ausgeschaltet.

An dem Parameter LADDR wird die Adresse der anzusprechenden seriellen Schnittstellenbaugruppe ET 200S Modbus/USS angegeben.

Der Ausgang DONE zeigt "Auftragsende ohne Fehler" an. ERROR zeigt einen aufgetretenen Fehler an. Im STATUS wird bei einem Fehler die entsprechende Ereignisnummer angezeigt. Ist kein Fehler aufgetreten, hat STATUS den Wert 0. DONE und ERROR/STATUS werden auch bei RESET des FB S\_SEND ausgegeben. Bei einem aufgetretenen Fehler wird das Binäregebnis BIE zurückgesetzt. Wird der Baustein ohne Fehler beendet, hat das Binäregebnis den Zustand "1".

## Modbus-Master-Leseauftrag

Da die Schnittstelle zwischen dem Anwenderprogramm und der Schnittstellenbaugruppe halbduplex arbeitet, müssen Sie folgendes beachten:

Nach einem positiv quittierten Modbus-Master-Leseauftrag müssen Sie zuerst mit dem Funktionsbaustein S\_RCV die Empfangsdaten von der Schnittstellenbaugruppe abholen, bevor Sie einen neuen Modbus-Master-Sendeauftrag starten.

## Anlauf

Der Parameter COM\_RST des FB S\_SEND dient dazu, dem FB einen Anlauf mitzuteilen.

Setzen Sie den Parameter COM\_RST im Anlauf-OB auf 1.

Rufen Sie den FB im zyklischen Betrieb auf, ohne den Parameter COM\_RST zu setzen oder rückzusetzen.

Wenn der Parameter COM\_RST gesetzt ist,

- ermittelt der FB Informationen über das Modul ET 200S Modbus/USS (Anzahl Byte im Peripheriebereich, in dezentraler Peripherie oder nicht).
- setzt sich der FB zurück und beendet dabei evtl. einen zuvor (vor dem letzten Stopp-Übergang der CPU) begonnenen Auftrag.

Wenn der FB die Information über das Modul ET 200S Modbus/USS ermittelt hat, setzt er den Parameter COM\_RST selbst zurück.

Die folgende Tabelle zeigt die AWL- und KOP-Darstellungen des FB3 S\_SEND.

---

### Hinweis

Der Eingang REQ ist flankengetriggert. Es genügt am Eingang REQ eine positive Flanke. Es muss nicht während der gesamten Übertragung das VKE (Verknüpfungsergebnis) auf "1" sein.

---

### Hinweis

Der Eingang EN\_R ist statisch auf "1" zu legen. Während des gesamten Empfangsauftrages muss der Parameter EN\_R mit dem VKE "1" (Verknüpfungsergebnis) versorgt werden.

---

### Hinweis

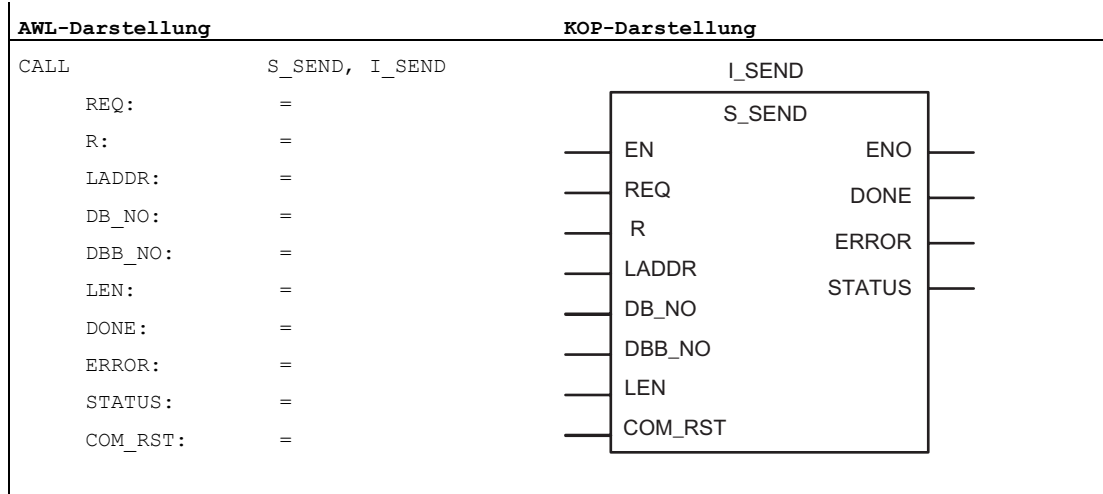
Der Funktionsbaustein S\_SEND hat keine Parameterprüfung. Bei ungültigen Parametern geht die CPU in den Betriebszustand STOP.

Bevor ein angestoßener Auftrag nach einem Zustandsübergang der CPU von STOP nach RUN von der Baugruppe bearbeitet werden kann, muss der ET 200S-CPU-Anlaufmechanismus des FB S\_SEND abgeschlossen sein. Ein in der Zwischenzeit angestoßener Auftrag geht nicht verloren. Er wird nach Abschluss der Anlaufkoordinierung zur Baugruppe übertragen.

---

### Aufruf von FB3

Die Tabelle zeigt die AWL- und KOP-Darstellungen des FB3 S\_SEND.



#### Hinweis

Die Parameter EN und ENO sind nur in der grafischen Darstellung vorhanden (bei KOP oder FUP). Für die Bearbeitung dieser Parameter wird vom Compiler das Binärergebnis BIE verwendet.

Das Binärergebnis BIE wird auf den Signalzustand "1" gesetzt, wenn der Baustein fehlerfrei beendet wurde. Liegt ein Fehler vor, wird das Binärergebnis BIE auf "0" gesetzt.

### Zuordnung im Datenbereich

Der FB S\_SEND arbeitet mit einem Instanz-DB I\_SEND zusammen. Die DB-Nummer wird beim Aufruf mitgegeben. Ein Zugriff auf die Daten im Instanz-DB ist nicht zulässig.

#### Hinweis

Ausnahme: Im Fehlerfall, STATUS == W#16#1E0F, können Sie einen genaueren Fehlerhinweis aus der Variablen SFCERR entnehmen. Diese Fehlervariable kann nur über einen symbolischen Zugriff auf den Instanz-DB geladen werden.



## Parameter von FB3 S\_SEND

Die Tabelle führt die Parameter von S\_SEND (FB3) auf.

Tabelle 3- 13 FB3: Parameter S\_SEND

Name	Art	Datentyp	Beschreibung	Zulässige Belegung, Bemerkung
REQ	INPUT	BOOL	Auftragsanstoß bei positiver Flanke	
R	INPUT	BOOL	Auftragsabbruch	Laufender Auftrag wird abgebrochen. Senden gesperrt.
LADDR	INPUT	INT	Basisadresse der seriellen Schnittstelle ET 200S	Die Basisadresse wird aus STEP 7 entnommen
DB_NO	INPUT	INT	Datenbausteinnummer	Sende-DB-Nr.: CPU-spezifisch, Null ist nicht zulässig
DBB_NO	INPUT	INT	Datenbytenummer	0 ≤ DBB_NO ≤ 8190 Übertragene Daten per Datenwort
LEN	INPUT	INT	Datenlänge	1 ≤ LEN ≤ 224 Angabe in Anzahl Byte
DONE <sup>1</sup>	OUTPUT	BOOL	Auftrag fertig ohne Fehler	Parameter STATUS = = 16#00
ERROR <sup>1</sup>	OUTPUT	BOOL	Auftrag fertig mit Fehler	Parameter STATUS enthält die Fehlerinformation
STATUS <sup>1</sup>	OUTPUT	WORD	Spezifikation des Fehlers	Bei ERROR = = 1 steht im Parameter STATUS die Fehlerinformation
COM_RST	IN_OUT	BOOL	Neustart des FB	

<sup>1</sup> Diese Parameter stehen nach korrektem Sendeauftrag **einen** CPU-Zyklus lang zur Verfügung.

### Zeitdiagramm für FB3 S\_SEND

Das folgende Bild zeigt das Verhalten der Parameter DONE und ERROR je nachdem, wie die Eingänge REQ und R verdrahtet sind.

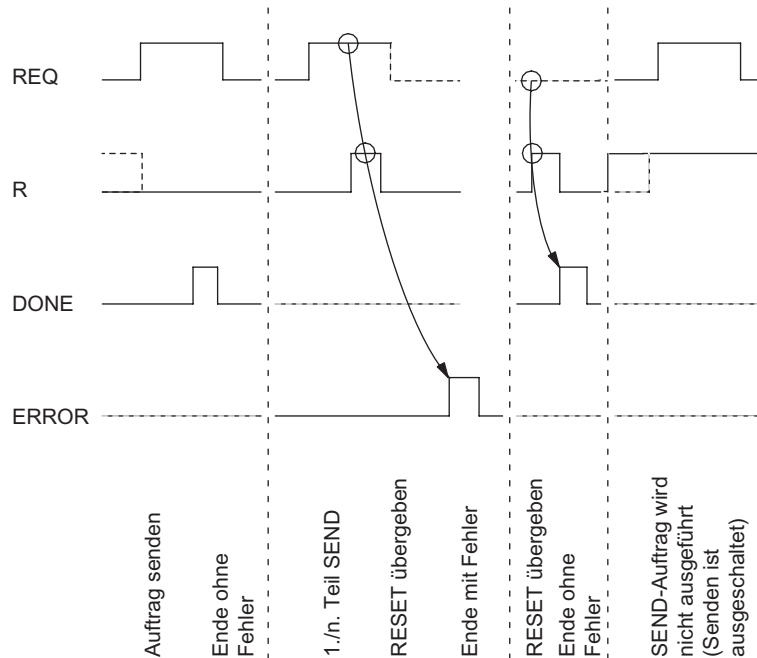


Bild 3-8 Zeitablaufdiagramm FB3 S\_SEND

#### Hinweis

Der Eingang REQ ist flankengetriggert. Es genügt am Eingang REQ eine positive Flanke. Es muss nicht während der gesamten Übertragung das VKE (Verknüpfungsergebnis) auf "1" sein.

### FB2 S\_RCV: Daten von einem Kommunikationspartner empfangen

Der FB S\_RCV überträgt Daten von der Baugruppe zu einem S7-Datenbereich, spezifiziert durch die Parameter DB\_NO und DBB\_NO. Der FB S\_RCV wird zur Datenübertragung im Zyklus oder alternativ in einem zeitgesteuerten Programm statisch (ohne Bedingungen) aufgerufen.

Mit (statisch) Signalzustand "1" am Parameter EN\_R wird die Überprüfung, ob Daten von der seriellen Schnittstelle zu lesen sind, freigegeben. Eine laufende Übertragung kann mit Signalzustand "0" am Parameter EN\_R abgebrochen werden. Der abgebrochene Empfangsauftrag wird mit einer Fehlermeldung (STATUS-Ausgang) beendet. Der Empfang ist ausgeschaltet, solange Signalzustand "0" am Parameter EN\_R ansteht. Je nach Datenmenge kann eine Datenübertragung über mehrere Aufrufe (Programmzyklen) laufen.

Erkennt der Funktionsbaustein Signalzustand "1" am Parameter R, dann wird der momentane Übertragungsauftrag abgebrochen und der FB S\_RCV in den Grundzustand versetzt. Der Empfang ist ausgeschaltet, solange Signalzustand "1" am Parameter R ansteht. Bei erneutem Signalzustand "0" wird das abgebrochene Telegramm erneut, von Anfang an, empfangen.

An dem Parameter LADDR wird die anzusprechende serielle Schnittstellenbaugruppe ET 200S Modbus/USS angegeben.

Der Ausgang NDR zeigt "Auftrag fertig ohne Fehler/Daten übernommen" an (alle Daten gelesen). ERROR zeigt einen aufgetretenen Fehler an. Ist ein Fehler aufgetreten, wird die entsprechende Fehlernummer in STATUS angezeigt, wenn der Empfangspuffer zu mehr als 2/3 voll ist. STATUS enthält nach jedem Aufruf von S\_RCV eine Warnung, wenn ERROR nicht gesetzt ist. Sind keine Fehler oder Warnungen aufgetreten, hat STATUS den Wert 0.

NDR und ERROR/STATUS werden auch bei RESET des FB S\_RCV ausgegeben (Parameter LEN == 16#00). Bei einem aufgetretenen Fehler wird das Binärergebnis BIE zurückgesetzt. Wird der Baustein ohne Fehler beendet, hat das Binärergebnis den Zustand "1".

## Anlauf

Der Parameter COM\_RST des FB S\_RCV dient dazu, dem FB einen Anlauf mitzuteilen.

Setzen Sie den Parameter COM\_RST im Anlauf-OB auf 1.

Rufen Sie den FB im zyklischen Betrieb auf, ohne den Parameter COM\_RST zu setzen oder rückzusetzen.

Wenn der Parameter COM\_RST gesetzt ist,

- ermittelt der FB Informationen über das Modul ET 200S Modbus/USS (Anzahl Byte im Peripheriebereich, in dezentraler Peripherie oder nicht).
- setzt sich der FB zurück und beendet dabei evtl. einen zuvor (vor dem letzten Stopp-Übergang der CPU) begonnenen Auftrag.

Wenn der FB die Information über das Modul ET 200S Modbus/USS ermittelt hat, setzt er den Parameter COM\_RST selbst zurück.

---

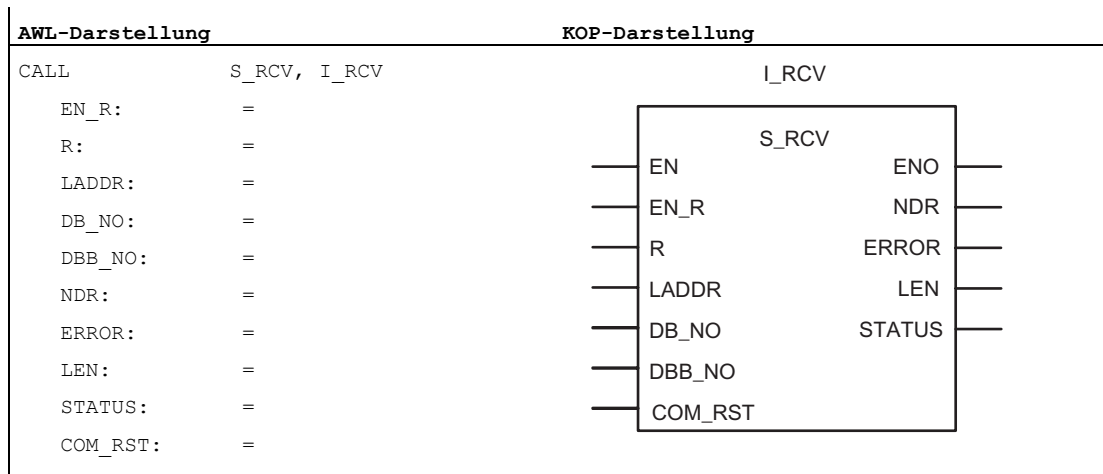
### Hinweis

Der Funktionsbaustein S\_RCV hat keine Parameterprüfung. Bei ungültigen Parametern kann die CPU in den Betriebszustand STOP gehen.

Bevor ein Auftrag nach einem Zustandsübergang der CPU von STOP nach RUN von der Baugruppe empfangen werden kann, muss der ET 200S-CPU-Anlaufmechanismus des FB S\_RCV abgeschlossen sein.

---

Die Tabelle zeigt die AWL- und KOP-Darstellungen des FB2 S\_RCV.



**Hinweis**

Die Parameter EN und ENO sind nur in der grafischen Darstellung vorhanden (bei KOP oder FUP). Für die Bearbeitung dieser Parameter wird vom Compiler das Binäresultat BIE verwendet.

Das Binäresultat BIE wird auf den Signalzustand "1" gesetzt, wenn der Baustein fehlerfrei beendet wurde. Liegt ein Fehler vor, wird das Binäresultat BIE auf "0" gesetzt.

**Zuordnung im Datenbereich**

Der FB S\_RCV arbeitet mit einem Instanz-DB I\_RCV zusammen. Die DB-Nummer wird beim Aufruf mitgegeben. Ein Zugriff auf die Daten im Instanz-DB ist nicht zulässig.

Die Tabelle führt die Parameter von FB2 S\_RCV auf.

**Hinweis**

Ausnahme: Im Fehlerfall, STATUS == W#16#1E0D, können Sie einen genaueren Fehlerhinweis aus der Variablen SFCERR entnehmen. Diese Fehlervariable kann nur über einen symbolischen Zugriff auf den Instanz-DB geladen werden.

Tabelle 3- 14 FB2: Parameter S\_RCV

Name	Art	Datentyp	Beschreibung	Zulässige Belegung, Bemerkung
EN_R	INPUT	BOOL	Freigabe für Daten lesen	
R	INPUT	BOOL	Auftragsabbruch	Laufender Auftrag wird abgebrochen. Empfang gesperrt.
LADDR	INPUT	INT	Basisadresse der seriellen Schnittstelle ET 200S	Die Basisadresse wird aus STEP 7 entnommen.
DB_NO	INPUT	INT	Datenbausteinnummer	Empfangs-DB-Nr.: CPU-spezifisch, Null ist nicht zulässig
DBB_NO	INPUT	INT	Datenbytenummer	0 ≤ DBB_NO ≤ 8190 Empfangene Daten per Datenwort
NDR <sup>1</sup>	OUTPUT	BOOL	Auftrag fertig ohne Fehler, Daten übernommen	Parameter STATUS == 16#00
ERROR <sup>1</sup>	OUTPUT	BOOL	Auftrag fertig mit Fehler	Parameter STATUS enthält die Fehlerinformation
LEN <sup>1</sup>	OUTPUT	INT	Länge des empfangenen Telegramms	1 ≤ LEN ≤ 224 Angabe in Anzahl Byte
STATUS <sup>1</sup>	OUTPUT	WORD	Spezifikation des Fehlers	Bei ERROR == 1 steht im Parameter STATUS die Fehlerinformation
COM_RST	IN_OUT	BOOL	Neustart des FB	

<sup>1</sup> Diese Parameter stehen nach korrektem Empfangsauftrag **einen** CPU-Zyklus lang zur Verfügung.

### Zeitdiagramm für FB2 S\_RCV

Das folgende Bild zeigt das Verhalten der Parameter NDR, LEN und ERROR je nachdem, wie die Eingänge EN\_R und R verdrahtet sind.

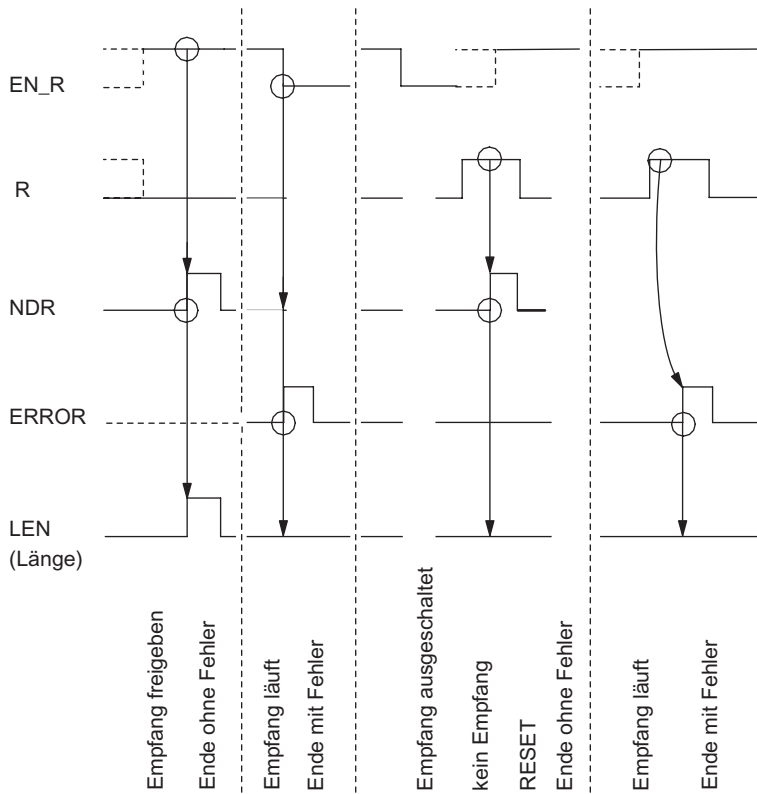


Bild 3-9 Zeitablaufdiagramm FB2 S\_RCV

#### Hinweis

Der Eingang EN\_R ist statisch auf "1" zu legen. Während des gesamten Empfangsauftrages muss der Parameter EN\_R mit dem VKE "1" (Verknüpfungsergebnis) versorgt werden.

### 3.5.3 Konfigurieren und Parametrieren des Modbus-Masters

#### Konfigurieren der Modbus-Baugruppe

Wenn Sie mit einem S7-Master mit der Baugruppe über ein PROFIBUS-Netzwerk kommunizieren, dann arbeiten Sie in der Hardware-Konfiguration in STEP 7, um die Baugruppe im PROFIBUS-Netzwerk einzurichten und die Kommunikationsparameter der Baugruppe einzustellen.

Wenn Sie den Modbus-Master im Hardware-Katalog wählen und sie in die ET 200S Basis in der Konfiguration des Netzwerks einfügen, werden die Bestellnummer der Baugruppe, die Nummer des Steckplatzes sowie die Adressen von Ein- und Ausgängen automatisch in die Konfigurationstabelle aufgenommen. Sie können dann den Eigenschaftsdialog des Modbus-Master aufrufen und dort die Kommunikationsart und andere Parameter einstellen.

#### Parametrieren des Master-Treibers

Die Tabelle führt die Parameter auf, die für den Modbus-Treiber der Baugruppe eingestellt werden können.

Tabelle 3- 15 Parameter für den Modbus-Master-Treiber

Parameter	Beschreibung	Wertebereich	Defaultwert
Diagnosealarm	Geben Sie an, ob die Baugruppe einen Diagnosealarm erzeugt, wenn ein schwerer Fehler auftritt.	<ul style="list-style-type: none"> <li>• nein</li> <li>• Ja</li> </ul>	nein
BREAK Erkennung aktivieren	Bei Leitungsbruch bzw. wenn kein Schnittstellenkabel angeschlossen ist erzeugt die Baugruppe die Fehlermeldung "Break".	<ul style="list-style-type: none"> <li>• Nein</li> <li>• Ja</li> </ul>	Nein
Art der Schnittstelle	Geben Sie die zu verwendende elektrische Schnittstelle an.	<ul style="list-style-type: none"> <li>• RS-232C</li> <li>• RS-422 (voll duplex)</li> <li>• RS-485 (halbduplex)</li> </ul>	RS-232C
Halb- und vollduplexe Vorbelegung der Empfangsleitung	Geben Sie die Vorbelegung der Empfangsleitung in den Betriebsarten RS-422 und RS-485 an. Nicht in der Betriebsart RS-232C.  Die Einstellung "Invertierte Pegel" wird nur im Ersatzteillfall zur Sicherstellung der Kompatibilität benötigt.	RS422: R(A) 5V / R(B) 0V (BREAK) R(A) 0V / R(B) 5V Invertierter Pegel  RS485: Keine R(A) 0V / R (B) 5V	RS422: R(A) 5V / R(B) 0V (BREAK)  RS485: R(A) 0V / R (B) 5V

3.5 Modbus-Master-Treiber

Parameter	Beschreibung	Wertebereich	Defaultwert
Datenflusskontrolle (mit voreingestellten Parametern; voreingestellte Werte im Anwenderprogramm ändern)	Sie können Daten mit Datenflusskontrolle senden und empfangen. Durch die Datenflusskontrolle wird die Datenübertragung synchronisiert, wenn ein Kommunikationspartner schneller arbeitet als der andere. Wählen Sie die Art der Datenflusskontrolle und stellen Sie die zugehörigen Parameter ein.  Hinweis: Bei der RS 485-Schnittstelle ist keine Datenflusskontrolle möglich. Datenflusskontrolle mit "Automatische Bedienung der V24-Signale" ist nur bei der RS-232C-Schnittstelle möglich.	<ul style="list-style-type: none"> <li>keine</li> <li>Automatische Bedienung der V.24-Signale</li> </ul>	keine
Übertragungsgeschwindigkeit	Wählen Sie die Geschwindigkeit der Datenübertragung in Bits pro Sekunde.	<ul style="list-style-type: none"> <li>110</li> <li>300</li> <li>600</li> <li>1.200</li> <li>2.400</li> <li>4.800</li> <li>9.600</li> <li>19.200</li> <li>38.400</li> <li>57.600</li> <li>76.800</li> <li>115.200</li> </ul>	9600
Stoppbits	Wählen Sie die Anzahl Stoppbits, die bei der Übertragung jedem Zeichen nachgesetzt werden und das Ende eines Zeichens kennzeichnen.	<ul style="list-style-type: none"> <li>1</li> <li>2</li> </ul>	1
Parität	Die Reihenfolge der Datenbits kann um ein Zeichen erweitert werden, so dass das Paritätsbit aufgenommen wird. Der zusätzliche Wert (0 oder 1) versetzt den Wert aller Bits (Datenbits und Paritätsbit) in einen definierten Zustand.  <b>Keine:</b> Daten werden ohne Paritätsbit gesendet.  <b>Ungerade:</b> Das Paritätsbit wird gesetzt, so dass die Gesamtzahl der Datenbits (einschließlich Paritätsbit) mit dem Signalzustand "1" ungerade ist.  <b>Gerade:</b> Das Paritätsbit wird gesetzt, so dass die Gesamtzahl der Datenbits (einschließlich Paritätsbit) mit dem Signalzustand "1" gerade ist.	<ul style="list-style-type: none"> <li>keine</li> <li>ungerade</li> <li>gerade</li> </ul>	Gerade
Reaktionszeit	Zulässige Zeit für die Antwort vom Slave.	50 ms bis 655.000 ms	2000 ms
Betriebsart	"Normaler Betrieb" "Störunterdrückung"	<ul style="list-style-type: none"> <li>normal</li> <li>Störunterdrückung</li> </ul>	normal



Parameter	Beschreibung	Wertebereich	Defaultwert
Multiplikator Zeichenverzug	Nutzt einen Multiplikator der Zeichenverzugszeit von 1-10.	1 bis 10	1
Empfangspuffer der seriellen Schnittstelle beim Anlauf löschen	Geben Sie an, ob der Empfangspuffer der seriellen Schnittstelle automatisch gelöscht werden soll, wenn die CPU vom Betriebszustand STOP nach RUN wechselt (CPU-Anlauf). Sie können auf diese Weise sicherstellen, dass der Empfangspuffer der seriellen Schnittstelle nur Telegramme enthält, die nach dem Anlauf der CPU empfangen wurden.	<ul style="list-style-type: none"> <li>• nein</li> <li>• ja</li> </ul>	ja

- **Vollduplex (RS422) Vierdraht-Betrieb**

In dieser Betriebsart werden Daten über die Sendeleitung T(A), T(B) gesendet und über die Empfangsleitung R(A), R(B) empfangen. Die Fehlerbehandlung erfolgt gemäß der über den Parameter "Treiber-Betriebsart" eingestellten Funktionalität (normal oder Störunterdrückung).

- **Halbduplex (RS485) Zweidraht-Betrieb**

In dieser Betriebsart wird vom Treiber ein Umschaltbetrieb der 2-Draht-Empfangsleitung R(A), R(B) der Schnittstelle zwischen Sende- und Empfangsbetrieb durchgeführt. Der Beginn eines Empfangstelegramms vom Slave wird mit der korrekt empfangenen Slave-Adresse erkannt. Bei Punkt-zu-Punkt-Kopplung wird als Vorbelegung der Empfangsleitung die Einstellung R(A) 0V, R(B) 5V empfohlen.

- **Vorbelegung der Empfangsleitung**

Dieser Parameter gibt den Grundzustand der Empfangsleitung für die Betriebsarten RS-422 und RS-485 an. Er wird nicht für die Betriebsart RS-232C verwendet.

- **R(A) 5V, R(B) 0V (BREAK)**

Die Zwei-Draht-Leitung R(A), R(B) wird von der Baugruppe wie folgt vorbelegt:  
R(A) --> +5V, R(B) --> 0V ( $V_A - V_B = +0,3$  V).

Dies bedeutet, dass bei Leitungsbruch BREAK-Pegel an der Baugruppe ansteht.

- **R(A) 0V, R(B) 5V (High)**

Die Zwei-Draht-Leitung R(A), R(B) wird von der Baugruppe wie folgt vorbelegt:  
R(A) --> 0V, R(B) --> +5 V ( $V_A - V_B = -0,3$  V).

Dies bedeutet, dass bei Leitungsbruch (bzw. im Ruhezustand wenn kein Slave sendet) HIGH-Pegel an der Baugruppe ansteht. Der Leitungszustand BREAK kann nicht erkannt werden.

- **Keine (nur bei RS485)**

Bei Mehrpunktverbindung ist eine Vorbelegung der Empfangsleitung ausgeschaltet.

- **Übertragungsgeschwindigkeit**

Die maximale Übertragungsgeschwindigkeit ist die Geschwindigkeit der Datenübertragung in Bit pro Sekunde (bps). Die maximale Übertragungsgeschwindigkeit der Baugruppe beträgt 38400 bps im Halbduplex-Betrieb.

- **Datenbits**

Die Anzahl der Datenbits beschreibt, auf wieviele Bits ein zu übertragendes Zeichen abgebildet wird. Es müssen immer 8 Datenbits eingestellt werden. Es muss immer ein 11-Bit-Zeichenrahmen verwendet werden. Wenn Sie "keine" Parität einstellen, müssen Sie 2 Stoppbits wählen.

- **Stoppbits**

Die Anzahl der Stoppbits definiert den kleinsten zeitlichen Abstand zwischen zwei zu übertragenden Zeichen. Es muss immer ein 11-Bit-Zeichenrahmen verwendet werden. Wenn Sie "keine" Parität einstellen, müssen Sie 2 Stoppbits wählen.

- **Parität**

Das Paritätsbit dient der Datensicherheit. Es ergänzt die Anzahl der übertragenen Datenbits je nach Parametrierung auf eine gerade oder ungerade Anzahl. Ist "keine" Parität eingestellt, wird kein Paritätsbit übertragen. Dies reduziert die Übertragungssicherheit. Es muss immer ein 11-Bit-Zeichenrahmen verwendet werden. Wenn Sie "keine" Parität einstellen, müssen Sie 2 Stoppbits wählen.

- **Antwortzeit**

Die Antwortüberwachungszeit ist die Zeit, die der Master nach der Ausgabe eines Anforderungstelegramms auf ein Antworttelegramm vom Slave wartet.

- **Normalbetrieb**

In dieser Betriebsart führen alle erkannten Übertragungsfehler bzw. BREAK vor und nach Empfangstelegrammen vom Slave zu einer entsprechenden Fehlermeldung.

- **Störunterdrückung**

Wird zu Beginn des Empfangstelegramms BREAK in der Empfangsleitung erkannt oder stellt der Baugruppen-Schnittstellenbaustein Übertragungsfehler fest, so ignoriert der Treiber diesen fehlerhaften Empfang. Der Beginn eines Empfangstelegramms vom Slave wird mit der korrekt empfangenen Slave-Adresse erkannt. Ebenfalls ignoriert werden Übertragungsfehler bzw. BREAK, die nach dem Ende des Empfangstelegramms (CRC-Code) auftreten.

- **Multiplikator Zeichenverzug**

Wenn ein Koppelpartner die zeitlichen Anforderungen der Modbus-Spezifikation nicht einhalten kann, besteht die Möglichkeit, die Zeichenverzugszeit  $t_{zVZ}$  durch den Multiplikationsfaktor  $f_{MUL}$  zu vervielfachen. Die Zeichenverzugszeit sollte nur verstellt werden, wenn der Koppelpartner die geforderten Zeiten nicht einhalten kann. Die resultierende Zeichenverzugszeit  $t_{zVZ}$  ergibt sich aus:

$$t_{zVZ} = t_{zVZ\_TAB} * f_{MUL} ;$$

$t_{zVZ\_TAB}$ : Tabellenwert für  $t_{zVZ}$

$f_{MUL}$ : Multiplikationsfaktor

---

### Hinweis

Beachten Sie auch die Themen zu Identifikationsdaten (Seite 61) und Nachladen von Firmware-Updates (Seite 63) .

---

### 3.5.4 Vom Modbus-Master verwendete Funktionscodes

#### Tabelle der Funktionscodes

Die Tabelle führt die Funktionscodes auf, die vom Modbus-Master-Treiber unterstützt werden.

Tabelle 3- 16 Parameter für den Modbus-Master-Treiber

Funktionscode	Beschreibung	Funktion in SIMATIC S7	
01	Read Output Status	Bitweise lesen	Merker M
		Bitweise lesen	Ausgänge A
		Bitweise lesen (16-Bit-Intervall)	Zeiten T
		Bitweise lesen (16-Bit-Intervall)	Zähler Z
02	Read Input Status	Bitweise lesen	Merker M
		Bitweise lesen	Eingänge E
03	Read Output Registers	Wortweise lesen	Datenbaustein DB
04	Read Input Registers	Wortweise lesen	Datenbaustein DB
05	Force Single Coil	Bitweise schreiben	Merker M
		Bitweise schreiben	Ausgänge A
06	Preset Single Register	Wortweise schreiben	Datenbaustein DB
07	Read Exception Status	Bitweise lesen	8-Bit-Status
08	Loop back diagnostic test	-	-
11	Fetch Communications Event Counter	2 Wörter lesen	Ereignisstatus und Zähler
12	Fetch Communications Event Log	70 Bytes lesen	Ereignisprotokoll
15	Force Multiple Coils	Bitweise schreiben (1...2040 Bit)	Merker M
		Bitweise schreiben (1...2040 Bit)	Ausgänge A
16	Preset Multiple Registers	Wortweise schreiben (1...127 Register)	Datenbaustein DB

### 3.5.5 Funktionscode 01 – Read Output Status

#### Zweck und Aufbau

<b>Funktion</b>	Diese Funktion ermöglicht das Lesen einzelner Bits vom Slave.
<b>Startadresse</b>	Der Parameter Bit-Startadresse wird vom Treiber nicht überprüft und unverändert gesendet.
<b>Bitanzahl</b>	Als Bitanzahl (number of coils) ist jeder Wert zwischen 1 und 1768 zulässig.
<b>LEN in Bytes</b>	6

#### Quell-DB SEND

Die Tabelle zeigt den Aufbau des SEND-Quellbereichs:

Adresse	Name	Art	Anfangswert	Kommentar
+0.0	Adresse	BYTE	B#16#5	Slave-Adresse
+1.0	Funktion	BYTE	B#16#1	Funktionscode
+2.0	Bit-Startadresse	WORD	W#16#0040	Bit-Startadresse
+4.0	Bitanzahl	INT	16	Bitanzahl

#### Ziel-DB RCV

Die Tabelle zeigt den Inhalt des RCV-Zielbereichs:

Adresse	Name	Art	Aktualwert	Kommentar
+0.0	data[1]	WORD	W#16#1701	Daten

Die Daten des Antworttelegramms werden vom Treiber wortweise in den Ziel-DB eingetragen. Das erste empfangene Byte wird als Low-Byte des ersten Wortes "data[1]" eingetragen, das dritte empfangene Byte als Low-Byte des zweiten Wortes "data[2]" usw. Werden weniger als 9 Bit gelesen oder ist nur ein Low-Byte gelesen worden, wird in das verbleibende High-Byte des letzten Wortes der Wert 00H eingetragen.

### 3.5.6 Funktionscode 02 – Read Input Status

#### Zweck und Aufbau

<b>Funktion</b>	Diese Funktion ermöglicht das Lesen einzelner Bits vom Slave.
<b>Startadresse</b>	Der Parameter Bit-Startadresse wird vom Treiber nicht überprüft und unverändert gesendet.
<b>Bitanzahl</b>	Als Bitanzahl (number of coils) ist jeder Wert zwischen 1 und 1768 zulässig.
<b>LEN in Bytes</b>	6

#### Quell-DB SEND

Die Tabelle zeigt den Aufbau des SEND-Quellbereichs:

Adresse	Name	Art	Anfangswert	Kommentar
+0.0	Adresse	BYTE	B#16#5	Slave-Adresse
+1.0	Funktion	BYTE	B#16#2	Funktionscode
+2.0	Bit-Startadresse	WORD	W#16#0120	Bit-Startadresse
+4.0	Bitanzahl	INT	24	Bitanzahl

#### Ziel-DB RCV

Die Tabelle zeigt den Inhalt des RCV-Zielbereichs:

Adresse	Name	Art	Aktualwert	Kommentar
+0.0	data[1]	WORD	W#16#2604	Daten
+2.0	data[2]	WORD	W#16#0048	Daten

Die Daten des Antworttelegramms werden vom Treiber wortweise in den Ziel-DB eingetragen. Das erste empfangene Byte wird als Low-Byte des ersten Wortes "data[1]" eingetragen, das dritte empfangene Byte als Low-Byte des zweiten Wortes "data[2]" usw.

Werden weniger als 9 Bit gelesen oder ist nur ein Low-Byte gelesen worden, wird in das verbleibende High-Byte des letzten Wortes der Wert 00H eingetragen.

### 3.5.7 Funktionscode 03 – Read Output Registers

#### Zweck und Aufbau

<b>Funktion</b>	Diese Funktion ermöglicht das Lesen einzelner Register vom Slave.
<b>Startadresse</b>	Der Parameter Register-Startadresse wird vom Treiber nicht überprüft und unverändert gesendet.
<b>Bitanzahl</b>	Es können maximal 110 Register (1 Register = 2 Byte) gelesen werden.
<b>LEN in Bytes</b>	6

#### Quell-DB SEND

Die Tabelle zeigt den Aufbau des SEND-Quellbereichs:

Adresse	Name	Art	Anfangswert	Kommentar
+0.0	Adresse	BYTE	B#16#5	Slave-Adresse
+1.0	Funktion	BYTE	B#16#3	Funktionscode
+2.0	Register-Start- adresse	WORD	W#16#0040	Register-Start- adresse
+4.0	Registeranzahl	INT	2	Registeranzahl

#### Ziel-DB RCV

Die Tabelle zeigt den Inhalt des RCV-Zielbereichs:

Adresse	Name	Art	Aktualwert	Kommentar
+0.0	data[1]	WORD	W#16#2123	Daten
+2.0	data[2]	WORD	W#16#2527	Daten

### 3.5.8 Funktionscode 04 – Read Input Registers

#### Zweck und Aufbau

<b>Funktion</b>	Diese Funktion ermöglicht das Lesen einzelner Register vom Slave.
<b>Startadresse</b>	Der Parameter Register-Startadresse wird vom Treiber nicht überprüft und unverändert gesendet.
<b>Bitanzahl</b>	Es können maximal 110 Register (1 Register = 2 Byte) gelesen werden.
<b>LEN in Bytes</b>	6

#### Quell-DB SEND

Die Tabelle zeigt den Aufbau des SEND-Quellbereichs:

Adresse	Name	Art	Anfangswert	Kommentar
+0.0	Adresse	BYTE	B#16#5	Slave-Adresse
+1.0	Funktion	BYTE	B#16#4	Funktionscode
+2.0	Register-Start- adresse	WORD	W#16#0050	Register-Start- adresse
+4.0	Registeranzahl	INT	3	Registeranzahl

#### Ziel-DB RCV

Die Tabelle zeigt den Inhalt des RCV-Zielbereichs:

Adresse	Name	Art	Aktualwert	Kommentar
+0.0	data[1]	WORD	W#16#2123	Daten
+2.0	data[2]	WORD	W#16#2527	Daten
+4.0	data[3]	WORD	W#16#3536	Daten

### 3.5.9 Funktionscode 05 – Force Single Coil

#### Zweck und Aufbau

<b>Funktion</b>	Mit dieser Funktion kann beim Slave ein einzelnes Bit gesetzt oder gelöscht werden.
<b>Bitadresse</b>	Der Parameter Bitadresse wird vom Treiber nicht überprüft und unverändert gesendet.
<b>Bitstatus</b>	Als Bitstatus sind folgende zwei Werte zulässig: FF00H => Bit setzen 0000H => Bit löschen
<b>LEN in Bytes</b>	6

#### Quell-DB SEND

Die Tabelle zeigt den Aufbau des SEND-Quellbereichs:

Adresse	Name	Art	Anfangswert	Kommentar
+0.0	Adresse	BYTE	B#16#5	Slave-Adresse
+1.0	Funktion	BYTE	B#16#5	Funktionscode
+2.0	Bitadresse	WORD	W#16#0019	Bitadresse
+4.0	Bitstatus	WORD	W#16#FF00	Bitstatus

Der Slave muss das Anforderungstelegramm unverändert an den Master zurückgeben (Echo).

#### Ziel-DB RCV

Die Tabelle zeigt den Inhalt des RCV-Zielbereichs:

Adresse	Name	Art	Aktualwert	Kommentar
+0.0	Adresse	BYTE	B#16#5	Slave-Adresse
+1.0	Funktion	BYTE	B#16#5	Funktionscode
+2.0	Bitadresse	WORD	W#16#0019	Bitadresse
+4.0	Bitstatus	WORD	W#16#FF00	Bitstatus



### 3.5.10 Funktionscode 06 – Preset Single Register

#### Zweck und Aufbau

<b>Funktion</b>	Mit diesem Befehl kann ein Slave-Register mit einem neuen Wert überschrieben werden.
<b>Registeradresse</b>	Der Parameter Registeradresse wird vom Treiber nicht überprüft und unverändert gesendet.
<b>Registerwert</b>	Als Registerwert kann jeder beliebige Wert verwendet werden.
<b>LEN in Bytes</b>	6

#### Quell-DB SEND

Die Tabelle zeigt den Aufbau des SEND-Quellbereichs:

Adresse	Name	Art	Anfangswert	Kommentar
+0.0	Adresse	BYTE	B#16#5	Slave-Adresse
+1.0	Funktion	BYTE	B#16#6	Funktionscode
+2.0	Registeradresse	WORD	W#16#0180	Registeradresse
+4.0	Registerwert	WORD	W#16#3E7F	Registerwert

#### Ziel-DB RCV

Die Tabelle zeigt den Inhalt des RCV-Zielbereichs:

Adresse	Name	Art	Aktualwert	Kommentar
+0.0	Adresse	BYTE	B#16#5	Slave-Adresse
+1.0	Funktion	BYTE	B#16#6	Funktionscode
+2.0	Registeradresse	WORD	W#16#0180	Registeradresse
+4.0	Registerwert	WORD	W#16#3E7F	Registerwert

### 3.5.11 Funktionscode 07 – Read Exception Status

#### Zweck und Aufbau

**Funktion** Mit diesem Funktionscode können 8 Ereignisbits vom angeschlossenen Slave gelesen werden. Die Anfangsbitnummer des Ereignisbits ist durch das angeschlossene Gerät festgelegt und muss somit nicht vom SIMATIC-Anwenderprogramm vorgegeben werden.

**LEN in Bytes** 2

#### Quell-DB SEND

Die Tabelle zeigt den Aufbau des SEND-Quellbereichs:

Adresse	Name	Art	Anfangswert	Kommentar
+0.0	Adresse	BYTE	B#16#5	Slave-Adresse
+1.0	Funktion	BYTE	B#16#7	Funktionscode

#### Ziel-DB RCV

Die Tabelle zeigt den Inhalt des RCV-Zielbereichs:

Adresse	Name	Art	Aktualwert	Kommentar
+0.0	data[1]	WORD	W#16#3Exx	Daten

Die einzelnen Bits des Antworttelegramms werden vom Treiber im High-Byte im Ziel-DB data[1] eingetragen. Das Low-Byte von data[1] bleibt unverändert. Als Länge am Parameter LEN wird der Wert 1 angezeigt. Die Empfangslänge ist immer 1.

### 3.5.12 Funktionscode 08 – Loop Back Diagnostic Test

#### Zweck und Aufbau

<b>Funktion</b>	Diese Funktion dient zur Überprüfung der Kommunikationsverbindung. Für diesen Funktionscode wird nur der Diagnosecode 0000 unterstützt.
<b>Diagnosecode</b>	Für den Parameter Diagnosecode ist nur der Wert 0000 zulässig.
<b>Testwert</b>	Als Testwert kann jeder beliebige Wert verwendet werden.
<b>LEN in Bytes</b>	6

#### Quell-DB SEND

Die Tabelle zeigt den Aufbau des SEND-Quellbereichs:

Adresse	Name	Art	Anfangswert	Kommentar
+0.0	Adresse	BYTE	B#16#5	Slave-Adresse
+1.0	Funktion	BYTE	B#16#8	Funktionscode
+2.0	Diagnosecode	WORD	B#16#0000	Diagnosecode
+4.0	Registerwert	WORD	B#16#A5C3	Testwert

#### Ziel-DB RCV

Die Tabelle zeigt den Inhalt des RCV-Zielbereichs:

Adresse	Name	Art	Aktualwert	Kommentar
+0.0	Adresse	BYTE	B#16#5	Slave-Adresse
+1.0	Funktion	BYTE	B#16#8	Funktionscode
+2.0	Diagnosecode	WORD	B#16#0000	Diagnosecode
+4.0	Testwert	WORD	B#16#A5C3	Testwert

### 3.5.13 Funktionscode 11 – Fetch Communications Event Counter

#### Zweck und Aufbau

<b>Funktion</b>	Mit diesem Funktionscode kann ein 2-Byte-Statuswort und ein 2-Byte-Ereigniszähler aus dem Slave gelesen werden.
<b>LEN in Bytes</b>	2

#### Quell-DB SEND

Die Tabelle zeigt den Aufbau des SEND-Quellbereichs:

Adresse	Name	Art	Anfangswert	Kommentar
+0.0	Adresse	BYTE	B#16#5	Slave-Adresse
+1.0	Funktion	BYTE	B#16#0B	Funktionscode

#### Ziel-DB RCV

Die Tabelle zeigt den Inhalt des RCV-Zielbereichs:

Adresse	Name	Art	Aktualwert	Kommentar
+0.0	data[1]	WORD	W#16#FEDC	Statuswort
+2.0	data[2]	WORD	W#16#0108	Ereigniszähler

### 3.5.14 Funktionscode 12 – Fetch Communications Event Log

#### Zweck und Aufbau

<b>Funktion</b>	Mit diesem Funktionscode kann folgendes aus dem Slave gelesen werden: -- 2-Byte-Statuswort -- 2-Byte-Ereigniszähler -- 2-Byte-Telegrammzähler -- 64-Byte-Ereignisbytes
<b>LEN in Bytes</b>	2

#### Quell-DB SEND

Die Tabelle zeigt den Aufbau des SEND-Quellbereichs:

Adresse	Name	Art	Anfangswert	Kommentar
+0.0	Adresse	BYTE	B#16#5	Slave-Adresse
+1.0	Funktion	BYTE	B#16#0C	Funktionscode

#### Ziel-DB RCV

Die Tabelle zeigt den Inhalt des RCV-Zielbereichs:

Adresse	Name	Art	Aktualwert	Kommentar
+0.0	data[1]	WORD	W#16#8765	Statuswort
+2.0	data[2]	WORD	W#16#0108	Ereigniszähler
+4.0	data[3]	WORD	W#16#0220	Telegrammzähler
+6.0	bytedata[1]	BYTE	B#16#01	Ereignisbyte 1
+7.0	bytedata[2]	BYTE	B#16#12	Ereignisbyte 2
:	:			:
+68.0	bytedata[63]	BYTE	B#16#C2	Ereignisbyte 63
+69.0	bytedata[64]	BYTE	B#16#D3	Ereignisbyte 64

### 3.5.15 Funktionscode 15 – Force Multiple Coils

#### Zweck und Aufbau

<b>Funktion</b>	Mit diesem Funktionscode können bis zu 1696 Bits im Slave geändert werden.
<b>Startadresse</b>	Der Parameter Bit-Startadresse wird vom Treiber nicht überprüft und unverändert gesendet.
<b>Bitanzahl</b>	Als Bitanzahl (number of coils) ist jeder Wert zwischen 1 und 1696 zulässig. Hiermit wird festgelegt, wie viele Bits im Slave überschrieben werden sollen. Der im Anforderungstelegramm enthaltene Parameter "Bytezähler" wird vom Treiber aufgrund des übergebenen Parameters "Bitanzahl" gebildet.
<b>LEN in Bytes</b>	> 6

#### Quell-DB SEND

Die Tabelle zeigt den Aufbau des SEND-Quellbereichs:

Adresse	Name	Art	Anfangswert	Kommentar
+0.0	Adresse	BYTE	B#16#5	Slave-Adresse
+1.0	Funktion	BYTE	B#16#0F	Funktionscode
+2.0	Bit-Startadresse	WORD	W#16#0058	Bit-Startadresse
+4.0	Bitanzahl	INT	10	Bitanzahl
+6.0	coil_state[1]	WORD	W#16#EFCD	Status Coil 5FH..58H/57H..50H

#### Ziel-DB RCV

Die Tabelle zeigt den Inhalt des RCV-Zielbereichs:

Adresse	Name	Art	Aktualwert	Kommentar
+0.0	Adresse	BYTE	B#16#5	Slave-Adresse
+1.0	Funktion	BYTE	B#16#F	Funktionscode
+2.0	Bitadresse	WORD	W#16#0058	Bitadresse
+4.0	Bitanzahl	INT	10	Bitanzahl

Der Treiber sendet die Daten vom Quell-Ziel-DB wortweise. Das High-Byte (Byte 1) der Wortadresse "EF" im DB wird als erstes gesendet, gefolgt vom Low-Byte (Byte 0) der Wortadresse "CD" des DB. Wird eine ungerade Anzahl Bytes gesendet, dann ist das letzte Byte das High-Byte (Byte 1).

### 3.5.16 Funktionscode 16 – Preset Multiple Registers

#### Zweck und Aufbau

<b>Funktion</b>	Der Funktionscode 16 ermöglicht es, mit einem Anforderungstelegramm bis zu 109 Register im Slave zu überschreiben.
<b>Startadresse</b>	Der Parameter Register-Startadresse wird vom Treiber nicht überprüft und unverändert gesendet.
<b>Registeranzahl</b>	Es können maximal 109 Register (1 Register = 2 Byte) gelesen werden. Der im Anforderungstelegramm enthaltene Parameter "Bytezähler" wird vom Treiber aufgrund des übergebenen Parameters "Registeranzahl" gebildet.
<b>LEN in Bytes</b>	> 6

#### Quell-DB SEND

Die Tabelle zeigt den Aufbau des SEND-Quellbereichs:

Adresse	Name	Art	Anfangswert	Kommentar
+0.0	Adresse	BYTE	B#16#5	Slave-Adresse
+1.0	Funktion	BYTE	B#16#10	Funktionscode
+2.0	Register-Startadresse	WORD	W#16#0060	Register-Startadresse
+4.0	Registeranzahl	INT	3	Registeranzahl
+6.0	reg_data[1]	WORD	W#16#41A1	Registerdaten
+8.0	reg_data[2]	WORD	W#16#42A2	Registerdaten
+10.0	reg_data[3]	WORD	W#16#43A3	Registerdaten

#### Ziel-DB RCV

Die Tabelle zeigt den Inhalt des RCV-Zielbereichs:

Adresse	Name	Art	Aktualwert	Kommentar
+0.0	Adresse	BYTE	B#16#5	Slave-Adresse
+1.0	Funktion	BYTE	B#16#10	Funktionscode
+2.0	Register-Startadresse	WORD	W#16#0060	Register-Startadresse
+4.0	Registeranzahl	INT	3	Registeranzahl

## 3.6 Modbus-Slave-Treiber

### 3.6.1 Bestandteile der Modbus-Slave-Kopplung

#### Einleitung

Dieser Treiber ermöglicht zusammen mit dem entsprechenden Funktionsbaustein den Aufbau einer Kommunikationsverbindung zwischen einem Modbus-Master-Steuerungssystem und der Modbus-Slave-Kommunikationsbaugruppe ET 200S in Form eines modbusfähigen Systems.

#### Prinzip der Datenübertragung

Zur Übertragung wird das Modbus-Protokoll im RTU-Format verwendet. Die Datenübertragung wird nach dem Master-Slave-Prinzip abgewickelt. Der Master wird während der Übertragung initialisiert, so dass die Baugruppe und die S7-CPU als Slaves betrieben werden. Die Funktionscodes 01, 02, 03, 04, 05, 06, 08, 15 und 16 können für die Kommunikation zwischen der Baugruppe und dem Mastersystem verwendet werden. Die Modbus-Adresse im Anforderungstelegramm des Masters wird vom Treiber wie eine S7 ausgewertet. D.h. es können die folgenden Bereiche aus der S7-CPU gelesen werden:

- Lesen und Schreiben von Merkern, Ausgängen, Datenbausteinen
- Lesen von Merkern, Eingängen, Zeiten, Zählern

Mit der vorliegenden Kopplung werden die Datenzugriffe des MODBUS-Protokolls auf die spezifischen Speicherbereiche der SIMATIC S7-CPU umgesetzt.

#### Datenaufbau

Vor der Projektkonfiguration Ihres S7-Datenaufbaus müssen Sie sicherstellen, dass die Daten mit den Anwenderprogrammen des Modbus-Mastersystems kompatibel sind.

#### Modbus-Slave-Kopplung

Die Modbus-Slave-Kopplung für die Baugruppe besteht aus zwei Teilen:

- Modbus-Slave-Treiber
- Modbus-Kommunikations-Funktionsbaustein für die SIMATIC S7-CPU

#### Modbus-Slave-Kommunikations-FB

Für die Modbus-Slave-Kopplung ist neben dem Modbus-Slave-Treiber ein spezieller Kommunikations-FB in der S7-CPU nötig.

Der Modbus-Kommunikations-FB bearbeitet alle für die Kopplung nötigen Funktionen.

Der FB81(S\_MODB) empfängt das Modbus-Protokoll und setzt die Modbus-Adressen in SIMATIC-Speicherbereiche um.

Im Anwenderprogramm ist der FB81 im zyklischen Programm aufzurufen. Der Modbus-Kommunikations-FB nutzt einen Instanz-Datenbaustein als Arbeitsbereich.



### 3.6.2 Datenübertragung beim Modbus-Slave ET 200S

#### Ablauf der Datenübertragung

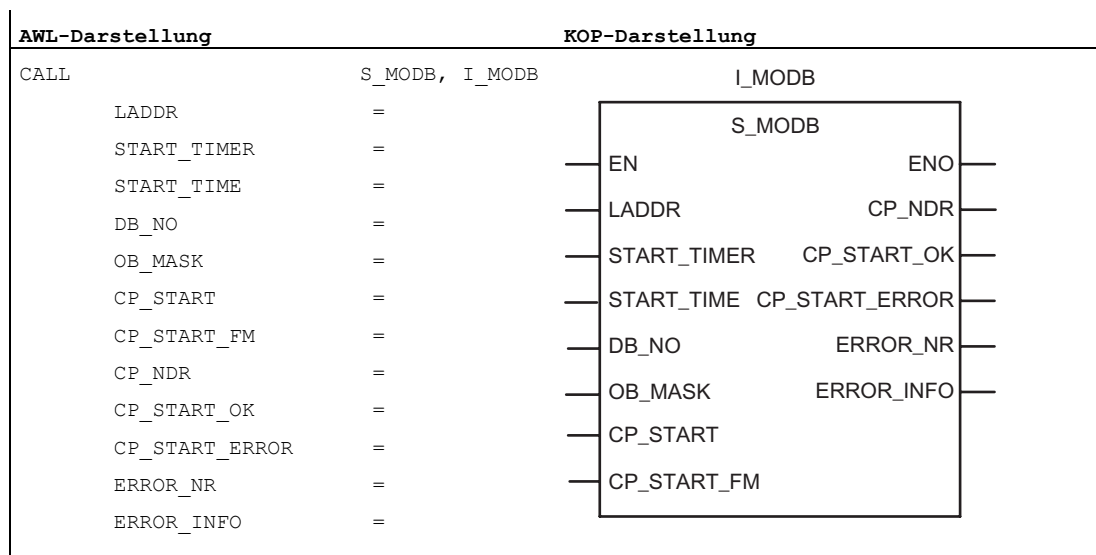
Für die Ausführung eines Modbus-Slave-Auftrags muss der FB S\_MODB zyklisch im Anwenderprogramm aktiviert werden. S\_MODB empfängt den Auftrag von der seriellen Schnittstellenbaugruppe ET200S Modbus/USS, führt den Auftrag aus und gibt die Antwort an die Baugruppe zurück. Die Kommunikation zwischen der CPU und der Baugruppe wird mittels der Funktionsbausteine S\_SEND und S\_RCV durchgeführt, die von S\_MODB aufgerufen werden.

Nach jedem Wiederanlauf der CPU muss das Anwenderprogramm eine Initialisierung des Modbus-Kommunikations-FBs durchführen. Die Initialisierung wird durch eine steigende Flanke am Eingang CP\_START aktiviert. Der FB zeichnet die Größen der Operandenbereiche E, A, M, T und Z der CPU im Instanz-Datenbaustein des FB auf. Nach erfolgreichem Abschluss der Initialisierung setzt der FB den Ausgang CP\_START\_OK.

Ein Initialisierungsfehler wird vom Ausgang CP\_START\_ERROR angegeben. In diesem Fall ist keine Modbus-Kommunikation möglich und alle Aufträge vom Modbus-Master werden mit einer Ausnahmecode-Meldung beantwortet.

S\_MODB nutzt eine Modbus-Datenumsetzungstabelle, die sich im Datenbaustein befindet, um die Modbus-Adressen auf die SIMATIC S7-Speicherbereiche abzubilden.

Mit dem Eingangsparameter OB\_MASK kann der Modbus-FB angewiesen werden, E/A-Zugriffsfehler zu maskieren. Bei einem Schreibzugriff auf nicht vorhandene Peripherie geht die CPU nicht in den Betriebszustand STOP und ruft keinen Fehler-OB auf. Der Zugriffsfehler wird vom FB erkannt und die Funktion wird mit einer Fehlerantwort an den Modbus-Master beendet.



---

**Hinweis**

Die Parameter EN und ENO sind nur in der grafischen Darstellung vorhanden (bei KOP oder FUP). Für die Bearbeitung dieser Parameter wird vom Compiler das Binärergebnis BIE verwendet.

Das Binärergebnis BIE wird auf Signalzustand "1" gesetzt, wenn der Baustein fehlerfrei beendet wurde. Liegt ein Fehler vor, wird das Binärergebnis BIE auf "0" gesetzt.

---

### 3.6.3 Datenbereiche in der SIMATIC CPU

#### Modbus-Datenumsetzungstabelle

Die in Telegrammen enthaltenen Modbus-Adressen werden von FB81(S\_MODB) "S7-mäßig" interpretiert und in den SIMATIC-Speicherbereich umgesetzt. Der Zugriff auf die einzelnen SIMATIC-Speicherbereiche kann vom Anwender mittels Übergabe eines DB als Eingabe für FB81(S\_MODB) angegeben werden (siehe Tabelle).

Tabelle 3- 17 Umsetzungstabelle

Adresse	Name	Art	Anfangswert	Aktualwert	Kommentar	Anwendbarer Funktionscode
0.0	aaaaa	WORD	W#16#0	W#16#0	Modbus-Adressbeginn	01, 05, 15
2.0	bbbbbb	WORD	W#16#0	W#16#7F7	Modbus-Adressende	
4.0	uuuuu	WORD	W#16#0	W#16#1F4	Merker	
6.0	cccc	WORD	W#16#0	W#16#7F8	Modbus-Adressbeginn	01, 05, 15
8.0	dddd	WORD	W#16#0	W#16#FEF	Modbus-Adressende	
10.0	oooo	WORD	W#16#0	W#16#15	Ausgänge	01, 05, 15
12.0	eeee	WORD	W#16#0	W#16#FF0	Modbus-Adressbeginn	
14.0	ffff	WORD	W#16#0	W#16#17E7	Modbus-Adressende	
16.0	tttt	WORD	W#16#0	W#16#28	Zeiten	
18.0	ggggg	WORD	W#16#0	W#16#17E8	Modbus-Adressbeginn	01, 05, 15
20.0	hhhhh	WORD	W#16#0	W#16#1FDF	Modbus-Adressende	
22.0	zzzzz	WORD	W#16#0	W#16#28	Zähler	
24.0	kkkkk	WORD	W#16#0	W#16#1FE0	Modbus-Adressbeginn	02
26.0	lllll	WORD	W#16#0	W#16#27D7	Modbus-Adressende	02
28.0	vvvvv	WORD	W#16#0	W#16#320	Merker	02
30.0	nnnnn	WORD	W#16#0	W#16#27D8	Modbus-Adressbeginn	02
32.0	rrrrr	WORD	W#16#0	W#16#2FCF	Modbus-Adressende	02
34.0	sssss	WORD	W#16#0	W#16#11	Eingänge	02
36.0	DB_Number_FC_03_06_16	WORD	W#16#0	W#16#6	DB	03, 06, 15
38.0	DB_Number_FC_04	WORD	W#16#0	W#16#2	DB	04
40.0	DB_Min	WORD	W#16#0	W#16#1	kleinste verwendete DB-Nummer	Grenzen
42.0	DB_Max	WORD	W#16#0	W#16#6	größte verwendete DB-Nummer	Grenzen
44.0	M_Min	WORD	W#16#0	W#16#1F4	kleinster verwendeter Merker	Grenzen
46.0	M_Max	WORD	W#16#0	W#16#4B0	größter verwendeter Merker	Grenzen
48.0	Q_Min	WORD	W#16#0	W#16#0	kleinster verwendeter Ausgang	Grenzen
50.0	Q_Max	WORD	W#16#0	W#16#64	größter verwendeter Ausgang	Grenzen

### 3.6.4 Konfigurieren der Parameter für die Kopplung

#### Parameter der Hardware-Konfiguration

Die folgenden Parameter und Betriebszustände müssen in der Hardware-Konfiguration für den Treiber eingestellt werden.

- Übertragungsgeschwindigkeit, Parität
- Slave-Adresse der Baugruppe
- Betriebszustand (Normal, Störunterdrückung)
- Multiplikationsfaktor für die Zeichenverzugszeit

#### Parameter am Eingangs-DB für FB81

Die unten aufgeführten Parameter müssen mit dem Eingangs-DB für FB81(S\_MODB) eingestellt werden.

- Adressbereiche für Funktionscodes 01, 05, 15
- Adressbereiche für Funktionscode 02
- Basis-DB-Nummer für Funktionscodes 03, 06, 16
- Basis-DB-Nummer für Funktionscode 04
- Grenzen für Schreibzugriffe

#### Parametrieren des Slave-Treibers

Die Tabelle führt die Parameter auf, die für den Modbus-Treiber der Baugruppe eingestellt werden können.

Tabelle 3- 18 Parameter für den Modbus-Slave-Treiber

Parameter	Beschreibung	Wertebereich	Defaultwert
Diagnosealarm	Geben Sie an, ob die Baugruppe einen Diagnosealarm erzeugt, wenn ein schwerer Fehler auftritt.	<ul style="list-style-type: none"> <li>• nein</li> <li>• ja</li> </ul>	nein
BREAK Erkennung aktivieren	Bei Leitungsbruch bzw. wenn kein Schnittstellenkabel angeschlossen ist erzeugt die Baugruppe die Fehlermeldung "Break".	<ul style="list-style-type: none"> <li>• Nein</li> <li>• Ja</li> </ul>	Nein
Art der Schnittstelle	Geben Sie die zu verwendende elektrische Schnittstelle an.	<ul style="list-style-type: none"> <li>• RS-232C</li> <li>• RS-422 (voll duplex)</li> <li>• RS-485 (halbduplex)</li> </ul>	RS-232C

Parameter	Beschreibung	Wertebereich	Defaultwert
Halb- und vollduplexe Vorbelegung der Empfangsleitung	Geben Sie die Vorbelegung der Empfangsleitung in den Betriebsarten RS-422 und RS-485 an. Nicht in der Betriebsart RS-232C.  Die Einstellung "Invertierte Pegel" wird nur im Ersatzteillfall zur Sicherstellung der Kompatibilität benötigt.	RS422: R(A) 5V / R(B) 0V (BREAK) R(A) 0V / R(B) 5V Invertierter Pegel  RS485: Keine R(A) 0V / R(B) 5V	RS422: R(A) 5V / R(B) 0V (BREAK)  RS485: R(A) 0V / R(B) 5V
Datenflusskontrolle (mit voreingestellten Parametern; voreingestellte Werte im Anwenderprogramm ändern)	Sie können Daten mit Datenflusskontrolle senden und empfangen. Durch die Datenflusskontrolle wird die Datenübertragung synchronisiert, wenn ein Kommunikationspartner schneller arbeitet als der andere. Wählen Sie die Art der Datenflusskontrolle und stellen Sie die zugehörigen Parameter ein.  Hinweis: Bei der RS 485-Schnittstelle ist keine Datenflusskontrolle möglich. Datenflusskontrolle mit "Automatische Bedienung der V24-Signale" ist nur bei der RS-232C-Schnittstelle möglich.	<ul style="list-style-type: none"> <li>• keine</li> <li>• Automatische Bedienung der V.24-Signale</li> </ul>	keine
Übertragungsgeschwindigkeit	Wählen Sie die Geschwindigkeit der Datenübertragung in Bits pro Sekunde.	<ul style="list-style-type: none"> <li>• 110</li> <li>• 300</li> <li>• 600</li> <li>• 1.200</li> <li>• 2.400</li> <li>• 4.800</li> <li>• 9.600</li> <li>• 19.200</li> <li>• 38.400</li> <li>• 57.600</li> <li>• 76.800</li> <li>• 115.200</li> </ul>	9600
Stoppbits	Wählen Sie die Anzahl Stoppbits, die bei der Übertragung jedem Zeichen nachgesetzt werden und das Ende eines Zeichens kennzeichnen.	<ul style="list-style-type: none"> <li>• 1</li> <li>• 2</li> </ul>	1

Parameter	Beschreibung	Wertebereich	Defaultwert
Parität	Die Reihenfolge der Datenbits kann um ein Zeichen erweitert werden, so dass das Paritätsbit aufgenommen wird. Der zusätzliche Wert (0 oder 1) versetzt den Wert aller Bits (Datenbits und Paritätsbit) in einen definierten Zustand. <b>Keine:</b> Daten werden ohne Paritätsbit gesendet. <b>Ungerade:</b> Das Paritätsbit wird gesetzt, so dass die Gesamtzahl der Datenbits (einschließlich Paritätsbit) mit dem Signalzustand "1" ungerade ist. <b>Gerade:</b> Das Paritätsbit wird gesetzt, so dass die Gesamtzahl der Datenbits (einschließlich Paritätsbit) mit dem Signalzustand "1" gerade ist.	<ul style="list-style-type: none"> <li>• keine</li> <li>• ungerade</li> <li>• gerade</li> </ul>	gerade
Slave-Adresse	Eigene Slave-Adresse der Baugruppe	1-247	222
Betriebsart	<ul style="list-style-type: none"> <li>• Normalbetrieb</li> <li>• Störunterdrückung</li> </ul>	<ul style="list-style-type: none"> <li>• normal</li> <li>• Störunterdrückung</li> </ul>	normal
Multiplikator Zeichenverzug	Nutzt einen Multiplikator der Zeichenverzugszeit von 1-10.	1 bis 10	1
Empfangspuffer der seriellen Schnittstelle beim Anlauf löschen	Geben Sie an, ob der Empfangspuffer der seriellen Schnittstelle automatisch gelöscht werden soll, wenn die CPU vom Betriebszustand STOP nach RUN wechselt (CPU-Anlauf). Sie können auf diese Weise sicherstellen, dass der Empfangspuffer der seriellen Schnittstelle nur Telegramme enthält, die nach dem Anlauf der CPU empfangen wurden.	<ul style="list-style-type: none"> <li>• nein</li> <li>• ja</li> </ul>	ja
<sup>1</sup> Die kürzeste Zeichenverzugszeit richtet sich nach der Baudrate.			

In der folgenden Liste finden Sie Erläuterungen zu einigen Parameter oder Werten:

• **Vollduplex (RS422) Vierdraht-Betrieb**

In dieser Betriebsart werden Daten über die Sendeleitung T(A), T(B) gesendet und über die Empfangsleitung R(A), R(B) empfangen. Die Fehlerbehandlung erfolgt gemäß der über den Parameter "Treiber-Betriebsart" eingestellten Funktionalität (normal oder Störunterdrückung).

• **Halbduplex (RS485) Zweidraht-Betrieb**

In dieser Betriebsart wird vom Treiber ein Umschaltbetrieb der 2-Draht-Empfangsleitung R(A), R(B) der Schnittstelle zwischen Sende- und Empfangsbetrieb durchgeführt. Der Beginn eines Empfangstelegramms vom Slave wird mit der korrekt empfangenen Slave-Adresse erkannt. Bei Punkt-zu-Punkt-Kopplung wird als Vorbelegung der Empfangsleitung die Einstellung R (A) 0V, R(B) 5V empfohlen.

- **Vorbelegung der Empfangsleitung**

Dieser Parameter gibt den Grundzustand der Empfangsleitung für die Betriebsarten RS-422 und RS-485 an. Er wird nicht für die Betriebsart RS-232C verwendet.

- **R(A) 5V, R(B) 0V (BREAK)**

Die Zwei-Draht-Leitung R(A), R(B) wird von der Baugruppe wie folgt vorbelegt:

R(A) --> +5V, R(B) --> 0V ( $V_A - V_B \geq +0,3$  V).

Dies bedeutet, dass bei Leitungsbruch BREAK-Pegel an der Baugruppe ansteht.

- **R(A) 0V, R(B) 5V (High)**

Die Zwei-Draht-Leitung R(A),R(B) wird von der Baugruppe wie folgt vorbelegt:

R(A) --> 0V, R(B) --> +5 V ( $V_A - V_B \leq -0,3$  V).

Dies bedeutet, dass bei Leitungsbruch (bzw. im Ruhezustand wenn kein Slave sendet) HIGH-Pegel an der Baugruppe ansteht. Der Leitungszustand BREAK kann nicht erkannt werden.

- **Keine** (nur bei RS485)

Bei Mehrpunktverbindung ist eine Vorbelegung der Empfangsleitung ausgeschaltet.

- **Übertragungsgeschwindigkeit**

Die Übertragungsgeschwindigkeit ist die Geschwindigkeit der Datenübertragung in Bit pro Sekunde (bps). Die Übertragungsgeschwindigkeit der Baugruppe beträgt 38400 bps im Halbduplex-Betrieb.

- **Datenbits**

Die Anzahl der Datenbits beschreibt, auf wieviele Bits ein zu übertragendes Zeichen abgebildet wird. Für diesen Treiber müssen immer 8 Datenbits eingestellt werden. Es muss immer ein 11-Bit-Zeichenrahmen verwendet werden. Wenn Sie "keine" Parität einstellen, müssen Sie 2 Stoppbits wählen.

- **Stoppbits**

Die Anzahl der Stoppbits definiert den kleinsten zeitlichen Abstand zwischen zwei zu übertragenden Zeichen. Es muss immer ein 11-Bit-Zeichenrahmen verwendet werden. Wenn Sie "keine" Parität einstellen, müssen Sie 2 Stoppbits wählen.

- **Parität**

Das Paritätsbit dient der Datensicherheit. Es ergänzt die Anzahl der übertragenen Datenbits, je nach Parametrierung, auf eine gerade oder ungerade Anzahl. Ist "keine" Parität eingestellt, wird kein Paritätsbit übertragen. Dies reduziert die Übertragungssicherheit. Es muss immer ein 11-Bit-Zeichenrahmen verwendet werden. Wenn Sie "keine" Parität einstellen, müssen Sie 2 Stoppbits wählen.

- **Slave-Adresse**

Hier wird die eigene Modbus-Slave-Adresse angegeben, auf die die Baugruppe antworten soll. Die Baugruppe beantwortet nur Telegramme, bei denen die empfangene Slave-Adresse mit der eigenen parametrisierten Slave-Adresse identisch ist. Telegramme an andere Slaves werden nicht geprüft und nicht beantwortet.

- **Normalbetrieb**

In dieser Betriebsart führen alle erkannten Übertragungsfehler bzw. BREAK vor und nach Empfangstelegrammen vom Slave zu einer entsprechenden Fehlermeldung.

- **Störunterdrückung**

Wird zu Beginn des Empfangstelegramms BREAK in der Empfangsleitung erkannt, oder stellt der Baugruppen-Schnittstellenbaustein Übertragungsfehler fest, so ignoriert der Treiber diesen fehlerhaften Empfang. Der Beginn eines Empfangstelegramms vom Slave wird mit der korrekt empfangenen Slave-Adresse erkannt. Ebenfalls ignoriert werden Übertragungsfehler bzw. BREAK, die nach dem Ende des Empfangstelegramms (CRC-Code) auftreten.

- **Multiplikator Zeichenverzug**

Wenn ein Koppelpartner die zeitlichen Anforderungen der Modbus-Spezifikation nicht einhalten kann, besteht die Möglichkeit, die Zeichenverzugszeit  $t_{ZVZ}$  durch den Multiplikationsfaktor  $f_{MUL}$  zu vervielfachen. Die Zeichenverzugszeit sollte nur verstellt werden, wenn der Koppelpartner die geforderten Zeiten nicht einhalten kann.

Die resultierende Zeichenverzugszeit  $t_{ZVZ}$  ergibt sich aus:

$$t_{ZVZ} = t_{ZVZ\_TAB} * f_{MUL}$$

$t_{ZVZ\_TAB}$  = Tabellenwert für  $t_{ZVZ}$

$f_{MUL}$  = Multiplikationsfaktor

---

**Hinweis**

Beachten Sie auch die Themen zu Identifikationsdaten (Seite 61) und Nachladen von Firmware-Updates (Seite 63).

---



### 3.6.5 Slave-Funktionscodes

#### Funktionscodes des Modbus-Slave-Treibers

Der Modbus-Slave-Treiber unterstützt die in der Tabelle aufgeführten Funktionscodes.

##### Hinweis

Alle in der folgenden Tabelle aufgeführten Modbus-Adressen beziehen sich auf die Übertragungstelebene und nicht auf die Anwenderschicht im Modbus-Mastersystem. Das heißt, dass die Modbus-Adressen in Übertragungstelegrammen bei 0000 Hex beginnen.

Tabelle 3- 19 Slave-Funktionscodes

Funktionscode	Beschreibung	Funktion in SIMATIC S7	
01	Read Coil Status	Bitweise lesen	Merker M
		Bitweise lesen	Ausgänge A
		Bitweise lesen (16-Bit-Intervall)	Zeiten T
		Bitweise lesen (16-Bit-Intervall)	Zähler Z
02	Read Input Status	Bitweise lesen	Merker M
		Bitweise lesen	Eingänge E
03	Read Holding Registers	Wortweise lesen	Datenbaustein DB
04	Read Input Registers	Wortweise lesen	Datenbaustein DB
05	Force Single Coil	Bitweise schreiben	Merker M
		Bitweise schreiben	Ausgänge A
06	Preset Single Register	Wortweise schreiben	Datenbaustein DB
08	Loop Back Test	-	-
15	Force Multiple Coils	Bitweise schreiben (1...2040 Bit)	Merker M
		Bitweise schreiben (1...2040 Bit)	Ausgänge A
16	Preset multiple (holding) registers	Wortweise schreiben (1...127 Register)	Datenbaustein DB

### 3.6.6 Funktionscode 01 – Read Coil (Output) Status

#### Zweck und Aufbau

Der Funktionscode 01 – Read Coil (Poutput) Status ist folgendermaßen charakterisiert:

<b>Funktion</b>	Diese Funktion ermöglicht das Lesen einzelner Bits aus den nachfolgend aufgeführten SIMATIC-Speicherbereichen durch das Modbus-Mastersystem.				
<b>Anforderungstelegramm</b>	ADDR	FUNC	start_address	bit_number	CRC
<b>Antworttelegramm</b>	ADDR	FUNC	start_address	n Byte DATA	CRC
<b>LEN in Bytes</b>	6				

#### start\_address

Die Modbus-Bitadresse "start\_address" wird vom Treiber interpretiert. Beispiel: FB81(S\_MODB) prüft, ob sich "start\_address" in einem der Bereiche befindet, der vom Umsetzungs-DB für FC 01, 05, 15 (von/bis: Merker, Ausgänge, Zeiten, Zähler) angegeben wurde.

Befindet sich die Modbus-Bitadresse start_address im Bereich	erfolgt der Zugriff auf den SIMATIC-Speicherbereich	
Von <i>aaaaa</i> bis <i>bbbb</i>	ab Merker	M <i>uuuuu.0</i>
Von <i>cccc</i> bis <i>dddd</i>	ab Ausgang	A <i>oooo.0</i>
Von <i>eeee</i> bis <i>ffff</i>	ab Zeit	T <i>tttt</i>
Von <i>gggg</i> bis <i>hhhh</i>	ab Zähler	Z <i>zzzz</i>

Die Adressberechnung für den Zugriff (Adressumsetzung) erfolgt dann folgendermaßen:

Zugriff beginnend mit SIMATIC	Umsetzungsformel	
Merkerbyte	$=((start\_address - aaaaa) / 8)$	+ <i>uuuuu</i>
Ausgangsbyte	$=((start\_address - cccc) / 8)$	+ <i>oooo</i>
Zeit	$=((start\_address - eeee) / 16)$	+ <i>tttt</i>
Zähler	$=((start\_address - gggg) / 16)$	+ <i>zzzz</i>

#### Zugriff auf Merker und Ausgänge

Beim Zugriff auf die SIMATIC-Bereiche Merker und Ausgänge wird die verbleibende Rest-Bit\_Nummer berechnet und dazu verwendet, das entsprechende Bit innerhalb des ersten/letzten Merker- oder Ausgangsbytes zu adressieren.

## Zugriff auf Zeiten und Zähler

Bei der Adressberechnung muss das Ergebnis

- (start\_address – eeeee) oder
- (start\_address – ggggg)

ohne Rest durch 16 teilbar sein (nur wortweiser Zugriff beginnend an Wortgrenze).

## bit\_number

Als bit\_number (Number of coils) sind Werte zwischen 1 und 1768 zulässig. Diese Anzahl von Bits wird gelesen.

Beim Zugriff auf die SIMATIC-Bereiche Zeiten und Zähler muss "bit\_number" durch 16 teilbar sei (nur wortweiser Zugriff).

## Anwendungsbeispiel

Tabelle 3- 20 Beispiel für die Umsetzung der Modbus-Adressierung:

Umsetzung der Modbus-Adressierung für Funktionscodes FC 01, 05, 15	
Modbus-Adresse im Übertragungstelegramm	SIMATIC-Speicherbereich
Von 0 bis 2047	ab Merker M 1000.0
Von 2048 bis 2559	ab Ausgang A 256.0
Von 4096 bis 4607	ab Zeit T 100
Von 4608 bis 5119	ab Zähler Z 200

## Quell-DB SEND

Die Tabelle zeigt den Aufbau des SEND-Quellbereichs:

Adresse	Name	Art	Anfangswert	Kommentar
+0.0	Adresse	BYTE	B#16#5	Slave-Adresse
+1.0	Funktion	BYTE	B#16#1	Funktionscode
+2.0	Bit-Startadresse	WORD	W#16#0040	Bit-Startadresse
+4.0	Bitanzahl	INT	16	Bitanzahl

**Ziel-DB RCV**

Die Tabelle zeigt den Inhalt des RCV-Zielbereichs:

Adresse	Name	Art	Aktualwert	Kommentar
+0.0	data[1]	WORD	W#16#1701	Daten

Die Daten des Antworttelegramms werden vom Treiber wortweise in den Ziel-DB eingetragen. Das erste empfangene Byte wird als Low-Byte des ersten Wortes "data[1]" eingetragen, das dritte empfangene Byte als Low-Byte des zweiten Wortes "data[2]" usw. Werden weniger als 9 Bit gelesen oder ist nur ein Low-Byte gelesen worden, wird in das verbleibende High-Byte des letzten Wortes der Wert 00H eingetragen.

**Adressberechnung:**

Die Modbus-Adresse "start\_address" 0040 Hex (64 dezimal) befindet sich im Bereich Merker:

$$\begin{aligned} \text{Merkerbyte} &= ((\text{start\_address} - \text{aaaaa}) / 8) + \text{uuuuu} \\ &= ((64 - 0) / 8) + 1000 \\ &= 1008; \end{aligned}$$

Die verbleibende Rest-Bit\_Nummer ergibt:

$$\begin{aligned} \text{Rest-Bit\_Nr.} &= ((\text{start\_address} - \text{aaaaa}) \% 8) \quad [\text{Modulo } 8] \\ &= ((64 - 0) \% 8) \\ &= 0; \end{aligned}$$

Es erfolgt ein Zugriff beginnend mit dem Merker M 1008.0 bis einschließlich M 1011.7.

**Bitanzahl:**

Die Modbus-Bitanzahl "bit\_number" 0020 Hex (32 dezimal) bedeutet, dass 32 Bit = 4 Byte gelesen werden sollen.

Die Tabelle führt weitere Beispiele für den Datenzugriff auf.

Tabelle 3- 21 Weitere Beispiele für den Datenzugriff

start_address Hex	Dezimal	Adressberechnung	Adresse
0000	0	Merker $((0 - 0) / 8) + 1000$	-> M1000.0
0021	33	Merker $((33 - 0) / 8) + 1000$	-> M1004.1
0400	1024	Merker $((1024 - 0) / 8) + 1000$	-> M1128.0
0606	1542	Merker $((1542 - 0) / 8) + 1000$	-> M1192.6
0840	2112	Ausgang $((2112 - 2048) / 8) + 256$	-> A264.0
09E4	2532	Ausgang $((2532 - 2048) / 8) + 256$	-> A316.4
1010	4112	Zeiten $((4112 - 4096) / 16) + 100$	-> T 101
10C0	4288	Zeiten $((4288 - 4096) / 16) + 100$	-> T112
1200	4608	Zähler $((4608 - 4608) / 16) + 200$	-> Z 200
13E0	5088	Zähler $((5088 - 4608) / 16) + 200$	-> Z 230

### 3.6.7 Funktionscode 02 – Read Input Status

#### Zweck und Aufbau

Der Funktionscode 02 – Read Input Status ist folgendermaßen charakterisiert:

<b>Funktion</b>	Diese Funktion ermöglicht das Lesen einzelner Bits aus den nachfolgend aufgeführten SIMATIC-Speicherbereichen durch das Modbus-Mastersystem.				
<b>Anforderungstelegramm</b>	ADDR	FUNC	start_address	bit_number	CRC
<b>Antworttelegramm</b>	ADDR	FUNC	Byte_count	n n Byte DATA	CRC
<b>LEN in Bytes</b>	6				

#### start\_address

Die Modbus-Bitadresse "start\_address" wird vom Treiber wie folgt interpretiert:

Der Treiber prüft, ob sich "start\_address" innerhalb einer der Bereiche befindet, die in den Umsetzungs-DB für FC 02 (von / bis: Merker, Eingänge) eingegeben wurden.

Befindet sich die Modbus-Bitadresse start_address im Bereich	erfolgt der Zugriff auf den SIMATIC-Speicherbereich	
Von kkkkk bis llll	ab Merker	M vvvvv.0
Von nnnnn bis rrrr	ab Eingang	E sssss. 0

Die Adressberechnung für den Zugriff (Adressumsetzung) erfolgt dann folgendermaßen:

Zugriff beginnend mit SIMATIC	Umsetzungsformel	
Merkerbyte	$= ((\text{start\_address} - \text{kkkk}) / 8)$	+ vvvvv
Eingangsbyte	$= ((\text{start\_address} - \text{nnnn}) / 8)$	+ sssss

#### Zugriff auf Merker und Eingänge

Beim Zugriff auf die SIMATIC-Bereiche Merker und Eingänge wird die verbleibende Rest-Bit\_Nummer berechnet und dazu verwendet, das entsprechende Bit innerhalb des ersten/letzten Merker- oder Eingangsbytes zu adressieren.

#### bit\_number

Als bit\_number (Number of coils) ist jeder Wert zwischen 1 und 1768 zulässig. Diese Anzahl von Bits wird gelesen.

**Anwendungsbeispiel**

Beispiel für die Umsetzung der Modbus-Adresszuordnung:

Tabelle 3- 22 Umsetzung der Modbus-Adressierung für Funktionscode FC 02

Modbus-Adresse im Übertragungstelegramm	SIMATIC-Speicherbereich	
Von 0 bis 4095	ab Merker	M 2000.0
Von 4096 bis 5119	ab Eingang 0	I 128.0

**Quell-DB SEND**

Die Tabelle zeigt den Aufbau des SEND-Quellbereichs:

Adresse	Name	Art	Anfangswert	Kommentar
+0.0	Adresse	BYTE	B#16#5	Slave-Adresse
+1.0	Funktion	BYTE	B#16#2	Funktionscode
+2.0	Bit-Startadresse	WORD	W#16#0120	Bit-Startadresse
+4.0	Bitanzahl	INT	24	Bitanzahl

**Ziel-DB RCV**

Die Tabelle zeigt den Inhalt des RCV-Zielbereichs:

Adresse	Name	Art	Aktualwert	Kommentar
+0.0	Data[1]	WORD	W#16#2604	Daten
+2.0	Data[2]	WORD	W#16#0048	Daten

Die Daten des Antworttelegramms werden vom Treiber wortweise in den Ziel-DB eingetragen. Das erste empfangene Byte wird als Low-Byte des ersten Wortes "data[1]" eingetragen, das dritte empfangene Byte als Low-Byte des zweiten Wortes "data[2]" usw.

Werden weniger als 9 Bit gelesen oder ist nur ein Low-Byte gelesen worden, wird in das verbleibende High-Byte des letzten Wortes der Wert 00H eingetragen.

**Adressberechnung:**

Die Modbus-Adresse "start\_address" 1030 Hex (4144 dezimal) befindet sich im Bereich Eingänge:

$$\begin{aligned}
 \text{Eingangbyte} &= ((\text{start\_address} - \text{nnnn}) / 8) + \text{ssss} \\
 &= ((4144 - 4096) / 8) + 128 \\
 &= 134;
 \end{aligned}$$

Die verbleibende Rest-Bit\_Nummer ergibt:

$$\begin{aligned} \text{Rest-Bit\_Nr.} &= ((\text{start\_address} - \text{aaaa}) \% 8) \quad [\text{Modulo } 8] \\ &= ((4144 - 4096) \% 8) \\ &= 0; \end{aligned}$$

Es erfolgt ein Zugriff beginnend mit dem Eingang E 134.0 bis einschließlich E 136.7.

**Bitanzahl:**

Die Modbus-Bitanzahl "bit\_number" 0018 Hex (24 dezimal) bedeutet, dass 24 Bit = 3 Byte gelesen werden sollen.

Die Tabelle führt weitere Beispiele für den Datenzugriff auf.

Tabelle 3- 23 Weitere Beispiele für den Datenzugriff

start_address Hex Dezimal		Adressberechnung	Adresse
0000	0	Merker $((0 - 0) / 8) + 2000$	-> M2000.0
0071	113	Merker $((113 - 0) / 8) + 2000$	-> M2014.1
0800	2048	Merker $((2048 - 0) / 8) + 2000$	-> M2256.0
0D05	3333	Merker $((3333 - 0) / 8) + 2000$	-> M2416.5
1000	4096	Eingang $((4096 - 4096) / 8) + 128$	-> E 128.0
10A4	4260	Eingang $((4260 - 4096) / 8) + 128$	-> E 148.4

### 3.6.8 Funktionscode 03 – Read Output Registers

#### Zweck und Aufbau

Der Funktionscode 03 – Read Output Registers ist folgendermaßen charakterisiert:

<b>Funktion</b>	Diese Funktion ermöglicht das Lesen von Datenworten aus einem Datenbaustein durch das Modbus-Mastersystem				
<b>Anforderungstelegramm</b>	ADDR	FUNC	start_address	register_number	CRC
<b>Antworttelegramm</b>	ADDR	FUNC	Byte_count n	n/2-Register DATA	CRC (High, Low)
<b>LEN in Bytes</b>	6				

#### start\_address

Die Modbus-Registeradresse "start\_register" wird vom Treiber wie folgt interpretiert:

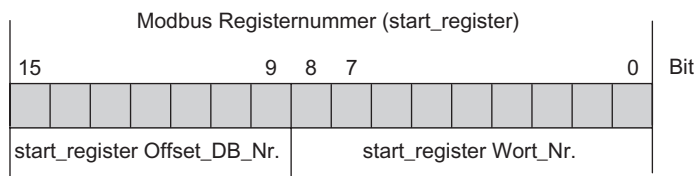


Bild 3-10 Interpretation der Modbus-Registernummer

Zur weiteren Adressbildung verwendet der FB81(S\_MODB) die Basis-DB-Nummer (aus DB xxxxx), die im Umsetzungs-DB für FC 03, 06, 16 eingegeben ist.

Die Adressberechnung für den Zugriff (Adressumsetzung) erfolgt dann in zwei Schritten:

Zugriff auf SIMATIC	Umsetzungsformel
Datenbaustein-DB (resultierender DB)	$=(\text{Basis-DB-Nummer } xxxxx + \text{start\_register Offset\_DB\_Nr.})$
Datenwort DBW	$=(\text{start\_register Wort\_Nr. } * 2)$

#### Berechnungs-Formel für start\_register

Wenn der resultierende DB bekannt ist, der gelesen werden soll, kann die im Mastersystem benötigte Modbus-Adresse start\_register nach folgender Formel berechnet werden:

$$\text{start\_register} = ((\text{resultierender DB} - \text{Basis-DB-Nummer}) * 512) + (\text{Datenwort\_DBW} / 2)$$

Hierbei dürfen nur geradzahlige Datenwortnummern zugrunde gelegt werden.

#### register\_number

Als register\_number (Number of registers) ist jeder Wert zwischen 1 und 110 zulässig. Diese Anzahl von Registern wird gelesen.



## Anwendungsbeispiel

Tabelle 3- 24 Umsetzung der Modbus-Adressierung für Funktionscodes FC 03, 06, 16

Modbus-Adresse im Übertragungstelegramm	SIMATIC-Speicherbereich
0	ab Datenbaustein DB 800 (Basis-DB-Nummer)

### Quell-DB SEND

Die Tabelle zeigt den Aufbau des SEND-Quellbereichs:

Adresse	Name	Art	Anfangswert	Kommentar
+0.0	Adresse	BYTE	B#16#5	Slave-Adresse
+1.0	Funktion	BYTE	B#16#3	Funktionscode
+2.0	Register-Start- adresse	WORD	W#16#0040	Register- Startadresse
+4.0	Registeranzahl	INT	2	Registeranzahl

### Ziel-DB RCV

Die Tabelle zeigt den Inhalt des RCV-Zielbereichs:

Adresse	Name	Art	Aktualwert	Kommentar
+0.0	Data[1]	WORD	W#16#2123	Daten
+2.0	Data[2]	WORD	W#16#2527	Daten



### 3.6.9 Funktionscode 04 – Read Input Registers

#### Zweck und Aufbau

Der Funktionscode 04 – Read Input Registers ist folgendermaßen charakterisiert:

<b>Funktion</b>	Diese Funktion ermöglicht das Lesen von Datenworten aus einem Datenbaustein durch das Modbus-Mastersystem					
<b>Anforderungstelegramm</b>	ADDR	FUNC	start_register	register_number	CRC	
<b>Antworttelegramm</b>	ADDR	FUNC	Byte_count n	n/2-Register DATA	CRC	(High, Low)
<b>LEN in Bytes</b>	6					

#### start\_address

Die Modbus-Registeradresse "start\_register" wird vom Treiber wie folgt interpretiert:

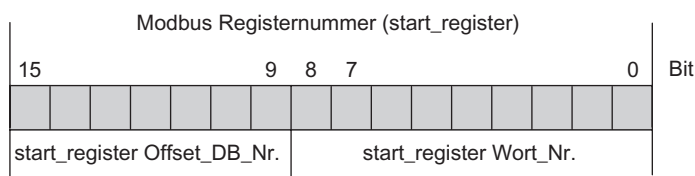


Bild 3-12 Interpretation der Modbus-Registernummer

Zur weiteren Adressbildung verwendet der FB81(S\_MODB) die Basis-DB-Nummer (aus DB xxxxx), die im Umsetzungs-DB für FC 04 eingegeben ist.

Die Adressberechnung für den Zugriff (Adressumsetzung) erfolgt dann in zwei Schritten:

Zugriff auf SIMATIC	Umsetzungsformel
Datenbaustein-DB (resultierender DB)	$=(\text{Basis-DB-Nummer } xxxxx + \text{start\_register Offset\_DB\_Nr.})$
Datenwort DBW	$=(\text{start\_register Wort\_Nr. } *2)$

#### Berechnungsformel für start\_register

Wenn der resultierende DB bekannt ist, der gelesen werden soll, kann die im Mastersystem benötigte Modbus-Adresse start\_register nach folgender Formel berechnet werden:

$$\text{start\_register} = ((\text{resultierender DB} - \text{Basis-DB-Nummer}) * 512) + (\text{Datenwort\_DBW} / 2)$$

Hierbei dürfen nur geradzahlige Datenwortnummern zugrunde gelegt werden.

#### register\_number

Als register\_number (Number of registers) ist jeder Wert zwischen 1 und 110 zulässig. Diese Anzahl von Registern wird gelesen.

## Anwendungsbeispiel

Tabelle 3- 26 Umsetzung der Modbus-Adressierung für Funktionscodes FC 04

Modbus-Adresse im Übertragungstelegramm	SIMATIC-Speicherbereich
0	ab Datenbaustein DB 900 (Basis-DB-Nummer)

## Quell-DB SEND

Die Tabelle zeigt den Aufbau des SEND-Quellbereichs:

Adresse	Name	Art	Anfangswert	Kommentar
+0.0	Adresse	BYTE	B#16#5	Slave-Adresse
+1.0	Funktion	BYTE	B#16#4	Funktionscode
+2.0	Register- Startadresse	WORD	W#16#0050	Register- Startadresse
+4.0	Registeranzahl	INT	3	Registeranzahl

## Ziel-DB RCV

Die Tabelle zeigt den Inhalt des RCV-Zielbereichs:

Adresse	Name	Art	Aktualwert	Kommentar
+0.0	Data[1]	WORD	W#16#2123	Daten
+2.0	Data[2]	WORD	W#16#2527	Daten
+4.0	Data[3]	WORD	W#16#3536	Daten



### 3.6.10 Funktionscode 05 – Force Single Coil

#### Zweck und Aufbau

Der Funktionscode 05 – Force Single Coil ist folgendermaßen charakterisiert:

<b>Funktion</b>	Diese Funktion ermöglicht das Schreiben eines Bits in den nachfolgend aufgeführten SIMATIC-Speicherbereichen durch das Modbus-Mastersystem				
<b>Anforderungstelegramm</b>	ADDR	FUNC	coil_address	DATA-on/off	CRC
<b>Antworttelegramm</b>	ADDR	FUNC	coil_address	DATA-on/off	CRC
<b>LEN in Bytes</b>	6				

#### coil\_address

Die Modbus-Bitadresse "coil\_address" wird vom Treiber wie folgt interpretiert:

FB81(S\_MODB) prüft, ob sich "coil\_address" in einem der Bereiche befindet, die im Umsetzungs-DB für FC 01, 05, 15 (von/bis: Merker, Ausgänge, Zeiten, Zähler) angegeben wurden.

Beindet sich die Modbus-Bitadresse start_address im Bereich	erfolgt der Zugriff auf den SIMATIC-Speicherbereich	
Von <i>aaaaa</i> bis <i>bbbbb</i>	ab Merker	M <i>uuuu</i> .0
Von <i>cccc</i> bis <i>dddd</i>	ab Ausgang	A <i>oooo</i> .0

Die Adressberechnung für den Zugriff (Adressumsetzung) erfolgt dann in zwei Schritten:

Zugriff beginnend mit SIMATIC	Umsetzungsformel	
Merkerbyte	$=((\text{start\_address} - \text{cccc}) / 8)$	+ <i>oooo</i>
Ausgangsbyte	$=((\text{start\_address} - \text{aaaa}) / 8)$	+ <i>uuuuu</i>

#### Zugriff auf Merker und Ausgänge

Beim Zugriff auf die SIMATIC-Bereiche Merker und Ausgänge wird die verbleibende Rest-Bit\_Nummer berechnet und dazu verwendet, das entsprechende Bit innerhalb des Merker- oder Ausgangsbytes zu adressieren.

#### Zugriff auf Zeiten und Zähler

Ein Zugriff auf die SIMATIC-Bereiche Zeiten und Zähler ist bei diesem Funktionscode FC 05 nicht zulässig und wird vom Treiber mit einem Fehlertelegramm abgewiesen.

**DATA-on/off**

Als DATA-on/off sind folgende zwei Werte zulässig:

FF00H = Bit setzen.

0000H = Bit löschen.

**Anwendungsbeispiel**

Tabelle 3- 28 Umsetzung der Modbus-Adressierung für Funktionscodes FC 01, 05, 15

Modbus-Adresse im Übertragungstelegramm	SIMATIC-Speicherbereich
Von 0 bis 2047	ab Merker M 1000.0
Von 2048 bis 2559	ab Ausgang A 256.0

**Quell-DB SEND**

Die Tabelle zeigt den Aufbau des SEND-Quellbereichs:

Adresse	Name	Art	Anfangswert	Kommentar
+0.0	Adresse	BYTE	B#16#5	Slave-Adresse
+1.0	Funktion	BYTE	B#16#5	Funktionscode
+2.0	Bitadresse	WORD	W#16#0019	Bitadresse
+4.0	Bitstatus	WORD	W#16#FF00	Bitstatus

Der Slave muss das Anforderungstelegramm unverändert an den Master zurückgeben (Echo).

**Ziel-DB RCV**

Die Tabelle zeigt den Inhalt des RCV-Zielbereichs:

Adresse	Name	Art	Aktualwert	Kommentar
+0.0	Adresse	BYTE	B#16#5	Slave-Adresse
+1.0	Funktion	BYTE	B#16#5	Funktionscode
+2.0	Bitadresse	WORD	W#16#0019	Bitadresse
+4.0	Bitstatus	WORD	W#16#FF00	Bitstatus

**Adressberechnung:**

Die Modbus-Adresse "coil\_address" 0809 Hex (2057 dezimal) befindet sich im Bereich Ausgänge:

$$\begin{aligned} \text{Ausgangsbyte} &= ((\text{coil\_address} - \text{cccc}) / 8) && + 00000 \\ &= ((2057 - 2048) / 8) && + 256 \\ &= 257 \end{aligned}$$

Die verbleibende Rest-Bit\_Nummer ergibt:

$$\begin{aligned} \text{Rest-Bit\_Nr.} &= ((\text{coil\_address} - \text{cccc}) \% 8) && [\text{Modulo } 8] \\ &= ((2057 - 2048) \% 8) \\ &= 1 ; \end{aligned}$$

Es erfolgt ein Zugriff auf den Ausgang A 257.1.

**Weitere Beispiele**

Weitere Zugriffsbeispiele auf Merker und Ausgänge können unter dem FC 01 nachgeschlagen werden.



### 3.6.11 Funktionscode 06 – Preset Single Register

#### Zweck und Aufbau

Der Funktionscode 06 – Preset Single Register ist folgendermaßen charakterisiert:

<b>Funktion</b>	Diese Funktion ermöglicht das Schreiben eines Datenworts in einem Datenbaustein der CPU durch das Modbus-Mastersystem				
<b>Anforderungstelegramm</b>	ADDR	FUNC	start_register	DATA value (High, Low)	CRC
<b>Antworttelegramm</b>	ADDR	FUNC	start_register	DATA value (High, Low)	CRC
<b>LEN in Bytes</b>	6				

#### start\_register

Die Modbus-Registeradresse "start\_register" wird vom Treiber wie folgt interpretiert:

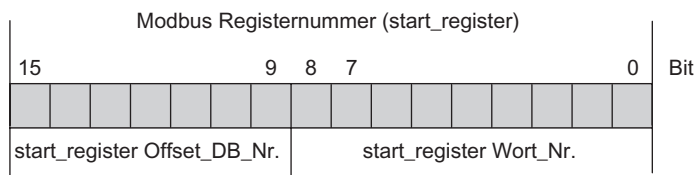


Bild 3-14 Interpretation der Modbus-Registernummer

Zur weiteren Adressbildung verwendet der FB81(S\_MODB) die Basis-DB-Nummer (ab DB xxxxx), die im Umsetzungs-DB für FC 03, 06, 16 eingegeben ist.

Die Adressberechnung für den Zugriff (Adressumsetzung) erfolgt dann in zwei Schritten:

Zugriff auf SIMATIC	Umsetzungsformel
Datenbaustein-DB (resultierender DB)	$=(\text{Basis-DB-Nummer xxxxx} + \text{start\_register-Offset\_DB\_Nr.})$
Datenwort DBW	$=(\text{start\_register-Wort\_Nr.} * 2)$

Wenn der resultierende DB bekannt ist, der gelesen werden soll, kann die im Mastersystem benötigte Modbus-Adresse start\_register nach folgender Formel berechnet werden:

$$\text{start\_register} = ((\text{resultierender DB} - \text{Basis-DB-Nummer}) * 512) + (\text{Datenwort\_DBW} / 2)$$

Hierbei dürfen nur geradzahlige Datennummern zugrunde gelegt werden.

#### DATA Value

Als DATA Value (Registerwert) kann jeder beliebige Wert verwendet werden.

**Anwendungsbeispiel für die Parametrierung:**

Tabelle 3- 29 Umsetzung der Modbus-Adressierung für Funktionscodes FC 03, 06, 16

Modbus-Adresse im Übertragungstelegramm	SIMATIC-Speicherbereich
0	ab Datenbaustein (Basis-DB-Nummer) DB 800

**Quell-DB SEND**

Die Tabelle zeigt den Aufbau des SEND-Quellbereichs:

Adresse	Name	Art	Anfangswert	Kommentar
+0.0	Adresse	BYTE	B#16#5	Slave-Adresse
+1.0	Funktion	BYTE	B#16#6	Funktionscode
+2.0	Registeradresse	WORD	W#16#0180	Registeradresse
+4.0	Registerwert	WORD	W#16#3E7F	Registerwert

**Ziel-DB RCV**

Die Tabelle zeigt den Inhalt des RCV-Zielbereichs:

Adresse	Name	Art	Aktualwert	Kommentar
+0.0	Adresse	BYTE	B#16#5	Slave-Adresse
+1.0	Funktion	BYTE	B#16#6	Funktionscode
+2.0	Registeradresse	WORD	W#16#0180	Registeradresse
+4.0	Registerwert	WORD	W#16#3E7F	Registerwert

**Adressberechnung:**

Die Modbus-Adresse "start\_register" 0180 Hex (384 dezimal) wird wie folgt interpretiert:

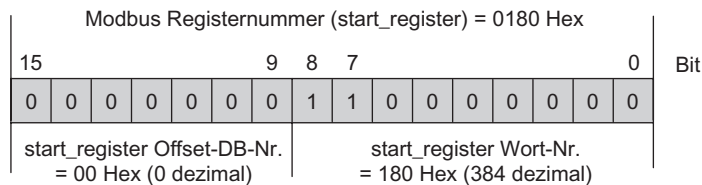


Bild 3-15 Interpretation der Modbus-Registernummer 0180 Hex

Datenbaustein-DB  
(resultierender DB) = (Basis-DB-Nummer xxxxx+start\_register Offset\_DB\_Nr.)  
= (800 + 0)  
= 800 ;

Datenwort DBW = (start\_register Wort\_Nr.\* 2)  
= (384 \* 2)  
= 768 ;

Es erfolgt ein Zugriff auf den DB 800, Datenwort DBW 768.

**Weitere Beispiele**

Weitere Zugriffsbeispiele finden Sie unter FC 03.

### 3.6.12 Funktionscode 08 – Loop Back Diagnostic Test

#### Zweck und Aufbau

Der Funktionscode 08 – Loop Back Diagnostic Test ist folgendermaßen charakterisiert:

<b>Funktion</b>	Diese Funktion dient zur Überprüfung der Kommunikationsverbindung. Sie hat keine Auswirkung auf die S7-CPU und die Anwenderprogramme und Anwenderdaten. Das empfangene Telegramm wird vom Treiber selbständig an das Mastersystem zurückgesendet.				
<b>Anforderungstelegramm</b>	ADDR	FUNC	Diagnosecode (High, Low)	Testdaten	CRC
<b>Antworttelegramm</b>	ADDR	FUNC	Diagnosecode (High, Low)	Testdaten	CRC
<b>Diagnosecode</b>	Es wird nur der Diagnosecode 0000 unterstützt.				
<b>Testdaten</b>	Jeder Wert (16 Bit).				
<b>LEN in Bytes</b>	6				

#### Anwendungsbeispiel

##### Quell-DB SEND

Die Tabelle zeigt den Aufbau des SEND-Quellbereichs:

Adresse	Name	Art	Anfangswert	Kommentar
+0.0	Adresse	BYTE	B#16#5	Slave-Adresse
+1.0	Funktion	BYTE	B#16#8	Funktionscode
+2.0	Diagnosecode	WORD	B#16#0000	Diagnosecode
+4.0	Registerwert	WORD	B#16#A5C3	Testwert

##### Ziel-DB RCV

Die Tabelle zeigt den Inhalt des RCV-Zielbereichs:

Adresse	Name	Art	Aktualwert	Kommentar
+0.0	Adresse	BYTE	B#16#5	Slave-Adresse
+1.0	Funktion	BYTE	B#16#8	Funktionscode
+2.0	Diagnosecode	WORD	B#16#0000	Diagnosecode
+4.0	Testwert	WORD	B#16#A5C3	Testwert

### 3.6.13 Funktionscode 15 – Force Multiple Coils

#### Zweck und Aufbau

Der Funktionscode 15 – Force Multiple Coils ist folgendermaßen charakterisiert:

<b>Funktion</b>	Diese Funktion ermöglicht das Schreiben mehrerer Bits in den nachfolgend aufgeführten SIMATIC-Speicherbereichen durch das Modbus-Mastersystem.						
<b>Anforderungs-telegramm</b>	ADDR	FUNC	start_adress	quantity	byte_count	n-DATA	CRC
<b>Antwort-telegramm</b>	ADDR	FUNC	start_adress		n Byte	DATA	CRC
<b>LEN in Bytes</b>	> 6						

#### start\_address

Die Modbus-Bitadresse "start\_address" wird vom Treiber wie folgt interpretiert:

FB81(S\_MODB) prüft, ob sich "start\_address" in einem der Bereiche befindet, die im Umsetzungs-DB für FC 01, 05, 15 (von/bis: Merker, Ausgänge, Zeiten, Zähler) angegeben wurden.

Beindet sich die Modbus-Bitadresse start_address im Bereich	erfolgt der Zugriff auf den SIMATIC-Speicherbereich	
Von <i>aaaaa</i> bis <i>bbbbb</i>	ab Merker	M <i>uuuu.0</i>
Von <i>cccc</i> bis <i>dddd</i>	ab Ausgang	A <i>oooo.0</i>

Die Adressberechnung für den Zugriff (Adressumsetzung) erfolgt dann folgendermaßen:

Zugriff beginnend mit SIMATIC	Umsetzungsformel	
Merkerbyte	$=((\text{start\_address} - \text{cccc}) / 8)$	+ <i>uuuu</i>
Ausgangsbyte	$=((\text{start\_address} - \text{aaaa}) / 8)$	+ <i>oooo</i>

#### Zugriff auf Merker und Ausgänge

Beim Zugriff auf die SIMATIC-Bereiche Merker und Ausgänge wird die verbleibende Rest-Bit\_Nummer berechnet und dazu verwendet, das entsprechende Bit innerhalb des Merker- oder Ausgangsbytes zu adressieren.

#### Zugriff auf Zeiten und Zähler

Ein Zugriff auf die SIMATIC-Bereiche Zeiten und Zähler ist bei diesem Funktionscode FC 15 nicht zulässig und wird vom Treiber mit einem Fehlertelegramm abgewiesen.

**Quantity**

Als Quantity (Bitanzahl) ist jeder Wert zwischen 1 und 1696 zulässig.

**DATA**

Das Feld DATA enthält Bitzustände (beliebige Werte).

**Anwendungsbeispiel**

Tabelle 3- 30 Umsetzung der Modbus-Adressierung für Funktionscodes FC 01, 05, 15

Modbus-Adresse im Übertragungstelegramm	SIMATIC-Speicherbereich
Von 0 bis 2047	ab Merker M 1000.0
Von 2048 bis 2559	ab Ausgang A 256.0

**Aktion**

Das Modbus-Mastersystem möchte die Merker M 1144.0 ... M 1144.7 und M 1145.0 ... M 1145.3 mit folgenden Bitzuständen beschreiben:

Merker	7	6	5	4	3	2	1	0	Bit
M 1144	ON	ON	OFF	OFF	ON	ON	OFF	ON	

Merker	7	6	5	4	3	2	1	0	Bit
M 1145	-	-	-	-	ON	OFF	OFF	ON	

**Quell-DB SEND**

Die Tabelle zeigt den Aufbau des SEND-Quellbereichs:

Adresse	Name	Art	Anfangswert	Kommentar
+0.0	Adresse	BYTE	B#16#5	Slave-Adresse
+1.0	Funktion	BYTE	B#16#0F	Funktionscode
+2.0	Bit-Startadresse	WORD	W#16#0058	Bit-Startadresse
+4.0	Bitanzahl	INT	10	Bitanzahl
+6.0	coil_state[1]	WORD	W#16#EFCD	Status Coil 5FH..58H/57H..50H

**Adressberechnung:**

Die Modbus-Adresse "coil\_address" 0480 Hex (1152 dezimal) befindet sich im Bereich Merker:

$$\begin{aligned} \text{Merkerbyte} &= ((\text{start\_address} - \text{aaaaa}) / 8) && + \text{uuuuu} \\ &= ((1152 - 0) / 8) && + 1000 \\ &= 1144; \end{aligned}$$

Die verbleibende Rest-Bit\_Nummer ergibt:

$$\begin{aligned} \text{Rest-Bit\_Nr.} &= ((\text{start\_address} - \text{aaaaa}) \% 8) && [\text{Modulo } 8] \\ &= ((1152 - 0) \% 8) \\ &= 0; \end{aligned}$$

Es erfolgt ein Zugriff auf Merker beginnend bei M 1144.0.

**Weitere Beispiele**

Weitere Zugriffsbeispiele auf Merker und Ausgänge können unter dem FC 01 nachgeschlagen werden.

### 3.6.14 Funktionscode 16 – Preset Multiple Registers

#### Zweck und Aufbau

Der Funktionscode 16 – Preset Multiple Registers ist folgendermaßen charakterisiert:

<b>Funktion</b>	Der Funktionscode ermöglicht das Schreiben von mehreren Datenworten in einem Datenbaustein der SIMATIC-CPU durch das Modbus-Mastersystem.						
<b>Anforderungs-telegramm</b>	ADDR	FUNC	start_register	quantity	byte_count N	n-DATA (High, Low)	CRC
<b>Antwort-telegramm</b>	ADDR	FUNC	start_register		quantity		CRC
<b>LEN in Bytes</b>	> 6						

#### start\_register

Die Modbus-Registeradresse "start\_register" wird vom Treiber wie folgt interpretiert:

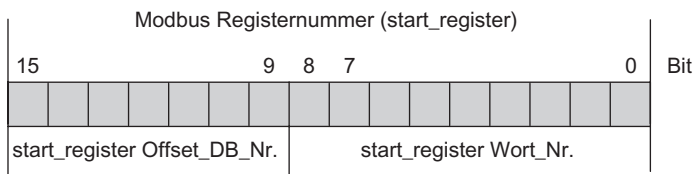


Bild 3-16 Interpretation der Modbus-Registernummer

Zur weiteren Adressbildung verwendet der FB81(S\_MODB) die Basis-DB-Nummer (ab DB xxxxx), die während der Parametrierung im Umsetzungs-DB für FC 03, 06, 16 eingegeben wurde.

Die Adressberechnung für den Zugriff (Adressumsetzung) erfolgt dann in zwei Schritten:

Zugriff auf SIMATIC	Umsetzungsformel
Datenbaustein-DB (resultierender DB)	=(Basis-DB-Nummer xxxxx + start_register-Offset_DB_Nr.)
Datenwort DBW	=(start_register Wort_Nr.* 2)

Wenn der resultierende DB bekannt ist, der geschrieben werden soll, kann die im Mastersystem benötigte Modbus-Adresse start\_register nach folgender Formel berechnet werden:

$$\text{start\_register} = ((\text{resultierender DB} - \text{Basis-DB-Nummer}) * 512) + (\text{Datenwort\_DBW} / 2)$$

Hierbei dürfen nur geradzahlige Datenwortnummern zugrunde gelegt werden.

#### Quantity

Als Quantity (Registeranzahl) ist jeder Wert zwischen 1 und 109 zulässig.



**DATA (High, Low)**

Als DATA (High, Low) (Registerwert) kann jeder beliebige Wert verwendet werden. Das Modbus-Mastersystem will die Datenworte DBW 100, DBW 102, DBW 104 von DB 800 mit den Werten CD09 Hex, DE1A Hex, EF2B Hex beschreiben.

**Anwendungsbeispiel**

Tabelle 3- 31 Umsetzung der Modbus-Adressierung für Funktionscodes FC 03, 06, 16

Modbus-Adresse im Übertragungstelegramm	SIMATIC-Speicherbereich	
0	ab Datenbaustein (Basis-DB-Nummer)	DB 800

**Quell-DB SEND**

Die Tabelle zeigt den Aufbau des SEND-Quellbereichs:

Adresse	Name	Art	Anfangswert	Kommentar
+0.0	Adresse	BYTE	B#16#5	Slave-Adresse
+1.0	Funktion	BYTE	B#16#10	Funktionscode
+2.0	Register- Startadresse	WORD	W#16#0060	Register- Startadresse
+4.0	Registeranzahl	INT	3	Registeranzahl
+6.0	reg_data[1]	WORD	W#16#41A1	Registerdaten
+8.0	reg_data[2]	WORD	W#16#42A2	Registerdaten
+10.0	reg_data[3]	WORD	W#16#43A3	Registerdaten

**Adressberechnung:**

Die Modbus-Adresse "start\_register" 0032 Hex (50 dezimal) wird wie folgt interpretiert:

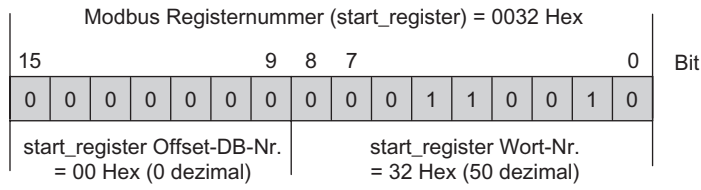


Bild 3-17 Interpretation der Modbus-Registernummer 0032 Hex

Datenbaustein-DB (resultierender DB) = (Basis-DB-Nummer xxxxx+start\_register Offset\_DB\_Nr.)  
 = (800 + 0)  
 = 800 ;

Datenwort DBW = (start\_register Wort\_Nr.\* 2)  
 = (50 \* 2)  
 = 100;

Es erfolgt ein Zugriff auf den DB 800, Datenwort DBW 100.

**Weitere Beispiele**

Weitere Zugriffsbeispiele finden Sie unter FC 03.

### 3.6.15 Bitorientierte Funktionscode-Umsetzung

#### Funktionscode 02

Der bitorientierte Funktionscode 02 erlaubt schreibgeschützten Zugriff auf die SIMATIC-Speicherbereiche Merker und Eingänge.

Über den Umsetzungs-DB kann festgelegt werden, von/bis welcher Modbus-Adresse auf Merker und auf Eingänge zugegriffen wird. Weiterhin kann parametrieren werden, ab welchem Datenelement im SIMATIC-Speicherbereich der Zugriff beginnen soll.

Die Modbus-Adressbereiche und SIMATIC-Speicherbereiche des FC 02 können unabhängig von denen der FC 01, 05, 15 ausgewählt werden.

Tabelle 3- 32 Adressbereiche

Modbus-Adresse im Übertragungstelegramm		SIMATIC-Speicherbereich	
Von kkkkk	Bis lllll	Merker	Ab
			M vvvv.0
Von nnnnn	Bis rrrr	Eingänge	Ab
			E sssss.0

### 3.6.16 Registerorientierte Funktionscode-Umsetzung

#### Funktionscodes 03, 06, 16

Die registerorientierten Funktionscodes 03, 06, 16 erlauben den lesenden und schreibenden Zugriff auf den SIMATIC-Speicherbereich Datenbausteine.

Die Berechnung der gewünschten Datenbausteinnummer erfolgt zweistufig.

1. Mit der Parametrieroberfläche kann eine Basis-DB-Nummer festgelegt werden. Dieser Basis-DB ist dann der erste DB, auf den zugegriffen werden kann.
2. Die im Telegramm übertragene Modbus-Adresse Start\_Register (Registernummer) wird wie folgt interpretiert:

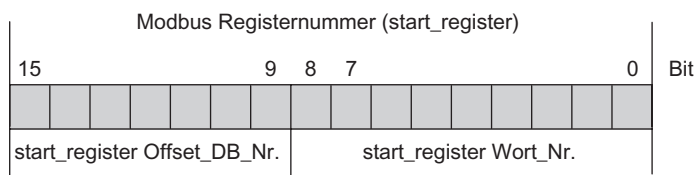


Bild 3-18 Interpretation der Modbus-Registernummer

### Resultierende DB-Nummer

Die resultierende DB-Nummer, auf die dann zugegriffen wird, ergibt sich aus:  
Basis-DB-Nummer + Offset-DB-Nummer.

Damit kann auf einen Datenbausteinbereich von 128 zusammenhängenden Datenbausteinen innerhalb des gesamten adressierbaren Datenbausteinbereichs (65535 DBs) zugegriffen werden.

### Wortnummer im DB

Innerhalb eines jeden Datenbausteins kann dann über die Wortnummer der Bereich von DBW 0 bis DBW 1022 adressiert werden.

Die von der Grundstruktur her byteweise organisierten DBs werden dabei durch den Treiber wortweise interpretiert.

### Besonderheiten beim Funktionscode 04

Der registerorientierte Funktionscode 04 erlaubt nur lesenden Zugriff auf den SIMATIC-Speicherbereich Datenbausteine.

Die Art und Weise des Zugriffs erfolgt wie bei den Funktionscodes 03, 06, 16 beschrieben.

Für den Funktionscode 04 ist mit dem Umsetzungs-DB eine eigene Basis-DB-Nummer frei parametrierbar. Damit kann ein zweiter unabhängiger 128 DBs umfassender Bereich selektiert werden.

Auf diese DBs ist jedoch nur ein lesender Zugriff möglich.

### 3.6.17 Freigeben/Sperren von Schreibzugriffen

#### Funktionscodes 05, 06, 15, 16

Für die schreibenden Funktionscodes 05, 06, 15 und 16 kann der Zugriff auf die entsprechenden SIMATIC-Speicherbereiche gesperrt bzw. eingeschränkt werden.

Mit dem Umsetzungs-DB kann ein Bereich festgelegt werden, der für Schreibzugriffe vom Modbus-Mastersystem freigegeben ist.

Versucht der Master auf SIMATIC-Speicherbereiche zuzugreifen, die außerhalb dieses freigegebenen Bereichs liegen, wird der Zugriff durch ein Fehlertelegramm (Ausnahme) abgewiesen. Die Tabelle zeigt das Freigeben von Schreibzugriffen.

Tabelle 3- 33 Freigeben von Schreibzugriffen

38.0	DB_Number _FC_04	WORD	W#16#0	W#16#2	DB	04
40.0	DB_Min	WORD	W#16#0	W#16#1	kleinste verwendete DB- Nummer	Grenzen
42.0	DB_Max	WORD	W#16#0	W#16#6	größte verwendete DB- Nummer	
44.0	M_Min	WORD	W#16#0	W#16#1F4	kleinster verwendeter Merker	
46.0	M_Max	WORD	W#16#0	W#16#4B0	größter verwendeter Merker	
48.0	Q_Min	WORD	W#16#0	W#16#0	kleinster verwendeter Ausgang	
50.0	Q_Max	WORD	W#16#0	W#16#64	größter verwendeter Ausgang	

### 3.6.18 Umsetzung der Modbus-Adressen für Bitfunktionen

#### Funktionscodes 01, 05, 15

Die bitorientierten Funktionscodes 01, 05 und 15 erlauben lesenden und schreibenden Zugriff auf die SIMATIC-Speicherbereiche Merker, Ausgänge, Zeiten, Zähler.

Zeiten und Zähler sind schreibgeschützt mit FC01.

Über den Umsetzungs-DB kann festgelegt werden, von/bis welcher Modbus-Adresse auf Ausgänge, Zeiten und Zähler zugegriffen wird. Weiterhin kann parametrierbar werden, ab welchem Datenelement im SIMATIC-Speicherbereich der Zugriff beginnen soll.

#### Übersicht über 01, 05, 15

Tabelle 3- 34 Umsetzung der Modbus-Adressierung für Funktionscodes FC 01, 05, 15

Parameter-DB		Eingang	Bedeutung
<b>SIMATIC-Bereich Merker</b>			
Modbus-Adresse im Übertragungstelegramm (Bitnummer)	von aaaa	von 0 bis 65535 (dezimal)	angefangen bei dieser Modbus-Adresse
	bis bbbb	von 0 bis 65535 (dezimal)	einschließlich dieser Modbus-Adresse
SIMATIC-Speicherbereich Merker (Merker)	ab M uuuuu.0	von 0 bis 65535 (dezimal)	ab diesem Merkerbyte
<b>SIMATIC-Bereich Ausgänge</b>			
Modbus-Adresse im Übertragungstelegramm (Bitnummer)	von cccc	von 0 bis 65535 (dezimal)	angefangen bei dieser Modbus-Adresse
	bis dddd	von 0 bis 65535 (dezimal)	einschließlich dieser Modbus-Adresse
SIMATIC-Speicherbereich Ausgänge (Nummer Ausgangsbyte)	ab A ooooo.0	von 0 bis 65535 (dezimal)	ab diesem Ausgangsbyte
<b>SIMATIC-Bereich Zeiten</b>			
Modbus-Adresse im Übertragungstelegramm (Bitnummer)	von eeee	von 0 bis 65535 (dezimal)	angefangen bei dieser Modbus-Adresse
	bis ffff	von 0 bis 65535 (dezimal)	einschließlich dieser Modbus-Adresse
SIMATIC-Speicherbereich Zeiten (Nummer der Zeit)	ab bis tttt	von 0 bis 65535 (dezimal)	ab dieser Zeit (= 16 Bitwort)
<b>SIMATIC-Bereich Zähler</b>			
Modbus-Adresse im Übertragungstelegramm (Bitnummer)	von gggg	von 0 bis 65535 (dezimal)	angefangen bei dieser Modbus-Adresse
	bis hhhh	von 0 bis 65535 (dezimal)	einschließlich dieser Modbus-Adresse
SIMATIC-Speicherbereich Zähler (Nummer des Zählers)	ab Z zzzzz	von 0 bis 65535 (dezimal)	ab diesem Zähler (= 16 Bitwort)

### Modbus-Adresse "Von/Bis"

Mit der "Von"-Adresse kann die Modbus-Adresse parametrieren werden, mit der der jeweilige Bereich z.B. Merker, Ausgänge, usw. beginnt (= erste Bitnummer des Bereichs).

Mit der "Bis"-Adresse kann die Modbus-Adresse parametrieren werden, mit der der jeweilige Bereich z.B. Merker, Ausgänge, usw. endet (= letzte Bitnummer des Bereichs).

Die "Von"/"Bis"-Adressen beziehen sich auf die Modbus-Adresse im Übertragungstelegramm (Bitnummern ab 0) bei den Funktionscodes FC 01, 05, 15.

Die einzelnen "Von/Bis"-Bereiche dürfen nicht überlappen.

Lücken zwischen den einzelnen "Von/Bis"-Bereichen sind zulässig.

### SIMATIC-Speicherbereich "ab"

Mit der Angabe "ab" kann der Anfang des SIMATIC-Bereichs festgelegt werden, auf den der Modbus-Bereich "Von/Bis" abgebildet wird (= erste Merkerbyte-, Ausgangsbyte-/Zeiten-/Zählernummer des SIMATIC-Bereichs).

### Beispiel für FC 01, 05, 15

Tabelle 3- 35 Umsetzung der Modbus-Adressierung für Funktionscodes FC 01, 05, 15

Parameter-DB		Eingang	Bedeutung
<b>SIMATIC-Bereich Merker</b>			
<i>Modbus-Adresse</i> im Übertragungstelegramm (Bitnummer)	von 0	von 0 bis 65535 (dezimal)	angefangen bei dieser Modbus-Adresse
	bis 2047	von 0 bis 65535 (dezimal)	einschließlich dieser Modbus-Adresse
SIMATIC-Speicherbereich Merker (Merker)	ab M 1000.0	von 0 bis 65535 (dezimal)	ab diesem Merkerbyte
<b>SIMATIC-Bereich Ausgänge</b>			
<i>Modbus-Adresse</i> im Übertragungstelegramm (Bitnummer)	von 2048	von 0 bis 65535 (dezimal)	angefangen bei dieser Modbus-Adresse
	bis 2559	von 0 bis 65535 (dezimal)	einschließlich dieser Modbus-Adresse
SIMATIC-Speicherbereich <i>Ausgänge</i> (Nummer Ausgangsbyte)	ab A 256.0	von 0 bis 65535 (dezimal)	ab diesem Ausgangsbyte
<b>SIMATIC-Bereich Zeiten</b>			
<i>Modbus-Adresse</i> im Übertragungstelegramm (Bitnummer)	von 4096	von 0 bis 65535 (dezimal)	angefangen bei dieser Modbus-Adresse
	bis 4255	von 0 bis 65535 (dezimal)	einschließlich dieser Modbus-Adresse
SIMATIC-Speicherbereich <i>Zeiten</i> (Nummer der Zeit)	ab T 100	von 0 bis 65535 (dezimal)	ab dieser Zeit (= 16 Bitwort)

3.6 Modbus-Slave-Treiber

Parameter-DB		Eingang	Bedeutung
<b>SIMATIC-Bereich Zähler</b>			
Modbus-Adresse im Übertragungstelegramm (Bitnummer)	von 4256	von 0 bis 65535 (dezimal)	angefangen bei dieser Modbus-Adresse
	bis 4415	von 0 bis 65535 (dezimal)	einschließlich dieser Modbus-Adresse
SIMATIC-Speicherbereich Zähler (Nummer des Zählers)	ab Z 120	von 0 bis 65535 (dezimal)	ab diesem Zähler (= 16 Bitwort)

Die Modbus-Adressen von 0 bis 2047 greifen auf SIMATIC-Merker ab Merker M 1000.0 zu. D.h.: Länge des Bereichs = 2048 Bit = 256 Byte, dies bedeutet letztes Merkerbit = M 1255.7.

Die Modbus-Adressen von 2048 bis 2559 greifen auf SIMATIC-Ausgänge ab Ausgang A 256.0 zu. D.h.: Länge des Bereichs = 512 Bit = 64 Byte, dies bedeutet letztes Ausgangsbit = A 319.7.

Die Modbus-Adressen von 4096 bis 4255 greifen auf SIMATIC-Zeiten ab Zeit T 100 zu. D.h.: Länge des Bereichs = 160 Bit = 10 Worte, dies bedeutet letzte Zeit = T 109.

Die Modbus-Adressen von 4256 bis 4415 greifen auf SIMATIC-Zähler ab Zähler Z 120 zu. D.h.: Länge des Bereichs = 160 Bit = 10 Worte, dies bedeutet letzter Zähler = Z 129.

Übersicht über FC02

Tabelle 3- 36 Umsetzung der Modbus-Adressierung für FC 02

Parameter-DB		Eingang	Bedeutung
<b>SIMATIC-Bereich Merker</b>			
Modbus-Adresse in Übertragungstelegramm (Bitnummer)	von	von 0 bis 65535 (dezimal)	angefangen bei dieser Modbus-Adresse
	bis	von 0 bis 65535 (dezimal)	einschließlich dieser Modbus-Adresse
SIMATIC-Bereich Merker	ab	von 0 bis 65535 (dezimal)	ab diesem Merkerbyte
<b>SIMATIC-Bereich Eingänge</b>			
Modbus-Adresse im Übertragungstelegramm (Bitnummer)	von	von 0 bis 65535 (dezimal)	angefangen bei dieser Modbus-Adresse
	bis	von 0 bis 65535 (dezimal)	einschließlich dieser Modbus-Adresse
SIMATIC-Speicherbereich Eingänge (Nummer Eingangsbyte)	ab E	von 0 bis 65535 (dezimal)	ab diesem Eingangsbyte



### Modbus-Adresse "Von/Bis"

Mit der "Von"-Adresse kann die Modbus-Adresse parametrieren werden, mit der der jeweilige Bereich z.B. Merker, Eingänge, usw. beginnt (= erste Bitnummer des Bereichs).

Mit der "Bis"-Adresse kann die Modbus-Adresse parametrieren werden, mit der der jeweilige Bereich endet (= letzte Bitnummer des Bereichs).

Die "Von/Bis"-Adressen beziehen sich auf die Modbus-Adresse im Übertragungstelegramm (Bitnummern ab 0) beim Funktionscode FC 02.

Die einzelnen "Von/Bis"-Bereiche dürfen nicht überlappen.

Lücken zwischen den einzelnen "Von/Bis"-Bereichen sind zulässig.

### SIMATIC-Speicherbereich "ab"

Mit der Angabe "ab" kann der Anfang des SIMATIC-Bereichs festgelegt werden, auf den der Modbus-Bereich "Von/Bis" abgebildet wird (= erste Merkerbyte-, Eingangsbytenummer des SIMATIC-Bereichs).

### Beispiel für FC 02

Tabelle 3- 37 Umsetzung der Modbus-Adressierung für FC 02

Parameter-DB		Eingang	Bedeutung
<b>SIMATIC-Bereich Merker</b>			
Modbus-Adresse in Übertragungstelegramm (Bitnummer)	von 0	von 0 bis 65535 (dezimal)	angefangen bei dieser Modbus-Adresse
	bis 4095	von 0 bis 65535 (dezimal)	einschließlich dieser Modbus-Adresse
SIMATIC-Bereich Merker	ab M 0.0	von 0 bis 65535 (dezimal)	ab diesem Merkerbyte
<b>SIMATIC-Bereich Eingänge</b>			
Modbus-Adresse im Übertragungstelegramm (Bitnummer)	von 4096	von 0 bis 65535 (dezimal)	angefangen bei dieser Modbus-Adresse
	bis 5119	von 0 bis 65535 (dezimal)	einschließlich dieser Modbus-Adresse
SIMATIC-Speicherbereich Eingänge (Nummer Eingangsbyte)	ab E 128.0	von 0 bis 65535 (dezimal)	ab diesem Eingangsbyte

Die Modbus-Adressen von 0 bis 4095 greifen auf SIMATIC-Merker ab Merker M 0.0 zu: d.h. Länge des Bereichs = 4096 Bit = 512 Byte, dies bedeutet letztes Merkerbit = M 511.7.

Die Modbus-Adressen von 4096 bis 5119 greifen auf SIMATIC-Eingänge ab Eingang E 128.0 zu: d.h. Länge des Bereichs = 1024 Bit = 128 Byte, dies bedeutet letztes Eingangsbit = E 255.7.

---

**Hinweis**

Die Eingabe des Wertes "ab Merker" ist völlig unabhängig von der Eingabe "ab Merker" bei den Funktionscodes 01, 05, 15.

Mit dem FC 02 kann also ein zweiter SIMATIC-Merkerbereich verwendet werden (nur lesend), der vom ersten völlig unabhängig ist.

---

### 3.6.19 Umsetzung der Modbus-Adressen für Registerfunktionen

#### Übersicht über FC 03, 06, 16

Tabelle 3- 38 Umsetzung der Modbus-Adressierung für FC 03, 06, 16

Parameter-DB		Eingang	Bedeutung
<b>SIMATIC-Bereich Datenbausteine</b>			
<i>Modbus-Adresse</i> = 0 im Übertragungstelegramm (Registernummer) bedeutet Zugriff auf:			
SIMATIC-Speicherbereich <i>Datenbaustein</i>	ab DB	von 1 bis 65535 (dezimal)	ab diesem Datenbaustein ab DBW 0 (= Basis-DB-Nummer)

#### ab DB

Mit der Eingabe "ab DB" kann der erste Datenbaustein des SIMATIC-Bereichs festgelegt werden, auf den zugegriffen wird (= Basis-DB-Nummer).

Auf diesen DB wird zugegriffen, wenn die Register-Nummer des Modbus-Telegramms den Wert 0 hat, beginnend beim Datenwort DBW 0.

Höhere Modbus-Register-Nummern greifen auf die folgenden Datenworte/Datenbausteine zu.

Es können bis zu 127 Folge-DBs adressiert werden.

Für den Zugriff auf die einzelnen Folge-DBs werden vom Treiber die Bits 9 – 15 der Modbus-Registernummer interpretiert.

#### Anwendungsbeispiel

Tabelle 3- 39 Umsetzung der Modbus-Adressierung für FC 03, 06, 16

Parameter-DB		Eingang	Bedeutung
<b>SIMATIC-Bereich Datenbausteine</b>			
<i>Modbus-Adresse</i> = 0 im Übertragungstelegramm (Registernummer) bedeutet Zugriff auf:			
SIMATIC-Speicherbereich <i>Datenbaustein</i>	ab DB 800	von 1 bis 65535 (dezimal)	ab diesem Datenbaustein ab DBW 0 (als Basis-DB-Nummer)

Mit der Modbus-Registeradresse 0 wird im SIMATIC-System auf den Datenbaustein 800 ab DBW 0 zugegriffen.

Höhere Modbus-Registeradressen ( $\geq 512$  usw.) greifen auf die folgenden DBs wie DB 801 usw. zu.

### Übersicht über FC 04

Tabelle 3- 40 Umsetzung der Modbus-Adressierung für FC 04

Parameter-DB		Eingang	Bedeutung
<b>SIMATIC-Bereich Datenbausteine</b>			
<i>Modbus-Adresse</i> = 0 im Übertragungstelegramm (Registernummer) bedeutet Zugriff auf:			
SIMATIC-Speicherbereich <i>Datenbausteine</i>	ab DB	von 1 bis 65535 (dezimal)	ab diesem Datenbaustein ab DBW 0 (als Basis-DB-Nummer)

#### ab DB

Mit der Eingabe "ab DB" kann der erste Datenbaustein des SIMATIC-Bereichs festgelegt werden, auf den zugegriffen wird (= Basis-DB-Nummer).

Auf diesen DB wird zugegriffen, wenn die Register-Nummer des Modbus-Telegramms den Wert 0 hat, beginnend beim Datenwort DBW 0.

Höhere Modbus-Register-Nummern greifen auf die folgenden Datenworte/Datenbausteine zu.

Es können bis zu 127 Folge-DBs adressiert werden. Für den Zugriff auf die einzelnen Folge-DBs werden vom Treiber die Bits 9 – 15 der Modbus-Registernummer interpretiert.

#### Hinweis

Die Eingabe des Wertes "ab DB" ist völlig unabhängig von der Eingabe "ab DB" bei den Funktionscodes 03, 06 und 16.

Mit dem FC 04 kann also ein zweiter SIMATIC-Datenbausteinbereich verwendet werden (nur lesend), der vom ersten völlig unabhängig ist.

### Beispiel für FC 04

Tabelle 3- 41 Umsetzung der Modbus-Adressierung für FC 04

Parameter-DB		Eingang	Bedeutung
<b>SIMATIC-Bereich Datenbausteine</b>			
<i>Modbus-Adresse</i> = 0 im Übertragungstelegramm (Registernummer) bedeutet Zugriff auf:			
SIMATIC-Speicherbereich <i>Datenbausteine</i>	ab DB 1200	von 1 bis 65535 (dezimal)	ab diesem Datenbaustein ab DBW 0 (als Basis-DB-Nummer)

Mit der Modbus-Registeradresse 0 wird im SIMATIC-System auf den Datenbaustein 1200 ab DBW 0 zugegriffen.

Höhere Modbus-Registeradressen ( $\geq 512$ , 1024 usw.) greifen auf die folgenden DBs wie DB 1201, 1202 usw. zu.

### 3.6.20 Grenzen für Schreibfunktionen

#### Übersicht über FC 05, 06, 16

Tabelle 3- 42 SIMATIC-Grenzen für Schreibzugriffe (FC 05, 06, 16)

Parameter-DB		Eingang	Bedeutung
Datenbausteine DB: resultierende DB-Nummer	DB MIN	von 1 bis 65535	erster freigegebener DB
	DB MAX	von 1 bis 65535	letzter freigegebener DB MAX=0 alle DBs gesperrt
Merker M (Nummer Merkerbyte)	M MIN	von 0 bis 65535	erstes freigegebenes Merkerbyte
	M MAX	von 1 bis 65535	letztes freigegebenes Merkerbyte MAX=0 alle Merker gesperrt
Ausgänge A (Nummer Ausgangsbyte)	Q MIN	von 0 bis 65535	erstes freigegebenes Ausgangsbyte
	Q MAX	von 1 bis 65535	letztes freigegebenes Ausgangsbyte MAX=0 alle Ausgänge gesperrt

#### SIMATIC-Speicherbereich MIN/MAX

Bei den schreibenden Funktionscodes können Unter- und Obergrenzen für den Zugriff (MIN / MAX) festgelegt werden. Ein schreibender Zugriff ist nur innerhalb dieses freigegebenen Bereichs möglich.

Wird als Obergrenze 0 angegeben, ist der komplette Bereich gesperrt.

Bitte achten Sie bei der Auswahl auf die jeweilige Bereichsgröße in der SIMATIC, die CPU-abhängig ist.

Ein Schreibzugriffsversuch vom Master auf einen Bereich außerhalb von Ober-/Untergrenze wird von der Baugruppe mit einem Fehlertelegramm abgewiesen.

Die MIN/MAX-Werte für den Bereich Datenbausteine sind als resultierende DB-Nummern anzugeben.

## Anwendungsbeispiel für FC 05, 06, 16

Tabelle 3- 43 SIMATIC-Grenzen für Schreibzugriffe (FC 05, 06, 16)

Parameter-DB		Eingang	Bedeutung
Datenbausteine DB: resultierende DB-Nummer	MIN 600	1 bis 65535	Erster freigegebener DB
	MAX 699	1 bis 65535	Letzter freigegebener DB MAX=0 alle DBs gesperrt
Merker M (Nummer Merkerbyte)	MIN 1000	0 bis 65535	Erstes freigegebenes Merkerbyte
	MAX 1127	1 bis 65535	Letztes freigegebenes Merkerbyte MAX=0 alle Merker gesperrt
Ausgänge A (Nummer Ausgangsbyte)	MIN 256	0 bis 65535	Erstes freigegebenes Ausgangsbyte
	MAX 319	1 bis 65535	Letztes freigegebenes Ausgangsbyte MAX=0 alle Ausgänge gesperrt

Auf die SIMATIC-Datenbausteine DB 600 bis DB 699 kann mit schreibenden Funktionscodes (FC 06, 16) zugegriffen werden.

Auf die SIMATIC-Merkerbytes MB 1000 bis MB 1127 (FC 05, 15) kann mit schreibenden Funktionscodes zugegriffen werden.

Auf die SIMATIC-Ausgangsbytes AB 256 bis AB 319 (FC 05, 15) kann mit schreibenden Funktionscodes zugegriffen werden.

## 3.7 Diagnose

### 3.7.1 Möglichkeiten der Diagnose

#### Prinzip

Mit den Diagnosefunktionen der seriellen Schnittstellenbaugruppe ET 200S Modbus/USS können Sie zu jedem Fehler, der während des Betriebs auftritt, die Fehlerursache ermitteln. Folgende Diagnosemöglichkeiten stehen Ihnen zur Verfügung:

- Diagnose über die Status-LEDs auf der Frontplatte der seriellen Schnittstellenbaugruppe ET 200S Modbus/USS
- Diagnose über den Ausgang STATUS der Funktionsbausteine
- Diagnose über PROFIBUS-Slave-Diagnose

### 3.7.2 Diagnoseinformationen der Status-LEDs

#### Funktion der Status-LEDs

Die folgenden Status-LEDs befinden sich auf der Frontplatte der seriellen Schnittstellenbaugruppe ET 200S Modbus/USS:

- **TX** (grün): Leuchtet, wenn die serielle Schnittstellenbaugruppe ET 200S Modbus/USS Daten über die Schnittstelle sendet.
- **RX** (grün): Leuchtet, wenn die serielle Schnittstellenbaugruppe ET 200S Modbus/USS Daten über die Schnittstelle empfängt.
- **SF** (rot): Sammelfehler LED zeigt einen der möglichen Fehler an:
  - Hardwarefehler
  - Parametrierungsfehler
  - Drahtbruch oder getrenntes Kabel zwischen der seriellen Schnittstellenbaugruppe ET 200S Modbus/USS und dem Kommunikationspartner: werden nur bei RS-422Slave-Diagnose-Schnittstellenverbindungen mit dem Parameter Vorbelegung der Empfangsleitung = R(A) 5V / R(B) 0V erkannt.
  - Kommunikationsfehler (Parität, Rahmenfehler, Pufferüberlauf)

### 3.7.3 Diagnosemeldungen der Funktionsbausteine

#### Aufbau der Diagnosemeldungen der Funktionsbausteine

Für eine Fehlerdiagnose besitzt jeder Funktionsbaustein einen Parameter STATUS. Jede STATUS-Meldungsnummer hat unabhängig vom verwendeten Funktionsbaustein die gleiche Bedeutung. Das folgende Bild zeigt den Aufbau des Parameters STATUS.

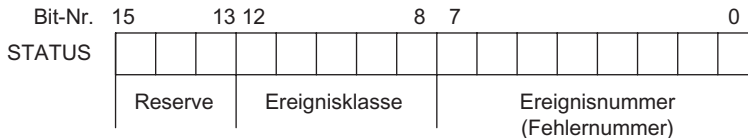


Bild 3-19 Aufbau des Parameters STATUS

Das folgende Bild zeigt als Beispiel den Inhalt des Parameters STATUS für das Ereignis "Auftragsabbruch wegen Neustart, Wiederanlauf oder Reset" (Ereignisklasse 1E<sub>H</sub>, Ereignisnummer 0D<sub>H</sub>).

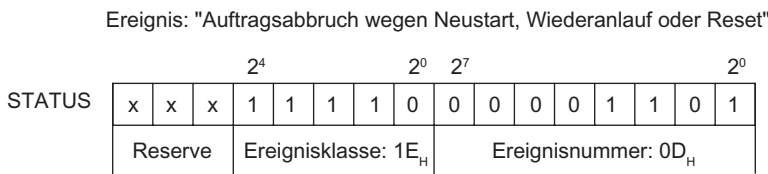


Bild 3-20 Beispiel: Parameter STATUS für Ereignisklasse 1EH, Ereignis 0DH

#### Aufrufen der Variablen SFCERR

Die Variable SFCERR enthält weitere Informationen zu den Fehlern 14 (1E 0E<sub>H</sub>) und 15 (1E 0F<sub>H</sub>) in Ereignisklasse 30.

Laden Sie die Variable SFCERR aus dem Instanz-DB des entsprechenden Funktionsbausteins.

Die Fehlermeldungen, die in der Variablen SFCERR eingetragen werden, werden im Abschnitt zu den Systemfunktionen SFC14 "DPRD\_DAT" und SFC15 "DPWR\_DAT" im Referenzhandbuch *Systemsoftware für S7 300/400, System- und Standardfunktionen* beschrieben.



## Bedeutung der Diagnosemeldungen der Funktionsbausteine

Die folgenden Tabellen beschreiben die Ereignisklassen, die Definitionen der Ereignisnummern und die empfohlene Abhilfe zu jeder Fehlerbedingung.

Tabelle 3- 44 Ereignisklasse 2 (0x02 Hex): Fehler bei Bearbeitung eines CPU-Auftrags

<b>Ereignisklasse 2 (0x02 Hex): "Fehler bei der Initialisierung"</b>			
<b>Ereignisnummer</b>	<b>Ereignisnummer (dezimal)</b>	<b>Ereignis</b>	<b>Abhilfe</b>
(02) 01 <sub>H</sub>	1	Keine (gültige) Parametrierung vorhanden.	Versorgen Sie die Baugruppe mit korrekten Parametern. Überprüfen Sie ggf. die ordnungsgemäße Installation der Anlage.

Tabelle 3- 45 Ereignisklasse 5 (05 Hex): Fehler bei Bearbeitung eines CPU-Auftrags

<b>Ereignisklasse 5 (05 Hex): Fehler bei Bearbeitung eines CPU-Auftrags</b>			
<b>Ereignisnummer</b>	<b>Ereignisnummer (dezimal)</b>	<b>Ereignis</b>	<b>Abhilfe</b>
(05) 02 <sub>H</sub>	2	Auftrag unzulässig in diesem Betriebszustand der seriellen Schnittstellenbaugruppe ET 200S Modbus/USS (Beispiel: Geräteschnittstelle ist nicht parametrierung).	Werten Sie den Diagnosealarm aus, und beheben Sie den Fehler entsprechend.
(05) 0E <sub>H</sub>	14	Ungültige Telegrammlänge	Das Sendetelegramm ist länger als 224 Byte. Der Sendeauftrag wurde von der Baugruppe ET 200S Modbus/USS abgebrochen. Wählen Sie eine kleinere Telegrammlänge.
(05) 30 <sub>H</sub>	48	Modbus-Master-Sendeauftrag abgelehnt, da die Antwort des Koppelpartners auf einen vorherigen lesenden Modbus-Master-Sendeauftrag noch nicht abgerufen wurde.	Nach einem erfolgreichen lesenden Modbus-Master-Sendeauftrag müssen Sie zuerst die Antwort des Koppelpartners aus dem Modul auslesen, bevor Sie einen neuen Modbus-Master-Sendeauftrag starten.
(05) 51 <sub>H</sub>	81	Rahmenablauffehler bei der Kommunikation zwischen der seriellen Schnittstellenbaugruppe ET 200S Modbus/USS und der CPU. Der Fehler ist beim Übertragen eines empfangenen Telegramms der seriellen Schnittstellenbaugruppe ET 200S SI in der CPU aufgetreten.	Die Baugruppe und die CPU haben die Übertragung abgebrochen. Wiederholen Sie den Empfangsauftrag. Die serielle Schnittstellenbaugruppe ET 200S Modbus/USS sendet die empfangene Meldung erneut.

Tabelle 3- 46 Ereignisklasse 8 (08 Hex): Empfangsfehler

Ereignisklasse 8 (08 Hex): Empfangsfehler			
Ereignisnummer	Ereignisnummer (dezimal)	Ereignis	Abhilfe
(08) 06 <sub>H</sub>	6	Zeichenverzugszeit überschritten. Zwei aufeinanderfolgende Zeichen wurden nicht innerhalb der Zeichenverzugszeit empfangen.	Partnergerät ist zu langsam oder gestört. Fehlverhalten des Partnergeräts ggf. mit Schnittstellentestgerät (FOXPG) untersuchen, das in die Übertragungsleitung eingeschaltet wird.
08 0A <sub>H</sub>	10	Überlauf des Empfangspuffers beim Master während des Empfangs des Antworttelegramms.	Prüfen Sie die Protokolleinstellungen des Slave.
(08) 0C <sub>H</sub>	12	Ein Übertragungsfehler (Paritätsfehler, Stoppbitfehler, Überlauffehler) wurde erkannt.	Störungen auf der Übertragungsleitung verursachen Telegrammwiederholungen und erniedrigen dadurch den Nutzdurchsatz. Die Gefahr eines nicht erkannten Fehlers steigt. Ändern Sie Ihren Systemaufbau bzw. die Leitungsverlegung. Überprüfen Sie die Verbindungsleitung der Kommunikationspartner bzw. überprüfen Sie, ob bei beiden Geräten Baudrate, Parität und Stoppbitanzahl gleich eingestellt sind.
(08) 0D <sub>H</sub>	13	BREAK: Empfangsleitung zum Partner ist unterbrochen.	Stellen Sie die Verbindung wieder her oder schalten Sie den Partner ein.
(08) 10 <sub>H</sub>	16	Paritätsfehler: Ist die SF-LED (rot) eingeschaltet, liegt eine Unterbrechung der Verbindungsleitung (Leitungsbreak) der beiden Kommunikationspartner vor.	Überprüfen Sie die Verbindungsleitung der Kommunikationspartner bzw. überprüfen Sie, ob bei beiden Geräten Baudrate, Parität und Stoppbitanzahl gleich eingestellt sind. Ändern Sie Ihren Systemaufbau bzw. die Leitungsverlegung.
(08) 11 <sub>H</sub>	17	Zeichenrahmenfehler: Ist die SF-LED (rot) eingeschaltet, liegt eine Unterbrechung der Verbindungsleitung (Leitungsbreak) der beiden Kommunikationspartner vor.	Überprüfen Sie die Verbindungsleitung der Kommunikationspartner bzw. überprüfen Sie, ob bei beiden Geräten Baudrate, Parität und Stoppbitanzahl gleich eingestellt sind. Ändern Sie Ihren Systemaufbau bzw. die Leitungsverlegung.
(08) 12 <sub>H</sub>	18	Nachdem die serielle Schnittstelle CTS auf OFF gestellt hat, wurden weitere Zeichen empfangen.	Parametrieren Sie den Kommunikationspartner neu oder entsorgen Sie die serielle Schnittstelle schneller.

Ereignisklasse 8 (08 Hex): Empfangsfehler			
Ereignisnummer	Ereignisnummer (dezimal)	Ereignis	Abhilfe
08 30 <sub>H</sub>	48	<p><b>Master:</b> Es wurde ein Anforderungstelegramm gesendet und die Antwortüberwachungszeit ist abgelaufen, ohne dass der Beginn eines Antworttelegramms erkannt wurde.</p> <p><b>Slave:</b> Broadcast bei diesem Funktionscode nicht zulässig.</p>	<p>Prüfen Sie, ob die Übertragungsleitung unterbrochen ist (evtl. Schnittstellenanalyse erforderlich).</p> <p>Prüfen Sie, dass in der Baugruppe und beim Kommunikationspartner die gleichen Einstellungen für die Protokollparameter Übertragungsgeschwindigkeit, Anzahl Datenbits, Parität und Anzahl Stoppbits vorhanden sind.</p> <p>Prüfen Sie, ob der Wert für die Antwortüberwachungszeit in PtP_PARAM groß genug ist.</p> <p>Prüfen Sie, ob die angegebene Slave-Adresse vorhanden ist.</p> <p>Das Modbus-Mastersystem darf Broadcast nur bei den dafür freigegebenen Funktionscodes verwenden.</p>
08 31 <sub>H</sub>	49	<p><b>Master:</b> Das erste Zeichen im Antworttelegramm des Slave unterscheidet sich von der Slave-Adresse, die im Anforderungstelegramm gesendet wurde (für den Normalbetrieb).</p> <p><b>Slave:</b> Empfangener Funktionscode nicht zulässig.</p>	<p>Der falsche Slave hat geantwortet.</p> <p>Prüfen Sie, ob die Übertragungsleitung unterbrochen ist (evtl. Schnittstellenanalyse erforderlich).</p> <p>Dieser Funktionscode kann für diesen Treiber nicht verwendet werden.</p>
08 32 <sub>H</sub>	50	Maximale Anzahl Bits oder Register überschritten oder Anzahl Bits kann nicht durch 16 dividiert werden, wenn auf die SIMATIC-Speicherbereiche Zeiten oder Zähler zugegriffen wird.	Begrenzen Sie die maximale Anzahl Bits auf 2040 und die maximale Anzahl Register auf 127. Zugriff auf SIMATIC-Zeiten/-Zähler nur in 16-Bit-Abständen.
08 33 <sub>H</sub>	51	Bit- oder Registeranzahl bei Funktionscode FC 15/16 und Telegrammelement byte_count passen nicht zusammen.	Korrigieren Sie die Bit-/Registeranzahl oder byte_count.
08 34 <sub>H</sub>	52	Unzulässige Bit-Codierung für "Bit setzen"/"Bit rücksetzen" erkannt.	Verwenden Sie für FC05 nur die Codierungen 0000Hex oder FF00Hex.
08 35 <sub>H</sub>	53	Unzulässiger Diagnose-Subcode (ungleich 0000Hex) bei Funktionscode FC 08 "Loop Back Test" erkannt.	Verwenden Sie für FC08 nur den Subcode 0000Hex.
08 36 <sub>H</sub>	54	Der intern gebildete Wert der CRC-16-Prüfsumme stimmt nicht mit der empfangenen CRC-Prüfsumme überein.	Überprüfen Sie die Bildung der CRC-Prüfsumme beim Modbus-Mastersystem.

Ereignisklasse 8 (08 Hex): Empfangsfehler			
Ereignisnummer	Ereignisnummer (dezimal)	Ereignis	Abhilfe
08 37 <sub>H</sub>	55	Telegrammablauffehler: Das Modbus-Mastersystem sendete ein neues Anforderungstelegramm ehe vom Treiber das letzte Antworttelegramm übertragen wurde.	Erhöhen Sie das Timeout auf das Slave-Antworttelegramm beim Modbus-Mastersystem.
08 50 <sub>H</sub>	80	Länge des Empfangstelegramms größer als 224 Byte bzw. größer als die parametrisierte Telegrammlänge.	Telegrammlänge des Partners anpassen.

Tabelle 3- 47 Ereignisklasse 14 (0E Hex) Allgemeine Verarbeitungsfehler <Parametrierung>

Ereignisklasse 14 (0E Hex) Allgemeine Verarbeitungsfehler <Parametrierung>			
Ereignisnummer	Ereignisnummer (dezimal)	Ereignis	Abhilfe
0E 20 <sub>H</sub>	32	Die Anzahl der Datenbits bei dieser Kopplung muss 8 betragen. Der Treiber ist nicht betriebsbereit.	Korrigieren Sie die Parametrierung des Treibers.
0E 21 <sub>H</sub>	33	Der parametrisierte Multiplikationsfaktor für die Zeichenverzugszeit ist nicht im Bereich von 1 bis 10. Der Treiber arbeitet mit der Standardeinstellung 1.	Korrigieren Sie die Parametrierung des Treibers.
0E 22 <sub>H</sub>	34	Die parametrisierte Betriebsart des Treibers ist unzulässig. Es ist "Normalbetrieb" oder "Störunterdrückungsbetrieb" anzugeben. Der Treiber ist nicht betriebsbereit.	Korrigieren Sie die Parametrierung des Treibers.
0E 23 <sub>H</sub>	35	<b>Master:</b> Es wurde ein unzulässiger Wert für die Antwortüberwachungszeit eingestellt: gültige Werte liegen zwischen 50 und 655000 ms. Der Treiber ist nicht betriebsbereit. <b>Slave:</b> Es wurde ein unzulässiger Wert für die Slave-Adresse eingestellt. Die Slave-Adresse 0 ist unzulässig. Der Treiber ist nicht betriebsbereit.	Korrigieren Sie die Parametrierung des Treibers.  Korrigieren Sie die Parametrierung des Treibers.
0E 2E <sub>H</sub>	46	Beim Lesen der Schnittstellenparameterdatei ist ein Fehler aufgetreten. Der Treiber ist nicht betriebsbereit.	Starten Sie den Master (Mains_ON) neu.

Tabelle 3- 48 Ereignisklasse 14 (0E Hex) Allgemeine Verarbeitungsfehler &lt;Verarbeitung eines Auftrags S\_SEND&gt;

<b>Ereignisklasse 14 (0E Hex) Allgemeine Verarbeitungsfehler &lt;Verarbeitung eines Auftrags S_SEND&gt;</b>			
<b>Ereignisnummer</b>	<b>Ereignisnummer (dezimal)</b>	<b>Ereignis</b>	<b>Abhilfe</b>
0E 40 <sub>H</sub>	64	Der für LEN bei S_SEND angegebene Wert ist zu klein.	Mindestlänge sind 2 Byte.
0E 41 <sub>H</sub>	65	Der für LEN bei S_SEND angegebene Wert ist zu klein. Eine größere Länge ist für den übertragenen Funktionscode erforderlich.	Die Mindestlänge für diesen Funktionscode beträgt 6 Byte.
0E 42 <sub>H</sub>	66	Der übertragene Funktionscode ist unzulässig.	Verwenden Sie nur zulässige Funktionscodes.
0E 43 <sub>H</sub>	67	Slave-Adresse 0 (= Broadcast) unzulässig bei diesem Funktionscode.	Verwenden Sie die Slave-Adresse 0 nur bei geeigneten Funktionscodes.
0E 44 <sub>H</sub>	68	Der Wert der übertragenen "Bitanzahl" liegt nicht im Bereich 1 bis 2040.	Die "Bitanzahl" muss sich im Bereich 1 bis 2040 befinden.
0E 45 <sub>H</sub>	69	Der Wert der übertragenen "Registeranzahl" liegt nicht im Bereich 1 bis 127.	Die "Registeranzahl" muss sich im Bereich 1 bis 127 befinden.
0E46 <sub>H</sub>	70	Funktionscode 15 oder 16: Die Werte der übertragenen "Bitanzahl" bzw. "Registeranzahl" liegen nicht im Bereich 1 bis 2040 bzw. 1 bis 127.	Die "Bitanzahl" bzw. "Registeranzahl" muss im Bereich 1 bis 2040 bzw. 1 bis 127 liegen.
0E 47 <sub>H</sub>	71	Funktionscode 15 oder 16: LEN für S_SEND entspricht nicht der übertragenen "Bitanzahl" bzw. "Registeranzahl".  LEN ist zu klein.	Erhöhen Sie LEN für SEND, bis eine ausreichende Menge Anwenderdaten an die Baugruppe übertragen wird. Eine größere Menge Benutzerdaten muss wegen der "Bitanzahl" bzw. "Registeranzahl" zur Baugruppe übertragen werden.
0E 48 <sub>H</sub>	72	Funktionscode 5: Der in Quell-DB SEND angegebene Code für "Bit setzen" (FF00H) bzw. "Bit löschen" (0000H) ist falsch.	Die einzigen zulässigen Codes sind "Bit setzen" (FF00H) "Bit löschen" oder 0000H.
0E 49 <sub>H</sub>	73	Funktionscode 8: Der in Quell-DB SEND angegebene Code für "Diagnosecode" ist falsch.	Der einzige zulässige Code ist "Diagnosecode" 0000H.
0E 4A <sub>H</sub>	74	Die Länge für diesen Funktionscode ist größer als die maximale Länge.	Im Handbuch finden Sie die maximalen Längenangaben zu jedem Funktionscode.

Tabelle 3- 49 Ereignisklasse 14 (0E Hex) Allgemeine Verarbeitungsfehler <Empfangsauswertung>

Ereignisklasse 14 (0E Hex) Allgemeine Verarbeitungsfehler <Empfangsauswertung>			
Ereignisnummer	Ereignisnummer (dezimal)	Ereignis	Anhilfe
0E 50 <sub>H</sub>	80	Master hat Antwort empfangen, ohne zu senden.	Ein Slave oder ein weiterer Master ist im Netz. Prüfen Sie, ob die Übertragungsleitung unterbrochen ist (evtl. Schnittstellenanalyse erforderlich).
0E 51 <sub>H</sub>	81	Funktionscode falsch: Der im Antworttelegramm empfangene Funktionscode unterscheidet sich vom gesendeten Funktionscode.	Prüfen Sie das Slave-Gerät.
0E 52 <sub>H</sub>	82	Byte-Unterlauf: Anzahl empfangener Zeichen ist kleiner als sich aus dem Byte-Zähler des Antworttelegramms ergeben würde bzw. ist kleiner als für diesen Funktionscode erwartet.	Prüfen Sie das Slave-Gerät.
0E 53 <sub>H</sub>	83	Byte-Überlauf: Anzahl empfangener Zeichen ist größer als sich aus dem Byte-Zähler des Antworttelegramms ergeben würde bzw. ist größer als für diesen Funktionscode erwartet.	Prüfen Sie das Slave-Gerät.
0E 54 <sub>H</sub>	84	Bytezähler falsch: Der im Antworttelegramm empfangene Bytezähler ist zu klein.	Prüfen Sie das Slave-Gerät.
0E 55 <sub>H</sub>	85	Bytezähler falsch: Der im Antworttelegramm empfangene Bytezähler ist falsch.	Prüfen Sie das Slave-Gerät.
0E 56 <sub>H</sub>	86	Echo falsch: Die vom Slave geechoteten Daten des Antworttelegramms (Bitanzahl, ...) unterscheiden sich von den im Anforderungstelegramm gesendeten Daten.	Prüfen Sie das Slave-Gerät.
0E 57 <sub>H</sub>	87	CRC-Prüfung falsch: Bei der Prüfung der CRC-16-Prüfsumme des Antworttelegramms vom Slave ist ein Fehler aufgetreten.	Prüfen Sie das Slave-Gerät.

Tabelle 3- 50 Ereignisklasse 14 (0E Hex) Allgemeine Verarbeitungsfehler <Empfang Ausnahmecodemeldung>

Ereignisklasse 14 (0E Hex) Allgemeine Verarbeitungsfehler <Empfang Ausnahmecodemeldung>			
Ereignisnummer	Ereignisnummer (dezimal)	Ereignis	Abhilfe
0E 61 <sub>H</sub>	97	Antworttelegramm mit Ausnahmecode 01: Unzulässige Funktion	Siehe Handbuch des Slave-Geräts.
0E 62 <sub>H</sub>	98	Antworttelegramm mit Ausnahmecode 02: Unzulässige Datenadresse	Siehe Handbuch des Slave-Geräts.

<b>Ereignisklasse 14 (0E Hex) Allgemeine Verarbeitungsfehler &lt;Empfang Ausnahmecodemeldung&gt;</b>			
<b>Ereignisnummer</b>	<b>Ereignisnummer (dezimal)</b>	<b>Ereignis</b>	<b>Abhilfe</b>
0E 63 <sub>H</sub>	99	Antworttelegramm mit Ausnahmecode 03: Unzulässiger Datenwert	Siehe Handbuch des Slave-Geräts.
0E 64 <sub>H</sub>	100	Antworttelegramm mit Ausnahmecode 04: Ausfall in zugehörigem Gerät	Siehe Handbuch des Slave-Geräts.
0E 65 <sub>H</sub>	101	Antworttelegramm mit Ausnahmecode 05: Quittierung	Siehe Handbuch des Slave-Geräts.
0E 66 <sub>H</sub>	102	Antworttelegramm mit Ausnahmecode 06: Belegt, Telegramm zurückgewiesen	Siehe Handbuch des Slave-Geräts.
0E 67 <sub>H</sub>	103	Antworttelegramm mit Ausnahmecode 07: Negative Quittierung	Siehe Handbuch des Slave-Geräts.

Tabelle 3- 51 Ereignisklasse 30 (1E Hex): Fehler während der Kommunikation zwischen serieller Schnittstelle und CPU

<b>Ereignisklasse 30 (1E Hex): Fehler während der Kommunikation zwischen serieller Schnittstelle und CPU</b>			
<b>Ereignisnummer</b>	<b>Ereignisnummer (dezimal)</b>	<b>Ereignis</b>	<b>Abhilfe</b>
(1E) 0D <sub>H</sub>	13	"Auftragsabbruch wegen Neustart, Wiederanlauf oder Reset"	
(1E) 0E <sub>H</sub>	14	Statischer Fehler bei Aufruf der SFC DP_RDDAT. Der Returnwert RET_VAL der SFC wird Ihnen in der Variablen SFCERR im Instanz-DB zur Auswertung zur Verfügung gestellt.	Laden Sie die Variable SFCERR aus dem Instanz-DB.
(1E) 0F <sub>H</sub>	15	Statischer Fehler bei Aufruf der SFC DP_WRDAT. Der Returnwert RET_VAL der SFC wird Ihnen in der Variablen SFCERR im Instanz-DB zur Auswertung zur Verfügung gestellt.	Laden Sie die Variable SFCERR aus dem Instanz-DB.
(1E) 10 <sub>H</sub>	16	Statischer Fehler bei Aufruf der SFC RD_LGADR. Der Returnwert RET_VAL der SFC wird Ihnen in der Variablen SFCERR im Instanz-DB zur Auswertung zur Verfügung gestellt.	Laden Sie die Variable SFCERR aus dem Instanz-DB.
(1E) 11 <sub>H</sub>	17	Statischer Fehler bei Aufruf der SFC RDSYSST. Der Returnwert RET_VAL der SFC wird Ihnen in der Variablen SFCERR im Instanz-DB zur Auswertung zur Verfügung gestellt.	Laden Sie die Variable SFCERR aus dem Instanz-DB.
(1E) 20 <sub>H</sub>	32	Parameter außerhalb des Bereichs.	Geben Sie für den Funktionsbaustein einen Parameter im zulässigen Bereich ein.
(1E) 41 <sub>H</sub>	65	Anzahl der am Parameter LEN der FBs angegebenen Bytes unzulässig	Halten Sie den Wertebereich von 1 bis 256 Byte ein.

**Auswerten der Variable SFCERR**

Nähere Informationen zu den aufgetretenen Fehlern (1E) 0E<sub>H</sub>, (1E) 0F<sub>H</sub>, (1E) 10<sub>H</sub> und (1E) 11<sub>H</sub> der Ereignisklasse 30 erhalten Sie über die Variable SFCERR.

Die Variable SFCERR können Sie aus dem Instanz-DB des entsprechenden Funktionsbausteins laden.

Die Fehlermeldungen, die in der Variablen SFCERR eingetragen werden, finden Sie bei den Systemfunktionen "DPRD\_DAT" und SFC15 "DPWR\_DAT" und RD\_LGADR im Referenzhandbuch *Systemsoftware für S7 300/400, System- und Standardfunktionen*.

**3.7.4 PROFIBUS-Slave-Diagnose**

**Einleitung**

Die Slave-Diagnose verhält sich nach der Norm EN 50170, Volume 2, PROFIBUS. Sie kann in Abhängigkeit vom DP-Master für alle DP-Slaves, die sich nach Norm verhalten, mit STEP 5 oder STEP 7 ausgelesen werden.

Die PROFIBUS-Slave-Diagnose umfasst die Baugruppendiagnose, den Baugruppenstatus und die kanalbezogene Diagnose. Ausführliche Informationen zur DP-Slave-Diagnose finden Sie im Handbuch *Dezentrales Peripheriesystem ET 200S, 6ES7 151-1AA10-8AA0*.

**Kanalbezogene Diagnose**

Die kanalbezogene Diagnose gibt Auskunft über Kanalfehler von Baugruppen und beginnt nach dem Baugruppenstatus. Die Tabelle führt die kanalbezogenen Fehlertypen auf.

Tabelle 3- 52 Kanalbezogene Fehlertypen der seriellen Schnittstellenbaugruppe ET 200S Modbus/USS

Ereignis (Fehlertyp)	Beschreibung	Empfohlene Maßnahme
00110: Drahtbruch	Draht gebrochen oder gelöst.	Überprüfen Sie die Verdrahtung zu den Klemmen. Überprüfen Sie das Kabel zum Partner.
00111: Überlauf	Pufferüberlauf; Meldungslängenüberlauf	Der FB S_RCV muss häufiger aufgerufen werden.
01000: Unterlauf	Meldung mit Länge 0 gesendet.	Überprüfen Sie, warum der Kommunikationspartner Telegramme ohne Nutzdaten sendet.
01001: Fehler	Interner Baugruppenfehler aufgetreten.	Tauschen Sie die Baugruppe aus.
10000: Parametrierungsfehler	Baugruppe ist nicht parametriert.	Korrigieren Sie die Parametrierung.
10110: Meldungsfehler	Rahmenfehler; Paritätsfehler	Prüfen Sie die Kommunikationseinstellungen.



### 3.7.5 Diagnosefunktionen Modbus-Slave

#### ERROR\_NR und ERROR\_INFO

Der Modbus-Kommunikations-FB hat folgende zwei Ausgangsparameter, mit denen aufgetretene Fehler angezeigt werden:

- Parameter ERROR\_NR
- Parameter ERROR\_INFO

Aufgetretene Fehler werden am Ausgang ERROR\_NR angezeigt. Am Ausgang ERROR\_INFO wird eine zusätzliche Detail-Information zu dem Fehler in ERROR\_NR ausgegeben.

#### Löschen der Fehler

Die Fehler werden bei einer steigenden Flanke an START gelöscht. Außerdem können die Fehleranzeigen durch den Anwender zu einem beliebigen Zeitpunkt gelöscht werden, wenn dies erforderlich ist.

#### FB-Fehlercodes

Die Fehlercodes 1 bis 99 haben folgende Bedeutung:

- **ERROR\_No 1 bis 9**

##### **Fehler bei der Initialisierung FB und CP**

Bei den Fehlernummern 1...9 wurde die Initialisierung mit Fehler beendet. Der Parameter START\_ERROR ist 1.

Es ist keine MODBUS-Kommunikation zum Mastersystem möglich.

- **ERROR\_No 10 bis 19**

##### **Fehler bei der Verarbeitung eines Funktionscodes**

Bei den Fehlernummern 10...19 ist bei der Verarbeitung eines Funktionscodes ein Fehler aufgetreten. Die Baugruppe hat dem Kommunikations-FB einen unzulässigen Bearbeitungsauftrag gesendet.

Der Fehler wird ebenfalls an den Treiber gemeldet.

Nachfolgende Bearbeitungsaufträge werden weiter bearbeitet.

- **ERROR\_No 90 bis 99**

##### **Sonstige Fehler**

Es ist ein Verarbeitungsfehler aufgetreten.

Der Fehler wird nicht an den Treiber gemeldet.

Nachfolgende Bearbeitungsaufträge werden weiter bearbeitet.

### 3.7.6 Fehler

#### Liste der Fehlernummern

Tabelle 3- 53 Fehler während der Initialisierung

Fehlernummer (dezimal)	ERROR_INFO	Ereignis	Abhilfe
0	0	Kein Fehler.	
1	SFC51->RET_VAL	Fehler beim Lesen der SZL mit der SFC51.	RET_VAL in ERROR_INFO analysieren, Ursache beseitigen.
2	S_SEND->STATUS, S_RCV->STATUS	Timeout bei Baugruppeninitialisierung oder Fehler bei Baugruppeninitialisierung (Fehler in Auftrag S_SEND).	Prüfen Sie, ob für diese Schnittstelle als Protokoll "MODBUS Slave" parametrierung wurde. Prüfen Sie, ob die im Kommunikations-FB angegebene "ID" korrekt ist. Analysieren Sie ERROR_INFO.

Tabelle 3- 54 Fehler bei der Verarbeitung eines Funktionscodes

Fehlernummer (dezimal)	ERROR_INFO	Ereignis	Abhilfe
11	Startadresse	Unzulässige Startadresse vom Treiber an den Kommunikations-FB übergeben.	Prüfen Sie die Modbus-Adresse des Modbus-Mastersystems.
12	Registeranzahl	Unzulässige Registeranzahl vom Treiber an den Kommunikations-FB übergeben. Registeranzahl = 0.	Prüfen Sie die Registeranzahl des Modbus-Mastersystems, starten Sie ggf. die Baugruppe neu (Mains_ON).
13	Registeranzahl	Unzulässige Registeranzahl vom Treiber an den Kommunikations-FB übergeben: Registeranzahl > 128.	Prüfen Sie die Registeranzahl des Modbus-Mastersystems, starten Sie ggf. die Baugruppe neu (Mains_ON).
14	Merker M – Endadresse	Zugriffsversuch auf den SIMATIC-Speicherbereich der Merker über das Bereichsende hinaus. Achtung: Die Bereichslänge in der SIMATIC-CPU ist vom Typ der CPU abhängig.	Verringern Sie die Modbus-Startadresse bzw. die Zugriffslänge im Modbus-Mastersystem.
15	Ausgänge A – Endadresse Eingänge E – Endadresse	Zugriffsversuch auf den SIMATIC-Speicherbereich der Ausgänge über das Bereichsende hinaus. Achtung: Die Bereichslänge in der SIMATIC-CPU ist vom Typ der CPU abhängig.	Verringern Sie die Modbus-Startadresse bzw. die Zugriffslänge im Modbus-Mastersystem.

Fehlernummer (dezimal)	ERROR_INFO	Ereignis	Abhilfe
16	Zeiten T – Endadresse	Zugriffsversuch auf den SIMATIC-Speicherbereich der Zeiten über das Bereichsende hinaus. Achtung: Die Bereichslänge in der SIMATIC-CPU ist vom Typ der CPU abhängig.	Verringern Sie die Modbus-Startadresse bzw. die Zugriffslänge im Modbus-Mastersystem.
17	Zähler Z – Endadresse	Zugriffsversuch auf den SIMATIC-Speicherbereich der Zähler über das Bereichsende hinaus. Achtung: Die Bereichslänge in der SIMATIC-CPU ist vom Typ der CPU abhängig.	Verringern Sie die Modbus-Startadresse bzw. die Zugriffslänge im Modbus-Mastersystem.
18	0	Unzulässiger SIMATIC-Speicherbereich vom Treiber an den Kommunikations-FB übergeben.	Starten Sie ggf. die Baugruppe neu (Mains_ON).
19		Fehler beim Zugriff auf die SIMATIC-Peripherie.	Prüfen Sie, ob die benötigte Peripherie vorhanden und fehlerfrei ist.
20	DB#	DB ist nicht vorhanden.	Nehmen Sie den DB in Ihr Projekt auf.
21	DB#	DB-Länge ungültig	Erhöhen Sie die DB-Länge.
22	DB#	DB# liegt unter dem DB-Mindestwert.	Ändern Sie den DB-Mindestwert.
23	DB#	DB# liegt über dem DB-Maximalwert.	Ändern Sie den DB-Maximalwert.
24	Merkeradresse	Merker unterhalb der Mindestgrenze.	Ändern Sie die Mindestgrenzen der Merker in Umsetzungs-DB.
25	Merkeradresse	Merker oberhalb der Maximalgrenze.	Ändern Sie die Höchstgrenzen der Merker in Umsetzungs-DB.
26	Ausgangsadresse	Ausgang unterhalb der Mindestgrenze.	Ändern Sie die Mindestgrenzen der Ausgänge in Umsetzungs-DB.
27	Ausgangsadresse	Ausgang oberhalb der Maximalgrenze.	Ändern Sie die Höchstgrenzen der Ausgänge in Umsetzungs-DB.

Tabelle 3- 55 Sonstige Fehler

Fehlernummer (dezimal)	ERROR_INFO	Ereignis	Abhilfe
90	S_SEND-> STATUS	Fehler bei der Übertragung eines Quittierungs-telegramms an den Treiber mit S_SEND.	Analysieren Sie die STATUS-Informationen.
94	S_RCV->STATUS	Fehler beim Lesen von SYSTAT mit S_RCV (STATUS).	Analysieren Sie die STATUS-Informationen.

## 3.8 USS-Master

### 3.8.1 Was ist der USS-Master?

#### Einleitung

Mit dem USS-Protokoll kann der Anwender die serielle Buskommunikation zwischen der Baugruppe ET 200S Modbus/USS als Master und mehreren Slave-Systemen aufbauen. Antriebe von Siemens können als Slaves am USS-Bus betrieben werden.

#### Merkmale des USS-Protokolls

Das USS-Protokoll verfügt über die folgenden wesentlichen Merkmale:

- Unterstützung der mehrpunktfähigen RS485-Kopplung
- Master-Slave-Zugriffstechnik
- System mit einem Master
- Maximal 32 Teilnehmer (max. 31 Slaves)
- Betrieb mit variablen oder festen Telegrammlängen
- Einfache, zuverlässige Telegramme
- Der gleiche Busbetrieb wie für PROFIBUS (DIN 19245 Teil 1)
- Datenschnittstelle zum Basis-Antriebswandler gemäß den PROFIL-Antrieben mit unterschiedlichen Geschwindigkeiten. D.h. bei Einsatz der USS werden die Informationen wie bei PROFIBUS-DP an den Antrieb übertragen.
- Einsatz für Anlauf, Wartung und Automation

## 3.8.2 USS-Protokoll

### Einleitung

Das USS-Protokoll ist ein einfaches serielles Datenübertragungsprotokoll, das auf die Anforderungen in der Antriebstechnologie zugeschnitten ist.

Das USS-Protokoll definiert eine Zugriffstechnik entsprechend dem Master-Slave-Prinzip für die Kommunikation über einen seriellen Bus. An den Bus können ein Master und bis zu 31 Slaves angeschlossen werden. Die einzelnen Slaves werden vom Master mit Hilfe eines Adresszeichens im Telegramm gewählt. Ein Slave kann nie senden, ohne zuvor vom Master angestoßen worden zu sein. Es ist also keine direkte Datenübertragung zwischen einzelnen Slaves möglich. Die Kommunikation arbeitet im Halbduplexbetrieb. Die Master-Funktion kann nicht übertragen werden. Das USS-System besitzt nur einen Master.

### Telegrammaufbau

Jedes Telegramm beginnt mit dem Startzeichen (STX), gefolgt von der Längenangabe (LGE) und dem Adressbyte (ADR). Danach folgt das Datenfeld. Das Telegramm endet mit dem Blockprüfzeichen (BCC).

STX	LGE	ADR	1	2	...	N	BCC
-----	-----	-----	---	---	-----	---	-----

Bei Daten im Netzdatenbaustein, die aus einem Wort (16 Bit) bestehen, wird zuerst das High-Byte, gefolgt vom Low-Byte gesendet. Bei Doppelwortdaten wird dementsprechend zuerst das High-Word und anschließend das Low-Word gesendet.

Das Protokoll identifiziert keine Tasks in den Datenfeldern.

### Datenverschlüsselung

Die Daten werden folgendermaßen verschlüsselt:

- STX: 1 Byte, Textbeginn, 02H
- LGE: 1 Byte, enthält die Telegrammlänge als Binärzahl
- ADR: 1 Byte, enthält die Slave-Adresse und den Telegrammtyp, binärcodiert
- Datenfelder: Je ein Byte, Inhalt sind taskabhängig
- BCC: 1 Byte, Blockprüfzeichen

### Datenübertragungsprozedur

Der Master sorgt für die zyklische Datenübertragung in Telegrammen. Der Master adressiert alle Slave-Teilnehmer nacheinander mit einem Task-Telegramm. Die angesprochenen Teilnehmer antworten mit einem Antworttelegramm. Entsprechend der Master-Slave-Prozedur muss der Slave das Antworttelegramm an den Master senden, nachdem er das Task-Telegramm empfangen hat. Erst dann kann der Master den nächsten Slave ansprechen.

### Allgemeiner Aufbau des Netzdatenbausteins

Der Netzdatenbaustein ist in zwei Bereiche unterteilt: Parameter (PKW) und Prozessdaten (PZD).

STX	LGE	ADR	Parameter (PKW)	Prozessdaten (PZD)	BCC
-----	-----	-----	-----------------	--------------------	-----

- **Parameterbereich (PKW)**

Der PKW-Bereich wickelt die Parameterübertragung zwischen zwei Kommunikationspartnern ab (z.B. Steuerung und Antrieb). Dies umfasst beispielsweise das Lesen und Schreiben von Parameterwerten und das Lesen von Parameterbeschreibungen und zugehörigem Text. Die PKW-Schnittstelle beinhaltet im allgemeinen Tasks für Bedienung und Anzeigen, Wartung und Diagnose.

- **Prozessdatenbereich (PZD)**

Der PZD-Bereich umfasst Signale, die für die Automatisierung erforderlich sind:

- Steuerworte und Sollwerte vom Master an den Slave
- Statusworte und Istwerte vom Slave an den Master

Die Inhalte des Parameterbereichs und des Prozessdatenbereichs werden von den Slave-Antrieben definiert. Weitere Informationen hierzu finden Sie in der Dokumentation des Antriebs.

### 3.8.3 Konfiguration und Parametrierung

#### Konfiguration und Parametrierung

Tabelle 3- 56 Parameter für den USS-Master

Parameter	Beschreibung	Wertebereich	Defaultwert
Diagnosealarm	Geben Sie an, ob die Baugruppe einen Diagnosealarm erzeugt, wenn ein schwerer Fehler auftritt.	<ul style="list-style-type: none"> <li>• nein</li> <li>• ja</li> </ul>	nein
BREAK Erkennung aktivieren	Bei Leitungsbruch bzw. wenn kein Schnittstellenkabel angeschlossen ist erzeugt die Baugruppe die Fehlermeldung "Break".	<ul style="list-style-type: none"> <li>• Nein</li> <li>• Ja</li> </ul>	Nein
Art der Schnittstelle	Geben Sie die zu verwendende elektrische Schnittstelle an.	<ul style="list-style-type: none"> <li>• RS-232</li> <li>• RS-485 (halbduplex)</li> </ul>	RS-485 (halbduplex)
Halbduplexe Vorbelegung der Empfangsleitung	Geben Sie die Vorbelegung der Empfangsleitung in der Betriebsart RS-485 an. Nicht in der Betriebsart RS-232C.  Die Einstellung "Invertierte Pegel" wird nur im Ersatzteifall zur Sicherstellung der Kompatibilität benötigt.	R(A) 5V / R(B) 0V R(A) 0V / R(B) 5V Invertierter Pegel Keine	R(A) 0V / R(B) 5V
Baudrate	Wählen Sie die Geschwindigkeit der Datenübertragung in Bits pro Sekunde.	<ul style="list-style-type: none"> <li>• 110</li> <li>• 300</li> <li>• 600</li> <li>• 1200</li> <li>• 2400</li> <li>• 4800</li> <li>• 9600</li> <li>• 19200</li> <li>• 38400</li> <li>• 57600</li> <li>• 76800</li> <li>• 115200</li> </ul>	9600

---

#### Hinweis

Beachten Sie auch die Themen zu Identifikationsdaten (Seite 61) und Nachladen von Firmware-Updates (Seite 63))

---

### 3.8.4 Funktionsübersicht

#### Ablauf der Netzdatenübertragung

Die Bausteine wickeln die Netzdatenübertragung zyklisch mit bis zu 31 Antriebs-Slaves entsprechend der in der Polling-Liste (Parametrierungs-DB) angegebenen Reihenfolge ab. Es ist pro Slave jeweils nur ein Auftrag aktiv. Die Netzdaten für jeden Slave werden vom Anwender in einem Datenbaustein (Netzdatenbaustein) abgelegt und von dort abgerufen. Sie werden entsprechend der Programmdefinition in der Polling-Liste über einen anderen Datenbereich (Kommunikationsprozessor-DB) zum Kommunikationsprozessor übertragen und von dort abgerufen.

Zwei Funktionsaufrufe sind für diese Prozedur erforderlich (ein Sende- und ein Empfangsbaustein). Eine weitere Funktion unterstützt das Erzeugen und Vorbesetzen der für die Kommunikation erforderlichen Datenbausteine.

#### Leistungsmerkmale:

- Erstellung von Datenbereichen für die Kommunikation je nach Buskonfiguration
- Vorbesetzung der Polling-Liste
- Telegrammaufbau entsprechend der USS-Spezifikation
- Netzdatenaustausch kann entsprechend der erforderlichen Netzdatenstruktur parametrisiert werden
- Ausführung und Überwachung von PKW-Aufträgen
- Abwicklung von Parameteränderungsberichten
- Überwachung des Gesamtsystems und Fehlerbehebung

Verschiedene Netzdatenstrukturen können genutzt werden, um Netzdaten zu senden.

Je nach der gewählten Struktur verfügen die Netzdaten über einen PZD-Bereich für die Prozessdaten und einen PKW-Bereich für die Parameterverarbeitung.

Im PKW-Bereich kann der Master Parameterwerte lesen und schreiben und der Slave kann Parameteränderungen über Parameteränderungsberichte anzeigen.

Der PZD-Bereich enthält für die Prozesssteuerung erforderliche Signale, z.B. Steuerworte und Sollwert vom Master an den Slave und Statusworte und Istwerte vom Slave an den Master.

Die ordnungsgemäße Reihenfolge von Funktionsaufrufen ist: S\_USST, S\_SEND, S\_RCV, S\_USSR. Dies ist wichtig, weil die Ausgänge der Funktionen S\_SEND und S\_RCV nur im aktuellen Zyklus des Automatisierungssystems gültig sind.

Das folgende Bild zeigt den Datenverkehr zwischen dem Anwenderprogramm und dem USS-Slave.



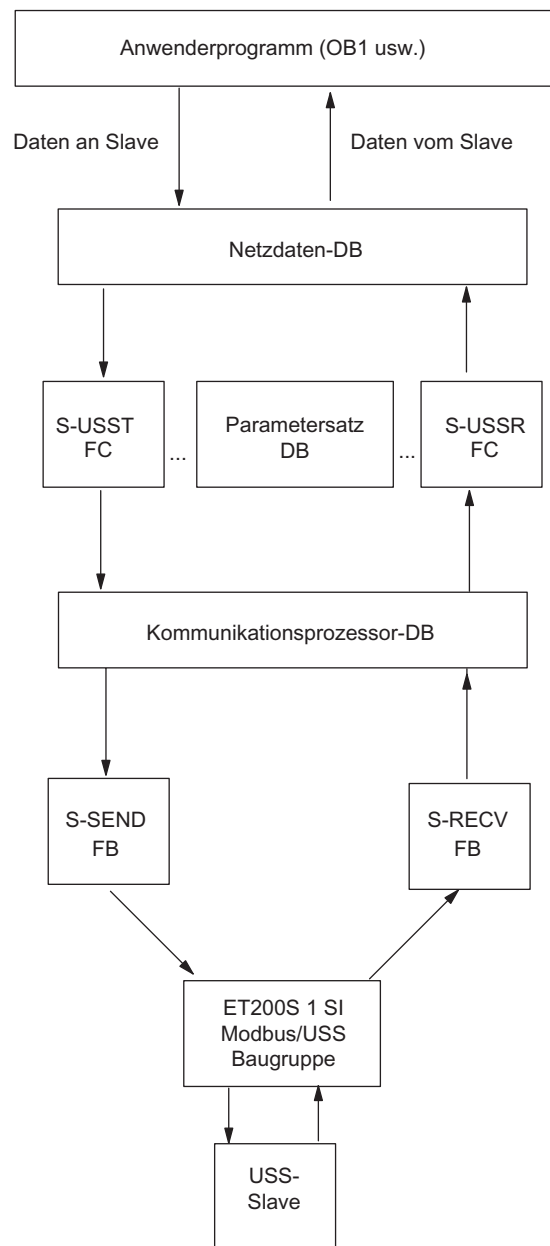


Bild 3-21 Datenverkehr zwischen dem Anwenderprogramm und dem USS-Slave

### 3.8.5 FC17 S\_USST: Daten an einen Slave senden

#### Beschreibung

Die FC S\_USST wickelt die Übertragung der Netzdaten (PZD- und evtl. PKW-Daten) an die Slaves je nach der verwendeten Netzdatenstruktur ab.

Die FC nimmt die Parametrierung des aktuellen Slaves aus der Polling-Liste (Parametrierungs-DB) und sendet die Daten aus dem Netzdaten-DB. Sie wertet das Kommunikationssteuerwort des aktuellen Slaves aus (Anstoß eines PKW-Auftrags bzw. Quittierung eines Parameteränderungsberichts), vervollständigt die USS-Sendedaten und überträgt sie an den Sendepuffer des Kommunikationsprozessor-DB. Schließlich stößt sie die Übertragung der Netzdaten an den Slave mittels FB S\_SEND an.

Erkennt die Funktion einen Parametrierungsfehler im Parametrierungs-DB, dann wird ein Fehlersignal in Byte Pafe 2 des Netzdaten-DB gespeichert.

Die FC17 wird einmal pro Zyklus des Automatisierungssystems aufgerufen.

### Programmaufbau von S\_USST

Die folgende Abbildung zeigt den Programmaufbau von S\_USST.

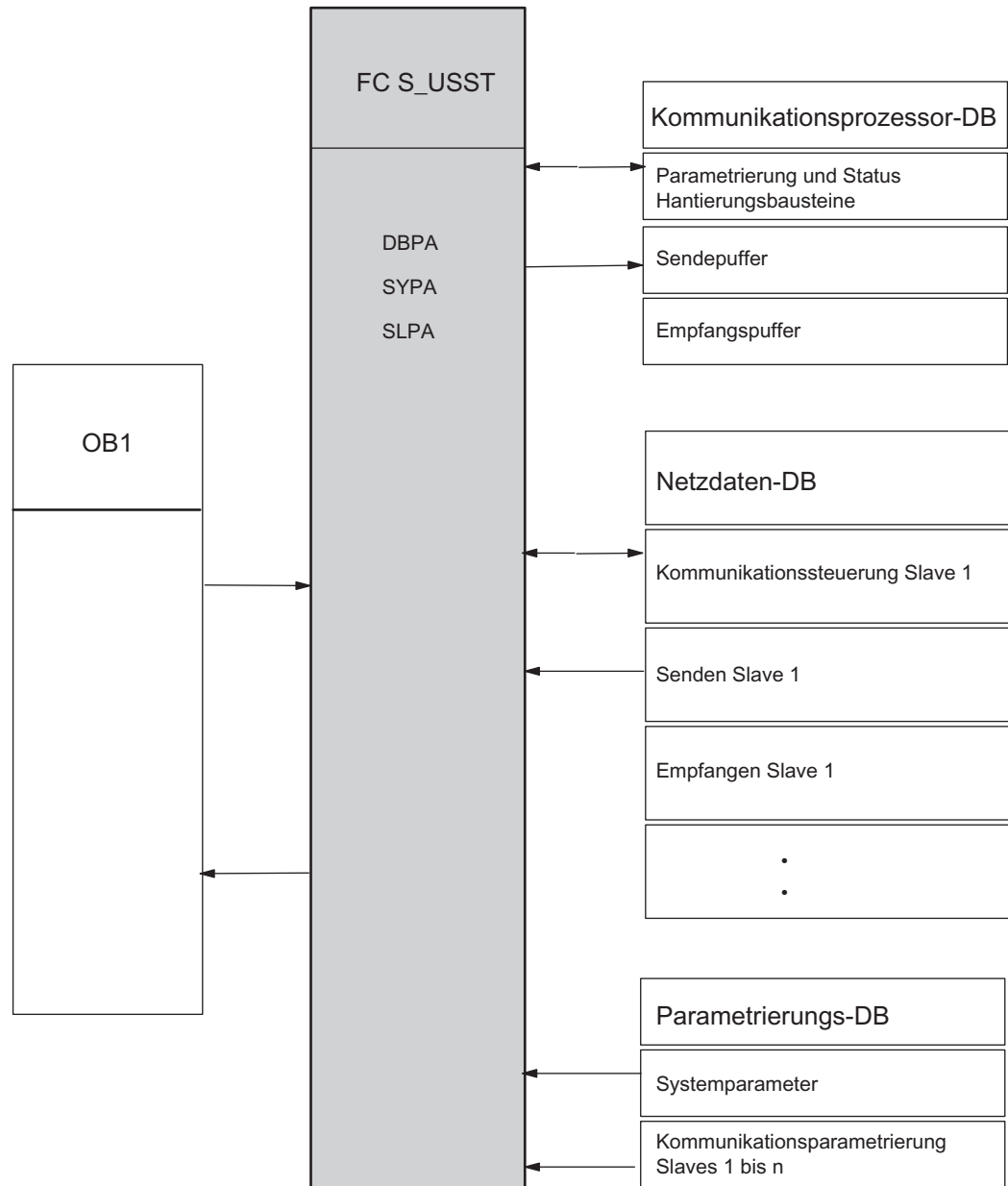
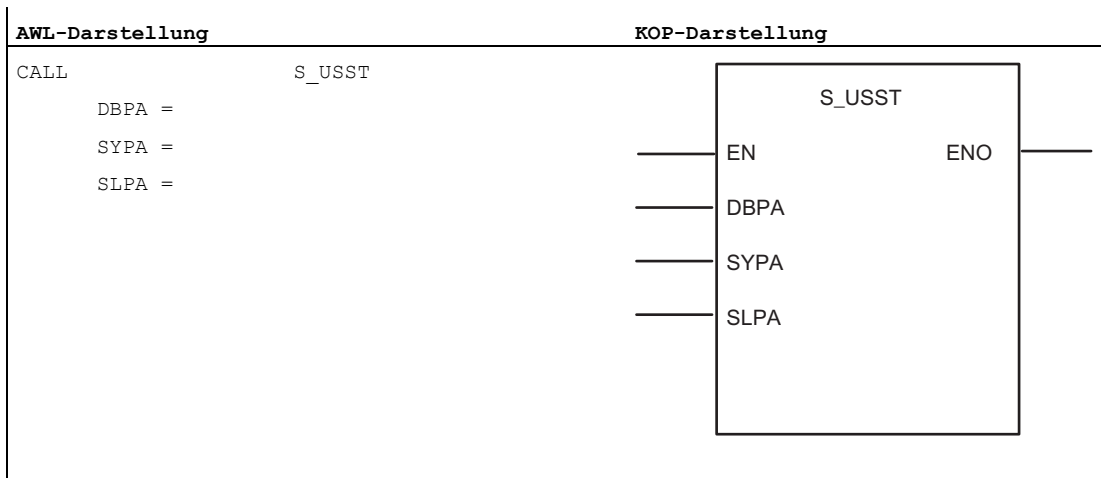


Bild 3-22 Diagnosefunktionen Modbus-Slave

Tabelle 3- 57 AWL- und KOP-Darstellungen



**Hinweis**

Die Parameter EN und ENO sind nur in der grafischen Darstellung vorhanden (bei KOP oder FUP). Für die Bearbeitung dieser Parameter wird vom Compiler das Binärergebnis BIE verwendet.

Das Binärergebnis BIE wird auf Signalzustand "1" gesetzt, wenn der Baustein fehlerfrei beendet wurde. Liegt ein Fehler vor, dann wird das Binärergebnis BIE auf "0" gesetzt.

**Parameter FC17 S\_USST**

Die Tabelle führt die Parameter von FC S\_USST auf.

Tabelle 3- 58 Parameter FC S\_USST

Name	Art	Datentyp	Beschreibung	Kommentar
DBPA	INPUT	INT	Bausteinnummer des Parametrierungs-DB	CPU-spezifisch (Null nicht zulässig)
SYPA	INPUT	INT	Anfangsadresse der Systemparameter im Parametrierungs-DB	0 <= SYPA <= 8174
SLPA	INPUT	INT	Anfangsadresse der Slave-Parameter im Parametrierungs-DB	0 <= SLPA <= 8184

### 3.8.6 FC18 S\_USSR: Daten von einem Slave empfangen

#### Beschreibung

Die FC S\_USSR wickelt den Empfang der Netzdaten (PZD- und evtl. PKW-Daten) von den Slaves je nach der verwendeten Netzdatenstruktur ab.

Die FC nimmt die Parametrierung des aktuellen Slaves aus der Polling-Liste (Parametrierungs-DB) und wertet das Statuswort des Bausteins TRANSMIT aus.

Wurde der aktuelle Auftrag fehlerfrei beendet (Bit 9 = 0 im Kommunikations-Statuswort des Netzdaten-DB), werden die eingehenden Daten vom Empfangspuffer des Kommunikationsprozessor-DB in den Netzdaten-DB übertragen und ausgewertet. Anschließend wird das Kommunikations-Statuswort im Netzdaten-DB aktualisiert.

Wurde der aktuelle Auftrag nicht fehlerfrei beendet (Bit 9 = 1 im Kommunikations-Statuswort des Netzdaten-DB), werden die Daten des aktuellen Slaves nicht vom Empfangspuffer des Kommunikationsprozessor-DB übernommen. Die FC18 gibt dies im Kommunikations-Statuswort des Netzdaten-DB an und trägt die Fehlerursache ins Kommunikations-Fehlerwort ein.

Erkennt der Baustein einen Parametrierungsfehler im Parametrierungs-DB, dann wird ein Fehlersignal in Byte Pafe 1 des Netzdaten-DB gespeichert.

Die FC18 wird einmal pro Zyklus des Automatisierungssystems aufgerufen.

### Programmaufbau von S\_USSR

Die folgende Abbildung zeigt den Programmaufbau von S\_USSR.

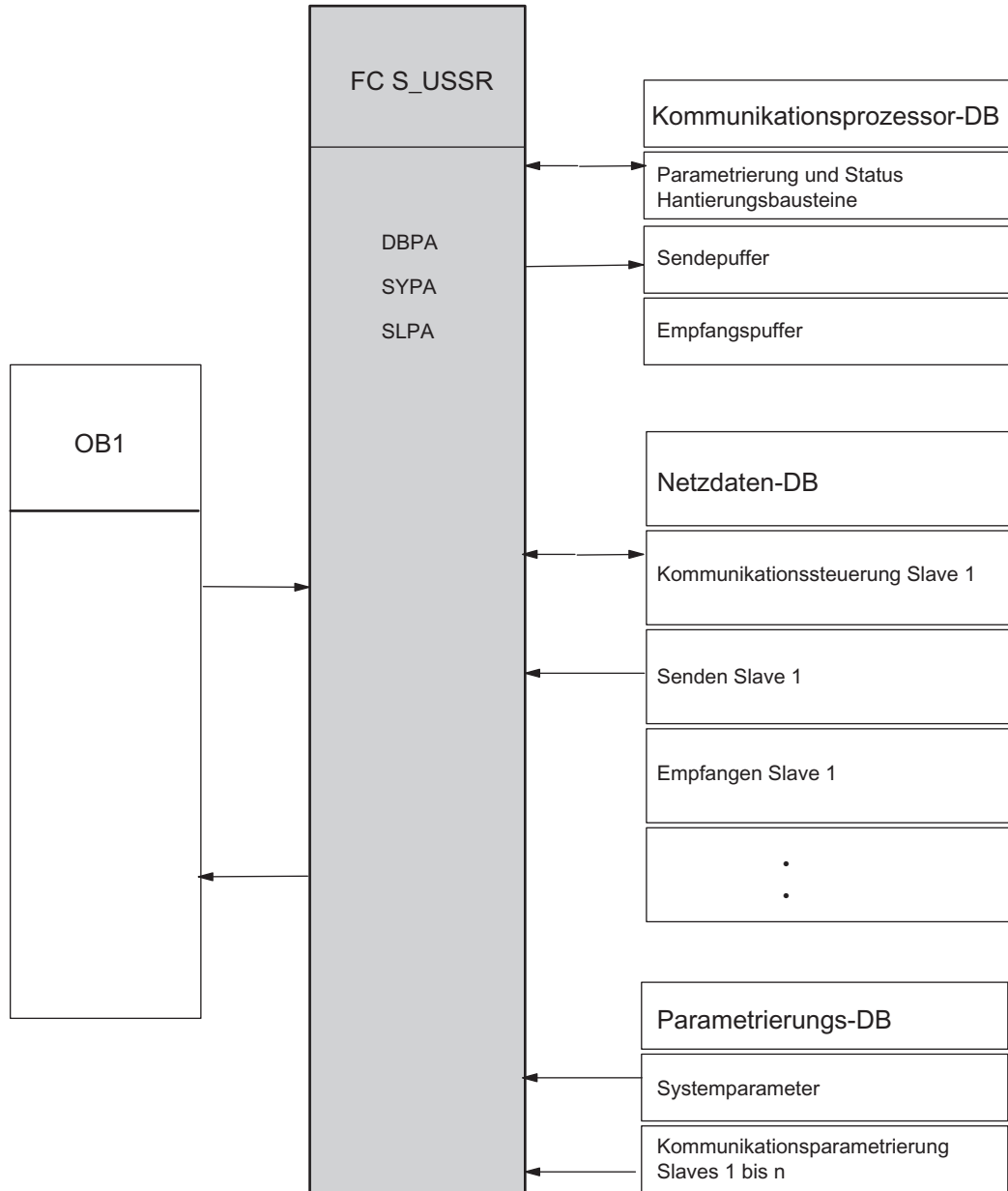
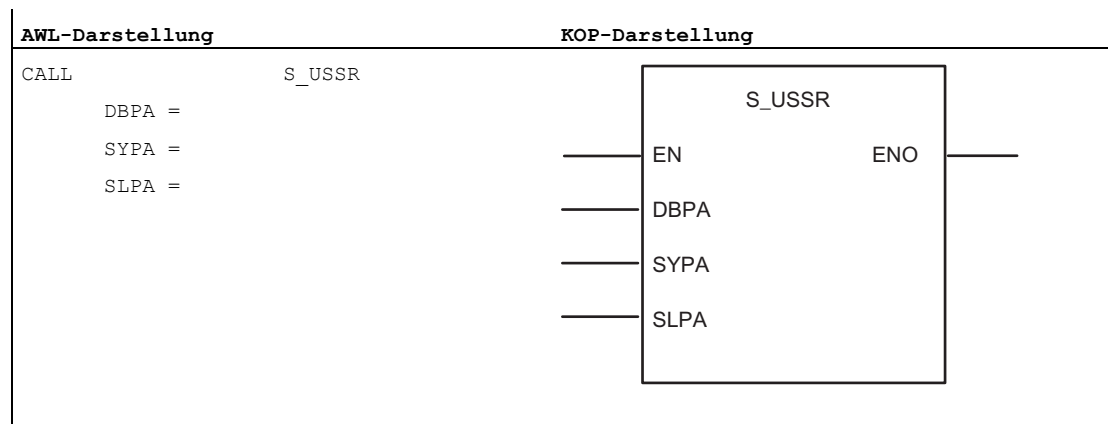


Bild 3-23 Programmaufbau von S\_USSR

Tabelle 3- 59 AWL- und KOP-Darstellungen



**Hinweis**

Die Parameter EN und ENO sind nur in der grafischen Darstellung vorhanden (bei KOP oder FUP). Für die Bearbeitung dieser Parameter wird vom Compiler das Binäresultat BIE verwendet.

Das Binäresultat BIE wird auf Signalzustand "1" gesetzt, wenn der Baustein fehlerfrei beendet wurde. Liegt ein Fehler vor, wird das Binäresultat BIE auf "0" gesetzt.

**Parameter FC18 S\_USSR**

Die Tabelle führt die Parameter von FC S\_USSR auf.

Tabelle 3- 60 Parameter FC S\_USSR

Name	Art	Datentyp	Beschreibung	Kommentar
DBPA	INPUT	INT	Bausteinnummer des Parametrierungs-DB	CPU-spezifisch (Null nicht zulässig)
SYPA	INPUT	INT	Anfangsadresse der Systemparameter im Parametrierungs-DB	0 <= SYPA <= 8174
SLPA	INPUT	INT	Anfangsadresse der Slave-Parameter im Parametrierungs-DB	0 <= SLPA <= 8184

Die Parameter der FC U\_USST entsprechen den Parametern der FC S\_USSR. Die beiden Funktionen greifen auf die gleiche Parametrierung (System- und Slave-Parameter) im Parametrierungs-DB zu und müssen deshalb identisch parametrierung werden.

### 3.8.7 FC19 S\_USSI: Initialisierung

#### Beschreibung

Die FC S\_USSI ist eine optionale Funktion.

Wird diese FC bei Anlauf des S7-Systems aufgerufen, werden die Datenbausteine Kommunikationsprozessor-DB, Netzdaten-DB und Parametrierungs-DB, die für die Kommunikation erforderlich sind, generiert. Der DBPA wird zusätzlich vorbesetzt. Die FC S\_USSI eignet sich für die Generierung und Vorbesetzung der genannten Datenbereich nur bei gleicher Netzdatenstruktur für alle Slaves.

Beim Aufruf prüft die FC zunächst die Plausibilität ihrer Parametrierung für Slaveanzahl, Netzdatenstruktur, Start-Teilnehmernummer und PKW-Wiederholungen. Erkennt der Baustein dabei einen Fehler, werden Generierung und Vorbesetzung der Datenbausteine nicht ausgeführt. Die CPU geht in den Betriebszustand STOP und der Anwender erhält eine Fehlermeldung über das Fehlerbyte der FC S\_USSI. Nach Behebung des Parametrierungsfehlers müssen alle bereits generierten Datenbausteine vor dem Neustart gelöscht werden.

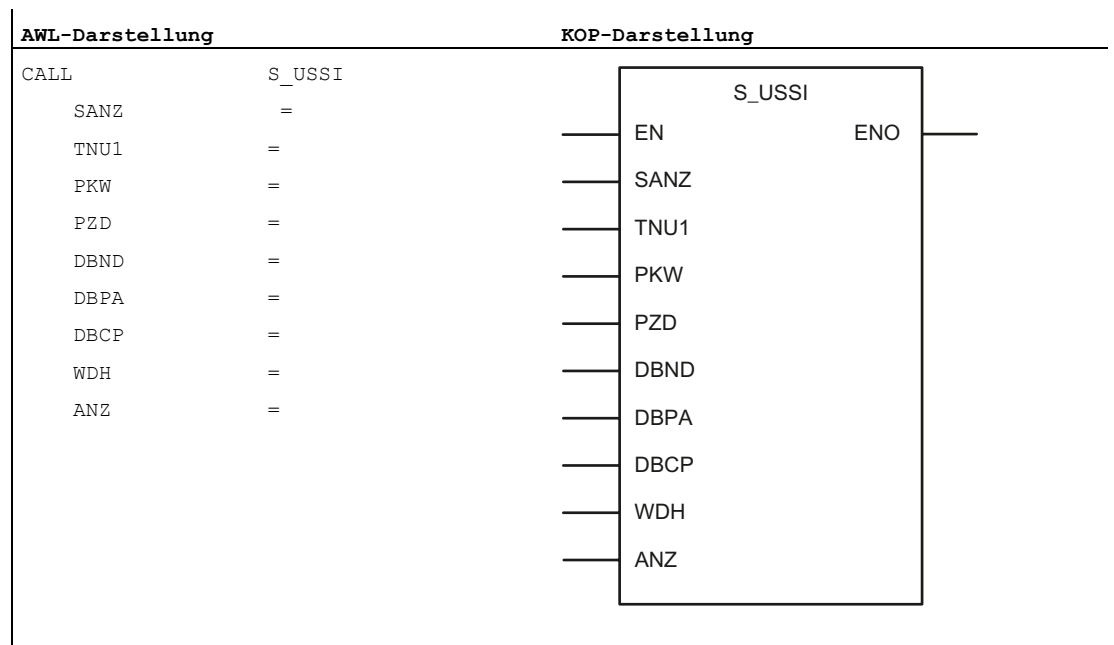
Nach der Plausibilitätsprüfung prüft der Baustein, ob die zu generierenden Datenbausteine bereits vorhanden sind:

- Sind die zu generierenden Datenbausteine noch nicht vorhanden, werden sie erzeugt und der DBPA wird vorbesetzt.
- Sind die zu generierenden Datenbausteine bereits vorhanden, wird die jeweilige Bausteinlänge geprüft. Ist der DB lang genug, dann wird die Vorbesetzung des Parametrierungs-DB neu erstellt und der Inhalt des Netzdaten-DB und des Kommunikationsprozessor-DB wird gelöscht. Ist ein DB zu kurz, geht die CPU in den Betriebszustand STOP. Der Anwender erkennt den fehlerhaften DB im Anzeigenbyte der FC S\_USSI. Zur Behebung des Fehlers müssen die drei Datenbausteine vollständig gelöscht werden. Beim nachfolgenden Neustart werden sie dann neu erzeugt und der Parametrierungs-DB wird vorbesetzt.

S\_USSI muss einmal während des Systemanlaufs aufgerufen werden (OB100).



Tabelle 3- 61 AWL- und KOP-Darstellungen



#### Hinweis

Die Parameter EN und ENO sind nur in der grafischen Darstellung vorhanden (bei KOP oder FUP). Für die Bearbeitung dieser Parameter wird vom Compiler das Binärergebnis BIE verwendet.

Das Binärergebnis BIE wird auf Signalzustand "1" gesetzt, wenn der Baustein fehlerfrei beendet wurde. Liegt ein Fehler vor, wird das Binärergebnis BIE auf "0" gesetzt.

**Parameter FC19 S\_USSI**

Die Tabelle führt die Parameter von FC S\_USSI auf.

Tabelle 3- 62 Parameter FC S\_USSI

Name	Art	Datentyp	Beschreibung	Kommentar
SANZ	INPUT	INT	Anzahl der Slaves mit gleicher Netzdatenstruktur (Systemparameter im DBPA)	1 <= SANZ <= 31
TNU1	INPUT	INT	Anfangsteilnehmernummer (Stationsnummer)	0 <= TNU1 <= 31
PKW	INPUT	INT	PKW, Anzahl	Anzahl der Wörter der PKW-Schnittstelle 0, 3 oder 4
PZD	INPUT	INT	PZD, Anzahl	Anzahl der Wörter der PZD-Schnittstelle 0 <= PZD <= 16
DBND	INPUT	INT	Nummer Netzdaten-DB	CPU-spezifisch (Null nicht zulässig)
DBPA	INPUT	INT	Nummer Parametrierungs-DB	CPU-spezifisch (Null nicht zulässig)
DBCP	INPUT	INT	Nummer Kommunikationsprozessor-DB	CPU-spezifisch (Null nicht zulässig)
WDH	INPUT	INT	Anzahl zulässiger Wiederholungen eines PKW-Auftrags	0 <= WDH <= 32767
ANZ	OUTPUT	BYTE	Fehlerbyte	0: kein Fehler 1: Anzahl Slaves zu groß 2: unzulässige Angaben für Netzdatenstruktur 3: Parametrierungs-DB zu kurz 4: Netzdaten-DB zu kurz 5: Fehler Stationsnummer 6: Kommunikationsprozessor-DB zu kurz 7: frei 8: Wiederholungszähler: falscher Wert

### 3.8.8 Netzdaten-DB

#### Beschreibung

Diese Datenbausteine können entweder mit der FC S\_USSI beim CPU-Anlauf erzeugt und vorbesetzt (nur DBPA) oder manuell eingegeben werden.

Der Netzdaten-DB bildet die Schnittstelle zwischen dem Kommunikations- und dem Steuerungsprogramm. Der Anwender muss diesen Baustein in ausreichender Länge "leer" bereitstellen. Nur die Sendedaten für einen Slave werden in den dem Slave vom Steuerungsprogramm zugeordneten Sendepuffer des Netzdaten-DB eingetragen. Die Antwortdaten vom Slave werden aus dem entsprechenden Empfangspuffer übernommen (nach Auswertung von Bit 9 im Kommunikations-Steuerwort). Statusworte erlauben die Kontrolle der Kommunikation, das Steuerwort den gezielten Anstoß eines Parametrierauftrags.

**Die Kommunikationsschnittstelle enthält für jeden Slave einmal folgende Daten:**

- Slavebezogene Kommunikationsdaten (Kommunikationssteuerung, Verfolgung, 6 Datenworte)
- Puffer für den laufenden PKW-Auftrag (nur wenn ein PKW-Bereich vorhanden ist)
- Sendepuffer für Netzdaten (maximal 20 Datenworte)
- Empfangspuffer für Netzdaten (maximal 20 Datenworte)

Die Länge der Sende- und Empfangspuffer richten sich nach der gewählten Netzdatenstruktur. Bei fehlender PKW-Schnittstelle entfällt der Puffer für den aktuellen PKW-Auftrag.

Die Gesamtlänge des benötigten Netzdaten-DB ist abhängig von der Anzahl Slaves und der verwendeten Netzdatenstruktur.

Anzahl Datenworte je Slave =  $2 \times (\text{PKW} + \text{PZD}) + \text{PKW} + 6$

mit  $\text{PKW} = 0, 3 \text{ oder } 4$  und  $0 \leq \text{PZD} \leq 16$

**Beispiel:** Ein Antrieb mit einem PKW-Bereich von 3 Worten und einem PZD-Bereich von 2 Worten belegt folglich 19 Datenworte im Netzdaten-DB.

Der Netzdaten-DB ist bei 31 Slaves und maximaler Netzdatenlänge 1550 Datenworte lang. DBW0 ist reserviert.

**Slave-Datenbelegung im Netzdaten-DB mit 4 Worten im PKW-Bereich und 0 bis 16 Worten im PZD-Bereich**

DBWn	Kommunikations-Steuerwort (KSTW)		Kommunikationssteuerung	
DBWn+2	Intern			
DBWn+4	Kommunikations-Statuswort		Kommunikationsverfolgung	
DBWn+6	Kommunikations-Fehlerwort		Fehlerstatus	
DBW n+8	Intern		PKW-Versuchszähler	
DBW n+10	Pafe 1-Byte, Pafe 2-Byte		Parameterfehler	
DBW n+12	Parameter-ID	PKE	Puffer für aktuellen PKW-Auftrag	
DBW n+14	Index	IND		
DBW n+16	Parameterwert 1	PWE1		
DBW n+18	Parameterwert 2	PWE2		
DBW n+20	Parameter-ID	PKE	PKW-Bereich	Sendepuffer
DBW n+22	Index	IND		
DBW n+24	Parameterwert 1	PWE1		
DBW n+26	Parameterwert 2	PWE2		
DBW n+28	Steuerwort (STW)	PZD1	PZD-Bereich (max. 16 Worte PZD)	
DBW n+30	Hauptsollwert (HSW)	PZD2		
DBW n+32	Sollwert/Zusatzsteuerwort	PZD3		
DBW n+34	Sollwert/Zusatzsteuerwort	PZD4		
...	...			
DBW n+58	Sollwert/Zusatzsteuerwort	PZD16	PKW-Bereich	
DBW n+60	Parameter-ID	PKE		
DBW n+62	Index	IND		
DBW n+64	Parameterwert 1	PWE1		
DBW n+66	Parameterwert 2	PWE2	PZD-Bereich (max. 16 Worte PZD)	
DBW n+68	Statuswort (ZSW)	PZD1		
DBW n+70	Hauptistwert (HIW)	PZD2		
DBW n+72	Istwert/Zusatzstatuswort	PZD3		
DBW n+74	Istwert/Zusatzstatuswort	PZD4		
...	...			
DBW n+98	Istwert/Zusatzstatuswort	PZD16		
		•		
(n = 2,4,6...)		•		

**Hinweis**

Gibt es keinen PKW-Bereich, dann sind der Puffer für aktuelle PKW-Aufträge und der PKW-Bereich im Sendepuffer nicht vorhanden.

### Kommunikations-Steuerwort KSTW (DBW n)

Die Bits im Kommunikations-Steuerwort koordinieren das Anwenderprogramm und die FC S\_USST FC.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

- **Bit 0: Anstoß PKW-Auftrag**  
 Bit 0 wird vom Anwender gesetzt, wenn ein neuer PKW-Auftrag im Sendepuffer steht und bearbeitet werden soll. Das Bit wird durch die FC zurückgesetzt, wenn der PKW-Auftrag übernommen wurde.
- **Bit 1: Annahme Parameteränderungsbericht**  
 Bit 1 wird vom Anwender gesetzt, wenn der Parameteränderungsbericht angenommen wurde. Das Bit wird durch die FC zurückgesetzt, um die Übernahme zu quittieren. Der Slave führt nach dieser Quittierung die unterbrochene Bearbeitung des laufenden Auftrags fort oder überträgt den nächsten Parameteränderungsbericht.

### Kommunikations-Statuswort (DBW n+4)

Die Bits im Kommunikations-Statuswort werden von den FCs S\_USST und S\_USSR gesetzt.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

- **Bit 0: PKW-Auftrag in Bearbeitung**  
 Bit 0 wird von der FC S\_USST gesetzt, wenn der PKW-Auftrag angenommen wurde und die Parameter-ID (PKE) eine gültige Auftrags-ID enthält. Das Bit wird durch die FC S\_USSR zurückgesetzt, wenn der PKW-Auftrag ausgeführt wurde (mit oder ohne Fehler) oder wenn die PKW-Schnittstelle gestört ist.
- **Bit 1: PKW-Auftrag fertig ohne Fehler**  
 Bit 1 wird von der FC S\_USSR gesetzt, wenn ein PKW-Auftrag fehlerfrei ausgeführt wurde. Die Antwort ist dem Empfangspuffer zu entnehmen. Das Bit wird von der FC S\_USST zurückgesetzt, wenn ein neuer PKW-Auftrag angestoßen wird.

#### Hinweis

Die PKW-Aufträge für den Slave werden in der Reihenfolge der Polling-Liste (DBPA) abgearbeitet. Es ist pro Slave jeweils nur ein Auftrag aktiv. Ist mehr als ein Slave in der Polling-Liste eingetragen, sind die Antwortdaten für einen neuen PKW-Auftrag nur bei einer steigenden Flanke von Bit 1 (bzw. Bit 2) verfügbar.

- Bit 2: PKW-Auftrag fertig mit Fehler

Bit 2 wird von der FC S\_USSR bei Antwort-ID im PKE gesetzt. Die Fehlernummer steht im PWE der Slave-Antwort. Das Bit wird von der FC S\_USST zurückgesetzt, wenn ein neuer PKW-Auftrag angestoßen wird.

---

#### Hinweis

Der vom Anwender zuletzt übertragene PKW-Auftrag wird nach der Bearbeitung in der Sendeschnittstelle gespeichert. Die Übertragung an den Slave wird wiederholt, bis ein neuer Auftrag eingegeben wird. Dadurch können zusätzliche Antworten im Anwenderprogramm erforderlich sein, wenn der Status-PKW-Auftrag mit Fehler (Bit 2) und PKW-Schnittstellenfehler (Bit 4) beendet wird.

---

- Bit 3: PKW-Auftrags-ID ungültig.

Bit 3 wird von der FC S\_USST gesetzt, wenn die Auftrags-ID 15 im PKE festgestellt wird oder wenn der Index 255 in die Auftrags-ID 4 eingetragen wird. Das Bit wird von der FC S\_USST zurückgesetzt, wenn der nächste PKW-Auftrag mit gültiger Auftrags-ID im PKE angestoßen wird.

- Bit 4: PKW-Schnittstelle mit Fehler (Zählerüberlauf).

Bit 4 wird von der FC S\_USSR gesetzt, wenn der PKW-Auftrag nicht innerhalb einer parametrierbaren Anzahl von Auftragswiederholungen (Parameter WDH im Parametrierungs-DB) vom Slave beantwortet wird oder bei Antwort-ID 8 im PKE. Das Bit wird von der FC S\_USSR zurückgesetzt, wenn ein neuer PKW-Auftrag angestoßen und ordnungsgemäß ausgeführt wird.

- Bit 5: Antwortdaten enthalten Parameteränderungsbericht.

Bit 5 wird von der FC S\_USSR gesetzt, wenn ein Parameteränderungsbericht vom Slave vorhanden ist (Antwort-ID 9 bis 12 und Umschaltbit 11 invertiert). Das Bit wird von der FC S\_USST zurückgesetzt, wenn der Anwender den Parameteränderungsbericht quittiert (Kommunikations-Steuerwort, Bit 1).

- Bit 6: Betriebsstörung beim Slave.

Bit 6 wird von der FC S\_USSR FC gesetzt und zurückgesetzt. Die FC wertet das Statuswort (Bit 3) des Slave aus.

- Bit 7: Warnung vom Slave steht an.

Bit 7 wird von der FC S\_USSR FC gesetzt und zurückgesetzt. Die FC wertet das Statuswort (Bit 7) des Slave aus.

- Bit 8: Steuerung vom Automatisierungssystem gefordert.

Bit 8 wird von der FC S\_USSR FC gesetzt und zurückgesetzt. Die FC wertet das Statuswort (Bit 9) und das Steuerwort (Bit 10) aus.

- Bit 9: Sammelfehler Kommunikation.

Bit 9 wird von der FC S\_USSR FC gesetzt und zurückgesetzt. Die FC wertet die Rückmeldungen der Standardbausteine S\_SEND und S\_RCV aus und prüft das empfangene Telegramm in Bezug auf ADR, STX, BCC und LGE. Die FC meldet hier zusätzlich die Überschreitung der Telegrammüberwachungszeit.

---

#### Hinweis

Die Empfangsdaten aus dem Netzdaten-DB sind nur gültig, wenn Bit 9 = 0.

---

### Aufbau des Kommunikations-Fehlerworts (DBW n+6)

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

- Bit 0: Adressierungsfehler (ADR)
- Bit 3: Telegrammstart nicht erkannt (erstes Zeichen kein STX)
- Bit 4: falsches Blockprüfzeichen (BCC)
- Bit 6: falsche Telegrammlänge (LGE)

Die Bits 0, 3, 4 und 6 werden von der FC S\_USSR gesetzt, wenn bei der Prüfung des empfangenen Telegramms ein Fehler erkannt wird (ADR, STX, BCC, LGE).

- Bit 7: Telegrammüberwachungszeit abgelaufen  
Bit 7 wird von der FC S\_USSR gesetzt, wenn die Zeit zwischen dem Senden des Telegramms vom Master an den Slave und die Ankunft der Antwort vom Slave die vom Programm berechnete zulässige Zeit überschreitet (Telegrammüberwachungszeit).

Die übrigen Bits werden nicht verwendet.

### Pafe 1-Byte

Fehlermeldung von der FC S\_USSR, Parametrierungsfehler im Parametrierungs-DB

- Wert 0: Kein Fehler
- Wert 1: Falsche Daten für PKW / PZD

### Pafe 2-Byte

Fehlermeldung von der FC S\_USST, Parametrierungsfehler im Parametrierungs-DB

- Wert 0: Kein Fehler
- Wert 1: Falsche Daten für PKW / PZD

### Parameter-ID PKE im Sendepuffer

Der Anwender muss die Parameternummer (Bits 0 bis 10) und die Auftrags-ID (Bits 12 bis 15) versorgen. Das Umschaltbit für den Parameteränderungsbericht (Bit 11) wird von den Funktionen S\_USSR und S\_USST maskiert.

### 3.8.9 Parametrierungs-DB

#### Beschreibung

Der Parametrierungs-DB enthält die für die Steuerung der Kommunikation notwendigen Programmparameter. Der Anwender muss diesen Baustein erzeugen und der Konfiguration des Kommunikationssystems entsprechend vorbesetzen (S\_USSI oder manuell). Die Slaves am Bus werden in der Reihenfolge des Eintrags im DBPA bearbeitet (Polling-Liste).

Ein Slave kann auch mehrmals in den Parametrierungs-DB eingetragen werden, wodurch dessen Priorität effektiv erhöht wird.

Die Länge des Parametrierungs-DB ist abhängig von der Anzahl n der in einem Buszyklus anzusprechenden Slaves.

$$\text{Anzahl Datenworte des Parametrierungs-DB} = (n \times 4) + 5.$$

Für jede Slavekommunikation werden 4 Datenworte benötigt, für die Systemparameter werden einmalig 4 Datenworte belegt. DBW0 ist reserviert.

DBW 0	Frei	Systemparameter
DBW 2	DBCP	
DBW 4	SANZ	
DBW 6	SLAV	
DBW 8	WDH	
DBW 10	Anzahl PKW, Anzahl PZD	Kommunikation Parametersatz Slave 1
DBW 12	TUN	
DBW 14	DBND	
DBW 16	KSTW	
DBW 18	Anzahl PKW, Anzahl PZD	Kommunikation Parametersatz Slave 2
DBW 20	TUN	
DBW 22	DBND	
DBW 24	KSTW	
		Kommunikation Parametersatz Slave n
	Anzahl PKW, Anzahl PZD	
	TUN	
	DBND	
DBW (n x 8 + 8)	KSTW	



## Systemparameter

DBCP	Bausteinnummer des Kommunikationsprozessor-DB
SANZ	Anzahl aller Slave-Parametersätze im Parametrierungs-DB. Sollen einzelne Slaves in einem Buszyklus häufiger als andere angesprochen werden, so müssen ihre Slaveparameter mehrmals in den Parametrierungs-DB eingetragen werden. Der Systemparameter SANZ muss entsprechend angepasst werden.
SLAV	Laufende Nummer des aktuellen Slaves. Wird von der FC S_USST und der FC S_USSR zur Ermittlung des aktuellen Parametersatzes benötigt. Dieses Datenwort muss mit 1 vorbesetzt werden. Dies wird von der FC S_USSI durchgeführt, sofern diese verwendet wird.
WDH	Anzahl der zulässigen Wiederholungen eines PKW-Auftrags (Wertebereich: 0 bis 32767). Wird der aktuelle PKW-Auftrag nicht innerhalb der Satznummer beendet, wird die PKW-Schnittstelle als gestört gemeldet.

## Slave-Kommunikationsparametrierung

Anzahl PKW, Anzahl PZD	Definition der Netzdatenstruktur Linkes Byte: Anzahl Worte für PKW-Bereich (0, 3, 4) Rechtes Byte: Anzahl Worte für PZD-Bereich (0 bis 16)  Davon abweichende Angaben werden als Parametrierungsfehler erkannt (von den FCs S_USST und S_USSR) und in das Pafe 1-Byte, Pafe 2-Byte des Netzdaten-DB eingetragen.
TUN	Teilnehmernummer, die der am Antrieb eingestellten Busadresse entspricht (0 bis 31).
DBND	Bausteinnummer des Netzdaten-DB.
KSTW	Adresse des Kommunikations-Steuerworts KSTW für den Slave im Netzdaten-DB.

### 3.8.10 Kommunikationsprozessor-DB

#### Aufbau des Kommunikationsprozessor-DBs

Über diesen Datenbaustein wird der Datenverkehr zwischen der CPU und der seriellen Schnittstellenbaugruppe ET 200S Modbus/USS abgewickelt. Der Anwender muss diesen Baustein in ausreichender Länge bereitstellen. Der Kommunikationsprozessor-DB muss mindestens 50 Worte lang sein (DBW 0 bis 98).

DBW 0	Kommunikationsstatus		SENDEN und EMPFANGEN
DBW 2	Maximale Anzahl Zyklen beim Warten auf Empfang	Zykluszähler für Timeout-Bildung beim Warten auf Empfang	FC17
DBW 4	Gemessene Startpause		FC17 FC17, OB1
DBW 6	Dauer des letzten Zyklus (OB1_MIN_CYCLE)		
DBW 8	Sendetelegrammlänge (LEN)		SENDEN
DBB10	Frei		
DBB 11 : : DBB 54	Sendepuffer		Telegramm an Baugruppe senden (Länge richtet sich nach der Netzdatenstruktur des aktuellen Slaves)
DBB 55 : : DBB 98	Empfangspuffer		Telegramm von der Baugruppe empfangen (Länge richtet sich nach der Netzdatenstruktur des aktuellen Slaves)

## Kommunikationsstatus DBW0

Das DBW0 enthält folgende Bits:

- Bit 0: Eingang REQ für S\_SEND.  
Dieses Bit wird zurückgesetzt, wenn Bit 8 gesetzt wird.
- Bit 1: Eingang R für S\_SEND.  
Dieses Bit wird zyklisch von S\_USST zurückgesetzt.
- Bit 2: Ausgang DONE von S\_SEND.
- Bit 3: Ausgang ERROR von S\_SEND.
- Bit 4: Eingang EN\_R für S\_RCV.  
Dieses Bit wird zyklisch von S\_USSR gesetzt.
- Bit 5: Eingang R für S\_RCV.  
Dieses Bit wird zyklisch von S\_USSR zurückgesetzt.
- Bit 6: Ausgang NDR von S\_RCV.
- Bit 7: Ausgang ERROR von S\_RCV.
- Bit 8: Auftrag in Bearbeitung (Bit DONE von S\_SEND gespeichert).  
Dieses Bit wird von S\_USST gesetzt und zurückgesetzt.

## Dauer des letzten Zyklus DBW6

Mit diesem Parameter misst S\_USST die Antwortzeit eines Slaves. Das Anwenderprogramm muss die Zykluszeit des Automatisierungssystems (OB1\_MIN\_CYCLE) vor jedem Aufruf von S\_USST in diesen Parameter kopieren.

## 3.9 Anlaufeigenschaften und Betriebszustände der seriellen Schnittstellenbaugruppe ET 200S Modbus/USS

### 3.9.1 Laden der Konfigurations- und Parametrierungsdaten

#### Datenhaltung

Beim Schließen der Hardware-Konfiguration werden die Daten automatisch in Ihrem STEP 7-Projekt abgelegt.

#### Laden von Konfiguration und Parametern

Die Konfigurations- und Parametrierungsdaten können Sie online vom Programmiergerät in die CPU laden. Mit dem Menübefehl "Zielsystem > Laden" werden die Daten in die CPU übertragen.

Die Baugruppenparameter der Baugruppe werden beim Anlauf der CPU und bei jedem STOP/RUN-Übergang automatisch in die Baugruppe übertragen, sobald diese über den S7-300-Rückwandbus erreichbar ist.

Die Parametrieroberfläche im remanenten Speicher der Baugruppe speichert den Treibercode. Deshalb ist ein Baugruppentausch ohne Programmiergerät nicht möglich.

#### Weitere Informationen

Im Benutzerhandbuch von STEP 7 ist ausführlich beschrieben, wie Sie:

- Konfiguration und Parameter speichern,
- Konfiguration und Parameter in die CPU laden,
- Konfiguration und Parameter lesen, ändern, kopieren und drucken.

### 3.9.2 Betriebszustände der seriellen Schnittstellenbaugruppe ET 200S Modbus/USS

#### Betriebszustände

Die serielle Schnittstellenbaugruppe ET 200S Modbus/USS bietet folgende Betriebszustände:

- **STOP:**

Im Zustand STOP der Baugruppe ist kein Protokoll-Treiber aktiv, alle Sende- und Empfangsaufträge von der CPU werden negativ quittiert. Die Baugruppe bleibt im Betriebszustand STOP, bis die Ursache für den Zustand STOP behoben ist (Beispiel: Drahtbruch oder ungültiger Parameter).

- **Rücksetzen von Parametern:**

Wenn Sie Parameter der Baugruppe rücksetzen, wird der Protokolltreiber initialisiert. Die Sammelfehler LED SF ist während des Rücksetzvorgangs eingeschaltet.

Es ist kein Sende- und Empfangsbetrieb möglich, in der Baugruppe gespeicherte Sende- und Empfangstelegramme gehen wegen des Treiber-Neustarts verloren. Die Kommunikation Baugruppe-CPU wird neu gestartet (laufende Telegramme werden abgebrochen).

Zum Abschluss des Rücksetzvorgangs der Parameter befindet sich die Baugruppe im Betriebszustand RUN und ist sende- und empfangsbereit.

- **RUN:**

Die Baugruppe bearbeitet die Sendeaufträge der CPU. Die vom Kommunikationspartner empfangenen Telegramme werden zum Lesen durch die CPU bereitgestellt.

### 3.9.3 Anlaufeigenschaften der seriellen Schnittstellenbaugruppe ET 200S Modbus/USS

#### Phasen des Anlaufs

Der Anlauf besteht aus zwei Phasen:

- **Initialisierung:** Sobald die Baugruppe an Spannung liegt, wird die serielle Schnittstelle initialisiert und wartet auf Parametrierungsdaten von der CPU.
- **Parametrierung:** Während der Parametrierung empfängt die serielle Schnittstellenbaugruppe ET 200S Modbus/USS die Baugruppenparameter, die Sie dem aktuellen Steckplatz in STEP 7 zugeordnet haben.

### 3.9.4 Verhalten der seriellen Schnittstellenbaugruppe ET 200S Modbus/USS bei Betriebszustandsübergängen der CPU

#### Verhalten nach Anlauf

Nach Anlauf der seriellen Schnittstellenbaugruppe ET 200S Modbus/USS werden alle Daten zwischen der CPU und der Baugruppe über die Funktionsbausteine ausgetauscht.

- **CPU-STOP:**

Im Betriebszustand CPU-STOP ist die Kommunikation über PROFIBUS nicht möglich. Eine laufende Datenübertragung zwischen der seriellen Schnittstellenbaugruppe ET 200S Modbus/USS und der CPU, sowohl Sende- als auch Empfangstelegramm, wird abgebrochen und ein Neustart der Verbindung eingeleitet.

- **CPU-Anlauf:**

Beim Anlauf überträgt die CPU Parameter an die Baugruppe.

Durch entsprechende Parametrierung können Sie den Empfangspuffer der Baugruppe im CPU-Anlauf automatisch löschen.

- **CPU-RUN:**

Im Zustand RUN der CPU ist ein uneingeschränkter Sende- und Empfangsbetrieb möglich. In den ersten FB-Durchläufen nach CPU-Neustart werden die Baugruppe und die jeweiligen FBs synchronisiert. Erst danach wird ein neuer S\_SEND oder S\_RCV ausgeführt.

#### Besonderheiten beim Senden von Telegrammen

Telegramme können nur im Betriebszustand RUN gesendet werden.

Schaltet die CPU während der Datenübertragung von der CPU zur Baugruppe in den Betriebszustand STOP, meldet S\_SEND den Fehler (05) 02<sub>H</sub> nach dem Wiederanlauf. Um dies zu verhindern, kann das Anwenderprogramm S\_SEND mit dem Eingang RESET aus dem Anlauf-OB aufrufen.

---

#### Hinweis

Die serielle Schnittstellenbaugruppe ET 200S Modbus/USS sendet Daten erst dann zum Kommunikationspartner, wenn sie alle Daten von der Baugruppe erhalten hat.

---

### Besonderheiten beim Empfangen von Telegrammen

Mit STEP 7 können Sie "Baugruppen-Empfangspuffer im Anlauf löschen = ja/nein" parametrieren.

- Haben Sie "Ja" parametrieren, wird der Empfangspuffer der seriellen Schnittstellenbaugruppe ET 200S Modbus/USS beim Übergang von STOP nach RUN der CPU automatisch gelöscht.
- Haben Sie "Nein" parametrieren, wird das Telegramm im Empfangspuffer der seriellen Schnittstellenbaugruppe ET 200S Modbus/USS gepuffert.

Schaltet die CPU während der Datenübertragung von der CPU zur seriellen Schnittstellenbaugruppe ET 200S Modbus/USS in den Betriebszustand STOP, meldet S\_RCV den Fehler (05) 02<sub>H</sub> nach Wiederanlauf. Um dies zu verhindern, kann das Anwenderprogramm S\_SEND mit dem Eingang RESET aus dem Anlauf-OB aufrufen. Bei "Empfangspuffer der seriellen Schnittstellenbaugruppe ET 200S Modbus/USS im Anlauf löschen = nein" wird das Telegramm erneut von der seriellen Schnittstellenbaugruppe ET 200S Modbus/USS an die CPU übertragen.

## 3.10 Technische Daten

### Allgemeine technische Daten

Für die serielle Schnittstellenbaugruppe ET 200S 1SI Modbus/USS gelten die allgemeinen technischen Daten, wie sie im Handbuch *Dezentrales Peripheriesystem ET 200S* im Kapitel "Allgemeine Technische Daten" beschrieben sind. Dieses Handbuch finden Sie hier:

<http://www.siemens.de/simatic-tech-doku-portal>

### Technische Daten der Protokolle und Schnittstelle

Tabelle 3- 63 Technische Daten zu Protokollen und Schnittstellen der Baugruppe ET 200S Modbus/USS

<b>Allgemeine technische Daten</b>	
Anzeigeelemente	LED grün, TX (Senden) LED grün, RX (Empfangen) LED rot, SF (Systemfehler)
Mitgelieferte Protokolltreiber	Modbus-Treiber USS-Treiber
Baudraten Modbus-Protokoll Baudraten USS-Treiber	110, 300, 600, 1.200, 2.400, 4.800, 9.600, 19.200, 38.400, 57.600, 76.800, 115.200
Zeichenrahmen (11 Bit)	Anzahl Bit pro Zeichen: 8 Anzahl Start-/Stopbits: 1 oder 2 Parität: keine, gerade, ungerade, beliebig
Speicherplatzbedarf der Standardbausteine (FBs)	Senden und Empfangen: ca. 4300 Byte
<b>Technische Daten der RS-232C-Schnittstelle</b>	
Schnittstelle	RS-232C, 8 Klemmen
RS-232C-Signale	TXD, RXD, RTS, CTS, DTR, DSR, DCD, PE Alle elektrisch von der internen Spannungsversorgung der Baugruppe ET 200S Modbus/USS getrennt.
Maximaler Übertragungsweg	15 m
<b>Technische Daten der RS-422/485-Schnittstelle</b>	
Schnittstelle	RS-422, 5 Klemmen RS-485, 3 Klemmen
RS-422-Signale RS-485-Signale	TXD (A)-, RXD (A)-, TXD (B)+, RXD (B)+, PE R/T (A), R/T (B), PE Alle elektrisch von der internen Spannungsversorgung der Baugruppe ET 200S Modbus/USS getrennt.
Maximaler Übertragungsweg	1200 m



## Technische Daten des Modbus/USS

Tabelle 3- 64 Allgemeine technische Daten der Baugruppe ET 200S Modbus/USS

<b>Allgemeine technische Daten</b>	
<b>Maße und Gewicht</b>	
Abmessungen B × H × T (in mm)	15 × 81 × 52
Gewicht	ca. 50 g
<b>Baugruppenspezifische Daten</b>	
RS-232C	
• Anzahl der Eingänge	4
• Anzahl Ausgänge	3
RS-422	
• Anzahl Eingangspaare	1
• Anzahl Ausgangspaare	1
RS-485	
• Anzahl E/A-Paare	1
Kabellänge	
• Geschirmt (RS-232C)	max. 15 m
• Geschirmt (RS-422/485)	max. 1200 m
Schutzart <sup>1</sup>	IEC 801-5
<b>Spannungen, Ströme, Potenziale</b>	
Versorgungsnennspannung der Elektronik (L+)	24V DC
• Verpolschutz	ja
<b>Potenzialtrennung</b>	
• zwischen Kanälen und Rückwandbus	ja
• zwischen Kanälen und Spannungsversorgung der Elektronik	ja
• zwischen Kanälen	nein
• zwischen Kanälen und PROFIBUS-DP	ja
Isolation geprüft mit	
• Kanäle gegen Rückwandbus und Lastspannung L+	DC 500 V
• Lastspannung L+ gegen Rückwandbus	AC 500 V
Stromquelle	
• aus Rückwandbus	max. 10 mA
• aus Spannungsversorgung L+	max. 80 mA, typ. 20 mA
Verlustleistung der Baugruppe	typ. 1,2 W
<b>Status, Alarme, Diagnose</b>	
Statusanzeige	grüne LED (TX) grüne LED (RX)

Allgemeine technische Daten	
Diagnosefunktionen	
<ul style="list-style-type: none"> <li>• Sammelfehleranzeige</li> <li>• Diagnoseinformationen können angezeigt werden</li> </ul>	rote LED (SF) möglich
<b>Ausgänge</b>	
Ausgang, RS-232C-Bereich	
<ul style="list-style-type: none"> <li>• für kapazitive Last</li> <li>• Kurzschlusschutz</li> <li>• Kurzschlussstrom</li> <li>• Spannung an den Ausgängen oder Eingängen zu PE (Erde)</li> </ul>	±max. 10 V max. 2500 pF ja ca. 60 mA max. 25 V
Ausgang, RS-422/485	
Lastwiderstand	
<ul style="list-style-type: none"> <li>• Kurzschlusschutz</li> <li>• Kurzschlussstrom</li> </ul>	min. 50 kΩ ja ca. 60 mA
<sup>1</sup> Externe Schutzgeräte erforderlich in den Eingangsleitungen der Anwenderspannung: <ul style="list-style-type: none"> <li>• Blitzductor Standard-Hutschienen-Adapter</li> <li>• Blitzductor Schutzmodultyp KT AD-24V</li> </ul>	

**Verarbeitungszeiten**

Die Zeit für die komplette Verarbeitung Master-Slave (mit Datenaktualisierungszeit) kann wie folgt ermittelt werden:

- Gesamtverarbeitungszeit (t<sub>8</sub>) = Verarbeitungszeit Master-Auftrag (t<sub>1</sub>) + Sendezeit Master-Auftrag (t<sub>2</sub>) + Verarbeitungszeit Slave-Auftrag (t<sub>3</sub>) + 1 CPU-Zyklus (Zeit zur Verarbeitung des Funktionscodes) (t<sub>4</sub>) + Verarbeitungszeit Slave-Antwort (t<sub>5</sub>) + Sendezeit Slave-Antwort (t<sub>6</sub>) + Verarbeitungszeit Master-Antwort (t<sub>7</sub>)

**Verarbeitungszeit Auftrag/Antwort**

Die Formel zum Berechnen der Sende- und Empfangszeit ist die gleiche für Master und Slave. Bei einer 8-Byte-Datenübertragung kann die Sende- und Empfangszeit folgendermaßen ermittelt werden:

- Ist CPU-Zyklus >> (E/A-Zyklus + 10 ms), dann Verarbeitungszeit = 1 CPU-Zyklus je 7 Bytes, sonst Verarbeitungszeit = (2 CPU-Zyklen + 3 E/A-Zyklen + 10 ms) je 7 Bytes

**Sende-/Empfangszeit für Auftrag/Antwort**

Die Zeit zum Senden bzw. Empfangen eines Auftrags bzw. einer Antwort wird folgendermaßen ermittelt:

- Sende-/Empfangszeit = 10 ms + Übertragungsgeschwindigkeit multipliziert mit der Anzahl Zeichen in der Meldung

Tabelle 3- 65 Beispiel für die Gesamtverarbeitungszeit:

Lesen	Baudrate	E/A-Zyklus	t <sub>1</sub>	t <sub>2</sub>	t <sub>3</sub>	t <sub>4</sub>	t <sub>5</sub>	t <sub>6</sub>	t <sub>7</sub>	t <sub>8</sub>
10 Wörter	9600Bits/s	2 ms	40 ms	12 ms	40 ms	40 ms	160 ms	29 ms	160 ms	483 ms

# Index

## A

- Anlaufeigenschaften, 245
- Anlaufeigenschaften und Betriebszustände
  - Betriebszustände, 245
  - Laden der Parametrierungsdaten, 244
- Anlaufeigenschaften und Betriebszustände
  - Verhalten der seriellen Schnittstellenbaugruppe ET 200S MODBUS/ USS bei Betriebszustandsübergängen der CPU, 246
- Anschlussbelegung des RS-232C-Anschlusskabels für 25-poligen Anschlussstecker, 123
- Anschlussbelegung des RS-232C-Anschlusskabels für 9-poligen Anschlussstecker, 122
- Anschlussbelegung des RS-422-Anschlusskabels für 15-poligen Anschlussstecker, 124
- Anschlussbelegung des RS-485-Anschlusskabels für 15-poligen Anschlussstecker, 125
- ASCII-Treiber
  - RS-232C-Begleitsignale, 49
- Ausnahmecodetelegramm, 132

## B

- Beispiel zur Inbetriebnahme
  - serielle Schnittstelle, 114
- Betriebsart der Kopplung für den Slave-Treiber
  - Datenbereich in der SIMATIC-CPU: Adresstransformation, 163
  - Freigeben/Sperren von Schreibzugriffen, 197
  - Freigeben/Sperren von Schreibzugriffen: Funktionscodes 05, 06, 15, 16, 197
  - Grenzen für Schreibfunktionen: SIMATIC-Speicherbereiche MIN/MAX, 205
  - Grenzen für Schreibzugriffe, 205
  - Grenzen für Schreibzugriffe: Beispiel für FC 05, 06, 16, 206
  - Grenzen für Schreibzugriffe: Übersicht über FC 05, 06, 15, 16, 205
  - Umsetzung der MODBUS-Adressen für Bitfunktionen, 198
  - Umsetzung der MODBUS-Adressen für Bitfunktionen: Beispiel für FC 01, 05, 15, 199
  - Umsetzung der MODBUS-Adressen für Bitfunktionen: Beispiel für FC 02, 201

- Umsetzung der MODBUS-Adressen für Bitfunktionen: Modbus-Adresse Von/Bis, 199, 201
- Umsetzung der MODBUS-Adressen für Bitfunktionen: SIMATIC-Speicherbereich 'ab', 199, 201
- Umsetzung der MODBUS-Adressen für Bitfunktionen: Übersicht über FC 01, 05, 15, 198
- Umsetzung der MODBUS-Adressen für Bitfunktionen: Übersicht über FC 02, 200
- Umsetzung der Modbus-Adressen für Registerfunktionen: ab DB, 204
- Umsetzung der MODBUS-Adressen für Registerfunktionen: Beispiel, 203
- Umsetzung der MODBUS-Adressen für Registerfunktionen: Beispiel für FC 04, 204
- Umsetzung der MODBUS-Adressen für Registerfunktionen: Übersicht über FC 03, 06, 16, 203
- Umsetzung der MODBUS-Adressen für Registerfunktionen: Übersicht über FC 04, 204
- Zugriff mit bitorientierten Funktionscodes, 195
- Zugriff mit bitorientierten Funktionscodes: Funktionscode 02, 195
- Zugriff mit registerorientierten Funktionscodes, 195
- Zugriff mit registerorientierten Funktionscodes: Funktionscode 04, 196
- Zugriff mit registerorientierten Funktionscodes: Resultierende DB-Nummer, 196
- Zugriff mit registerorientierten Funktionscodes: Wortnummer im DB, 196

## C

- CRC-Prüfung, 131

## D

- Datenfeld DATA, 131
  - Bytecount, 131
  - Coil\_Start Address, 131
  - Number\_of\_Coils, 131
  - Number\_of\_Registers, 131
  - Register\_Start Address, 131
- Datenübertragung beim Modbus-Master ET 200S, 133
- Diagnose, 207
  - Aufbau der Diagnosemeldungen der Funktionsbausteine, 208

Aufrufen der Variablen SFCERR, 208  
 Diagnoseinformationen der Status-LEDs, 207  
 Diagnosemeldungen der Funktionsbausteine, 209  
 Ereignisklasse 14 (0E Hex) Ladbare Treiber –  
 Allgemeine Verarbeitungsfehler  
 <Parametrierung>, 212  
 Ereignisklasse 14 (0E Hex) Ladbare Treiber –  
 Allgemeine Verarbeitungsfehler <Verarbeitung  
 eines Auftrags S\_SEND>, 213  
 Ereignisklasse 30 (1EH): Fehler während der  
 Kommunikation zwischen SI und CPU, 215  
 Kanalbezogene Fehlertypen der seriellen  
 Schnittstellenbaugruppe ET 200S, 216  
 PROFIBUS-Slave-Diagnose, 216  
 Diagnose des Kommunikations-FB  
 Diagnose über die Parameter ERROR\_NR,  
 ERROR\_INFO, 217  
 Diagnose über die Parameter ERROR\_NR,  
 ERROR\_INFO:ERROR\_No 1...9, 217  
 Diagnose über die Parameter ERROR\_NR,  
 ERROR\_INFO:ERROR\_No 10...19, 217  
 Diagnose über die Parameter ERROR\_NR,  
 ERROR\_INFO:ERROR\_No 90...99, 217  
 Diagnosefunktionen, 217  
 ERROR\_NR, ERROR\_INFO, 217  
 Fehler während der Initialisierung, 218  
 Diagnosedes des Kommunikations-FB  
 Löschen der Fehler, 217

## E

Einsetzbare Schnittstellen und Protokolle, 133

## F

FB2 S\_RCV, 138  
 Zeitdiagramm, 142  
 Zuordnung im Datenbereich, 140  
 FB3 S\_SEND, 133  
 Aufruf, 136  
 Parameter, 137  
 Zeitdiagramm, 138  
 Zuordnung im Datenbereich, 136  
 Funktionscodes, 130

## H

Halbduplexbetrieb, 24

## K

Konfigurieren der Modbus-Baugruppe, 143, 167  
 Kurzanleitung zur Inbetriebnahme  
 serielle Schnittstelle, 12

## M

Master-Funktionscodes, 130  
 Master-Funktionscode 01 -- Read Exception  
 Status, 154  
 Master-Funktionscode 01 -- Read Output  
 Status, 148  
 Master-Funktionscode 01 -- Read Output  
 Status:Quell-DB SEND, 148  
 Master-Funktionscode 01 -- Read Output  
 Status:Ziel-DB RCV, 148  
 Master-Funktionscode 02 -- Read Input Status, 149  
 Master-Funktionscode 02 -- Read Input  
 Status:Quell-DB SEND, 149  
 Master-Funktionscode 02 -- Read Input Status:Ziel-  
 DB RCV, 149  
 Master-Funktionscode 03 -- Read Output  
 Registers, 150  
 Master-Funktionscode 03 -- Read Output  
 Registers:Ziel-DB RCV, 150  
 Master-Funktionscode 04 -- Read Input  
 Registers, 151  
 Master-Funktionscode 04 -- Read Input  
 Registers:Ziel-DB RCV, 151  
 Master-Funktionscode 05 -- Force Single Coil, 152  
 Master-Funktionscode 05 -- Force Single Coil:Quell-  
 DB SEND, 152  
 Master-Funktionscode 05 -- Force Single Coil:Ziel-  
 DB RCV, 152  
 Master-Funktionscode 06 -- Preset Multiple  
 Registers, 159  
 Master-Funktionscode 06 -- Preset Single  
 Register, 153  
 Master-Funktionscode 06 -- Preset Single  
 Register:Quell-DB SEND, 153  
 Master-Funktionscode 06 -- Preset Single  
 Register:Ziel-DB RCV, 153  
 Master-Funktionscode 07 -- Read Exception  
 Status:Quell-DB SEND, 154  
 Master-Funktionscode 07 -- Read Exception  
 Status:Ziel-DB RCV, 154  
 Master-Funktionscode 08 -- Loop Back Diagnostic  
 Test, 155  
 Master-Funktionscode 08 -- Loop Back Diagnostic  
 Test:Quell-DB SEND, 155  
 Master-Funktionscode 08 -- Loop Back Diagnostic  
 Test:Ziel-DB RCV, 155

Master-Funktionscode 11 -- Fetch Communications Event Counter, 156  
 Master-Funktionscode 11 -- Fetch Communications Event Counter:Ziel-DB RCV, 156  
 Master-Funktionscode 12 -- Fetch Communications Event Log, 157  
 Master-Funktionscode 12 -- Fetch Communications Event Log:Ziel-DB RCV, 157  
 Master-Funktionscode 15 -- Force Multiple Coils, 158  
 Master-Funktionscode 15 -- Force Multiple Coils:Quell-DB SEND, 158  
 Master-Funktionscodes 03 -- Read Output Registers:Quell-DB SEND, 150  
 Master-Funktionscodes 04 -- Read Input Registers:Quell-DB SEND, 151  
 Master-Funktionscodes 11 -- Fetch Communications Event Counter:Quell-DB SEND, 156  
 Master-Funktionscodes 12 -- Fetch Communications Event Log:Quell-DB SEND, 157  
 Master-Funktionscodes 16 -- Preset Multiple Registers:Quell-DB SEND, 159  
 Modbus-Slave-Treiber, 160

## P

Parametrieren des Master-Treibers, 143  
   Antwortzeit, 146  
   Datenbits, 146  
   Halbduplex (RS485) Zweidraht-Betrieb, 145, 166  
   Multiplikator Zeichenverzug, 146  
   Normalbetrieb, 146  
   Parität, 146  
   Stopbits, 146  
   Störunterdrückung, 146  
   Übertragungsgeschwindigkeit, 145  
   Vorbelegung der Empfangsleitung, 145, 167  
 Produktübersicht  
   Bestellnummer, 111  
 Prozedur 3964(R), 33

## Q

Quell-DB SEND  
   Master-Funktionscode 01, 148  
   Master-Funktionscode 02, 149  
   Master-Funktionscode 03, 150  
   Master-Funktionscode 04, 151  
   Master-Funktionscode 05, 152  
   Master-Funktionscode 06, 153  
   Master-Funktionscode 07, 154

Master-Funktionscode 08, 155  
 Master-Funktionscode 11, 156  
 Master-Funktionscode 12, 157  
 Master-Funktionscode 15, 158  
 Master-Funktionscode 16, 159

## R

RS-232C-Begleitsignale, 49  
 RS-232C-Kommunikation  
   Anschlussbelegung, 120  
 RS-422-Kommunikation  
   Anschlussbelegung, 121  
 RS-485-Kommunikation  
   Anschlussbelegung, 121

## S

Salve-Funktionscodes  
   -Funktionscode 04 -- Read Input Registers, 179  
 Schaltbilder mit Anschlussbelegung, 120  
 Schnittstellen  
   RS-232C, 126  
   RS-422/485, 128  
 Serielle Schnittstellenbaugruppe  
   Technische Daten, 108, 109, 249  
 Signale  
   Automatische Bedienung der Begleitsignale, 127  
   RS-232C, 126  
   Zeitdiagramm für Begleitsignale, 128  
 Slave  
   Adresse, 130  
 Slave-Anwendungsbeispiel  
   Funktionscode 06 – Preset Single Register, 186  
   Slave-Funktionscode 02, 174  
 Slave-Anwendungsbeispiele  
   Funktionscode 05, 183  
   Funktionscode 08, 188  
   Funktionscode 15:Umsetzung von Modbus-Adresszuordnung, 190  
   Funktionscode 16, 193  
   Slave-Funktionscode 01, 171  
   Slave-Funktionscode 03, 177  
   Slave-Funktionscode 04, 180  
 Slave-Funktionscodes, 130, 169  
   Funktionscode 01 – Read Coil (Output) Status, 170  
   Funktionscode 01 – Read Coil (Output) Status:Anwendungsbeispiel, 171  
   Funktionscode 01 – Read Coil (Output) Status:bit\_number, 171  
   Funktionscode 01 – Read Coil (Output) Status:start\_address, 170

- Funktionscode 01 – Read Coil (Output)
- Status:Zugriff auf Merker und Ausgänge, 170
- Funktionscode 01 – Read Coil (Output)
- Status:Zugriff auf Zeiten und Zähler, 171
- Funktionscode 02 – Read Input Status, 173
- Funktionscode 02 – Read Input
- Status:Anwendungsbeispiel, 174
- Funktionscode 02 – Read Input
- Status:bit\_number, 173
- Funktionscode 02 – Read Input
- Status:start\_address, 173
- Funktionscode 03 -- Read Output Registers, 176
- Funktionscode 03 – Read Output
- Registers:Anwendungsbeispiel, 177
- Funktionscode 03 – Read Output
- Registers:Berechnungs-Formel für start\_register, 176
- Funktionscode 03 – Read Output
- Registers:register\_number, 176
- Funktionscode 03 – Read Output
- Registers:start\_address, 176
- Funktionscode 04 – Read Input
- Registers:Anwendungsbeispiel, 180
- Funktionscode 04 – Read Input
- Registers:register\_number, 179
- Funktionscode 04 – Read Input
- Registers:start\_address, 179
- Funktionscode 05 -- Force Single Coil, 182
- Funktionscode 05 – Force Single
- Coil:Anwendungsbeispiel, 183
- Funktionscode 05 – Force Single
- Coil:coil\_address, 182
- Funktionscode 05 – Force Single Coil:DATA-on/off, 183
- Funktionscode 05 – Force Single Coil:Zugriff auf Merker und Ausgänge, 182
- Funktionscode 05 – Force Single Coil:Zugriff auf Zeiten und Zähler, 182
- Funktionscode 06 -- Preset Single Register, 185
- Funktionscode 06 – Preset Single
- Register:Anwendungsbeispiel, 186
- Funktionscode 06 – Preset Single Register:DATA Value, 185
- Funktionscode 06 – Preset Single
- Register:start\_register, 185
- Funktionscode 08 -- Loop Back Diagnostic Test, 188
- Funktionscode 08 – Loop Back Diagnostic Test:Anwendungsbeispiel, 188
- Funktionscode 15 -- Force Multiple Coils, 189
- Funktionscode 15 – Force Multiple Coils: Anwendungsbeispiel für Umsetzung von Modbus-Adresszuordnung, 190
- Funktionscode 15 – Force Multiple Coils:DATA, 190
- Funktionscode 15 – Force Multiple Coils:Quantity, 190
- Funktionscode 15 – Force Multiple Coils:start\_address, 189
- Funktionscode 15 – Force Multiple Coils:Zugriff auf Merker und Ausgänge, 189
- Funktionscode 15 – Force Multiple Coils:Zugriff auf Zeiten und Zähler, 189
- Funktionscode 16 – Preset Multiple Coils:Quantity, 192
- Funktionscode 16 -- Preset Multiple Registers, 192
- Funktionscode 16 – Preset Multiple Registers:Anwendungsbeispiel, 193
- Funktionscode 16 – Preset Multiple Registers:DATA (High, Low), 193
- Funktionscode 16 – Preset Multiple Registers:start\_register, 192
- Slave-Treiber
- Bestandteile:Datenaufbau, 160
- Bestandteile:MODBUS-Slave-Kommunikations-FB, 160
- Bestandteile:MODBUS-Slave-Kopplung, 160
- Parameter:Datenbits, 167
- Parameter:Konfigurieren, 164
- Parameter:Multiplikator Zeichenverzug, 168
- Parameter:Normalbetrieb, 167
- Parameter:Parametrieren des Slave-Treibers, 164
- Parameter:Parität, 167
- Parameter:Slave-Adresse, 167
- Parameter:Stoppbits, 167
- Parameter:Störunterdrückung, 168
- Parameter:Übertragungsgeschwindigkeit, 167

## T

- Technische Daten
  - Protokolle und Schnittstelle, 248
  - Serielle Schnittstellenbaugruppe, 108
- Telegrammaufbau, 129
- Telegrammende, 131

## U

- Übertragungsprotokoll, 129
- Übertragungstelegramm, 130
- USS-Master, 220
  - FC17 S\_USST: Daten an einen Slave senden, 226
  - FC17 S\_USST: Daten an einen Slave senden:Parameter, 228
  - FC18 S\_USSR: Daten von einem Slave empfangen, 229

- FC18 S\_USSR: Daten von einem Slave empfangen:Parameter, 231
- FC19 S\_USSI: Initialisierung, 232
- FC19 S\_USSI: Initialisierung:Parameter, 234
- Funktionsübersicht, 224
- Kommunikationsprozessor-DB, 242
- Kommunikationsprozessor-DB:Dauer des letzten Zyklus DBW6, 243
- Kommunikationsprozessor-DB:Kommunikationsstatus DBW0, 243
- Netzdaten-DB, 235
- Netzdaten-DB:Aufbau des Kommunikations-Fehlerworts (DBW n+6), 239
- Netzdaten-DB:Kommunikations-Steuerwort (DBWn+4), 237
- Netzdaten-DB:Kommunikations-Steuerwort KSTW(DBWn), 237
- Netzdaten-DB:Pafe 1-Byte, 239
- Netzdaten-DB:Pafe 2-Byte, 239
- Netzdaten-DB:Parameter-ID PKE im Sendepuffer, 239
- Netzdaten-DB:Slave-Datenbelegung, 236
- Parametrierungs-DB, 240
- Parametrierungs-DB:Slave-Kommunikationsparametrierung, 241
- Parametrierungs-DB:Systemparameter, 241
- Reihenfolge Funktionsaufrufe, 224
- USS-Protokoll, 221
- USS-Protokoll: Netzdatenbaustein, 222
- USS-Protokoll:Datenübertragungsprozedur, 221
- USS-Protokoll:Datenverschlüsselung, 221
- USS-Protokoll:Telegrammaufbau, 221
- USS-Protokoll
  - Allgemeiner Aufbau des Netzdatenbausteins:Parameterbereich (PKW), 222
  - Allgemeiner Aufbau des Netzdatenbausteins:Prozessdatenbereich (PZD), 222
- Master-Funktionscode 06, 153
- Master-Funktionscode 07, 154
- Master-Funktionscode 08, 155
- Master-Funktionscode 11, 156
- Master-Funktionscode 12, 157

## V

- Verdrahtungsrichtlinien, 120
- Vollduplexbetrieb, 24

## Z

- Ziel-DB RCV
  - Master-Funktionscode 01, 148
  - Master-Funktionscode 02, 149
  - Master-Funktionscode 03, 150
  - Master-Funktionscode 04, 151
  - Master-Funktionscode 05, 152

