

SIEMENS

SIMATIC

S7-300/S7-400

Ladbarer Treiber für Punkt-zu-Punkt-CPs: MODBUS-Protokoll, RTU-Format, S7 ist Master

Betriebsanleitung

<u>Vorwort</u>	1
<u>Produktbeschreibung</u>	2
<u>Montage</u>	3
<u>Inbetriebnahme Treiber</u>	4
<u>Übertragungsprotokoll</u>	5
<u>Funktionscodes</u>	6
<u>Schnittstelle CPU - CP</u>	7
<u>Diagnose beim Treiber</u>	8
<u>Applikationsbeispiel</u>	9
<u>Technische Daten</u>	A
<u>Verdrahtungspläne Mehrpunkt</u>	B
<u>Literaturhinweis</u>	C

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.

 GEFAHR
bedeutet, dass Tod oder schwere Körperverletzung eintreten wird , wenn die entsprechenden Vorsichtsmaßnahmen nicht getroffen werden.
 WARNUNG
bedeutet, dass Tod oder schwere Körperverletzung eintreten kann , wenn die entsprechenden Vorsichtsmaßnahmen nicht getroffen werden.
 VORSICHT
mit Warndreieck bedeutet, dass eine leichte Körperverletzung eintreten kann, wenn die entsprechenden Vorsichtsmaßnahmen nicht getroffen werden.
VORSICHT
ohne Warndreieck bedeutet, dass Sachschaden eintreten kann, wenn die entsprechenden Vorsichtsmaßnahmen nicht getroffen werden.
ACHTUNG
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 zu dieser Dokumentation zugehörige Produkt/System darf nur von für die jeweilige Aufgabenstellung **qualifiziertem Personal** gehandhabt werden unter Beachtung der für die jeweilige Aufgabenstellung zugehörigen Dokumentation, insbesondere der darin enthaltenen Sicherheits- und Warnhinweise. Qualifiziertes Personal ist auf Grund seiner Ausbildung und Erfahrung befähigt, im Umgang mit diesen Produkten/Systemen Risiken zu erkennen und mögliche Gefährdungen zu vermeiden.

Bestimmungsgemäßer Gebrauch von Siemens-Produkten

Beachten Sie Folgendes:

 WARNUNG
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

1	Vorwort	7
2	Produktbeschreibung	11
2.1	Einsatzmöglichkeiten	11
2.2	Hardware- und Softwarevoraussetzungen	13
2.3	Grundsätzliches zum GOULD-MODBUS-Protokoll	14
3	Montage	17
3.1	Einsetzen des Dongles	17
3.2	Schnittstellen-Anschluss	18
4	Inbetriebnahme Treiber	21
4.1	Inbetriebnahme Treiber	21
4.2	Installieren des Treibers auf dem STEP 7-PG/-PC	21
4.3	Deinstallieren des Treibers	22
4.4	Konfigurieren der Kopplung	22
4.4.1	Konfigurieren der Kopplung	22
4.4.2	Konfigurieren einer Kopplung mit dem CP 341	22
4.4.3	Konfigurieren einer Kopplung mit dem CP 441-2	23
4.5	Parametrieren des CP	24
4.5.1	Parametrieren des CP 341	24
4.5.2	Parametrieren des CP 441-2	25
4.6	Verbindungsprojektierung	27
4.7	Parametrieren des ladbaren Treibers	28
4.7.1	Modbus Master-Protokoll	28
4.7.2	RS422/485 (X27) - Schnittstelle	32
4.8	Laden der Konfigurations- und Parametrierungsdaten beim CP 341	34
4.9	Treiber in den CP 341 laden	35
4.10	Laden der Konfigurations- und Parametrierungsdaten beim CP 441-2	36
4.11	Anlaufverhalten des CP	36
4.12	Parametrierung "Anlauf der CPU"	37
5	Übertragungsprotokoll	39

6	Funktionscodes	47
6.1	Funktionscode 01 - Read Output Status.....	47
6.2	Funktionscode 02 - Read Input Status.....	49
6.3	Funktionscode 03 - Read Output Registers.....	51
6.4	Funktionscode 04 - Read Input Registers.....	53
6.5	Funktionscode 05 - Force Single Coil.....	55
6.6	Funktionscode 06 - Preset Single Register.....	57
6.7	Funktionscode 07 - Read Exception Status.....	59
6.8	Funktionscode 08 - Loop Back Diagnostic Test.....	61
6.9	Funktionscode 11 - Fetch Communications Event Counter	63
6.10	Funktionscode 12 - Fetch Communications Event Log	64
6.11	Funktionscode 15 - Force Multiple Coils.....	66
6.12	Funktionscode 16 - Preset Multiple Registers	68
7	Schnittstelle CPU - CP	71
7.1	Schnittstelle CPU - CP für CP 341.....	71
7.1.1	Datenübergabe von der CPU an den CP mit P_SND_RK (CP 341).....	72
7.1.2	Datenübergabe vom CP an die CPU mit P_RCV_RK (CP 341).....	74
7.2	Schnittstelle CPU - CP für CP 441-2	75
7.2.1	Datenübergabe von der CPU an den CP mit BSEND (CP 441-2).....	75
7.2.2	Datenübergabe vom CP an die CPU mit BRCV (CP 441-2)	78
8	Diagnose beim Treiber	79
8.1	Diagnosemöglichkeiten beim CP 341	80
8.1.1	Diagnose über die Anzeigeelemente des CP 341	80
8.1.2	Diagnosemeldungen der Funktionsbausteine des CP 341.....	81
8.2	Diagnosemöglichkeiten beim CP 441-2.....	82
8.2.1	Diagnose über die Anzeigeelemente des CP 441-2.....	82
8.2.2	Diagnosemeldungen der Systemfunktionsbausteine des CP 441-2	83
8.2.3	Diagnose über den Fehlermeldebereich SYSTAT des CP 441-2.....	84
8.3	Tabelle der Fehler/Ereignisse	86
8.3.1	Fehlercode im SYSTAT bei "CPU-Auftrags-Fehler"	86
8.3.2	Fehlercode im SYSTAT bei "Empfangsfehlern".....	87
8.3.3	Fehlercode im SYSTAT bei "allgemeinen Verarbeitungsfehlern".....	88
9	Applikationsbeispiel	93
9.1	Applikationsbeispiel für CP 341	93
9.1.1	Applikationsbeispiel für CP 341	93
9.1.2	Verwendete Bausteine	93
9.1.3	Programmbeschreibung.....	95
9.1.4	Programmbeispiel	96
9.2	Applikationsbeispiel für CP 441-2	101
9.2.1	Verwendete Bausteine	101
9.2.2	Programmbeschreibung.....	103
9.2.3	Programmbeispiel	105

A	Technische Daten	113
	A.1 Technische Daten	113
B	Verdrahtungspläne Mehrpunkt	119
C	Literaturhinweis	121
	Glossar	123
	Index	131

Vorwort

Zweck des Handbuchs

Die Informationen dieses Handbuchs ermöglichen es Ihnen, eine Kopplung zwischen einem CP als "modbusfähigem" Master und einem Modbus-Slave-Steuerungssystem aufzubauen und in Betrieb zu nehmen.

Erforderliche Grundkenntnisse

Zum Verständnis des Handbuchs sind allgemeine Kenntnisse auf dem Gebiet der Automatisierungstechnik erforderlich.

Außerdem werden Kenntnisse über die Verwendung von Computern oder PC-ähnlichen Arbeitsmitteln (z. B. Programmiergeräten) unter dem Betriebssystem Microsoft® Windows® und Kenntnisse in STEP 7-Programmierung vorausgesetzt.

Gültigkeitsbereich des Handbuchs

Das vorliegende Handbuch ist gültig für folgende Software:

Produkt	Bestellnummer	Ab Version
Ladbarer Treiber für Punkt-zu-Punkt-CPs	6ES7870-1AA01-0YA0	3.0

Hinweis

Das vorliegende Handbuch enthält die Beschreibung von Treiber und Funktionsbaustein, wie sie zum Zeitpunkt der Herausgabe des Handbuchs gültig ist.

Wegweiser

Im vorliegenden Handbuch sind die Funktion des ladbaren Treibers und die Einbindung in die Hardware und Software des Kommunikationsprozessors CP 341 und CP 441-2 beschrieben.

Das Handbuch beinhaltet folgende Themen:

- Produktbeschreibung / Montage
- Inbetriebnahme Treiber / Installieren / Parametrieren
- Schnittstelle CPU – CP
- Übertragungsprotokoll / Funktionscodes
- Diagnose beim Treiber
- Applikationsbeispiel

Konventionen

In der vorliegenden Dokumentation wird im Folgenden die Bezeichnung CP oder CP 341 bzw. CP 441-2 verwendet.

Besondere Hinweise

Der in diesem Handbuch beschriebene Treiber stellt ein ladbares Protokoll für den CP dar, das anstelle der Standard-Protokolle 3964R, RK512, ASCII, Drucker verwendet werden kann.

Hinweis

Bei dem vorliegenden Treiber können Abläufe in der Kommunikation zwischen CP und CPU modifiziert oder erweitert sein.

Insbesondere die zur Diagnose vorhandenen Ereignisklassen und Ereignisnummern können modifiziert und erweitert sein.

Beachten Sie weiterhin, dass im vorliegenden Handbuch immer nur die Modifikationen und Erweiterungen gegenüber den Standardfunktionen beschrieben sind. Die grundsätzlichen Informationen entnehmen Sie dem Handbuch des eingesetzten CP.

Um eine sichere Anwendung des Treibers zu erreichen, ist also die genaue Kenntnis der Funktionsweise des CP Voraussetzung.

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>

Produktbeschreibung

2.1 Einsatzmöglichkeiten

Einordnen in die Systemumgebung

Der vorliegende Treiber ist ein Software-Produkt für die Kommunikationsprozessoren CP 341 (S7-300) und CP 441-2 (S7-400).

CP 341 und CP 441-2 sind in den Automatisierungssystemen S7 einsetzbar und können serielle Kommunikationsverbindungen zu Partnersystemen herstellen.

Funktion des Treibers

Mit diesem Treiber wird eine Kommunikationsverbindung zwischen der Kommunikationsbaugruppe CP 341 bzw. CP 441-2 und "modbusfähigen" Steuerungssystemen, z. B. Modicon-Steuerungen oder Honeywell TDC 3000, ermöglicht.

Zur Übertragung wird das **GOULD - MODBUS -Protokoll** im **RTU-Format** verwendet. Die Datenübertragung wird nach dem Master-Slave-Prinzip abgewickelt. Der **Master (SIMATIC S7)** hat bei der Übertragung die Initiative.

Bei der Kommunikation zwischen CP und Fremdsystem können Sie dabei die **Funktionscodes 01, 02, 03, 04, 05, 06, 07, 08, 11, 12, 15** und **16** verwenden.

Einsetzbare Schnittstellen und Protokolle

Die beiden seriellen Schnittstellen des CP 441-2 können Sie unabhängig voneinander mit unterschiedlichen Standardprotokollen oder ladbaren Protokollen betreiben.

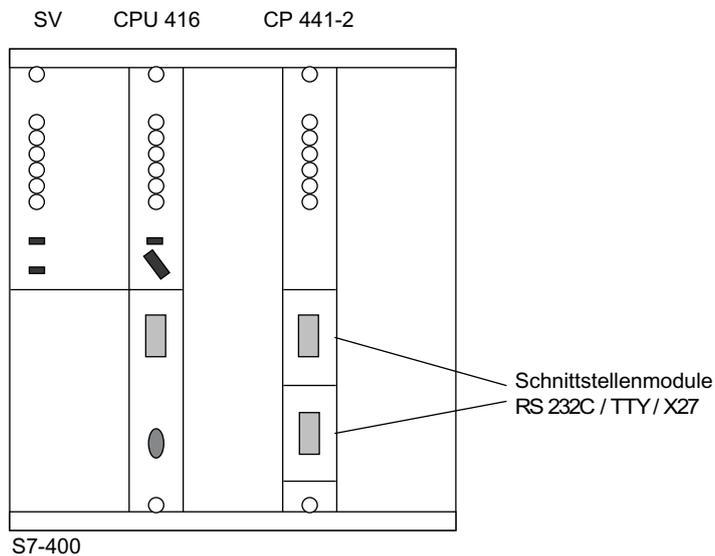
Als Schnittstellenphysik ist bei den CPs RS232, TTY oder RS422/485 (X27) möglich.

Die RS422/485 (X27)-Schnittstelle können Sie bei diesem Treiber sowohl im 2-Draht-Betrieb als auch im 4-Draht-Betrieb einsetzen.

Im 2-Draht-Betrieb ist es möglich, bis zu 32 Slaves an einen Master im Halbduplex-Betrieb anzuschließen und damit eine Mehrpunktverbindung, ein Netzwerk, zu bilden.

Mögliche Anlagenkonfiguration

Im folgenden Bild ist eine mögliche Anlagenkonfiguration schematisch dargestellt.



Datenkonsistenz

Der Datenaustausch zwischen der S7-CPU und dem CP erfolgt blockweise über integrierte Systemfunktionen.

Beachten Sie hierzu das Kapitel "Schnittstelle CPU - CP (Seite 71)" in diesem Handbuch.

2.2 Hardware- und Softwarevoraussetzungen

Verwendbare Baugruppe

Der Treiber ist auf dem CP 341 und auf dem CP 441-2 mit der MLFB 6ES7441-2AA02-0AE0 oder neuer ablauffähig.

Den CP 441-1 mit der MLFB 6ES7441-1AA0x-0AE0 und den CP 441-2 mit der MLFB 6ES7441-2AA00-0AE0 oder 6ES7441-2AA01-0AE0 können Sie nicht mit ladbaren Treibern betreiben.

Dongle

Für den Betrieb des CP mit ladbaren Treibern benötigen Sie einen Dongle. Der Dongle ist im Lieferumfang des Treibers enthalten.

Ladespeicher der CPU (Memory Card)

Bei Verwendung des CP 441-2 werden die ladbaren Treiber nach ihrer Parametrierung in den Ladespeicher der CPU geladen und beim Anlauf der CPU in den CP-Speicher übertragen.

In der CPU muss daher ein ausreichend großer Ladespeicher vorhanden sein. Hierfür wird in der CPU eine RAM- oder FLASH-**Memory Card** MLFB 6ES7952-... benötigt.

Für **jede** CP-Schnittstelle, bei der dieser ladbare Treiber parametrierung wurde, werden ca. 25 KB CPU-Ladespeicher benötigt.

Bei Verwendung des CP 341 werden die ladbaren Treiber direkt in den CP 341 geladen. Es wird deshalb kein Ladespeicher auf der S7-300 CPU benötigt. Beachten Sie aber, dass deshalb ein Baugruppentausch ohne PG nicht möglich ist.

Softwareausgabestände

Eine installierte Version von STEP 7 Basis V5.3 oder höher.

Eine installierte Version des Optionspakets Punkt-zu-Punkt-Kopplungen-Parametrieren CP PtP Param V5.1 oder höher.

Datenstrukturen

Vor dem Projektieren Ihrer S7-Datenstrukturen sollte eine Abstimmung mit den Anwenderprogrammen der MODBUS-Slave-Systeme erfolgen. Klären Sie, welche Funktionscodes und welche Modbus-Adressen verwendet werden.

2.3 Grundsätzliches zum GOULD-MODBUS-Protokoll

Funktionscodes

Die Art des Datenaustausches zwischen MODBUS-Systemen wird durch Funktionscodes FCs gesteuert.

Datenaustausch

Mit den folgenden FCs ist ein **Bit-orientierter** Datenaustausch möglich:

- FC 01 Read coil (output) status,
- FC 02 Read input status,
- FC 05 Force single coil,
- FC 15 Force multiple coils.

Mit den folgenden FCs ist ein **Register-orientierter** Datenaustausch möglich:

- FC 03 Read holding registers,
- FC 04 Read input registers,
- FC 06 Preset single register,
- FC 16 Preset multiple registers.

Datenbereiche

Die einzelnen FCs arbeiten in der Regel nach folgendem Schema:

Funktionscode	Daten	Datentyp		Zugriffsart
01, 05, 15	coil (output) status	Bit	Ausgang	Lesen/schreiben
02	input status	Bit	Eingang	Nur lesen
03, 06, 16	holding register	Register (16 Bit)	Ausgangs- Register	Lesen/schreiben
04	input register	Register (16 Bit)	Eingangs- Register	Nur lesen

Adress-Darstellung

Analog zur Aufteilung in schreibbare und lesbare und nur lesbare Bereiche können die Daten in der Anwenderschicht gemäß folgendem Schema dargestellt sein:

Funktionscode	Datentyp	Adress-Darstellung in der Anwenderschicht (dez.)
01, 05, 15	Output Bit	0xxxx
02	Input Bit	1xxxx
04	Input Register	3xxxx
03, 06, 16	Holding Register	4xxxx

Im **Übertragungstelegramm** auf der seriellen Übertragungsleitung werden die Adressen des MODBUS-Anwendersystems auf **0** bezogen.

Im **MODBUS-Anwendersystem** selbst werden die Adressen beginnend mit **1** gezählt!

Beispiel:

- Das erste Holding Register im Anwendersystem wird als Register **40001** dargestellt. Im Übertragungstelegramm wird als Register Address bei FC 03, 06 und 16 der Wert 0000 Hex übertragen.
- Das 127. Coil stellt im Anwendersystem das coil **00127** dar und hat im Übertragungstelegramm die Coil Address 007E Hex.

Montage

3.1 Einsetzen des Dongles

Einleitung

Für den Betrieb des CP mit ladbaren Treibern benötigen Sie einen Dongle. Bei gestecktem Dongle können Sie Treiber nachladen. Beim CP 441-2 können Sie für beide Schnittstellen Treiber nachladen.

Stecken des Dongles

Zum Stecken des Dongles müssen Sie den CP aus dem Baugruppenträger nehmen. An der Rückseite des CP, oberhalb des Steckers für den Rückwandbus, befindet sich ein Modulschacht, in den Sie den Dongle einsetzen können.

3.2 Schnittstellen-Anschluss

RS 232C / TTY

Es kann eine Punkt-zu-Punkt-Verbindung zu einem Slave-System aufgebaut werden.

Hinweise zum Schnittstellen-Anschluss siehe Handbuch "Punkt-zu-Punkt-Kopplung CP 341 bzw. CP 441-2".

X27/RS485 (Zweidraht)

Es kann direkt eine Mehrpunkt-Verbindung, ein Netzwerk, mit bis zu 32 Slaves an einem Master-System aufgebaut werden.

Auf dem CP führt der Treiber den Umschaltbetrieb der Empfangs-Zweidraht-Leitung zwischen Senden und Empfangen durch.

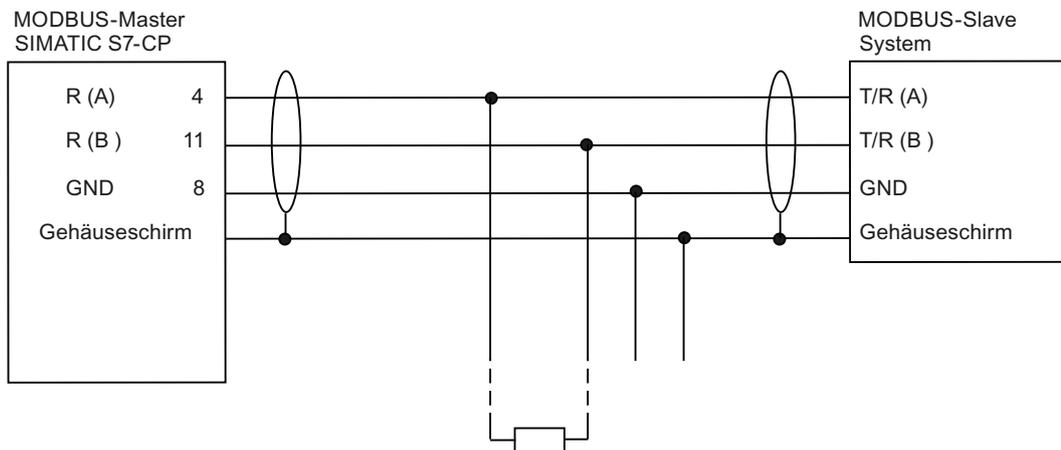


Bild 3-1 Schematischer Anschluss: 1 Master-System, 1 Slave-System am Bus

Weitere Hinweise zum Schnittstellen-Anschluss siehe Handbuch "Punkt-zu-Punkt-Kopplung CP 341 bzw. CP 441-2".

X27/RS422 (Vierdraht)

Es kann eine Punkt-zu-Punkt-Verbindung zu einem Slave-System aufgebaut werden.

Der direkte Aufbau einer Mehrpunkt-Verbindung, Netzwerk, mit mehreren Slaves ist dann möglich, wenn dies durch die Hardware der MODBUS-Slave-Systeme unterstützt wird. Die MODBUS-Slave-Systeme müssen ihre Sender in den hochohmigen Zustand schalten können, wenn sie nicht senden.

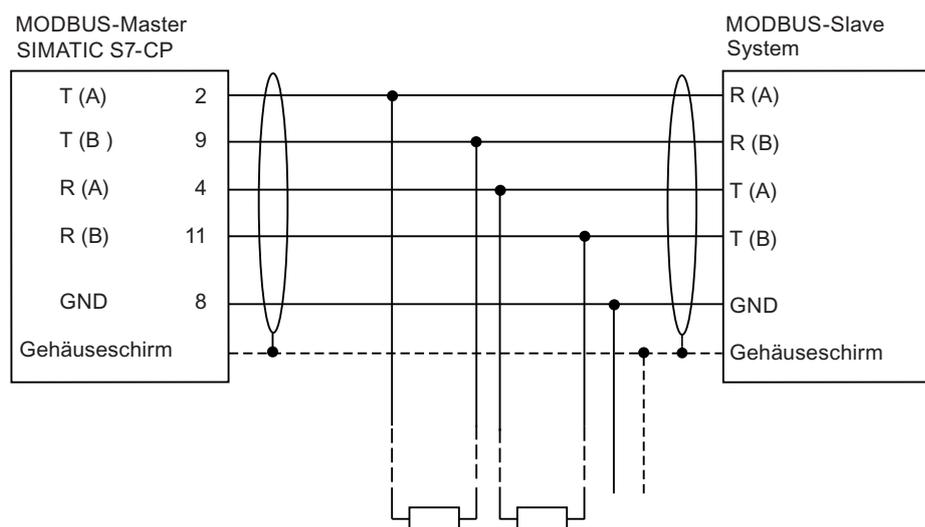


Bild 3-2 Schematischer Anschluss: 1 Master-System, 1 Slave-System

Weitere Hinweise zum Schnittstellen-Anschluss siehe Kapitel "Verdrahtungspläne Mehrpunkt (Seite 119)" und Handbuch "Punkt-zu-Punkt-Kopplung CP 341 bzw. CP 441-2".

Inbetriebnahme Treiber

4.1 Inbetriebnahme Treiber

Allgemeines

Die im Folgenden verwendeten Angaben zu STEP 7 beziehen sich auf die STEP 7-Version 5.3.

Bei späteren Versionen können Abläufe, Namens- und Verzeichnisangaben geändert sein.

4.2 Installieren des Treibers auf dem STEP 7-PG/-PC

Vorgehen

Bei der Installation des Treibers, bestehend aus Treibercode und treiberspezifischen Maskendateien, gehen Sie folgendermaßen vor:

1. Legen Sie die MODBUS-Master-CD in das CD-ROM Laufwerk ein.
2. Starten Sie unter Windows den Dialog zur Installation von Software durch Doppelklick auf das Symbol "Software" in "Systemsteuerung".
3. Wählen Sie im Dialog das CD-ROM-Laufwerk und die Datei **Setup.exe** aus und starten Sie den Installationsvorgang.
4. Befolgen Sie Schritt für Schritt die Anweisungen, die Ihnen das Installationsprogramm anzeigt.

Ergebnis: Der Treiber und die Parametriermasken sind im folgenden Verzeichnis installiert: **Step7\S7ftp\S7Driver**.

Das Verzeichnis enthält u. a. folgende Dateien:

- S7wfpa1a.dll
- S7wfpa1x.cod
- S7wfpa2x.cod

4.3 Deinstallieren des Treibers

Vorgehen

Sie können den Treiber unter Windows über "Systemsteuerung", "Software" und "Deinstallieren" wieder aus dem STEP 7-Paket deinstallieren.

Anschließend können Sie im Verzeichnis Step7\S7ftp\S7Driver überprüfen, ob alle Dateien S7wfpa1?.*, S7wfpa2?.*, S7wfpa3?.* gelöscht sind.

Hinweis

Vor einer Deinstallation des Pakets "**Parametrieroberfläche CP Punkt-zu-Punkt-Kopplungen parametrieren**" müssen Sie alle ladbaren Treiber deinstallieren.

4.4 Konfigurieren der Kopplung

4.4.1 Konfigurieren der Kopplung

Einleitung

Konfigurieren einer Kopplung ist die Anordnung der Hardware in der Konfigurationstabelle mit HW Konfig. Sie konfigurieren die Kopplung mit der STEP 7-Software.

4.4.2 Konfigurieren einer Kopplung mit dem CP 341

S7-Projekt

Vor dem Konfigurieren müssen Sie mit STEP 7 ein **S7-Projekt** anlegen.

Projektkomponenten

Fügen Sie die benötigten Projektkomponenten mit dem SIMATIC-Manager in das geöffnete Projekt ein:

- SIMATIC 300-Station

Selektieren Sie vor jedem Einfügen das geöffnete Projekt durch Anklicken.

Einfügen > Station > SIMATIC 300-Station

Hardware konfigurieren

Das Konfigurieren der Hardware ist die Festlegung der Hardware-Komponenten selbst sowie die Festlegung ihrer Eigenschaften.

Durch Selektion der SIMATIC 300-Station und einen Doppelklick auf "Hardware" bzw. "Bearbeiten > Objekt öffnen" starten Sie die Hardware-Konfiguration.

Fügen Sie mit "Einfügen > Hardwarekomponenten" folgende Komponenten ein:

- Aus SIMATIC 300 ein RACK-300, eine PS-300 und eine CPU-300
- Aus CP-300 den CP PtP mit der entsprechenden Bestellnummer

Das Vorgehen beim Konfigurieren von S7-300-Baugruppen ist ausführlich im Benutzerhandbuch für STEP 7 beschrieben.

4.4.3 Konfigurieren einer Kopplung mit dem CP 441-2

Einleitung

Für eine PtP-Kopplung müssen Sie neben der SIMATIC 400-Station die Koppelpartner-Station, die PtP-Knoten sowie das PtP-Netz konfigurieren.

S7-Projekt

Vor dem Konfigurieren müssen Sie mit STEP 7 ein **S7-Projekt** anlegen.

Projektkomponenten

Fügen Sie die benötigten Projektkomponenten mit dem SIMATIC-Manager in das geöffnete Projekt ein:

- SIMATIC-Station, Andere Station, PtP-Netz.
Selektieren Sie vor jedem Einfügen das geöffnete Projekt durch Anklicken.
 - **Einfügen > Station > SIMATIC 400-Station**
für das eigene S7-Programm (Rack, PS, CPU, CP 441-2, ...),
 - **Einfügen > Station > Andere Station**
für den Koppelpartner,
 - **Einfügen > Subnetz > PtP**
für ein PtP-Netz zwischen der SIMATIC 400-Station und dem Koppelpartner.

Hardware konfigurieren

Das Konfigurieren der Hardware ist die Festlegung der Hardware-Komponenten selbst sowie die Festlegung ihrer Eigenschaften.

Durch Selektion der SIMATIC 400-Station und einen Doppelklick auf "Hardware" bzw. "Bearbeiten > Objekt öffnen" starten Sie die Hardware-Konfiguration.

Fügen Sie mit "Einfügen > Hardwarekomponenten" folgende Komponenten ein:

- Aus SIMATIC 400 ein RACK-400, eine PS-400 und eine CPU-400
- Aus CP-400 den CP PtP mit der entsprechenden Bestellnummer

Das Vorgehen beim Konfigurieren von S7 400-Baugruppen ist ausführlich im Benutzerhandbuch für STEP 7 beschrieben.

4.5 Parametrieren des CP

Parametrieren des CP

Nachdem Sie die Baugruppen mit "Hardware Konfigurieren" in Ihrem Baugruppenträger angeordnet haben, müssen diese parametriert werden.

Die Parametrieroberfläche des CP starten Sie in "Hardware konfigurieren" durch Doppelklick auf den CP oder durch Markieren des CP und Anwahl des Menübefehls **Bearbeiten > Objekteigenschaften**.

4.5.1 Parametrieren des CP 341

So gehen Sie vor

1. Eigenschaften - CP > Grundparameter

Durch Anwahl der Schaltfläche "**Parameter**" mit einem 1-fach Klick gelangen Sie in die Oberfläche zur Protokollauswahl "**Punkt-zu-Punkt-Kopplung parametrieren**". Hier können Sie das gewünschte Übertragungs-Protokoll auswählen.

Nach Auswahl des **Protokolls** können Sie die **Parametrierung des Treibers** vornehmen. Starten Sie durch Doppelklick auf das Brief-Symbol.

Das Auswählen des Protokolls und die Parametrierung der Dialoge für den ladbaren Treiber ist im Kapitel " Parametrieren des ladbaren Treibers (Seite 28) " ausführlich beschrieben.

Nach erfolgter Parametrierung gelangen Sie zurück in den Dialog "**Eigenschaften - CP**".

2. Eigenschaften - CP > Adressen

Im Register "**Adressen**" des Dialogs Eigenschaften - CP brauchen Sie **keine** Einstellungen vorzunehmen.

3. Eigenschaften - CP > Allgemein

Im Register "**Allgemein**" des Dialogs Eigenschaften - CP brauchen Sie **keine** Einstellungen vorzunehmen.

Im Dialog "Eigenschaften - CP" können Sie mit "OK" die Parametrierung des CP abschließen und gelangen dann zurück in den Dialog "Hardware konfigurieren".

Speichern Sie die Parametrierung ab und schließen Sie "Hardware konfigurieren". Sie gelangen zurück in das Hauptmenü des STEP 7-Projekts.

4.5.2 Parametrieren des CP 441-2

So gehen Sie vor

1. Eigenschaften > CP 441-2 > Grundparameter

Im Register "**Grundparameter**" geben Sie jetzt die gewünschte "**Schnittstelle**" der CP 441-Baugruppe an (1=obere, 2=untere Schnittstelle). Wählen Sie bei "**Modul**" das gesteckte Schnittstellenmodul aus.

Durch Anwahl der Schaltfläche "**Parameter**" mit einem 1-fach Klick gelangen Sie in die Oberfläche zur Protokollauswahl "**Punkt-zu-Punkt-Kopplung parametrieren**".

Hier wählen Sie das gewünschte Übertragungs-Protokoll.

Nach Auswahl des **Protokolls** können Sie die **Parametrierung des Treibers** vornehmen. Starten Sie durch Doppelklick auf das Brief-Symbol.

Das Auswählen des Protokolls und die Parametrierung der Dialoge für den ladbaren Treiber ist im Kapitel "Parametrieren des Treibers" ausführlich beschrieben.

Nach erfolgter Parametrierung gelangen Sie zurück in den Dialog "**Eigenschaften - CP 441-2**".

2. Eigenschaften > CP 441-2 > Adressen

Im Register "**Adressen**" des Dialogs Eigenschaften - CP 441-2 brauchen Sie **keine** Einstellungen vorzunehmen.

3. Eigenschaften > CP 441-2 > Allgemein

Im Register "**Allgemein**" des Dialogs Eigenschaften - CP 441-2 müssen Sie angeben, an welches **PtP-Netz** die Schnittstellen des CP angeschlossen werden.

PtP(1) bezieht sich dabei auf die obere Schnittstelle, PtP(2) auf die untere Schnittstelle des CP.

Durch Anklicken der Schaltfläche **PtP(1)** oder **PtP(2)** gelangen Sie in den Dialog zur Projektierung des Subnetzes.

Selektieren Sie das gewünschte **Subnetz** und aktivieren Sie die Checkbox "Teilnehmer ist mit dem ausgewählten Netz verbunden".

Das ausgewählte Subnetz stellt die Verbindung zwischen der CP-Schnittstelle und der Koppelpartner-Schnittstelle dar.

Über die Schaltfläche "OK" gelangen Sie zurück in den Dialog "Eigenschaften - CP 441-2". Hier können Sie mit "OK" die Parametrierung des CP abschließen und gelangen dann zurück in den Dialog "Hardware konfigurieren".

Speichern Sie die Parametrierung ab und schließen Sie "Hardware konfigurieren". Sie gelangen zurück in das Hauptmenü des STEP 7-Projekts.

Parametrieren des Koppelpartners

Nachdem Sie, wie unter "Projektkomponenten: Einfügen > Andere Station" beschrieben, die Koppelpartnerstation in Ihr STEP 7-Projekt eingefügt haben, müssen Sie die Objekteigenschaften dieser Fremd-Station festlegen.

Ausgehend vom geöffneten STEP 7-Projekt können Sie die Koppelpartner-Station (Andere Station) durch Anklicken selektieren.

Wählen Sie den Menüpunkt **Bearbeiten > Objekteigenschaften**.

Sie gelangen dann in den Dialog "Eigenschaften - Andere Station".

1. Eigenschaften > Andere Station > Teilnehmerliste

In Registerkarte "**Teilnehmerliste**" drücken Sie den Knopf "Neu".

In der Typauswahl selektieren Sie "PTP-Teilnehmer" und drücken "OK".

Es erscheint der Dialog "Netzanschluss".

Selektieren Sie das zugehörige **Subnetz**, das die Verbindung zwischen CP-Schnittstelle und Koppelpartner-Schnittstelle darstellt, und aktivieren Sie die Checkbox "Teilnehmer ist mit dem ausgewählten Netz verbunden".

Durch Drücken der Schaltfläche "OK" gelangen Sie zurück in die Registerkarte "Teilnehmerliste".

2. Eigenschaften > Andere Station > Allgemein

Im Register "**Allgemein**" brauchen Sie keine Einstellungen vorzunehmen.

Über die Schaltfläche "OK" gelangen Sie zurück in das Hauptmenü des STEP 7-Projekts.

Eine Fremdstation kann auch mehrere Schnittstellen (=PtP-Teilnehmer) haben und an verschiedene PtP-Verbindungen angeschlossen sein.

4.6 Verbindungsprojektierung

Einleitung

Dieses Kapitel ist nur für den CP 441-2 relevant. Wenn Sie einen CP 341 einsetzen, können Sie dieses Kapitel überschlagen.

Kommunikationsverbindung

Der CP ist das Bindeglied für die Verbindung zwischen einer S7-CPU und einem über eine Punkt-zu-Punkt-Kopplung angeschlossenen Kommunikationspartner/Bus. Für die Verbindung der jeweiligen seriellen Schnittstelle zum Koppelpartner/Bus müssen Sie eine Verbindungsprojektierung vornehmen.

Verbindung projektieren

Selektieren Sie im STEP 7-Projekt in der geöffneten eigenen S7 400-Station die CPU und öffnen Sie durch Doppelklicken auf "**Verbindungen**" die Verbindungsprojektierung.

Es erscheint der Dialog "Verbindungen projektieren".

Nach der Anwahl von **Einfügen > Verbindung** gelangen Sie in den Dialog "Neue Verbindung". Wählen Sie hier für die neue Verbindung den Verbindungspartner (Andere Station) und als Verbindung "PtP-Verbindung" an.

Quittieren Sie mit "OK". Sie gelangen jetzt in den Dialog "Objekteigenschaften - Verbindung".

Objekteigenschaften der Verbindung

Es wird Ihnen eine **ID** vorgegeben, die Sie nach Ihren Wünschen ändern können.

Wählen Sie "Kommunikationsrichtung "3: Lokal <-> Partner".

Der parametrisierte Verbindungsweg wird angezeigt.

Beide Angaben einer CPU-Nummer sind für den Betrieb dieses Treibers irrelevant.

Mit "OK" werden die Eingaben übernommen.

Speichern Sie die Verbindungsprojektierung und schließen Sie den Dialog "Verbindungen projektieren".

Beachten Sie, dass die Verbindungs-ID (Lokale ID) im Anwenderprogramm beim Aufruf der SFBs zu verwenden ist.

4.7 Parametrieren des ladbaren Treibers

Öffnen der Parametrieroberfläche CP-PtP

Durch Selektion der SIMATIC-Station und Doppelklick "Hardware" bzw. "Bearbeiten > Objekt öffnen" wird zunächst "Hardware konfigurieren" gestartet.

Markieren Sie den CP und wählen Sie **Bearbeiten > Objekteigenschaften**.

Nach der Auswahl der Schnittstelle (nur CP 441-2) und des Schnittstellen-Moduls (nur CP 441-2) gelangen Sie durch Anwahl der Schaltfläche **"Parameter"** in den Dialog zur Protokollauswahl.

Protokollauswahl

Neben den Standardprotokollen werden in der Selektbox auch alle installierten ladbaren Treiber angezeigt. Wählen Sie **"MODBUS Master"** für diesen Treiber.

Durch Doppelklick auf das Symbol Briefkasten für das Übertragungsprotokoll gelangen Sie in den Dialog zur Einstellung der protokollspezifischen Parameter.

Treiberspezifische Parameter

Die nachfolgend beschriebenen Parameter können bei diesem Treiber in den einzelnen Masken eingestellt werden.

4.7.1 Modbus Master-Protokoll

Übersicht Übertragungsparameter

Tabelle 4- 1 Geschwindigkeit, Zeichenrahmen

Parameter	Beschreibung	Wertebereich	Defaultwert
Baudrate	Geschwindigkeit der Datenübertragung in Bit/Sekunde	300 600 1200 2400 4800 9600 19200 38400 76800	9600
Zusätzliche Baudrate beim CP 341 mit diesen Bestellnummern: <ul style="list-style-type: none"> • 6ES7341-1xH01-0AE0 • 6ES7341-1xH02-0AE0 		57600	
Zusätzliche Baudraten beim CP 441-2 mit diesen Bestellnummern: <ul style="list-style-type: none"> • 6ES7441-2AA03-0AE0 • 6ES7441-2AA04-0AE0 		57600 115200	

Parameter	Beschreibung	Wertebereich	Defaultwert
Datenbits	Bit pro Zeichen	8	8
Stoppbits	Anzahl Stoppbits	1 2	1
Parität	Es wird kein Paritätsbit übertragen.	Keine	Gerade
	Die Anzahl der Datenbits wird auf ungerade ergänzt.	Ungerade	
	Die Anzahl der Datenbits wird auf gerade ergänzt.	Gerade	

Baudrate

Die Baudrate ist die Geschwindigkeit der Datenübertragung in Bit pro Sekunde (Baud).

Beachten Sie beim CP 441-2 die maximale Summenbaudrate. Die Summenbaudrate ergibt sich aus der Addition der parametrisierten Baudraten für die beiden Schnittstellen.

Die maximale Baudrate für die TTY-Schnittstelle beträgt 19200 Baud.

Datenbits

Die Anzahl der Datenbits beschreibt, auf wie viele Bits ein zu übertragendes Zeichen abgebildet wird.

Stoppbits

Die Anzahl der Stoppbits definiert den kleinsten zeitlichen Abstand zwischen zwei zu übertragenden Zeichen.

Parität

Das Paritätsbit dient der Datensicherheit; es ergänzt die Anzahl der übertragenen Datenbits je nach Parametrierung auf eine gerade oder ungerade Anzahl.

Wenn "keine" Parität eingestellt ist, wird kein Paritätsbit übertragen. Dies reduziert die Übertragungssicherheit.

Übersicht Protokoll-Parameter

Tabelle 4- 2 Protokoll-Parameter

Parameter	Beschreibung	Wertebereich	Defaultwert
Antwortüberwachungszeit	Überwachungszeit, mit der der Beginn der Antwort vom Slave überwacht wird.	5 bis 65500	2000
Betriebsartauswahl bei RS485 aktivieren	Auswahl von "Normal-Betrieb" auch in der Betriebsart Halbduplex (RS485) Zweidraht-Betrieb ermöglichen	ja nein	nein
Betriebsart	"Normal-Betrieb" "Störungsunterdrückung"	Normal-Betrieb Störungsunterdrückung	<ul style="list-style-type: none"> • bei "Voll duplex (RS422) Vierdraht-Betrieb": Normal-Betrieb • bei "Halbduplex (RS485) Zweidraht-Betrieb": Störungsunterdrückung
Multiplikator Zeichenverzugszeit	Multiplikationsfaktor für die baudratenabhängige Zeichenverzugszeit	1 bis 10	1

Antwortüberwachungszeit

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

Betriebsartauswahl bei RS485 aktivieren

Nach dem Aktivieren der "Betriebsartauswahl bei RS485" ist auch bei Auswahl von "Halbduplex (RS485) Zweidraht-Betrieb" im Register "Schnittstelle" das Umschalten der Betriebsart auf "Normal-Betrieb" möglich.

Normal-Betrieb

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

Das erste Zeichen eines Telegramms muss eine gültige Slave-Adresse sein.

Das Telegrammende wird ausschließlich durch den Ablauf der Zeichenverzugszeit erkannt.

Störungsunterdrückung

Wenn in dieser Betriebsart zu Beginn des Empfangstelegramms am Empfänger BREAK erkannt wird oder der CP-Schnittstellenbaustein Übertragungsfehler feststellt, führt dies im Anwenderprogramm nicht zu einer Fehlermeldung.

Ebenfalls ignoriert werden Übertragungsfehler bzw. BREAK, die nach dem Ende des Empfangstelegramms (CRC-Code) auftreten.

Der Beginn des Empfangstelegramms vom Koppelpartner wird mit der korrekt empfangenen Slaveadresse erkannt!

Multiplikator Zeichenverzugszeit

Wenn ein Koppelpartner die zeitlichen Anforderungen der MODBUS-Spezifikation nicht einhalten kann, besteht hier die Möglichkeit, die Zeichenverzugszeit ZVZ durch den Multiplikationsfaktor f_{MUL} zu vervielfachen. Verstellen Sie die Zeichenverzugszeit nur, 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 ZVZ (siehe Punkt "Übertragungsprotokoll")
- f_{MUL} : Multiplikationsfaktor

4.7.2 RS422/485 (X27) - Schnittstelle

Übersicht

Tabelle 4- 3 RS422/485 (X27) - Schnittstelle

Parameter	Beschreibung	Wertebereich	Defaultwert
Betriebsart	Festlegung, ob die RS422/485 (X27)-Schnittstelle voll duplex (RS422) oder halbduplex (RS485) betrieben werden soll	<ul style="list-style-type: none"> • Vollduplex (RS422) Vierdraht-Betrieb • Halbduplex (RS485) Zweidraht-Betrieb 	Vollduplex (RS422) Vierdraht-Betrieb
Vorbelegung der Empfangsleitung	<p>keine: Die Zweidraht-Leitung R(A), R(B) ist nicht vorbelegt. In diesem Fall sollte die Vorbelegung durch den Koppelpartner erfolgen.</p>	<ul style="list-style-type: none"> • Keine • Signal R(A) 5V / Signal R(B) 0V (Breakerkennung)¹ • Signal R(A) 0V / Signal R(B) 5V 	<ul style="list-style-type: none"> • bei "Vollduplex (RS422) Vierdraht-Betrieb": Signal R(A) 5V / Signal R(B) 0V (Breakerkennung) ¹ • bei "Halbduplex (RS485) Zweidraht-Betrieb": Signal R(A) 0V / Signal R(B) 5V
	<p>Signal R(A) 5V / Signal R(B) 0V (Breakerkennung): Bei dieser Vorbelegung ist bei "Vollduplex (RS422) Vierdraht-Betrieb" Breakerkennung möglich. Nicht anwählbar für "Halbduplex (RS485) Zweidraht-Betrieb".</p>		
	<p>Signal R(A) 0V / Signal R(B) 5V: Diese Vorbelegung entspricht dem Ruhezustand (kein Sender aktiv) bei "Halbduplex (RS485) Zweidraht-Betrieb". Bei dieser Vorbelegung ist keine Breakerkennung möglich.</p>		
<p>¹ Nur bei "Vollduplex (RS 422) Vierdraht-Betrieb"</p>			

"Vollduplex (RS422) Vierdraht-Betrieb"

In dieser Betriebsart wird auf der Sendeleitung T(A)-, T(B)+ gesendet und auf der Empfangsleitung R(A)-, R(B)+ empfangen.

Die Fehlerbehandlung erfolgt gemäß der beim Parameter "Treiber-Betriebsart" eingestellten Funktionalität (Normal-Betrieb oder Störungsunterdrückung).

"Halbduplex (RS485) Zweidraht-Betrieb"

In dieser Betriebsart schaltet der Treiber die 2-Draht-Empfangsleitung R(A)-, R(B)+ der Schnittstelle zwischen Sende- und Empfangsbetrieb um.

Als Voreinstellung werden in dieser Betriebsart alle erkannten Übertragungsfehler bzw. BREAK vor und nach Empfangstelegrammen ignoriert.

BREAK-Pegel während Telegrammpausen wird ebenfalls ignoriert.

Der Beginn eines Empfangstelegramms vom Koppelpartner wird mit der korrekt empfangenen Slaveadresse erkannt!

Als Vorbelegung der Empfangsleitung wird für einen Busteilnehmer die Einstellung "Signal R(A) 0V, Signal R(B) 5V" empfohlen. Alle anderen Busteilnehmer sollten bei "Vorbelegung der Empfangsleitung" "keine" wählen.

Vorbelegung der Empfangsleitung

- "Keine"

Die Zweidraht-Leitung R(A)-, R(B)+ ist nicht vorbelegt. In diesem Fall sollte die Vorbelegung durch den Koppelpartner erfolgen.

- Vorbelegung "R(A) 5V, R(B) 0V (Breakerkennung)"

Die Zweidrahtleitung R(A)-, R(B)+ wird vom CP wie folgt vorbelegt:

R(A) --> +5V, R(B) --> 0V ($V_A - V_B \geq +0,3V$).

Dies bedeutet, dass bei Leitungsbruch BREAK-Pegel am CP ansteht.

Diese Option ist nur im "Vollduplex (RS422) Vierdraht-Betrieb" anwählbar.

- Vorbelegung "R(A) 0V, R(B) 5V"

Die Zweidrahtleitung R(A)-, R(B)+ wird vom CP wie folgt vorbelegt:

R(A) --> 0V, R(B) --> +5V ($V_A - V_B \leq -0,3V$).

Dies bedeutet, dass bei Leitungsbruch bzw. im Ruhezustand, wenn niemand sendet, HIGH-Pegel am CP ansteht. Der Leitungszustand BREAK kann nicht erkannt werden.

Auswahl der Parameter

Wählen Sie die für Ihre Kopplung notwendigen Einstellungen aus und verlassen Sie die einzelnen Masken mit "OK".

4.8 Laden der Konfigurations- und Parametrierungsdaten beim CP 341

Datenhaltung

Beim Schließen von "Hardware Konfigurieren" werden die Daten automatisch in Ihr STEP 7 Projekt abgelegt.

Konfiguration und Parameter laden

Die Konfigurations- und Parametrierungsdaten können Sie nun online vom Programmiergerät auf die CPU laden. Mit dem Menübefehl **Zielsystem > Laden** werden die Daten an die CPU übertragen.

Die Baugruppenparameter des CP werden im Anlauf der CPU und bei jedem STOP/RUN-Übergang automatisch von der CPU an den CP übertragen, sobald dieser über den S7 300-Rückwandbus erreichbar ist.

Der Treibercode wird nicht in der CPU sondern direkt mit der Parametrieroberfläche auf den CP 341 im remanenten Speicher abgelegt. Beachten Sie, dass deshalb ein Baugruppentausch ohne PG nicht möglich ist.

Weitere Informationen

Im Benutzerhandbuch für STEP 7 ist Folgendes ausführlich beschrieben:

- Konfiguration und Parameter speichern.
- Konfiguration und Parameter in die CPU laden.
- Konfiguration und Parameter lesen, ändern, kopieren und drucken.

4.9 Treiber in den CP 341 laden

Voraussetzung

Sie haben eine Online-Verbindung zur CPU.

Treiber laden

1. Wählen Sie im Fenster "Punkt-zu-Punkt-Kopplung parametrieren" in der Klappliste "Protokoll" den gewünschten ladbaren Treiber.
2. Klicken Sie auf die Ikone "Treiber laden".

Im Fenster "Treiber laden auf CP341" sehen Sie die online auf der Baugruppe geladene Treiberversion und die von Ihnen offline auf dem PG gewählte Treiberversion.
3. Klicken Sie auf die Schaltfläche "Treiber laden" und bestätigen Sie mit "Ja".
Der Treiber wird in den CP 341 geladen.
Nach dem Laden wird die Information "Treiberversion online auf der Baugruppe" aktualisiert.
Wenn sich der von Ihnen geladene Treiber bereits auf dem CP 341 befindet, wird das Laden mit der Meldung "Treiber schon vorhanden" abgebrochen. In diesem Fall bestätigen Sie mit "OK" und schließen Sie das Fenster "Treiber laden auf CP341".

Wenn Treiberdateien fehlen oder fehlerhaft sind, erhalten Sie die Fehlermeldung "Treiber laden wurde von der Baugruppe zurückgewiesen". In diesem Fall müssen Sie die Treiber erneut installieren.

4.10 Laden der Konfigurations- und Parametrierungsdaten beim CP 441-2

Datenhaltung

Beim Schließen von "Hardware Konfigurieren" bzw. "Verbindungen projektieren" werden die Daten einschließlich Baugruppenparameter und Treibercode automatisch in Ihr STEP 7 Projekt abgelegt.

Konfiguration und Parameter laden

Die Konfigurations- und Parametrierungsdaten können Sie nun online vom Programmiergerät auf die CPU laden. Mit dem Menübefehl **Zielsystem > Laden** werden die Daten an die CPU übertragen.

Die Baugruppenparameter des CP werden im Anlauf der CPU automatisch von der CPU an den CP übertragen, sobald dieser über den S7 400-Rückwandbus erreichbar ist.

Weitere Informationen

Im Benutzerhandbuch für STEP 7 ist Folgendes ausführlich beschrieben:

- Konfiguration und Parameter speichern.
- Konfiguration und Parameter in die CPU laden.
- Konfiguration und Parameter lesen, ändern, kopieren und drucken.

4.11 Anlaufverhalten des CP

Einleitung

Der Anlauf des CP unterteilt sich in folgende Phasen:

- Initialisierung bei Netz-Ein des CP
- Parametrierung

Initialisierung

Sobald der CP an Spannung liegt, wird nach Durchlauf eines Hardwaretestprogramms die Firmware auf dem CP für den Betrieb vorbereitet.

Parametrierung

Bei der Parametrierung empfängt der CP die dem aktuellen Steckplatz zugeordneten Baugruppenparameter.

Der CP ist jetzt betriebsbereit.

4.12 Parametrierung "Anlauf der CPU"

Einleitung

Dieses Kapitel ist nur für den CP 441-2 relevant. Wenn Sie einen CP 341 einsetzen, können Sie dieses Kapitel überschlagen.

Hardware konfigurieren

Wählen Sie zur Vermeidung von Problemen beim Anlauf CPU – CP bei der **Parametrierung** der **CPU** mit "**Hardware Konfigurieren**" folgende Einstellung:

Nach Start der Parametrierung durch Doppelklick auf die CPU oder durch Markieren der CPU und Anwahl des Menübefehls **Bearbeiten > Objekteigenschaften** gelangen Sie in die Page "Eigenschaften - CPU".

Stellen Sie im Register "**Anlauf**" unter "**Überwachungszeit für**" beim Punkt "**Übertragen der Parameter an Baugruppe (100ms)**:" einen Mindestwert von *3000* (= 300 s) ein

Grund: Bei Parametrierung einer CP 441-2-Schnittstelle mit einem ladbaren Treiber wird neben der Parametrierung zusätzlich der Treibercode an den CP übertragen. Der gesamte Ladevorgang wird durch die oben genannte Zeit überwacht und muss ausreichend groß bemessen sein. .

Übertragungsprotokoll

Allgemeines

Zum Einsatz kommt eine codetransparente, asynchrone Halbduplexprozedur.
Die Datenübertragung erfolgt ohne Handshake.

Master-Slave-Beziehung

Der CP initiiert als Master die Übertragung und wartet nach Ausgabe des Anforderungstelegramms die parametrisierte Antwortüberwachungszeit auf ein Antworttelegramm vom Slave.

Telegrammaufbau

Der Datenverkehr "Master-Slave" bzw. "Slave-Master" beginnt mit der **Slaveadresse**, gefolgt vom **Funktionscode**. Dann werden die Daten übertragen. Der Aufbau des Datenfeldes ist abhängig vom verwendeten Funktionscode. Am Ende des Telegramms wird der CRC-Check übertragen.

ADDRESS	FUNCTION	DATA	CRC-CHECK
Byte	Byte	n Byte	2 Byte

ADDRESS	MODBUS-Slave-Adresse
FUNCTION	MODBUS-Funktions-Code
DATA	Telegramm Daten: Byte_Count, Coil_Number, Data
CRC-CHECK	Telegramm-Prüfsumme

Slaveadresse

Die Slaveadresse kann im Bereich von 1 bis 255 liegen. Mit der Adresse wird ein definierter Slave am Bus angesprochen.

Broadcast Message

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

Broadcast-Messages sind nur in Verbindung mit den schreibenden **Funktionscodes 05, 06, 15 bzw. 16** zulässig.

Bei einer Broadcast-Message wird vom Slave kein Antworttelegramm gesendet.

Funktionscode

Der Funktionscode definiert die Bedeutung des Telegramms. Durch den Funktionscode ist auch der Telegrammaufbau festgelegt. Folgende Funktionscodes werden vom CP unterstützt:

Funtionscode	Funktion gemäß MODBUS-Spezifikation
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

Datenfeld DATA

Im Datenfeld DATA werden die funktionscode-spezifischen Daten übertragen:

- Bytecount, Coil_Startaddress, Register_Startaddress; Number_of_Coils, Number_of_Registers,

Siehe hierzu Kapitel "Funktionscodes (Seite 47)".

CRC-Check

Den Telegrammabschluss bildet die zwei Byte lange CRC-16-Prüfsumme, gebildet nach dem Polynom $x^{16} + x^{15} + x^2 + 1$.

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

Telegrammende

Der ladbare Treiber erkennt das Telegrammende, wenn während der Zeit, die dreieinhalb Zeichen zur Übermittlung benötigen (3,5-fache Zeichenverzugszeit), keine Übertragung stattfindet (siehe MODBUS Protocol Reference Guide).

Dieser Telegrammende-TIME_OUT ist somit baudratenabhängig.

Baudrate	TIME_OUT
76800 Baud	0,5 ms
38400 Baud	1 ms
19200 Baud	2 ms
9600 Baud	4 ms
4800 Baud	8 ms
2400 Baud	16 ms
1200 Baud	32 ms
600 Baud	64 ms
300 Baud	128 ms

Im "Normal-Betrieb" wird nach Eintreffen des Telegrammende-TIME_OUTs das vom Koppel-partner empfangene Modbus-Telegramm ausgewertet und formal geprüft.

Bei "Störungsunterdrückung" wird das Telegrammende durch ein formal richtiges Empfangs-telegramm mit korrektem CRC-Code erkannt.

Exception Responses

Wenn der Slave einen Fehler im Anforderungstelegramm des Masters erkennt, z. B. Registeradresse nicht erlaubt, setzt er das höchstwertige Bit im Funktionscode des Antworttelegramms.

Nachfolgend wird ein Byte Fehlercode, Exception Code, übertragen, das die Fehlerursache beschreibt.

Sinn und Bedeutung der genannten Parameter entnehmen Sie der Beschreibung "GOULD MODICON Modbus Protocol".

Exception Code Telegramm

Das Fehlercode-Antworttelegramm vom Slave hat folgenden Aufbau:

- z. B. Slave-Adresse 5, Funktionscode 5, Exception Code 2

Antworttelegramm vom Slave EXCEPTION_CODE_xx:

05H	Slave-Adresse
85H	Funktionscode
02H	Exception Code (1...7)
xxH	CRC-Prüfcode "Low"
xxH	CRC-Prüfcode "High"

Wenn der Treiber ein Fehlercode-Antworttelegramm empfängt, wird der laufende Auftrag mit Fehler beendet.

Zusätzlich wird in den SYSTAT-Bereich eine dem empfangenen Fehlercode (Exception Code 1-7) entsprechende Fehlernummer eingetragen.

Es erfolgt kein Eintrag in einen BRCV-Ziel-Datenbaustein.

Folgende Fehlercodes sind gemäß MODBUS-Spezifikation definiert:

Fehlercode	Bedeutung nach MODBUS-Spezifikation	Ursache - Kurzbeschreibung *
1	Illegal function	Unzulässiger Funktionscode
2	Illegal data address	Unzulässige Daten-Adresse beim Slave
3	Illegal data value	Unzulässiger Daten-Wert beim Slave
4	Failure in associated device	Interner Fehler beim Slave
5	Acknowledge	Funktion wird ausgeführt
6	Busy, Rejected message	Slave ist nicht empfangsbereit
7	Negative acknowledgement	Funktion nicht ausführbar
* Detaillierte Ursache beim Slave untersuchen.		

RS 232C-Begleitsignale

Auf dem CP sind bei Einsatz des RS 232C-Schnittstellenmoduls folgende RS 232C-Begleitsignale vorhanden:

DCD	(Eingang)	Data carrier detect	Datenträger erkannt
DTR	(Ausgang)	Data terminal ready	CP betriebsbereit
DSR	(Eingang)	Data set ready	Kommunikationspartner betriebsbereit
RTS	(Ausgang)	Request to send	CP sendebereit
CTS	(Eingang)	Clear to send	Kommunikationspartner kann Daten vom CP empfangen (Antwort auf RTS = ON des CP)
RI	(Eingang)	Ring indicator	Rufzeichen

Nach dem Einschalten des CP befinden sich die Ausgangssignale im Zustand OFF (inaktiv).

Die Bedienung der Steuersignale DTR/DSR und RTS/CTS können Sie mit der Parametrieroberfläche **Punkt-zu-Punkt-Kopplung parametrieren** parametrieren oder über Funktionen (FBs) im Anwenderprogramm steuern.

RS 232C-Begleitsignale bedienen

Die RS 232C-Begleitsignale können bedient werden:

- Bei parametrierter automatischen Bedienung aller RS 232C-Begleitsignale
- Über die Funktionen FB V24_STAT und FB V24_SET

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 die Funktion FB V24_SET möglich!

Das Lesen aller RS 232C-Begleitsignale über die Funktion FB V24_STAT ist hingegen immer möglich.

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

Automatische Bedienung der Begleitsignale

Die Realisierung der automatischen Bedienung der RS 232C-Begleitsignale auf dem CP wird folgendermaßen durchgeführt:

- Sobald der CP durch Parametrierung in eine Betriebsart mit automatischen Bedienung der RS 232C-Begleitsignale gebracht wurde, setzt er die Leitungen RTS auf OFF und DTR auf ON (CP 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 entsprechenden Fehlermeldung abgebrochen.

- Wenn ein Sendeauftrag ansteht, 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.
- Wenn beim Senden innerhalb der Datenausgabewartezeit die Leitung CTS nicht auf ON gesetzt wird oder innerhalb des Sendevorgangs ein Wechsel von CTS auf OFF erfolgt, 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. Es wird nicht auf den Wechsel von CTS auf OFF gewartet.
- Ein Empfang von Daten über die RS 232C-Schnittstelle ist möglich, sobald die Leitung DSR auf ON gesetzt ist. Droht der Empfangspuffer des CP überzulaufen, erfolgt keine Reaktion des CP.
- Bei einem Wechsel von DSR = ON auf OFF wird sowohl ein laufender Sendeauftrag als auch das Empfangen von Daten mit einer Fehlermeldung abgebrochen. Im Diagnosepuffer des CP wird die Meldung "DSR = OFF" eingetragen (automatische Bedienung der V24-Signale).

Hinweis

Die automatische Bedienung der RS 232C-Begleitsignale ist nur im Halbduplexbetrieb möglich. 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 die Funktion FB V24_SET möglich!

Hinweis

Stellen Sie die "RTS-Wegnahmezeit" in der Parametrieroberfläche so ein, dass der die letzten Zeichen des Telegramms vollständig empfangen kann, bevor RTS und damit der Sendewunsch weggenommen wird. Dimensionieren Sie die "Datenausgabewartezeit" so, dass der Kommunikationspartner in Empfangsbereitschaft gehen kann, bevor die Zeit abläuft.

Zeitdiagramm

Das folgende Bild zeigt den zeitlichen Ablauf eines Sendeauftrags:

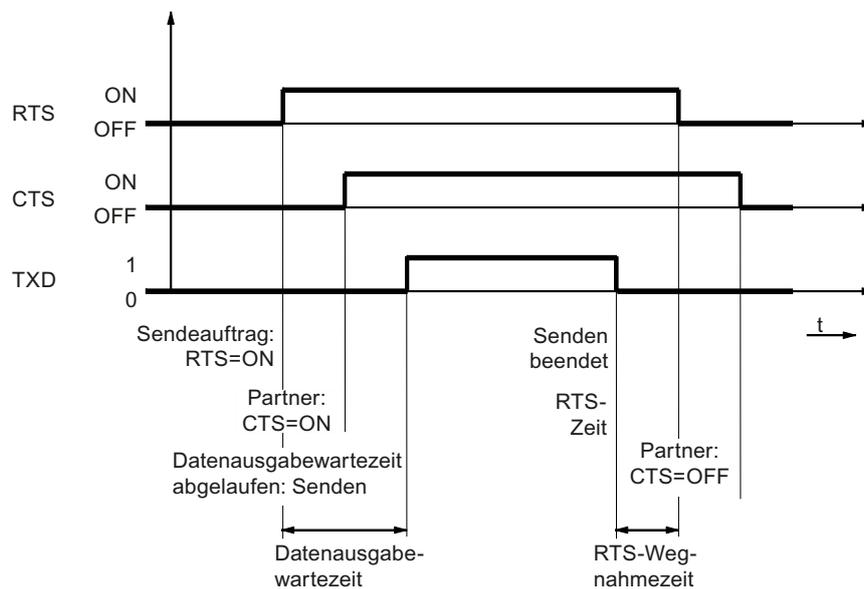


Bild 5-1 Zeitdiagramm bei automatischer Bedienung der RS 232C-Begleitsignale

Funktionscodes

6.1 Funktionscode 01 - Read Output Status

Funktion

Diese Funktion ermöglicht das Lesen einzelner Bits vom Slave.

Start-Adresse

Der Parameter **Bit-Startadresse** wird vom Treiber nicht überprüft und unverändert gesendet.

Bit-Anzahl

Als **Bit-Anzahl**, Number of coils, ist jeder Wert zwischen **1** und **2040** erlaubt.

SEND-Quell-DB

Aufbau des SEND-Quellbereichs:

Adresse	Name	Typ	Anfangswert	Kommentar
+0.0	address	BYTE	B#16#5	Slave-Adresse
+1.0	function	BYTE	B#16#1	Funktionscode
+2.0	bit_startadr	WORD	W#16#0040	Bit-Startadresse
+4.0	bit_anzahl	INT	16	Bit-Anzahl

Beispiel

Anforderungstelegramm FUNCTION 01:

05H	Slave-Adresse
01H	Funktionscode
00H	Bit-Startadresse "High"
40H	Bit-Startadresse "Low"
00H	Bit-Anzahl "High"
10H	Bit-Anzahl "Low"
xxH	CRC-Prüfcode "Low"
xxH	CRC-Prüfcode "High"

Antworttelegramm vom Slave FUNCTION 01:

05H	Slave-Adresse
01H	Funktionscode
02H	Bytezähler
01H	<Daten>
17H	<Daten>
xxH	CRC-Prüfcode "Low"
xxH	CRC-Prüfcode "High"

RCV-Ziel-DB

Inhalt des RCV-Zielbereichs:

Adresse	Name	Typ	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 1. empfangene Byte wird als Low-Byte des 1. Worts "data[1]" eingetragen, das 3. empfangene Byte als Low-Byte des 2. Worts "data[2]", usw.

Wenn weniger als 9 Bit gelesen werden oder wenn nur noch ein Low-Byte gelesen wurde, wird in das verbleibende High-Byte des letzten Worts der Wert **00H** eingetragen.

6.2 Funktionscode 02 - Read Input Status

Funktion

Diese Funktion ermöglicht das Lesen einzelner Bits vom Slave.

Start-Adresse

Der Parameter **Bit-Startadresse** wird vom Treiber nicht überprüft und unverändert gesendet.

Bit-Anzahl

Als **Bit-Anzahl**, Number of coils, ist jeder Wert zwischen **1** und **2040** erlaubt.

SEND-Quell-DB

Aufbau des SEND-Quellbereichs:

Adresse	Name	Typ	Anfangswert	Kommentar
+0.0	address	BYTE	B#16#5	Slave-Adresse
+1.0	function	BYTE	B#16#2	Funktionscode
+2.0	bit_startadr	WORD	W#16#0120	Bit-Startadresse
+4.0	bit_anzahl	INT	24	Bit-Anzahl

Beispiel

Anforderungstelegramm FUNCTION 02:

05H	Slave-Adresse
02H	Funktionscode
01H	Bit-Startadresse "High"
20H	Bit-Startadresse "Low"
00H	Bit-Anzahl "High"
18H	Bit-Anzahl "Low"
xxH	CRC-Prüfcode "Low"
xxH	CRC-Prüfcode "High"

Antworttelegramm vom Slave FUNCTION 02:

05H	Slave-Adresse
02H	Funktionscode
03H	Bytezähler
04H	<Daten>
26H	<Daten>
48H	<Daten>
xxH	CRC-Prüfcode "Low"
xxH	CRC-Prüfcode "High"

RCV-Ziel-DB

Inhalt des RCV-Zielbereichs:

Adresse	Name	Typ	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 1. empfangene Byte wird als Low-Byte des 1. Wortes "data[1]" eingetragen, das 3. empfangene Byte als Low-Byte des 2. Wortes "data[2]", usw.

Wenn weniger als 9 Bit gelesen werden oder wenn nur noch ein Low-Byte gelesen wurde, wird in das verbleibende High-Byte des letzten Worts der Wert **00H** eingetragen.

6.3 Funktionscode 03 - Read Output Registers

Funktion

Diese Funktion ermöglicht das Lesen einzelner Register vom Slave.

Start-Adresse

Der Parameter **Register-Startadresse** wird vom Treiber nicht überprüft und unverändert gesendet.

Register-Anzahl

Es können 1 bis **maximal 127 Register** gelesen werden (1 Register = zwei Byte).

SEND-Quell-DB

Aufbau des SEND-Quellbereichs:

Adresse	Name	Typ	Anfangswert	Kommentar
+0.0	address	BYTE	B#16#5	Slave-Adresse
+1.0	function	BYTE	B#16#3	Funktionscode
+2.0	reg_startadr	WORD	W#16#0040	Register-Startadresse
+4.0	reg_anzahl	INT	2	Register-Anzahl

Beispiel

Anforderungstelegramm FUNCTION 03:

05H	Slave-Adresse
03H	Funktionscode
00H	Register-Startadresse "High"
40H	Register-Startadresse "Low"
00H	Register-Anzahl "High"
02H	Register-Anzahl "Low"
xxH	CRC-Prüfcode "Low"
xxH	CRC-Prüfcode "High"

Antworttelegramm vom Slave FUNCTION 03:

05H	Slave-Adresse
03H	Funktionscode
04H	Bytezähler
21H	Registeradresse 40H Daten "High"
23H	Registeradresse 40H Daten "Low"
25H	Registeradresse 41H Daten "High"
27H	Registeradresse 41H Daten "Low"
xxH	CRC-Prüfcode "Low"
xxH	CRC-Prüfcode "High"

RCV-Ziel-DB

Inhalt des RCV-Zielbereichs:

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

6.4 Funktionscode 04 - Read Input Registers

Funktion

Diese Funktion ermöglicht das Lesen einzelner Register vom Slave.

Start-Adresse

Der Parameter **Register-Startadresse** wird vom Treiber nicht überprüft und unverändert gesendet.

Register-Anzahl

Es können 1 bis **maximal 127 Register** gelesen werden (1 Register = zwei Byte).

SEND-Quell-DB

Aufbau des SEND-Quellbereichs:

Adresse	Name	Typ	Anfangswert	Kommentar
+0.0	address	BYTE	B#16#5	Slave-Adresse
+1.0	function	BYTE	B#16#4	Funktionscode
+2.0	reg_startadr	WORD	W#16#0050	Register-Startadresse
+4.0	reg_anzahl	INT	3	Register-Anzahl

Beispiel

Anforderungstelegramm FUNCTION 04:

05H	Slave-Adresse
04H	Funktionscode
00H	Register-Startadresse "High"
50H	Register-Startadresse "Low"
00H	Register-Anzahl "High"
03H	Register-Anzahl "Low"
xxH	CRC-Prüfcode "Low"
xxH	CRC-Prüfcode "High"

Antworttelegramm vom Slave FUNCTION 04:

05H	Slave-Adresse
04H	Funktionscode
04H	Bytezähler
31H	Registeradresse 50H Daten "High"
32H	Registeradresse 50H Daten "Low"
33H	Registeradresse 51H Daten "High"
34H	Registeradresse 51H Daten "Low"
35H	Registeradresse 52H Daten "High"
36H	Registeradresse 52H Daten "Low"
xxH	CRC-Prüfcode "Low"
xxH	CRC-Prüfcode "High"

RCV-Ziel-DB

Inhalt des RCV-Zielbereichs:

Adresse	Name	Typ	Aktualwert	Kommentar
+0.0	data[1]	WORD	W#16#3132	Daten
+2.0	data[2]	WORD	W#16#3334	Daten
+4.0	data[3]	WORD	W#16#3536	Daten

6.5 Funktionscode 05 - Force Single Coil

Funktion

Mit dieser Funktion kann beim Slave ein einzelnes Bit gesetzt oder gelöscht werden.

Bit-Adresse

Der Parameter **Bit-Adresse** wird vom Treiber nicht überprüft und unverändert gesendet.

Bit-Status

Als **Bit-Status** sind folgende zwei Werte zulässig:

- FF00H = Bit Setzen
- 0000H = Bit Löschen

SEND-Quell-DB

Aufbau des SEND-Quellbereichs:

Adresse	Name	Typ	Anfangswert	Kommentar
+0.0	address	BYTE	B#16#5	Slave-Adresse
+1.0	function	BYTE	B#16#5	Funktionscode
+2.0	bit_address	WORD	W#16#0019	Bit-Adresse
+4.0	bit_state	WORD	W#16#FF00	Bit-Status

Beispiel

Anforderungstelegramm FUNCTION 05:

05H	Slave-Adresse
05H	Funktionscode
00H	Bit-Adresse "High"
19H	Bit-Adresse "Low"
FFH	Bit Setzen
00H	
xxH	CRC-Prüfcode "Low"
xxH	CRC-Prüfcode "High"

Antworttelegramm vom Slave FUNCTION 05:

05H	Slave-Adresse
05H	Funktionscode
00H	Bit-Adresse "High"
19H	Bit-Adresse "Low"
FFH	Bit-Status "High"
00H	Bit-Status "Low"
xxH	CRC-Prüfcode "Low"
xxH	CRC-Prüfcode "High"

6.6 Funktionscode 06 - Preset Single Register

Funktion

Mit diesem Befehl kann ein Slave-Register mit einem neuen Wert überschrieben werden.

Register-Adresse

Der Parameter **Register-Adresse** wird vom Treiber nicht überprüft und unverändert gesendet.

Register-Wert

Als **Register-Wert** kann jeder beliebige Wert verwendet werden.

SEND-Quell-DB

Aufbau des SEND-Quellbereichs:

Adresse	Name	Typ	Anfangswert	Kommentar
+0.0	address	BYTE	B#16#5	Slave-Adresse
+1.0	function	BYTE	B#16#6	Funktionscode
+2.0	reg_address	WORD	W#16#0180	Register-Adresse
+4.0	reg_value	WORD	W#16#3E7F	Register-Wert

Beispiel

Anforderungstelegramm FUNCTION 06:

05H	Slave-Adresse
06H	Funktionscode
01H	Register-Adresse "High"
80H	Register-Adresse "Low"
3EH	Register-Wert "High"
7FH	Register-Wert "Low"
xxH	CRC-Prüfcode "Low"
xxH	CRC-Prüfcode "High"

Antworttelegramm vom Slave FUNCTION 06:

05H	Slave-Adresse
06H	Funktionscode
01H	Register-Adresse "High"
80H	Register-Adresse "Low"
3EH	Register-Wert "High"
7FH	Register-Wert "Low"
xxH	CRC-Prüfcode "Low"
xxH	CRC-Prüfcode "High"

6.7 Funktionscode 07 - Read Exception Status

Funktion

Mit diesem Funktionscode können 8 Ereignis-Bits vom angeschlossenen Slave gelesen werden.

Die Anfangsbitnummer der Ereignis-Bits ist durch das angeschlossene Gerät festgelegt und muss somit nicht vom SIMATIC-Anwenderprogramm vorgegeben werden.

SEND-Quell-DB

Aufbau des SEND-Quellbereichs:

Adresse	Name	Typ	Anfangswert	Kommentar
+0.0	address	BYTE	B#16#5	Slave-Adresse
+1.0	function	BYTE	B#16#7	Funktionscode

Beispiel

Anforderungstelegramm FUNCTION 07:

05H	Slave-Adresse
07H	Funktionscode
xxH	CRC-Prüfcode "Low"
xxH	CRC-Prüfcode "High"

Antworttelegramm vom Slave FUNCTION 07:

05H	Slave-Adresse
07H	Funktionscode
3EH	<Daten>
xxH	CRC-Prüfcode "Low"
xxH	CRC-Prüfcode "High"

RCV-Ziel-DB

Inhalt des RCV-Zielbereichs:

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

Das einzelne Byte des Antworttelegramms wird vom Treiber im **High-Byte** im Ziel-DB data[1] eingetragen.

Das Low-Byte von data[1] bleibt unverändert.

Als Länge am Parameter LEN des BRCV wird der Wert 1 angezeigt.

6.8 Funktionscode 08 - Loop Back Diagnostic Test

Funktion

Diese Funktion dient zur Überprüfung der Kommunikations-Verbindung.
Bei diesem Funktionscode wird nur der **Diagnostic Code 0000** unterstützt !

Diagnostic-Code

Für den Parameter **Diagnostic-Code** ist nur der Wert 0000 zulässig.

Test-Wert

Als **Test-Wert** kann jeder beliebige Wert verwendet werden.

SEND-Quell-DB

Aufbau des SEND-Quellbereichs:

Adresse	Name	Typ	Anfangswert	Kommentar
+0.0	address	BYTE	B#16#5	Slave-Adresse
+1.0	function	BYTE	B#16#8	Funktionscode
+2.0	diag_code	WORD	W#16#0000	Diagnostic-Code
+4.0	test_value	WORD	W#16#A5C3	Test-Wert

Beispiel

Anforderungstelegramm FUNCTION 08:

05H	Slave-Adresse
08H	Funktionscode
00H	Diagnostic-Code "High"
00H	Diagnostic-Code "Low"
A5H	Test-Wert "High"
C3H	Test-Wert "Low"
xxH	CRC-Prüfcode "Low"
xxH	CRC-Prüfcode "High"

Antworttelegramm vom Slave FUNCTION 08:

05H	Slave-Adresse
08H	Funktionscode
00H	Diagnostic-Code "High"
00H	Diagnostic-Code "Low"
A5H	Test-Wert "High"
C3H	Test-Wert "Low"
xxH	CRC-Prüfcode "Low"
xxH	CRC-Prüfcode "High"

Der Slave muss das vom Master empfangene Anforderungstelegramm als Echo unverändert zurücksenden.

Das Antworttelegramm wird nicht in einen RCV-DB eingetragen.

6.9 Funktionscode 11 - Fetch Communications Event Counter

Funktion

Mit diesem Funktionscode kann ein 2 Byte "Status-Word" und ein 2 Byte "Event-Counter" vom Slave gelesen werden.

Sinn und Bedeutung der genannten Parameter sind der Beschreibung "GOULD MODICON Modbus Protocol" zu entnehmen.

SEND-Quell-DB

Aufbau des SEND-Quellbereichs:

Adresse	Name	Typ	Anfangswert	Kommentar
+0.0	address	BYTE	B#16#5	Slave-Adresse
+1.0	function	BYTE	B#16#0B	Funktionscode

Beispiel

Anforderungstelegramm FUNCTION 11:

05H	Slave-Adresse
0BH	Funktionscode
xxH	CRC-Prüfcode "Low"
xxH	CRC-Prüfcode "High"

Antworttelegramm vom Slave FUNCTION 11:

05H	Slave-Adresse
0BH	Funktionscode
FEH	Status-Word "High"
DCH	Status-Word "Low"
01H	Event-Counter "High"
08H	Event-Counter "Low"
xxH	CRC-Prüfcode "Low"
xxH	CRC-Prüfcode "High"

RCV-Ziel-DB

Inhalt des RCV-Zielbereichs:

Adresse	Name	Typ	Aktualwert	Kommentar
+0.0	data[1]	WORD	W#16#FEDC	Status-Word
+2.0	data[2]	WORD	W#16#0108	Event-Counter

6.10 Funktionscode 12 - Fetch Communications Event Log

Funktion

Mit diesem Funktionscode kann vom Slave Folgendes gelesen werden.

- 2 Byte "Status-Word"
- 2 Byte "Event-Counter",
- 2 Byte "Message-Counter" und
- 64 Byte "Event-Bytes"

Sinn und Bedeutung der genannten Parameter sind der Beschreibung "GOULD MODICON Modbus Protocol" zu entnehmen.

SEND-Quell-DB

Aufbau des SEND-Quellbereichs:

Adresse	Name	Typ	Anfangswert	Kommentar
+0.0	address	BYTE	B#16#5	Slave-Adresse
+1.0	function	BYTE	B#16#0C	Funktionscode

Beispiel

Anforderungstelegramm FUNCTION 12:

05H	Slave-Adresse
0CH	Funktionscode
xxH	CRC-Prüfcode "Low"
xxH	CRC-Prüfcode "High"

Antworttelegramm vom Slave FUNCTION 12:

05H	Slave-Adresse
0CH	Funktionscode
46H	Bytezähler
87H	Status-Word "High"
65H	Status-Word "Low"
01H	Event-Counter "High"
08H	Event-Counter "Low"
02H	Message-Counter "High"
20H	Message-Counter "Low"
01H	Event-Byte 1
12H	Event-Byte 2
:	:
C2H	Event-Byte 63

D3H	Event-Byte 64
xxH	CRC-Prüfcode "Low"
xxH	CRC-Prüfcode "High"

RCV-Ziel-DB

Inhalt des RCV-Zielbereichs:

Adresse	Name	Typ	Aktualwert	Kommentar
+0.0	data[1]	WORD	W#16#8765	Status-Word
+2.0	data[2]	WORD	W#16#0108	Event-Counter
+4.0	data[3]	WORD	W#16#0220	Message-Counter
+6.0	bytedata[1]	BYTE	B#16#01	Event-Byte 1
+7.0	bytedata[2]	BYTE	B#16#12	Event-Byte 2
:	:			:
+68.0	bytedata[63]	BYTE	B#16#C2	Event-Byte 63
+69.0	bytedata[64]	BYTE	B#16#D3	Event-Byte 64

6.11 Funktionscode 15 - Force Multiple Coils

Funktion

Mit diesem Funktionscode können bis zu 2040 Bits im Slave geändert werden.

Start-Adresse

Der Parameter **Bit-Startadresse** wird vom Treiber nicht überprüft und unverändert gesendet.

Bit-Anzahl

Als **Bit-Anzahl**, Number of coils, ist jeder Wert zwischen **1** und **2040** erlaubt.

Hiermit wird festgelegt, wie viele Bits im Slave überschrieben werden sollen.

Der im Anforderungstelegramm enthaltene Parameter "Bytezähler" wird vom Treiber aufgrund des übergebenen Parameters "Bit-Anzahl" gebildet.

SEND-Quell-DB

Aufbau des SEND-Quellbereichs:

Adresse	Name	Typ	Anfangswert	Kommentar
+0.0	address	BYTE	B#16#5	Slave-Adresse
+1.0	function	BYTE	B#16#0F	Funktionscode
+2.0	bit_startadr	WORD	W#16#0058	Bit-Startadresse
+4.0	bit_anzahl	INT	10	Bit-Anzahl
+6.0	coil_state[1]	WORD	W#16#EFCD	Status Coil 5FH..58H/57H..50H

Beispiel

Anforderungstelegramm FUNCTION 15:

05H	Slave-Adresse
0FH	Funktionscode
00H	Bit-Adresse "High"
50H	Bit-Adresse "Low"
00H	Bit-Anzahl "High"
0AH	Bit-Anzahl "Low"
02H	Bytezähler
CDH	Status Coil 50H..57H
EFH	Status Coil 58H..59H
xxH	CRC-Prüfcode "Low"
xxH	CRC-Prüfcode "High"

Antworttelegramm vom Slave FUNCTION 15:

05H	Slave-Adresse
0FH	Funktionscode
00H	Bit-Adresse "High"
50H	Bit-Adresse "Low"
00H	Bit-Anzahl "High"
0AH	Bit-Anzahl "Low"
xxH	CRC-Prüfcode "Low"
xxH	CRC-Prüfcode "High"

6.12 Funktionscode 16 - Preset Multiple Registers

Funktion

Der Funktionscode 16 ermöglicht es, mit einem Anforderungstelegramm bis zu **127 Register** im Slave zu überschreiben.

Start-Adresse

Der Parameter **Register-Startadresse** wird vom Treiber nicht überprüft und unverändert gesendet.

Register-Anzahl

Es können **1 bis maximal 127 Register** gelesen werden (1 Register = zwei Byte).

Der im Anforderungstelegramm enthaltene Parameter "Bytezähler" wird vom Treiber aufgrund des übergebenen Parameters "Register-Anzahl" gebildet.

SEND-Quell-DB

Aufbau des SEND-Quellbereichs:

Adresse	Name	Typ	Anfangswert	Kommentar
+0.0	address	BYTE	B#16#5	Slave-Adresse
+1.0	function	BYTE	B#16#10	Funktionscode
+2.0	reg_startadr	WORD	W#16#0060	Register-Startadresse
+4.0	reg_anzahl	INT	3	Register-Anzahl
+6.0	reg_data[1]	WORD	W#16#41A1	Register-Daten
+8.0	reg_data[2]	WORD	W#16#42A2	Register-Daten
+10.0	reg_data[3]	WORD	W#16#43A3	Register-Daten

Beispiel

Anforderungstelegramm FUNCTION 16:

05H	Slave-Adresse
10H	Funktionscode
00H	Register-Adresse "High"
60H	Register-Adresse "Low"
00H	Register-Anzahl "High"
03H	Register-Anzahl "Low"
06H	Bytezähler
41H	<reg_data[1]> "High"
A1H	<reg_data[1]> "Low"
42H	<reg_data[2]> "High"

A2H	<reg_data[2]> "Low"
43H	<reg_data[3]> "High"
A3H	<reg_data[3]> "Low"
xxH	CRC-Prüfcode "Low"
xxH	CRC-Prüfcode "High"

Antworttelegramm vom Slave FUNCTION 16:

05H	Slave-Adresse
10H	Funktionscode
00H	Register-Adresse "High"
60H	Register-Adresse "Low"
00H	Register-Anzahl "High"
03H	Register-Anzahl "Low"
xxH	CRC-Prüfcode "Low"
xxH	CRC-Prüfcode "High"

Schnittstelle CPU - CP

7.1 Schnittstelle CPU - CP für CP 341

Verwendete SFBs

Die Datenübergabe zwischen CP und CPU erfolgt mit den FBs **P_SND_RK** und **P_RCV_RK**.

Der FB **P_SND_RK** wird mit einer Flanke am Eingang **REQ** aktiviert, wenn Daten ausgegeben werden sollen.

Der FB **P_RCV_RK** wird mit **EN_R=1** empfangsbereit gemacht.

Ein **P_RCV_RK** ist bei allen lesenden Funktionscodes notwendig.

Gleichzeitig bearbeitbare Aufträge

Im Anwenderprogramm darf für jeden eingesetzten CP 341 gleichzeitig immer nur ein FB **P_SND_RK** und ein FB **P_RCV_RK** aufgerufen werden.

Einbindung der Aufträge in das Anwenderprogramm

Zur vollständigen Abwicklung eines einzigen MODBUS-Master-Auftrags müssen Sie die Bausteine **P_SND_RK** und **P_RCV_RK** jeweils mehrfach aufrufen. Die Anzahl der erforderlichen Aufrufe ist abhängig vom zu schreibenden bzw. vom zu lesenden Datenvolumen. Sie beschleunigen den Datenaustausch, wenn Sie die Bausteine **P_SND_RK** und **P_RCV_RK** im OB1-Zyklus der CPU aufrufen. Bei einem Aufruf der Bausteine in "langsamen" Weckalarmen dauert der Datenaustausch länger.

7.1.1 Datenübergabe von der CPU an den CP mit P_SND_RK (CP 341)

Aktivierung

Die Ausführung eines MODBUS-Funktionscodes wird mit einem SFB **P_SND_RK** mit einer **Flanke** am Eingang **REQ** aktiviert.

Am Parameter SF geben Sie 'S' für SEND an.

An LADDR wird die logische Baugruppenadresse angegeben.

Als Bereichstyp der Partner-CPU muss 'X' für erweiterte Datenbausteine angegeben werden. Für die übrigen Parameter der Partner-CPU (R_...) müssen keine Werte angegeben werden.

Dadurch werden die zur Ausführung des Funktionscodes nötigen Parameter an den Treiber übergeben.

Datenquelle

Bei Aktivierung des P_SND_RK wird der mit den Parametern **DB_NO** und **DBB_NO** angegebene **Quell-Datenbereich** mit der Länge **LEN** an den CP übertragen.

Längenangabe

Die Länge **LEN** ist abhängig vom verwendeten Funktionscode.

Funktionscode	Länge LEN in Byte
01	6
02	6
03	6
04	6
05	6
06	6
07	2
08	6
11	2
12	2
15	>6
16	>6

Wenn bei den einzelnen Funktionscodes die aufgelisteten Datenmengen nicht übergeben werden, wird der Auftrag nicht ausgeführt und der P_SND_RK mit einer Flanke am Ausgang ERROR beendet.

SEND Quell-DB

Im Quell-Datenbereich müssen die zur Ausführung eines Funktionscodes benötigten Parameter als Nutzdaten eingetragen sein.

Der detaillierte Aufbau des jeweiligen P_SND_RK-Quell-DB ist bei der Beschreibung der einzelnen Funktionscodes im Kapitel "Funktionscodes (Seite 47)" dargestellt.

Telegrammgenerierung

Die Anforderungs-Telegramme an den Slave werden anhand der übergebenen P_SND_RK-Quell-Daten aufgebaut und vom CP gesendet.

Zunächst überprüft der Treiber, ob die am P_SND_RK angegebene Länge LEN mit der Länge für diesen Funktionscode korrespondiert.

Wenn nicht, wird der Auftrag nicht ausgeführt und mit einer Flanke am Ausgang ERROR des P_SND_RK beendet.

Bei Verwendung von anderen Funktionscodes als den aufgeführten wird der aktivierte Auftrag ebenfalls nicht ausgeführt und mit ERROR am P_SND_RK beendet.

Die im Anforderungs-Telegramm enthaltenen Elemente "Bytezähler" und "CRC-Check" werden vom CP gebildet, ein Eintrag in den P_SND_RK-Quell-DB ist nicht erforderlich.

Auftragsabschluss bei schreibenden Funktionen

Bei schreibenden Funktionscodes wird der aktivierte P_SND_RK nach dem fehlerfreien Empfang des Antworttelegramms beendet. Dies wird dem SIMATIC-Anwenderprogramm mit einer Flanke am Ausgang **DONE** des P_SND_RK mitgeteilt.

Wenn beim Telegrammverkehr **Fehler** erkannt wurden oder der Slave mit einem **Fehlercode**-Antworttelegramm antwortet, wird dies durch eine Flanke am Ausgang **ERROR** gemeldet.

Auftragsabschluss bei lesenden Funktionen

Bei lesenden Funktionen wird der aktivierte P_SND_RK nach dem fehlerfreien Empfang des Antworttelegramms **und** vollständiger Übergabe der empfangenen Daten an die CPU beendet.

Dies wird dem SIMATIC-Anwenderprogramm mit einer Flanke am Ausgang **DONE** des P_SND_RK mitgeteilt.

Zu diesem Zeitpunkt sind die empfangenen Daten also bereits in der CPU vorhanden.

Wenn beim Telegrammverkehr **Fehler** erkannt wurden oder der Slave mit einem **Fehlercode**-Antworttelegramm antwortet, wird dies durch eine Flanke am Ausgang **ERROR** gemeldet.

In diesem Fall werden keine Empfangsdaten an die CPU übergeben.

STATUS-Eintrag bei Auftragsende

In den Fällen, in denen ein Auftrag mit **ERROR** am P_SND_RK beendet wird, erfolgt ein zusätzlicher Eintrag eines Fehlercodes im Parameter **STATUS**.

Anhand dieses Fehlercodes kann die genaue Fehlerursache ermittelt werden.

7.1.2 Datenübergabe vom CP an die CPU mit P_RCV_RK (CP 341)

Voraussetzung

Alle **lesenden** Funktionscodes benötigen einen P_RCV_RK.

Datenziel

Bei aktivierter Empfangsbereitschaft des FB P_RCV_RK übernimmt dieser die empfangenen Daten vom CP und trägt sie in das durch die Parameter **DB_NO** und **DBB_NO** spezifizierte Datenziel ein.

Anzeige des Daten-Empfangs

Der Empfang von Daten in der CPU wird dem Anwender durch eine Flanke am Ausgang **NDR** mitgeteilt.

Zu diesem Zeitpunkt steht im Parameter **LEN** die Länge des empfangenen Datenblocks.

Der Abschluss des gesamten Modbus-Auftrags kann am Ausgang **DONE** des FB P_SND_RK erkannt werden.

Verhalten im Fehlerfall

Bei Empfangs- oder Übergabefehlern werden keine Daten an die CPU übergeben. Der P_SND_RK wird in diesem Fall mit einer Flanke am Ausgang **ERROR** beendet.

P_RCV_RK Ziel-DB

Die bei einem lesenden Funktionscode empfangenen Nutzdaten werden in den P_RCV_RK-Zielbereich eingetragen.

Der detaillierte Aufbau des jeweiligen P_RCV_RK-Ziel-DB ist bei der Beschreibung der einzelnen Funktionscodes im Kapitel " Funktionscodes (Seite 47) " dargestellt.

Die Länge der eingetragenen Daten wird am Parameter **LEN** des P_RCV_RK angezeigt.

7.2 Schnittstelle CPU - CP für CP 441-2

Verwendete SFBs

Die Datenübergabe zwischen CP und CPU erfolgt mit den SFBs **BSEND** und **BRCV**.

Der SFB BSEND wird mit einer Flanke am Eingang **REQ** aktiviert, wenn Daten ausgegeben werden sollen.

Der SFB BRCV wird mit **EN_R=1** empfangsbereit gemacht.

Ein BRCV ist bei allen lesenden Funktionscodes notwendig.

7.2.1 Datenübergabe von der CPU an den CP mit BSEND (CP 441-2)

Kommunikationsverbindung

Der **Parameter ID** beschreibt die eindeutige Kommunikationsverbindung zu einem Kommunikationspartner. Hier muss die lokale ID aus der Verbindungsprojektierung angegeben werden.

Bausteinbeziehung

Der **Parameter R_ID** beschreibt die eindeutige Bausteinbeziehung innerhalb einer Kommunikationsverbindung.

Bei diesem Treiber können für die R_ID am BSEND **beliebige** Werte **0..255** eingetragen werden.

Bei lesenden Aufträgen muss am zugehörigen BRCV die gleiche R_ID wie beim BSEND parametrisiert sein.

Aktivierung

Die Ausführung eines MODBUS-Funktionscodes wird mit einem SFB **BSEND** mit einer **Flanke** am Eingang **REQ** aktiviert.

Dadurch werden die zur Ausführung des Funktionscodes nötigen Parameter an den Treiber übergeben.

Datenquelle

Bei Aktivierung des BSEND wird der am Parameter **SD_1** angegebene **Quell-Datenbereich** mit der Länge **LEN** an den CP übertragen.

Längenangabe

Die Länge **LEN** ist abhängig vom verwendeten Funktionscode.

Funktionscode	Länge LEN in Byte
01	6
02	6
03	6
04	6
05	6
06	6
07	2
08	6
11	2
12	2
15	>6
16	>6

Wenn bei den einzelnen Funktionscodes die aufgelisteten Datenmengen nicht übergeben werden, wird der Auftrag nicht ausgeführt und der BSEND mit einer Flanke am Ausgang ERROR beendet.

BSEND Quell-DB

Im Quell-Datenbereich müssen die zur Ausführung eines Funktionscodes benötigten Parameter als Nutzdaten eingetragen sein.

Der detaillierte Aufbau des jeweiligen BSEND-Quell-DB ist bei der Beschreibung der einzelnen Funktionscodes im Kapitel "Funktionscodes" dargestellt.

Telegrammgenerierung

Die Anforderungs-Telegramme an den Slave werden anhand der übergebenen BSEND-Quell-Daten aufgebaut und vom CP gesendet.

Zunächst überprüft der Treiber, ob die am BSEND angegebene Länge LEN mit der Länge für diesen Funktionscode korrespondiert.

Wenn nicht, wird der Auftrag nicht ausgeführt und mit einer Flanke am Ausgang ERROR des BSEND beendet.

Bei Verwendung von anderen Funktionscodes als den aufgeführten wird der aktivierte Auftrag ebenfalls nicht ausgeführt und mit ERROR am BSEND beendet.

Die im Anforderungs-Telegramm enthaltenen Elemente "Bytezähler" und "CRC-Check" werden vom CP gebildet, ein Eintrag in den BSEND-Quell-DB ist nicht erforderlich.

Auftragsabschluss bei schreibenden Funktionen

Bei schreibenden Funktionscodes wird der aktivierte BSEND nach dem fehlerfreien Empfang des Antworttelegramms beendet.

Dies wird dem SIMATIC-Anwenderprogramm mit einer Flanke am Ausgang **DONE** des BSEND mitgeteilt.

Wenn beim Telegrammverkehr **Fehler** erkannt wurden oder der Slave mit einem **Fehlercode**-Antworttelegramm antwortet, wird dies durch eine Flanke am Ausgang **ERROR** gemeldet.

Auftragsabschluss bei lesenden Funktionen

Bei lesenden Funktionen wird der aktivierte BSEND nach dem fehlerfreien Empfang des Antworttelegramms **und** vollständiger Übergabe der empfangenen Daten an die CPU beendet.

Dies wird dem SIMATIC-Anwenderprogramm mit einer Flanke am Ausgang **DONE** des BSEND mitgeteilt.

Zu diesem Zeitpunkt sind die empfangenen Daten also bereits in der CPU vorhanden.

Wurden beim Telegrammverkehr **Fehler** erkannt oder antwortete der Slave mit einem **Fehlercode**-Antworttelegramm, wird dies durch eine Flanke am Ausgang **ERROR** gemeldet.

In diesem Fall werden keine Empfangsdaten an die CPU übergeben.

SYSTAT-Eintrag bei Auftragsende

In den Fällen, in denen ein Auftrag mit **ERROR** am BSEND beendet wird, erfolgt ein zusätzlicher Eintrag eines Fehlercodes in den **SYSTAT**-Bereich.

Anhand dieses Fehlercodes kann die genaue Fehlerursache ermittelt werden.

7.2.2 Datenübergabe vom CP an die CPU mit BRCV (CP 441-2)

Kommunikationsverbindung

Der **Parameter ID** beschreibt die eindeutige Kommunikationsverbindung zu einem Kommunikationspartner. Hier muss die lokale ID aus der Verbindungsprojektierung angegeben werden.

Bausteinbeziehung

Der **Parameter R_ID** beschreibt die eindeutige Bausteinbeziehung innerhalb einer Kommunikationsverbindung.

Alle **lesenden** Funktionscodes benötigen einen BRCV.

Die am **BRCV** parametrisierte **R_ID** muss die **gleiche R_ID** wie beim zugehörigen BSEND aufweisen, mit der dieser Auftrag aktiviert wurde (beliebige Werte 0 ... 255).

So können im SIMATIC-Anwenderprogramm mehrere BSEND-/BRCV-Paare programmiert werden.

Die vom Modbus-Slave empfangenen Antwort-Telegramme werden dann in unterschiedliche Zielbereiche abgelegt, in Abhängigkeit von der bei diesem Auftrag verwendeten R_ID.

Datenziel

Bei aktivierter Empfangsbereitschaft des SFB BRCV übernimmt dieser die empfangenen Daten vom CP und trägt sie in das am Parameter **RD_1** angegebene Datenziel ein. Das Datenziel ist somit variabel.

Anzeige des Daten-Empfangs

Der Empfang von Daten in der CPU wird dem Anwender durch eine Flanke am Ausgang **NDR** mitgeteilt.

Zu diesem Zeitpunkt steht im Parameter **LEN** die Länge des empfangenen Datenblocks.

Der Abschluss des gesamten Modbus-Auftrags kann am Ausgang **DONE** des SFB BSEND erkannt werden.

Verhalten im Fehlerfall

Bei Empfangs- oder Übergabefehlern werden keine Daten an die CPU übergeben. Der BSEND wird in diesem Fall mit einer Flanke am Ausgang **ERROR** beendet.

BRCV Ziel-DB

Die bei einem lesenden Funktionscode empfangenen Nutzdaten werden in den BRCV-Zielbereich eingetragen.

Der detaillierte Aufbau des jeweiligen BRCV-Ziel-DB ist bei der Beschreibung der einzelnen Funktionscodes im Kapitel Funktionscodes (Seite 47)"Funktionscodes" dargestellt.

Die Länge der eingetragenen Daten wird am Parameter **LEN** des BRCV angezeigt.

Diagnose beim Treiber

Diagnosefunktionen

Die Diagnosefunktionen des CP erlauben Ihnen eine schnelle Lokalisierung aufgetretener Fehler. Folgende Diagnosemöglichkeiten stehen zur Verfügung:

- Diagnose über die Anzeigeelemente des CP
- Diagnose über den STATUS-Ausgang der Funktionsbausteine
- Diagnose über den Fehlermeldebereich SYSTAT (nur beim CP 441-2)
- Diagnosepuffer des CP

Anzeigeelemente (LED)

Die Anzeigeelemente informieren Sie über den Betriebszustand bzw. über mögliche Fehlerzustände des CP. Die Anzeigeelemente geben Ihnen einen ersten Überblick über aufgetretene interne bzw. externe Fehler sowie schnittstellenspezifische Fehler.

STATUS-Ausgang der FBs / SFBs

Für eine Fehlerdiagnose besitzt jeder Funktionsbaustein und jeder Systemfunktionsbaustein einen STATUS-Ausgang. Durch Lesen des STATUS-Ausgangs erhalten Sie Aussagen zu Fehlern, die bei der Kommunikation aufgetreten sind. Den STATUS-Parameter können Sie im Anwenderprogramm auswerten.

Fehlermeldebereich SYSTAT (nur für CP 441-2)

Der Fehlermeldebereich SYSTAT ist ein Speicherbereich auf dem CP 441-2, in den alle vom CP erkannten Fehler und Ereignisse detailliert eingetragen werden. Über die Programmierung des Systemfunktionsbausteins STATUS im Anwenderprogramm können Sie den SYSTAT-Bereich auslesen.

Diagnosepuffer des CP

Alle Fehler und Ereignisse, die in Kapitel "Tabelle der Fehler/Ereignisse (Seite 86) beschrieben sind, werden auch in den Diagnosepuffer des CP eingetragen. Wie Sie den Diagnosepuffer auslesen können, können Sie dem Handbuch des CP entnehmen.

8.1 Diagnosemöglichkeiten beim CP 341

8.1.1 Diagnose über die Anzeigeelemente des CP 341

Anzeigeelemente

Über die Anzeigeelemente des CP 341 erhalten Sie Auskunft über den CP 341. Es gibt folgende Anzeigen:

Sammelstörungsanzeigen	
• SF (rot)	Ein Fehler ist aufgetreten oder Neuparametrierung
Sonderanzeigen	
• TXD (grün)	Senden aktiv; leuchtet auf, wenn der CP 341 Nutzdaten über die Schnittstelle sendet.
• RXD (grün)	Empfangen aktiv; leuchtet auf, wenn der CP 341 Nutzdaten über die Schnittstelle empfängt.

Sammelstörungsanzeige SF

Die Sammelstörungsanzeige SF leuchtet immer nach NETZ-EIN und erlischt nach der Initialisierung. Wenn für den CP 341 Parametrierungsdaten erstellt wurden, leuchtet die SF-LED während der Neuparametrierung noch einmal kurz auf.

Die Sammelstörungsanzeige SF leuchtet, wenn folgende Fehler aufgetreten sind:

- Hardwarefehler
- Firmwarefehler
- Parametrierfehler
- BREAK (Die Empfangsleitung zwischen CP 341 und Kommunikationspartner ist unterbrochen.)

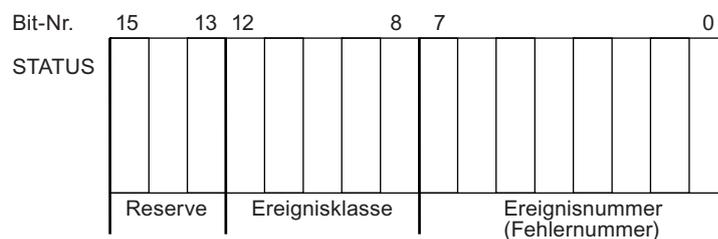
8.1.2 Diagnosemeldungen der Funktionsbausteine des CP 341

Einleitung

Für eine Fehlerdiagnose besitzt jeder Funktionsbaustein einen STATUS-Parameter. Jede STATUS-Meldungsnummer hat unabhängig vom verwendeten Funktionsbaustein die gleiche Bedeutung.

Nummernschema Ereignisklasse/Ereignisnummer

Das folgende Bild zeigt den Aufbau des Parameters STATUS.



Die einzelnen Fehler/Ereignisse sind im Kapitel "Tabelle der Fehler/Ereignisse (Seite 86)" aufgelistet.

8.2 Diagnosemöglichkeiten beim CP 441-2

8.2.1 Diagnose über die Anzeigeelemente des CP 441-2

Anzeigeelemente

Über die Anzeigeelemente des CP 441-2 erhalten Sie Auskunft über den CP 441-2. Es gibt folgende Anzeigen:

Sammelstörungsanzeigen	
INTF	Interner Fehler
EXTF	Externer Fehler
Sonderanzeigen	
TXD	Senden aktiv; leuchtet auf, wenn der CP 441-2 Nutzdaten über die Schnittstelle sendet.
RXD	Empfangen aktiv; leuchtet auf, wenn der CP 441-2 Nutzdaten über die Schnittstelle empfängt.
Schnittstellenfehleranzeigen	
FAULT	Schnittstellenfehler

Fehlermeldungen der Anzeigeelemente

Die folgende Tabelle beschreibt die Fehlermeldungen der Anzeigeelemente.

Fehleranzeige	Fehlerbeschreibung	Abhilfe
INTF leuchtet	CP 441-2 meldet internen Fehler, z. B. Hardware- oder Softwarefehler.	Programmieren Sie den SFB STATUS für detaillierte Informationen.
EXTF leuchtet	CP 441-2 meldet externen Fehler, z. B. BREAK auf der Empfangsleitung.	Programmieren Sie den SFB STATUS für detaillierte Informationen.
FAULT aus	Schnittstelle betriebsbereit bzw. Schnittstellenmodul nicht gesteckt	-
FAULT blinkt langsam	Schnittstelle ist initialisiert und betriebsbereit, aber keine Kommunikation über den S7 400-Rückwandbus möglich.	Überprüfen Sie die Konfiguration und Verbindungsprojektion.
FAULT blinkt schnell	Parameter falsch oder falsches bzw. fehlerhaftes Schnittstellenmodul gesteckt. (Modul und Schnittstellenparameter passen nicht zusammen).	Überprüfen Sie die Parametereinstellung in der Parametrieroberfläche und/oder das Schnittstellenmodul.
FAULT leuchtet	Keine Schnittstellenparameter vorhanden oder schwerer Fehler im Modul (Hardware).	Nehmen Sie die Parametrierung mit der Parametrieroberfläche vor bzw. prüfen Sie das Schnittstellenmodul.

8.2.2 Diagnosemeldungen der Systemfunktionsbausteine des CP 441-2

Einleitung

Für eine Fehlerdiagnose besitzt jeder Systemfunktionsbaustein einen STATUS-Parameter. Jede STATUS-Meldungsnummer hat unabhängig vom verwendeten Systemfunktionsbaustein die gleiche Bedeutung. In der Tabelle sind die für den CP wichtigsten STATUS-Meldungen beschrieben. Die vollständige und aktuelle Beschreibung der STATUS-Meldungen finden Sie im Referenzhandbuch "Systemsoftware für S7-300/400 System- und Standardfunktionen".

Meldungen am STATUS-Ausgang des SFB

STATUS	Fehlerbeschreibung
0	Kein Fehler
1	Kommunikationsprobleme zwischen CP und CPU
2	Negative Quittung, Funktion nicht ausführbar, z. B. Koppelpartner antwortet nicht oder quittiert negativ.
3	R-ID auf dieser Kommunikationsverbindung nicht bekannt, Betriebsmittel nicht vorhanden.
4	Die Anzahl der Datenbereiche oder einzelne Datentypen stimmen nicht überein.
5	Resetanforderung eingetroffen
6	Remoter Baustein befindet sich im Zustand disabled.
7	Remoter Partner ist im falschen Zustand.
8	Der Zugriff auf ein remotes Objekt wurde abgelehnt, Zugriffsfehler im Server aufgetreten (GET/PUT).
9	Overrun-Warnung (ERROR=0): Empfangsdaten wurden von neueren Daten überschrieben.
10	Zugriff auf lokalen Anwenderspeicher nicht möglich, z. B. DB gelöscht.
11	Warnung (ERROR=0): Ein neuer Auftrag ist unwirksam, da der vorangegangene Auftrag noch nicht abgeschlossen ist.
12	Instanz ist inkompatibel zum Systemcall; keine Instanz, sondern normaler DB aufgerufen.
13	Fehler in der Formatbeschreibung
14	Die referenzierte Verbindung (Applikationsbeziehung) existiert nicht, die ID ist unbekannt. Es muss die lokale ID aus der Verbindungsprojektierung angegeben werden.
15	Die über die ID referenzierte Verbindung ist aufgebaut.
16	Die Verbindung kann wegen Ressourcenmangel nicht aufgebaut werden.

STATUS-Ausgang anzeigen und auswerten

Den STATUS-Ausgang der Systemfunktionsbausteine können Sie über die Variablen-tabelle von STEP 7 anzeigen und auswerten.

Hinweis

Durch Lesen des SYSTAT-Bereichs mit dem STATUS-Auftrag erhalten Sie detaillierte Aussagen zu Fehlern und Ereignissen, die bei der Kommunikation zwischen dem CP, der zugeordneten CPU und dem angeschlossenen Kommunikationspartner aufgetreten sind.

8.2.3 Diagnose über den Fehlermeldebereich SYSTAT des CP 441-2

Einleitung

Der Fehlermeldebereich SYSTAT ist ein Datenbereich des CP 441-2, in den alle vom CP erkannten Fehler und Ereignisse detailliert eingetragen werden. Der SYSTAT-Bereich enthält für jede Schnittstelle sechs Ereignisse sowie Angaben zum Betriebszustand des CP und zum Zustand des SYSTAT-Bereichs.

Auslesen des SYSTAT-Bereichs

Der SYSTAT-Bereich kann mit dem SFB STATUS ausgelesen werden. Am Parameter ID ist die gewünschte Verbindung einzutragen, von der die aktuellen Ereignisse gelesen werden sollen. Am Ausgang LOCAL werden 16 Byte Diagnosedaten der Kopplung übergeben, die Parameter PHYS und LOG werden bei PtP-Kopplungen nicht verwendet.

Hinweis

Da der Auftrag STATUS asynchron zu den übrigen auf einer Verbindung laufenden Aufträgen abläuft, kann den Fehlermeldungen kein SFB mit einer bestimmten R_ID zugeordnet werden. Das heißt, es kann zwar angezeigt werden, welche Fehler auf der Verbindung aufgetreten sind, nicht aber, von welchem SFB-Aufruf der Fehler ausgelöst wurde.

Aufbau des SYSTAT-Bereichs

Im SYSTAT-Bereich werden die ersten sechs Fehler/Ereignisse eingetragen, die der CP erkennt. Weitere Fehler/Ereignisse können erst wieder eingetragen werden, wenn der SYSTAT-Bereich gelöscht wurde.

Die Fehler/Ereignisse werden im Parameter LOCAL folgendermaßen eingetragen:

Byte 0	Betriebszustand des CP (02H für RUN, 05H für defekt)	
Byte 1	Reserviert	
Byte 2	Bit 0 - F	Fehler im SYSTAT eingetragen
	Bit 1 - U	Fehlerüberlauf
	Bit 2 - B	BREAK
Byte 3	Reserviert	
Byte 4/5	Ereignis 1	
Byte 6/7	Ereignis 2	
Byte 8/9	Ereignis 3	
Byte 10/11	Ereignis 4	
Byte 12/13	Ereignis 5	
Byte 14/15	Ereignis 6	

Löschen des SYSTAT-Bereichs

Nach dem Auslesen des SYSTAT-Bereichs mit dem SFB STATUS werden die SYSTAT-Meldungen automatisch gelöscht.

Nummernschema

Das Nummernschema für die Ereignisse im SYSTAT-Bereich ist wie folgt strukturiert:

Bit-Nr.	15		13	12				8	7						0
	Reserve			Ereignisklasse				Ereignisnummer/Fehlernummer							

Die genaue Aufschlüsselung der Ereignisklassen und Ereignisnummern finden Sie in den folgenden Kapiteln bzw. im Handbuch Punkt-zu-Punkt-Kopplungen CP 441-2.

Hinweis

Bei ladbaren Treibern sind die Ereignisklassen/Ereignisnummern für den SYSTAT-Bereich gegenüber den Standardtreibern teilweise modifiziert.

In den **folgenden Kapiteln** finden Sie alle **modifizierten Ereignisklassen/Ereignisnummern** in ihrer **treiberspezifischen Bedeutung**.

Wenn Sie ein Ereignis nicht in dieser Bedienungsanleitung finden, entspricht es dem der Standardkopplungen und ist im Handbuch CP 441-2 beschrieben.

8.3 Tabelle der Fehler/Ereignisse

Ereignisklassen

Die nachfolgend aufgeführten Ereignisklassen sind definiert:

Ereignis-klasse	Beschreibung	Beschrieben in
1	Hardwarefehler auf dem CP	CP-Handbuch
2	Fehler bei der Initialisierung	CP-Handbuch
3	Fehler bei der Parametrierung der PBK	CP-Handbuch
4	Vom CP erkannte Fehler im Datenverkehr CP - CPU	CP-Handbuch
5	Fehler bei der Bearbeitung eines CPU-Auftrags	CP-Handbuch, Treiber-Handbuch
6	Fehler bei der Bearbeitung eines Partnerauftrags	CP-Handbuch
7	Sendefehler	CP-Handbuch
8	Empfangsfehler	Treiber-Handbuch
9	Fehlertelegramm vom Koppelpartner empfangen	Nicht verwendet
10	Fehler im Reaktionstelegramm des Partners, die vom CP erkannt werden.	Nicht verwendet
14	Allgemeine Verarbeitungsfehler des ladbaren Treibers	Treiber-Handbuch

8.3.1 Fehlercode im SYSTAT bei "CPU-Auftrags-Fehler"

Ereignisklasse 5 (05H) "CPU-Auftragsfehler"			
Ereignis-klasse/ -nr. (Hex)	Ereignis-nummer (Dezimal)	Ereignistext	Abhilfe
05 18H	24	<ul style="list-style-type: none"> Übertragungslänge beim Senden ist zu groß (> 4 kByte) Die Übertragungslänge beim SEND ist zu klein. 	Parameter LEN am SEND überprüfen.

8.3.2 Fehlercode im SYSTAT bei "Empfangsfehlern"

Ereignisklasse 8 (08H) "Empfangsfehler"			
Ereignis- klasse/ -nr. (Hex)	Ereignis- nummer (Dezimal)	Ereignistext	Abhilfe
08 06H	6	Zeichenverzugszeit ZVZ überschritten	Fehler beim Partnergerät oder Störungen auf der Übertragungsleitung beseitigen.
08 0CH	12	Bei einem Zeichen wurde ein Übertragungsfehler erkannt (Paritätsfehler, Overrunfehler, Stoppbitfehler (Frame)).	Prüfen Sie, ob Störungen auf die Übertragungsleitung einwirken. Ändern Sie gegebenenfalls den Systemaufbau bzw. die Leitungsverlegung. Prüfen Sie, ob die Protokollparameter Baudrate, Datenbitanzahl, Parität, Stoppbitanzahl bei CP und Koppelpartner gleich eingestellt sind.
08 0DH	13	BREAK Die Empfangsleitung zum Partnergerät ist unterbrochen.	Stellen Sie die Verbindung zwischen den Geräten her oder schalten Sie das Partnergerät ein. Prüfen Sie bei TTY-Betrieb, ob im Ruhezustand Linienstrom fließt. Überprüfen und ändern Sie gegebenenfalls bei einer RS422/485 (X27)-Verbindung die Vorbelegung der 2-Draht Empfangsleitung R(A), R(B).
08 30H	48	Die Antwortüberwachungszeit nach Senden eines Anforderungstelegramms ist abgelaufen, ohne dass der Beginn des Antworttelegramms erkannt wurde.	Prüfen Sie, ob die Übertragungsleitung gestört ist (gegebenenfalls Schnittstellen-Analyzer verwenden). Prüfen Sie, ob die Protokollparameter Baudrate, Daten-bitanzahl, Parität, Stoppbitanzahl bei CP und Koppelpartner gleich eingestellt sind. Prüfen Sie, ob die mit PtP_PARAM parametrisierte Antwortüberwachungszeit groß genug eingestellt ist. Prüfen Sie, ob die angegebene Slave-Adresse vorhanden ist.
08 31H	49	Das erste Zeichen des Antworttelegramms vom Slave ist ungleich der im Anforderungstelegramm gesendeten Slave-Adresse (bei Betriebsart "Normal").	Ein falscher Slave hat geantwortet. Prüfen Sie, ob die Übertragungsleitung gestört ist (gegebenenfalls Schnittstellen-Analyzer verwenden).
08 32H	50	Überlauf des Empfangspuffers im CP bei Empfang des Antworttelegramms.	Überprüfen Sie die Protokolleinstellungen beim Slave.

8.3.3 Fehlercode im SYSTAT bei "allgemeinen Verarbeitungsfehlern"

Ereignisklasse 14 (0EH) "Ladbarer Treiber - Allgemeine Verarbeitungsfehler"			
Ereignis- klasse/ -nr. (Hex)	Ereignis- nummer (Dezimal)	Ereignistext	Abhilfe
0E 01H	1	Fehler bei der Initialisierung des treiberspezifischen SCC-Prozesses.	Treiber neu parametrieren und laden.
0E 02H	2	Fehler bei Anlauf des Treibers: Falscher SCC-Prozess aktiv (SCC-Treiber). Der Treiber ist mit diesem SCC-Treiber nicht funktionsfähig.	Treiber neu parametrieren und laden.
0E 03H	3	Fehler bei Anlauf des Treibers: Falscher Datenübergabe-Prozess aktiv (Schnittstelle zu den SFBs). Der Treiber ist mit diesem Datenübergabe-Prozess nicht funktionsfähig.	Treiber neu parametrieren und laden.
0E 04H	4	Fehler bei Anlauf des Treibers: Unzulässiges Schnittstellenmodul Der Treiber kann mit dem parametrierten Schnittstellenmodul nicht betrieben werden.	Parametrierung überprüfen und korrigieren.
0E 05H	5	Fehler bei Treiber-Dongle: Es ist kein Dongle gesteckt oder der gesteckte Dongle ist fehlerhaft. Der Treiber ist nicht betriebsbereit.	Überprüfen Sie, ob im CP ein Treiber-Dongle gesteckt ist. Wenn ein Dongle gesteckt ist, ersetzen Sie den fehlerhaften Dongle durch einen korrekten Dongle.
0E 06H	6	Fehler bei Treiber-Dongle: Der Dongle hat keinen gültigen Inhalt. Der Treiber ist nicht betriebsbereit.	Besorgen Sie sich bei Ihrer SIEMENS Niederlassung, von der Sie den Treiber erhalten haben, einen korrekten Dongle.
:	:		
0E 10H	16	Interner Fehler Prozedur: Default-Zweig im Prozedur-Automat.	Neuanlauf des CP (Netz_EIN)
0E 11H	17	Interner Fehler Prozedur: Default-Zweig bei Prozedurzustand Senden/Empfang.	Neuanlauf des CP (Netz_EIN)
0E 12H	18	Interner Fehler Aktiv-Automat: Default-Zweig	Neuanlauf des CP (Netz_EIN)
0E 13H	19	Interner Fehler Passiv-Automat: Default-Zweig	Neuanlauf des CP (Netz_EIN)

Ereignisklasse 14 (0EH) "Ladbarer Treiber - Allgemeine Verarbeitungsfehler <Parametrierung>"			
Ereignis- klasse/ -nr. (Hex)	Ereignis- nummer (Dezimal)	Ereignistext	Abhilfe
0E 20H	32	Die Anzahl der Datenbits bei dieser Kopplung muss 8 betragen. Der Treiber ist nicht betriebsbereit.	Korrigieren Sie die Parametrierung des Treibers.
0E 21H	33	Der parametrierte Multiplikationsfaktor für die Zeichenverzugszeit ist nicht im Bereich von 1 bis 10. Der Treiber arbeitet mit der Standardeinstellung von 1.	Korrigieren Sie die Parametrierung des Treibers.
0E 22H	34	Die parametrierte Betriebsart des Treibers ist unzulässig. Es ist "Normal"- oder "Störungsunterdrückungs"-Betrieb anzugeben. Der Treiber ist nicht betriebsbereit.	Korrigieren Sie die Parametrierung des Treibers.
0E 23H	35	Unzulässiger Wert für die Antwortüberwachungszeit parametriert: Zulässige Werte 5 ... 65500ms. Der Treiber ist nicht betriebsbereit.	Korrigieren Sie die Parametrierung des Treibers.
:	:		
0E 2EH	46	Beim Lesen der Schnittstellenparameter-Datei ist ein Fehler aufgetreten. Der Treiber ist nicht betriebsbereit.	Neuanlauf des CP (Netz_EIN).

Ereignisklasse 14 (0EH) "Ladbarer Treiber - Allgemeine Verarbeitungsfehler <CPU-CP>"			
Ereignis- klasse/ -nr. (Hex)	Ereignis- nummer (Dezimal)	Ereignistext	Abhilfe
0E 30H	48	Interner Fehler Übergabe an CPU: Unerwartete Quittung Passiv.	Kann bei sporadischem Auftreten ignoriert werden.
0E 31H	49	Time Out bei Übergabe von Daten an die CPU	Schnittstelle CP-CPU überprüfen.
0E 32H	50	Fehler bei Übergabe von Daten mit RCV an die CPU: Ursache analysieren - Detailfehler steht vor diesem Eintrag im SYSTAT.	Schnittstelle CP-CPU überprüfen.
0E 33H	51	Interner Fehler bei Datenübergabe an CPU: Unzulässiger Automatenzustand	Schnittstelle CP-CPU überprüfen.
:	:		
0E 3CH	60	Unzulässiger Auftrag bei diesem Treiber.	Nur SFB SEND, RCV, STATUS (nur bei CP 441-2) zulässig

Ereignisklasse 14 (0EH) "Ladbarer Treiber - Allgemeine Verarbeitungsfehler <Bearbeitung BSEND-Auftrag>"			
Ereignis- klasse/ -nr. (Hex)	Ereignis- nummer (Dezimal)	Ereignistext	Abhilfe
0E 40H	64	Beim Parameter LEN am SFB SEND wurde ein zu kleiner Wert angegeben.	Die Mindestlänge ist 2 Byte.
0E 41H	65	Beim Parameter LEN am SFB SEND wurde ein zu kleiner Wert angegeben. Bei dem übergebenen Funktionscode ist eine größere Länge nötig.	Die Mindestlänge ist bei diesem Funktionscode 6 Byte.
0E 42H	66	Der übergebene Funktionscode ist nicht zulässig.	Nur die im Kapitel " Funktionscodes (Seite 47) " aufgelisteten Funktionscodes sind zulässig.
0E 43H	67	Die Slave-Adresse 0 (= Broadcast) ist bei diesem Funktionscode nicht zulässig.	Verwenden Sie Slave-Adresse 0 nur bei den hierfür geeigneten Funktionscodes.
0E 44H	68	Der Wert des übergebenen Parameters "Bit-Anzahl" ist nicht im Bereich 1... 2040.	Der Parameter "Bit-Anzahl" muss im Bereich 1 ... 2040 liegen.
0E 45H	69	Der Wert des übergebenen Parameters "Register-Anzahl" ist nicht im Bereich 1 ... 127.	Der Parameter "Register-Anzahl" muss im Bereich 1 ... 127 liegen.
0E 46H	70	Funktionscode 15 oder 16: Der Wert des übergebenen Parameters "Bit-Anzahl" bzw. "Register-Anzahl" ist nicht im Bereich 1 ... 2040 bzw. 1 ... 127.	Der Parameter "Bit-Anzahl" bzw. "Register-Anzahl" muss im Bereich 1 ... 2040 bzw. 1 ... 127 liegen.
0E 47H	71	Funktionscode 15 oder 16: Der Parameter LEN am SFB SEND korrespondiert nicht mit dem übergebenen Parameter "Bit-Anzahl" bzw. "Register-Anzahl". Der Parameter LEN ist zu klein.	Vergrößern Sie den Parameter LEN am SEND, bis eine ausreichende Anzahl von Nutzdaten an den CP übergeben wird. Aufgrund von "Bit-Anzahl" bzw. "Register-Anzahl" muss eine größere Anzahl von Nutzdaten an den CP übergeben werden.
0E 48H	72	Funktionscode 05: Die im SEND-Quell-DB angegebene Codierung für "Bit-Setzen" (FF00H) oder "Bit-Löschen" (0000H) ist falsch.	Es sind nur die Codierungen FF00H und 0000H erlaubt.
0E 49H	73	Funktionscode 08: Der im SEND-Quell-DB angegebene "Diagnostic Code" ist falsch.	Es ist nur der "Diagnostic Code" 0000H erlaubt.
:	:		
0E 4FH	79	CP 441: Die am SFB SEND angegebene R_ID ist bei diesem Treiber unzulässig.	Nur R_IDs 0 ... 255 (00 00 00 00 ... 00 00 00 FFH) verwenden.
		CP 341: Die am SFB SEND RK angegebene R_TYP ist bei diesem Treiber unzulässig.	Als R_TYP muss "X" eingetragen werden.

Ereignisklasse 14 (0EH) "Ladbarer Treiber - Allgemeine Verarbeitungsfehler <Empfangsauswertung>"			
Ereignis- klasse/ -nr. (Hex)	Ereignis- nummer (Dezimal)	Ereignistext	Abhilfe
0E 50H	80	Slave-Adresse falsch: Die empfangene Slave-Adresse ist ungleich der gesendeten Slave-Adresse.	Ein falscher Slave hat geantwortet. Prüfen Sie, ob die Übertragungs- leitung gestört ist (gegebenenfalls Schnittstellen-Analyzer verwenden).
0E 51H	81	Funktionscode falsch: Der im Antworttelegramm empfangene Funktionscode ist ungleich dem gesendeten Funktionscode.	Überprüfen Sie das Slave-Gerät.
0E 52H	82	Byte-Underflow: Es wurden weniger Zeichen empfangen, als sich durch den Bytezähler des Antworttelegramms ergeben würden, bzw. als bei diesem Funktionscode erwartet werden.	Überprüfen Sie das Slave-Gerät.
0E 53H	83	Byte-Overflow: Es wurden mehr Zeichen empfangen, als sich durch den Bytezähler des Antworttelegramms ergeben würden, bzw. als bei diesem Funktionscode erwartet werden.	Überprüfen Sie das Slave-Gerät.
0E 54H	84	Bytezähler zu klein: Der im Antworttelegramm empfangene Bytezähler ist zu klein.	Überprüfen Sie das Slave-Gerät.
0E 55H	85	Bytezähler falsch: Der im Antworttelegramm empfangene Bytezähler ist falsch.	Überprüfen Sie das Slave-Gerät.
0E 56H	86	Echo falsch: Die vom Slave geechoteten Daten des Antworttelegramms (Bit-Anzahl, ...) sind ungleich den im Anforderungstelegramm gesendeten Daten.	Überprüfen Sie das Slave-Gerät.
0E 57H	87	CRC-Check fehlerhaft: Bei der Überprüfung der CRC-16-Checksumme des Antworttelegramms vom Slave wurde ein Fehler festgestellt.	Überprüfen Sie das Slave-Gerät.

Ereignisklasse 14 (0EH) "Ladbarer Treiber - Allgemeine Verarbeitungsfehler <Empfang Exceptioncode-Telegramm>"			
Ereignis- klasse/ -nr. (Hex)	Ereignis- nummer (Dezimal)	Ereignistext	Abhilfe
0E 61H	97	Antworttelegramm mit Exception Code 01: Illegal function	Siehe "Handbuch des Slave-Geräts"
0E 62H	98	Antworttelegramm mit Exception Code 02: Illegal data address	Siehe "Handbuch des Slave-Geräts"
0E 63H	99	Antworttelegramm mit Exception Code 03: Illegal data value	Siehe "Handbuch des Slave-Geräts"
0E 64H	100	Antworttelegramm mit Exception Code 04: Failure in associated device	Siehe "Handbuch des Slave-Geräts"
0E 65H	101	Antworttelegramm mit Exception Code 05: Acknowledge	Siehe "Handbuch des Slave-Geräts"
0E 66H	102	Antworttelegramm mit Exception Code 06: Busy, Rejected message	Siehe "Handbuch des Slave-Geräts"
0E 67H	103	Antworttelegramm mit Exception Code 07: Negative Acknowledgement	Siehe "Handbuch des Slave-Geräts"

Applikationsbeispiel

9.1 Applikationsbeispiel für CP 341

9.1.1 Applikationsbeispiel für CP 341

Allgemein

Im Folgenden ist die Verwendung der FBs P_SND_RK und P_RCV_RK in einem einfachen Programmbeispiel dargestellt.

Das Programmbeispiel wird bei der Installation des Modbus Masters im STEP 7-Verzeichnis EXAMPLES unter dem Namen *Modma* abgelegt.

Das S7 Programm soll als Informationsquelle dienen und kann nicht als verbindlicher Lösungsvorschlag kundenspezifischer Anlagenkonfigurationen betrachtet werden.

Das Programmbeispiel wurde bewusst einfach gestaltet. Auf symbolische Darstellung wurde verzichtet, um die prinzipielle Struktur deutlich zu machen.

9.1.2 Verwendete Bausteine

Verwendete Bausteine

Folgende Bausteine sind im Programmbeispiel verwendet.

Baustein	Symbol	Kommentar
OB 1	Cycle Execution	Zyklische Programmbearbeitung
OB 100	Complete Restart	Anlauf-OB für Neustart
FC 10	Initiation	FC für Anlauf-OB
FC 21	Execute Send Jobs	FC mit Aufruf des SFB P_SND_RK
FC 23	Execute Receive Jobs	FC mit Aufruf des SFB P_RCV_RK
DB 50	IDB_P_SND_RK	Instanz-DB für P_SND_RK
DB 70	IDB_P_RCV_RK	Instanz-DB für P_RCV_RK
DB 40	Work DB Send	Arbeits-DB für FC21 und P_SND_RK
DB 41	Work DB Receive	Arbeits-DB für FC23 und P_RCV_RK
DB 42	SOURCE_DB	P_SND_RK-Quell-DB mit Sendedaten
DB 43	DESTINATION_DB	P_RCV_RK-Ziel-DB für Empfangsdaten

Verwendete Daten

Folgende Operanden (Merker, Datenbits oder Datenworte) sind im Programmbeispiel verwendet.

Operand	Kommentar
M120.7	Anstoßbit für die Ausführung eines P_SND_RK-Auftrags
DB40.DBX 0.0	Steuerparameter REQuest: Zur Aktivierung eines P_SND_RK
DB40.DBX 0.1	Steuerparameter Reset: Zum Abbrechen des aktuellen P_SND_RK
DB40.DBX 0.4	Zustandsparameter DONE: Anzeige, dass der aktuelle P_SND_RK ohne Fehler abgeschlossen wurde.
DB40.DBX 0.5	Zustandsparameter ERROR: Anzeige, dass der aktuelle P_SND_RK mit Fehler abgeschlossen wurde.
DB40.DBW 6	Erfolgszähler bei P_SND_RK
DB40.DBW 8	Fehlerzähler bei P_SND_RK
DB40.DBW 10	Länge LEN des an den CP zu übergebenden P_SND_RK-Quell-Datenbereichs in Byte
DB40.DBW 12	STATUS-Anzeige am P_SND_RK
DB40.DBW 14	Gespeicherte P_SND_RK-STATUS-Anzeige
DB41.DBX 0.0	Steuerparameter EN_R: Zur Empfangsfreigabe für den P_RCV_RK
DB41.DBX 0.4	Zustandsparameter NDR: Anzeige, dass der aktuelle P_RCV_RK neue Daten vom CP empfangen hat.
DB41.DBX 0.5	Zustandsparameter ERROR: Anzeige, dass der aktuelle P_RCV_RK mit Fehler beendet wurde.
DB41.DBW 4	Gespeicherte Länge LEN des P_RCV_RK
DB41.DBW 6	Erfolgszähler bei P_RCV_RK
DB41.DBW 8	Fehlerzähler bei P_RCV_RK
DB41.DBW 10	Länge LEN des vom CP empfangenen P_RCV_RK-Ziel-Datenbereichs in Byte
DB41.DBW 12	STATUS-Anzeige am P_RCV_RK
DB41.DBW 14	Gespeicherte P_RCV_RK-STATUS-Anzeige

9.1.3 Programmbeschreibung

Allgemein

Das Programmbeispiel besteht aus:

- Anlauf-Baustein OB 100, FC 10
- Zyklischer Teil OB 1 mit Aufruf von
- Funktionsbaustein FC 21 zur Datenübergabe CPU nach CP (**Senden**)
- FC 23 zum Daten-**Empfang** CP nach CPU

Die Parameter für die programmierten Systemfunktionsbausteine P_SND_RK und P_RCV_RK sind in den Arbeits-DBs **DB 40** und **DB 41** abgelegt.

Die Sendedaten, der SEND-Quellbereich, sind im **DB 42** enthalten.

Vom Koppelpartner empfangene Daten werden im **DB 43**, dem RCV-Zielbereich, eingetragen.

P_SND_RK-Auftrag

Im zyklischen Teil des Programms kann durch Setzen des Merkers **M 120.7** (z. B. durch VARIABLE STEUERN) ein **P_SND_RK-Auftrag** aktiviert werden. Es werden die im P_SND_RK-Quellbereich DB 42 enthaltenen Daten mit Länge LEN an den CP übergeben. Der Anstoßmerker M 120.7 wird sofort wieder zurückgesetzt.

Nach Ausführung des P_SND_RK-Auftrags wird bei Abschluss ohne Fehler ein Erfolgswähler, bei Abschluss mit Fehler ein Fehlerwähler inkrementiert.

P_RCV_RK-Auftrag

Im FC 23 ist ein SFB **P_RCV_RK** programmiert, bei dem die **Empfangsfreigabe** immer **"1"** ist, um Empfangsdaten vom Koppelpartner empfangen zu können. Die Empfangsdaten werden in den **P_RCV_RK-Zielbereich** eingetragen, die Anzahl der eingetragenen Daten wird am Parameter **LEN** angezeigt.

Bei Übernahme der Daten ohne Fehler wird ein Erfolgswähler, bei Abschluss mit Fehler ein Fehlerwähler inkrementiert.

Bei den genannten Aufträgen P_SND_RK und P_RCV_RK wird der Ausgangsparameter **STATUS** gespeichert, wenn ein Wert ungleich 0 gemeldet wird.

9.1.4 Programmbeispiel

Programmbeispiel

Folgende Bausteine sind nachfolgend aufgelistet:

Baustein	Kommentar
OB 100	Anlauf-OB für Neustart
FC 10	Anlauf-FC für OB 100
OB 1	Zyklische Programmbearbeitung
FC 21	FC mit Aufruf des FB P_SND_RK
FC 23	FC mit Aufruf des FB P_RCV_RK

Programm-Anlauf

OB 100 Complete Restart

```

L 272 //logical address
T DB40.DBW 16 //for SEND
T DB41.DBW 16 //and RCV
UC FC 10 //Call of FC for Initiation
    
```

FC 10 Initiation

```

// -----
// Reset Control Bits
// -----
L B#16#0
T DB40.DBB 0 //SEND-Work-DB
T DB41.DBB 0 //RCV-Work-DB
// -----
// Reset counters/STATUS
// -----
L W#16#0
T DB40.DBW 6 //SEND- Work -DB
T DB40.DBW 8
T DB40.DBW 12
T DB40.DBW 14
T DB41.DBW 6 //RCV- Work -DB
T DB41.DBW 8
T DB41.DBW 12
T DB41.DBW 14
    
```

Zyklischer-Programmablauf

OB 1 Cyclic-OB

```
UC FC 21 //Call of SEND
UC FC 23 //Call of RCV
```

FC 21 Execute SEND-Jobs

```
// -----
// Interlockings for SEND
// -----
U M 120.7 //Trigger SEND
UN DB40.DBX 0.0 //SEND_REQ
UN DB40.DBX 0.4 //SEND_DONE
UN DB40.DBX 0.5 //SEND_ERROR
R M 120.7 //Reset Trigger SEND
S DB40.DBX 0.0 //Set SEND_REQ
// -----
// Generate edge SEND_REQ
// -----
U(
O DB40.DBX 0.4 //SEND_DONE
O DB40.DBX 0.5 //SEND_ERROR
)
U DB40.DBX 0.0 //SEND_REQ
R DB40.DBX 0.0 //SEND with REQ=0
// -----
// Supply LEN
// -----
L W#16#20 //Length SEND-Data
T DB40.DBW 10 //SEND-LEN
// -----
// SEND with Instance-DB
// -----
CALL FB 8 , DB50
SF :=
REQ :=DB40.DBX0.0
R :=DB40.DBX0.1
LADDR :=DB40.DBW16
DB_NO :=42
DBB_NO :=10
LEN :=DB40.DBW10
R_CPU_NO :=
R_TYP :='X'
```

FC 21 Execute SEND-Jobs

```

R_NO :=
R_OFFSET :=
R_CF_BYT :=
R_CF_BIT :=
DONE :=DB40.DBX0.4
ERROR :=DB40.DBX0.5
STATUS :=DB40.DBW12
// -----
// Check "Complete without error"
// -----
U DB40.DBX 0.4 //DONE ?
SPBN CON1 //if NO
L DB40.DBW 6 // "Complete without error"
+1 //increment counter
T DB40.DBW 6
: // :
: //further user
: // functions
: // :
SPA LEAV
// -----
// Check "Complete with error"
// -----
CON1: U DB40.DBX 0.5 //ERROR ?
SPBN CON2 //if NO
L DB40.DBW 8 // "Complete with error"
+1 // increment counter
T DB40.DBW 8
: // :
: //Error-Handling
: // :
L 0
L DB40.DBW 12 //if STATUS <>0
==I
SPB LEAV
T DB40.DBW 14 //save STATUS
SPA LEAV
// -----
// Check "Error in STATUS"
// -----
CON2: L 0
L DB40.DBW 12 //if STATUS <>0
==I
SPB LEAV

```

FC 21 Execute SEND-Jobs

```
T DB40.DBW 14 //save STATUS
: // :
: //Error-Handling
: // :
LEAV: CLR
```

FC 23 Carry out RCV-Receive

```
// -----
// Enable Receive Data
// -----
SET
= DB41.DBX 0.0 //RCV with EN_R=1
// -----
// RCV with Instance-DB
// -----
CALL FB 7 , DB70
EN_R :=DB41.DBX0.0
R:=
LADDR :=DB41.DBW16
DB_NO :=43
DBB_NO :=0
L_TYP :=
L_NO :=
L_OFFSET :=
L_CF_BYT :=
L_CF_BIT :=
NDR :=DB41.DBX0.4
ERROR :=DB41.DBX0.5
LEN :=DB41.DBW10
STATUS :=DB41.DBW12
// -----
// Check "Receive without error"
// -----
U DB41.DBX 0.4 //NDR ?
SPBN CON1 //if NO
L DB41.DBW 6 //"Receive without error"
+1 //increment counter
T DB41.DBW 6
L DB41.DBW 10 //save
T DB41.DBW 4 //Receive-Length LEN
SPA LEAV
// -----
```

FC 23 Carry out RCV-Receive

```
// Check "Receive with error"
// -----
CON1: U DB41.DBX 0.5           //ERROR ?
SPBN CON2                     //if NO
L DB41.DBW 8                  //"Receive with error"
+1                             //increment counter
T DB41.DBW 8
L 0
L DB41.DBW 12                 //if STATUS <>0
==I
SPB LEAV
T DB41.DBW 14                 //save STATUS
SPA LEAV
// -----
// Check "Error in STATUS"
// -----
CON2: L 0
L DB41.DBW 12                 //if STATUS <>0
==I
SPB LEAV
T DB41.DBW 14                 //save STATUS
LEAV: CLR
```

9.2 Applikationsbeispiel für CP 441-2

Allgemein

Im Folgenden ist die Verwendung der SFBs BSEND, BRCV und STATUS in einem einfachen Programmbeispiel dargestellt.

Das Programmbeispiel wird bei der Installation des Modbus Masters im STEP 7-Verzeichnis EXAMPLES unter dem Namen *Modma* abgelegt.

Das S7-400 Programm soll als Informationsquelle dienen und kann nicht als verbindlicher Lösungsvorschlag kundenspezifischer Anlagenkonfigurationen betrachtet werden.

Das Programmbeispiel wurde bewusst einfach gestaltet. Auf symbolische Darstellung wurde verzichtet, um die prinzipielle Struktur deutlich zu machen.

9.2.1 Verwendete Bausteine

Verwendete Bausteine

Folgende Bausteine sind im Programmbeispiel verwendet.

Baustein	Symbol	Kommentar
OB 1	Cycle Execution	Zyklische Programmbearbeitung
OB 100	Complete Restart	Anlauf-OB für Neustart
FC 100	Initiation	FC für Anlauf-OB
FC 210	Execute BSEND Jobs	FC mit Aufruf des SFB BSEND
FC 230	Execute BRCV Jobs	FC mit Aufruf des SFB BRCV
FC 250	Execute STATUS Jobs	FC mit Aufruf des SFB STATUS
DB 22	IDB_STATUS	Instanz-DB für STATUS
DB 50	IDB_BSEND	Instanz-DB für BSEND
DB 70	IDB_BRCV	Instanz-DB für BRCV
DB 400	Work DB BSEND	Arbeits-DB für FC 210 und BSEND
DB 401	Work DB BRCV	Arbeits-DB für FC 230 und BRCV
DB 410	SOURCE_DB	BSEND-Quell-DB mit Sendedaten
DB 430	DESTINATION_DB	BRCV-Ziel-DB für Empfangsdaten
DB 450	Work DB STATUS	Arbeits-DB und SYSTAT-Empfangs-DB für FC 250 und STATUS

Verwendete Daten

Folgende Operanden (Merker, Datenbits oder Datenworte) sind im Programmbeispiel verwendet.

Operand	Kommentar
M 119.7	Anstoßbit für die Ausführung eines STATUS-Auftrags
M 120.7	Anstoßbit für die Ausführung eines BSEND-Auftrags
DB400.DBX 0.0	Steuerparameter REQuest: <ul style="list-style-type: none"> Zur Aktivierung eines BSEND
DB400.DBX 0.1	Steuerparameter Reset: <ul style="list-style-type: none"> Zum Abbrechen des aktuellen BSEND
DB400.DBX 0.4	Zustandsparameter DONE: <ul style="list-style-type: none"> Anzeige, dass der aktuelle BSEND ohne Fehler abgeschlossen wurde.
DB400.DBX 0.5	Zustandsparameter ERROR: <ul style="list-style-type: none"> Anzeige, dass der aktuelle BSEND mit Fehler abgeschlossen wurde.
DB400.DBW 6	Erfolgszähler bei BSEND
DB400.DBW 8	Fehlerzähler bei BSEND
DB400.DBW 10	Länge LEN des an den CP zu übergebenden BSEND-Quell-Datenbereichs in Byte
DB400.DBW 12	STATUS-Anzeige am BSEND
DB400.DBW 14	Gespeicherte BSEND-STATUS-Anzeige
DB400.DBD 16	Parameter R_ID beim BSEND
DB401.DBX 0.0	Steuerparameter EN_R: <ul style="list-style-type: none"> Zur Empfangsfreigabe für den BRCV
DB401.DBX 0.4	Zustandsparameter NDR: <ul style="list-style-type: none"> Anzeige, dass der aktuelle BRCV neue Daten vom CP empfangen hat.
DB401.DBX 0.5	Zustandsparameter ERROR: <ul style="list-style-type: none"> Anzeige, dass der aktuelle BRCV mit Fehler beendet wurde.
DB401.DBW 4	Gespeicherte Länge LEN des BRCV
DB401.DBW 6	Erfolgszähler bei BRCV
DB401.DBW 8	Fehlerzähler bei BRCV
DB401.DBW 10	Länge LEN des vom CP empfangenen BRCV-Ziel-Datenbereichs in Byte
DB401.DBW 12	STATUS-Anzeige am BRCV
DB401.DBW 14	Gespeicherte BRCV-STATUS-Anzeige
DB401.DBD 16	Parameter R_ID beim BRCV
DB450.DBX 0.0	Steuerparameter REQuest: <ul style="list-style-type: none"> Zur Aktivierung des STATUS
DB450.DBX 0.4	Zustandsparameter NDR: <ul style="list-style-type: none"> Anzeige, dass der STATUS neue SYSTAT-Daten vom CP übernommen hat.

Operand	Kommentar
DB450.DBX 0.5	Zustandsparameter ERROR: <ul style="list-style-type: none"> Anzeige, dass der aktuelle STATUS mit Fehler beendet wurde.
DB450.DBW 6	Erfolgszähler bei STATUS
DB450.DBW 8	Fehlerzähler bei STATUS
DB450.DBW 12	STATUS-Anzeige am STATUS
DB450.DBW 14	Gespeicherte STATUS-Anzeige
DB450.DBW 16	Parameter PHYS: <ul style="list-style-type: none"> Physikalischer Zustand der Schnittstelle (bei PtP-Kopplungen nicht verwendet)
DB450.DBW 18	Parameter LOG: Logischer Zustand der Schnittstelle (bei PtP-Kopplungen nicht verwendet)
DB450.DBW 20 : DB450.DBW 34	Parameter LOCAL: <ul style="list-style-type: none"> Zielbereich für den empfangenen SYSTAT-Bereich der Schnittstelle
DB450.DBW 40 : DB450.DBW 54	Gespeicherter SYSTAT-Bereich

9.2.2 Programmbeschreibung

Allgemein

Das Programmbeispiel besteht aus:

- Anlauf-Baustein OB 100, FC 100
- Zyklischer Teil OB 1 mit Aufruf von
- Funktionsbaustein FC 210 zur Datenübergabe CPU nach CP (**Senden**)
- FC 230 zum Daten-**Empfang** CP nach CPU
- FC 250 zum Lesen des **SYSTAT**-Bereichs.

Die Parameter für die programmierten Systemfunktionsbausteine BSEND, BRCV und STATUS sind in den Arbeits-DBs **DB 400** (BSEND), **DB401** (BRCV), **DB450** (STATUS) abgelegt.

Die Sendedaten (BSEND-Quellbereich) sind im **DB 410** enthalten. Vom Koppelpartner empfangene Daten werden im **DB 430** (BRCV-Zielbereich) eingetragen.

An den SFBs BSEND, BRCV, STATUS ist als **ID** der Wert **1000** (Hex) angegeben. Die IDs werden in der **Verbindungsprojektierung** beginnend mit 1000 (Hex) durchnummeriert. Wenn sich bei Ihrer Verbindungsprojektierung eine andere ID ergeben hat, müssen Sie diese ID an den jeweiligen SFBs angeben. (siehe auch Kapitel "Verbindungsprojektierung (Seite 27)").

BSEND-Auftrag

Im zyklischen Teil des Programms kann durch Setzen des Merkers **M 120.7** (z. B. durch VARIABLE STEUERN) ein **BSEND-Auftrag** aktiviert werden. Es werden die im BSEND-Quellbereich DB 410 enthaltenen Daten mit Länge LEN an den CP übergeben. Der Anstoßmerker M 120.7 wird sofort wieder zurückgesetzt.

Nach Ausführung des BSEND-Auftrags wird bei Abschluss ohne Fehler ein Erfolgszähler, bei Abschluss mit Fehler ein Fehlerzähler inkrementiert.

BRCV-Auftrag

Im FC 230 ist ein SFB **BRCV** programmiert, bei dem die **Empfangsfreigabe** immer "1" ist, um Empfangsdaten vom Koppelpartner empfangen zu können. Die Empfangsdaten werden in den **BRCV-Zielbereich** eingetragen, die Anzahl der eingetragenen Daten wird am Parameter **LEN** angezeigt.

Bei Übernahme der Daten ohne Fehler wird ein Erfolgszähler, bei Abschluss mit Fehler ein Fehlerzähler inkrementiert.

SYSTAT lesen

Durch Setzen des Merkers **M 119.7** (z. B. durch VARIABLE STEUERN) kann ein **SYSTAT-Lese-Auftrag** aktiviert werden. Dieser Anstoßmerker wird sofort wieder zurückgesetzt.

Im FC 250 ist ein SFB **STATUS** programmiert, der die Daten des SYSTAT-Bereichs des CP in das am STATUS-Baustein angegebene Ziel einträgt.

Bei Ausführung des STATUS-Auftrags ohne Fehler wird ein Erfolgszähler, bei Abschluss mit Fehler ein Fehlerzähler inkrementiert.

Bei den genannten Aufträgen BSEND, BRCV und SYSTAT-Lesen wird der Ausgangsparameter **STATUS** gespeichert, wenn ein Wert ungleich 0 gemeldet wird.

9.2.3 Programmbeispiel

Programmbeispiel

Folgende Bausteine sind nachfolgend aufgelistet:

Baustein	Kommentar
OB 100	Anlauf-OB für Neustart
FC 100	Anlauf-FC für OB100
OB 1	Zyklische Programmbearbeitung
FC 210	FC mit Aufruf des SFB BSEND
FC 230	FC mit Aufruf des SFB BRCV
FC 250	FC mit Aufruf des SFB STATUS

Programm-Anlauf

OB 100 Complete Restart

```
CALL FC 100 //Initialisation
```

FC 100 Initiation

```
// -----
// Reset Control Bits
// -----
L B#16#0
T DB400.DBB 0 //BSEND-Work-DB
T DB401.DBB 0 //BRCV-Work-DB
T DB450.DBB 0 //STATUS-Work-DB
// -----
// Reset counters/STATUS
// -----
L W#16#0
T DB400.DBW 6 //BSEND-Work-DB
T DB400.DBW 8
T DB400.DBW 12
T DB400.DBW 14
T DB401.DBW 6 //BRCV-Work-DB
T DB401.DBW 8
T DB401.DBW 12
T DB401.DBW 14
T DB450.DBW 6 //STATUS-Work-DB
T DB450.DBW 8
T DB450.DBW 12
T DB450.DBW 14
```

Zyklischer-Programmablauf

OB 1 Cyclic-OB

```

UC FC 210 //Call of BSEND
UC FC 230 //Call of BRCV
UC FC 250 //Call of STATUS
    
```

FC 210 Execute BSEND-Jobs

```

// -----
// Interlockings for BSEND
// -----
U M 120.7 //Trigger BSEND
UN DB450.DBX 0.0 //REQuest STATUS
UN DB400.DBX 0.0 //BSEND_REQ
UN DB400.DBX 0.4 //BSEND_DONE
UN DB400.DBX 0.5 //BSEND_ERROR
R M 120.7 //Reset Trigger BSEND
S DB400.DBX 0.0 //Set BSEND_REQ
// -----
// Generate edge BSEND:EQ
// -----
U(
O DB400.DBX 0.4 //BSEND_DONE
O DB400.DBX 0.5 //BSEND_ERROR
)
U DB400.DBX 0.0 //BSEND_REQ
R DB400.DBX 0.0 //BSEND with REQ=0
// -----
// Supply R_ID, LEN
// -----
L DW#16#1 //Use R_ID = 1
T DB400.DBD 16 //as BSEND-R_ID
L W#16#6 //Length BSEND-Data
T DB400.DBW 10 //BSEND-LEN
// -----
// BSEND with Instance-DB
// -----
CALL SFB 12 , DB50
REQ :=DB400.DBX0.0
R :=DB400.DBX0.1
ID :=W#16#1000
R_ID :=DB400.DBD16
DONE :=DB400.DBX0.4
ERROR :=DB400.DBX0.5
    
```

FC 210 Execute BSEND-Jobs

```

STATUS :=DB400.DBW12
SD_1 :=P#DB410.DBX 10.0 WORD 1
LEN :=DB400.DBW10
// -----
// Check "Complete without error"
// -----
U DB400.DBX 0.4 //DONE ?
SPBN CON1 //if NO
L DB400.DBW 6 //"Complete without error"
+1 //increment counter
T DB400.DBW 6
: // :
: //further User functions
: // :
SPA LEAV
// -----
Check "Complete with error"
// -----
CON1 U DB400.DBX 0.5 //ERROR ?
SPBN CON2 //if NO
L DB400.DBW 8 //"Complete with error"
+1 //increment counter
T DB400.DBW 8
: // :
: //Error-Handling
: // :
L 0
L DB400.DBW 12 //if STATUS <>0
==I
SPB LEAV
T DB400.DBW 14 //save STATUS
SPA LEAV
// -----
Check "Error in STATUS"
// -----
CON2: L 0
L DB400.DBW 12 //if STATUS <>0
==I
SPB LEAV
T DB400.DBW 14 //save STATUS
: // :
: //Error-Handling
: // :
LEAV: CLR

```

FC 230 Carry out BRCV-Receive

```
// -----
// Supply R_ID
// -----
L DW#16#1 //use BRCV-R_ID = 1
T DB401.DBD 16 //(same as BSEND-R_ID)
// -----
// Enable Receive Data
// -----
SET
= DB401.DBX 0.0 //BRCV with EN_R=1
// -----
// BRCV with Instance-DB
// -----
CALL SFB 13 , DB70
EN_R :=DB401.DBX0.0
ID :=W#16#1000
R_ID :=DB401.DBD16
NDR :=DB401.DBX0.4
ERROR :=DB401.DBX0.5
STATUS :=DB401.DBW12
RD_1 :=P#DB430.DBX 0.0 WORD 128
LEN :=DB401.DBW10
// -----
// Check "Receive without error"
// -----
U DB401.DBX 0. 4 //NDR ?
SPBN CON1 //if NO
L DB401.DBW 6 //"Receive without error"
+1 //increment counter
T DB401.DBW 6
L DB401.DBW 10 //save
T DB401.DBW 4 //Receive-Length LEN
SPA LEAV
// -----
// Check "Receive with error"
// -----
CON1: U DB401.DBX 0.5 //ERROR ?
SPBN CON2 //if NO
L DB401.DBW 8 //"Receive with error"
+1 //increment counter
T DB401.DBW 8
L 0
```

FC 230 Carry out BRCV-Receive

```
L DB401.DBW 12          //if STATUS <>0
==I
SPB LEAV
T DB401.DBW 14          //save STATUS
SPA LEAV
// -----
// Check "Error in STATUS"
// -----
CON2: L 0
L DB401.DBW 12          //if STATUS <>0
==I
SPB LEAV
T DB401.DBW 14          //save STATUS
LEAV: CLR
```

FC 250 Carry out STATUS-Job

```
// -----
// Interlockings for STATUS
// -----
U M 119.7                //Trigger STATUS
UN DB400.DBX 0.0         //BSEND-REQ active?
UN DB450.DBX 0.0         //STATUS_REQ active?
R M 119.7                //reset Trigger STATUS
S DB450.DBX 0.0         //activate STATUS_REQ
// -----
// Generate edge STATUS_REQ
// -----
U(
O DB450.DBX 0.4          //STATUS_NDR
O DB450.DBX 0.5          //STATUS_ERROR
)
U DB450.DBX 0.0          //STATUS_REQ
R DB450.DBX 0.0          //STATUS with REQ=0
// -----
// STATUS with Instanz-DB (= Read
SYSTAT)
// -----
CALL SFB 22 , DB22
REQ :=DB450.DBX0.0
ID :=W#16#1000
NDR :=DB450.DBX0.4
ERROR :=DB450.DBX0.5
```

FC 250 Carry out STATUS-Job

```

STATUS :=DB450.DBW12
PHYS :=P#DB450.DBX 16.0 BYTE 2
LOG :=P#DB450.DBX 18.0 BYTE 2
LOCAL :=P#DB450.DBX 20.0 BYTE 16
// -----
// Check "New Data received"
// -----
U DB450.DBX 0.4 //NDR ?
SPBN CON1 //if NO
L DB450.DBW 6 // "Complete without error"
+1 //increment counter
T DB450.DBW 6
U DB450.DBX 22.0 //Bit0: Error exists?
SPBN LEAV //if NO
// -----
// Save SYSTAT
// -----
L DB450.DBW 22
T DBW 42
L DBD 24
T DBD 44
L DBD 28
T DBD 48
L DBD 32
T DBD 52
SPA LEAV
// -----
// Check "Complete with error"
// -----
CON1: U DB450.DBX 0.5 //ERROR ?
SPBN CON2 //if NO
L DB450.DBW 8 // "Complete with error"
+1 //increment counter
T DB450.DBW 8
L 0
L DB450.DBW 12 //if STATUS <>0:
==I
SPB LEAV
T DB450.DBW 14 //save STATUS
SPA LEAV
// -----
// Check "Error in STATUS"
// -----
CON2: L 0

```

FC 250 Carry out STATUS-Job

```
L DB450.DBW 12 //if STATUS <>0
==I
SPB LEAV
T DB450.DBW 14 //save STATUS
LEAV: CLR
```


Technische Daten

A.1 Technische Daten

Übertragungszeiten

Die folgenden Tabellen enthalten gemessene Übertragungszeiten für die unterschiedlichen Funktionscodes.

Für die Zeitmessung wurde ein Automatisierungssystem S7-300 mit einer CPU 315-2 DP (6ES7315-2AF01-0AB0) und einem CP 341 verwendet und als Partner ein Automatisierungssystem S7-400 mit einer CPU 414 (6ES7414-1XG01-0AB0) und einem CP 441-2. Gemessen wurden folgende Zeiten:

- Die Verarbeitungszeit vom Anstoß des Auftrags im Anwenderprogramm einschließlich der Verarbeitungszeit auf dem Master.
- Die Übertragungszeit des Auftrags über die serielle Schnittstelle an den Partner.
- Die Zeit zur Bearbeitung auf dem Slave.
- Die Zeit, die für die Übertragung der Quittung auf der seriellen Schnittstelle benötigt wird.

Zur Berechnung der Zeit für eine vollständige Übertragung müssen die vier Zeiten addiert werden.

Wenn Sie als Partner einen anderen Master oder Slave einsetzen, müssen Sie statt der Zeiten in der Tabelle die entsprechenden Zeiten des verwendeten Masters oder Slaves verwenden. Die Zeiten für Auftrag und für Quittung bleiben gleich, sie sind nur von der verwendeten Baudrate abhängig.

Master ist CP 341

Funktionscode 1 (Lesen) – Read Coil (Output) Status (Zeiten in msec)

Baudrate (Baud)	300			
Nutzdaten	Master CP 341	Auftrag	Slave CP 441-2	Quittung
1 Byte	236	257	188	184
10 Byte	236	257	190	515
20 Byte	238	257	190	882
50 Byte	244	257	193	1986
100 Byte	280	257	199	3824
200 Byte	286	257	207	7502
255 Byte	288	257	216	9487

Baudrate (Baud)	9600			
Nutzdaten	Master CP 341	Auftrag	Slave CP 441-2	Quittung
1 Byte	33	8	40	6
10 Byte	33	8	43	16
20 Byte	35	8	44	28
50 Byte	42	8	45	62
100 Byte	56	8	56	120
200 Byte	75	8	64	235
255 Byte	82	8	77	296

Baudrate (Baud)	76800			
Nutzdaten	Master CP 341	Auftrag	Slave CP 441-2	Quittung
1 Byte	35	1	23	1
10 Byte	36	1	25	2
20 Byte	37	1	26	3
50 Byte	46	1	27	8
100 Byte	61	1	30	15
200 Byte	82	1	39	29
255 Byte	92	1	48	37

Master ist CP 341

Funktionscode 15 (Schreiben) – Force Multiple Coils (Zeiten in msec)

Baudrate (Baud)	300			
Nutzdaten	Master CP 341	Auftrag	Slave CP 441-2	Quittung
1 Byte	225	331	223	257
10 Byte	227	662	224	257
20 Byte	227	1030	228	257
50 Byte	227	2132	232	257
100 Byte	229	3971	236	257
200 Byte	230	7648	243	257
255 Byte	237	9634	255	257

Baudrate (Baud)	9600			
Nutzdaten	Master CP 341	Auftrag	Slave CP 441-2	Quittung
1 Byte	64	11	62	8
10 Byte	64	21	63	8
20 Byte	69	32	64	8
50 Byte	69	67	68	8
100 Byte	72	124	70	8
200 Byte	75	239	76	8
255 Byte	75	301	86	8

Baudrate (Baud)	76800			
Nutzdaten	Master CP 341	Auftrag	Slave CP 441-2	Quittung
1 Byte	60	1	56	1
10 Byte	60	3	58	1
20 Byte	62	4	58	1
50 Byte	64	9	60	1
100 Byte	67	16	67	1
200 Byte	72	30	77	1
255 Byte	77	38	84	1

Master ist CP 441

Funktionscode 1 (Lesen) – Read Coil (Output) Status (Zeiten in msec)

Baudrate (Baud)	300			
Nutzdaten	Master CP 441-2	Auftrag	Slave CP 341	Quittung
1 Byte	229	257	179	184
10 Byte	229	257	179	512
20 Byte	229	257	180	882
50 Byte	232	257	182	1986
100 Byte	236	257	192	3842
200 Byte	243	257	208	7501
255 Byte	251	257	214	9487

Baudrate (Baud)	9600			
Nutzdaten	Master CP 441-2	Auftrag	Slave CP 341	Quittung
1 Byte	74	8	18	6
10 Byte	75	8	19	16
20 Byte	77	8	19	27
50 Byte	83	8	24	62
100 Byte	90	8	34	119
200 Byte	92	8	48	235
255 Byte	95	8	56	296

Baudrate (Baud)	76800			
Nutzdaten	Master CP 441-2	Auftrag	Slave CP 341	Quittung
1 Byte	73	1	13	1
10 Byte	74	1	13	2
20 Byte	76	1	13	4
50 Byte	86	1	20	8
100 Byte	93	1	29	15
200 Byte	95	1	45	29
255 Byte	97	1	50	37

Master ist CP 441

Funktionscode 15 (Schreiben) – Force Multiple Coils (Zeiten in msec)

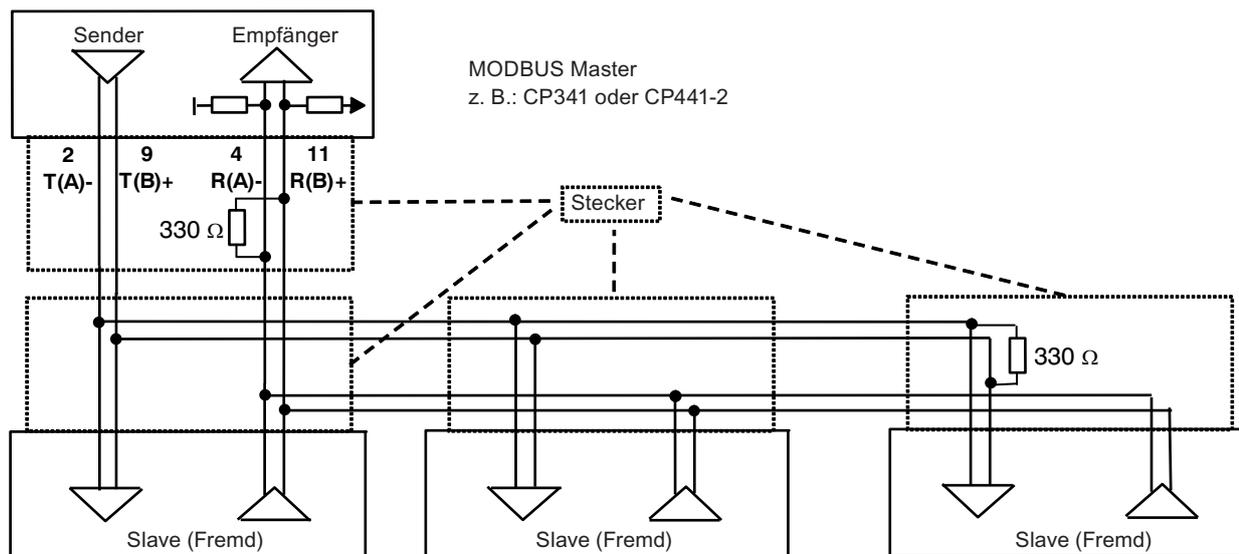
Baudrate (Baud)	300			
Nutzdaten	Master CP 441-2	Auftrag	Slave CP 341	Quittung
1 Byte	205	331	199	257
10 Byte	206	662	200	257
20 Byte	206	1028	201	257
50 Byte	208	2132	212	257
100 Byte	211	3971	223	257
200 Byte	217	7648	238	257
255 Byte	221	9634	243	257

Baudrate (Baud)	9600			
Nutzdaten	Master CP 441-2	Auftrag	Slave CP 341	Quittung
1 Byte	48	10	41	8
10 Byte	48	20	41	8
20 Byte	50	32	43	8
50 Byte	52	67	48	8
100 Byte	55	124	56	8
200 Byte	63	239	74	8
255 Byte	67	301	88	8

Baudrate (Baud)	76800			
Nutzdaten	Master CP 441-2	Auftrag	Slave CP 341	Quittung
1 Byte	58	1	40	1
10 Byte	61	3	43	1
20 Byte	62	4	43	1
50 Byte	63	8	44	1
100 Byte	64	15	50	1
200 Byte	66	30	69	1
255 Byte	68	38	85	1

Verdrahtungspläne Mehrpunkt

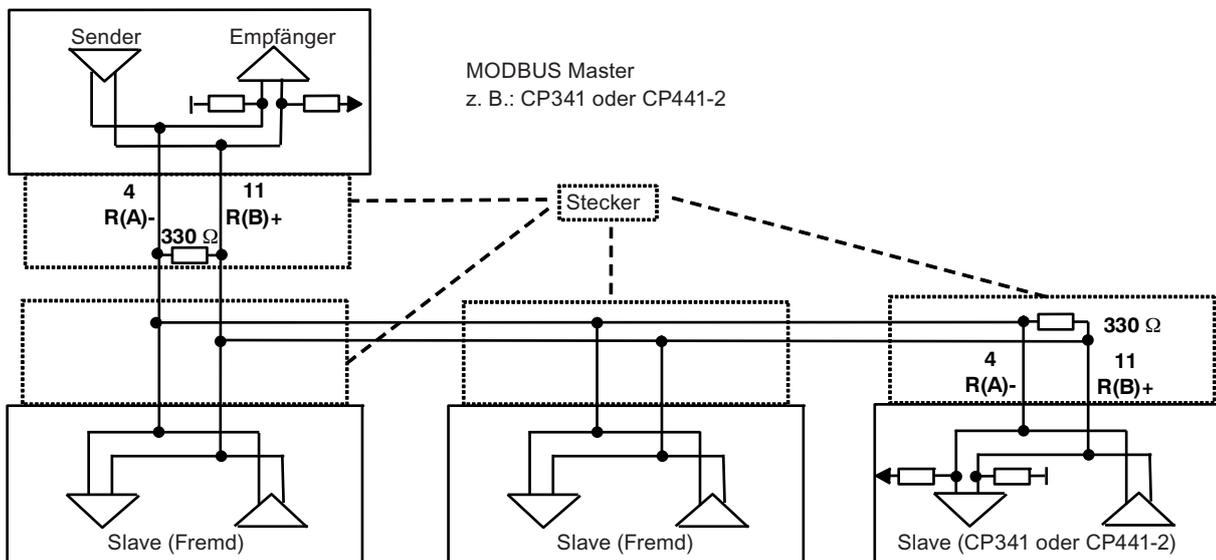
Verdrahtungsplan RS422 Mehrpunkt (MODBUS Multipoint)



ACHTUNG

In der Betriebsart **RS422** können sowohl CP 341 als auch CP 441-2 **nur als "Master"** betrieben werden, da sie ihre Sendeleitungen nicht in den Zustand "Tri State" schalten können!

Verdrahtungsplan RS485 Mehrpunkt (MODBUS Multipoint)



Für beide Betriebsarten gilt:

- Die GND-Leitung (PIN 8 bei CP 341/CP 441-2) muss zusätzlich über den gesamten Leitungsverlauf verbunden werden.
- Der Gehäuseschirm ist überall aufzulegen.
- Am jeweils letzten Empfänger einer Teilnehmerkette ist ein Abschlusswiderstand von ca. 330 Ω im Stecker einzulöten.
- Empfohlener Kabeltyp: LIYCY 3 x 2 x 0,14, R(A)/R(B) bzw. T(A)/T(B) verdreht
- Eine Verdrahtung mit "Stichleitungen" ist nicht zulässig.

Literaturhinweis

Modbus Protocol

- /1/ Gould Modbus Protocol
Reference Guide
PI-MBUS-300 Rev B
GOULD Electronics

Glossar

ANLAUF

Der Betriebszustand ANLAUF wird beim Übergang vom Betriebszustand STOP in den Betriebszustand RUN durchlaufen. Er kann durch folgende Ereignisse ausgelöst werden:

- Durch Betätigen des Betriebsartenschalters
- Nach Netz-Ein
- Durch Bedienung am Programmiergerät

Man unterscheidet zwischen den Anlaufarten Kaltstart, Neustart und Wiederanlauf.

Anwenderprogramm

Das Anwenderprogramm enthält alle Anweisungen und Vereinbarungen für die Signalverarbeitung, durch die eine Anlage oder ein Prozess gesteuert werden können. Das Anwenderprogramm wird bei SIMATIC S7 strukturiert aufgebaut und in kleinere Einheiten, die Bausteine, unterteilt.

Arbeitsspeicher

Der Arbeitsspeicher ist ein RAM-Speicher in der CPU, auf den der Prozessor während der Bearbeitung des Anwenderprogramms zugreift.

Automatisierungssystem

Ein Automatisierungssystem ist eine Speicherprogrammierbare Steuerung, die aus mindestens einer Zentralbaugruppe, verschiedenen Ein- und Ausgabebaugruppen sowie Bedien- und Beobachtungsgeräten besteht.

Baugruppe

Baugruppen sind steckbare Leiterplatten für Automatisierungssysteme.

Baugruppenparameter

Baugruppenparameter sind Werte, mit denen das Verhalten der Baugruppe eingestellt werden kann. Man unterscheidet zwischen statischen und dynamischen Baugruppenparametern.

Baustein

Bausteine sind durch ihre Funktion, ihre Struktur oder ihren Verwendungszweck abgegrenzte Teile des Anwenderprogramms. Es gibt bei STEP 7

- Codebausteine (FB, FC, OB, SFB, SFC)
- Datenbausteine (DB, SDB)
- Anwenderdefinierte Datentypen (UDT)

Bausteinanruf

Als Bausteinanruf bezeichnet man die Verzweigung der Programmbearbeitung in den aufgerufenen Baustein.

Bausteinparameter

Bausteinparameter sind Platzhalter innerhalb mehrfach nutzbarer Bausteine, die beim Aufruf des betreffenden Bausteins mit aktuellen Werten versorgt werden.

Betriebssystem der CPU

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

Betriebszustand

Die Automatisierungsfamilie SIMATIC S7 kennt vier unterschiedliche Betriebszustände:

STOP, ANLAUF und RUN. Die Funktionalität der CPUs in den einzelnen Betriebszuständen ist unterschiedlich.

CPU

Central Processing Unit = Zentralbaugruppe des S7-Automatisierungssystems mit Steuerwerk, Rechenwerk, Speicher, Systemprogramm und Schnittstellen zu Peripheriebaugruppen.

CRC

Cyclic Redundancy Check = Prüfsumme mit großer Fehlererkennungswahrscheinlichkeit.

Datenbaustein (DB)

Datenbausteine sind Bausteine, die Daten und Parameter enthalten, mit denen das Anwenderprogramm arbeitet. Sie enthalten im Gegensatz zu allen anderen Bausteinen keine Anweisungen. Es gibt globale Datenbausteine und Instanzdatenbausteine.

Auf die in den Datenbausteinen enthaltenen Daten kann absolut oder symbolisch zugegriffen werden. Komplexe Daten können strukturiert abgelegt werden.

Datentyp

Mithilfe eines Datentyps wird festgelegt, wie der Wert einer Variablen oder Konstanten im Anwenderprogramm verwendet werden soll. Bei SIMATIC S7 stehen dem Anwender zwei Arten von Datentypen nach IEC 1131-3 zur Verfügung:

- Elementare Datentypen
- Zusammengesetzte Datentypen

Default-Einstellung

Die Default-Einstellung ist eine sinnvolle Grundeinstellung, die immer dann verwendet wird, wenn kein anderer Wert eingegeben wird.

Diagnoseereignisse

Ein Diagnoseereignis führt zu einem Eintrag in den Diagnosepuffer der CPU. Die Diagnoseereignisse unterscheiden sich wie folgt:

- Fehler auf einer Baugruppe
- Fehler in der Prozessverdrahtung
- Systemfehler in der CPU
- Betriebszustandsübergänge der CPU
- Fehler im Anwenderprogramm
- Anwenderdefinierte Diagnoseereignisse.

Diagnosefunktionen

Die Diagnosefunktionen umfassen die gesamte Systemdiagnose und beinhalten Erkennen, Auswerten und Melden von Fehlern innerhalb des Automatisierungssystems.

Diagnosepuffer

Der Diagnosepuffer ist ein gepufferter Speicherbereich z. B. bei CPUs, der als Ringpuffer organisiert ist. Diagnoseereignisse sind in der Reihenfolge des Auftretens abgelegt.

Funktionsbausteine (FBs)

Funktionsbausteine sind Bestandteile des Anwenderprogramms und sind gemäß der IEC-Norm "Bausteine mit Gedächtnis". Das Gedächtnis für den Funktionsbaustein ist ein zugeordneter Datenbaustein, der "Instanz-Datenbaustein". Funktionsbausteine sind parametrierbar, Sie können sie mit und ohne Parameter nutzen.

Hardware

Als Hardware bezeichnet man die gesamte physikalische und technische Ausstattung eines Automatisierungssystems.

Instanz-Datenbaustein

Ein Instanz-Datenbaustein speichert die Formalparameter und statischen Daten von Funktionsbausteinen. Ein Instanz-Datenbaustein kann einem FB-Aufruf oder einer Aufrufhierarchie von Funktionsbausteinen zugeordnet sein.

Interrupt

Interrupt ist die Bezeichnung für die Unterbrechung der Programmbearbeitung im Prozessor eines Automatisierungssystems durch einen von außen anstehenden Alarm.

Kommunikationsprozessor

Ein Kommunikationsprozessor ist eine programmierbare Baugruppe für Kommunikationsaufgaben, z. B. Vernetzung oder Punkt-zu-Punkt-Kopplung.

Konfigurierung

Konfigurieren bedeutet Auswählen und Zusammenstellen einzelner Komponenten eines Automatisierungssystems.

Laden aus PG

Laden von Ladeobjekten, z. B. Codebausteinen, vom Programmiergerät in den Ladespeicher der Zentralbaugruppe (CPU).

Laden in PG

Laden von Ladeobjekten, z. B. Codebausteinen, aus dem Ladespeicher der Zentralbaugruppe in das Programmiergerät.

Online/Offline

Bei Online besteht eine Datenverbindung zwischen Automatisierungssystem und Programmiergerät, bei Offline nicht.

Online-Hilfe

STEP 7 bietet Ihnen die Möglichkeit, sich während des Arbeitens mit der Programmiersoftware kontextabhängige Hilfetexte am Bildschirm anzeigen zu lassen.

Operand

Ein Operand ist Teil einer STEP 7-Anweisung und sagt aus, womit der Prozessor etwas tun soll. Ein Operand kann sowohl absolut als auch symbolisch adressiert werden.

Parameter

Ein Parameter ist

- eine Variable eines STEP 7-Codebausteins,
- eine Variable zur Einstellung des Verhaltens einer Baugruppe.
Jede Baugruppe besitzt im Lieferzustand eine sinnvolle Grundeinstellung, die durch das Konfigurieren der Hardware verändert werden kann.

Es gibt 2 Arten von Parametern: statische und dynamische Parameter.

Parametrieroberfläche CP: Punkt-zu-Punkt-Kopplung parametrieren

Über die Parametrieroberfläche *CP Punkt-zu-Punkt-Kopplung parametrieren* parametrieren Sie die Schnittstelle des Kommunikationsprozessors und die treiberspezifischen Parameter.

Für jeden ladbaren Treiber wird der Standardumfang erweitert.

Parametrierung

Parametrieren ist das Einstellen des Verhaltens einer Baugruppe.

Protokoll

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

Prozedur

Eine Prozedur ist der Ablauf einer Datenübertragung nach einem bestimmten Protokoll.

Prozessabbild

Das Prozessabbild ist ein besonderer Speicherbereich im Automatisierungssystem. Am Anfang des zyklischen Programms werden die Signalzustände der Eingabebaugruppen zum Prozessabbild der Eingänge übertragen. Am Ende des zyklischen Programms wird das Prozessabbild der Ausgänge als Signalzustand zu den Ausgabebaugruppen übertragen.

Punkt-zu-Punkt-Kopplung

Bei der Punkt-zu-Punkt-Kopplung bildet der Kommunikationsprozessor die Schnittstelle zwischen einer Speicherprogrammierbaren Steuerung und einem Kommunikationspartner.

Rack

Ein Rack ist ein Baugruppenträger, der Steckplätze für Baugruppen enthält.

Schnittstellenmodul

Das CP 441-2 Schnittstellenmodul setzt die Signale physikalisch um. Durch Austausch der steckbaren Schnittstellenmodule können Sie den Kommunikationsprozessor an die Physik des Kommunikationspartners anpassen.

Software

Die Software ist die Gesamtheit aller Programme, die auf einem Rechner eingesetzt werden. Zur Software gehören das Betriebssystem und die Anwenderprogramme.

STEP 7

STEP 7 ist die Programmiersoftware von SIMATIC S7.

Systembausteine

Systembausteine unterscheiden sich von den anderen Bausteinen dadurch, dass sie im System S7-300/400 bereits integriert sind und für bereits definierte Systemfunktionen zur Verfügung stehen. Es gibt Systemdatenbausteine, Systemfunktionen und Systemfunktionsbausteine.

Systemfunktionen (SFCs)

Eine Systemfunktion SFC ist eine im Betriebssystem der S7-CPU integrierte Funktion ohne Gedächtnis, die bei Bedarf im Anwenderprogramm wie eine Funktion FC aufgerufen werden kann.

Systemfunktionsbausteine (SFBs)

Ein Systemfunktionsbaustein SFB ist ein im Betriebssystem der S7-CPU integrierter Funktionsbaustein mit Gedächtnis, der bei Bedarf im Anwenderprogramm wie ein Funktionsbaustein FB aufgerufen werden kann.

Tool

Ein Tool ist ein Software-Werkzeug zum Projektieren und Programmieren.

Variable

Eine Variable ist ein Datum mit variablem Inhalt, das im STEP 7-Anwenderprogramm verwendet werden kann. Eine Variable besteht aus einem Operanden, z. B. M 3.1, und einem Datentyp, z. B. Bool, und kann mit einem Symbol, z. B. BAND_EIN, gekennzeichnet werden.

Verbindungsprojektierung (nur CP 441-2)

Unter Verbindungsprojektierung versteht man die Angabe einer Verbindungs-ID im Systemfunktionsbaustein. Über die Verbindungs-ID können die Systemfunktionsbausteine zwischen zwei Kommunikationsendpunkten kommunizieren.

Zyklische Programmbearbeitung

Bei der zyklischen Programmbearbeitung läuft das Anwenderprogramm in einer sich ständig wiederholenden Programmschleife ab, die Zyklus genannt wird.

Zykluszeit

Die Zykluszeit ist die Zeit, welche die CPU für eine einmalige Bearbeitung des Anwenderprogramms benötigt.

Index

A

Adressdarstellung, 15
Anlaufverhalten, 36
 Ladevorgang, 37
Antwortüberwachungszeit, 30
Anzeigeelemente (LED), 79

B

Baudrate, 28
Betriebsart, 30
Broadcast, 39

C

CPU Parametrierung, 37
CRC, 40

D

De-Installation, 22
Diagnose, 79
Dongle, 13, 17

F

Funktionscode, 11
 FC-01, 14, 47
 FC-02, 14, 49
 FC-03, 14, 51
 FC-04, 14, 53
 FC-05, 14, 55
 FC-06, 14, 57
 FC-07, 59
 FC-08, 61
 FC-11, 63
 FC-12, 64
 FC-15, 14, 66
 FC-16, 14, 68

L

Ladespeicher, 13

M

Mehrpunktverbindung, 18

P

Parametrieren, 24
Parametrierung, 24, 28
Parität, 29

S

Schnittstelle
 X27, 32
Schnittstellenmodule
 RS 232C, 11, 18
 TTY, 18
 X27, 18
Systat, 73, 77
SYSTAT, 84
 Ereignisklasse, 85
 Ereignisnummer, 85

T

Telegrammaufbau, 39

U

Übertragungsfehler, 33
Übertragungszeiten, 113

V

Verbindungs-ID, 27, 75, 103
Verbindungsprojektierung, 27

Z

Zeichenverzugszeit ZVZ, 30, 41

