

System 300S

CP | 341-1AH01 | Handbuch

HB130 | CP | 341-1AH01 | de | 18-41

CP 341 RS232



YASKAWA Europe GmbH
Hauptstraße 185
65760 Eschborn
Deutschland
Tel.: +49 6196 569-300
Fax: +49 6196 569-398
E-Mail: info@yaskawa.eu.com
Internet: www.yaskawa.eu.com

Inhaltsverzeichnis

| | | |
|----------|--|-----------|
| 1 | Allgemeines | 5 |
| 1.1 | Copyright © YASKAWA Europe GmbH..... | 5 |
| 1.2 | Über dieses Handbuch..... | 6 |
| 1.3 | Sicherheitshinweise..... | 7 |
| 2 | Grundlagen | 8 |
| 2.1 | Sicherheitshinweis für den Benutzer..... | 8 |
| 2.2 | Grundlagen - ISO/OSI-Schichtenmodell..... | 9 |
| 2.3 | CP 341-1AH01..... | 10 |
| 2.4 | Allgemeine Daten..... | 11 |
| 2.4.1 | Einsatz unter erschwerten Betriebsbedingungen..... | 12 |
| 3 | Montage und Aufbaurichtlinien | 13 |
| 3.1 | Einbaumaße..... | 13 |
| 3.2 | Montage Standard-Bus..... | 14 |
| 3.3 | Aufbaurichtlinien..... | 16 |
| 4 | Hardwarebeschreibung | 19 |
| 4.1 | Leistungsmerkmale..... | 19 |
| 4.2 | Aufbau..... | 20 |
| 4.3 | Technische Daten..... | 26 |
| 5 | Einsatz | 29 |
| 5.1 | Schnelleinstieg..... | 29 |
| 5.2 | Hardware-Konfiguration..... | 31 |
| 5.2.1 | Eigenschaften..... | 32 |
| 5.3 | Kommunikation mit dem Anwenderprogramm..... | 34 |
| 5.4 | Zugriff auf RS232-Begleitsignale..... | 36 |
| 5.5 | Firmwareupdate..... | 38 |
| 5.5.1 | Firmwareupdate über Siemens Parametriertool..... | 38 |
| 5.5.2 | Firmwareupdate bei Einsatz einer SPEED7 CPU..... | 39 |
| 5.5.3 | CP-Firmwarestand anzeigen..... | 40 |
| 6 | Kommunikationsprotokolle | 41 |
| 6.1 | Übersicht..... | 41 |
| 6.2 | ASCII..... | 42 |
| 6.2.1 | ASCII - Parameter | 43 |
| 6.3 | 3964(R) | 46 |
| 6.3.1 | Grundlagen 3964(R)..... | 46 |
| 6.3.2 | Vorgehensweise..... | 47 |
| 6.3.3 | 3964(R) - Parameter | 48 |
| 6.4 | Modbus..... | 51 |
| 6.4.1 | Grundlagen Modbus..... | 51 |
| 6.4.2 | Modbus Master - Parameter..... | 52 |
| 6.4.3 | Modbus Master - Funktionsweise..... | 58 |
| 6.4.4 | Modbus Master - Funktionscodes..... | 62 |
| 6.4.5 | Modbus Slave - Parameter..... | 70 |
| 6.4.6 | Modbus Slave - Funktionsweise..... | 74 |
| 6.4.7 | Modbus Slave - Kommunikation mit Anwenderprogramm..... | 77 |
| 6.4.8 | Modbus Slave - Funktionscodes..... | 83 |

| | | |
|----------|--|-----------|
| 7 | Diagnose und Fehlerverhalten..... | 96 |
| 7.1 | Übersicht Diagnosefunktionen..... | 96 |
| 7.2 | Diagnose über FB <i>STATUS</i> | 97 |
| 7.3 | Diagnose über Diagnosepuffer..... | 107 |
| 7.4 | Diagnose über Diagnosealarm..... | 108 |

1 Allgemeines

1.1 Copyright © YASKAWA Europe GmbH

All Rights Reserved

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

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

Zur Genehmigung von Vervielfältigung oder Verteilung wenden Sie sich bitte an:
YASKAWA Europe GmbH, European Headquarters, Hauptstraße 185, 65760 Eschborn, Germany

Tel.: +49 6196 569 300

Fax.: +49 6196 569 398

E-Mail: info@yaskawa.eu.com

Internet: www.yaskawa.eu.com



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

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

EG-Konformitätserklärung

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

Informationen zur Konformitätserklärung

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

Warenzeichen

VIPA, SLIO, System 100V, System 200V, System 300V, System 300S, System 400V, System 500S und Commander Compact sind eingetragene Warenzeichen der YASKAWA Europe GmbH.

SPEED7 ist ein eingetragenes Warenzeichen der YASKAWA Europe GmbH.

SIMATIC, STEP, SINEC, TIA Portal, S7-300, S7-400 und S7-1500 sind eingetragene Warenzeichen der Siemens AG.

Microsoft und Windows sind eingetragene Warenzeichen von Microsoft Inc., USA.

Portable Document Format (PDF) und Postscript sind eingetragene Warenzeichen von Adobe Systems, Inc.

Alle anderen erwähnten Firmennamen und Logos sowie Marken- oder Produktnamen sind Warenzeichen oder eingetragene Warenzeichen ihrer jeweiligen Eigentümer.

Dokument-Support

Wenden Sie sich an Ihre Landesvertretung der YASKAWA Europe GmbH, wenn Sie Fehler anzeigen oder inhaltliche Fragen zu diesem Dokument stellen möchten. Sie können YASKAWA Europe GmbH über folgenden Kontakt erreichen:

E-Mail: Documentation.HER@yaskawa.eu.com

Über dieses Handbuch

Technischer Support

Wenden Sie sich an Ihre Landesvertretung der YASKAWA Europe GmbH, wenn Sie Probleme mit dem Produkt haben oder Fragen zum Produkt stellen möchten. Ist eine solche Stelle nicht erreichbar, können Sie den Yaskawa Kundenservice über folgenden Kontakt erreichen:

YASKAWA Europe GmbH,
European Headquarters, Hauptstraße 185, 65760 Eschborn, Germany
Tel.: +49 6196 569 500 (Hotline)
E-Mail: support@yaskawa.eu.com

1.2 Über dieses Handbuch

Zielsetzung und Inhalt

Das Handbuch beschreibt den CP 341-1AH01 aus dem System 300S von Yaskawa. Beschrieben wird Aufbau, Projektierung und Anwendung.

| Produkt | Best.-Nr. | ab Stand: | |
|--------------|-----------|-----------|--------|
| | | CP-HW | CP-FW |
| CP 341 RS232 | 341-1AH01 | 01 | V1.3.1 |

Zielgruppe

Das Handbuch ist geschrieben für Anwender mit Grundkenntnissen in der Automatisierungstechnik.

Aufbau des Handbuchs

Das Handbuch ist in Kapitel gegliedert. Jedes Kapitel beschreibt eine abgeschlossene Thematik.

Orientierung im Dokument

Als Orientierungshilfe stehen im Handbuch zur Verfügung:

- Gesamt-Inhaltsverzeichnis am Anfang des Handbuchs
- Verweise mit Seitenangabe

Verfügbarkeit

Das Handbuch ist verfügbar in:

- gedruckter Form auf Papier
- in elektronischer Form als PDF-Datei (Adobe Acrobat Reader)

Piktogramme Signalwörter

Wichtige Textteile sind mit folgenden Piktogrammen und Signalworten hervorgehoben:



GEFAHR!

Unmittelbare oder drohende Gefahr. Personenschäden sind möglich.



VORSICHT!

Bei Nichtbefolgen sind Sachschäden möglich.



Zusätzliche Informationen und nützliche Tipps.

1.3 Sicherheitshinweise

Bestimmungsgemäße Verwendung

Das System ist konstruiert und gefertigt für:

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



GEFAHR!

Das Gerät ist nicht zugelassen für den Einsatz

- in explosionsgefährdeten Umgebungen (EX-Zone)

Dokumentation

Handbuch zugänglich machen für alle Mitarbeiter in

- Projektierung
- Installation
- Inbetriebnahme
- Betrieb



VORSICHT!

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

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

Entsorgung

Zur Entsorgung des Geräts nationale Vorschriften beachten!

2 Grundlagen

2.1 Sicherheitshinweis für den Benutzer

Handhabung elektrostatisch gefährdeter Baugruppen

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



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

Versenden von Baugruppen

Verwenden Sie für den Versand immer die Originalverpackung.

Messen und Ändern von elektrostatisch gefährdeten Baugruppen

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

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

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



VORSICHT!

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

2.2 Grundlagen - ISO/OSI-Schichtenmodell

Übersicht

Das ISO/OSI-Schichtenmodell basiert auf einem Vorschlag, der von der International Standards Organization (ISO) entwickelt wurde. Es stellt den ersten Schritt zur internationalen Standardisierung der verschiedenen Protokolle dar. Das Modell trägt den Namen ISO-OSI-Schichtenmodell. OSI steht für **O**pen **S**ystem **I**nterconnection, die Kommunikation offener Systeme. Das ISO/OSI-Schichtenmodell ist keine Netzwerkarchitektur, da die genauen Dienste und Protokolle, die in jeder Schicht verwendet werden, nicht festgelegt sind. Sie finden in diesem Modell lediglich Informationen über die Aufgaben, welche die jeweilige Schicht zu erfüllen hat. Jedes offene Kommunikationssystem basiert heutzutage auf dem durch die Norm ISO 7498 beschriebenen ISO/OSI Referenzmodell. Das Referenzmodell strukturiert Kommunikationssysteme in insgesamt 7 Schichten, denen jeweils Teilaufgaben in der Kommunikation zugeordnet sind. Dadurch wird die Komplexität der Kommunikation auf verschiedene Ebenen verteilt und somit eine größere Übersichtlichkeit erreicht.

Folgende Schichten sind definiert:

- Schicht 7 - Application Layer (Anwendung)
- Schicht 6 - Presentation Layer (Darstellung)
- Schicht 5 - Session Layer (Sitzung)
- Schicht 4 - Transport Layer (Transport)
- Schicht 3 - Network Layer (Netzwerk)
- Schicht 2 - Data Link Layer (Sicherung)
- Schicht 1 - Physical Layer (Bitübertragung)

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

Schicht 1 - Bitübertragungsschicht (physical layer)

Die Bitübertragungsschicht beschäftigt sich mit der Übertragung von Bits über einen Kommunikationskanal. Allgemein befasst sich diese Schicht mit den mechanischen, elektrischen und prozeduralen Schnittstellen und mit dem physikalischen Übertragungsmedium, das sich unterhalb der Bitübertragungsschicht befindet:

- Wie viel Volt entsprechen einer logischen 0 bzw. 1?
- Wie lange muss die Spannung für ein Bit anliegen?
- Pinbelegung der verwendeten Schnittstelle.

Schicht 2 - Sicherungsschicht (data link layer)

Diese Schicht hat die Aufgabe, die Übertragung von Bitstrings zwischen zwei Teilnehmern sicherzustellen. Dazu gehören die Erkennung und Behebung bzw. Weitermeldung von Übertragungsfehlern, sowie die Flusskontrolle. Die Sicherungsschicht verwandelt die zu übertragenden Rohdaten in eine Datenreihe. Hier werden Rahmengrenzen beim Sender eingefügt und beim Empfänger erkannt. Dies wird dadurch erreicht, dass am Anfang und am Ende eines Rahmens spezielle Bitmuster gesetzt werden. In der Sicherungsschicht wird häufig noch eine Flussregelung und eine Fehlererkennung integriert. Die Datensicherungsschicht ist in zwei Unterschichten geteilt, die LLC- und die MAC-Schicht. Die MAC (**M**edia **A**ccess **C**ontrol) ist die untere Schicht und steuert die Art, wie Sender einen einzigen Übertragungskanal gemeinsam nutzen. Die LLC (**L**ogical **L**ink **C**ontrol) ist die obere Schicht und stellt die Verbindung für die Übertragung der Datenrahmen von einem Gerät zum anderen her.

Schicht 3 - Netzwerkschicht (network layer)

Die Netzwerkschicht wird auch Vermittlungsschicht genannt. Die Aufgabe dieser Schicht besteht darin, den Austausch von Binärdaten zwischen nicht direkt miteinander verbundenen Stationen zu steuern. Sie ist für den Ablauf der logischen Verknüpfungen von Schicht 2-Verbindungen zuständig. Dabei unterstützt diese Schicht die Identifizierung der einzelnen Netzwerkadressen und den Auf- bzw. Abbau von logischen Verbindungskanälen. IP basiert auf Schicht 3. Eine weitere Aufgabe der Schicht 3 besteht in der priorisierten Übertragung von Daten und die Fehlerbehandlung von Datenpaketen. IP (Internet Protokoll) basiert auf Schicht 3.

| | |
|---|---|
| Schicht 4 - Transportschicht (transport layer) | Die Aufgabe der Transportschicht besteht darin, Netzwerkstrukturen mit den Strukturen der höheren Schichten zu verbinden, indem sie Nachrichten der höheren Schichten in Segmente unterteilt und an die Netzwerkschicht weiterleitet. Hierbei wandelt die Transportschicht die Transportadressen in Netzwerkadressen um. Gebräuchliche Transportprotokolle sind: TCP, SPX, NWLink und NetBEUI. |
| Schicht 5 - Sitzungsschicht (session layer) | Die Sitzungsschicht wird auch Kommunikationssteuerungsschicht genannt. Sie erleichtert die Kommunikation zwischen Service-Anbieter und Requestor durch Aufbau und Erhaltung der Verbindung, wenn das Transportsystem kurzzeitig ausgefallen ist. Auf dieser Ebene können logische Benutzer über mehrere Verbindungen gleichzeitig kommunizieren. Fällt das Transportsystem aus, so ist es die Aufgabe, gegebenenfalls eine neue Verbindung aufzubauen. Darüber hinaus werden in dieser Schicht Methoden zur Steuerung und Synchronisation bereitgestellt. |
| Schicht 6 - Darstellungsschicht (presentation layer) | Auf dieser Ebene werden die Darstellungsformen der Nachrichten behandelt, da bei verschiedenen Netzsystemen unterschiedliche Darstellungsformen benutzt werden. Die Aufgabe dieser Schicht besteht in der Konvertierung von Daten in ein beiderseitig akzeptiertes Format, damit diese auf den verschiedenen Systemen lesbar sind. Hier werden auch Kompressions-/Dekompressions- und Verschlüsselungs-/ Entschlüsselungsverfahren durchgeführt. Man bezeichnet diese Schicht auch als Dolmetscherdienst. Eine typische Anwendung dieser Schicht ist die Terminalemulation. |
| Schicht 7 - Anwendungsschicht (application layer) | Die Anwendungsschicht stellt sich als Bindeglied zwischen der eigentlichen Benutzeranwendung und dem Netzwerk dar. Sowohl die Netzwerk-Services wie Datei-, Druck-, Nachrichten-, Datenbank- und Anwendungs-Service als auch die zugehörigen Regeln gehören in den Aufgabenbereich dieser Schicht. Diese Schicht setzt sich aus einer Reihe von Protokollen zusammen, die entsprechend den wachsenden Anforderungen der Benutzer ständig erweitert werden. |

2.3 CP 341-1AH01

Aufbau/Maße

Maße Grundgehäuse:

- 1fach breit: (BxHxT) in mm: 40x125x120

Kompatibilität

- Der CP 341-1AH01 ist funktionskompatibel zum Siemens CP 341 (6ES7 341-1AH01-0AE0).
- Die Projektierung erfolgt im Siemens SIMATIC Manager.

2.4 Allgemeine Daten

Konformität und Approbation

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

Personenschutz und Geräteschutz

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

Umgebungsbedingungen gemäß EN 61131-2

| | | |
|-----------------------------|---------------|--|
| Klimatisch | | |
| Lagerung /Transport | EN 60068-2-14 | -25...+70°C |
| Betrieb | | |
| Horizontaler Einbau hängend | EN 61131-2 | 0...+60°C |
| Horizontaler Einbau liegend | EN 61131-2 | 0...+40°C |
| Vertikaler Einbau | EN 61131-2 | 0...+40°C |
| Luftfeuchtigkeit | EN 60068-2-30 | RH1 (ohne Betauung, relative Feuchte 10 ... 95%) |
| Verschmutzung | EN 61131-2 | Verschmutzungsgrad 2 |
| Aufstellhöhe max. | - | 2000m |
| Mechanisch | | |
| Schwingung | EN 60068-2-6 | 1g, 9Hz ... 150Hz |
| Schock | EN 60068-2-27 | 15g, 11ms |

Allgemeine Daten > Einsatz unter erschwerten Betriebsbedingungen

| Montagebedingungen | | |
|--------------------|---|-------------------------|
| Einbauort | - | Im Schaltschrank |
| Einbaulage | - | Horizontal und vertikal |

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

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

2.4.1 Einsatz unter erschwerten Betriebsbedingungen



Ohne zusätzlich schützende Maßnahmen dürfen die Produkte nicht an Orten mit erschwerten Betriebsbedingungen; z.B. durch:

- Staubentwicklung
- chemisch aktive Substanzen (ätzende Dämpfe oder Gase)
- starke elektrische oder magnetische Felder

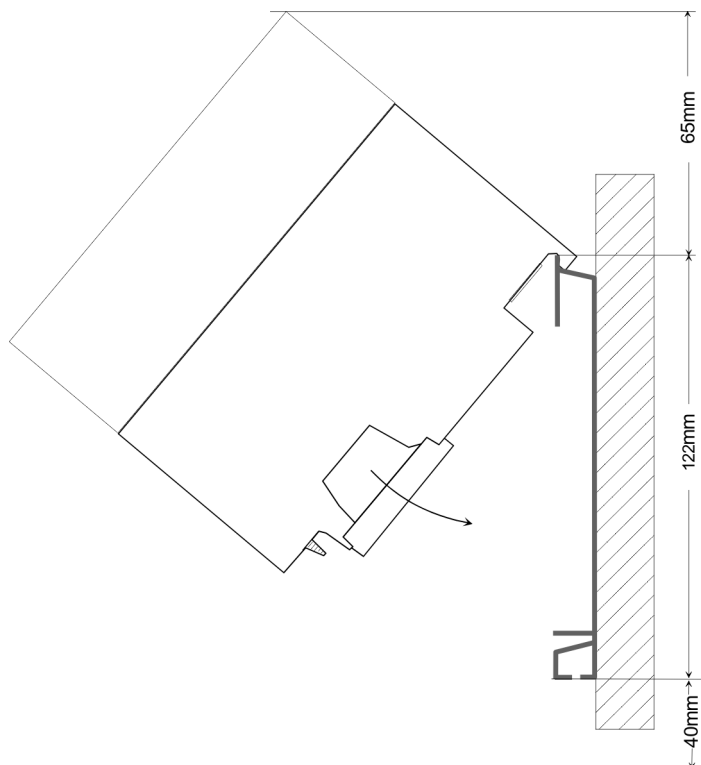
eingesetzt werden!

3 Montage und Aufbaurichtlinien

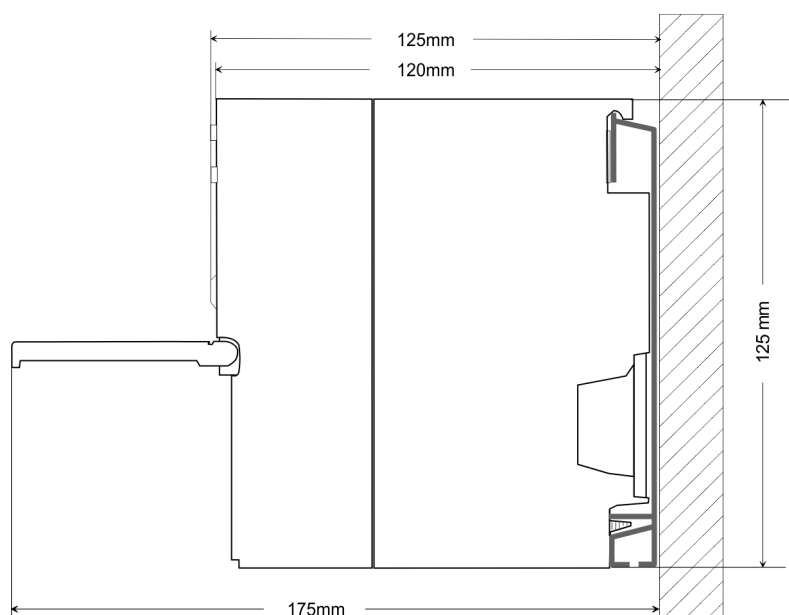
3.1 Einbaumaße

Maße Grundgehäuse 1fach breit (BxHxT) in mm: 40 x 125 x 120

Montagemaße



Maße montiert



3.2 Montage Standard-Bus

Allgemein

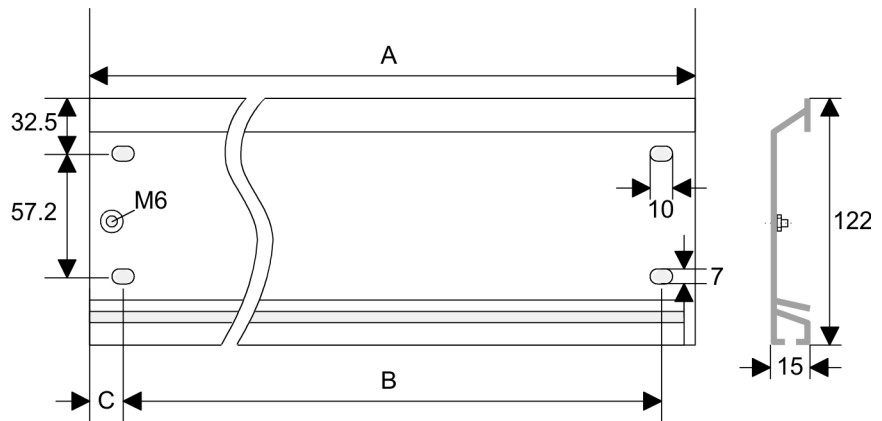
Die einzelnen Module werden direkt auf eine Profilschiene montiert und über den Rückwandbus-Verbinder verbunden. Vor der Montage ist der Rückwandbus-Verbinder von hinten an das Modul zu stecken. Die Rückwandbus-Verbinder sind im Lieferumfang der Peripherie-Module enthalten.

Profilschiene

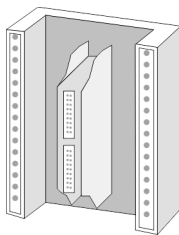
| Bestellnummer | A | B | C |
|---------------|------|---------------------|-----|
| 390-1AB60 | 160 | 140 | 10 |
| 390-1AE80 | 482 | 466 | 8,3 |
| 390-1AF30 | 530 | 500 | 15 |
| 390-1AJ30 | 830 | 800 | 15 |
| 390-9BC00* | 2000 | Bohrungen nur links | 15 |

*) Verpackungseinheit 10 Stück

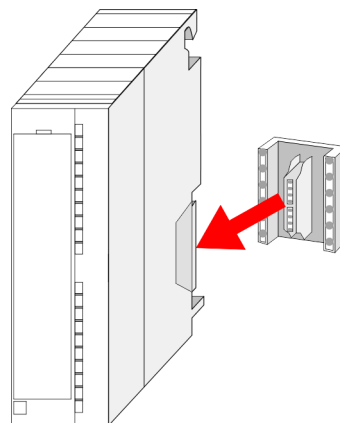
Maße in mm



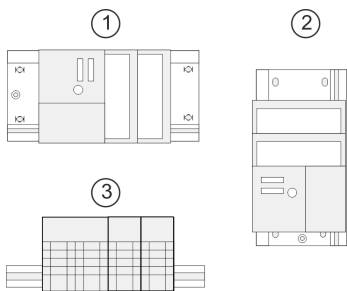
Busverbinder



Für die Kommunikation der Module untereinander wird beim System 300S ein Rückwandbus-Verbinder eingesetzt. Die Rückwandbus-Verbinder sind im Lieferumfang der Peripherie-Module enthalten und werden vor der Montage von hinten an das Modul gesteckt.



Montagemöglichkeiten

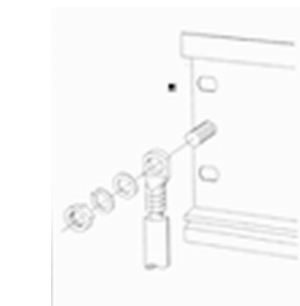


Beachten Sie bitte die hierbei zulässigen Umgebungstemperaturen:

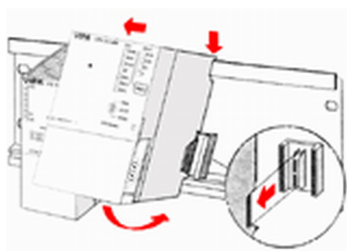
- 1 waagrecht Aufbau: von 0 bis 60°C
- 2 senkrechter Aufbau: von 0 bis 40°C
- 3 liegender Aufbau: von 0 bis 40°C

Vorgehensweise

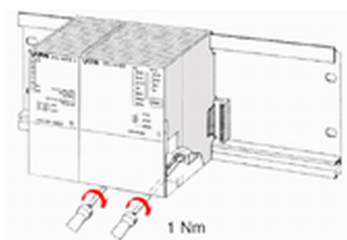
Sofern Sie keine SPEED-Bus-Module einsetzen, erfolgt die Montage nach folgender Vorgehensweise:



1. ➤ Verschrauben Sie die Profilschiene mit dem Untergrund (Schraubengröße: M6) so, dass mindestens 65mm Raum oberhalb und 40mm unterhalb der Profilschiene bleibt.
2. ➤ Achten Sie bei geerdetem Untergrund auf eine niederohmige Verbindung zwischen Profilschiene und Untergrund.
3. ➤ Verbinden Sie die Profilschiene mit dem Schutzleiter. Für diesen Zweck befindet sich auf der Profilschiene ein Stehbolzen mit M6-Gewinde.
4. ➤ Der Mindestquerschnitt der Leitung zum Schutzleiter muss 10mm² betragen.



5. ➤ Hängen Sie die Spannungsversorgung ein und schieben Sie diese nach links bis an den Erdungsbolzen der Profilschiene.
6. ➤ Schrauben Sie die Spannungsversorgung fest.
7. ➤ Nehmen Sie einen Rückwandbus-Verbinder und stecken Sie ihn wie gezeigt von hinten an die CPU.
8. ➤ Hängen Sie die CPU rechts von der Spannungsversorgung ein und schieben Sie diese bis an die Spannungsversorgung.



9. ➤ Klappen Sie die CPU nach unten und schrauben Sie die CPU wie gezeigt fest.
10. ➤ Verfahren Sie auf die gleiche Weise mit Ihren Peripherie-Modulen, indem Sie jeweils einen Rückwandbus-Verbinder stecken, Ihr Modul rechts neben dem Vorgänger-Modul einhängen, dieses nach unten klappen, in den Rückwandbus-Verbinder des Vorgängermoduls einrasten lassen und das Modul festschrauben.

3.3 Aufbaurichtlinien

Allgemeines

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

Was bedeutet EMV?

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

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

Mögliche Störeinträge

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

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

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

Man unterscheidet:

- galvanische Kopplung
- kapazitive Kopplung
- induktive Kopplung
- Strahlungskopplung

Grundregeln zur Sicherstellung der EMV

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

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

- Achten Sie auf die einwandfreie Befestigung der Leitungsschirme.
 - Datenleitungen sind geschirmt zu verlegen.
 - Analogleitungen sind geschirmt zu verlegen. Bei der Übertragung von Signalen mit kleinen Amplituden kann das einseitige Auflegen des Schirms vorteilhaft sein.
 - Leitungen für Frequenzumrichter, Servo- und Schrittmotore sind geschirmt zu verlegen.
 - Legen Sie die Leitungsschirme direkt nach dem Schrankeintritt großflächig auf eine Schirm-/Schutzleiterschiene auf, und befestigen Sie die Schirme mit Kabelschellen.
 - Achten Sie darauf, dass die Schirm-/Schutzleiterschiene impedanzarm mit dem Schrank verbunden ist.
 - Verwenden Sie für geschirmte Datenleitungen metallische oder metallisierte Steckergehäuse.
- Setzen Sie in besonderen Anwendungsfällen spezielle EMV-Maßnahmen ein.
 - Erwägen Sie bei Induktivitäten den Einsatz von Löschiern.
 - Beachten Sie, dass bei Einsatz von Leuchtstofflampen sich diese negativ auf Signalleitungen auswirken können.
- Schaffen Sie ein einheitliches Bezugspotenzial und erden Sie nach Möglichkeit alle elektrischen Betriebsmittel.
 - Achten Sie auf den gezielten Einsatz der Erdungsmaßnahmen. Das Erden der Steuerung dient als Schutz- und Funktionsmaßnahme.
 - Verbinden Sie Anlagenteile und Schränke mit Ihrer SPS sternförmig mit dem Erde/Schutzleitersystem. Sie vermeiden so die Bildung von Erdschleifen.
 - Verlegen Sie bei Potenzialdifferenzen zwischen Anlagenteilen und Schränken ausreichend dimensionierte Potenzialausgleichsleitungen.

Schirmung von Leitungen

Elektrische, magnetische oder elektromagnetische Störfelder werden durch eine Schirmung geschwächt; man spricht hier von einer Dämpfung. Über die mit dem Gehäuse leitend verbundene Schirmschiene werden Störströme auf Kabelschirme zur Erde hin abgeleitet. Hierbei ist darauf zu achten, dass die Verbindung zum Schutzleiter impedanzarm ist, da sonst die Störströme selbst zur Störquelle werden.

Bei der Schirmung von Leitungen ist folgendes zu beachten:

- Verwenden Sie möglichst nur Leitungen mit Schirmgeflecht.
- Die Deckungsdichte des Schirmes sollte mehr als 80% betragen.
- In der Regel sollten Sie die Schirme von Leitungen immer beidseitig auflegen. Nur durch den beidseitigen Anschluss der Schirme erreichen Sie eine gute Störunterdrückung im höheren Frequenzbereich. Nur im Ausnahmefall kann der Schirm auch einseitig aufgelegt werden. Dann erreichen Sie jedoch nur eine Dämpfung der niedrigen Frequenzen. Eine einseitige Schirmanbindung kann günstiger sein, wenn:
 - die Verlegung einer Potenzialausgleichsleitung nicht durchgeführt werden kann.
 - Analogsignale (einige mV bzw. μA) übertragen werden.
 - Folienschirme (statische Schirme) verwendet werden.
- Benutzen Sie bei Datenleitungen für serielle Kopplungen immer metallische oder metallisierte Stecker. Befestigen Sie den Schirm der Datenleitung am Steckergehäuse. Schirm nicht auf den PIN 1 der Steckerleiste auflegen!
- Bei stationärem Betrieb ist es empfehlenswert, das geschirmte Kabel unterbrechungsfrei abzuisolieren und auf die Schirm-/Schutzleiterschiene aufzulegen.
- Benutzen Sie zur Befestigung der Schirmgeflecht Kabelschellen aus Metall. Die Schellen müssen den Schirm großflächig umschließen und guten Kontakt ausüben.
- Legen Sie den Schirm direkt nach Eintritt der Leitung in den Schrank auf eine Schirmschiene auf. Führen Sie den Schirm bis zu Ihrer SPS weiter, legen Sie ihn dort jedoch nicht erneut auf!



VORSICHT!

Bitte bei der Montage beachten!

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

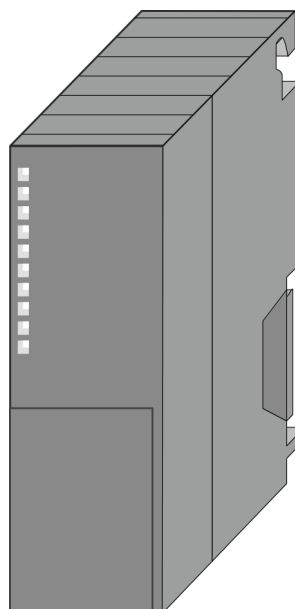
Abhilfe: Potentialausgleichsleitung.

4 Hardwarebeschreibung

4.1 Leistungsmerkmale

CP 341 RS232

- RS232-Schnittstelle potenzialgetrennt zum Rückwandbus
- Funktionskompatibel zu Siemens CP 341 (6ES7 341-1AH01-0AE0)
- Unterstützt werden die Protokolle:
 - ASCII
 - 3964(R)
 - Modbus Master ASCII / RTU (kein Hardware-Dongle erforderlich)
 - Modbus Slave RTU (kein Hardware-Dongle erforderlich)
- Parametrierung CP 341 über Parametrierpaket von Siemens
 - CP 341: Punkt-zu-Punkt-Kopplung parametrieren ab V 5.0
- Bis zu 250 Telegramme innerhalb der 1024Byte großen Empfangs- bzw. Sendepuffer
- Datenübertragungsrate parametrierbar bis 76,8kBit/s
- Spannungsversorgung über Rückwandbus

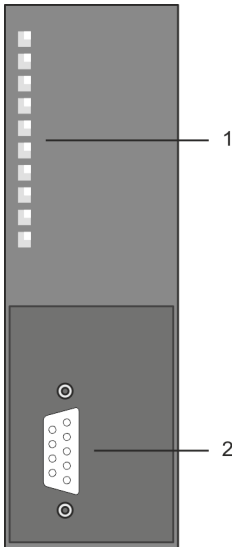


Bestelldaten

| Typ | Bestellnummer | Beschreibung |
|--------------|---------------|--|
| CP 341 RS232 | 341-1AH01 | CP 341 mit RS232-Schnittstelle Protokolle: ASCII, 3964(R), Modbus Master (ASCII / RTU), Modbus Slave (RTU) |

4.2 Aufbau

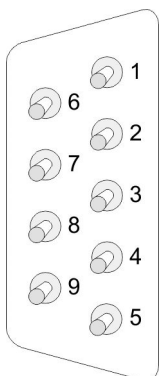
CP 341-1AH01



- 1 LED Statusanzeigen
Folgende Komponente befindet sich unter der Frontklappe:
- 2 RS232-Schnittstelle

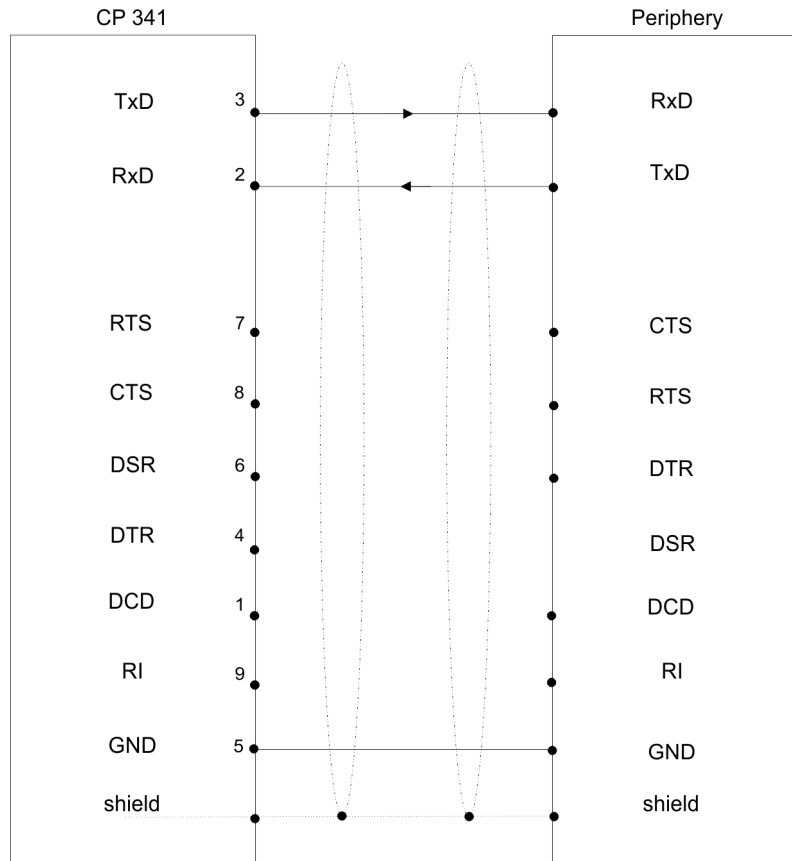
RS232-Schnittstelle

- Pinkompatibel zu Siemens CP 341 (6ES7 341-1AH01-0AE0)
- Logische Zustände als Spannungspegel
- Punkt-zu-Punkt-Kopplung mit serieller Vollduplex-Übertragung
- Datenübertragung bis 15m Entfernung
- Datenübertragungsrate bis 76800bit/s

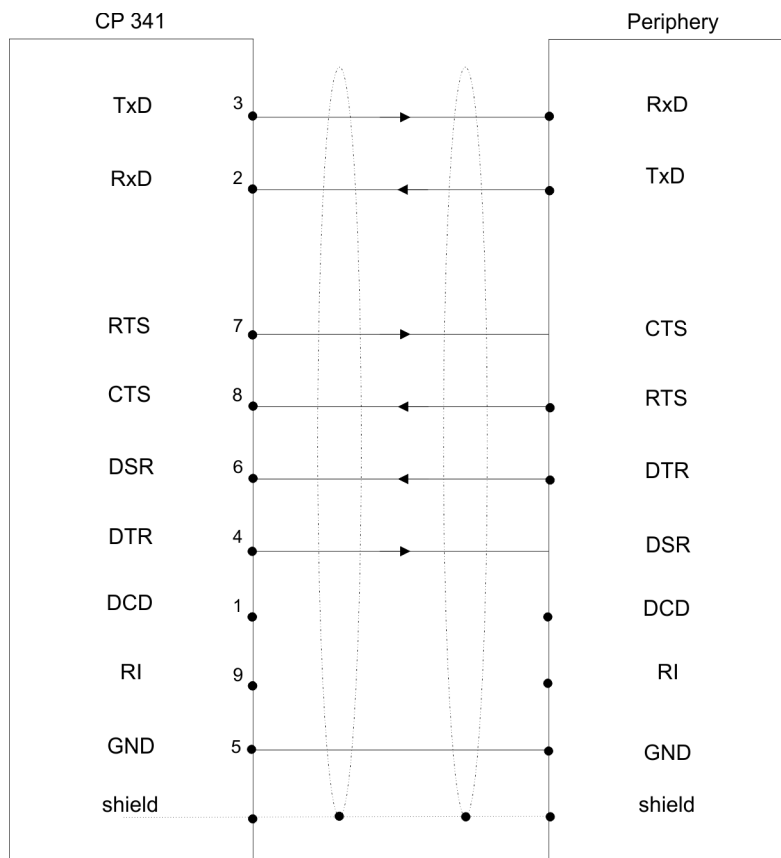
X2: 9poliger SubD-Stecker

| Pin | Bezeichnung | | Ein-/Ausgang | Beschreibung |
|-----|-------------|---------------------|--------------|--|
| 1 | DCD | Data Carrier Detect | Eingang | Trägersignal bei Anschluss eines Modems. |
| 2 | RxD | Receive Data | Eingang | <ul style="list-style-type: none"> ■ Empfangsdaten: <ul style="list-style-type: none"> – Empfangsleitung wird vom Kommunikationspartner auf logisch "1" gehalten |
| 3 | TxD | Transmit Data | Ausgang | <ul style="list-style-type: none"> ■ Sendedaten: <ul style="list-style-type: none"> – Sendeleitung wird vom CP im Ruhezustand auf logisch "1" gehalten |
| 4 | DTR | Data Terminal Ready | Ausgang | CP ist betriebsbereit |
| 5 | GND | Signal Ground | - | GND (GND_ISO) Nullbezugspunkt |
| 6 | DSR | Data Set Ready | Eingang | <ul style="list-style-type: none"> ■ "ON": Kommunikationspartner ist eingeschaltet und betriebsbereit ■ "OFF": Kommunikationspartner ist nicht eingeschaltet, nicht betriebsbereit |
| 7 | RTS | Request to send | Ausgang | <ul style="list-style-type: none"> ■ RTS auf "ON": <ul style="list-style-type: none"> – CP sendebereit ■ RTS auf "OFF": <ul style="list-style-type: none"> – CP sendet nicht |
| 8 | CTS | Clear to send | Eingang | Kommunikationspartner kann Daten vom CP empfangen, der CP erwartet dies als Antwort auf RTS gleich "ON". |
| 9 | RI | Ring indicator | Eingang | Klingelzeichen (Modem) |

RS232-Verkabelung ohne Hardware Handshake

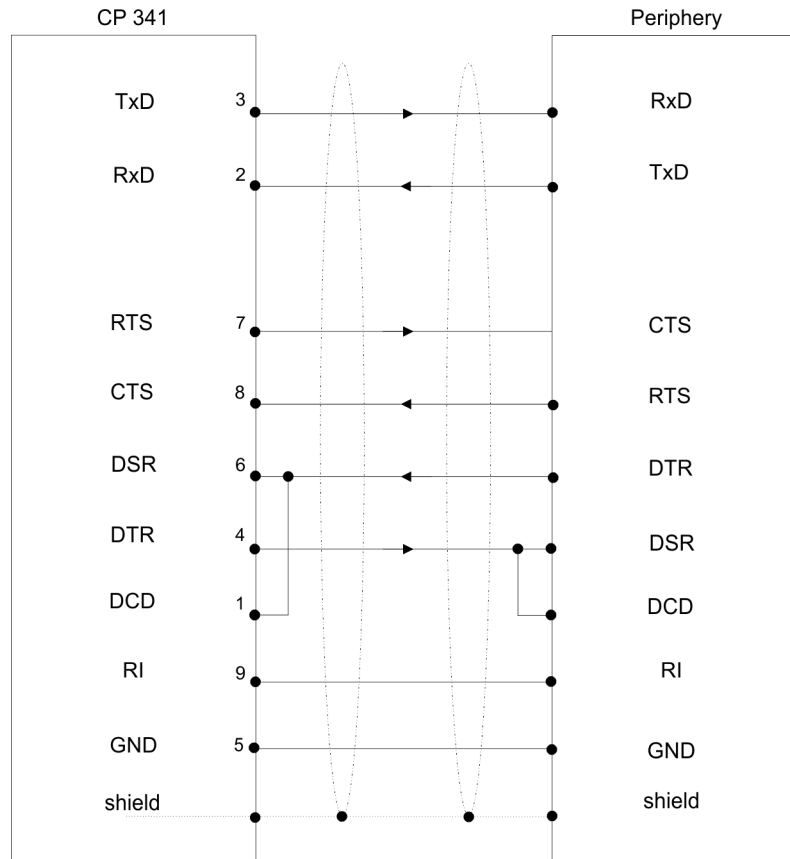


RS232-Verkabelung mit Hardware Handshake



- Verbinden Sie niemals Kabelschirm und GND miteinander, da die Schnittstellen zerstört werden könnten.
- GND muss in jedem Fall auf beiden Seiten verbunden werden, weil sonst ebenfalls eine Zerstörung der Schnittstellen-Module möglich ist.

RS232-Nullmodemkabel







Spannungsversorgung

Der CP 341-1AH01 bezieht seine Spannungsversorgung über den Rückwandbus.
 ↪ Kap. 4.3 "Technische Daten" Seite 26

LEDs

Der CP 341-1AH01 besitzt verschiedene LEDs, die der Busdiagnose dienen und den eigenen Betriebszustand anzeigen. Abhängig von der Betriebsart geben diese nach folgendem Schema Auskunft über den Betriebszustand des CP:

| Name | Farbe | Bedeutung |
|------|--|--|
| PWR |  grün | Signalisiert eine anliegende Betriebsspannung |
| SF |  rot | Sammelfehleranzeige oder Neuparametrierung <ul style="list-style-type: none">■ Sammelfehleranzeige leuchtet bei:<ul style="list-style-type: none">– Hardwarefehler– Firmwarefehler– Parametrierfehler– BREAK (Empfangsleitung zwischen CP und Kommunikationspartner unterbrochen) |
| TxD |  grün | ■ Daten senden (Transmit data) blinkt, wenn der CP Nutzdaten über die Schnittstelle sendet |
| RxD |  grün | ■ Daten empfangen (Receive data) blinkt, wenn der CP über die Schnittstelle Nutzdaten empfängt |

**Firmwareupdate**

- Während eines Firmwareupdates leuchten auf dem CP die LEDs SF, TxD und RxD.
- Der Firmwareupdate ist abgeschlossen, sobald TxD und RxD erlöschen.

4.3 Technische Daten

| Artikelnr. | 341-1AH01 |
|--|----------------------------------|
| Bezeichnung | CP 341 - Kommunikationsprozessor |
| SPEED-Bus | - |
| Stromaufnahme/Verlustleistung | |
| Stromaufnahme aus Rückwandbus | 160 mA |
| Verlustleistung | 0,8 W |
| Status, Alarm, Diagnosen | |
| Statusanzeige | ja |
| Alarmer | nein |
| Prozessalarm | nein |
| Diagnosealarm | ja, parametrierbar |
| Diagnosefunktion | nein |
| Diagnoseinformation auslesbar | möglich |
| Versorgungsspannungsanzeige | ja |
| Sammelfehleranzeige | rote SF-LED |
| Kanalfehleranzeige | keine |
| Funktionalität Sub-D Schnittstellen | |
| Bezeichnung | X2 |
| Physik | RS232 |
| Anschluss | 9poliger SubD Stecker |
| Potenzialgetrennt | ✓ |
| MPI | - |
| MP ² I (MPI/RS232) | - |
| Punkt-zu-Punkt-Kopplung | ✓ |
| 5V DC Spannungsversorgung | - |
| 24V DC Spannungsversorgung | - |
| Funktionalität Sub-D Schnittstellen | |
| Bezeichnung | - |
| Physik | - |
| Anschluss | - |
| Potenzialgetrennt | - |
| MPI | - |
| MP ² I (MPI/RS232) | - |
| Punkt-zu-Punkt-Kopplung | - |
| 5V DC Spannungsversorgung | - |
| 24V DC Spannungsversorgung | - |

| Artikelnr. | 341-1AH01 |
|-------------------------------------|--------------------------|
| Point-to-Point Kommunikation | |
| PtP-Kommunikation | ✓ |
| Schnittstelle potentialgetrennt | ✓ |
| Schnittstelle RS232 | ✓ |
| Schnittstelle RS422 | - |
| Schnittstelle RS485 | - |
| Anschluss | 9poliger SubD Stecker |
| Übertragungsgeschwindigkeit, min. | - |
| Übertragungsgeschwindigkeit, max. | 76,8 kbit/s |
| Leitungslänge, max. | 15 m |
| Point-to-Point Protokolle | |
| Protokoll ASCII | ✓ |
| Protokoll STX/ETX | ✓ |
| Protokoll 3964(R) | ✓ |
| Protokoll RK512 | - |
| Protokoll USS Master | - |
| Protokoll Modbus Master | ✓ |
| Protokoll Modbus Slave | ✓ |
| Spezielle Protokolle | - |
| Datengrößen | |
| Eingangsbytes | 16 |
| Ausgangsbytes | 16 |
| Parameterbytes | (16 + 106) |
| Diagnosebytes | 4 |
| Gehäuse | |
| Material | PPE |
| Befestigung | Profilschiene System 300 |
| Mechanische Daten | |
| Abmessungen (BxHxT) | 40 mm x 125 mm x 120 mm |
| Gewicht Netto | 170 g |
| Gewicht inklusive Zubehör | - |
| Gewicht Brutto | - |
| Umgebungsbedingungen | |
| Betriebstemperatur | 0 °C bis 60 °C |
| Lagertemperatur | -25 °C bis 70 °C |
| Zertifizierungen | |

Technische Daten

| | |
|------------------------|------------------|
| Artikelnr. | 341-1AH01 |
| Zertifizierung nach UL | ja |
| Zertifizierung nach KC | ja |

5 Einsatz

5.1 Schnelleinstieg

Übersicht

Die Einbindung des CP in Ihr SPS-System sollte nach folgender Vorgehensweise erfolgen:

1. ➤ Montage und Inbetriebnahme
2. ➤ Hardwarekonfiguration (Einbindung CP in CPU)
3. ➤ Protokollparameter über Parametrier-Plugin
4. ➤ Kommunikation mit dem Anwenderprogramm

Montage und Inbetriebnahme

1. ➤ Bauen Sie Ihr System 300 mit einer CPU 31x und dem CP 341 auf.
2. ➤ Verdrahten Sie das System, indem Sie die Leitungen für Spannungsversorgung, Signale und serielle Kommunikation anschließen.

Eine detaillierte Beschreibung zu diesem Thema finden Sie

↳ *Kap. 3 "Montage und Aufbaurichtlinien" Seite 13*

3. ➤ Schalten Sie die Spannungsversorgung ein.
⇒ Nach kurzer Hochlaufzeit befindet sich der CP ohne Protokoll im System.
4. ➤ Starten Sie den Siemens SIMATIC Manager und gehen Sie mit der CPU online. Näheres hierzu finden Sie im Handbuch zu Ihrer CPU.

Hardwarekonfiguration

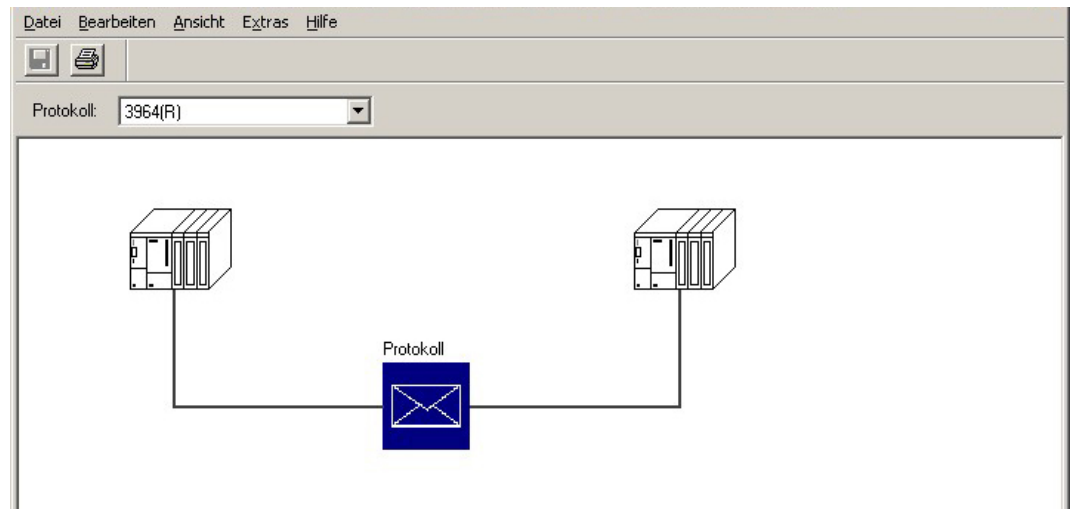
1. ➤ Zur Hardware-Konfiguration wechseln Sie im Siemens SIMATIC Manager in Ihrem Projekt in den Hardware-Konfigurator.
2. ➤ Platzieren Sie eine Profilschiene mit der entsprechenden CPU und den zugehörigen Modulen.
3. ➤ Ziehen Sie zur Projektierung des CP 341-1AH01 von Yaskawa den entsprechenden CP 341 von Siemens mit der Best.-Nr. 6ES7 341-1AH01-0AE0 auf den zugehörigen Steckplatz.
4. ➤ Stellen Sie über den CP-Eigenschaften-Dialog die Adresse und mittels des Parameter-Plugin "Punkt-zu-Punkt-Kopplung parametrieren" das Übertragungs-Protokoll und die protokollspezifischen Parameter ein.




Bitte beachten Sie, dass die Adresse für Ein- und Ausgänge identisch ist. Über diese Adresse haben Sie Zugriff auf den CP über Ihr Anwenderprogramm.

Protokollparameter

Für die Parametrierung der Protokoll-Parameter ist das Parametrier-Plugin "Punkt-zu-Punkt-Kopplung parametrieren" erforderlich.



Dieses Plugin können Sie von Siemens beziehen.

1. ➤ Über die Schaltfläche [Parameter...] im Eigenschaften-Dialog des CP starten Sie das Parametrier-Plugin "Punkt-zu-Punkt-Kopplung parametrieren".
2. ➤ Stellen Sie unter "Protokoll" das gewünschte Protokoll ein.
3. ➤ Klicken Sie zur Protokoll-Parametrierung auf  und stellen Sie die gewünschten Protokoll-Parameter ein.
4. ➤ Nachdem Sie alle protokollspezifischen Parameter geändert haben, sind die Parameter zu speichern.
5. ➤ Kehren Sie in den Eigenschaften-Dialog zurück.
 - ⇒ Speichern und übersetzen Sie Ihr Projekt.

Ladbare Protokolltreiber

Sie haben auch die Möglichkeit mittels ladbarer Protokolltreiber die Auswahl der Protokolle des Parametrier-Plugins zu erweitern. Näheres hierzu finden Sie bei dem entsprechenden Protokoll beschrieben.

Kommunikation mit dem Anwenderprogramm

- Bei den Standard-Protokollen erfolgt die Kommunikation mittels der Hantierungsbausteine FB 7 und FB 8, die zusammen mit dem Parametrier-Plugin installiert werden.
- Durch zyklischen Aufruf dieser Bausteine können Sie mit dem CP zyklisch Daten senden und empfangen. Auf dem CP erfolgt die Umsetzung der Übertragungsprotokolle zum Kommunikationspartner.
- Für diese FBs ist jeweils ein Instanz-DB erforderlich. Dieser ist beim Aufruf des entsprechenden FB mit anzugeben. Die Daten für die Kommunikation sind jeweils in einem Sende- bzw. Empfangs-DB abzulegen.
- Für die Steuerung der Kommunikation besitzen die FBs Steuerbits. Hiermit können Sie mit der entsprechenden Programmierung für den jeweiligen CP die Kommunikation starten, anhalten oder rücksetzen. Für die Fehler-Auswertung stellen die FBs Statusbits zur Verfügung.



Bitte beachten Sie, dass bei dem ladbaren Protokoll Modbus Slave für die Kommunikation der FB 80 - MODB_341 zum Einsatz kommt. Innerhalb diesem werden FB 7 und FB 8 aufgerufen.

Zugriff RS232-Begleitsignale

Bei Einsatz des ASCII-Treibers haben Sie die Möglichkeit mit dem FC 5 und FC 6 auf die RS232-Begleitsignale zuzugreifen. ↪ *Kap. 5.4 "Zugriff auf RS232-Begleitsignale" Seite 36*

5.2 Hardware-Konfiguration

Übersicht

- Die hier gemachten Angaben beziehen sich auf Module, die sich zusammen mit der CPU am gleichen Bus befinden. Damit die gesteckten Module gezielt angesprochen werden können, müssen ihnen bestimmte Adressen in der CPU zugeordnet werden. Die Adresszuordnung und die Parametrierung der direkt gesteckten Module erfolgt im Siemens SIMATIC Manager.
- Navigieren Sie hierzu im Hardware-Katalog zum gewünschten CP und platzieren Sie diesen in Ihrer S7-300 Station.

Vorgehensweise

1. ➤ Starten Sie den Siemens SIMATIC Manager.
2. ➤ Wechseln Sie in den Hardware-Konfigurator.
3. ➤ Platzieren Sie eine Profilschiene, indem Sie diese aus dem Hardware-Katalog in Ihr Projektfenster ziehen.
4. ➤ Projektieren Sie Ihre CPU und die entsprechenden Module.
Ziehen Sie hierzu die gewünschten Module aus dem Hardware-Katalog auf den zugehörigen Steckplatz der Profilschiene.
5. ➤ Ziehen Sie zur Projektierung des Yaskawa CP 341-1AH01 den entsprechenden CP 341 von Siemens mit der Best.-Nr. 6ES7 341-1AH01-0AE0 auf den zugehörigen Steckplatz.
6. ➤ Stellen Sie über die CP-"Eigenschaften" Ihr Übertragungsprotokoll und die protokollspezifischen Parameter ein (siehe "Protokollparameter"). Notieren Sie sich die "Adresse" ab welcher der CP eingebunden wird. Dieser Wert ist für die Einbindung in Ihrem Anwenderprogramm erforderlich. ↪ *Kap. 5.3 "Kommunikation mit dem Anwenderprogramm" Seite 34*
7. ➤ Speichern und übersetzen Sie Ihr Projekt und übertragen Sie Ihr Projekt in Ihre CPU.

5.2.1 Eigenschaften

CP 341-1AH01

Zum Aufruf der Eigenschaften doppelklicken Sie in Ihrem Projekt im Hardware-Konfigurator auf Ihren CP. Über die nachfolgend beschriebenen Register können Sie die Parameter des CP 341 von Yaskawa entsprechend einstellen. Zur Parametrierung ist das Parametrier-Plugin "Punkt-zu-Punkt-Kopplung parametrieren" erforderlich. Das Plugin können Sie von Siemens beziehen. Zur Installation starten Sie dieses und folgen Sie den Anweisungen.

Allgemein

- Kurzbezeichnung
 - Die Kurzbezeichnung mit der Information darunter sind identisch zu den Angaben im Fenster "Hardware-Katalog".
- Bestell-Nr.
 - Hier sehen Sie die Bestellnummer des Siemens CP 341. Bitte verwenden Sie für die Projektierung des CP 341-1AH01 von Yaskawa den Siemens-CP mit der Bestell-Nr. 6ES7 341-1AH01-0AE0.
- Name
 - Hier steht die Kurzbezeichnung des CP, die Sie nach Ihren Vorgaben ändern können. Wenn Sie die Bezeichnung ändern, erscheint die neue Bezeichnung in Ihrem Projekt in der Konfigurationstabelle.
- Kommentar
 - Hier können Sie den Einsatzzweck der Baugruppe eingeben.

Adressen

- Eingänge / Ausgänge
 - Durch Vorgabe einer Anfangs-Adresse für den Ein- bzw. Ausgabebereich können Sie den Beginn des Adressbereichs bestimmen, ab dem der CP im Adressbereich der CPU eingebunden wird. Hierbei belegt der CP für Ein- und Ausgabedaten jeweils 16Byte. Dieser Wert ist zur Einbindung in Ihrem Anwenderprogramm erforderlich. Bitte beachten Sie beim CP, dass die Basis-Adressen für Ein- und Ausgabe identisch sind.
- Prozessabbild
 - Das *Prozessabbild* bietet die Möglichkeit während der zyklischen Programmbearbeitung auf ein konsistentes Abbild des Prozesssignals zugreifen zu können.
 - Wenn im Feld *Prozessabbild* der Eintrag "---" sichtbar ist, so bedeutet dies, dass der angegebene Adressbereich außerhalb des Prozessabbilds liegt. Sobald sich der Eintrag innerhalb des Prozessabbilds befindet, wird dies mit dem Eintrag "OB1-PA" angezeigt.

Grundparameter

- Alarmgenerierung / Reaktion auf CPU-STOP
 - Hier können Sie das Alarmverhalten der Baugruppe einstellen. Wenn Sie hier "JA" anwählen, wird der Diagnose-Alarm freigegeben.

Parameter...

Über diese Schaltfläche können Sie das Plugin zur Parametrierung von Punkt-zu-Punkt-Verbindungen öffnen.

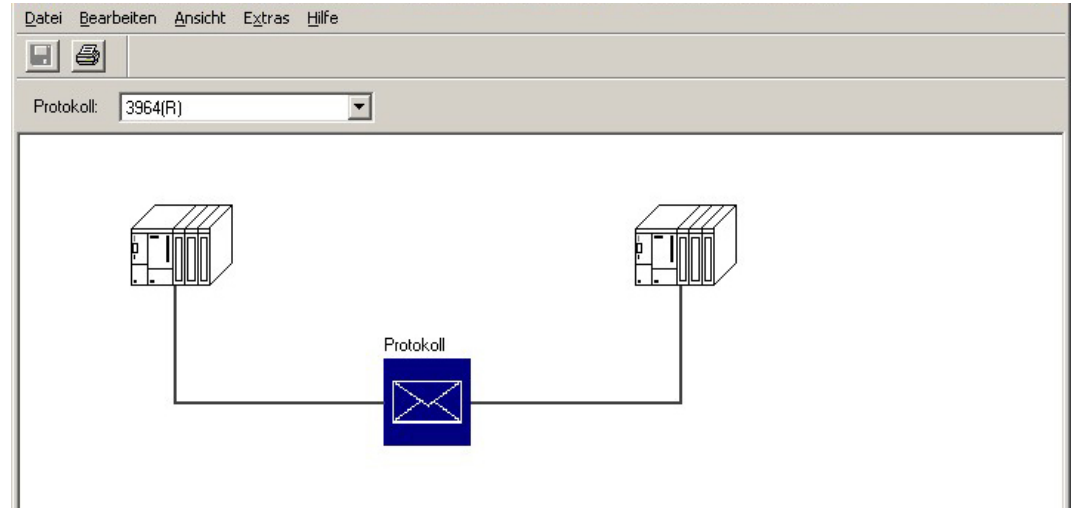



Bitte beachten Sie, dass Sie hierzu das Parametrier-Plugin "Punkt-zu-Punkt-Kopplung parametrieren" installieren müssen. Das Plugin können Sie von Siemens beziehen.


Nachfolgend ist die grundsätzliche Vorgehensweisen beim Einsatz beschrieben. Nähere Informationen zur Installation und zum Einsatz entnehmen Sie bitte der zugehörigen Dokumentation von Siemens.

Vorgehensweise

1. Starten Sie nach der Installation über die Schaltfläche [Parameter...] im Eigenschaften-Dialog des CP das Parametrier-Plugin "Punkt-zu-Punkt- Kopplung parametrieren".
2. Stellen Sie unter "Protokoll das gewünschte Protokoll ein. Je nach ausgewähltem Protokoll bietet das Parametrier-Plugin die Möglichkeit die Parameter für Datenempfang und Schnittstelle einzustellen.




3.  Bitte beachten Sie, dass solange das Plugin geöffnet ist, der Eigenschaften-Dialog des CP gesperrt ist.

4. Klicken Sie zur Protokoll-Parametrierung auf  und stellen Sie die gewünschten Protokoll-Parameter ein. Eine nähere Beschreibung der Protokolle finden Sie unter: [☞ Kap. 6 "Kommunikationsprotokolle" Seite 41](#)
5. Nachdem sie alle protokollspezifischen Parameter geändert haben, sind die Parameter zu speichern.

Ladbare Protokolltreiber

Sie haben auch die Möglichkeit mittels ladbarer Protokolltreiber die Auswahl der Protokolle des Parametrier-Plugins zu erweitern. Näheres hierzu finden Sie bei dem entsprechenden Protokoll beschrieben. [☞ Kap. 6 "Kommunikationsprotokolle" Seite 41](#)

Speichern

1. Nachdem Sie alle protokollspezifischen Parameter geändert haben, sind die Parameter mit "Datei → Speichern" bzw. über  zu speichern.
 - ⇒ Die Parameter werden nur dann in Ihr Projekt übernommen, wenn Sie diese zuvor speichern.
2. Mit "Datei → Beenden" wird das Plugin geschlossen und der CP Eigenschaften-Dialog wieder freigegeben. Speichern Sie die von Ihnen erstellte Konfiguration mit "Station → Speichern und übersetzen" in Ihrem Projekt.
3. Übertragen Sie die Konfiguration in Ihre CPU.

5.3 Kommunikation mit dem Anwenderprogramm

Übersicht

Für die Verarbeitung der Verbindungsaufträge auf SPS-Seite ist ein Anwenderprogramm in der CPU erforderlich. Hierbei kommen zur Kommunikation zwischen CPU, CP und einem Kommunikationspartner folgende Yaskawa-spezifischen Bausteine zum Einsatz:

| Baustein | Symbol | Kommentar |
|----------|----------|---|
| FB 7 | P_RCV_RK | Baustein für den Empfang von Daten von einem Kommunikationspartner. |
| FB 8 | P_SND_RK | Baustein für das Senden von Daten an einen Kommunikationspartner. |



Bitte beachten Sie, dass diese Bausteine intern den FC bzw. SFC 192 CP_S_R aufrufen. Dieser darf nicht überschrieben werden! Der direkte Aufruf eines internen Bausteins führt zu Fehler im entsprechenden Instanz-DB!



VORSICHT!

- Ein Aufruf dieser Bausteine im Prozess- oder Diagnosealarm ist nicht zulässig.
- Bitte beachten Sie, dass die FBs keine Parameterprüfung besitzen, d.h. bei falscher Parametrierung kann die CPU in den Zustand STOP verzweigen.



Näheres zum Einsatz dieser Bausteine finden Sie im Handbuch "SPEED7 Operationsliste" von Yaskawa.

FB 80 - MODB_341 bei Modbus Slave Protokoll

Bei Einsatz des Modbus Slave Protokolls kommt der Kommunikations-FB 80 - MODB_341 zum Einsatz. Innerhalb des FB 80 werden die FB 7 und FB 8 aufgerufen. Näheres zur Installation und zum Einsatz des FB 80 finden Sie im Teil "Kommunikationsprotokolle" unter Modbus Slave. ↪ *Kap. 6.4.7.1 "Daten senden FB 80 - MODB_341" Seite 78*

Installation

Die Funktionsbausteine sind zusammen mit dem Plugin "Punkt-zu-Punkt- Kopplung parametrieren" von Siemens online erhältlich.

- Die Installation der Funktionsbausteine erfolgt zusammen mit dem Plugin.
- Starten Sie das Installationsprogramm und folgen Sie den Anweisungen.
- Nach der Installation befinden sich die Funktionsbausteine in der Baustein-Bibliothek.
- Die Bibliothek öffnen Sie im Siemens SIMATIC Manager über: "Datei → Öffnen → Bibliotheken" über "CP PtP".
- Die Bausteine finden Sie unter "Blocks" vom CP 341.
 - ⇒ Für den Einsatz eines Funktionsbausteins ist dieser in Ihr Projekt zu kopieren.

Datenkonsistenz

Die Datenkonsistenz ist durch die Blockgröße bei der Datenübertragung zwischen CPU und CP auf 32Byte begrenzt. Für die konsistente Datenübertragung von mehr als 32Byte müssen Sie folgendes beachten:

- FB 8 - P_SND_RK:
 - Greifen Sie auf den Sende-DB erst wieder zu, wenn die Daten komplett übertragen wurden (*DONE* = 1).
- FB 7 - P_RCV_RK:
 - Greifen Sie auf den Empfangs-DB erst wieder zu, wenn die Daten komplett empfangen wurden (*NDR* = 1). Sperren Sie den Empfangs-DB danach solange (*EN_R* = 0), bis Sie die Daten bearbeitet haben.

Kommunikationsprinzip

Durch zyklischen Aufruf von FB 7 und FB 8 können Sie mit dem CP zyklisch Daten senden und empfangen. Auf dem CP erfolgt die Umsetzung der Übertragungsprotokolle zum Kommunikationspartner, welche Sie mittels der Hardwarekonfiguration parametrieren können.



Näheres zum Einsatz dieser Bausteine finden Sie im Handbuch "SPEED7 Operationsliste" von Yaskawa.

5.4 Zugriff auf RS232-Begleitsignale

Übersicht

Für den Datentransfer mit dem ASCII-Treiber können Sie die gleichen Funktionen verwenden wie unter "Kommunikation mit dem Anwenderprogramm" beschrieben. Zusätzlich können Sie beim ASCII-Treiber bei Einsatz des CP 341-1AH01 auf die RS232-Begleitsignale zugreifen. Hierbei kommen folgende Funktionsbausteine zum Einsatz:

| Baustein | Symbol | Kommentar | Protokoll |
|----------|-----------------------|--|-----------|
| FC 5 | V24_STAT (V ≥ 2.0) | Die Funktion ermöglicht Ihnen, die Signalzustände an der RS232-Schnittstelle des CP zu lesen. | ASCII |
| FC 6 | V24_SET (V ≥ 2.0) | Die Funktion ermöglicht Ihnen, die Ausgänge der RS232-Schnittstelle des CP zu setzen / rücksetzen. | ASCII |



VORSICHT!

- Ein Aufruf dieser Bausteine im Prozess- oder Diagnosealarm ist nicht zulässig.
- Bitte beachten Sie, dass Sie für den FC 5 und FC 6 ausschließlich Versionen ≥ 2.0 verwenden dürfen, da es ansonsten zu Datenverfälschungen kommen kann.

RS232-Begleitsignale

Auf der RS232-Schnittstelle des CP sind folgende Begleitsignale vorhanden:

| Signal | Ein-/Ausgang | Zugriff | Beschreibung |
|--------|--------------|---|---|
| DCD | Eingang | FC 5 - V24_STAT | Data Carrier detect - Empfangssignalepegel |
| DTR | Ausgang | FC 5 - V24_STAT FC 6 - V24_SET ■ Datenfluss: keine, XON/XOFF, RTS/CTS | Data Terminal ready - CP ist betriebsbereit |
| DSR | Eingang | FC 5 - V24_STAT | Data set ready - Kommunikationspartner ist betriebsbereit |
| RTS | Ausgang | FC 5 - V24_STAT FC 6 - V24_SET ■ Datenfluss: keine, XON/XOFF | Request to send - CP ist sendebereit |
| CTS | Eingang | FC 5 - V24_STAT FC 6 - V24_SET ■ Datenfluss: keine, XON/XOFF | Clear to send - Kommunikationspartner kann Daten vom CP empfangen |
| RI | Eingang | FC 5 - V24_STAT | Ring Indicator - Rufzeichen |

Signalzustand nach dem Einschalten

Nach dem Einschalten des CP sind alle Ausgangssignale inaktiv. Sie haben die Möglichkeit mittels der Parametrieroberfläche "Punkt-zu-Punkt- Kopplung parametrieren" bzw. über die nachfolgend aufgeführten FCs die Begleitsignale zu steuern. Lesender Zugriff auf die Begleitsignale mit dem FC 5 ist jederzeit möglich.

Prinzip der automatischen Bedienung der Begleitsignale

Die automatische Bedienung der RS232-Begleitsignale ist nur im Halbduplex-Betrieb möglich. Sobald Sie im Parametrierungstool unter "Übertragung" den Parameter "Automat. Bedienung der V24-Signale" aktivieren, ist ein schreibender Zugriff mit dem FC 6 nicht möglich.

- Die automatische Bedienung der RS232-Begleitsignale erfolgt nach folgendem Prinzip:
 - Sobald der CP über die Parametrierung auf "Automat. Bedienung der V24-Signale" eingestellt wurde, setzt der CP die Leitungen RTS auf OFF und DTR auf ON und zeigt somit seine Betriebsbereitschaft an.
 - In diesem Zustand können Telegramme gesendet und empfangen werden.

Senden

1. ➤ Steht ein Sendeauftrag an, wird RTS auf ON gesetzt und die parametrierte *Datenausgabewartezeit* gestartet.
 - ⇒ Nach Ablauf der *Datenausgabewartezeit* und CTS = ON werden die Daten über die RS232-Schnittstelle gesendet.
2. ➤ Wird beim Senden innerhalb der *Datenausgabewartezeit* die Leitung CTS nicht auf ON gesetzt bzw. erfolgt innerhalb des Sendevorgangs ein Wechsel von CTS auf OFF.
 - ⇒ Der Sendeauftrag wird abgebrochen und eine entsprechende Fehlermeldung generiert.
3. ➤ Nach dem Senden der Daten wird nach Ablauf der parametrierten RTS Wegnahmezeit die Leitung RTS auf OFF gesetzt. Es erfolgt kein Warten auf den Wechsel von CTS auf OFF.

Empfangen

1. ➤ Sobald die Leitung DSR auf ON gesetzt ist, ist Empfang von Daten über die RS232-Schnittstelle möglich.
 - ⇒ Bei Überlauf des Empfangspuffers erfolgt keine Reaktion des CP.
2. ➤ Bei einem Wechsel von DSR von ON nach OFF wird sowohl ein laufender Sendeauftrag als auch das Empfangen von Daten mit einer Fehlermeldung abgebrochen.
 - ⇒ Hierbei wird im Diagnosepuffer des CP die Meldung "DSR = OFF automat. Bedienung d. V24-Signale" eingetragen.

Datenausgabewartezeit / RTS-Wegnahmezeit

Die Datenausgabewartezeit ist so zu dimensionieren, dass der Kommunikationspartner in Empfangsbereitschaft gehen kann, bevor die Zeit abläuft. Stellen Sie die RTS-Wegnahmezeit so ein, dass der Kommunikationspartner die letzten Zeichen des Telegramms vollständig empfangen kann, bevor RTS (Sendewunsch) wieder deaktiviert wird.

FC 5 / FC 6



Näheres zum Einsatz dieser Bausteine finden Sie im Handbuch "SPEED7 Operationsliste" von Yaskawa.

5.5 Firmwareupdate

Übersicht

- Zur Funktionserweiterung und Fehlerbehebung haben Sie die Möglichkeit ein Firmwareupdate von Yaskawa in den Betriebssystemspeicher des CP zu laden. Ein Firmwareupdate erfolgt mittels der CPU über das Siemens Parametrier tool "Punkt-zu-Punkt-Kopplung parametrieren".
- Bei Einsatz einer Yaskawa SPEED7 CPU aus dem System 300S können Sie ab CPU-Firmware-Stand V 3.4.0 auch mittels einer entsprechend vorbereiteten MMC über die CPU ein Firmwareupdate durchführen.


5.5.1 Firmwareupdate über Siemens Parametrier tool

Bei Einsatz des Siemens Parametrier tools müssen folgende Voraussetzungen erfüllt sein:

- Siemens STEP®7 ab V 4.02 ist installiert.
- Parametrier tool "Punkt-zu-Punkt-Kopplung parametrieren" ab V 5.0 ist installiert.
- Der CP muss in der CPU in einem gültigen Projekt konfiguriert sein.
- Die CPU muss online mit Ihrem Projektier-PC verbunden sein.

Vorgehensweise

1. ➔ Schalten Sie die CPU in den STOP-Zustand.
2. ➔ Starten Sie das Parametrier tool "Punkt-zu-Punkt-Kopplung parametrieren". Doppelklicken Sie hierzu auf den entsprechenden CP und klicken Sie im "Eigenschaften"-Dialog auf die Schaltfläche [Parameter...].
3. ➔ Öffnen Sie den Dialog für das Firmwareupdate mit "Extras ➔ Firmware" Update.
 - ⇒ Sobald der CP erreichbar ist, wird der aktuelle Stand der CP Firmware unter "aktueller BG-FW-Stand" angezeigt. Kann keine Firmware-Version ermittelt werden (CP ist offline), wird "-----" angezeigt.
4. ➔ Wählen Sie über die Schaltfläche [Datei suchen...] die zu ladende Firmware aus. Die aktuellste CP-Firmware finden Sie im Service- Bereich unter www.yaskawa.eu.com.
5. ➔

 *Bitte beachten Sie, dass die Firmware aus 3 Dateien besteht. Wählen Sie hier die Datei HEADER.UPD.*

 - ⇒ Sie sehen unter "ausgewählter FW-Stand" die Version der ausgewählten Firmware-Datei.
6. ➔ Starten Sie mit [Firmware laden] den Ladevorgang auf den CP.

Mit der Bestätigung des Ladevorgangs wird die ausgewählte Firmware in den CP geladen. [Abbruch] führt zum sofortigen Ladeabbruch.

Das Laden wird durch Leuchten der LEDs SF, TxD und RxD angezeigt. Bevor eine schon bestehende Firmware überschrieben wird, überprüft der CP, ob es sich um eine gültige Firmware von Yaskawa für den CP handelt.

 - ⇒ Nach Abschluss des Ladevorgangs erlöschen die LEDs TxD und RxD.
7. ➔ Zur Aktivierung der neuen Firmware müssen Sie jetzt Power OFF/ON durchführen.

Transferanzeige

- Während des Firmwaretransfers wird der Fortschritt unter "Erledigt" in Form eines Balkens und in % angezeigt.
 - Auf dem entsprechenden CP leuchten die LEDs SF, TxD und RxD.

5.5.2 Firmwareupdate bei Einsatz einer SPEED7 CPU

- Sie haben die Möglichkeit unter Einsatz einer MMC für den CP ein Firmwareupdate durchzuführen. Diese Funktionalität ist ab der CPU Firmware-Version V 3.4.0 möglich. Hierzu muss sich in der SPEED7 CPU beim Hochlauf eine entsprechend vorbereitete MMC befinden.
- Damit eine Firmwaredatei beim Hochlauf erkannt und zugeordnet werden kann, ist für jede update-fähige Komponente und jeden Hardware-Ausgabestand ein pkg-Dateiname reserviert, der mit "px" beginnt und sich in einer 6-stelligen Ziffer unterscheidet. Bei jedem updatefähigen Modul finden Sie den pkg-Dateinamen unter der Frontklappe auf einem Aufkleber auf der rechten Seite des Moduls.
- Sobald sich beim Hochlauf eine entsprechende pkg-Datei auf der MMC befindet, werden alle der pkg-Datei zugeordneten Komponenten auf dem Rückwandbus und in der CPU mit der entsprechenden Firmware beschrieben.
- Die 2 aktuellsten Firmwarestände finden Sie auf www.yaskawa.eu.com im Service-Bereich.



VORSICHT!

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

Firmwarestand über Web-Seite der SPEED7 CPU ausgeben

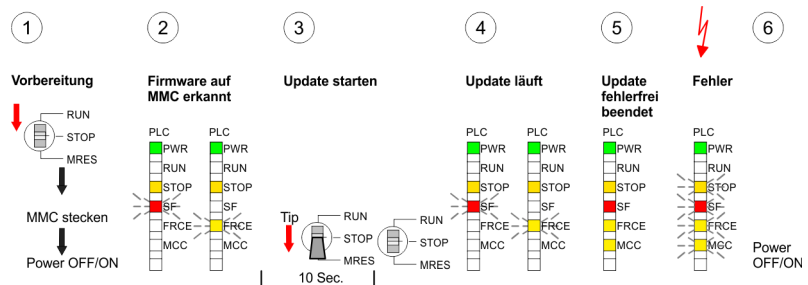
1. Die SPEED7 CPU hat eine Web-Seite integriert, die auch Informationen zum Firmwarestand der angebenen Komponenten bereitstellt. Über den Ethernet-PG/OP-Kanal haben Sie Zugriff auf diese Web-Seite.
2. Zur Aktivierung des PG/OP-Kanals müssen Sie diesem IP-Parameter zuweisen. Dies kann im Siemens SIMATIC Manager entweder über eine Hardware-Konfiguration erfolgen, die Sie über MMC bzw. MPI einspielen oder über Ethernet durch Angabe der MAC-Adresse unter "*Zielsystem* → *Ethernet-Adresse*" vergeben.
 - ⇒ Danach können Sie mit einem Web-Browser über die angegebene IP Adresse auf den PG/OP-Kanal zugreifen. Näheres hierzu finden Sie im Handbuch zu Ihrer CPU unter "Zugriff auf den Webserver".

Firmware laden und auf MMC übertragen

1. Gehen Sie auf www.yaskawa.eu.com
2. Klicken Sie auf "*Service* → *Download* → *Firmware*"
3. Wählen Sie den entsprechenden CP aus und laden Sie die .zip-Datei Px000080.pkg auf Ihren PC.
4. Entpacken Sie die zip-Datei und kopieren Sie die extrahierte Datei auf Ihre MMC.
5. Übertragen Sie auf diese Weise alle erforderlichen Firmware-Dateien auf Ihre MMC.

Firmware von MMC in SPEED 7 CPU übertragen

1. ➤ Bringen Sie den RUN-STOP-Schalter Ihrer CPU in Stellung STOP.
Schalten Sie die Spannungsversorgung aus.
Stecken Sie die MMC mit den Firmware-Dateien in die CPU. Achten Sie hierbei auf die Steckrichtung der MMC.
Schalten Sie die Spannungsversorgung ein.
2. ➤ Nach einer kurzen Hochlaufzeit zeigt das abwechselnde Blinken der LEDs SF und FRCE an, dass auf der MMC mindestens eine Firmware-Datei gefunden wurde, die sich vom aktuellen Stand unterscheidet.
3. ➤ Sie starten die Übertragung der Firmware, sobald Sie innerhalb von 10s den RUN/STOP-Schalter kurz nach MRES tippen und dann den Schalter in der STOP-Position belassen.
4. ➤ Während des Update-Vorgangs blinken die LEDs SF und FRCE abwechselnd und die MCC-LED leuchtet. Dieser Vorgang kann mehrere Minuten dauern.
5. ➤ Das Update ist fehlerfrei beendet, wenn die LEDs PWR, STOP, SF, FRCE und MCC leuchten.
⇒ Blinken diese schnell, ist ein Fehler aufgetreten.
6. ➤ Schalten Sie die Spannungsversorgung aus und wieder ein.
Jetzt prüft die CPU, ob noch weitere Firmware-Updates durchzuführen sind. Ist dies der Fall, blinken, wiederum nach einer kurzen Hochlaufzeit, die LEDs SF und FRCE. Fahren Sie mit Punkt 3 fort.



- ⇒ Das Update ist fehlerfrei beendet, wenn die LEDs PWR, STOP, SF, FRCE und MCC leuchten.

5.5.3 CP-Firmwarestand anzeigen

Sie haben die Möglichkeit über den Baugruppenzustand des Siemens SIMATIC Managers den aktuellen Hardware- und Firmware-Ausgabestand, die Px-Nummer und die Version zu ermitteln.

1. ➤ Gehen Sie hierzu im Hardware-Konfigurator auf "Station → Online öffnen" und wählen Sie den entsprechenden CP.
2. ➤ Wie schon beschrieben, können Sie bei Einsatz einer SPEED7 CPU über die Webserver der CPU den aktuellen Firmwarestand ermitteln.

6 Kommunikationsprotokolle

6.1 Übersicht

Serielle Übertragung eines Zeichens

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

Zeichenrahmen

Beim Bidirektionalen Datenverkehr wird zwischen *Halbduplex*- und *Vollduplex*-Betrieb unterschieden.

- Im *Halbduplex*-Betrieb werden zu einem Zeitpunkt Daten entweder gesendet oder empfangen.
- Ein gleichzeitiger Datenaustausch kann nur im *Vollduplex*-Betrieb erfolgen.

Jedem zu übertragenden Zeichen geht ein Synchronisierimpuls als Startbit voraus. Das Ende des Zeichentransfers bildet das Stopbit. Neben Start- und Stopbit sind weitere parametrierbare Vereinbarungen zwischen den Kommunikationspartnern für eine serielle Datenübertragung erforderlich.

Dieser Zeichenrahmen besteht aus folgenden Elementen:

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

Protokolle

Der CP wickelt die serielle Datenübertragung selbständig ab. Hierzu ist der CP mit Treibern für folgende Protokolle ausgestattet:

- ASCII
- 3964(R)



Bitte beachten Sie, dass der CP 341-1AH01 von Yaskawa die Rechnerkopplung RK512 nicht unterstützt.

Zusätzlich werden folgende ladbare Protokoll-Treiber unterstützt:

- Modbus Master RTU
- Modbus Master ASCII
- Modbus Slave RTU

Auf den Folgeseiten sind alle unterstützten Protokolle beschrieben. ↪ *Kap. 6.2 "ASCII" Seite 42*



Bei Einsatz ladbarer Treiber werden aus softwaretechnischen Gründen die Treiber von Siemens in den CP übertragen, dort aber nicht installiert. Da im CP 341-1AH01 VIPA-eigene Treiber installiert sind, ist die Verwendung von Siemensüblichen Hardware-Dongles nicht erforderlich.

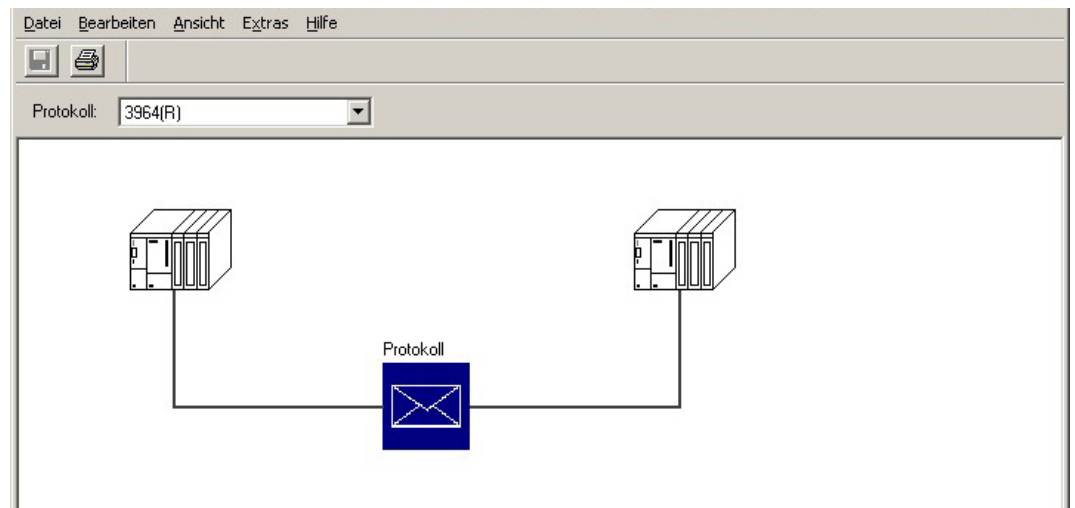
6.2 ASCII

Funktionsweise

- Die Datenkommunikation über ASCII ist eine einfache Form des Datenaustauschs und kann mit einer Multicast/Broadcast-Funktion verglichen werden.
- Die logische Trennung der Telegramme erfolgt über die Zeichenverzugszeit (ZVZ). Innerhalb dieser Zeit muss der Sender sein Telegramm an den Empfänger geschickt haben. Ein Telegramm wird erst dann an die CPU weitergereicht, wenn dieses vollständig empfangen wurde.
- Zusätzlich zur Zeichenverzugszeit haben Sie die Möglichkeit über die Parametrierung des ASCII-Treibers ein weiteres Endekriterium für den Empfang zu bestimmen.
- Da bei ASCII-Übertragung neben der Verwendung des Paritätsbit keine weiteren Maßnahmen zur Datensicherung erfolgen, ist der Datentransfer zwar sehr effizient aber nicht gesichert. Mit der Parität wird das Kippen eines Bits in einem Zeichen abgesichert. Kippen mehrere Bits eines Zeichens, kann dieser Fehler nicht mehr erkannt werden.

Vorgehensweise

1. Über die Schaltfläche [Parameter...] im Eigenschaften-Dialog des CP starten Sie das Parametrier-Plugin "Punkt-zu-Punkt-Kopplung parametrieren".
2. Hier können Sie die Parameter für Übertragungsprotokoll, Datenempfang und Schnittstelle einstellen.



3. Stellen Sie zuerst unter *Protokoll* das gewünschte Protokoll "ASCII" ein.
4. Zur Protokoll-Parametrierung klicken Sie auf .
 - ⇒ Nachfolgend sind diese Parameter beschrieben. Informationen hierzu finden Sie auch in der Online-Hilfe des Parametrier-Plugins.

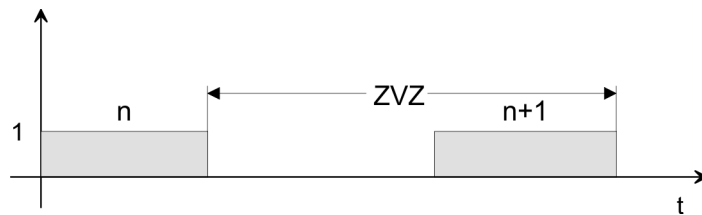
6.2.1 ASCII - Parameter

Hier können Sie die Parameter für den ASCII-Treiber einstellen. Unter ASCII sind die Parameter zum Zeichnungsrahmen und zur Baudrate auf allen Kommunikationspartnern gleich einzustellen.

Endekennung eines Telegramms

Bei der ASCII-Übertragung kann das Ende des Empfangstelegramms auf folgende Arten erkannt werden:

- nach Ablauf der Zeichenverzugszeit (Default-Wert)
- nach Empfang einer festen Zeichenanzahl
- nach Empfang der/des Endezeichens
 - Je nach Modus können Sie hier entsprechende Parameter vorgeben.



Die Zeichenverzugszeit (ZVZ) definiert den max. zulässigen zeitlichen Abstand zwischen zwei Zeichen.

| Parameter | Beschreibung | Defaultwert | |
|------------------------|---|-------------|--|
| Zeichenverzugszeit ZVZ | Die kleinste ZVZ ist abhängig von der Baudrate: | 4ms | |
| | Geschwindigkeit (Bit/s) | | ZVZ (ms) |
| | 300 | | 130 |
| | 600 | | 65 |
| | 1200 | | 32 |
| | 2400 | | 16 |
| | 4800 | | 8 |
| | 9600 | | 4 |
| | 19200, 57600, 76800 | | 2 |
| | | | Wertebereich: 2ms ... 65535ms in 1ms Schritten |
| Telegrammlänge | Für den Empfang mit fester Zeichenanzahl können Sie hier eine feste Telegrammlänge angeben. <ul style="list-style-type: none"> ■ Wertebereich: 1 ... 1024Bytes | 240 | |
| Sendepause... | Hier können Sie das Einhalten einer Sendepause zur Einsynchronisation deaktivieren. <ul style="list-style-type: none"> ■ Wertebereich: aktiviert, deaktiviert | aktiviert | |

Senden mit Endezeichen

Sofern Sie in der Endeerkennung "Endezeichen" aktiviert haben, können Sie hier Endezeichen definieren oder die am FB parametrisierte Länge vorgeben.

| Parameter | Beschreibung | Defaultwert |
|-----------------|--|---|
| Endezeichen 1/2 | Für die Kommunikation mit Endezeichen können Sie maximal 2 Endezeichen parametrieren. Mit einem Endezeichen begrenzen Sie die Länge des jeweiligen Telegramms. ■ Wertebereich: 0...7Fh/FFh (7/8 Datenbits) | Endezeichen 1:3 (03h = ETX) Endezeichen 2:0 |


Geschwindigkeit

Hier können Sie aus einer Auswahlliste die Geschwindigkeit für die Datenübertragung vorgeben.

| Parameter | Beschreibung | Defaultwert |
|-------------------|--|-------------|
| Baudrate in Bit/s | Übertragungsgeschwindigkeit in Bit/s ■ Wertebereich: 300, 600, 1200, 2400, 4800, 9600, 19200, 38400, 57600, 76800 | 9600 |

Zeichenrahmen

Die Daten zwischen den Kommunikationspartnern werden über die serielle Schnittstelle in einem Zeichenrahmen übertragen. Somit kann beim Empfänger jedes übertragene Zeichen erkannt und kontrolliert werden.



Bitte beachten Sie, dass die folgenden Parameter bei allen Kommunikationspartnern gleich einzustellen sind.

| Parameter | Beschreibung | Defaultwert |
|-----------|--|-------------|
| Datenbits | Anzahl der Bits, auf die ein Zeichen abgebildet wird. ■ Wertebereich: 7, 8 | 8 |
| Stopbits | Die Stopbits werden bei der Übertragung jedem zu übertragenden Zeichen nachgesetzt und kennzeichnen das Ende eines Zeichens. ■ Wertebereich: 1, 2 | 1 |
| Parität | Das Paritätsbit ergänzt durch seinen Wert "0" oder "1" die Summe aller Bits (Daten- und Paritätsbit) auf einen definierten Zustand. ■ Wertebereich: keine, ungerade, gerade | gerade |

ASCII Übertragung

Arbeitet ein Kommunikationspartner unter ASCII schneller als der andere, können Sie über die Datenflusskontrolle die Kommunikation der Teilnehmer synchronisieren. In diesem Register haben Sie die Möglichkeit die Art der Datenflusskontrolle und die zugehörigen Parameter einzustellen.

Datenflusskontrolle

| Parameter | Beschreibung | Defaultwert |
|---------------------|--|-------------|
| Datenflusskontrolle | <ul style="list-style-type: none"> Wertebereich: keine, XON/XOFF, RTS/CTS, Autom. Bedienung der V24-Signale | keine |

Datenflusskontroll-Parameter

| Parameter | Beschreibung | Defaultwert |
|---|---|-------------|
| XON-Zeichen | Bei "XON/XOFF" Code für Zeichen XON <ul style="list-style-type: none"> Wertebereich: 0...7Fh/FFh (7/8 Datenbits) | 11(DC1) |
| XOFF-Zeichen | Bei "XON/XOFF" Code für Zeichen XOFF <ul style="list-style-type: none"> Wertebereich: 0...7Fh/FFh (7/8 Datenbits) | 13(DC3) |
| Warten auf XON nach XOFF (Wartezeit auf CTS=ON) | Zeit, die der CP beim Senden auf das Zeichen CTS=ON des Partners warten soll. <ul style="list-style-type: none"> Wertebereich: 20 ... 65535ms in 10ms Schritten | 20 000ms |
| RTS-Wegnahmezeit | Bei "Autom. Bedienung der V24-Signale" Zeit, die nach dem Senden erwartet werden soll, bis der CP die Leitung RTS auf OFF setzt. <ul style="list-style-type: none"> Wertebereich: 0 ... 65535ms in 10ms Schritten | 10ms |
| Datenausgabewartezeit | Bei "Autom. Bedienung der V24-Signale" Zeit, die der CP beim Senden nach dem Setzen der Leitung RTS auf ON, auf CTS=ON des Partners warten soll, bevor der Sendevorgang gestartet wird. <ul style="list-style-type: none"> Wertebereich: 0 ... 65535ms in 10ms Schritten | 10ms |

ASCII Datenempfang

Der CP verwendet zur Pufferung von empfangenen Telegrammen einen Ringpuffer. Hierbei wird immer das jeweils älteste Telegramm vom CP an die CPU übertragen.

| Parameter | Beschreibung | Defaultwert |
|-------------------------------|--|-------------|
| Gepufferte Empfangstelegramme | Anzahl der Telegramme, die im CP-Empfangspuffer gepuffert werden sollen. <ul style="list-style-type: none"> Wertebereich: 1 ... 250 | 250 |
| Überschreiben verhindern | Wenn der Parameter "Gepufferte Empfangstelegramme" auf "1" eingestellt ist, können Sie diesen Parameter deaktivieren. Auf diese Weise wird immer ein aktuelles Telegramm an die CPU übertragen. <ul style="list-style-type: none"> Wertebereich: aktiviert, deaktiviert | aktiviert |

6.3 3964(R)

6.3.1 Grundlagen 3964(R)

Funktionsweise

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

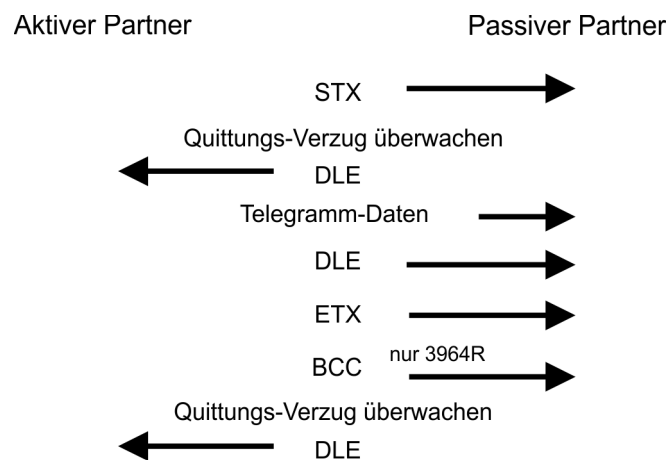
Folgende Steuerzeichen werden ausgewertet:

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



Wird ein DLE als Informationszeichen übertragen, so wird dieses zur Unterscheidung vom Steuerzeichen DLE beim Verbindungsauf- und -abbau auf der Sendeleitung doppelt gesendet (DLE-Verdoppelung). Der Empfänger macht die DLE-Verdoppelung wieder rückgängig. Unter 3964(R) muss dem Kommunikationspartner eine niedrigere Priorität zugeordnet sein. Wenn beide Kommunikationspartner gleichzeitig einen Sendeauftrag erteilen, dann stellt der Partner mit niedriger Priorität seinen Sendeauftrag zurück.

Ablauf



Sie können pro Telegramm maximal 250Byte übertragen.

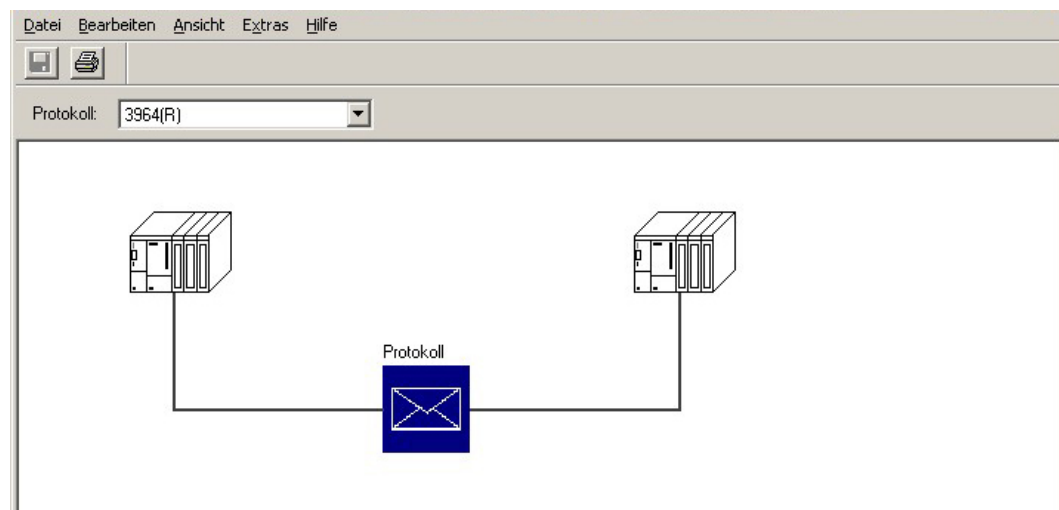
Time-out-Zeiten

QVZ wird überwacht zwischen STX und DLE sowie zwischen BCC und DLE. ZVZ wird während des gesamten Telegramm-Empfangs überwacht. Bei Verstreichen der QVZ nach STX wird erneut STX gesendet, nach 5 Versuchen wird ein NAK gesendet und der Verbindungsaufbau abgebrochen. Dasselbe geschieht, wenn nach einem STX ein NAK oder ein beliebiges Zeichen empfangen wird. Bei Verstreichen der QVZ nach dem Telegramm (nach BCC-Byte) oder bei Empfang eines Zeichens ungleich DLE werden der Verbindungsaufbau und das Telegramm wiederholt. Auch hier werden 5 Versuche unternommen, danach ein NAK gesendet und die Übertragung abgebrochen.

| | |
|---|--|
| Passivbetrieb | Wenn der Treiber auf den Verbindungsaufbau wartet und ein Zeichen ungleich STX empfängt, sendet er NAK. Bei Empfang eines Zeichens NAK sendet der Treiber keine Antwort. Wird beim Empfang die ZVZ überschritten, wird ein NAK gesendet und auf erneuten Verbindungsaufbau gewartet. Wenn der Treiber beim Empfang des STX noch nicht bereit ist, sendet er ein NAK. |
| Block-Check-Character (BCC-Byte) | Zur weiteren Datensicherung wird bei 3964R am Ende des Telegramms ein Block-Check-Charakter angehängt. Das BCC-Byte wird durch eine XOR-Verknüpfung über die Daten des gesamten Telegramms einschließlich DLE/ETX gebildet. Beim Empfang eines BCC-Bytes, das vom selbst ermittelten abweicht, wird anstatt des DLEs ein NAK gesendet. |
| Initialisierungskonflikt | Versuchen beide Partner gleichzeitig innerhalb der QVZ einen Verbindungsaufbau, so sendet der Partner mit der niedrigeren Priorität das DLE und geht auf Empfang. |
| Data Link Escape (DLE-Zeichen) | Das DLE-Zeichen in einem Telegramm wird vom Treiber verdoppelt, d.h. es wird DLE/DLE gesendet. Beim Empfang werden doppelte DLEs als ein DLE im Puffer abgelegt. Als Ende des Telegramms gilt immer die Kombination DLE/ETX/BCC (nur bei 3964R). Die Steuercodes: <ul style="list-style-type: none"> ■ 02h = STX ■ 03h = ETX ■ 10h = DLE ■ 15h = NAK |

6.3.2 Vorgehensweise

1. ➤ Über die Schaltfläche [Parameter...] im Eigenschaften-Dialog des CP starten Sie das Parametrier-Plugin "Punkt-zu-Punkt-Kopplung parametrieren".
2. ➤ Hier können Sie die Parameter für Übertragungsprotokoll, Datenempfang und Schnittstelle einstellen.



3. ➤ Stellen Sie zuerst unter *Protokoll* das gewünschte Protokoll "3964(R)" ein.
4. ➤ Zur Protokoll-Parametrierung klicken Sie auf .
 ⇒ Nachfolgend sind diese Parameter beschrieben. Informationen hierzu finden Sie auch in der Online-Hilfe des Parametrier-Plugins.

6.3.3 3964(R) - Parameter

Hier können Sie die Parameter für den 3964(R)-Protokoll-Treiber einstellen.



Bitte beachten Sie, dass Sie die Parameter zum Blockcheck, zur Baudrate und zum Zeichnungsrahmen mit Ausnahme der Priorität auf allen Kommunikationspartnern gleich einstellen.

Der CP unterstützt folgende Protokollvarianten:

- Standard-Werte ohne Blockcheck: 3964
- Standard-Werte mit Blockcheck: 3964R
- Parametrierbar ohne Blockcheck: 3964
- Parametrierbar mit Blockcheck: 3964R

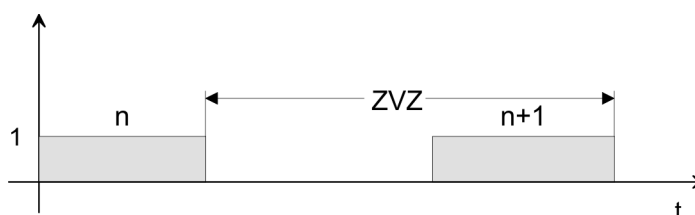
Protokoll

- Defaultmäßig ist "Standard-Werte mit Blockcheck" eingestellt:
 - Zeichenverzugszeit: 220ms
 - Quittungsverzugszeit: 2000ms
 - Aufbauversuche: 6
 - Übertragungsversuche: 6

| Parameter | Beschreibung | Defaultwert |
|-------------------------|---|-------------|
| mit Blockcheck | <p>Zur Erhöhung der Datensicherheit können Sie mit einem Blockprüfzeichen (BCC = Block Check Character) arbeiten.</p> <p>Erkennt der CP die Zeichenfolge DLE ETX BCC, beendet er den Empfang. Der CP vergleicht das empfangene Blockprüfzeichen BCC mit der intern gebildeten Längsparität.</p> <p>Ist das Blockprüfzeichen korrekt und kein anderer Empfangsfehler aufgetreten, sendet der CP das Zeichen DLE (bei Fehler wird das Zeichen NAK an den Kommunikationspartner gesendet).</p> <p>Erkennt der CP bei deaktiviertem Blockcheck die Zeichenfolge DLE ETX, beendet er den Empfang und sendet DLE für einen fehlerfrei (oder NAK für einen fehlerhaft) empfangenen Block an den Kommunikationspartner.</p> <ul style="list-style-type: none"> ■ Wertebereich: aktiviert, deaktiviert | aktiviert |
| Standardwerte verwenden | <p>Im aktivierten Zustand sind die Protokollparameter mit Defaultwerten belegt. Ist dieser Parameter deaktiviert, können Sie die Protokollparameter frei einstellen.</p> <ul style="list-style-type: none"> ■ Wertebereich: aktiviert, deaktiviert | aktiviert |

Protokoll-Parameter

Die ZVZ definiert den max. zulässigen zeitlichen Abstand zwischen zwei Zeichen innerhalb eines Telegramms.




| Parameter | Beschreibung | Defaultwert | |
|--|--|---|----------|
| Zeichenverzugszeit (ZVZ) | Bitte beachten Sie, dass die ZVZ in Abhängigkeit von der Übertragungsgeschwindigkeit auf einen bestimmten Mindestwert begrenzt ist: | 220ms | |
| | Geschwindigkeit (Bit/s) | | ZVZ (ms) |
| | 300 | | 60 |
| | 600 | | 40 |
| | 1200 | | 30 |
| | 2400 ... 76800 | | 20 |
| ■ Wertebereich: 20...65535ms in 10ms Schritten | | | |
| Quittungsverzugszeit (QVZ) | Hier geben Sie die Zeit an, die maximal beim Verbindungsauf- und -abbau bis zur Quittierung des Kommunikationspartners verstreichen darf. Bitte beachten Sie, dass die QVZ in Abhängigkeit von der Übertragungsgeschwindigkeit auf einen bestimmten Mindestwert begrenzt ist: | 2000ms (550ms bei 3964 ohne Block-check) | |
| | Geschwindigkeit (Bit/s) | | QVZ (ms) |
| | 300 | | 60 |
| | 600 | | 40 |
| | 1200 | | 30 |
| | 2400 ... 76800 | | 20 |
| ■ Wertebereich: 20...65535ms in 10ms Schritten | | | |
| Aufbauversuche | Über diesen Parameter definieren Sie die maximale Anzahl der Versuche des CP eine Verbindung aufzubauen. Überschreitet die Anzahl vergeblicher Versuche die maximale Anzahl, wird das Verfahren abgebrochen und der Fehler am STATUS-Ausgang des FBs angezeigt. ■ Wertebereich: 1...255 | 6 | |
| Übertragungsversuche | Mit diesem Parameter bestimmen Sie die maximale Anzahl der Versuche ein Telegramm zu übertragen. Überschreitet die Anzahl vergeblicher Versuche die maximale Anzahl, wird das Verfahren abgebrochen und der Fehler am STATUS-Ausgang des FBs angezeigt. ■ Wertebereich: 1...255 | 6 | |

Geschwindigkeit Hier können Sie aus einer Auswahlliste die Geschwindigkeit für die Datenübertragung vorgeben.

| Parameter | Beschreibung | Defaultwert |
|-------------------|--|-------------|
| Baudrate in Bit/s | Übertragungsgeschwindigkeit in Bit/s <ul style="list-style-type: none"> Wertebereich: 300, 600, 1200, 2400, 4800, 9600, 19200, 38400, 57600, 76800 | 9600 |

Zeichenrahmen Die Daten zwischen den Kommunikationspartnern werden über die serielle Schnittstelle in einem Zeichenrahmen übertragen. Somit kann beim Empfänger jedes übertragene Zeichen erkannt und kontrolliert werden.



Bitte beachten Sie, dass die folgenden Parameter bei allen Kommunikationspartnern gleich einzustellen sind.

| Parameter | Beschreibung | Defaultwert |
|-----------|---|-------------|
| Datenbits | Anzahl der Bits, auf die ein Zeichen abgebildet wird. <ul style="list-style-type: none"> Wertebereich: 7, 8 | 8 |
| Stopbits | Die Stopbits werden bei der Übertragung jedem zu übertragenden Zeichen nachgesetzt und kennzeichnen das Ende eines Zeichens. <ul style="list-style-type: none"> Wertebereich: 1, 2 | 1 |
| Parität | Das Paritätsbit ergänzt durch seinen Wert "0" oder "1" die Summe aller Bits (Daten- und Paritätsbit) auf einen definierten Zustand. <ul style="list-style-type: none"> Wertebereich: keine, ungerade, gerade | gerade |
| Priorität | Wenn beide Kommunikationspartner gleichzeitig einen Sendeauftrag erteilen, dann stellt der Partner mit der niedrigen Priorität seinen Sendeauftrag zurück. Für die Datenübertragung müssen Sie an einem Kommunikationspartner hohe und beim anderen niedrige Priorität einstellen. <ul style="list-style-type: none"> Wertebereich: hoch, niedrig | hoch |

3964(R) Datenempfang

- CP-Empfangspuffer im Anlauf löschen:
 - (Defaultwert: "CP-Empfangspuffer im Anlauf löschen" nicht aktiv).
- Diesen Parameter können sie nicht aktivieren.
- Der Empfangspuffer des CP 341-1AH01 wird beim CPU-Übergang von STOP nach RUN (CPU-Anlauf) nicht gelöscht.

6.4 Modbus

6.4.1 Grundlagen Modbus

Übersicht

Das Protokoll Modbus ist ein Kommunikationsprotokoll, das eine hierarchische Struktur mit einem Master und mehreren Slaves festlegt.

Physikalisch arbeitet Modbus über eine serielle Halbduplex-Verbindung als Punkt-zu-Punkt unter RS232 oder als Mehrpunkt-Verbindung unter RS485.

Master-Slave-Kommunikation

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

Telegramm-Aufbau

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

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

Broadcast mit Slave-Adresse = 0

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

Nur Schreibaufträge dürfen als Broadcast gesendet werden.

ASCII-, RTU Modus

Bei Modbus gibt es zwei unterschiedliche Übertragungsmodi:

- ASCII-Modus:
 - Jedes Byte wird im 2 Zeichen ASCII-Code übertragen. Die Daten werden durch Anfang- und Ende-Zeichen gekennzeichnet. Dies macht die Übertragung transparent, aber auch langsam.
- RTU-Modus:
 - Jedes Byte wird als ein Zeichen übertragen. Hierdurch erreichen Sie einen höheren Datendurchsatz als im ASCII-Modus. Anstelle von Anfang- und Ende-Zeichen wird eine Zeitüberwachung eingesetzt.

Die Modus-Wahl erfolgt bei der Parametrierung.

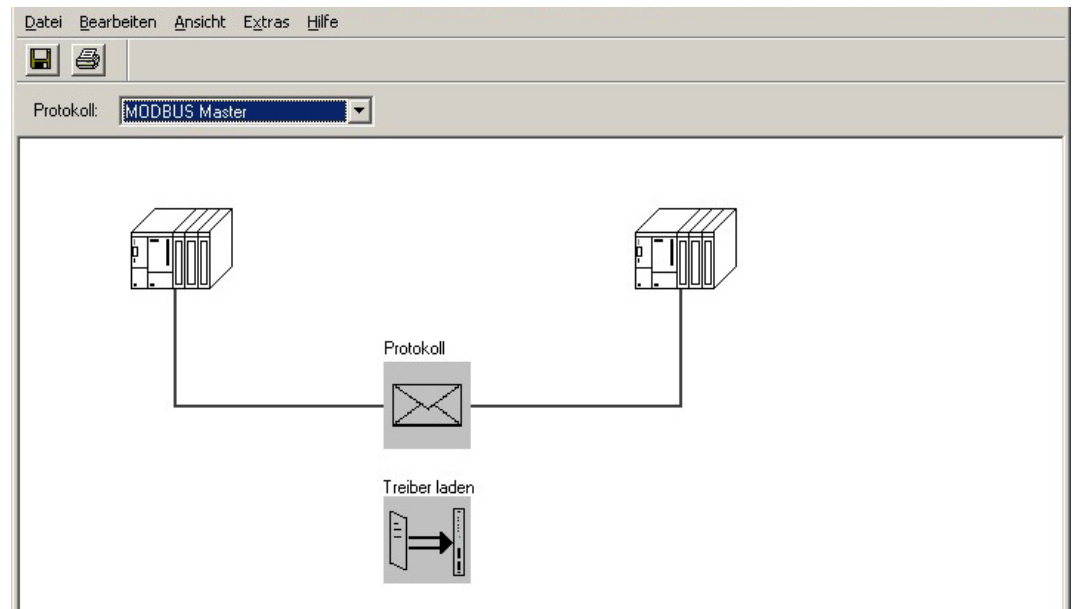
6.4.2 Modbus Master - Parameter


Modbus über Ladbare Treiber

- Für den Einsatz von Modbus Master auf dem CP 341-1AH01 ist ein ladbarer Treiber erforderlich. Diesen können Sie von der Siemens Webseite downloaden.
- Beim Einsatz ladbarer Treiber werden aus softwaretechnischen Gründen die Treiber von Siemens in den CP übertragen, dort aber nicht installiert.
- Da im CP Yaskawa eigene Treiber installiert sind, ist die Verwendung von Siemens üblichen Hardware-Dongles nicht erforderlich. Zur Installation des Treibers beenden Sie den Siemens SIMATIC Manager, öffnen Sie die Treiberdatei und folgen Sie den Anweisungen.

Vorgehensweise

1. Starten Sie nach der Installation des Treibers den Siemens SIMATIC Manager mit Ihrem Projekt.
2. Über die Schaltfläche [Parameter...] im Eigenschaften-Dialog des CP starten Sie das Parametrier-Plugin "Punkt-zu-Punkt-Kopplung parametrieren".
3. Hier können Sie die Parameter für Übertragungsprotokoll, Datenempfang und Schnittstelle einstellen.



4. Stellen Sie zuerst unter *Protokoll* das gewünschte Modbus-Protokoll ein:
 - Modbus Master RTU → "MODBUS Master"
 - Modbus Master ASCII → "MODBUS ASCII Master"
5. Zur Protokoll-Parametrierung klicken Sie auf .
 - ⇒ Nachfolgend sind diese Parameter beschrieben. Informationen hierzu finden Sie auch in der Online-Hilfe des Parametrier-Plugins.

Allgemein

- In diesem Dialogfenster erhalten Sie alle Informationen zum ladbaren Treiber. Hier können Sie nichts ändern.
- Unter Ladbarer Treiber finden Sie den Modbus-Typ gefolgt vom Übertragungsformat.
- Unter KP bzw. SCC offline auf dem PG wird Ihnen der Name und die Version des Kommunikationstreibers bzw. des seriellen Low-Level Transfer-Treibers angezeigt.

6.4.2.1 Modbus Master (RTU)

Geschwindigkeit Hier können Sie aus einer Auswahlliste die Geschwindigkeit für die Datenübertragung vorgeben.

| Parameter | Beschreibung | Defaultwert |
|-------------------|--|-------------|
| Baudrate in Bit/s | Übertragungsgeschwindigkeit in Bit/s <ul style="list-style-type: none"> ■ Wertebereich: 300, 600, 1200, 2400, 4800, 9600, 19200, 38400, 57600, 76800 | 9600 |

Zeichenrahmen Die Daten zwischen den Kommunikationspartnern werden über die serielle Schnittstelle in einem Zeichenrahmen übertragen. Somit kann beim Empfänger jedes übertragene Zeichen erkannt und kontrolliert werden.



Bitte beachten Sie, dass die folgenden Parameter bei allen Kommunikationspartnern gleich einzustellen sind.

| Parameter | Beschreibung | Defaultwert |
|-----------|--|-------------|
| Datenbits | Anzahl der Bits, auf die ein Zeichen abgebildet wird. Beim Modbus RTU-Protokoll sind 8 Datenbits fest eingestellt. <ul style="list-style-type: none"> ■ Wertebereich: 8 (fix) | 8 |
| Stopbits | Die Stopbits werden bei der Übertragung jedem zu übertragenden Zeichen nachgesetzt und kennzeichnen das Ende eines Zeichens. <ul style="list-style-type: none"> ■ Wertebereich: 1, 2 | 1 |
| Parität | Das Paritätsbit ergänzt durch seinen Wert "0" oder "1" die Summe aller Bits (Daten- und Paritätsbit) auf einen definierten Zustand. <ul style="list-style-type: none"> ■ Wertebereich: keine, ungerade, gerade | gerade |

Protokoll-Parameter

| Parameter | Beschreibung | Defaultwert |
|----------------------------------|--|---------------|
| Antwortüberwachungszeit | <p>Hier ist eine Wartezeit in ms vorzugeben, die der CP nach der Ausgabe eines Anforderungstelegramms auf ein Antworttelegramm vom Modbus-Slave wartet.</p> <ul style="list-style-type: none"> ■ Wertebereich: 5 ... 65500ms | 2000 |
| Betriebsart | <p>Hier können Sie die Betriebsart für den Treiber vorgeben.</p> <p>Im <i>Normal-Betrieb</i> führen alle Übertragungsfehler und Leitungsunterbrechungen sofort zu einer Fehlerbehandlung, auch wenn der Treiber sich im Leerlauf (Idle-Mode) befindet.</p> <p>In der Betriebsart <i>Störungsunterdrückung</i> werden Übertragungsfehler und Leitungsunterbrechungen im Leerlauf des Treibers nicht erkannt. Sobald aber der Treiber den Idle-Mode verlässt, führen Übertragungsfehler und Leitungsunterbrechungen zu einer Fehlerbehandlung.</p> <ul style="list-style-type: none"> ■ Wertebereich: Normal-Betrieb, Störungsunterdrückung | Normalbetrieb |
| Multiplikator Zeichenverzugszeit | <p>Kann ein Kommunikationspartner die Zeichenverzugszeit der Modbus-Spezifikation nicht einhalten, so haben Sie hier die Möglichkeit die Zeichenverzugszeit durch den <i>Multiplikator</i> zu vervielfachen.</p> <ul style="list-style-type: none"> ■ Wertebereich: 1 ... 10 | 1 |

6.4.2.2 Modbus Master (ASCII)

Geschwindigkeit Hier können Sie aus einer Auswahlliste die Geschwindigkeit für die Datenübertragung vorgeben.

| Parameter | Beschreibung | Defaultwert |
|-------------------|--|-------------|
| Baudrate in Bit/s | Übertragungsgeschwindigkeit in Bit/s <ul style="list-style-type: none"> ■ Wertebereich: 300, 600, 1200, 2400, 4800, 9600, 19200, 38400, 57600, 76800 | 9600 |

Zeichenrahmen Die Daten zwischen den Kommunikationspartnern werden über die serielle Schnittstelle in einem Zeichenrahmen übertragen. Somit kann beim Empfänger jedes übertragene Zeichen erkannt und kontrolliert werden.



Bitte beachten Sie, dass die folgenden Parameter bei allen Kommunikationspartnern gleich einzustellen sind.

| Parameter | Beschreibung | Defaultwert |
|-----------|--|-------------|
| Datenbits | Anzahl der Bits, auf die ein Zeichen abgebildet wird. Beim Modbus RTU-Protokoll sind 8 Datenbits fest eingestellt. <ul style="list-style-type: none"> ■ Wertebereich: 8 (fix) | 8 |
| Stopbits | Die Stopbits werden bei der Übertragung jedem zu übertragenden Zeichen nachgesetzt und kennzeichnen das Ende eines Zeichens. <ul style="list-style-type: none"> ■ Wertebereich: 1, 2 | 1 |
| Parität | Das Paritätsbit ergänzt durch seinen Wert "0" oder "1" die Summe aller Bits (Daten- und Paritätsbit) auf einen definierten Zustand. <ul style="list-style-type: none"> ■ Wertebereich: keine, ungerade, gerade | gerade |

Protokoll-Parameter

| Parameter | Beschreibung | Defaultwert |
|----------------------|---|------------------|
| Character Delay Time | <p>Hier ist eine Zeichenverzugszeit in ms vorzugeben.</p> <p>Die Zeichenverzugszeit ist die Zeit, die maximal zwischen zwei Zeichen in einem Modbus-Telegramm vergehen darf.</p> <p>Die Zeitüberwachung wird von der empfangenden Station durchgeführt. Bei Überschreiten der Zeit wird das Telegramm verworfen und es erfolgt eine Fehlerrückmeldung.</p> <ul style="list-style-type: none"> ■ Wertebereich: 1 ... 6500ms | 1000 |
| Response Time-out | <p>Hier ist die Antwortüberwachungszeit in ms vorzugeben, die der CP nach der Ausgabe eines Anforderungstelegramms auf ein Antworttelegramm vom Modbus-Slave wartet.</p> <ul style="list-style-type: none"> ■ Wertebereich: 5 ... 65500ms | 2000 |
| Turnaround Delay | <p>Hier stellen Sie die Zeit ein, welche mindestens zwischen zwei Broadcasts liegen muss.</p> <p>Mit der Einstellung 0 ist die Verzögerung deaktiviert.</p> <ul style="list-style-type: none"> ■ Wertebereich: 0 ... 65535ms | 0 |
| Operating Mode | <p>Hier können Sie die Betriebsart für den Treiber vorgeben.</p> <p>Unter <i>Normal Operation</i> (Normal-Betrieb) führen alle Übertragungsfehler und Leitungsunterbrechungen sofort zu einer Fehlerbehandlung, auch wenn der Treiber sich im Leerlauf (Idle-Mode) befindet.</p> <p>In der Betriebsart <i>Interference Suppression</i> (Störungsunterdrückung) werden Übertragungsfehler und Leitungsunterbrechungen im Leerlauf des Treibers nicht erkannt. Sobald aber der Treiber den Idle-Mode verlässt, führen Übertragungsfehler und Leitungsunterbrechungen zu einer Fehlerbehandlung.</p> <ul style="list-style-type: none"> ■ Wertebereich: Normal-Betrieb, Störungsunterdrückung | Normal Operation |
| with 32-Bit Register | <p>Mit den registerorientierten Funktionen FC 03, 06, 16 können Sie auch auf 32Bit-Register zugreifen.</p> <p>Durch Setzen dieses Parameters aktivieren Sie den Treiber, Register mit einer Länge von 4Byte zu verarbeiten.</p> <p>Den Zugriff auf 16Bit bzw. 32Bit bestimmen Sie über das Byte, welches den Funktionscode beinhaltet.</p> <p>Durch Setzen des 6. Bits im Funktionscode greifen Sie auf ein 32Bit großes Register zu.</p> <p>Ist das 6. Bit nicht gesetzt, erfolgt der Zugriff auf ein 16Bit großes Register.</p> <ul style="list-style-type: none"> ■ Wertebereich: aktiviert, deaktiviert | deaktiviert |

Datenübertragung

Der Modbus-Treiber bietet Ihnen die Möglichkeit mit *Datenflusskontrolle* zu senden und zu empfangen. Bei aktivierter Datenflusskontrolle können Sie zwischen den Kommunikationspartnern den Datenfluss synchronisieren, sofern diese unterschiedliche Bearbeitungsgeschwindigkeiten haben.

Datenflusskontrolle

| Parameter | Beschreibung | Defaultwert |
|--|---|-------------|
| Automatische Bedienung der V24-Signale | Hier können Sie die <i>Datenflusskontrolle</i> aktivieren bzw. deaktivieren. Im aktivierten Zustand werden die Datenflusskontroll-Parameter aktiviert und können geändert werden. <ul style="list-style-type: none"> ■ Wertebereich: aktiviert, deaktiviert | deaktiviert |

Datenflusskontroll-Parameter

Hier können Sie die Parameter für die Datenflusskontrolle ändern.

| Parameter | Beschreibung | Defaultwert |
|-----------------------|--|-------------|
| RTS-Wegnahmezeit | Bei "Autom. Bedienung der V24-Signale" Zeit, die nach dem Senden gewartet werden soll, bis der CP die Leitung RTS auf OFF setzt. <ul style="list-style-type: none"> ■ Wertebereich: 0 ... 65535ms in 10ms Schritten | 10ms |
| Datenausgabewartezeit | Bei "Autom. Bedienung der V24-Signale" Zeit, die der CP beim Senden nach dem Setzen der Leitung RTS auf ON, auf CTS=ON des Partners warten soll, bevor der Sendevorgang gestartet wird. <ul style="list-style-type: none"> ■ Wertebereich: 0 ... 65535ms in 10ms Schritten | 10ms |

6.4.3 Modbus Master - Funktionsweise

Übersicht

Bei Modbus erfolgt die Datenübertragung ohne Handshake. Der Master initiiert die Übertragung und wartet nach Ausgabe des Anforderungstelegramms die parametrisierte Antwortüberwachungszeit auf ein Antworttelegramm vom Slave. Die Art des Datenaustauschs zwischen Modbus-Systemen wird durch Funktionscodes gesteuert. Die Telegrammlänge ist abhängig vom verwendeten Funktionscode.

Telegrammaufbau

Modbus verwendet für die Kommunikation folgenden Telegrammaufbau:

| ADDR | FUNC | DATA | CRC-CHECK |
|------|------|--------|-----------|
| Byte | Byte | n Byte | Wort |

ADDR

Modbus-Slave-Adresse, welche im Bereich 1...255 liegen kann. Mit der Slave-Adresse 0 (Broadcast-Message) spricht der Master alle am Bus befindlichen Slaves an. Dies ist aber nur mit schreibenden Funktionscodes möglich. Hierbei sendet der Slave kein Antwort-Telegramm.

FUNC

Über den Modbus Funktionscode definieren Sie die Bedeutung und den Aufbau des Telegramms.

Modbus Funktionscodes Folgende Funktionscodes werden vom Treiber unterstützt:

| FC | Funktion | Aktion in der SPS | |
|----|-----------------------------------|--|------------------------|
| 01 | Read coil status | Bitweise lesen | Merker M |
| | | Bitweise lesen | Ausgänge A |
| | | Bitweise lesen (16Bit-Raster) | Zeiten T |
| | | Bitweise lesen (16Bit-Raster) | Zähler Z |
| 02 | Read input status | Bitweise lesen | Merker M |
| | | Bitweise lesen | Eingänge E |
| 03 | Read holding registers | Wortweise lesen | Datenbaustein DB |
| 04 | Read input registers | Wortweise lesen | Datenbaustein DB |
| 05 | Force single coil | Bitweise schreiben | Merker M |
| | | Bitweise schreiben | Ausgänge A |
| 06 | Preset single register | Wortweise schreiben | |
| 07 | Read exception status | Bitweise lesen | Ereignis-Bits |
| 08 | Loop back test | - | - |
| 11 | Fetch communication event counter | Status-Wort und Event-Counter lesen | Status, Event |
| 12 | Fetch communication event log | Eweiterten Status lesen | Status, Event, Message |
| 15 | Force multiple coils | Bitweise schreiben (1...2040Bits) | Merker M |
| | | Bitweise schreiben (1...2040Bits) | Ausgänge A |
| 16 | Preset multiple registers | Wortweise schreiben (1...127 Register) | Datenbaustein DB |

DATA Hier werden die für den Funktionscode spezifischen Daten übertragen. Nähere Informationen zum Aufbau dieses Feldes finden Sie bei den Funktionscodes. ↪ *Kap. 6.4.4 "Modbus Master - Funktionscodes" Seite 62*

CRC-CHECK

- Den Telegrammabschluss bildet die 2Byte lange Prüfsumme.
 - Hierbei wird zuerst das Low- und dann das High-Byte übertragen.
- Der Treiber für Modbus Master erkennt dann das Telegrammende, wenn nach der 3,5-fachen Zeichenverzugszeit keine Übertragung stattfindet.

Hieraus ergeben sich baudratenabhängig folgende Time-Out-Zeiten:

| Baudrate in Baud | Time-Out-Zeit in ms |
|------------------|---------------------|
| 76800 | 0,5 |
| 38400 | 1 |
| 19200 | 2 |
| 9600 | 4 |
| ... | ... |
| 300 | 128 |

Byte-Reihenfolge im Wort

Für die Byte-Reihenfolge im Wort gilt: Wort = High-Byte | Low-Byte

Antwort des Slaves

Ist kein Fehler aufgetreten, wird der Funktionscode zurückgeliefert.

Erkennt der Slave einen Fehler im Anforderungstelegramm, so setzt er das höchstwertigste Bit im Funktionscode (Funktionscode OR 80h) des Antworttelegramms und sendet dieses zusammen mit einem Byte, welches den Fehlercode beinhaltet zurück.

Slave-Antwort:

- OK → Funktionscode
- Fehler → Funktionscode OR 80h & Fehlercode

Fehlercodes

| Fehlercode | Bedeutung nach Modbus-Spezifikation | Ursache |
|------------|-------------------------------------|-------------------------------------|
| 1 | Illegal function | Unzulässiger Funktionscode |
| 2 | Illegal data address | Unzulässige Datenadresse beim Slave |
| 3 | Illegal data value | Unzulässiger Datenwert 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 |

Kommunikation mit dem Anwenderprogramm

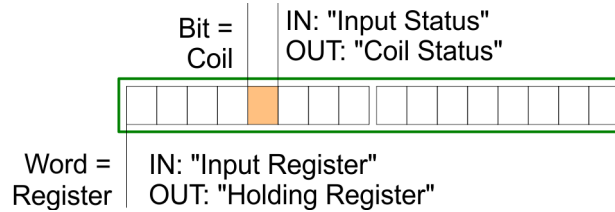
Für die Verarbeitung der Verbindungsaufträge ist in der CPU ein Anwenderprogramm erforderlich. Hierbei kommen zur Kommunikation zwischen CPU, CP und einem Kommunikationspartner die Bausteine FB 7 - P_RCV_RK und FB 8 - P_SND_RK zum Einsatz.



Näheres zum Einsatz dieser Bausteine finden Sie im Handbuch "SPEED7 Operationsliste" von Yaskawa.

6.4.4 Modbus Master - Funktionscodes

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



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

Modbus Funktionscodes Folgende Funktionscodes werden vom Treiber unterstützt:

| FC | Funktion | Aktion in der SPS | |
|----|-----------------------------------|--|------------------------|
| 01 | Read coil status | Bitweise lesen | Merker M |
| | | Bitweise lesen | Ausgänge A |
| | | Bitweise lesen (16Bit-Raster) | Zeiten T |
| | | Bitweise lesen (16Bit-Raster) | Zähler Z |
| 02 | Read input status | Bitweise lesen | Merker M |
| | | Bitweise lesen | Eingänge E |
| 03 | Read holding registers | Wortweise lesen | Datenbaustein DB |
| 04 | Read input registers | Wortweise lesen | Datenbaustein DB |
| 05 | Force single coil | Bitweise schreiben | Merker M |
| | | Bitweise schreiben | Ausgänge A |
| 06 | Preset single register | Wortweise schreiben | |
| 07 | Read exception status | Bitweise lesen | Ereignis-Bits |
| 08 | Loop back test | - | - |
| 11 | Fetch communication event counter | Status-Wort und Event-Counter lesen | Status, Event |
| 12 | Fetch communication event log | Eweiterten Status lesen | Status, Event, Message |
| 15 | Force multiple coils | Bitweise schreiben (1...2040Bits) | Merker M |
| | | Bitweise schreiben (1...2040Bits) | Ausgänge A |
| 16 | Preset multiple registers | Wortweise schreiben (1...127 Register) | Datenbaustein DB |

32Bit-Zugriff bei Modbus Master ASCII

1. ➤ Unter Modbus Master ASCII können Sie mit den registerorientierten Funktionen FC 03, 06, 16 auch auf 32Bit-Register zugreifen.
2. ➤ Hierzu ist über die Parametrierung der Protokoll-Eigenschaften unter "Modbus Master" der Parameter "with 32-Bit Register" zu aktivieren.
3. ➤ Im aktivierten Zustand haben Sie jetzt die Möglichkeit über einen "modifizierten" Funktionscode auf 32Bit große Register zuzugreifen.
4. ➤ Indem Sie das 6. Bit der Funktionscodes setzen, greifen Sie auf 32Bit große Register zu.

Ist das 6. Bit nicht gesetzt, erfolgt der Zugriff auf 16Bit große Register.

| FC | bei 16Bit-Zugriff | bei 32Bit-Zugriff |
|----|-------------------|-------------------|
| 03 | 03h | 43h |
| 06 | 06h | 46h |
| 16 | 10h | 50h |



- Bitte beachten Sie, dass der Zustand des 6. Bits den Funktionscode, welcher gesendet wird, nicht beeinflusst.
- Dies soll lediglich den Master informieren, welche Datengrößen dieser zu bearbeiten hat.
- Weiter ist zu beachten, dass Sie auch im Modbus Slave den 32Bit-Zugriff zu aktivieren haben.

6.4.4.1 FC 01 - Read Coil Status

Diese Funktion ermöglicht das Lesen einzelner Bits aus dem Ausgabebereich des Slaves.

Send-Quell-DB

| Adresse | Name | Typ | Kommentar |
|---------|------------|------|------------------|
| +0.0 | ADDR | BYTE | Slave-Adresse |
| +1.0 | FUNC | BYTE | Funktionscode |
| +2.0 | start_addr | WORD | Bit-Startadresse |
| +4.0 | bit_number | INT | Bit-Anzahl |

- *start_addr*
 - *start_addr* wird vom Treiber nicht überprüft und unverändert gesendet.
- *bit_number*
 - Als *bit_number* ist jeder Wert zwischen 1...2040 (ASCII: 1...2008) erlaubt.

RCV-Ziel-DB

| Adresse | Name | Typ | Kommentar |
|---------|---------|------|-----------|
| +0.0 | data[1] | WORD | Daten |
| +2.0 | data[2] | WORD | Daten |
| ... | ... | ... | ... |

- Die Daten des Antworttelegramms werden vom Treiber wortweise in den Ziel-DB eingetragen.
- Hierbei wird das 1. Byte als Low-Byte des 1. Wortes von "data[1]", das 3. Byte als Low-Byte des 2. Wortes "data[2]", usw. eingetragen.
- Werden weniger als 9Bit gelesen oder ist nur noch ein Low-Byte gelesen worden, so wird in das verbleibende High-Byte der Wert 00h eingetragen.

6.4.4.2 FC 02 - Read Input Status

Diese Funktion ermöglicht das Lesen einzelner Bits aus dem Eingabebereich des Slave.

Send-Quell-DB

| Adresse | Name | Typ | Kommentar |
|---------|------------|------|------------------|
| +0.0 | ADDR | BYTE | Slave-Adresse |
| +1.0 | FUNC | BYTE | Funktionscode |
| +2.0 | start_addr | WORD | Bit-Startadresse |
| +4.0 | bit_number | INT | Bit-Anzahl |

- *start_addr*
 - *start_addr* wird vom Treiber nicht überprüft und unverändert gesendet.
- *bit_number*
 - Als *bit_number* ist jeder Wert zwischen 1...2040 (ASCII: 1...2008) erlaubt.

RCV-Ziel-DB

| Adresse | Name | Typ | Kommentar |
|---------|---------|------|-----------|
| +0.0 | data[1] | WORD | Daten |
| +2.0 | data[2] | WORD | Daten |
| ... | ... | ... | ... |

- Die Daten des Antworttelegramms werden vom Treiber wortweise in den Ziel-DB eingetragen.
- Hierbei wird das 1. Byte als Low-Byte des 1. Wortes von "data[1]", das 3. Byte als Low-Byte des 2. Wortes "data[2]", usw. eingetragen.
- Werden weniger als 9Bit gelesen, so wird in das verbleibende High-Byte der Wert 00h eingetragen.

6.4.4.3 FC 03 - Read Output Registers

Diese Funktion ermöglicht das Lesen einzelner Register aus dem Ausgabebereich des Slave.

Send-Quell-DB

| Adresse | Name | Typ | Kommentar |
|---------|-----------------|------|-----------------------|
| +0.0 | ADDR | BYTE | Slave-Adresse |
| +1.0 | FUNC | BYTE | Funktionscode |
| +2.0 | start_register | WORD | Register-Startadresse |
| +4.0 | register_number | INT | Register-Anzahl |

- *start_register*
 - *start_register* wird vom Treiber nicht überprüft und unverändert gesendet.
- *register_number*
 - Sie können 1...127 (ASCII: 1...125) Register (Worte) lesen.

Rcv-Ziel-DB

| Adresse | Name | Typ | Kommentar |
|---------|---------|------|-----------|
| +0.0 | data[1] | WORD | Daten |
| +2.0 | data[2] | WORD | Daten |
| ... | ... | ... | ... |

6.4.4.4 FC 04 - Read Input Registers

Diese Funktion ermöglicht das Lesen einzelner Register aus dem Eingabebereich des Slaves.

Send-Quell-DB

| Adresse | Name | Typ | Kommentar |
|---------|-----------------|------|-----------------------|
| +0.0 | ADDR | BYTE | Slave-Adresse |
| +1.0 | FUNC | BYTE | Funktionscode |
| +2.0 | start_register | WORD | Register-Startadresse |
| +4.0 | register_number | INT | Register-Anzahl |

- *start_register*
 - *start_register* wird vom Treiber nicht überprüft und unverändert gesendet.
- *register_number*
 - Sie können 1...127 (ASCII: 1...125) Register (Worte) lesen.

RCV-Ziel-DB

| Adresse | Name | Typ | Kommentar |
|---------|---------|------|-----------|
| +0.0 | data[1] | WORD | Daten |
| +2.0 | data[2] | WORD | Daten |
| ... | ... | ... | ... |

6.4.4.5 FC 05 - Force Single Coil

Mit dieser Funktion können Sie in einem Slave ein einzelnes Bit im Ausgabebereich setzen bzw. löschen.

Send-Quell-DB

| Adresse | Name | Typ | Kommentar |
|---------|------------|------|---------------|
| +0.0 | ADDR | BYTE | Slave-Adresse |
| +1.0 | FUNC | BYTE | Funktionscode |
| +2.0 | coil_addr | WORD | Bit-Adresse |
| +4.0 | coil_state | WORD | Bit-Status |

- *coil_addr*
 - *coil_addr* wird vom Treiber nicht überprüft und unverändert gesendet.
- *coil_state*
 - *coil_state* wird vom Treiber nicht überprüft und unverändert gesendet.
Bei *coil_state* sind folgende Werte zulässig:
0000h → Bit = 0
FF00h → Bit = 1

6.4.4.6 FC 06 - Preset Single Register

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

Send-Quell-DB

| Adresse | Name | Typ | Kommentar |
|---------|----------------|------|------------------|
| +0.0 | ADDR | BYTE | Slave-Adresse |
| +1.0 | FUNC | BYTE | Funktionscode |
| +2.0 | start_register | WORD | Register-Adresse |
| +4.0 | register_value | WORD | Register-Wert |

- *start_register*
 - *start_register* wird vom Treiber nicht überprüft und unverändert gesendet.
- *register_value*
 - Als *register_value* kann jeder beliebige Wert verwendet werden.

6.4.4.7 FC 07 - Read Exception State

- Mit diesem Funktionscode können 8 Ereignis-Bits vom angeschlossenen Slave gelesen werden.
- Die Anfangsbitnummer der Ereignis-Bits ist durch den angeschlossenen Slave festgelegt und muss somit nicht vom Anwenderprogramm vorgegeben werden.

Send-Quell-DB

| Adresse | Name | Typ | Kommentar |
|---------|------|------|---------------|
| +0.0 | ADDR | BYTE | Slave-Adresse |
| +1.0 | FUNC | BYTE | Funktionscode |

RCV-Ziel-DB

| Adresse | Name | Typ | Kommentar |
|---------|---------|------|-----------|
| +0.0 | data[1] | WORD | 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.

6.4.4.8 FC 08 - Loop Back Diagnostic Test

- Diese Funktion dient zur Überprüfung der Kommunikations-Verbindung.
- Hierbei muss der Slave das vom Master empfangene Telegramm unverändert zurücksenden.
- Das Antworttelegramm wird nicht in einem RCV-DB eingetragen.

Send-Quell-DB

| Adresse | Name | Typ | Kommentar |
|---------|------------|------|-----------------|
| +0.0 | ADDR | BYTE | Slave-Adresse |
| +1.0 | FUNC | BYTE | Funktionscode |
| +2.0 | diag_code | WORD | Diagnostic Code |
| +4.0 | test_value | WORD | Test Wert |

- *diag_code*
 - Die Funktion unterstützt ausschließlich *diag_code* = 0000h.
- *test_value*
 - Beliebiger 16Bit-Wert als Test Wert.

6.4.4.9 FC 11 - Fetch Communications Event Counter

- Mit diesem Funktionscode haben Sie Zugriff auf die System-Worte "Status- Word" und "Event-Counter" eines Slave.
- Diese Worte sind im "Gould Modbus Protokoll" näher beschrieben.

Send-Quell-DB

| Adresse | Name | Typ | Kommentar |
|---------|------|------|---------------|
| +0.0 | ADDR | BYTE | Slave-Adresse |
| +1.0 | FUNC | BYTE | Funktionscode |

RCV-Ziel-DB

| Adresse | Name | Typ | Kommentar |
|---------|---------|------|---------------|
| +0.0 | data[1] | WORD | Status-Word |
| +2.0 | data[2] | WORD | Event-Counter |

6.4.4.10 FC 12 - Fetch Communication Event Log

- Mit diesem Funktionscode haben Sie Zugriff auf die System-Worte "Status-Word", "Event-Counter" und "Message-Counter" sowie auf die 64Byte "Event-Byte" des Slaves.
- Auch hier finden Sie nähere Informationen im "Gould Modbus Protokoll".

Send-Quell-DB

| Adresse | Name | Typ | Kommentar |
|---------|------|------|---------------|
| +0.0 | ADDR | BYTE | Slave-Adresse |
| +1.0 | FUNC | BYTE | Funktionscode |

RCV-Ziel-DB

| Adresse | Name | Typ | Kommentar |
|---------|--------------|------|-----------------|
| +0.0 | data[1] | WORD | Status-Word |
| +2.0 | data[2] | WORD | Event-Counter |
| +4.0 | data[3] | WORD | Message-Counter |
| +6.0 | bytedata[1] | BYTE | Event-Byte 1 |
| +7.0 | bytedata[2] | BYTE | Event-Byte 2 |
| ... | ... | ... | ... |
| +69.0 | bytedata[64] | BYTE | Event-Byte 64 |

6.4.4.11 FC 15 - Force Multiple Coils

Mit diesem Funktionscode können bis zu 2040 (ASCII: 1976) Bits im Slave geändert werden.

Send-Quell-DB

| Adresse | Name | Typ | Kommentar |
|---------|---------------|------|------------------|
| +0.0 | ADDR | BYTE | Slave-Adresse |
| +1.0 | FUNC | BYTE | Funktionscode |
| +2.0 | start_addr | WORD | Bit-Startadresse |
| +4.0 | bit_number | INT | Bit-Anzahl |
| +6.0 | coil_state[1] | WORD | Status Coil |

- *start_addr*
 - *start_addr* wird vom Treiber nicht überprüft und unverändert gesendet.
- *bit_number*
 - Als *bit_number* ist jeder Wert zwischen 1...2040 (ASCII: 1...1976) erlaubt. Hier geben Sie an, wie viele Bits im Slave zu überschreiben sind.
- *coil_state[1]*
 - Status Coil:
5Fh...58h
57h...50h

6.4.4.12 FC 16 - Preset Multiple Registers

Diese Funktion ermöglicht es, mit einem Anforderungstelegramm bis zu 127 (ASCII: 123) Register im Slave zu überschreiben.

Send-Quell-DB

| Adresse | Name | Typ | Kommentar |
|---------|-----------------|------|-----------------------|
| +0.0 | ADDR | BYTE | Slave-Adresse |
| +1.0 | FUNC | BYTE | Funktionscode |
| +2.0 | start_register | WORD | Register-Startadresse |
| +4.0 | register_number | INT | Register-Anzahl |
| +6.0 | data[1] | WORD | Register-Daten |
| +8.0 | data[2] | WORD | Register-Daten |
| +10.0 | data[3] | WORD | Register-Daten |
| ... | ... | ... | ... |

- *start_register*
 - *start_register* wird vom Treiber nicht überprüft und unverändert gesendet.
- *register_number*
 - Als *register_number* sind die Werte 1...127 (ASCII: 1...123) erlaubt. Hier geben Sie die Anzahl der Register (1 Register = zwei Bytes) an, welche gelesen werden sollen.

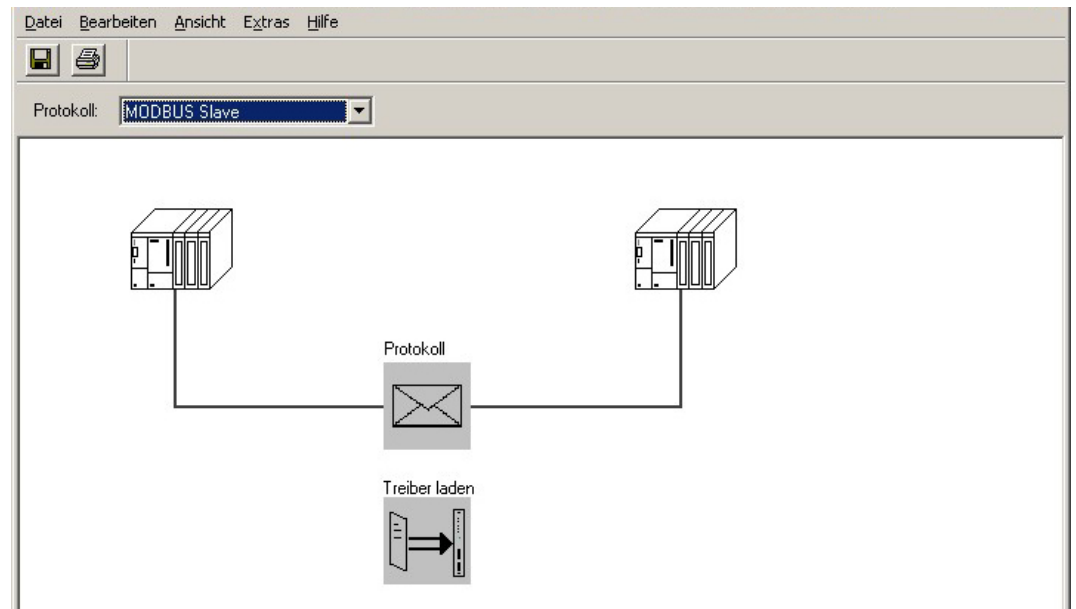
6.4.5 Modbus Slave - Parameter


Modbus über Ladbare Treiber

- Für den Einsatz von Modbus Slave auf dem CP 341-1AH01 ist ein ladbarer Treiber erforderlich.
- Diesen können Sie von der Siemens Webseite downloaden. Bei Einsatz ladbarer Treiber werden aus softwaretechnischen Gründen die Treiber von Siemens in den CP übertragen, dort aber nicht installiert.
- Da im CP Yaskawa-eigene Treiber installiert sind, ist die Verwendung von Siemens üblichen Hardware-Dongle nicht erforderlich.
- Zur Installation des Treibers beenden Sie den Siemens SIMATIC Manager, öffnen Sie die Treiberdatei und folgen Sie den Anweisungen.

Vorgehensweise

1. Starten Sie nach der Installation des Treibers den Siemens SIMATIC Manager mit Ihrem Projekt.
2. Über die Schaltfläche [Parameter...] im Eigenschaften-Dialog des CP starten Sie das Parametrier-Plugin "Punkt-zu-Punkt-Kopplung parametrieren".
3. Hier können Sie die Parameter für Übertragungsprotokoll, Datenempfang und Schnittstelle einstellen.



4. Stellen Sie zuerst unter *Protokoll* das gewünschte Protokoll "Modbus Slave" ein.
5. Zur Protokoll-Parametrierung klicken Sie auf .
 - ⇒ Nachfolgend sind diese Parameter beschrieben. Informationen hierzu finden Sie auch in der Online-Hilfe des Parametrier-Plugins.

Allgemein

- In diesem Dialogfenster erhalten Sie alle Informationen zum ladbaren Treiber. Hier können Sie nichts ändern.
- Unter Ladbarer Treiber finden Sie den Modbus-Typ gefolgt vom Übertragungsformat.
- Unter KP bzw. SCC offline auf dem PG wird Ihnen der Name und die Version des Kommunikationstreibers bzw. des seriellen Low-Level Transfer-Treibers angezeigt.

6.4.5.1 Parameter

Geschwindigkeit

Hier können Sie aus einer Auswahlliste die Geschwindigkeit für die Datenübertragung vorgeben.

| Parameter | Beschreibung | Defaultwert |
|-------------------|--|-------------|
| Baudrate in Bit/s | Übertragungsgeschwindigkeit in Bit/s <ul style="list-style-type: none"> ■ Wertebereich: 300, 600, 1200, 2400, 4800, 9600, 19200, 38400, 57600, 76800 | 9600 |

Zeichenrahmen

Die Daten zwischen den Kommunikationspartnern werden über die serielle Schnittstelle in einem Zeichenrahmen übertragen. Somit kann beim Empfänger jedes übertragene Zeichen erkannt und kontrolliert werden.



Bitte beachten Sie, dass die folgenden Parameter bei allen Kommunikationspartnern gleich einzustellen sind.

| Parameter | Beschreibung | Defaultwert |
|-----------|--|-------------|
| Datenbits | Anzahl der Bits, auf die ein Zeichen abgebildet wird. Beim Modbus RTU-Protokoll sind 8 Datenbits fest eingestellt. <ul style="list-style-type: none"> ■ Wertebereich: 8 (fix) | 8 |
| Stopbits | Die Stopbits werden bei der Übertragung jedem zu übertragenden Zeichen nachgesetzt und kennzeichnen das Ende eines Zeichens. <ul style="list-style-type: none"> ■ Wertebereich: 1, 2 | 1 |
| Parität | Das Paritätsbit ergänzt durch seinen Wert "0" oder "1" die Summe aller Bits (Daten- und Paritätsbit) auf einen definierten Zustand. <ul style="list-style-type: none"> ■ Wertebereich: keine, ungerade, gerade | gerade |

Protokoll-Parameter

| Parameter | Beschreibung | Defaultwert |
|----------------------------------|--|---------------|
| Slave-Adresse | Hier ist die eigene Slave-Adresse einzustellen, auf die der CP antworten soll. <ul style="list-style-type: none"> ■ Wertebereich: 1 ... 255 | 222 |
| Betriebsart | Hier können Sie die Betriebsart für den Treiber vorgeben. Im <i>Normalbetrieb</i> führen alle Übertragungsfehler und Leitungsunterbrechungen sofort zu einer Fehlerbehandlung, auch wenn der Treiber sich im Leerlauf (Idle-Mode) befindet. In der Betriebsart <i>Störungsunterdrückung</i> werden Übertragungsfehler und Leitungsunterbrechungen im Leerlauf des Treibers nicht erkannt. Sobald aber der Treiber den Idle-Mode verlässt, führen Übertragungsfehler und Leitungsunterbrechungen zu einer Fehlerbehandlung. <ul style="list-style-type: none"> ■ Wertebereich: Normalbetrieb, Störungsunterdrückung | Normalbetrieb |
| Multiplikator Zeichenverzugszeit | Kann ein Kommunikationspartner die Zeichenverzugszeit der Modbus-Spezifikation nicht einhalten, so haben Sie hier die Möglichkeit die Zeichenverzugszeit durch den Multiplikator zu vervielfachen. <ul style="list-style-type: none"> ■ Wertebereich: 1 ... 10 | 1 |

FC 01, 05, 15, 02

- In diesem Dialogfenster können Sie den Modbus-Adressen der bit orientierten Funktionscodes FC 01, 05 und 15 Adressbereiche in der CPU zuordnen.
- Mit diesen Funktionscodes haben Sie Zugriff auf Merker, Ausgänge, Zeiten und Zähler der CPU. Bei Zeiten und Zähler ist der lesende Zugriff ausschließlich mit Funktionscode FC 01 möglich.
- Über FC 02 ordnen Sie für lesenden Zugriff den Modbus Adress-Bereichen einen Merker- und Eingangsbereich in der CPU zu.

FC 03, 06, 16, 04

- Mit den Register orientierten Funktionscodes FC 03, 06 und 16 haben Sie lesenden und schreibenden Zugriff auf Datenbausteine in Ihrer CPU. Hier können Sie angeben, ab welcher DB-Nr. die Modbus-Adressen beginnend bei 0 zugeordnet sind.
- Sie haben Zugriff auf bis zu 128 zusammenhängende DBs. Über den Register orientierten Funktionscode FC 04 haben Sie nur lesenden Zugriff auf Datenbausteine in Ihrer CPU. Hier können Sie einen weiteren 128 DBs umfassenden Bereich bestimmen.
- Näheres hierzu finden Sie bei den entsprechenden Funktionscodes.

Grenzen

- Für die schreibenden Funktionscodes FC 05, 06, 15 und 16 müssen Sie den Zugriff auf die entsprechenden Bereiche freigeben.
- Per Default sind alle Bereiche für den schreibenden Zugriff gesperrt, d.h. alle Werte sind 0.
- Versucht der Master CPU-Ausgabebereiche zu beschreiben, welche außerhalb des freigegebenen Bereichs liegen, wird der Zugriff durch ein entsprechendes Fehlertelegramm abgewiesen.

Datenübertragung

Der Modbus-Treiber bietet Ihnen die Möglichkeit mit *Datenflusskontrolle* zu senden und zu empfangen. Bei aktivierter Datenflusskontrolle können Sie zwischen den Kommunikationspartnern den Datenfluss synchronisieren, sofern diese unterschiedliche Bearbeitungsgeschwindigkeiten haben.

Datenflusskontrolle

| Parameter | Beschreibung | Defaultwert |
|--|---|-------------|
| Automatische Bedienung der V24-Signale | Hier können Sie die <i>Datenflusskontrolle</i> aktivieren bzw. deaktivieren. Im aktivierten Zustand werden die Datenflusskontroll-Parameter aktiviert und können geändert werden. <ul style="list-style-type: none"> ■ Wertebereich: aktiviert, deaktiviert | deaktiviert |

Datenflusskontroll-Parameter

Hier können Sie die Parameter für die Datenflusskontrolle ändern.

| Parameter | Beschreibung | Defaultwert |
|-----------------------|--|-------------|
| RTS-Wegnahmezeit | Bei "Autom. Bedienung der V24-Signale" Zeit, die nach dem Senden gewartet werden soll, bis der CP die Leitung RTS auf OFF setzt. <ul style="list-style-type: none"> ■ Wertebereich: 0 ... 65535ms in 10ms Schritten | 10ms |
| Datenausgabewartezeit | Bei "Autom. Bedienung der V24-Signale" Zeit, die der CP beim Senden nach dem Setzen der Leitung RTS auf ON, auf CTS=ON des Partners warten soll, bevor der Sendevorgang gestartet wird. <ul style="list-style-type: none"> ■ Wertebereich: 0 ... 65535ms in 10ms Schritten | 10ms |

6.4.6 Modbus Slave - Funktionsweise

Übersicht

- Bei Modbus erfolgt die Datenübertragung ohne Handshake. Der Master initiiert die Übertragung und wartet nach Ausgabe des Anforderungstelegramms die parametrisierte Antwortüberwachungszeit auf ein Antworttelegramm vom Slave. Die Art des Datenaustauschs zwischen Modbus-Systemen wird durch Funktionscodes gesteuert.
- Auf Modbus-Slave-Seite transformiert der Protokolltreiber die Modbus-Adresse aus dem Telegramm des Masters auf Speicherbereiche der CPU. Die entsprechenden Bereichs-Zuordnungen können Sie in der Parametrierung durchführen.
- Der Datenaustausch zwischen CP und CPU erfolgt mit dem Modbus Kommunikations FB 80 - MODB_341.
 - Von diesem werden intern FB 7 - P_PRC_RK und FB 8 - P_SND_RK aufgerufen.
- Für die Kommunikation sind auf Slave-Seite die Bausteine FB 7 - P_PRC_RK und FB 8 - P_SND_RK in das Projekt zu übernehmen.

Telegrammaufbau

Modbus verwendet für die Kommunikation folgenden Telegrammaufbau:

| ADDR | FUNC | DATA | CRC-CHECK |
|------|------|--------|-----------|
| Byte | Byte | n Byte | Wort |

ADDR

Modbus-Slave-Adresse, welche im Bereich 1...255 liegen kann. Mit der Slave-Adresse 0 (Broadcast-Message) spricht der Master alle am Bus befindlichen Slaves an. Dies ist aber nur mit schreibenden Funktionscodes möglich. Hierbei sendet der Slave kein Antwort-Telegramm.

FUNC

Über den Modbus Funktionscode definieren Sie die Bedeutung und den Aufbau des Telegramms. Folgende Funktionscodes werden vom Modbus Slave Treiber unterstützt:

| FC | Funktion | Aktion in der SPS | |
|----|---------------------------|--|------------------|
| 01 | Read coil status | Bitweise lesen | Merker M |
| | | Bitweise lesen | Ausgänge A |
| | | Bitweise lesen (16Bit-Raster) | Zeiten T |
| | | Bitweise lesen (16Bit-Raster) | Zähler Z |
| 02 | Read input status | Bitweise lesen | Merker M |
| | | Bitweise lesen | Eingänge E |
| 03 | Read holding registers | Wortweise lesen | Datenbaustein DB |
| 04 | Read input registers | Wortweise lesen | Datenbaustein DB |
| 05 | Force single coil | Bitweise schreiben | Merker M |
| | | Bitweise schreiben | Ausgänge A |
| 06 | Preset single register | Wortweise schreiben | Datenbaustein DB |
| 08 | Loop back test | - | - |
| 15 | Force multiple coils | Bitweise schreiben (1...2040Bits) | Merker M |
| | | Bitweise schreiben (1...2040Bits) | Ausgänge A |
| 16 | Preset multiple registers | Wortweise schreiben (1...127 Register) | Datenbaustein DB |



Bitte beachten Sie dass Sie, sobald Sie über Funktionscodes schreibend auf den Slave zugreifen möchten, diesen Bereich über die Protokoll-Parametrierung im Dialogfenster "Grenzen" freigeben.

DATA Hier werden die für den Funktionscode spezifischen Daten übertragen. Nähere Informationen zum Aufbau dieses Feldes finden Sie bei den Funktionscodes. ↪ *Kap. 6.4.8 "Modbus Slave - Funktionscodes" Seite 83*

CRC-CHECK

- Den Telegrammabschluss bildet die 2Byte lange Prüfsumme.
 - Hierbei wird zuerst das Low- und dann das High-Byte übertragen.
- Der Treiber für Modbus Master erkennt dann das Telegrammende, wenn nach der 3,5-fachen Zeichenverzugszeit keine Übertragung stattfindet.

Hieraus ergeben sich baudratenabhängig folgende Time-Out-Zeiten:

| Baudrate in Baud | Time-Out-Zeit in ms |
|------------------|---------------------|
| 76800 | 0,5 |
| 38400 | 1 |
| 19200 | 2 |
| 9600 | 4 |
| ... | ... |
| 300 | 128 |

Byte-Reihenfolge im Wort Für die Byte-Reihenfolge im Wort gilt: Wort = High-Byte | Low-Byte

Antwort des Slaves Ist kein Fehler aufgetreten, wird der Funktionscode zurückgeliefert. Erkennt der Slave einen Fehler im Anforderungstelegramm, so setzt er das höchstwertigste Bit im Funktionscode (Funktionscode OR 80h) des Antworttelegramms und sendet dieses zusammen mit einem Byte, welches den Fehlercode beinhaltet zurück.

Slave-Antwort:

- OK → Funktionscode
- Fehler → Funktionscode OR 80h & Fehlercode

Fehlercodes

Folgende Fehlercodes sind gemäß Modbus-Spezifikation definiert:

| Fehlercode | Bedeutung nach Modbus-Spezifikation | Ursache |
|------------|-------------------------------------|-------------------------------------|
| 1 | Illegal function | Unzulässiger Funktionscode |
| 2 | Illegal data address | Unzulässige Datenadresse beim Slave |
| 3 | Illegal data value | Unzulässiger Datenwert 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 |

6.4.7 Modbus Slave - Kommunikation mit Anwenderprogramm

Übersicht

- Für die Verarbeitung der Verbindungsaufträge ist auf der Slave-Seite ein Anwenderprogramm in der CPU erforderlich.
- Der Datenaustausch zwischen CP und CPU erfolgt mit dem Modbus Kommunikations FB 80 - MODB_341.
 - Von diesem werden intern FB 7 - P_RCV_RK und FB 8 - P_SND_RK aufgerufen.
- Für die Kommunikation sind auf Slave-Seite immer die Bausteine FB 7 - P_RCV_RK und FB 8 - P_SND_RK in das Projekt zu übernehmen.
- Alle für den Modbus Kommunikations FB 80 relevanten Daten liegen in einem Instanz-DB. Dieser DB ist zugleich Instanz-DB für alle innerhalb des FB aufgerufenen Bausteine. Auf den Instanz-DB dürfen Sie nur lesend zugreifen.



VORSICHT!

- Ein Aufruf des FB 80 - MODB_341 im Prozess- oder Diagnosealarm ist nicht zulässig.
- Bitte beachten Sie, dass der FB keine Parameterprüfung besitzt, d.h. bei falscher Parametrierung kann die CPU in den Zustand STOP verzweigen.

Installation

1. Der Funktionsbaustein FB 80 wird zusammen mit dem Protokoll-Treiber installiert.
2. Wenn nicht schon geschehen, beenden Sie den Siemens SIMATIC Manager, starten Sie die Installations-Datei des Treibers und folgen Sie den Anweisungen.
3. Nach der Installation finden Sie den FB 80 - MODB_341 in der Baustein- Bibliothek.
4. Die Bibliothek öffnen Sie im Siemens SIMATIC Manager über: "Datei → Öffnen → Bibliotheken" und hier "Modbus".

Kommunikationsprinzip

- Durch zyklischen Aufruf des FB 80 - MODB_341 können Sie mit dem Slave-CP zyklisch Anforderungstelegramme vom Master empfangen und Daten senden.
- Auf dem CP erfolgt die Umsetzung der entsprechenden Modbus-Adresse in den Speicherbereich der CPU.
- Über die Parametrierung in der Hardware-Konfiguration können Sie die Bereichszuordnungen durchführen.
- Der FB 80 - MODB_341 ist nachfolgend beschrieben. ↪ *Kap. 6.4.7.1 "Daten senden FB 80 - MODB_341" Seite 78*

Reaktionszeit

- Für die schreibenden Funktionscodes (FC 05, FC 15) gilt:
 - Reaktionszeit = AG-Zyklus + Zeit CP → CPU + Zeit CPU → CP
- Für alle übrigen Funktionscodes gilt:
 - Reaktionszeit = Zeit CP → CPU + Zeit CPU → CP
- Der CP sendet das Antworttelegramm an das Master-System erst nach dem Datentransfer CPU → CP.
 - Die Standard-Antwortüberwachungszeit von 2s kann hierbei eingehalten werden.

6.4.7.1 Daten senden FB 80 - MODB_341

Der FB 80 - MODB_341 ist im Anwenderprogramm zyklisch aufzurufen. Hierbei empfängt dieser das Anforderungstelegramm des Modbus Master, ordnet die Modbus-Adresse dem entsprechenden Speicherbereich in der CPU zu und stellt die angeforderten Daten dem Master zur Verfügung.

Parameter

| Parameter | Deklaration | Datentyp | Beschreibung |
|----------------|----------------|----------|--|
| LADDR | Input | INT | Logische Basisadresse des CP - entspricht der Adresse aus der Hardware-Konfiguration des CP. |
| START_TIMER | Input | TIMER | Timer-Nr. für die Überwachungszeit der Initialisierung |
| START_TIME | Input | S5TIME | Zeitwert für die Überwachungszeit |
| OB_MASK | Input | BOOL | Peripheriezugriffsfehler maskieren, Alarme verzögert |
| CP_START | Input / Output | BOOL | FB-Initialisierung starten |
| CP_START_FM | Input / Output | BOOL | Flankenmerker <i>CP_START</i> |
| CP_START_NDR | Input / Output | BOOL | Info: Schreibanforderung vom CP |
| CP_START_OK | Input / Output | BOOL | Initialisierung ohne Fehler beendet (Zeit lag innerhalb der Überwachungszeit) |
| CP_START_ERROR | Input / Output | BOOL | Initialisierung mit Fehler beendet (Zeit lag außerhalb der Überwachungszeit) |
| ERROR_NR | Input / Output | WORD | Fehlernummer |
| ERROR_INFO | Input / Output | WORD | Fehlerzusatzinformation |

- LADDR
 - Geben Sie hier die logische Basisadresse des CP an.
 - Diese entspricht der Adresse aus der Hardware-Konfiguration des CP.
- START_TIMER, START_TIME
 - Nach NetzeIN erfordert der CP mehrere Sekunde, bis dieser betriebsbereit ist. Die während dieser Zeit vom FB durchgeführten Initialisierungsversuche werden mit Fehler beendet. Aus diesem Grund wiederholt der FB seinen Initialisierungsauftrag innerhalb der unter *START_TIME* parametrisierten Überwachungszeit mit dem unter *START_TIMER* angegebenen Timer.
- OB_MASK
 - Durch Aktivierung von *OB_MASK* (=True) können Sie Zugriffsfehler auf den Peripheriebereich der CPU maskieren. Hierbei geht die CPU bei Zugriff auf einen nicht vorhandenen Peripheriebereich nicht in STOP bzw. ruft nicht den Fehler-OB auf. Der Zugriffsfehler wird aber erkannt und die Funktion mit einer Fehlermeldung an den CP beendet.
- CP_START
 - Nach jedem Neustart oder Wiederanlauf der CPU muss eine Initialisierung des FB 80 - MODB_341 durchgeführt werden. Die Initialisierung aktivieren Sie mit einer steigenden Flanke an *CP_START*.
- CP_START_FM, CP_START_NDR
 - *CP_START_FM* ist der Flankenmerker für *CP_START*. Gesetzt bei einer Schreib-anforderung vom CP.

- CP_START_OK, CP_START_ERROR
 - Sobald der SEND-Auftrag ohne Fehler beendet ist, wird der Ausgang CP_START_OK gesetzt und die FB-Initialisierung ist abgeschlossen.
 - Wird der SEND-Auftrag mit Fehler beendet, wird CP_START zurückgesetzt und CP_START_ERROR gesetzt.
- ERROR_NR, ERROR_INFO
 - Mit ERROR_NR und ERROR_INFO erhalten Sie nähere Informationen zu einem aufgetretenen Fehler.
 - Mit einer steigenden Flanke an CP_START werden die Fehler wieder gelöscht.

ERROR_NR 1 ... 2

- Fehler bei der Initialisierung FB und CP
 - Bei den Fehlernummern 1 ... 2 wurde die Initialisierung mit Fehler beendet.
 - Der Parameter CP_START_ERROR ist 1.
 - Es ist keine Modbus Kommunikation zum Master-System möglich.

| ERROR_NR (dezimal) | ERROR_INFO | Fehlertext |
|-----------------------|------------------------------------|--|
| 0 | 0 | kein Fehler |
| 1 | SFC 51 → RET_VAL | <ul style="list-style-type: none"> ■ Fehler beim Lesen der SZL mit dem SFC 51. <ul style="list-style-type: none"> – <i>Abhilfe: RET_VAL in ERROR_INFO analysieren, Ursache beseitigen.</i> |
| 2 | SFB 12 → STATUS SFB 22 → STATUS | <ul style="list-style-type: none"> ■ TimeOut bei CP Initialisierung oder Fehler bei CP Initialisierung (Fehler bei BSEND-Auftrag). <ul style="list-style-type: none"> – <i>Abhilfe: Prüfen Sie, ob für die CP-Schnittstelle als Protokoll "Modbus Slave" parametrisiert wurde.</i> – <i>Prüfen Sie, ob die am Kommunikations-FB angegebene "ID" korrekt ist.</i> – <i>Analysieren Sie ERROR_INFO.</i> |

ERROR_NR 10 ... 19

- Fehler bei der Bearbeitung eines Funktionscodes
 - Bei den Fehlernummern 10 ... 19 ist bei der Bearbeitung eines Funktionscodes ein Fehler aufgetreten.
 - Der CP hat dem Kommunikations-FB einen unzulässigen Bearbeitungsauftrag gesendet.
 - Der Fehler wird ebenfalls an den Treiber gemeldet und nachfolgende Bearbeitungsaufträge werden weiter bearbeitet.

| ERROR_NR (dezimal) | ERROR_INFO | Fehlertext |
|-----------------------|-------------------|---|
| 10 | Bearbeitungs-Code | <ul style="list-style-type: none"> ■ Unzulässige Bearbeitungsfunktion vom Treiber an den Kommunikations-FB übergeben. <ul style="list-style-type: none"> – <i>Abhilfe: Neuanlauf des CP (NetzEin)</i> |
| 11 | Start-Adresse | <ul style="list-style-type: none"> ■ Unzulässige Start-Adresse vom Treiber an den Kommunikations-FB übergeben. <ul style="list-style-type: none"> – <i>Abhilfe: Modbus-Adresse vom Modbus-Master-System kontrollieren.</i> |

| ERROR_NR (dezimal) | ERROR_INFO | Fehlertext |
|-----------------------|--------------------------|--|
| 12 | Register-Anzahl | <ul style="list-style-type: none"> ■ Unzulässige Register-Anzahl vom Treiber an den Kommunikations-FB übergeben: Register-Anzahl = 0. <ul style="list-style-type: none"> – <i>Abhilfe: Register-Anzahl vom Modbus-Master-System kontrollieren, gegebenenfalls Neuanlauf des CP (NetzEIN).</i> |
| 13 | Register-Anzahl | <ul style="list-style-type: none"> ■ Unzulässige Register-Anzahl vom Treiber an den Kommunikations-FB übergeben: Register-Anzahl > 128 <ul style="list-style-type: none"> – <i>Abhilfe: Register-Anzahl vom Modbus-Master-System kontrollieren, gegebenenfalls Neuanlauf des CP (NetzEIN).</i> |
| 14 | Merker M - End-Adresse | <ul style="list-style-type: none"> ■ Zugriffsversuch auf den Speicherbereich "Merker" über das Bereichsende hinaus. <ul style="list-style-type: none"> – Achtung: Die Bereichslänge der CPU ist CPU-Typ-abhängig. – <i>Abhilfe: Modbus-Start-Adresse bzw. Zugriffslänge im Modbus-Master-System verringern.</i> |
| 15 | Ausgänge A - End-Adresse | <ul style="list-style-type: none"> ■ Zugriffsversuch auf den Speicherbereich "Ausgänge" über das Bereichsende hinaus. <ul style="list-style-type: none"> – Achtung: Die Bereichslänge der CPU ist CPU-Typ-abhängig. – <i>Abhilfe: Modbus-Start-Adresse bzw. Zugriffslänge im Modbus-Master-System verringern.</i> |
| 16 | Zeiten T - End-Adresse | <ul style="list-style-type: none"> ■ Zugriffsversuch auf den Speicherbereich "Zeiten (Timer)" über das Bereichsende hinaus. <ul style="list-style-type: none"> – Achtung: Die Bereichslänge der CPU ist CPU-Typ-abhängig. – <i>Abhilfe: Modbus-Start-Adresse bzw. Zugriffslänge im Modbus-Master-System verringern.</i> |
| 17 | Zähler Z - End-Adresse | <ul style="list-style-type: none"> ■ Zugriffsversuch auf den Speicherbereich "Zähler (Counter)" über das Bereichsende hinaus. <ul style="list-style-type: none"> – Achtung: Die Bereichslänge der CPU ist CPU-Typ-abhängig. – <i>Abhilfe: Modbus-Start-Adresse bzw. Zugriffslänge im Modbus-Master-System verringern.</i> |
| 18 | 0 | <ul style="list-style-type: none"> ■ Unzulässiger Speicherbereich vom Treiber an den Kommunikations-FB übergeben. <ul style="list-style-type: none"> – <i>Abhilfe: Gegebenenfalls Neuanlauf des CP (NetzEIN).</i> |
| 19 | | <ul style="list-style-type: none"> ■ Fehler beim Zugriff auf die Peripherie <ul style="list-style-type: none"> – <i>Abhilfe: Überprüfen, ob die benötigte Peripherie vorhanden und fehlerfrei ist.</i> |

ERROR_NR 90 ... 99

- Sonstige Fehler
 - Es ist ein Verarbeitungsfehler aufgetreten, welcher nicht an den Treiber gemeldet wird.
Nachfolgende Bearbeitungs-Aufträge werden weiter bearbeitet.

| ERROR_NR (dezimal) | ERROR_INFO | Fehlertext |
|-----------------------|-----------------|--|
| 90 | SFB 12 → STATUS | <ul style="list-style-type: none"> ■ Fehler beim Senden einer Quittungsmeldung an den Treiber mit dem SFB 12 (BSEND) <ul style="list-style-type: none"> – <i>Abhilfe: STATUS Information analysieren.</i> |
| 91 | SFB 22 → STATUS | <ul style="list-style-type: none"> ■ Fehler beim Lesen des SYSTAT mit dem SFB 22 (STATUS). <ul style="list-style-type: none"> – <i>Abhilfe: STATUS Information analysieren.</i> |
| 92 | FB 7 → STATUS | <ul style="list-style-type: none"> ■ Fehler beim Ausführen eines RECEIVE/FETCH-Aufrufs mit dem FB 7 (RCV_RK). <ul style="list-style-type: none"> – <i>Abhilfe: FB 7-STATUS Information analysieren.</i> |

Beispielprogramm**OB 100**

```
UN M 100.0 // CP_START setzen
S M 100.0 //
U M 100.1 // CP_START_FM rücksetzen
R M 100.1 //
```

OB 1

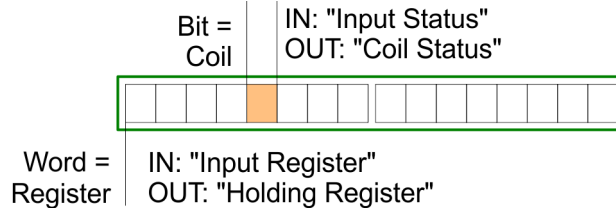
```
Call FB 80 , DB80 // Modbus Slave CP341 FB
LADDR: =256 // Basisadresse des CP
START_TIMER: =T120 // Timer Anlauf
START_TIME: =S5T#5S // Zeitwert Anlauf
OB_MASK: =TRUE // Fehler maskieren
CP_START: =M100.0 // Initialisierung Anlauf
CP_START_FM: =M100.1 // Flankenmerker
CP_NDR: =M100.2 // Neuer Schreibauftrag CP
CP_START_OK: =M100.3 // Init. CP-FB ohne Fehler
CP_START_ERROR: =M100.4 // Init. CP mit Fehler
CP_ERROR_NR: =MW102 // Fehler-Nr.
CP_ERROR_INFO: =MW104 // Fehler Zusatzinformation
```

6.4.7.1.1 Datenkonsistenz

- Der Datenaustausch zwischen CPU und CP erfolgt blockweise über FB 7 - P_RCV_RK und FB 8 - P_SND_RK.
 - Hierbei beträgt die Blockgröße 32Byte.
- Eine Datenkonsistenz ist ausschließlich für die Blockgröße von 32Byte gegeben.
- Bei größeren Datenmengen werden die Daten zeitlich versetzt in der genannten Blockgröße ausgetauscht. Zwischen diesen Datenblöcken besteht keine Konsistenz, da diese zwischenzeitlich vom Anwenderprogramm bearbeitet werden können.
- Ein Zugriff auf den CPU-Speicherbereich erfolgt immer dann, wenn der FB 7 - P_RCV_RK durchlaufen wird.
- Ist eine Datenkonsistenz beim Lesen/Schreiben von Registern/Bits erforderlich, muss die mit einem einzelnen Telegramm übertragene Datenmenge auf die Blockgröße von 32Byte beschränkt werden.
- Beispielsweise 16 Register bei FC 03, 04, 16 oder maximal 256Bits bei FC 01, 02, 15.
- Ansonsten müssen Sie über Ihr Anwenderprogramm sicherstellen, dass zusammengehörende Datenbereiche konsistent übertragen werden.

6.4.8 Modbus Slave - Funktionscodes

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



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

Modbus Funktionscodes Folgende Modbus Funktionscodes werden vom Treiber unterstützt:

| FC | Funktion | Aktion in der SPS | |
|----|---------------------------|--|------------------|
| 01 | Read coil status | Bitweise lesen | Merker M |
| | | Bitweise lesen | Ausgänge A |
| | | Bitweise lesen (16Bit-Raster) | Zeiten T |
| | | Bitweise lesen (16Bit-Raster) | Zähler Z |
| 02 | Read input status | Bitweise lesen | Merker M |
| | | Bitweise lesen | Eingänge E |
| 03 | Read holding registers | Wortweise lesen | Datenbaustein DB |
| 04 | Read input registers | Wortweise lesen | Datenbaustein DB |
| 05 | Force single coil | Bitweise schreiben | Merker M |
| | | Bitweise schreiben | Ausgänge A |
| 06 | Preset single register | Wortweise schreiben | Datenbaustein DB |
| 08 | Loop back test | - | - |
| 15 | Force multiple coils | Bitweise schreiben (1...2040Bits) | Merker M |
| | | Bitweise schreiben (1...2040Bits) | Ausgänge A |
| 16 | Preset multiple registers | Wortweise schreiben (1...127 Register) | Datenbaustein DB |



- Bei allen Funktionscodes, welche auf DBs in der CPU zugreifen (FC 03, 04, 06, 16), unterstützt der Modbus Slave Treiber eine Datenbausteinlänge von 512 Worte.
- Mit einem Telegramm können Sie immer nur auf einen DB zugreifen.
- Ansonsten bekommen Sie eine Fehlermeldung.

6.4.8.1 FC 01 - Read Coil Status

Diese Funktion ermöglicht das Lesen einzelner Bits aus dem Ausgabe-Speicherbereich der CPU über den Modbus-Master.

Anforderungstelegramm

| | | | | |
|------|------|------------|------------|-----|
| ADDR | FUNC | start_addr | bit_number | CRC |
|------|------|------------|------------|-----|

Antworttelegramm

| | | | | |
|------|------|--------------|-------------|-----|
| ADDR | FUNC | byte_count n | n byte data | CRC |
|------|------|--------------|-------------|-----|

start_addr

- Die Modbus Bit-Adresse *start_addr* beinhaltet den Beginn des Bereichs, auf den Sie in der CPU zugreifen möchten.
- Die entsprechende Adresszuordnung der CPU-Speicherbereiche führen Sie über die Eigenschaften "FC 01, 05, 15" in der Parametrierung des CP durch.
 - Hier können Sie einer "Modbus-Adresse im Übertragungstelegramm" kurz *Param-Start-Adresse* einen "SIMATIC Speicherbereich" kurz SPS-Bereich zuordnen.

Umrechnung Merker und Ausgänge

- $Byteadresse = ((start_addr - Param-Start-Adresse) / 8) + SPS-Bereich$
- Beim Zugriff auf Merker bzw. Ausgänge in der SPS wird die verbleibende Restbitnummer berechnet und dazu verwendet, um das entsprechende Bit innerhalb des Merker- bzw. Ausgangsbereichs zu adressieren.
- $Restbitnummer = (start_addr - Param-Start-Adresse) \% 8$ [Modulo 8]

Umrechnung Zähler und Zeiten

- $Wortadresse = ((Startadresse - Param-Start-Adresse) / 16) + SPS-Bereich$
- Bei der Adressberechnung muss das Ergebnis *Startadresse - Param-Start-Adresse* ohne Rest durch 16 teilbar sein. Es ist nur wortweiser Zugriff beginnend an Wortgrenzen erlaubt.

bit_number

- Als *bit_number* sind Werte zwischen 1 und 2040 erlaubt.
- Diese Anzahl von Bits wird gelesen.
- Bei Zugriff auf Zeiten und Zähler muss die *bit_number* durch 16 teilbar sein.
- Sie haben maximal Zugriff auf 16 Zeiten bzw. Zähler.

Beispiel

Umsetzung Modbus-Adressierung bei FC 01, 05, 15

| "Modbus-Adresse im Übertragungstelegramm" <i>Param-Start-Adresse</i> | "SIMATIC-Speicherbereich" <i>SPS-Bereich</i> |
|---|---|
| von 0 ... 1023 | ab Merker M 1000.0 |
| von 1024 ... 2047 | ab Ausgang A 100.0 |
| von 2048 ... 4057 | ab Zeit T 100 |
| von 4064 ... 4096 | ab Zähler Z 200 |

- Adressberechnung:
 - Byteadresse = $((start_addr - Param-Start-Adresse) / 8) + SPS-Bereich$
 - Restbitnummer = $(start_addr - Param-Start-Adresse) \% 8$ [Modulo 8]

| start_addr | | Zugriff | Berechnung | | | | Bereich in SPS |
|------------|---------|---------|---------------|------|-------|---|----------------|
| hex | dezimal | | | | | | |
| 0000h | 0 | Merker | (0 - 0) | / 8 | +1000 | → | M 1000.0 |
| 0001h | 1 | Merker | (1 - 0) | / 8 | +1000 | → | M 1000.1 |
| 01F1h | 497 | Merker | (497 - 0) | / 8 | +1000 | → | M 1062.1 |
| 0400h | 1024 | Ausgang | (1024 - 1024) | / 8 | +100 | → | A 100.0 |
| 0401h | 1025 | Ausgang | (1025 - 1024) | / 8 | +100 | → | A 100.1 |
| 07DAh | 2010 | Ausgang | (2010 - 1024) | / 8 | +100 | → | A 223.2 |
| 0800h | 2048 | Zeiten | (2048 - 2048) | / 16 | +100 | → | T 100 |
| 0801h | 2064 | Zeiten | (2064 - 2048) | / 16 | +100 | → | T 101 |
| 0C80h | 3200 | Zeiten | (3200 - 2048) | / 16 | +100 | → | T 172 |
| 0FE0h | 4064 | Zähler | (4064 - 4064) | / 16 | +200 | → | Z 200 |
| 0FF0h | 4080 | Zähler | (4080 - 4064) | / 16 | +200 | → | Z 201 |
| 1000h | 4096 | Zähler | (4096 - 4064) | / 16 | +200 | → | Z 202 |

6.4.8.2 FC 02 - Read Input Status

Diese Funktion ermöglicht das Lesen einzelner Bits aus dem Eingabe-Speicherbereich der CPU über den Modbus-Master.

Anforderungstelegramm

| | | | | |
|------|------|------------|------------|-----|
| ADDR | FUNC | start_addr | bit_number | CRC |
|------|------|------------|------------|-----|

Antworttelegramm

| | | | | |
|------|------|--------------|-------------|-----|
| ADDR | FUNC | byte_count n | n byte data | CRC |
|------|------|--------------|-------------|-----|

start_addr

- Die Modbus Bit-Adresse *start_addr* beinhaltet den Beginn des Bereichs, auf den Sie in der CPU zugreifen möchten.
- Die entsprechende Adresszuordnung der CPU-Speicherbereiche führen Sie über die Eigenschaften "FC 02" in der Parametrierung des CP durch.
 - Hier können Sie einer "Modbus-Adresse im Übertragungstelegramm" kurz *Param-Start-Adresse* einen "SIMATIC Speicherbereich" kurz SPS-Bereich zuordnen.

Umrechnung

- $Byteadresse = ((start_addr - Param-Start-Adresse) / 8) + SPS-Bereich$

Beim Zugriff auf Merker bzw. Ausgänge in der SPS wird die verbleibende Restbitnummer berechnet und dazu verwendet, um das entsprechende Bit innerhalb des Merker- bzw. Ausgangsbereichs zu adressieren.

$$Restbitnummer = (start_addr - Param-Start-Adresse) \% 8 \text{ [Modulo 8]}$$

bit_number

- Als *bit_number* sind Werte zwischen 1 und 2040 erlaubt.

Beispiel

Umsetzung Modbus-Adressierung bei FC 02

| "Modbus-Adresse im Übertragungstelegramm" <i>Param-Start-Adresse</i> | "SIMATIC-Speicherbereich" <i>SPS-Bereich</i> |
|---|---|
| von 0 ... 1023 | ab Merker M 1000.0 |
| von 1024 ... 2047 | ab Eingang E 100.0 |

| start_addr | | Zugriff | Berechnung | | | | Bereich in SPS |
|------------|---------|---------|---------------|-----|-------|---|----------------|
| hex | dezimal | | | | | | |
| 0000h | 0 | Merker | (0 - 0) | / 8 | +1000 | → | M 1000.0 |
| 0001h | 1 | Merker | (1 - 0) | / 8 | +1000 | → | M 1000.1 |
| 01F1h | 497 | Merker | (497 - 0) | / 8 | +1000 | → | M 1062.1 |
| 0400h | 1024 | Eingang | (1024 - 1024) | / 8 | +100 | → | E 100.0 |
| 0401h | 1025 | Eingang | (1025 - 1024) | / 8 | +100 | → | E 100.1 |
| 07DAh | 2010 | Eingang | (2010 - 1024) | / 8 | +100 | → | E 223.2 |

6.4.8.3 FC 03 - Read Output Registers

Diese Funktion ermöglicht das Lesen von Datenworten aus einem Datenbaustein der CPU über den Modbus-Master.

Anforderungstelegramm

| | | | | |
|------|------|----------------|-----------------|-----|
| ADDR | FUNC | start_register | register_number | CRC |
|------|------|----------------|-----------------|-----|

Antworttelegramm

| | | | | |
|------|------|--------------|----------------------------------|-----|
| ADDR | FUNC | byte_count n | n/2-register data (High, Low) | CRC |
|------|------|--------------|----------------------------------|-----|

start_register

Die Modbus-Register-Adresse *start_register* wird vom Treiber wie folgt interpretiert:

| start_register | | | | | | | | | | | | | | | |
|------------------------------|----|----|----|----|----|---|---|-------------------------|---|---|---|---|---|---|---|
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| start_register-offset_DB_no. | | | | | | | | start_register-word_no. | | | | | | | |

- Mit *start_register* geben Sie den DB an, auf den Sie in der CPU zugreifen möchten.
- Die entsprechende DB-Zuordnung der CPU führen Sie über die Eigenschaft "FC 03, 06, 16" in der Parametrierung des CP durch.
 - Hier können Sie der fixen "Modbus-Adresse im Übertragungstelegramm" 0 eine Basis-DB-Nummer im "SIMATIC Speicherbereich" zuordnen.

Umrechnung

- Datenbaustein DB = Basis-DB-Nummer + *start_register-offset_DB_no.*
- Datenwort DBW = *start_register-word_no.* x 2

Ist der DB und das entsprechende Datenwort DBW bekannt, ab dem Sie auf den DB zugreifen möchten, so können Sie nach folgender Formel den Wert für *start_register* berechnen:

■ $start_register = (DB - \text{Basis-DB-Nummer}) \times 512 + (DBW / 2)$



Bitte beachten Sie, dass Sie immer nur geradzahlige Datenwort-Nummern für DBW verwenden dürfen.

register_number

- Als *register_number* ist jeder Wert zwischen 1 und 127 erlaubt.
 - Diese Anzahl von Registern wird gelesen.
- Es gilt: Maximale *register_number* = 512 - *start_register*

Beispiel

Umsetzung Modbus-Adressierung bei FC 03, 06, 16

| | |
|---|---------------------------|
| "Modbus-Adresse im Übertragungstelegramm" | "SIMATIC-Speicherbereich" |
| <i>Param-Start-Adresse</i> | <i>SPS-Bereich</i> |
| von 0 | ab DB 800 |

Umrechnung

Die Umrechnung erfolgt z.B. für *start_register* = 80 (0050h) nach folgender Vorgehensweise:

| start_register = 0050h | | | | | | | | | | | | | | | |
|------------------------------------|----|----|----|----|----|---|---|-------------------------------|---|---|---|---|---|---|---|
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| start_register-offset_DB_no. = 00h | | | | | | | | start_register-word-no. = 50h | | | | | | | |

- Datenbaustein DB = Basis-DB-Nummer + start_register-offset_DB_no.
 - Datenbaustein DB = 800 + 0 = 800
- Datenwort DBW = start_register-word-no. x 2
 - Datenwort DBW = 80 x 2 = 160

Weitere Werte

| start_register | | offset_DB_no. | word_no. | | Basis-DB-Nummer | DB | DBW |
|----------------|---------|---------------|----------|---------|-----------------|---------|---------|
| hex | dezimal | dezimal | hex | dezimal | dezimal | dezimal | dezimal |
| 0000h | 0 | 0 | 000h | 0 | 800 | 800 | 0 |
| 01FAh | 500 | 0 | 1F4h | 500 | 800 | 800 | 1000 |
| 0384h | 900 | 1 | 184h | 388 | 800 | 801 | 776 |
| 03FFh | 1023 | 1 | 1FFh | 511 | 800 | 801 | 1022 |

6.4.8.4 FC 04 - Read Input Registers

- Diese Funktion ist identisch mit FC 03.
- Hier können Sie auf Datenworte eines weiteren Datenbausteins zugreifen. Die entsprechende DB-Zuordnung der CPU führen Sie über die Eigenschaft "FC 04" in der Parametrierung des CP durch. Hier können Sie der fixen "Modbus-Adresse im Übertragungstelegramm" 0 eine Basis-DB- Nummer im "SIMATIC Speicherbereich" zuordnen.
- Näheres hierzu finden Sie unter FC 03 beschrieben. ↪ *Kap. 6.4.8.3 "FC 03 - Read Output Registers" Seite 87*

6.4.8.5 FC 05 - Force Single Coil

Anforderungstelegramm

| | | | | |
|------|------|-----------|-------------|-----|
| ADDR | FUNC | coil_addr | Data_on/off | CRC |
|------|------|-----------|-------------|-----|

Antworttelegramm

| | | | | |
|------|------|-----------|-------------|-----|
| ADDR | FUNC | coil_addr | Data_on/off | CRC |
|------|------|-----------|-------------|-----|

coil_addr

- Die Modbus-Bit-Adresse *coil_addr* beinhaltet den Beginn des Bereichs, auf den Sie in der CPU zugreifen möchten.
- Die entsprechende Adresszuordnung der CPU-Speicherbereiche führen Sie über die Eigenschaft "FC 01, 05, 15" in der Parametrierung des CP durch.
 - Hier können Sie einer "Modbus-Adresse im Übertragungstelegramm" kurz *Param-Start-Adresse* einen "SIMATIC Speicherbereich" kurz SPS-Bereich zuordnen.

Umrechnung

- Byteadresse = $((coil_addr - Param-Start-Adresse) / 8) + SPS-Bereich$
- Beim Zugriff auf Merker bzw. Ausgänge in der SPS wird die verbleibende Restbitnummer berechnet und dazu verwendet, um das entsprechende Bit innerhalb des Merker- bzw. Ausgangsbereichs zu adressieren.
 - Restbitnummer = $(coil_addr - Param-Start-Adresse) \% 8$ [Modulo 8]

Data_on/off

- Als *Data_on/off* sind folgende zwei Werte zulässig:
 - FF00h: Bit setzen
 - 0000h: Bit löschen

Beispiel

Umsetzung Modbus-Adressierung bei FC 01, 05, 15

| | |
|---|---------------------------|
| "Modbus-Adresse im Übertragungstelegramm" | "SIMATIC-Speicherbereich" |
| <i>Param-Start-Adresse</i> | <i>SPS-Bereich</i> |
| von 0 ... 1023 | ab Merker M 1000.0 |
| von 1024 ... 2047 | ab Ausgang A 100.0 |

Adressberechnung:

| start_addr | | Zugriff | Berechnung | | | | Bereich in SPS |
|------------|---------|---------|---------------|-----|-------|---|----------------|
| hex | dezimal | | | | | | |
| 0000h | 0 | Merker | (0 - 0) | / 8 | +1000 | → | M 1000.0 |
| 0001h | 1 | Merker | (1 - 0) | / 8 | +1000 | → | M 1000.1 |
| 01F1h | 497 | Merker | (497 - 0) | / 8 | +1000 | → | M 1062.1 |
| 0400h | 1024 | Ausgang | (1024 - 1024) | / 8 | +100 | → | A 100.0 |
| 0401h | 1025 | Ausgang | (1025 - 1024) | / 8 | +100 | → | A 100.1 |
| 07DAh | 2010 | Ausgang | (2010 - 1024) | / 8 | +100 | → | A 223.2 |

6.4.8.6 FC 06 - Preset Single Register

Die Funktion ermöglicht das Schreiben eines Datenworts in einem Datenbaustein der CPU über den Modbus-Master.



Bitte beachten Sie, dass Sie den Bereich, auf den Sie schreibend zugreifen möchten, über die Protokoll-Parametrierung im Dialogfenster "Grenzen" freigeben.

Anforderungstelegramm

| | | | | |
|------|------|----------------|---------------------------|-----|
| ADDR | FUNC | start_register | Data_value (High, Low) | CRC |
|------|------|----------------|---------------------------|-----|

Antworttelegramm

| | | | | |
|------|------|----------------|---------------------------|-----|
| ADDR | FUNC | start_register | Data_value (High, Low) | CRC |
|------|------|----------------|---------------------------|-----|

start_register

Die Modbus-Register-Adresse *start_register* wird vom Treiber wie folgt interpretiert:

| start_register | | | | | | | | | | | | | | | |
|------------------------------|----|----|----|----|----|---|---|-------------------------|---|---|---|---|---|---|---|
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| start_register-offset_DB_no. | | | | | | | | start_register-word_no. | | | | | | | |

- Mit *start_register* geben Sie den DB an, auf den Sie in der CPU zugreifen möchten.
- Die entsprechende DB-Zuordnung der CPU führen Sie über die Eigenschaft "FC 03, 06, 16" in der Parametrierung des CP durch.
 - Hier können Sie der fixen "Modbus-Adresse im Übertragungstelegramm" 0 eine Basis-DB-Nummer im "SIMATIC Speicherbereich" zuordnen.

Umrechnung

- Datenbaustein DB = Basis-DB-Nummer + start_register-offset_DB_no.
- Datenwort DBW = start_register-word_no. x 2

Ist der DB und das entsprechende Datenwort DBW bekannt, ab dem Sie auf den DB zugreifen möchten, so können Sie nach folgender Formel den Wert für *start_register* berechnen:

- $start_register = (DB - \text{Basis-DB-Nummer}) \times 512 + (DBW / 2)$



Bitte beachten Sie, dass Sie immer nur geradzahlige Datenwort-Nummern für DBW verwenden dürfen.

Data_value

- Als *Data_value* ist jeder 16Bit-Wert erlaubt.
 - Dies ist der Register-Wert, welcher zu schreiben ist.

Beispiel

Umsetzung Modbus-Adressierung bei FC 03, 06, 16

| | |
|---|---------------------------|
| "Modbus-Adresse im Übertragungstelegramm" | "SIMATIC-Speicherbereich" |
| <i>Param-Start-Adresse</i> | <i>SPS-Bereich</i> |
| von 0 | ab DB 800 |

Umrechnung

Die Umrechnung erfolgt z.B. für *start_register* = 80 (0050h) nach folgender Vorgehensweise:

| start_register = 0050h | | | | | | | | | | | | | | | |
|------------------------------------|----|----|----|----|----|---|---|-------------------------------|---|---|---|---|---|---|---|
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| start_register-offset_DB_no. = 00h | | | | | | | | start_register-word-no. = 50h | | | | | | | |

- Datenbaustein DB = Basis-DB-Nummer + start_register-offset_DB_no.
 - Datenbaustein DB = 800 + 0 = 800
- Datenwort DBW = start_register-word-no. x 2
 - Datenwort DBW = 80 x 2 = 160

Weitere Werte

| start_register | | offset_DB_no. | word_no. | | Basis-DB-Nummer | DB | DBW |
|----------------|---------|---------------|----------|---------|-----------------|---------|---------|
| hex | dezimal | dezimal | hex | dezimal | dezimal | dezimal | dezimal |
| 0000h | 0 | 0 | 000h | 0 | 800 | 800 | 0 |
| 01FAh | 500 | 0 | 1F4h | 500 | 800 | 800 | 1000 |
| 0384h | 900 | 1 | 184h | 388 | 800 | 801 | 776 |
| 03FFh | 1023 | 1 | 1FFh | 511 | 800 | 801 | 1022 |

6.4.8.7 FC 08 - Loop Back Diagnostic Test

Diese Funktion dient zur Überprüfung der Kommunikations-Verbindung. Sie hat keinerlei Auswirkung auf das Anwenderprogramm. Das empfangene Telegramm wird vom Treiber selbständig an den Master zurückgesendet.

Anforderungstelegramm

| | | | | |
|------|------|--------------------------------|-----------|-----|
| ADDR | FUNC | diagnostic_code (High, Low) | test_data | CRC |
|------|------|--------------------------------|-----------|-----|

Antworttelegramm

| | | | | |
|------|------|--------------------------------|-----------|-----|
| ADDR | FUNC | diagnostic_code (High, Low) | test_data | CRC |
|------|------|--------------------------------|-----------|-----|

diagnostic_code Die Funktion unterstützt ausschließlich *diagnostic_code* = 0000.

test_data Beliebiger 16Bit Wert.

6.4.8.8 FC 15 - Force Multiple Coils

Diese Funktion ermöglicht das Schreiben mehrerer Bits im Ausgabebereich der CPU über Modbus-Master.



Bitte beachten Sie, dass Sie den Bereich, auf den Sie schreibend zugreifen möchten, über die Protokoll-Parametrierung im Dialogfenster "Grenzen" freigeben.

Anforderungstelegramm

| | | | | | | |
|------|------|------------|----------|--------------|--------|-----|
| ADDR | FUNC | start_addr | quantity | byte_count n | n-Data | CRC |
|------|------|------------|----------|--------------|--------|-----|

Antworttelegramm

| | | | | |
|------|------|------------|----------|-----|
| ADDR | FUNC | start_addr | quantity | CRC |
|------|------|------------|----------|-----|

start_addr

- Die Modbus-Bit-Adresse *start_addr* beinhaltet den Beginn des Bereichs, auf den Sie in der CPU zugreifen möchten.
- Die entsprechende Adresszuordnung der CPU-Speicherbereiche führen Sie über die Eigenschaft "FC 01, 05, 15" in der Parametrierung des CP durch.
 - Hier können Sie einer "Modbus-Adresse im Übertragungstelegramm" kurz *Param-Start-Adresse* einen "SIMATIC Speicherbereich" kurz SPS-Bereich zuordnen.

Umrechnung

- Byteadresse = $((start_addr - Param-Start-Adresse) / 8) + SPS-Bereich$
- Beim Zugriff auf Merker bzw. Ausgänge in der SPS wird die verbleibende Restbitnummer berechnet und dazu verwendet, um das entsprechende Bit innerhalb des Merker- bzw. Ausgangsbereichs zu adressieren.
 - Restbitnummer = $(start_addr - Param-Start-Adresse) \% 8$ [Modulo 8]

quantity

Als *quantity* (Bitanzahl) ist jeder Wert zwischen 1 und 2040 erlaubt.

byte_count n


Der *byte_count n* (Bytezähler) wird automatisch aufgrund der Bitanzahl gebildet.

n-Data

n-Data beinhaltet Bit-Zustände (beliebige Werte).

6.4.8.9 FC 16 - Preset Multiple Registers

Diese Funktion ermöglicht das Schreiben mehrerer Datenworte in einen Datenbaustein der CPU über Modbus-Master.

 Bitte beachten Sie, dass Sie den Bereich, auf den Sie schreibend zugreifen möchten, über die Protokoll-Parametrierung im Dialogfenster "Grenzen" freigeben.

Anforderungstelegramm

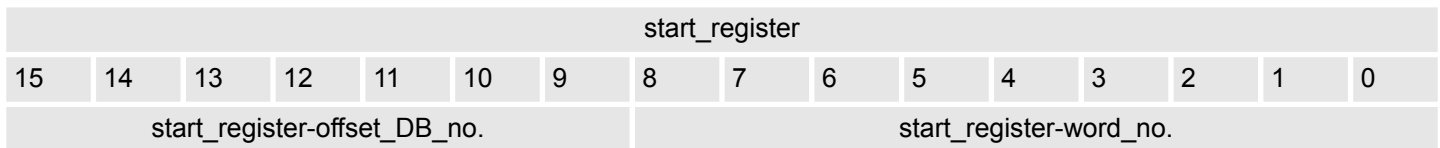
| | | | | | | |
|------|------|----------------|----------|--------------|-----------------------|-----|
| ADDR | FUNC | start_register | quantity | byte_count n | n-Data (High, Low) | CRC |
|------|------|----------------|----------|--------------|-----------------------|-----|

Antworttelegramm

| | | | | |
|------|------|------------|----------|-----|
| ADDR | FUNC | start_addr | quantity | CRC |
|------|------|------------|----------|-----|

start_register

Die Modbus-Register-Adresse *start_register* wird vom Treiber wie folgt interpretiert:




- Die Modbus-Register-Adresse *start_register* beinhaltet den DB und das 1. Datenwort, auf das Sie in der CPU zugreifen möchten.
- Die entsprechende DB-Zuordnung der CPU führen Sie über die Eigenschaft "FC 03, 06, 16" in der Parametrierung des CP durch.
 - Hier können Sie der fixen "Modbus-Adresse im Übertragungstelegramm" 0 eine Basis-DB-Nummer im "SIMATIC Speicherbereich" zuordnen.

Umrechnung

- Datenbaustein DB = *Basis-DB-Nummer* + *start_register-offset_DB_no.*
- Datenwort DBW = *start_register-word_no.* x 2

Ist der DB und das entsprechende Datenwort DBW bekannt, ab dem Sie auf den DB zugreifen möchten, so können Sie nach folgender Formel den Wert für *start_register* berechnen:

■ $start_register = (DB - \text{Basis-DB-Nummer}) \times 512 + (DBW / 2)$

 Bitte beachten Sie, dass Sie immer nur geradzahlige Datenwort-Nummern für DBW verwenden dürfen.

quantity

- Als *quantity* (Register-Anzahl) ist jeder Wert zwischen 1 und 127 erlaubt.
 - Es gilt: Maximale *quantity* = 512 - *start_register*

byte_count n

Der *byte_count n* (Bytezähler) wird automatisch aufgrund der Bitanzahl gebildet.

n-Data (High, Low) Als *n-Data* (High, Low) kann jeder beliebige Wert verwendet werden.

Beispiel Umsetzung Modbus-Adressierung bei FC 03, 06, 16

| | |
|---|---------------------------|
| "Modbus-Adresse im Übertragungstelegramm" | "SIMATIC-Speicherbereich" |
| <i>Param-Start-Adresse</i> | <i>SPS-Bereich</i> |
| von 0 | ab DB 800 |

Umrechnung Die Umrechnung erfolgt z.B. für *start_register* = 80 (0050h) nach folgender Vorgehensweise:

| | | | | | | | | | | | | | | | |
|------------------------------------|----|----|----|----|----|---|---|-------------------------------|---|---|---|---|---|---|---|
| start_register = 0050h | | | | | | | | | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| start_register-offset_DB_no. = 00h | | | | | | | | start_register-word-no. = 50h | | | | | | | |

- Datenbaustein DB = *Basis-DB-Nummer* + *start_register-offset_DB_no.*
 - Datenbaustein DB = 800 + 0 = 800
- Datenwort DBW = *start_register-word-no.* x 2
 - Datenwort DBW = 80 x 2 = 160

Weitere Werte

| start_register | | offset_DB_no. | word_no. | | Basis-DB-Nummer | DB | DBW |
|----------------|---------|---------------|----------|---------|-----------------|---------|---------|
| hex | dezimal | dezimal | hex | dezimal | dezimal | dezimal | dezimal |
| 0000h | 0 | 0 | 000h | 0 | 800 | 800 | 0 |
| 01FAh | 500 | 0 | 1F4h | 500 | 800 | 800 | 1000 |
| 0384h | 900 | 1 | 184h | 388 | 800 | 801 | 776 |
| 03FFh | 1023 | 1 | 1FFh | 511 | 800 | 801 | 1022 |

7 Diagnose und Fehlerverhalten

7.1 Übersicht Diagnosefunktionen

Übersicht

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

- Diagnose über die CP-LEDs
- Diagnose über FB-STATUS
- Diagnose über Diagnosepuffer des CP
- Diagnose über Diagnosealarm

Diagnose über die CP-LEDs

Die CP-LEDs geben Ihnen einen ersten Überblick über aufgetretene interne bzw. externe Fehler sowie schnittstellenspezifische Fehler. Nähere Informationen zu den LEDs und deren Funktion finden Sie unter "Hardwarebeschreibung" und unter "Firmwareupdate".

Diagnose über STATUS der FBs

- Für die Fehlerdiagnose besitzen die Bausteine FB 7 - P_RCV_RK und FB 8 - P_SND_RK den Parameter STATUS.
 - Durch Zugriff auf STATUS erhalten Sie Aussagen zu Fehlern, die bei der Kommunikation aufgetreten sind.
 - Den STATUS-Ausgang können Sie im Anwenderprogramm auswerten.
 - Die Diagnoseereignisse an STATUS trägt der CP auch in seinen Diagnosepuffer ein.

Diagnose über Diagnosepuffer des CP

Alle Fehler des CP werden im Diagnosepuffer des CP eingetragen. Wie auch bei der CPU können Sie über die Zielsystemfunktionen auf den Diagnosepuffer des CP zugreifen.



Eine Fehlermeldung wird nur dann ausgegeben, wenn gleichzeitig das Bit ERROR (Auftragsende mit Fehler) gesetzt ist. Ansonsten ist STATUS "0".

Diagnose über Diagnosealarm

- Der CP kann auf der ihm zugeordneten CPU einen Diagnosealarm auslösen. Hierbei stellt der CP 4Byte Diagnoseinformationen der CPU zur Verfügung.
 - Auf diese Daten haben Sie Zugriff, indem Sie den Diagnosepuffer der CPU auslesen bzw. mit dem OB 82 auf die Diagnose reagieren.
- Die Diagnosen werden auch im Diagnosepuffer des CP eingetragen.
- Beim Auftreten eines Diagnosealarms leuchtet die rote SF-LED.

7.2 Diagnose über FB STATUS

Übersicht

- Für die Fehlerdiagnose besitzen die Funktionsbausteine FB 7 - P_RCV_RK und FB 8 - P_SND_RK den Parameter STATUS.
- Jede Meldung von STATUS hat unabhängig vom verwendeten Funktionsbaustein die gleiche Bedeutung.

Das STATUS-Wort hat folgende Struktur:

STATUS



Ereignisklassen und -Nummern

Nachfolgend sind in tabellarischer Form die Ereignisklassen mit zugehörigen Ereignisnummern aufgeführt:

| Ereignisklasse 00h "Anlauf des CP" | |
|------------------------------------|--|
| Ereignisklasse / Nr. | Beschreibung |
| 00 03h | PtP-Parameter übernommen |
| 00 04h | Parameter schon auf CP (Zeitstände gleich) |
| 00 07h | Zustandsübergang CPU in STOP |
| 00 08h | Zustandsübergang CPU in RUN/ANLAUF |

| Ereignisklasse 01h "Hardwarefehler auf dem CP" | |
|--|---|
| Ereignisklasse / Nr. | Beschreibung |
| 01 01h | Fehler beim Test des Betriebssystem-EPROM des CP <i>Abhilfe: CP ist defekt und muss getauscht werden.</i> |
| 01 02h | RAM-Test des CP fehlerhaft <i>Abhilfe: CP ist defekt und muss getauscht werden.</i> |
| 01 03h | Auftragsschnittstelle des CP defekt <i>Abhilfe: CP ist defekt und muss getauscht werden.</i> |
| 01 10h | Fehler in der CP-Firmware <i>Abhilfe: Schalten Sie den CP aus und wieder ein, ggf. ist der CP zu tauschen.</i> |

| Ereignisklasse 02h "Fehler bei der Initialisierung" | |
|---|--|
| Ereignisklasse / Nr. | Beschreibung |
| 02 0Fh | Beim Start der parametrisierten Kommunikation wird eine unzulässige Parametrierung festgestellt. Die Schnittstelle konnte nicht parametrisiert werden. Bitte beachten Sie, dass RK512 vom Yaskawa-CP nicht unterstützt wird. Bei parametrierter RK512 erhalten Sie diese Fehlermeldung. <i>Abhilfe: Verwenden Sie kein RK512. Korrigieren Sie die unzulässige Parametrierung und führen Sie einen Neustart durch.</i> |

Diagnose über FB STATUS

Ereignisklasse 03h "Fehler bei der Parametrierung der FBs" (wird nicht im Diagnosepuffer angezeigt)

| Ereignisklasse / Nr. | Beschreibung |
|----------------------|---|
| 03 01h | <p>Quell-/Zielfadentyp nicht zulässig bzw. nicht vorhanden Bereich (Anfangsadresse, Länge) nicht zulässig DB nicht vorhanden bzw. nicht zulässig(z.B. DB 0) oder anderer Datentyp nicht vorhanden bzw. nicht zulässig.</p> <p><i>Abhilfe: Parametrierung auf CPU und CP prüfen und evtl. korrigieren.</i></p> |

Ereignisklasse 04h "Vom CP erkannte Fehler im Datenverkehr CP - CPU"

| Ereignisklasse / Nr. | Beschreibung |
|----------------------|---|
| 04 03h | <p>Fehlerhafter, unbekannter oder nicht erlaubter Datentyp (z.B. falsche Parametrierung des FB)</p> <p><i>Abhilfe: Programm überprüfen, z.B. falsche Parametrierung des FBs.</i></p> |
| 04 07h | <p>Fehler bei der Datenübertragung zwischen CPU und CP.</p> <p><i>Abhilfe: Wird dieser Fehler ständig gemeldet, müssen Sie überprüfen, ob die von Ihnen im Anwenderprogramm aufgerufenen FBs richtig parametrier sind.</i></p> <p><i>Wird der Fehler unmittelbar nach NetzEIN gemeldet, ist zu diesem Zeitpunkt noch keine Verbindung zur CPU aufgebaut. Bei 3964(R) und beim ASCII-Treiber wird die Datenübertragung vom empfangenden CP wiederholt, bis die Daten zur CPU hin übertragen sind.</i></p> <p><i>Erfolgt die Meldung sporadisch während der laufenden Datenübertragung, kann die CPU zeitweise die Daten nicht übernehmen. Bei 3964(R) und beim ASCII-Treiber wird die Datenübertragung dann vom empfangenden CP wiederholt, bis die Daten zur CPU hin übertragen sind.</i></p> |
| 04 08h | <p>Fehler bei der Datenübertragung zwischen CPU und CP (Empfang).</p> <ul style="list-style-type: none"> ■ CPU ist temporär überlastet, Auftrag wird wiederholt. <i>Abhilfe: Anzahl der Kommunikations-Aufrufe reduzieren.</i> ■ Auf Datenbereich der CPU kann nicht zugegriffen werden, z.B. weil Empfangsbaustein zu selten aufgerufen wird. <i>Abhilfe: Empfangsbaustein häufiger aufrufen.</i> ■ Auf Datenbereich der CPU kann temporär nicht zugegriffen werden, z.B. weil Empfangsbaustein zwischendurch gesperrt wurde (EN = false). <i>Abhilfe: Überprüfen, ob Empfangsbaustein zu lange gesperrt wird.</i> |
| 04 09h | <p>Empfang von Daten nicht möglich. Fehler bei der Datenübertragung zwischen CPU und CP (Empfang). Nach mehrmaligen Versuchen wurde Auftrag nach 10s abgebrochen:</p> <ul style="list-style-type: none"> ■ Empfangsbaustein wurde nicht aufgerufen. <i>Abhilfe: Überprüfen Sie ihr Anwenderprogramm, ob der Empfangsbaustein durchlaufen wird.</i> ■ Empfangsbaustein ist gesperrt. <i>Abhilfe: Überprüfen Sie, ob der Empfangsbaustein gesperrt ist.</i> ■ Auf Datenbereich der CPU kann nicht zugegriffen werden <i>Abhilfe: Überprüfen Sie, ob der Datenbereich, in den die Daten übertragen werden sollten, vorhanden ist.</i> ■ Datenbereich der CPU ist zu kurz. <i>Abhilfe: Überprüfen Sie die Länge des Datenbereichs.</i> |
| 04 0Ah | <p>Fehler bei der Datenübertragung zwischen CPU und CP.</p> <p>Die Datenübertragung wurde durch RESET abgebrochen, weil:</p> <ul style="list-style-type: none"> ■ Ziel-DB ist nicht vorhanden ■ Ziel-DB ist zu kurz ■ RESET-Bit am FB ist gesetzt. <p><i>Abhilfe: Ziel-DB im Anwenderprogramm erzeugen bzw. vorhandenen Ziel-DB verlängern.</i></p> |

Ereignisklasse 05h "Fehler bei Bearbeitung eines CPU-Auftrags"

| Ereignisklasse / Nr. | Beschreibung |
|----------------------|---|
| 05 01h | <p>Laufender Auftrag wurde durch Neuanlauf des CP abgebrochen.</p> <p><i>Abhilfe: Bei NetzEIN ist keine Hilfe möglich. Beim Umparametrieren des CP vom PG aus sollten Sie vor dem Beschreiben einer Schnittstelle darauf achten, dass von der CPU aus keine Aufträge mehr laufen.</i></p> |
| 05 02h | <p>Auftrag ist in diesem Betriebszustand des CP nicht erlaubt (z.B. Geräteschnittstelle nicht parametrierbar).</p> <p><i>Abhilfe: Parametrieren Sie die Geräteschnittstelle.</i></p> |
| 05 14h | <p>Anfangsadressen zu hoch angegeben für gewünschten Datentyp oder Anfangsadresse bzw. DB/DX-Nr. zu niedrig.</p> <p><i>Abhilfe: Entnehmen Sie den Auftrags Tabellen die zulässigen Anfangsadressen und DB/DX-Nummern, die im Programm angegeben werden dürfen.</i></p> |
| 05 17h | <p>Übertragungslänge > 1kByte ist für CP zu groß oder Länge für Schnittstellenparameter zu klein.</p> <p><i>Abhilfe: Spalten Sie den Auftrag auf mehrere Aufträge mit kleinerer Länge.</i></p> |
| 05 18h | <p>Nur bei Modbus Master</p> <p>Übertragungslänge beim Senden ist zu groß (> 4kByte) oder Übertragungslänge beim Empfang ist zu klein.</p> <p><i>Abhilfe: Parameter LEN am SEND überprüfen.</i></p> |

Ereignisklasse 07h "Sendefehler"

| Ereignisklasse / Nr. | Beschreibung |
|----------------------|---|
| 07 01h | <p>Senden der ersten Wiederholung:</p> <ul style="list-style-type: none"> ■ Beim Senden des Telegramms wurde ein Fehler erkannt. ■ Der Partner forderte durch ein negatives Quittungszeichen (NAK) eine Wiederholung an. <p><i>Abhilfe: Eine Wiederholung ist kein Fehler, jedoch kann sie ein Hinweis sein, dass Störungen auf der Übertragungsleitung auftreten oder ein Fehlverhalten des Partnergerätes vorliegt. Wenn nach der max. Wiederholungsanzahl des Telegramm immer noch nicht übertragen werden konnte, wird eine Fehler-Nr. gemeldet, die den Fehler beschreibt, der zuerst auftrat.</i></p> |
| 07 02h | <p>Nur bei 3964(R)</p> <p>Fehler beim Verbindungsaufbau: nachdem STX gesendet wurde, wurde NAK oder ein beliebiges Zeichen (außer DLE oder STX) empfangen.</p> <p><i>Abhilfe: Fehlverhalten des Partnergerätes ggf. mit Schnittstellentestgerät untersuchen, das in die Übertragungsleitung eingeschaltet wird.</i></p> |
| 07 03h | <p>Nur bei 3964(R)</p> <p>Quittungsverzögerzeit (QVZ) überschritten: Nach Senden von STX kam innerhalb der Quittungsverzögerzeit keine Antwort vom Partner.</p> <p><i>Abhilfe: Partnergerät ist zu langsam oder nicht empfangsbereit, oder es liegt z.B. ein Bruch der Sendeleitung vor. Fehlverhalten des Partnergerätes ggf. mit Schnittstellentestgerät nachweisen, das in die Übertragungsleitung eingeschaltet wird.</i></p> |

| Ereignisklasse 07h "Sendefehler" | |
|----------------------------------|---|
| Ereignisklasse / Nr. | Beschreibung |
| 07 04h | <p>Nur bei 3964(R)</p> <p>Abbruch durch Partner: während des laufenden Sendebetriebs wurden vom Partner ein oder mehrere Zeichen empfangen.</p> <p><i>Abhilfe: Prüfen Sie, ob der Partner ebenfalls Fehler anzeigt, da evtl. nicht alle Sendedaten angekommen sind (z.B. Bruch in der Sendeleitung) oder schwere Störung vorliegen, oder es liegt ein Fehlverhalten des Partnergerätes vor. Dies ggf. mit Schnittstellentestgerät nachweisen, das in die Übertragungsleitung eingeschaltet wird.</i></p> |
| 07 06h | <p>Nur bei 3964(R)</p> <p>Fehler bei Verbindungsende:</p> <ul style="list-style-type: none"> ■ Das Telegramm wurde vom Partner am Ende mit NAK oder einem beliebigen Zeichen (außer DLE) abgelehnt. ■ Das Quittungszeichen (DLE) wurde zu früh empfangen. <p><i>Abhilfe: Prüfen Sie, ob der Partner ebenfalls Fehler anzeigt, da evtl. nicht alle Sendedaten angekommen sind (z.B. Bruch in der Sendeleitung) oder schwere Störung vorliegen, oder es liegt ein Fehlverhalten des Partnergerätes vor. Dies ggf. mit Schnittstellentestgerät nachweisen, das in die Übertragungsleitung eingeschaltet wird.</i></p> |
| 07 07h | <p>Nur bei 3964(R)</p> <p>Quittungsverzugszeit am Verbindungsende/Antwortüberwachungszeit nach Sendetelegramm überschritten.</p> <p>Nach Verbindungsabbau mit DLE ETX kam innerhalb der QVZ keine Antwort vom Partner.</p> <p><i>Abhilfe: Partnergerät ist zu langsam oder gestört. Dies ggf. mit Schnittstellentestgerät nachweisen, das in die Übertragungsleitung eingeschaltet wird.</i></p> |
| 07 08h | <p>Nur bei ASCII-Treiber</p> <p>Die Wartezeit auf XON bzw. CTS = ON ist abgelaufen.</p> <p><i>Abhilfe: Der Kommunikationspartner ist gestört, zu langsam oder offline geschaltet. Überprüfen Sie den Kommunikationspartner oder ändern Sie ggf. die Parametrierung.</i></p> |
| 07 09h | <p>Kein Verbindungsaufbau möglich, die Anzahl der erlaubten Aufbauversuche wurde überschritten.</p> <p><i>Abhilfe: Überprüfen Sie das Schnittstellenkabel oder die Übertragungsparameter. Überprüfen Sie auch beim Partner, ob die Empfangsfunktion zwischen CPU und CP richtig parametrier ist.</i></p> |
| 07 0Ah | <p>Die Daten konnten nicht übertragen werden, die erlaubte Anzahl der Übertragungsversuche wurde überschritten.</p> <p><i>Abhilfe: Überprüfen Sie das Schnittstellenkabel oder die Übertragungsparameter.</i></p> |

| Ereignisklasse 08h "Empfangsfehler" | |
|-------------------------------------|---|
| Ereignisklasse / Nr. | Beschreibung |
| 08 01h | <p>Erwarten der ersten Wiederholung:</p> <p>Beim Empfangen eines Telegramms wurde ein Fehler erkannt und der CP forderte durch eine negative Quittierung (NAK) beim Partner eine Wiederholung an.</p> <p><i>Abhilfe: Eine Wiederholung ist kein Fehler, jedoch kann Sie ein Hinweis sein, dass Störungen auf der Übertragungsleitung auftreten oder ein Fehlverhalten des Partnergerätes vorliegt. Wenn nach der max. Wiederholungsanzahl das Telegramm immer noch nicht übertragen werden konnte, wird eine Fehler-Nr. gemeldet, die den Fehler beschreibt, der zuerst auftrat.</i></p> |
| 08 02h | <p>Nur bei 3964(R)</p> <p>Fehler beim Verbindungsaufbau:</p> <ul style="list-style-type: none"> ■ In Ruhestellung wurden ein oder mehrere beliebige Zeichen (außer NAK oder STX) empfangen. ■ Nach einem empfangenen STX wurden vom Partner weitere Zeichen gesendet, ohne die Antwort DLE abzuwarten. <p>Nach NetzEIN des Partners:</p> <ul style="list-style-type: none"> ■ Während der Partner eingeschaltet wird, empfängt der CP ein undefiniertes Zeichen. <p><i>Abhilfe: Fehlverhaltendes Partnergerät ggf. mit Schnittstellentestgerät nachweisen, das in die Übertragungsleitung eingeschaltet wird.</i></p> |
| 08 05h | <p>Nur bei 3964(R)</p> <p>Logischer Fehler während des Empfangs:</p> <p>Nach Empfang von DLE wurde ein weiteres beliebiges Zeichen empfangen (außer DLE, ETX).</p> <p><i>Abhilfe: Prüfen Sie, ob der Partner DLE im Telegrammkopf und im Datenstring immer verdoppelt bzw. der Verbindungsabbau mit DLE ETX vorgenommen wird. Fehlverhaltendes Partnergerätes ggf. mit Schnittstellentestgerät nachweisen, das in die Übertragungsleitung eingeschaltet wird.</i></p> |
| 08 06h | <p>Zeichenverzugszeit (ZVZ) überschritten:</p> <ul style="list-style-type: none"> ■ Zwei aufeinanderfolgende Zeichen wurden nicht innerhalb der ZVZ empfangen oder <p>Nur bei 3964(R)</p> <ul style="list-style-type: none"> ■ 1. Zeichen nach dem Senden von DLE beim Verbindungsaufbau wurde nicht innerhalb der ZVZ empfangen. <p><i>Abhilfe: Partnergerät ist zu langsam oder gestört. Dies mit Schnittstellentestgerät nachweisen, das in die Übertragungsleitung eingeschaltet wird.</i></p> |
| 08 08h | <p>Nur bei 3964(R)</p> <p>Fehler beim Blockprüfzeichen BCC:</p> <p>Der intern gebildete Wert des BCC stimmt nicht mit dem vom Partner am Verbindungsende empfangenen BCC überein.</p> <p><i>Abhilfe: Prüfen Sie, ob die Verbindung stark gestört ist, in diesem Fall werden auch gelegentlich Fehlercodes zu beobachten sein. Fehlverhaltendes Partnergerätes ggf. mit Schnittstellentestgerät nachweisen, das in die Übertragungsleitung eingeschaltet wird.</i></p> |
| 08 0Ah | <p>Ein freier Empfangspuffer ist nicht vorhanden.</p> <p><i>Abhilfe: Der FB P_RCV_RK muss häufiger aufgerufen werden.</i></p> |

Ereignisklasse 08h "Empfangsfehler"

| Ereignisklasse / Nr. | Beschreibung |
|----------------------|--|
| 08 0Ch | <p>Übertragungsfehler:</p> <ul style="list-style-type: none"> Ein Übertragungsfehler (Paritäts-, Stoppbit- oder Überlauffehler) wurde erkannt. <p>Nur bei 3964(R)</p> <ul style="list-style-type: none"> Wird in Ruhestellung ein gestörtes Zeichen empfangen, wird der Fehler sofort gemeldet, damit Störeinflüsse auf der Übertragungsleitung frühzeitig erkannt werden können. Falls dies während des Sende- oder Empfangsbetriebes auftritt, werden Wiederholungen gestartet. <p><i>Abhilfe: Störungen auf der Übertragungsleitung verursachen Telegrammwiederholungen und erniedrigen dadurch den Nutzdatendurchsatz. Die Gefahr eines nicht erkannten Fehlers steigt. Ändern Sie ihren Systemaufbau bzw. die Leitungsverlegung. Überprüfen Sie die Verbindungsleitung der Kommunikationspartner bzw. überprüfen Sie, ob bei beiden Geräten Baudrate, Parität und Stoppbitanzahl gleich eingestellt sind.</i></p> |
| 08 0Dh | <p>BREAK</p> <p>Empfangsleitung zum Partner ist unterbrochen.</p> <p><i>Abhilfe: Stellen Sie die Verbindung wieder her oder schalten Sie den Partner ein.</i></p> |
| 08 15h | <p>Einstellung der Übertragungsversuche des CP und des Kommunikationspartners stimmen nicht überein.</p> <p><i>Abhilfe: Parametrieren Sie beim Partner die gleiche Anzahl der Übertragungsversuche wie am CP. Fehlverhaltendes Partnergerät ggf. mit Schnittstellentestgerät nachweisen, das in die Übertragungsleitung eingeschaltet wird.</i></p> |
| 08 16h | <ul style="list-style-type: none"> Die Länge eines empfangenen Telegramms war länger als die maximal vereinbarte Länge. <p><i>Abhilfe: Korrektur beim Partner erforderlich.</i></p> <ul style="list-style-type: none"> Die Länge des parametrierten Empfangsfachs ist zu kurz. <p><i>Abhilfe: Länge des Empfangsfachs vergrößern.</i></p> |
| 08 18h | <p>Nur bei (Modbus) ASCII-Treiber</p> <p>DSR = OFF bzw. CTS = OFF</p> <p><i>Abhilfe: Vor oder während eines Sendevorgangs sind die Signale DSR bzw. CTS vom Partner auf "OFF" geschaltet worden. Überprüfen Sie die Steuerung der RS 232-Begleitsignale beim Partner.</i></p> |
| 08 30h | <p>Nur bei Modbus Master</p> <p>Die Antwortüberwachungszeit nach Senden eines Anforderungstelegramms ist abgelaufen ohne dass der Beginn des Antworttelegramms erkannt wurde.</p> <p><i>Abhilfe: Prüfen Sie, ob die Übertragungsleitung gestört ist (ggf. Schnittstellen-Analyzer verwenden).</i></p> <p><i>Prüfen Sie, ob die Protokollparameter Baudrate, Datenbitanzahl, Parität, Stoppbitanzahl bei CP und Koppelpartner gleich eingestellt sind.</i></p> <p><i>Prüfen Sie, ob die mit PtP_PARAM parametrierte Antwortüberwachungszeit groß genug eingestellt ist.</i></p> <p><i>Prüfen Sie, ob die angegebene Slave-Adresse vorhanden ist.</i></p> |
| 08 31h | <p>Nur bei Modbus Master RTU</p> <p>Das erste Zeichen des Antworttelegramms vom Slave ist ungleich der im Anforderungstelegramm gesendeten Slave-Adresse (bei Betriebsart "normal").</p> <p><i>Abhilfe: Ein falscher Slave hat geantwortet.</i></p> <p><i>Prüfen Sie, ob die Übertragungsleitung gestört ist (ggf. Schnittstellen-Analyzer verwenden).</i></p> |

| Ereignisklasse 08h "Empfangsfehler" | |
|-------------------------------------|--|
| Ereignisklasse / Nr. | Beschreibung |
| 08 32h | <p>Nur bei Modbus Master</p> <p>Überlauf des Empfangspuffers im CP bei Empfang des Antworttelegramms.</p> <p><i>Abhilfe: Überprüfen Sie die Protokolleinstellungen beim Slave.</i></p> |
| 08 33h | <p>Nur bei Modbus Master ASCII</p> <p>Ein falsches Startzeichen wurde empfangen. Es war kein ":" (3Ah).</p> <p><i>Abhilfe: Überprüfen Sie die Protokolleinstellungen beim Slave.</i></p> |
| 08 34h | <p>Nur bei Modbus Master ASCII</p> <p>Ein Startzeichen wurde innerhalb eines Telegramms empfangen. Der 1. Teil des Telegramms wurde verworfen und der Empfang mit dem 2. Startzeichen neu gestartet.</p> <p><i>Abhilfe: Überprüfen Sie, ob die Übertragungsleitung eine Unterbrechung hat. Bei Unterbrechungen wird ein Sendeauftrag nicht unterbrochen. Ein Fehler erscheint nur im Diagnosepuffer des CP.</i></p> |

| Ereignisklasse 14 (0Eh) "Ladbarer Treiber - Allgemeine Verarbeitungsfehler <Bearbeitung BSEND-Auftrag>" | |
|---|--|
| Ereignisklasse / Nr. | Beschreibung |
| 0E 31h | <p>Nur bei Modbus Slave</p> <p>TimeOut bei Übergabe von Daten an die CPU.</p> <p><i>Abhilfe: Schnittstelle CP-CPU überprüfen.</i></p> |
| 0E 38h | <p>Nur bei Modbus Slave</p> <p>Beim Zugriff auf einen der CPU-Bereiche "Merker", "Ausgang", "Zeiten", "Zähler", "Eingang" mit dem Funktionscode FC 01 oder FC 02 ist ein Fehler aufgetreten: z.B. Eingang nicht vorhanden oder Leseversuch über das Bereichsende hinaus.</p> <p><i>Abhilfe: Prüfen Sie, ob der angesprochene CPU-Bereich vorhanden ist und ob nicht versucht wurde, über das Bereichsende hinaus zuzugreifen.</i></p> |
| 0E 39h | <p>Nur bei Modbus Slave</p> <p>Beim Zugriff auf den CPU-Bereich "Datenbaustein" mit dem Funktionscode FC 03, 04, 06, 16 ist ein Fehler aufgetreten: Datenbaustein nicht vorhanden oder zu kurz.</p> <p><i>Abhilfe: Prüfen Sie, ob der angesprochene Datenbaustein vorhanden und ausreichend lang ist.</i></p> |
| 0E 40h | <p>Nur bei Modbus Master</p> <p>Beim Parameter LEN am SFB SEND wurde ein zu kleiner Wert angegeben.</p> <p><i>Abhilfe: Die Mindestlänge ist 2Byte.</i></p> |
| 0E 41h | <p>Nur bei Modbus Master</p> <p>Beim Parameter LEN am SFB SEND wurde ein zu kleiner Wert angegeben. Bei dem übergebenen Funktionscode ist eine größere Länge nötig.</p> <p><i>Abhilfe: Die Mindestlänge ist bei diesem Funktionscode 6Byte.</i></p> |
| 0E 42h | <p>Nur bei Modbus Master</p> <p>Der übergebene Funktionscode ist nicht zulässig.</p> <p><i>Abhilfe: Nur die im Kapitel "Funktionscodes" aufgelisteten Funktionscodes sind zulässig.</i></p> <p>🔗 <i>Kap. 6.4.4 "Modbus Master - Funktionscodes" Seite 62</i></p> |

| Ereignisklasse 14 (0Eh) "Ladbarer Treiber - Allgemeine Verarbeitungsfehler <Bearbeitung BSEND-Auftrag>" | |
|---|--|
| Ereignisklasse / Nr. | Beschreibung |
| 0E 43h | <p>Nur bei Modbus Master</p> <p>Die Slave-Adresse 0 (=Broadcast) ist bei diesem FC nicht zulässig.</p> <p><i>Abhilfe: Verwenden Sie Slave-Adresse 0 nur bei den hierfür geeigneten Funktionscodes.</i></p> |
| 0E 44h | <p>Nur bei Modbus Master</p> <p>Der Wert des übergebenen Parameters "Bit-Anzahl" ist nicht im Bereich 1...2040 (Modbus Master ASCII: 1...2008).</p> <p><i>Abhilfe: Korrigieren Sie Ihren Quell-DB.</i></p> |
| 0E 45h | <p>Nur bei Modbus Master</p> <p>Der Wert des übergebenen Parameters "Register-Anzahl" ist nicht im Bereich 1...127 (Modbus Master ASCII 1...125, mit 32Bit 1...62).</p> <p><i>Abhilfe: Korrigieren Sie Ihren Quell-DB.</i></p> |
| 0E 46h | <p>Nur bei Modbus Master</p> <p>Funktionscode 15 oder 16: Der Wert des übergebenen Parameters "Bit-Anzahl" bzw. "Register-Anzahl" ist nicht im Bereich 1...2040 bzw. 1...127 (Modbus Master ASCII 1...1976 bzw. 1...123, mit 32Bit 1...61).</p> <p><i>Abhilfe: Korrigieren Sie Ihren Quell-DB.</i></p> |
| 0E 47h | <p>Nur bei Modbus Master</p> <p>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.</p> <p><i>Abhilfe: 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.</i></p> |
| 0E 48h | <p>Nur bei Modbus Master</p> <p>Funktionscode 05: Die im SEND-Quell_DB angegebene Codierung für "Bit-Setzen" (FF00h) oder "Bit-Löschen" (0000h) ist falsch.</p> <p><i>Abhilfe: Es sind nur die Codierungen FF00h und 0000h erlaubt.</i></p> |
| 0E 49h | <p>Nur bei Modbus Master</p> <p>Funktionscode 08: Der im SEND-Quell_DB angegebene "Diagnostic-Code" ist falsch.</p> <p><i>Abhilfe: Es ist nur der "Diagnostic Code" 0000h erlaubt.</i></p> |
| 0E 4Ah | <p>Nur bei Modbus Master ASCII</p> <p>Zugriff auf 32Bit-Register ist nur mit FC 03, 06, 16 möglich.</p> <p>Hier ist das Bit 6 im FC des DB gesetzt.</p> <p><i>Abhilfe: Korrigieren Sie Ihren Quell-DB.</i></p> |
| 0E 4Fh | <p>Nur bei Modbus Master</p> <p>Die am SFB SEND RK angegebene R_TYP ist bei diesem Treiber unzulässig.</p> <p><i>Abhilfe: Als R_TYP muss "X" eingetragen werden.</i></p> |

| Ereignisklasse 14 (0Eh) "Ladbarer Treiber - Allgemeine Verarbeitungsfehler <Bearbeitung BSEND-Auftrag>" | |
|---|--|
| Ereignisklasse / Nr. | Beschreibung |
| 0E 50h | <p>Nur bei Modbus Master</p> <p>Slave-Adresse falsch: Die empfangene Slave-Adresse ist ungleich der gesendeten Slave-Adresse.</p> <p><i>Abhilfe: Ein falscher Slave hat geantwortet. Prüfen Sie, ob die Übertragungsleitung gestört ist (ggf. Schnittstellen-Analyzer verwenden).</i></p> |
| 0E 51h | <p>Nur bei Modbus Master</p> <p>Funktionscode falsch: Der im Antworttelegramm empfangene Funktionscode ist ungleich dem gesendeten Funktionscode.</p> <p><i>Abhilfe: Überprüfen Sie das Slavegerät.</i></p> |
| 0E 52h | <p>Nur bei Modbus Master</p> <p>Byte-Underflow: Es wurden weniger Zeichen empfangen als sich durch den Bytezähler des Antworttelegramms ergeben würden, bzw. als bei diesem Funktionscode erwartet werden.</p> <p><i>Abhilfe: Überprüfen Sie das Slavegerät.</i></p> |
| 0E 53h | <p>Nur bei Modbus Master</p> <p>Es wurden mehr Zeichen empfangen als sich durch den Bytezähler des Antworttelegramms ergeben würden, bzw. als bei diesem Funktionscode erwartet werden.</p> <p><i>Abhilfe: Überprüfen Sie das Slavegerät.</i></p> |
| 0E 54h | <p>Nur bei Modbus Master</p> <p>Bytezähler zu klein: Der im Antworttelegramm empfangene Bytezähler ist zu klein.</p> <p><i>Abhilfe: Überprüfen Sie das Slavegerät.</i></p> |
| 0E 55h | <p>Nur bei Modbus Master</p> <p>Bytezähler falsch: Der im Antworttelegramm empfangene Bytezähler ist falsch.</p> <p><i>Abhilfe: Überprüfen Sie das Slavegerät.</i></p> |
| 0E 56h | <p>Nur bei Modbus Master</p> <p>Echo falsch: Die vom Slave zurückgesendeten Daten des Antworttelegramms (Bit- Anzahl, ...) sind ungleich der Daten im Anforderungstelegramm.</p> <p><i>Abhilfe: Überprüfen Sie das Slavegerät.</i></p> |
| 0E 57h | <p>Nur bei Modbus Master</p> <p>CRC-Check fehlerhaft (Modbus Master ASCII: LRC check fehlerhaft): Bei der Überprüfung der CRC-(LRC)-Checksumme des Antworttelegramms vom Slave wurde ein Fehler festgestellt.</p> <p><i>Abhilfe: Überprüfen Sie das Slavegerät.</i></p> |
| 0E 58h | <p>Nur bei Modbus Master ASCII</p> <p>Ein empfangenes Zeichen innerhalb des Telegramms ist kein ASCII- Zeichen (0...9, A...F).</p> <p><i>Abhilfe: Überprüfen Sie das Slavegerät. Stellen Sie sicher, dass sich der Slave im ASCII- und nicht im RTU-Mode befindet.</i></p> |
| 0E 61h | <p>Nur bei Modbus Master</p> <p>Antworttelegramm mit Exception Code 01: Illegal Function</p> <p><i>Abhilfe: Siehe Handbuch des Slavegerätes.</i></p> |

Diagnose über FB STATUS

Ereignisklasse 14 (0Eh) "Ladbarer Treiber - Allgemeine Verarbeitungsfehler <Bearbeitung BSEND-Auftrag>"

| Ereignisklasse / Nr. | Beschreibung |
|----------------------|--|
| 0E 62h | Nur bei Modbus Master Antworttelegramm mit Exception Code 02: Illegal Data Address <i>Abhilfe: Siehe Handbuch des Slavegerätes.</i> |
| 0E 63h | Nur bei Modbus Master Antworttelegramm mit Exception Code 03: Illegal Data Value <i>Abhilfe: Siehe Handbuch des Slavegerätes.</i> |
| 0E 64h | Nur bei Modbus Master Antworttelegramm mit Exception Code 04: Failure in associated device <i>Abhilfe: Siehe Handbuch des Slavegerätes.</i> |
| 0E 65h | Nur bei Modbus Master Antworttelegramm mit Exception Code 05: Acknowledge <i>Abhilfe: Siehe Handbuch des Slavegerätes.</i> |
| 0E 66h | Nur bei Modbus Master Antworttelegramm mit Exception Code 06: Busy, Rejected message <i>Abhilfe: Siehe Handbuch des Slavegerätes.</i> |
| 0E 67h | Nur bei Modbus Master Antworttelegramm mit Exception Code 07: Negative Acknowledgment <i>Abhilfe: Siehe Handbuch des Slavegerätes.</i> |

Ereignisklasse 30 (1Eh) "Kommunikationsfehler zwischen CP und CPU über Rückwandbus"

| Ereignisklasse / Nr. | Beschreibung |
|----------------------|--|
| 1E 0Dh | Auftragsabbruch wegen Neustart, Wiederanlauf oder Reset. |
| 1E 0Eh | Statischer Fehler beim Aufruf des SFC 59 "RD-REC" (Datensatz lesen). Der Returnwert RET_VAL des SFCs wird Ihnen in der Variablen SFCERR im Instanz-DB zur Auswertung zur Verfügung gestellt. <i>Abhilfe: Laden Sie die Variable SFCERR aus dem Instanz-DB.</i> |
| 1E 0Fh | Statischer Fehler beim Aufruf des SFC 58 "WD-REC" (Datensatz schreiben). Der Returnwert RET_VAL des SFCs wird Ihnen in der Variablen SFCERR im Instanz-DB zur Auswertung zur Verfügung gestellt. <i>Abhilfe: Laden Sie die Variable SFCERR aus dem Instanz-DB.</i> |
| 1E 41h | Anzahl der am Parameter LEN der FBs angegebenen Bytes unzulässig. <i>Abhilfe: Halten Sie den Wertebereich von 1 ... 1024Bytes ein.</i> |

7.3 Diagnose über Diagnosepuffer

Übersicht

Der CP besitzt einen eigenen Diagnosepuffer. Hier werden alle Diagnoseeinträge des CP in der Reihenfolge ihres Auftretens eingetragen.

Folgende Fehler können angezeigt werden:

- Fehler in Hardware bzw. Firmware
- Fehler bei der Initialisierung und Parametrierung
- Fehler bei der Ausführung eines CPU-Auftrags
- Fehler bei der Datenübertragung (Sende- und Empfangsfehler)



- *Der Diagnosepuffer ist als Ringpuffer für maximal 9 Diagnoseeinträge aufgebaut.*
- *Ist der Diagnosepuffer voll, wird bei einem neuen Diagnosepuffereintrag der älteste Eintrag gelöscht.*
- *Somit steht der jüngste Eintrag immer an erster Stelle.*
- *Bei einem NetzAUS bzw. beim Umparametrieren des CP geht der Inhalt des Diagnosepuffers verloren.*

Diagnosepuffer am PG auslesen

Über den Siemens SIMATIC Manager können Sie mittels der Zielsystemfunktionen auf den Diagnosepuffer des CP zugreifen. Der Zugriff erfolgt nach folgender Vorgehensweise:

1. ➤ Starten Sie den Siemens SIMATIC Manager mit Ihrem Projekt.
2. ➤ Wählen Sie die entsprechende Station an und öffnen Sie diese im Hardware-Konfigurator.
3. ➤ Markieren Sie den gewünschten CP und wählen Sie *"Zielsystem → Baugruppenzustand"*.
 - ⇒ Der Registerdialog "Baugruppenzustand" zum CP wird angezeigt.
4. ➤ Wechseln Sie hier in die Registerseite "Diagnosepuffer".
 - ⇒ Hier werden die neuesten Diagnoseereignisse des CP im Klartext angezeigt.

Diagnosemeldung

- Unter "Details" können zusätzliche Informationen zu einer Fehlerursache stehen. Den Nummerncode finden Sie im Feld "Ereignis-ID". Der vordere Teil ist immer F1C8h.
- Im hinteren Teil finden Sie *Ereignisklasse* und die *Ereignisnummer*. ↪ *Kap. 7.2 "Diagnose über FB STATUS" Seite 97*
- Durch Auswahl der Schaltfläche [Hilfe zum Ereignis] wird die entsprechende *Abhilfe* eingeblendet, die Sie weiter oben in der Tabelle finden. Über die Schaltfläche [Aktualisieren] können Sie die Diagnosedaten erneut vom CP lesen.

7.4 Diagnose über Diagnosealarm

Übersicht

Der CP 341-1AH01 kann einen Diagnosealarm auf der zugeordneten CPU auslösen und somit eine Fehlfunktion des CP anzeigen. Über die Parametrierung können Sie vorgeben, ob der CP im Fehlerfall einen Diagnosealarm auslösen soll oder nicht. Defaultmäßig ist die Alarmgenerierung deaktiviert.

Bei aktiviertem Alarm können folgende Ereignisse einen Diagnosealarm auslösen:

- Drahtbruch auf der RxD-Leitung
- Fehler in der Parametrierung

Diagnosealarm

Im Fehlerfall stellt der CP über den Rückwandbus Diagnosedaten zur Verfügung. Diese werden von der CPU gelesen und im Diagnosepuffer der CPU abgelegt. Sie können jederzeit den CPU-Diagnosepuffer mit Ihrem Projektier-PC mittels der Zielsystemfunktionen auslesen. Beim Auftreten eines Diagnosealarms leuchtet die SF-LED und der OB 82 wird aufgerufen.

OB 82

- Sobald ein Fehler auftritt werden die Diagnosedaten als Startinformation an den OB 82 übergeben und dieser aufgerufen. Hier haben Sie die Möglichkeit durch entsprechende Programmierung auf die Diagnose zu reagieren.
- Haben Sie keinen OB 82 programmiert, geht die CPU bei einem Diagnosealarm automatisch in den Betriebszustand STOP über.

Diagnoseinformationen

Der CP stellt 4Byte Diagnoseinformationen zu Verfügung. Je nach Ereignis sind die 4Byte wie folgt belegt:

| Ereignis | Byte 0 | Byte 1 | Byte 2 | Byte 3 |
|-------------------|--------|--------|--------|--------|
| Drahtbruch an RxD | 25h | 0Ch | 02h | 00h |
| Parametrierfehler | 83h | 0Ch | 00h | 00h |