

# SPEED7 Library

OPL\_SP7-LIB | SW90GS0MA V10.011 | Handbuch

HB00 | OPL\_SP7-LIB | SW90GS0MA V10.011 | de | 24-02

Baustein Bibliothek - Serial Communication



YASKAWA Europe GmbH  
Philipp-Reis-Str. 6  
65795 Hattersheim  
Deutschland  
Tel.: +49 6196 569-300  
Fax: +49 6196 569-398  
E-Mail: [info@yaskawa.eu](mailto:info@yaskawa.eu)  
Internet: [www.yaskawa.eu.com](http://www.yaskawa.eu.com)

## Inhaltsverzeichnis

<b>1</b>	<b>Allgemeines.</b>	<b>4</b>
1.1	Copyright © YASKAWA Europe GmbH.	4
1.2	Über dieses Handbuch.	5
<b>2</b>	<b>Wichtige Hinweise.</b>	<b>6</b>
2.1	Allgemein.	6
2.2	Intern verwendete Bausteine.	6
2.3	Kein optimierter Bausteinzugriff.	7
2.4	Deklarationstypen.	7
<b>3</b>	<b>Bibliothek einbinden.</b>	<b>8</b>
3.1	Einbinden in Siemens SIMATIC Manager.	8
3.2	Einbinden in Siemens TIA Portal.	9
<b>4</b>	<b>Bausteinparameter.</b>	<b>10</b>
4.1	HW-Kennung - HW_ID.	10
4.2	RET_VAL und BUSY bei asynchron arbeitenden Bausteinen.	11
4.3	Allgemeine und spezifische Fehlercodes RET_VAL.	11
<b>5</b>	<b>Serielle Kommunikation - "Serial Communication".</b>	<b>14</b>
5.1	Serielle Kommunikation - "Serial Communication".	14
5.1.1	SFC 207 - SER_CTRL - Modemfunktionalität PtP.	14
5.1.2	FC/SFC 216 - SER_CFG - Parametrierung PtP.	15
5.1.3	FC/SFC 217 - SER_SND - Senden an PtP.	19
5.1.4	FC/SFC 218 - SER_RCV - Empfangen von PtP.	24
5.1.5	FB 1 - RECEIVE_ASCII - Empfangen mit definierter Länge von PtP.	26
5.1.6	FB 7 - P_RCV_RK - Empfangen von CP 341.	27
5.1.7	FB 8 - P_SND_RK - Senden an CP 341.	28
5.2	CP040.	30
5.2.1	Übersicht.	30
5.2.2	FB 60 - SEND - Senden an System SLIO CP 040.	31
5.2.3	FB 61 - RECEIVE - Empfangen von System SLIO CP 040.	33
5.2.4	FB 65 - CP040_COM - Kommunikation SLIO CP 040.	36
5.3	CP240.	41
5.3.1	FC 0 - SEND_ASCII_STX_3964 - Senden an CP 240.	41
5.3.2	FC 1 - RECEIVE_ASCII_STX_3964 - Empfangen von CP 240.	42
5.3.3	FC 8 - STEUERBIT - Modemfunktionalität CP 240.	44
5.3.4	FC 9 - SYNCHRON_RESET - Synchronisation CPU und CP 240.	45
5.3.5	FC 11 - ASCII_FRAGMENT - Fragment Datenempfang CP 240.	47

# 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, Philipp-Reis-Str. 6, 65795 Hattersheim, Deutschland

Tel.: +49 6196 569 300

Fax.: +49 6196 569 398

E-Mail: [info@yaskawa.eu](mailto:info@yaskawa.eu)

Internet: [www.yaskawa.eu.com](http://www.yaskawa.eu.com)

### 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

SLIO, System 300S und SPEED7 sind eingetragene Warenzeichen der YASKAWA Europe GmbH.

CAN ist ein eingetragenes Warenzeichen der CAN in Automation e. V. (CiA).

DeviceNet ist ein eingetragenes Warenzeichen der ODVA, Inc.

EtherCAT ist ein eingetragenes Warenzeichen der Beckhoff Automation GmbH.

Modbus ist ein eingetragenes Warenzeichen der Schneider Electric.

PROFINET und PROFIBUS sind eingetragene Warenzeichen der PROFIBUS and PROFINET International (PI).

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

Alle genannten Microsoft Windows, Office und Server-Produkte sind eingetragene Warenzeichen von Microsoft Inc., USA.

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

**Allgemeine  
Nutzungsbedingungen**

Es wurden alle Anstrengungen unternommen, um sicherzustellen, dass die in diesem Dokument enthaltenen Informationen zum Zeitpunkt der Veröffentlichung vollständig und richtig sind. Fehlerfreiheit kann nicht garantiert werden, das Recht auf Änderungen der Informationen bleibt jederzeit vorbehalten. Eine Informationspflicht gegenüber dem Kunden über etwaige Änderungen besteht nicht. Der Kunde ist aufgefordert, seine Dokumente aktiv aktuell zu halten. Der Einsatz der Produkte mit zugehöriger Dokumentation hat immer in Eigenverantwortung des Kunden unter Berücksichtigung der geltenden Richtlinien und Normen zu erfolgen. Die vorliegende Dokumentation beschreibt alle heute bekannten Hard- und Software-Einheiten und Funktionen. Es ist möglich, dass Einheiten beschrieben sind, die beim Kunden nicht vorhanden sind. Der genaue Lieferumfang ist im jeweiligen Kaufvertrag beschrieben.

**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](mailto:Documentation.HER@yaskawa.eu)

**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, Philipp-Reis-Str. 6, 65795 Hattersheim, Deutschland  
Tel.: +49 6196 569 500 (Hotline)  
E-Mail: [support@yaskawa.eu](mailto:support@yaskawa.eu)

## 1.2 Über dieses Handbuch

**Zielsetzung und Inhalt**

Das Handbuch beschreibt die Baustein-Bibliothek *"Serial Communication"*:

- Das Handbuch ist geschrieben für Anwender mit Grundkenntnissen in der Automatisierungstechnik.
- Das Handbuch ist in Kapitel gegliedert. Jedes Kapitel beschreibt eine abgeschlossene Thematik.
- Als Orientierungshilfe stehen im Handbuch zur Verfügung:
  - Gesamt-Inhaltsverzeichnis am Anfang des Handbuchs
  - Verweise mit Seitenangabe

**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.*

## 2 Wichtige Hinweise

### 2.1 Allgemein



*Nachfolgend finden Sie wichtige Hinweise, die grundsätzlich beim Einsatz der Bausteine zu beachten sind.*

### 2.2 Intern verwendete Bausteine



#### VORSICHT

Folgende Bausteine werden intern verwendet und dürfen nicht überschrieben werden! Der direkte Aufruf eines internen Bausteins führt zu Fehler im entsprechenden Instanz-DB! Bitte verwenden Sie für den Aufruf immer die zugehörige Funktion.

FC/SFC	Bezeichnung	Beschreibung
FC/SFC 131	TSEND_	wird intern für FB 63 verwendet
FC/SFC 132	TRECV_	wird intern für FB 64 verwendet
FC/SFC 133	TCON_	wird intern für FB 65 verwendet
FC/SFC 134	TDISCON_	wird intern für FB 66 verwendet
FC/SFC 135	TUSEND_	wird intern für FB 67 verwendet
FC/SFC 136	TURECV_	wird intern für FB 68 verwendet
FC/SFC 192	CP_S_R	wird intern für FB 7 und FB 8 verwendet
FC/SFC 196	AG_CNTRL	wird intern für FC 10 verwendet
FC/SFC 198	USEND_	wird intern für FB 8 verwendet
FC/SFC 199	URCV_	wird intern für FB 9 verwendet
FC/SFC 200	AG_GET	wird intern für FB/SFB 14 verwendet
FC/SFC 201	AG_PUT	wird intern für FB/SFB 15 verwendet
FC/SFC 202	AG_BSEND	wird intern für FB/SFB 12 verwendet
FC/SFC 203	AG_BRCV	wird intern für FB/SFB 13 verwendet
FC/SFC 204	IP_CONF	wird intern für FB 55 IP_CONF verwendet
FC/SFC 205	AG_SEND	wird intern für FC 5 AG_SEND verwendet
FC/SFC 206	AG_RECV	wird intern für FC 6 AG_RECV verwendet
FC/SFC 253	IBS_ACCESS	wird intern für SPEED-Bus-INTERBUS-Master verwendet
SFB 238	EC_RWOD	wird intern für EtherCAT-Kommunikation verwendet
SFB 239	FUNC	wird intern für FB 240, FB 241 verwendet

## 2.3 Kein optimierter Bausteinzugriff



*Bitte beachten Sie, dass die Bausteine für den Einsatz in S7-1500 CPUs von Siemens keinen optimierten Bausteinzugriff unterstützen! Bei verwendeten Instanz- und Datenbausteinen ist der optimierte Bausteinzugriff zu deaktivieren!*

### Bausteinzugriff einstellen

1. ➔ Öffnen Sie im Siemens TIA Portal in der *Projektnavigation* die *"Programmbausteine"*.
2. ➔ Wählen Sie den Baustein an, für welchen Sie den Bausteinzugriff ändern möchten und wählen Sie *"Kontextmenü → Eigenschaften"*.  
➔ Der "Eigenschaften"-Dialog des Bausteins wird geöffnet.
3. ➔ Wählen Sie *"Attribute"* an.
4. ➔ Deaktivieren Sie den Parameter *"Optimierter Bausteinzugriff"*.
5. ➔ Bestätigen Sie Ihre Eingabe mit [OK].

Näheres hierzu finden Sie im Handbuch zum Siemens TIA Portal.

## 2.4 Deklarationstypen

Bitte beachten Sie, dass die Schreibweisen der Deklarationstypen in Siemens STEP7 und TIA Portal sich unterscheiden. In dieser Dokumentation wird die Schreibweise für Siemens STEP7 verwendet. Eine Gegenüberstellung der Schreibweisen können Sie der nachfolgenden Tabelle entnehmen.

Siemens TIA Portal	Siemens STEP7
Input	IN
Output	OUT
InOut	IN_OUT
Static	STAT
Temp	TEMP

### 3 Bibliothek einbinden

#### Baustein-Bibliothek Serial Communication

Die Baustein-Bibliothek finden Sie im "Download Center" auf [www.yaskawa.eu.com](http://www.yaskawa.eu.com) unter "Controls Library" als "Baustein-Bibliothek Serial Communication - SW90GS0MA" zum Download. Die Bibliothek liegt als gepackte zip-Dateien vor. Sobald Sie die Bausteine verwenden möchten, müssen Sie diese in Ihr Projekt importieren.



*Bitte verwenden Sie immer das zu Ihrer Bibliothek zugehörige Handbuch. Solange es keine beschreibungsrelevante Änderungen gibt, können im Handbuch die Versionsangaben der Bibliothek und der zugehörigen Dateien von denen der Bibliothek abweichen.*

#### Folgende Bausteinbibliotheken stehen zur Verfügung

Datei	Beschreibung
SerialCom_S7_V0005.zip	<ul style="list-style-type: none"> <li>■ Bausteinbibliothek für Siemens SIMATIC Manager.</li> <li>■ Für den Einsatz in CPUs von Yaskawa bzw. S7-300 CPUs von Siemens.</li> </ul>
SerialCom_TIA_V0006.zip	<ul style="list-style-type: none"> <li>■ Bausteinbibliothek für Siemens TIA Portal V14/V15.</li> <li>■ Für den Einsatz in CPUs von Yaskawa bzw. S7-300 CPUs von Siemens.</li> </ul>
SerialCom_TIA_1500_V0005.zip	<ul style="list-style-type: none"> <li>■ Bausteinbibliothek für Siemens TIA Portal V14/V15.</li> <li>■ Für den Einsatz in S7-1500 CPUs von Siemens.</li> </ul>

#### 3.1 Einbinden in Siemens SIMATIC Manager

##### Übersicht

Die Einbindung in den Siemens SIMATIC Manager erfolgt nach folgenden Schritten:

1. ZIP-Datei laden
2. Bibliothek "dearchivieren"
3. Bibliothek öffnen und Bausteine in Projekt übertragen

##### ZIP-Datei laden

Navigieren Sie auf der Webseite zu der gewünschten ZIP-Datei, laden und speichern Sie diese in Ihrem Arbeitsverzeichnis.

##### Bibliothek dearchivieren

1. Starten Sie den Siemens SIMATIC Manager mit Ihrem Projekt.
2. Öffnen Sie mit "Datei → Dearchivieren" das Dialogfenster zur Auswahl der ZIP-Datei.
3. Wählen Sie die entsprechende ZIP-Datei an und klicken Sie auf [Öffnen].
4. Geben Sie ein Zielverzeichnis an, in dem die Bausteine abzulegen sind.
5. Starten Sie den Entpackvorgang mit [OK].

##### Bibliothek öffnen und Bausteine in Projekt übertragen

1. Öffnen Sie die Bibliothek nach dem Entpackvorgang.
2. Öffnen Sie Ihr Projekt und kopieren Sie die erforderlichen Bausteine aus der Bibliothek in das Verzeichnis "Bausteine" Ihres Projekts.
  - ➔ Nun haben Sie in Ihrem Anwenderprogramm Zugriff auf die Bausteine.



*Werden anstelle der SFCs FCs verwendet, so werden diese von den System 300S CPUs von Yaskawa ab Firmware 3.6.0 unterstützt.*



## 3.2 Einbinden in Siemens TIA Portal

### Übersicht

Die Einbindung in das Siemens TIA Portal erfolgt nach folgenden Schritten:

1. ZIP-Datei laden
2. ZIP-Datei entpacken
3. Bibliothek "dearchivieren"
4. Bibliothek öffnen und Bausteine in Projekt übertragen

### ZIP-Datei laden

1. Navigieren Sie auf der Webseite zu der ZIP-Datei, welche zu Ihrer Programmversion passt.

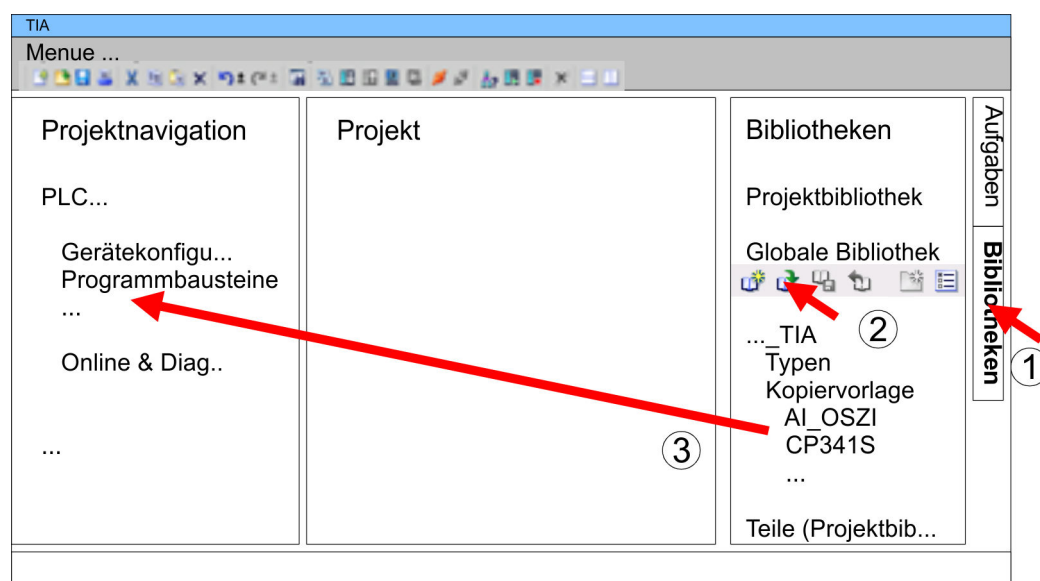
2. Laden und speichern Sie diese in Ihrem Arbeitsverzeichnis.

### ZIP-Datei entpacken

- Entpacken Sie die ZIP-Datei mit Ihrem Entpackprogramm in ein Arbeitsverzeichnis für das Siemens TIA Portal.

### Bibliothek öffnen und Bausteine in Projekt übertragen

1. Starten Sie das Siemens TIA Portal mit Ihrem Projekt.
2. Wechseln sie in die *Projektansicht*.
3. Wählen Sie auf der rechten Seite die Task-Card "Bibliotheken".
4. Klicken Sie auf "Globale Bibliothek".
5. Klicken Sie auf "Globale Bibliothek öffnen".
6. Navigieren Sie zu ihrem Arbeitsverzeichnis und laden Sie die Datei ...\_TIA.al1x.



7. Kopieren Sie die erforderlichen Bausteine aus der Bibliothek in das Verzeichnis "Programmbausteine" in der *Projektnavigation* Ihres Projekts. Nun haben Sie in Ihrem Anwenderprogramm Zugriff auf die Bausteine.

## 4 Bausteinparameter

### 4.1 HW-Kennung - HW\_ID

#### HW-Kennung

- Den Parameter *HW\_ID* zur Angabe der *HW-Kennung* gibt es nur bei Einsatz in S7-1500 CPUs von Siemens.
- Bei der Konfiguration einer Hardware-Komponente wird für jedes Objekt der Hardware-Konfiguration automatisch eine Hardware-Kennung als *HW-Kennung* vergeben.
- Die *HW-Kennung* umfasst Module, Ports, Schnittstellen und E/A-Bereiche von Bus-Systemen.
- Die *HW-Kennung* ist eine dezimale Ganzzahl-Konstante vom Datentyp *HW\_IO*.
- Bei der *HW-Kennung* wird nicht zwischen Ein- und Ausgabebereich unterschieden.
- Mit Hilfe der *HW-Kennung* können Sie die entsprechenden Hardware-Komponenten adressieren.

#### HW-Kennung ermitteln

Die *HW-Kennung* für die jeweilige Komponente können Sie mit folgender Vorgehensweise ermitteln:

1. ➤ Öffnen Sie in der *Projektnavigation* die "*Gerätekonfiguration*".
2. ➤ Selektieren Sie die gewünschte Hardware-Komponente, deren *HW-Kennung* Sie ermitteln möchten.
3. ➤ Klicken Sie im *Inspektor*-Fenster auf "*Allgemein*".
  - ➔ Die "*HW-Kennung*" wird angezeigt. Diese können Sie bei der Beschaltung der Bausteine in den Parameter *HW\_ID* übernehmen.

#### HW-Kennung und Systemkonstanten

Sie können auch die *HW-Kennung* über die "*Systemkonstanten*" ermitteln. Über "*Systemkonstanten*" im *Inspektor*-Fenster werden alle HW-Kennungen eines in der Gerätesicht markierten Objekts mit *Name* und *Typ* aufgelistet. *Name* und *Typ* werden bei der Zuordnung der HW-Kennung automatisch generiert. Hierbei besitzt *Name* einen hierarchischen Aufbau mit maximal 4 Hierarchieebenen, wobei jede Ebene durch ein "~" getrennt wird. Den Namen der Komponente der entsprechenden Hierarchieebene können Sie jederzeit über die Eigenschaften anpassen.

#### HW-Kennung im Anwenderprogramm

- Bei der Erstellung Ihres Anwenderprogramms können Sie über Doppelklick auf den entsprechenden Ein- bzw. Ausgabe-Parameter aus einer Liste aller möglichen Hardware-Komponenten die entsprechende Hardware-Komponente zuordnen.
- Bei einem Prozessalarm können sie über die Startinformationen die *HW-Kennung* als "*ID*" der alarmanauslösenden Hardware-Komponente ermitteln.

## 4.2 RET\_VAL und BUSY bei asynchron arbeitenden Bausteinen

### Ausgangsparameter RET\_VAL und BUSY

Bei asynchron arbeitenden Bausteinen erstreckt sich die Funktionsausführung über mehrere Bausteinaufrufe. Hierbei wird über die Ausgangsparameter RET\_VAL und BUSY der Zustand der Auftragsausführung angezeigt:

- **Erstaufwurf mit REQ = 1**
  - Bei freien Systemressourcen und korrekten Eingangsparametern wird BUSY auf 1 gesetzt und in RET\_VAL W#16#7001 eingetragen.
  - Sind die Systemressourcen belegt oder die Eingangsparameter fehlerhaft, wird BUSY auf 0 gesetzt und in RET\_VAL der entsprechende Fehlercode eingetragen.
- **Zwischenaufwurf**
  - BUSY bleibt auf 1 gesetzt und in RET\_VAL wird W#16#7002 eingetragen. Dies signalisiert, dass sich der Auftrag noch in Bearbeitung befindet.
- **Letzter Aufruf**
  - Nach fehlerfreier Ausführung wird BUSY auf 0 gesetzt und in RET\_VAL 0 eingetragen. Bitte beachten Sie, dass manche Bausteine über RET\_VAL die Anzahl der übertragenen Daten rückmelden. Nähere Informationen hierzu finden Sie in den entsprechenden Bausteinbeschreibungen.
  - Im Fehlerfall wird BUSY auf 0 gesetzt und in RET\_VAL der entsprechende Fehlercode eingetragen.

### Eingangsparameter REQ

Der Eingangsparameter REQ dient ausschließlich dem Anstoß eines Auftrags:

- Durch REQ = 1 im entsprechenden Baustein wird ein noch nicht aktiver Auftrag ausgeführt.
- Bei jedem Folgeaufruf des Bausteins wird REQ nicht ausgewertet.

### Zusammenhang der Parameter

Aufruf-Nr.	Aufrufart	REQ	RET_VAL	BUSY
1	Erstaufwurf	1	W#16#7001	1
			Fehlercode im Fehlerfall	0
2 ... n-1	Zwischenaufwurf	nicht relevant	W#16#7002	1
n	Letzter Aufruf	nicht relevant	W#16#0000 <sup>1</sup>	0
			Fehlercode im Fehlerfall	

1) Bei manchen Bausteinen die Anzahl der übertragenen Daten - siehe entsprechende Bausteinbeschreibung.

## 4.3 Allgemeine und spezifische Fehlercodes RET\_VAL

### Übersicht

Der Rückgabewert *RET\_VAL* einer Systemfunktion stellt einen der beiden folgenden Fehlercodes zur Verfügung:

- *Allgemeiner Fehlercode*, der sich auf jeden beliebigen SFC beziehen kann.
- *Spezifischer Fehlercode*, der sich auf den jeweiligen SFC bezieht.

Es handelt sich beim Datentyp des Ausgangsparameters *RET\_VAL* zwar um eine Ganzzahl (INT), doch die Fehlercodes der Systemfunktionen werden nach hexadezimalen Werten gegliedert.

Wenn Sie einen Rückgabewert auswerten und den Wert mit den Fehlercodes vergleichen, so lassen Sie sich den Fehlercode im Hexadezimalformat ausgeben.

**RET\_VAL (Rückgabewert)**

Die folgende Tabelle zeigt den Aufbau eines Fehlercodes:

Bit	Bedeutung
7 ... 0	Ereignisnummer bzw. Fehlerklasse und Einzelfehler
14 ... 8	Bit 14 ... 8 = "0": <b>Spezifischer Fehlercode</b> Den spezifischen Fehlercode finden Sie in der Beschreibung der einzelnen SFCs. Bit 14 ... 8 > "0": <b>Allgemeiner Fehlercode</b> Die möglichen allgemeinen Fehlercodes finden Sie auf der folgenden Seite.
15	Bit 15 = "1": zeigt an, dass ein Fehler aufgetreten ist.

**Spezifischer Fehlercode**

Dieser Fehlercode zeigt an, dass ein Fehler, der zu einer bestimmten Systemfunktion gehört, während der Bearbeitung aufgetreten ist.

Ein spezifischer Fehlercode besteht aus:

- Fehlerklasse zwischen 0 und 7
- Einzelfehler zwischen 0 und 15

Bit	Bedeutung
3 ... 0	Einzelfehler
6 ... 4	Fehlerklasse
7	Bit 7 = "1"
14 ... 8	Bit 14 ... 8 = "0"
15	Bit 15 = "1": zeigt an, dass ein Fehler aufgetreten ist.

**Allgemeine Fehlercodes  
RET\_VAL**

Der Parameter *RET\_VAL* verschiedener SFCs liefert keine spezifischen, sondern nur allgemeine Fehlerinformationen zurück.

Der allgemeine Fehlercode enthält Fehlerinformationen, die bei allen Systemfunktionen auftreten können. Ein allgemeiner Fehlercode besteht aus den beiden folgenden Nummern:

- Eine Parameternummer zwischen 1 und 111, wobei 1 den ersten Parameter, 2 den zweiten Parameter usw. des aufgerufenen SFC anzeigt.
- Eine Ereignisnummer zwischen 0 und 127. Die Ereignisnummer zeigt einen synchronen Fehler an.

Bit	Bedeutung
7 ... 0	Ereignisnummer
14 ... 8	Parameternummer
15	Bit 15 = "1": zeigt an, dass ein Fehler aufgetreten ist.

**Allgemeine Fehlercodes**

In der folgenden Tabelle werden die allgemeinen Fehlercodes eines Rückgabewerts erläutert. Die Darstellung erfolgt im Hexadezimalformat, wobei der Buchstabe x in jeder Codenummer nur als Platzhalter dient und die Nummer des Parameters der Systemfunktion darstellt, die den Fehler verursacht hat.

Fehlercode	Beschreibung
8x7Fh	Interner Fehler. Dieser Fehlercode zeigt einen internen Fehler am Parameter x an. Dieser Fehler wurde nicht vom Anwender verursacht und kann von ihm auch nicht behoben werden.
8x01h	Unzulässige Syntaxkennung bei einem ANY-Parameter.
8x22h	Bereichslängenfehler beim Lesen eines Parameters.
8x23h	Bereichslängenfehler beim Schreiben eines Parameters. Dieser Fehlercode zeigt an, dass sich der Parameter x vollständig oder teilweise außerhalb des Operandenbereichs befindet oder die Länge eines Bitfeldes bei einem ANY-Parameter nicht durch 8 teilbar ist.
8x24h	Bereichsfehler beim Lesen eines Parameters.
8x25h	Bereichsfehler beim Schreiben eines Parameters. Dieser Fehlercode zeigt an, dass sich der Parameter x in einem Bereich befindet, der für die Systemfunktion unzulässig ist. Die Beschreibung der jeweiligen Funktion gibt die Bereiche an, die für die Funktion unzulässig sind.
8x26h	Der Parameter enthält eine zu große Nummer einer Zeitzelle. Dieser Fehlercode zeigt an, dass die Zeitzelle, die in Parameter x angegeben wird, nicht vorhanden ist.
8x27h	Der Parameter enthält eine zu große Nummer einer Zählerzelle (Nummernfehler des Zählers). Dieser Fehlercode zeigt an, dass die Zählerzelle, die in Parameter x angegeben wird, nicht vorhanden ist.
8x28h	Ausrichtungsfehler beim Lesen eines Parameters.
8x29h	Ausrichtungsfehler beim Schreiben eines Parameters. Dieser Fehlercode zeigt an, dass der Verweis auf den Parameter x ein Operand ist, dessen Bitadresse ungleich 0 ist.
8x30h	Der Parameter befindet sich in dem schreibgeschützten Global-DB.
8x31h	Der Parameter befindet sich in dem schreibgeschützten Instanz-DB. Dieser Fehlercode zeigt an, dass der Parameter x sich in einem schreibgeschützten Datenbaustein befindet. Wenn der Datenbaustein von der Systemfunktion selbst geöffnet wurde, gibt die Systemfunktion immer den Wert 8x30h aus.
8x32h	Der Parameter enthält eine zu große DB-Nummer (Nummernfehler des DBs).
8x34h	Der Parameter enthält eine zu große FC-Nummer (Nummernfehler des FCs).
8x35h	Der Parameter enthält eine zu große FB-Nummer (Nummernfehler des FBs). Dieser Fehlercode zeigt an, dass der Parameter x eine Bausteinnummer enthält, die größer ist als die maximal zulässige Bausteinnummer.
8x3Ah	Der Parameter enthält die Nummer eines DBs, der nicht geladen ist.
8x3Ch	Der Parameter enthält die Nummer eines FCs, der nicht geladen ist.
8x3Eh	Der Parameter enthält die Nummer eines FBs, der nicht geladen ist.
8x42h	Es ist ein Zugriffsfehler aufgetreten, während das System einen Parameter aus dem Peripheriebereich der Eingänge auslesen wollte.
8x43h	Es ist ein Zugriffsfehler aufgetreten, während das System einen Parameter in den Peripheriebereich der Ausgänge schreiben wollte.
8x44h	Fehler beim n-ten ( $n > 1$ ) Lesezugriff nach Auftreten eines Fehlers.
8x45h	Fehler beim n-ten ( $n > 1$ ) Schreibzugriff nach Auftreten eines Fehlers. Dieser Fehlercode zeigt an, dass der Zugriff auf den gewünschten Parameter verweigert wird.

## 5 Serielle Kommunikation - "Serial Communication"

### 5.1 Serielle Kommunikation - "Serial Communication"

#### 5.1.1 SFC 207 - SER\_CTRL - Modemfunktionalität PtP

##### Beschreibung



*Bitte beachten Sie, dass dieser Baustein von SPEED7 CPUs nicht unterstützt wird!*

*Der Einsatz in S7-1500 CPUs von Siemens wird ebenfalls nicht unterstützt!*

Bei Einsatz des ASCII-Protokolls über die RS232-Schnittstelle haben Sie mit diesem Baustein zur Laufzeit Zugriff auf die seriellen Modemleitungen. Abhängig vom Parameter *FLOWCONTROL*, den Sie über *SFC 216 (SER\_CFG)* vorgeben, bietet der Baustein folgende Funktionalität:

	Lesen	Schreiben
<i>FLOWCONTROL=0:</i>	DTR, RTS, DSR, RI, CTS, CD	DTR, RTS
<i>FLOWCONTROL&gt;0:</i>	DTR, RTS, DSR, RI, CTS, CD	nicht möglich

##### Parameter

Name	Deklaration	Typ	Beschreibung
WRITE	IN	BYTE	<ul style="list-style-type: none"> <li>■ Bit 0: Neuer Zustand DTR</li> <li>■ Bit 1: Neuer Zustand RTS</li> </ul>
MASKWRITE	IN	BYTE	<ul style="list-style-type: none"> <li>■ Bit 0: Zustand an DTR übergeben</li> <li>■ Bit 1: Zustand an RTS übergeben</li> </ul>
READ	OUT	BYTE	Status (CTS, DSR, RI, CD, DTR, RTS)
READDELTA	OUT	BYTE	Status Änderung seit letztem Zugriff
RETVAL	OUT	WORD	Rückgabewert (0 = OK)

##### WRITE

Mit diesem Parameter geben Sie den Status für DTR und RTS vor, den Sie über *MASKWRITE* aktivieren können. Das Byte hat folgende Belegung:

- Bit 0 = DTR
- Bit 1 = RTS
- Bit 7 ... Bit 2: reserviert

##### MASKWRITE

Hier wird mit "1" der Status des entsprechenden Parameters übernommen. Das Byte hat folgende Belegung:

- Bit 0 = DTR
- Bit 1 = RTS
- Bit 7 ... Bit 2: reserviert

**READ** *READ* liefert den aktuellen Status der Modem-Leitungen zurück. *READDELTA* liefert den Status der Modem-Leitungen zurück, die sich seit dem letzten Zugriff geändert haben. Die Bytes haben folgenden Aufbau:

Bit-Nr.	7	6	5	4	3	2	1	0
Read	x	x	RTS	DTR	CD	RI	DSR	CTS
ReadDelta	x	x	x	x	CD	RI	DSR	CTS

**RETVAL (Rückgabewert)**

Wert	Beschreibung
0000h	kein Fehler
8x24h	Fehler in SFC-Parameter x, mit x: <ul style="list-style-type: none"> <li>1: Fehler in <i>WRITE</i></li> <li>2: Fehler in <i>MASKWRITE</i></li> <li>3: Fehler in <i>READ</i></li> <li>4: Fehler in <i>READDELTA</i></li> </ul>
809Ah	Schnittstelle ist nicht vorhanden
809Bh	Schnittstelle ist nicht konfiguriert (SFC 216)

### 5.1.2 FC/SFC 216 - SER\_CFG - Parametrierung PtP

**Beschreibung**

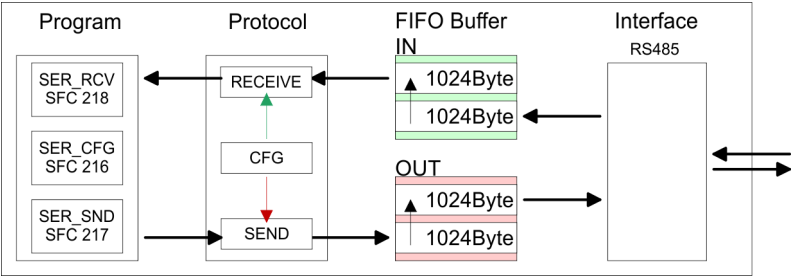


*Bitte beachten Sie, dass dieser Baustein ausschließlich für den Einsatz in CPUs von Yaskawa bzw. in S7-300 CPUs von Siemens geeignet ist!*

Über eine Hardware-Konfiguration können Sie unter Objekteigenschaften über den Parameter "Funktion RS485" den in der SPEED7-CPU integrierten DP-Master deaktivieren und die RS485-Schnittstelle für PtP-Kommunikation (point to point) freigeben. Die RS485-Schnittstelle im PtP-Betrieb ermöglicht die serielle Prozessankopplung zu verschiedenen Ziel- oder Quellsystemen. Die Parametrierung erfolgt zur Laufzeit unter Einsatz des FC/SFC 216 (SER\_CFG). Hierbei sind für alle Protokolle mit Ausnahme von ASCII die Parameter in einem DB abzulegen.

**Kommunikation**

- Daten, die von der CPU in den entsprechenden Datenkanal geschrieben werden, werden in einen FIFO-Sendepuffer (first in first out) mit einer Größe von 2x1024Byte abgelegt und von dort über die Schnittstelle ausgegeben.
- Empfängt die Schnittstelle Daten, werden diese in einem FIFO-Empfangspuffer mit einer Größe von 2x1024Byte abgelegt und können dort von der CPU gelesen werden.
- Sofern Daten mittels eines Protokolls übertragen werden, erfolgt die Einbettung der Daten in das entsprechende Protokoll automatisch. Im Gegensatz zu ASCII und STX/ETX erfolgt bei den Protokollen 3964R, USS und Modbus die Datenübertragung mit Quittierung der Gegenseite.
- Durch erneuten Aufruf des FC/SFC 217 SER\_SND bekommen Sie über RETVAL einen Rückgabewert geliefert, der unter anderem auch aktuelle Informationen über die Quittierung der Gegenseite beinhaltet. Zusätzlich ist bei USS und Modbus nach einem SER\_SND das Quittungstelegramm durch Aufruf des FC/SFC 218 SER\_RCV auszulesen.



Parameter

Parameter	Deklaration	Datentyp	Beschreibung
PROTOCOL	IN	BYTE	1=ASCII, 2=STX/ETX, 3=3964R
PARAMETER	IN	ANY	Zeiger zu den Protokoll-Parametern
BAUDRATE	IN	BYTE	Nr. der Baudrate
CHARLEN	IN	BYTE	0=5Bit, 1=6Bit, 2=7Bit, 3=8Bit
PARITY	IN	BYTE	0=Non, 1=Odd, 2=Even
STOPBITS	IN	BYTE	1=1Bit, 2=1,5Bit, 3=2Bit
FLOWCONTROL	IN	BYTE	1 - siehe Hinweis
RETVAL	OUT	WORD	Rückgabewert (0 = OK)

Alle Zeitangaben für Timeouts sind als Hexadezimaler Wert anzugeben. Den Hex-Wert erhalten Sie, indem Sie die gewünschte Zeit in Sekunden mit der Baudrate multiplizieren.

Beispiel:

- Gewünschte Zeit 8ms bei einer Baudrate von 19200Baud
- Berechnung:  $19200\text{Bit/s} \times 0,008\text{s} \approx 154\text{Bit} \rightarrow (9\text{Ah})$
- Als Hex-Wert ist 9Ah vorzugeben.

PROTOCOL

Geben Sie hier das Protokoll an, das verwendet werden soll. Zur Auswahl stehen:

- 1: ASCII
- 2: STX/ETX
- 3: 3964R
- 4: USS Master
- 5: Modbus RTU Master
- 6: Modbus ASCII Master

PARAMETER (als DB)

Bei eingestelltem ASCII-Protokoll wird dieser Parameter ignoriert. Für die Protokolle geben Sie hier einen DB an, der die Kommunikationsparameter beinhaltet und für die jeweiligen Protokolle STX/ETX, 3964R, USS und Modbus folgenden Aufbau hat:

Datenbaustein bei STX/ETX			
DBB0:	STX1	BYTE	(1. Start-Zeichen in hexadezimaler Form)
DBB1:	STX2	BYTE	(2. Start-Zeichen in hexadezimaler Form)
DBB2:	ETX1	BYTE	(1. Ende-Zeichen in hexadezimaler Form)



DBB3:	ETX2	BYTE	(2. Ende-Zeichen in hexadezimaler Form)
DBW4:	TIMEOUT	WORD	(max. zeitlicher Abstand zwischen 2 Telegrammen)



Das Zeichen für Start bzw. Ende sollte immer ein Wert kleiner 20 sein, ansonsten wird das Zeichen ignoriert!

Tragen Sie immer für nicht benutzte Zeichen FFh ein!

#### Datenbaustein bei 3964R

DBB0:	Prio	BYTE	(Die Priorität beider Partner muss unterschiedlich sein)
DBB1:	ConnAttmptNr	BYTE	(Anzahl der Verbindungsaufbauversuche)
DBB2:	SendAttmptNr	BYTE	(Anzahl der Telegrammwiederholungen)
DBB4:	CharTimeout	WORD	(Zeichenverzugszeit)
DBW6:	ConfTimeout	WORD	(Quittungsverzugszeit)

#### Datenbaustein bei USS

DBW0:	Timeout	WORD	(Verzugszeit)
-------	---------	------	---------------

#### Datenbaustein bei Modbus-Master

DBW0:	Timeout	WORD	(Antwort-Verzugszeit)
-------	---------	------	-----------------------

## BAUDRATE

Geschwindigkeit der Datenübertragung in Bit/s (Baud).

04h:	1200Baud	05h:	1800Baud	06h:	2400Baud	07h:	4800Baud
08h:	7200Baud	09h:	9600Baud	0Ah:	14400Baud	0Bh:	19200Baud
0Ch:	38400Baud	0Dh:	57600Baud	0Eh:	115200Baud		

## CHARLEN

Anzahl der Datenbits, auf die ein Zeichen abgebildet wird.

0: 5Bit	1: 6Bit	2: 7Bit	3: 8Bit
---------	---------	---------	---------

## PARITY

Die Parität ist je nach Wert gerade oder ungerade. Zur Paritätskontrolle werden die Informationsbits um das Paritätsbit erweitert, das durch seinen Wert ("0" oder "1") den Wert aller Bits auf einen vereinbarten Zustand ergänzt. Ist keine Parität vereinbart, wird das Paritätsbit auf "1" gesetzt, aber nicht ausgewertet.

0: NONE	1: ODD	2: EVEN
---------	--------	---------

## STOPBITS

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

1: 1Bit	2: 1,5Bit <sup>1</sup>	3: 2Bit
---------	------------------------	---------

1) Nur zulässig bei *CHARLEN* = 0 (5Bit)

## FLOWCONTROL

Der Parameter *FLOWCONTROL* wird ignoriert. Beim Senden ist *RTS*=1, beim Empfangen ist *RTS*=0.



### Sonderfunktion in System MICRO CPU

Ab der Firmware-Version 2.4.4 können Sie in einer System MICRO CPU zwischen RS422- und RS485-Kommunikation umschalten.

0: RS422-Kommunikation

1: RS485-Kommunikation

## RETVAL FC/SFC 216 (Rückgabewert)

Rückgabewerte, die der Baustein liefert:

Fehlercode	Beschreibung
0000h	kein Fehler
809Ah	Schnittstelle ist nicht vorhanden bzw. Schnittstelle wird für PROFIBUS verwendet.
8x24h	Fehler in FC/SFC-Parameter x, mit x: 1: Fehler in <i>PROTOKOLL</i> 2: Fehler in <i>PARAMETER</i> 3: Fehler in <i>BAUDRATE</i> 4: Fehler in <i>CHARLENGTH</i> 5: Fehler in <i>PARITY</i> 6: Fehler in <i>STOPBITS</i> 7: Fehler in <i>FLOWCONTROL</i> (Parameter fehlt)
809xh	Fehler in Wert des FC/SFC-Parameter x, mit x: 1: Fehler in <i>PROTOKOLL</i> 3: Fehler in <i>BAUDRATE</i> 4: Fehler in <i>CHARLENGTH</i> 5: Fehler in <i>PARITY</i> 6: Fehler in <i>STOPBITS</i>
8092h	Zugriffsfehler auf Parameter-DB (DB zu kurz)
828xh	Fehler in Parameter x von DB-Parameter mit x: 1: Fehler im 1. Parameter 2: Fehler im 2. Parameter ...

### 5.1.3 FC/SFC 217 - SER\_SND - Senden an PtP

#### Beschreibung



*Bitte beachten Sie, dass dieser Baustein ausschließlich für den Einsatz in CPUs von Yaskawa bzw. in S7-300 CPUs von Siemens geeignet ist!*

Mit diesem Baustein werden Daten über die serielle Schnittstelle gesendet. Durch erneuten Aufruf des FC/SFC 217 SER\_SND bekommen Sie bei 3964R, USS und Modbus über RETVAL einen Rückgabewert geliefert, der unter anderem auch aktuelle Informationen über die Quittierung der Gegenseite beinhaltet. Zusätzlich ist bei USS und Modbus nach einem SER\_SND das Quittungstelegramm durch Aufruf des FC/SFC 218 SER\_RCV auszulesen.

#### Parameter

Parameter	Deklaration	Datentyp	Beschreibung
DATAPTR	IN	ANY	Zeiger auf Sendedaten
DATALEN	OUT	WORD	Länge der Sendedaten
RETVAL	OUT	WORD	Rückgabewert (0 = OK)

#### DATAPTR

Geben Sie hier einen Bereich vom Typ Pointer für den Sendepuffer an, in den die Daten, die gesendet werden sollen, abzulegen sind. Anzugeben sind Typ, Anfang und Länge.

Beispiel:

- Daten liegen in DB5 ab 0.0 mit einer Länge von 124Byte
- DataPtr:=P#DB5.DBX0.0 BYTE 124

#### DATALEN

- Wort, in dem die Anzahl der gesendeten Bytes abgelegt wird.
- Werden unter **ASCII** die Daten intern mittels FC/SFC 217 schneller an die serielle Schnittstelle übertragen als sie gesendet werden können, kann aufgrund eines Pufferüberlaufs die zu sendende Datenlänge von **DATALEN** abweichen. Dies sollte im Anwenderprogramm berücksichtigt werden!
- Bei **STX/ETX**, **3964R**, **Modbus** und **USS** wird immer die unter **DATAPTR** angegebene Länge oder 0 eingetragen.

#### RETVAL FC/SFC 217 (Rückgabewerte)

Rückgabewerte, die der Baustein liefert:

Fehlercode	Beschreibung
0000h	Daten gesendet - fertig
1000h	Nichts gesendet (Datenlänge 0)
20xxh	Protokoll wurde fehlerfrei ausgeführt mit xx-Bitmuster für Diagnose
7001h	Daten liegen im internen Puffer - aktiv (busy)
7002h	Transfer - aktiv
80xxh	Protokoll wurde fehlerhaft ausgeführt mit xx-Bitmuster für Diagnose (keine Quittung der Gegenseite)
90xxh	Protokoll wurde nicht ausgeführt mit xx-Bitmuster für Diagnose (keine Quittung der Gegenseite)

Fehlercode	Beschreibung
8x24h	Fehler in FC/SFC-Parameter x, mit x: 1: Fehler in <i>DATAPTR</i> 2: Fehler in <i>DATALEN</i>
8122h	Fehler in Parameter <i>DATAPTR</i> (z.B. DB zu kurz)
807Fh	Interner Fehler
809Ah	Schnittstelle nicht vorhanden bzw. Schnittstelle wird für PROFIBUS verwendet
809Bh	Schnittstelle nicht konfiguriert

**Protokollspezifische  
RETVAL-Werte**

**ASCII**

Wert	Beschreibung
9000h	Pufferüberlauf (keine Daten gesendet)
9002h	Daten sind zu kurz (0Byte)

**STX/ETX**

Wert	Beschreibung
9000h	Pufferüberlauf (keine Daten gesendet)
9001h	Daten sind zu lang (>1024Byte)
9002h	Daten sind zu kurz (0Byte)
9004h	Unzulässiges Zeichen

**3964R**

Wert	Beschreibung
2000h	Senden fertig ohne Fehler
80FFh	NAK empfangen - Fehler in der Kommunikation
80FEh	Datenübertragung ohne Quittierung der Gegenseite oder mit fehlerhafter Quittierung
9000h	Pufferüberlauf (keine Daten gesendet)
9001h	Daten sind zu lang (>1024Byte)
9002h	Daten sind zu kurz (0Byte)

**USS**

Fehlercode	Beschreibung
2000h	Senden fertig ohne Fehler
8080h	Empfangspuffer voll (kein Platz für Quittung)
8090h	Quittungsverzugszeit überschritten
80F0h	Falsche Checksumme in Rückantwort

Fehler-code	Beschreibung
80FEh	Falsches Startzeichen in der Rückantwort
80FFh	Falsche Slave-Adresse in der Rückantwort
9000h	Pufferüberlauf (keine Daten gesendet)
9001h	Daten sind zu lang (>1024Byte)
9002h	Daten sind zu kurz (<2Byte)

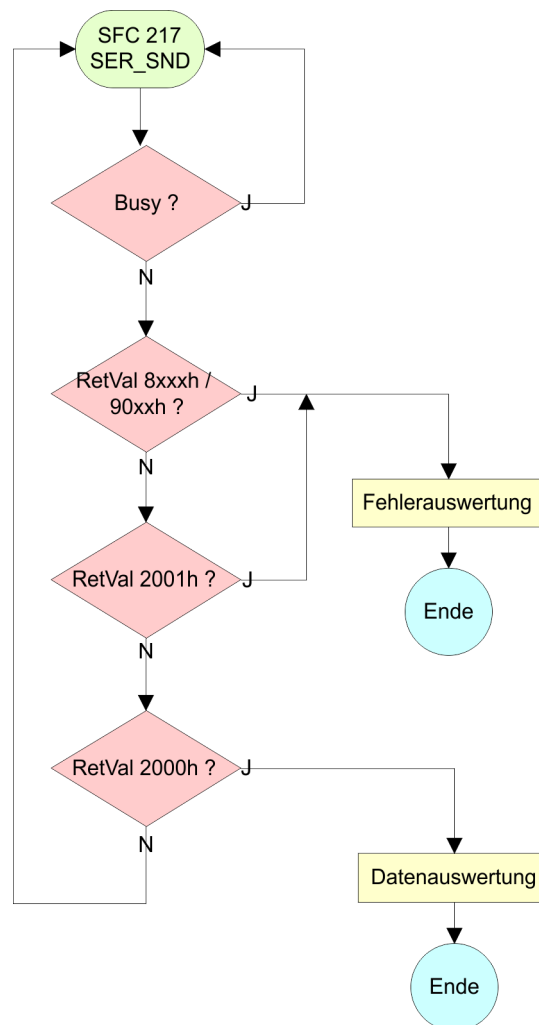
**Modbus RTU/ASCII Master**

Fehler-code	Beschreibung
2000h	Senden fertig (positive Slave-Rückmeldung vorhanden)
2001h	Senden fertig (negative Slave-Rückmeldung vorhanden)
8080h	Empfangspuffer voll (kein Platz für Quittung)
8090h	Quittungsverzugszeit überschritten
80F0h	Falsche Checksumme in Rückantwort
80FDh	Länge der Rückantwort ist zu lang
80FEh	Falscher Funktionscode in der Rückantwort
80FFh	Falsche Slave-Adresse in der Rückantwort
9000h	Pufferüberlauf (keine Daten gesendet)
9001h	Daten sind zu lang (>1024Byte)
9002h	Daten sind zu kurz (<2Byte)

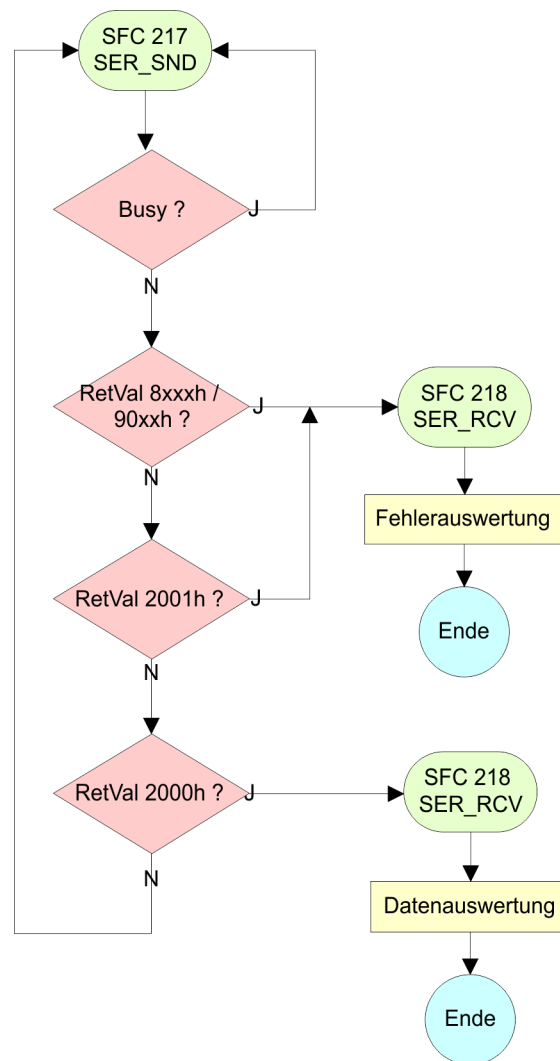
## Prinzip der Programmierung

Nachfolgend soll kurz die Struktur zur Programmierung eines Sendeauftrags für die verschiedenen Protokolle gezeigt werden.

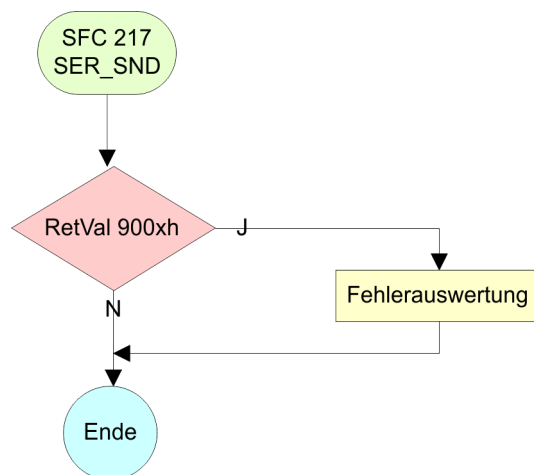
3964R



## USS / Modbus



## ASCII / STX/ETX



## 5.1.4 FC/SFC 218 - SER\_RCV - Empfangen von PtP

### Beschreibung



*Bitte beachten Sie, dass dieser Baustein ausschließlich für den Einsatz in CPUs von Yaskawa bzw. in S7-300 CPUs von Siemens geeignet ist!*

Mit diesem Baustein werden Daten über die serielle Schnittstelle empfangen. Bei den Protokollen USS und Modbus können Sie durch Aufruf des FC/SFC 218 SER\_RCV nach einem SER\_SND das Quittungstelegramm auslesen.

### Parameter

Parameter	Deklaration	Datentyp	Beschreibung
DATAPTR	IN	ANY	Zeiger auf Empfangspuffer
DATALEN	OUT	WORD	Länge der empfangenen Daten
ERROR	OUT	WORD	Fehler-Nr.
RETVAL	OUT	WORD	Rückgabewert (0 = OK)

### DATAPTR

Geben Sie hier einen Bereich vom Typ Pointer für den Empfangspuffer an, in den die Daten, die empfangen werden, abzulegen sind. Anzugeben sind Typ, Anfang und Länge.

Beispiel:

- Daten sind in DB5 ab 0.0 mit einer Länge von 124Byte abzulegen
- DataPtr:=P#DB5.DBX0.0 BYTE 124

### DATALEN

- Wort, in dem die Anzahl der empfangenen Bytes abgelegt wird.
- Bei **STX/ETX** und **3964R** wird immer die Länge der empfangenen Nutzdaten oder 0 eingetragen.
- Unter **ASCII** wird hier die Anzahl der gelesenen Zeichen eingetragen. Dieser Wert kann von der Telegrammlänge abweichen.

### ERROR

In diesem Wort erfolgt ein Eintrag im Fehlerfall. Folgende Fehlermeldungen können protokollabhängig generiert werden:

#### ASCII

Bit	Fehler	Beschreibung
0	overrun	Überlauf, ein Zeichen konnte nicht schnell genug aus der Schnittstelle gelesen werden kann
1	framing error	Fehler, der anzeigt, dass ein definierter Bitrahmen nicht übereinstimmt, die zulässige Länge überschreitet oder eine zusätzliche Bitfolge enthält (Stoppbitfehler)
2	parity	Paritätsfehler
3	overflow	Der Puffer ist voll.

#### STX/ETX

Bit	Fehler	Beschreibung
0	overflow	Das empfangene Telegramm übersteigt die Größe des Empfangspuffers.



Bit	Fehler	Beschreibung
1	char	Es wurde ein Zeichen außerhalb des Bereichs 20h ... 7Fh empfangen.
3	overflow	Der Puffer ist voll.

**3964R / Modbus RTU/ASCII Master**

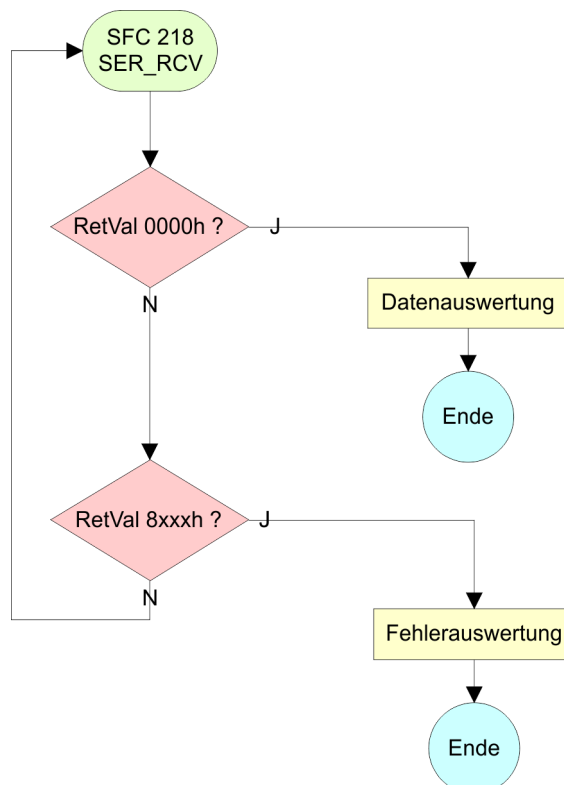
Bit	Fehler	Beschreibung
0	overflow	Das empfangene Telegramm übersteigt die Größe des Empfangspuffers.

**RETVAL FC/SFC 218  
(Rückgabewert)**

Fehlercode	Beschreibung
0000h	kein Fehler
1000h	Empfangspuffer ist zu klein (Datenverlust)
8x24h	Fehler in FC/SFC-Parameter x, mit x: 1: Fehler in <i>DATAPTR</i> 2: Fehler in <i>DATALEN</i> 3: Fehler in <i>ERROR</i>
8122h	Fehler in Parameter <i>DATAPTR</i> (z.B. DB zu kurz)
809Ah	Schnittstelle nicht vorhanden bzw. Schnittstelle wird für PROFIBUS verwendet
809Bh	Schnittstelle ist nicht konfiguriert

**Prinzip der Programmierung**

Nachfolgend sehen Sie die Grundstruktur zur Programmierung eines Receive-Auftrags. Diese Struktur können Sie für alle Protokolle verwenden.



## 5.1.5 FB 1 - RECEIVE\_ASCII - Empfangen mit definierter Länge von PtP

### Beschreibung



*Bitte beachten Sie, dass dieser Baustein ausschließlich für den Einsatz in CPUs von Yaskawa bzw. in S7-300 CPUs von Siemens geeignet ist!*

Dieser FB sammelt die Daten die über die interne serielle Schnittstelle im PtP-Betrieb empfangen werden und kopiert diese in den mittels *EMPF\_PUFFER* angegebenen Telegrammpuffer. Wurde das komplette Telegramm empfangen, wird *EMPF\_FERTIG* gesetzt und der FB verlassen. Das Einlesen der Daten, kann mehrere FB-Aufrufe erfordern. Das nächste Telegramm wird erst eingelesen, wenn das Bit *EMPF\_FERTIG* vom Anwender zurückgesetzt wurde. Mit diesem FB können nur Telegramme mit fester Länge empfangen werden.

### Parameter

Parameter	Deklaration	Datentyp	Beschreibung
EMPF_PUFFER	IN	ANY	Zeiger auf DB, in den das empfangene Telegramm übertragen wird.
ER_BYTE	OUT	WORD	Fehlercode
EMPF_FERTIG	IN_OUT	BOOL	Status

### EMPF\_PUFFER

Geben Sie hier einen Bereich vom Typ Pointer an, in den die empfangenen Daten zu kopieren sind. Anzugeben sind Typ, Anfang und Länge.

Beispiel:

- Daten sind in DB5 ab 0.0 mit einer Länge von 124Byte abzulegen
  - DataPtr:=P#DB5.DBX0.0 BYTE 124

### ER\_BYTE

In diesem Wort erfolgt ein Eintrag im Fehlerfall.

Fehlercode	Beschreibung
0003h	DB mit Telegrammpuffer nicht vorhanden.
0004h	DB mit Telegrammpuffer ist zu kurz.
7000h	Empfangspuffer ist zu klein - Daten wurden gelöscht!
8000h	Pointerangabe in <i>EMPF_PUFFER</i> fehlerhaft oder nicht vorhanden.
9001h	DB-Angabe in <i>EMPF_PUFFER</i> fehlerhaft oder nicht vorhanden.
9002h	Längen-Angabe in <i>EMPF_PUFFER</i> fehlerhaft oder nicht vorhanden.

### 5.1.6 FB 7 - P\_RCV\_RK - Empfangen von CP 341

#### Beschreibung



*Bitte beachten Sie, dass dieser Baustein ausschließlich für den Einsatz in CPUs von Yaskawa bzw. in S7-300 CPUs von Siemens geeignet ist!*

Der FB 7 P\_RCV\_RK überträgt Daten vom CP in einen Datenbereich der CPU, spezifiziert durch die Parameter *DB\_NO*, *DBB\_NO* und *LEN*. Der FB wird zur Datenübertragung im Zyklus oder in einem zeitgesteuerten Programm aufgerufen. Bitte beachten Sie, dass dieser Baustein intern den FC bzw. SFC 192 CP\_S\_R aufruft. Dieser darf nicht überschrieben werden! Der direkte Aufruf eines internen Bausteins führt zu Fehler im entsprechenden Instanz-DB!

#### Parameter

Parameter	Deklaration	Datentyp	Beschreibung
EN_R	IN	BOOL	Freigabe für Daten lesen
R	IN	BOOL	Auftragsabbruch - der laufende Auftrag wird abgebrochen und Empfang wird gesperrt.
LADDR	IN	INT	Logische Basisadresse des CP - entspricht der Adresse aus der Hardware-Konfiguration des CP.
DB_NO	IN	INT	Datenbausteinnummer - Nummer des Empfangs-DB, Null ist nicht erlaubt.
DBB_NO	IN	INT	Datenbytenummer - Empfangsdaten ab Datenbyte $0 \leq DBB\_NO \leq 8190$
L_...	OUT	-	Diese Parameter haben unter ASCII und 3964(R) keine Bedeutung, können aber von ladbaren Protokollen belegt sein.
NDR1 <sup>1</sup>	OUT	BOOL	Auftrag fertig ohne Fehler, Daten übernommen Parameter <i>STATUS</i> = 00h
ERROR <sup>1</sup>	OUT	BOOL	Auftrag fertig mit Fehler Parameter <i>STATUS</i> enthält die Fehlerinformation
LEN <sup>1</sup>	OUT	BOOL	Länge des empfangenen Telegramms in Byte $1 \leq LEN \leq 1024$
STATUS <sup>1</sup>	OUT	WORD	Spezifikation des Fehlers bei <i>ERROR</i> = 1

1) Parameter steht bis zum nächsten Aufruf des FBs zur Verfügung.

#### Auftrag freigeben und abbrechen

- Mit Signalzustand "1" an *EN\_R* wird die Überprüfung, ob Daten vom CP zu lesen sind, freigegeben. Je nach Datenmenge kann eine Datenübertragung über mehrere Programmzyklen laufen.
- Sie können jederzeit mit dem Signalzustand "0" an *EN\_R* eine laufende Übertragung abbrechen. Hierbei wird der abgebrochene Empfangsauftrag mit einer Fehlermeldung (*STATUS*) beendet.
- Solange "0" an *EN\_R* ansteht, ist der Empfang ausgeschaltet. Mit Signalzustand "1" an *R* können Sie einen laufenden Auftrag abbrechen und den FB in den Grundzustand zurückversetzen. Steht statisch der Signalzustand "1" am Eingang *R* an, so ist der Empfang ausgeschaltet.

### Mechanismus zur Anlauf-synchronisation

Der FB 7 besitzen einen Mechanismus für die Anlaufsynchroisation zwischen CPU und CP, der automatisch beim erstmaligen Aufruf des FB ausgeführt wird. Bevor ein angestoßener Auftrag nach einem STOP-RUN-Übergang vom CP bearbeitet werden kann, muss die Anlauf-Synchronisation zwischen CP und CPU abgeschlossen sein. Ein in der Zwischenzeit angestoßener Auftrag wird nach der Synchronisation zum CP übertragen.



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

### Fehleranzeige

- Der Ausgang **NDR** zeigt "Auftrag fertig ohne Fehler/Daten übernommen" an. Bei **ERROR** wird die entsprechende Ereignisnummer in **STATUS** angezeigt. Ist kein Fehler aufgetreten, hat **STATUS** den Wert "0".
- **NDR** und **ERROR/STATUS** werden auch bei **RESET** des FB ausgegeben. Bei einem aufgetretenen Fehler wird das Binärergebnis **BEI** zurückgesetzt. Wird der Baustein ohne Fehler beendet, hat **BEI** den Zustand "1".
- Bitte beachten Sie, dass die Parameter **NDR**, **ERROR** und **STATUS** immer nur für einen Bausteinaufruf verfügbar sind. Zur weiteren Auswertung sollten Sie diese in einen freien Datenbereich kopieren.

### Adressierung

Mit **LADDR** geben Sie die Adresse des anzusprechenden CP an. Dies ist die Adresse, die Sie über die Hardware-Konfiguration für den CP vergeben haben. Bitte beachten Sie beim CP, dass die Basis-Adressen für Ein- und Ausgabe identisch sind.

### Datenbereich

Der FB 7 - P\_RCV\_RK arbeitet mit einem Instanz-DB I\_RCV\_RK zusammen. Dieser hat eine Länge von 60Byte. Die DB-Nr. geben Sie mit dem Aufruf mit. Ein Zugriff auf die Daten im Instanz-DB ist nicht zulässig.

## 5.1.7 FB 8 - P\_SND\_RK - Senden an CP 341

### Beschreibung



*Bitte beachten Sie, dass dieser Baustein ausschließlich für den Einsatz in CPUs von Yaskawa bzw. in S7-300 CPUs von Siemens geeignet ist!*

Der FB 8 - P\_SND\_RK überträgt einen Datenblock von einem Datenbaustein zum CP, spezifiziert durch die Parameter **DB\_NO**, **DBB\_NO** und **LEN**. Hierbei ist der FB zur Datenübertragung im Zyklus oder statisch in einem zeitgesteuerten Programm aufzurufen. Bitte beachten Sie, dass dieser Baustein intern den FC bzw. SFC 192 CP\_S\_R aufruft. Dieser darf nicht überschrieben werden! Der direkte Aufruf eines internen Bausteins führt zu Fehler im entsprechenden Instanz-DB!

### Parameter

Parameter	Deklaration	Datentyp	Beschreibung
SF	IN	CHAR	S = Send, F = Fetch. Bei ASCII und 3964(R) kann hier der Defaultwert "S" für Senden übernommen werden.
REQ	IN	BOOL	Auftragsanstoß bei positiver Flanke
R	IN	BOOL	Auftragsabbruch - der laufende Auftrag wird abgebrochen und Senden wird gesperrt.

Parameter	Deklaration	Datentyp	Beschreibung
LADDR	IN	INT	Logische Basisadresse des CP - entspricht der Adresse aus der Hardware-Konfiguration des CP.
DB_NO	IN	INT	Datenbausteinnummer - Nummer des Sende-DB, Null ist nicht erlaubt.
DBB_NO	IN	INT	Datenbytenummer - Sendedaten ab Datenbyte $0 \leq DBB\_NO \leq 8190$
LEN	IN	INT	Länge des zu sendenden Telegramms in Byte. $1 \leq LEN \leq 1024$
R_...	IN	-	Diese Parameter haben unter ASCII und 3964(R) keine Bedeutung, können aber von ladbaren Protokollen belegt sein. Bei Einsatz unter Modbus ist hier "X" einzutragen.
DONE <sup>1</sup>	OUT	BOOL	Auftrag fertig ohne Fehler, Daten gesendet Parameter <i>STATUS</i> = 00h
ERROR <sup>1</sup>	OUT	BOOL	Auftrag fertig mit Fehler Parameter <i>STATUS</i> enthält die Fehlerinformation
STATUS <sup>1</sup>	OUT	WORD	Spezifikation des Fehlers bei <i>ERROR</i> = 1

1) Parameter steht bis zum nächsten Aufruf des FBs zur Verfügung.

### Auftrag freigeben und abbrechen

- Mit einer positiven Flanke am Eingang *REQ* des FB 8 - P\_SND\_RK wird die Übertragung der Daten angestoßen. Je nach Datenmenge kann eine Datenübertragung über mehrere Programmzyklen laufen.
- Sie können jederzeit mit dem Signalzustand "1" an *R* einen laufenden Auftrag abbrechen und den FB in den Grundzustand zurückversetzen. Bitte beachten Sie, dass hierbei die Daten, die der CP bereits erhalten hat, noch an den Kommunikationspartner gesendet werden.
- Steht statisch der Signalzustand "1" am Eingang *R* an, so ist das Senden ausgeschaltet.

### Mechanismus zur Anlauf-synchronisation

Der FB 8 besitzen einen Mechanismus für die Anlaufsynchronisation zwischen CPU und CP, der automatisch beim erstmaligen Aufruf des FB ausgeführt wird. Bevor ein angestoßener Auftrag nach einem STOP-RUN-Übergang der CPU vom CP bearbeitet werden kann, muss die Anlauf-Synchronisation zwischen CP und CPU abgeschlossen sein. Ein in der Zwischenzeit angestoßener Auftrag wird nach der Synchronisation zum CP übertragen.



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

### Fehleranzeige

- Der Ausgang *DONE* zeigt "Auftragsende ohne Fehler" an. Bei *ERROR* wird die entsprechende Ereignisnummer in *STATUS* angezeigt. Ist kein Fehler aufgetreten, hat *STATUS* den Wert "0".
- *DONE* und *ERROR/STATUS* werden auch bei *RESET* des FB ausgegeben. Bei einem aufgetretenen Fehler wird das Binärergebnis BIE zurückgesetzt. Wird der Baustein ohne Fehler beendet, hat BIE den Zustand "1".
- Bitte beachten Sie, dass die Parameter *DONE*, *ERROR* und *STATUS* immer nur für einen Bausteinaufruf verfügbar sind. Zur weiteren Auswertung sollten Sie diese in einen freien Datenbereich kopieren.

## Adressierung

Mit **LADDR** geben Sie die Adresse des anzusprechenden CP an. Dies ist die Adresse, die Sie über die Hardware-Konfiguration für den CP vergeben haben. Bitte beachten Sie beim CP, dass die Basis-Adressen für Ein- und Ausgabe identisch sind.

## Datenbereich

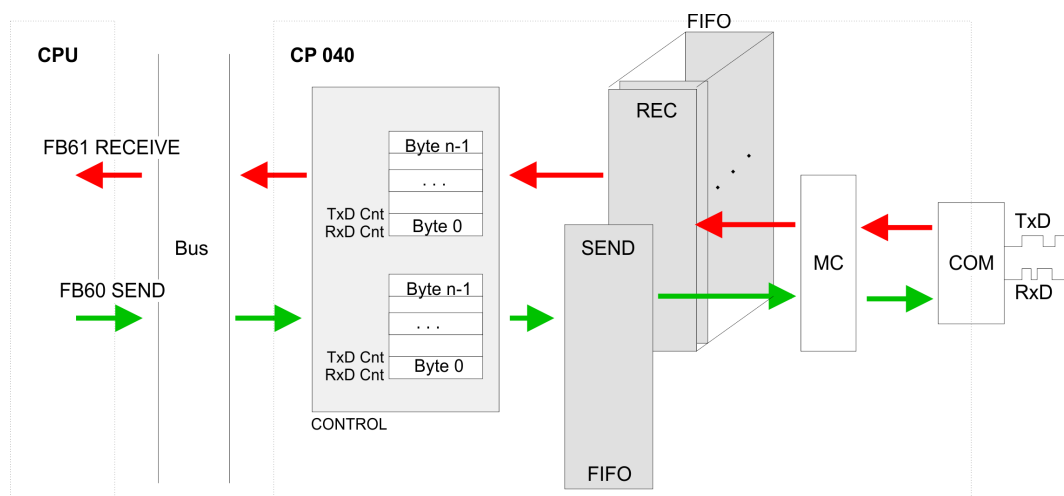
Der FB 8 - P\_SND\_RK arbeitet mit einem Instanz-DB I\_SND\_RK zusammen. Dieser hat eine Länge von 62Byte. Die DB-Nr. geben Sie mit dem Aufruf mit. Ein Zugriff auf die Daten im Instanz-DB ist nicht zulässig.

## 5.2 CP040

### 5.2.1 Übersicht

#### Kommunikationsprinzip

- Durch zyklischen Aufruf von FB 60 SEND und FB 61 RECEIVE bzw. FB 65 CP040\_COM können Sie mit dem CP zyklisch Daten senden und empfangen.
- Auf dem CP erfolgt die Umsetzung der Übertragungsprotokolle zum Kommunikationspartner, welche Sie mittels der Hardwarekonfiguration parametrieren können.
- Ein zu sendendes Telegramm wird in der CPU, abhängig von der IO-Size, in Blöcke unterteilt und über den Datenkanal an den CP übergeben. Im CP werden diese Blöcke im Sendepuffer zusammengesetzt und bei Vollständigkeit des Telegramms über die serielle Schnittstelle gesendet.
- Der Austausch von empfangenen Telegrammen über den Rückwandbus erfolgt asynchron.
- Ist ein komplettes Telegramm über die serielle Schnittstelle eingetroffen, so wird dies in einem 1024Byte großen Ringpuffer abgelegt. Aus der Länge des noch freien Ringpuffers ergibt sich die max. Länge eines Telegramms.
- Je nach Parametrierung können bis zu 250 Telegramme gepuffert werden, wobei deren Gesamtlänge 1024 nicht überschreiten darf.
- Ist der Puffer voll, werden neu ankommende Telegramme verworfen.
- Ein komplettes Telegramm wird in Blöcke, abhängig von der parametrierten IO-Size unterteilt und an den Rückwandbus übergeben.
- Das Zusammensetzen der Datenblöcke hat in der CPU zu erfolgen.
- Da der Datenaustausch über den Rückwandbus asynchron abläuft, wird ein Software-Handshake zwischen dem CP und der CPU eingesetzt. Hierzu besitzen beide Hantierungsbausteine den gemeinsamen Parameter CONTROL. Für diesen Parameter ist das selbe Merker-Byte zu verwenden.



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



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

## 5.2.2 FB 60 - SEND - Senden an System SLIO CP 040

### Beschreibung

Dieser FB dient zur Datenausgabe von der CPU an den System SLIO CP 040. Hierbei legen Sie über die Bezeichner *DB\_NO*, *DBB\_NO* und *LEN* den Sendebereich fest. Über eine positive Flanke an *REQ* wird das Senden angestoßen und die Daten werden gesendet.

### Parameter

Name	Deklaration	Typ	Beschreibung
REQ	IN	BOOL	Sendefreigabe bei positiver Flanke.
R	IN	BOOL	Synchron Reset auslösen.
LADDR / HW_ID	IN	INT / HW_IO	<ul style="list-style-type: none"> <li>■ LADDR <ul style="list-style-type: none"> <li>– Logische Basisadresse des CP.</li> <li>– Bei Einsatz in CPUs von Yaskawa bzw. S7-300 CPUs von Siemens.</li> </ul> </li> <li>■ HW_ID <ul style="list-style-type: none"> <li>– <i>HW-Kennung</i> zur Adressierung des CP.</li> <li>– Bei Einsatz in S7-1500 CPUs von Siemens.</li> </ul> </li> </ul>
DB_NO	IN	INT	Datenbausteinnummer der Sendedaten.
DBB_NO	IN	INT	Datenbytenummer - Sendedaten ab Datenbyte.
LEN	IN	INT	Länge des zu sendenden Telegramms in Byte.
IO_SIZE	IN	WORD	Parametrierte IO Größe des Moduls.
DONE <sup>1</sup>	OUT	BOOL	Sende-Auftrag fertig ohne Fehler.
ERROR <sup>1</sup>	OUT	BOOL	Sende-Auftrag fertig mit Fehler. Parameter <i>STATUS</i> enthält die Fehlerinformation.
STATUS <sup>1</sup>	OUT	WORD	Spezifikation des Fehlers bei <i>ERROR</i> = 1.
CONTROL	IN_OUT	BYTE	Geteiltes Byte mit RECEIVE Hantierungsbaustein: SEND (Bit 0 ... 3), RECEIVE (Bit 4 ... 7).

1) Parameter steht bis zum nächsten Aufruf des FBs zur Verfügung.

### REQ

Request - Sendefreigabe:

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

<b>R</b>	<p>Synchron Reset:</p> <ul style="list-style-type: none"><li>■ Für die Initialisierung ist SEND im Anlauf-OB mit allen Parametern und mit gesetztem <i>R</i> einmalig aufzurufen.</li><li>■ Sie können jederzeit mit dem Signalzustand "1" an <i>R</i> einen laufenden Auftrag abbrechen und den FB in den Grundzustand zurückversetzen. Bitte beachten Sie, dass hierbei die Daten, die der CP bereits erhalten hat, noch an den Kommunikationspartner gesendet werden.</li><li>■ Steht statisch der Signalzustand "1" am Eingang <i>R</i> an, so ist das Senden ausgeschaltet.</li></ul>
<b>LADDR</b>	<p>Peripherieadresse:</p> <ul style="list-style-type: none"><li>■ Dieser Parameter ist nur bei Einsatz in CPUs von Yaskawa bzw. S7-300 CPUs von Siemens verfügbar.</li><li>■ Mit <i>LADDR</i> geben Sie die Adresse des anzusprechenden CP an. Dies ist die Adresse, die Sie über die Hardware- Konfiguration für den CP vergeben haben.</li></ul>
<b>HW_ID</b>	<p>HW-Kennung:</p> <ul style="list-style-type: none"><li>■ Dieser Parameter ist nur bei Einsatz in S7-1500 CPUs von Siemens verfügbar.</li><li>■ Geben Sie unter <i>HW_ID</i> die <i>HW-Kennung</i> an, mit deren Hilfe Ihr Modul entsprechend adressiert werden kann. ➔ "<a href="#">HW-Kennung - HW_ID</a>"...Seite 10</li></ul>
<b>DB_NO</b>	<p>Datenbaustein-Nummer:</p> <ul style="list-style-type: none"><li>■ Nummer des Datenbausteins, der die zu sendenden Daten beinhaltet.</li><li>■ Null ist nicht erlaubt.</li></ul>
<b>DBB_NO</b>	<p>Datenbyte-Nummer:</p> <ul style="list-style-type: none"><li>■ Nummer des Datenbytes im Datenbaustein, ab dem die Sendedaten abgelegt sind.</li></ul>
<b>LEN</b>	<p>Länge:</p> <ul style="list-style-type: none"><li>■ Länge der Nutzdaten, welche zu übertragen sind.</li><li>■ Es gilt: <math>1 \leq LEN \leq 1024</math>.</li></ul>
<b>IO_SIZE</b>	<p>Größe E/A-Bereich:</p> <ul style="list-style-type: none"><li>■ Geben Sie hier die Größe des E/A-Bereichs an. Abhängig vom übergeordneten System belegt der CP für Ein- und Ausgabe jeweils folgende Anzahl an Bytes im Adress-Bereich:<ul style="list-style-type: none"><li>– PROFIBUS: 8Byte, 20Byte oder 60Byte wählbar</li><li>– PROFINET: 20Byte oder 60Byte wählbar</li><li>– CANopen: 8Byte</li><li>– EtherCAT: 60Byte</li><li>– DeviceNET: 60Byte</li><li>– ModbusTCP: 60Byte</li></ul></li></ul>
<b>DONE</b>	<p>DONE:</p> <ul style="list-style-type: none"><li>■ wird gesetzt bei Auftrag fertig ohne Fehler und <i>STATUS</i> = 0000h.</li></ul>



**ERROR****ERROR:**

- wird gesetzt bei Auftrag fertig mit Fehler. Hierbei enthält *STATUS* die entsprechende Fehlerinformation.

**STATUS**

Bei fehlerfreier Funktion *STATUS* = 0000h oder 8181h. Im Fehlerfall finden Sie hier den entsprechenden Fehlercode. Solange *ERROR* gesetzt ist bleibt der Wert in *STATUS* bestehen. Folgende Statusmeldungen sind möglich:

STATUS	Beschreibung
0000h	Kein Fehler vorhanden.
0202h	Mögliche Fehlerquellen: <ul style="list-style-type: none"> <li>■ Hantierungsbaustein und CP sind nicht synchron (Abhilfe: Synchron Reset auslösen)</li> <li>■ <i>IO_SIZE</i> ist ungültig (<i>IO_SIZE</i> = 0 oder <i>IO_SIZE</i> &gt; 60).</li> </ul>
0301h	DB ist ungültig.
0517h	<i>LEN</i> ist ungültig ( <i>LEN</i> = 0 oder <i>LEN</i> > 1024).
070Ah	Übertragung fehlgeschlagen, Partner antwortet nicht, oder hat den Auftrag negativ quittiert.
8090h	<i>HW_ID</i> ist unbekannt.
80A0h	Beim Zugriff auf die Peripherie wurde ein Zugriffsfehler erkannt.
80A1h	
8181h	Auftrag läuft (Status und keine Fehlermeldung).
8323h	Sende-DB vorhanden, aber zu kurz.
833Ah	Sende-DB nicht lesbar (DB nicht vorhanden oder DB im optimierten Bausteinzugriff). ➔ " <a href="#">Kein optimierter Bausteinzugriff</a> " ...Seite 7

**CONTROL**

Die Hantierungsbausteine SEND und RECEIVE verwenden für den Handshake den gemeinsamen Parameter *CONTROL*. Weisen Sie diesem Parameter ein gemeinsames Merker-Byte zu.

**Fehleranzeige**

- Der Ausgang *DONE* zeigt "Auftragsende ohne Fehler" an. Bei *ERROR* wird die entsprechende Ereignisnummer in *STATUS* angezeigt. Ist kein Fehler aufgetreten, hat *STATUS* den Wert "0".
- *DONE*, *ERROR* und *STATUS* werden auch bei RESET des FB ausgegeben. Bei einem aufgetretenen Fehler wird das Binärergebnis BIE zurückgesetzt. Wird der Baustein ohne Fehler beendet, hat BIE den Zustand "1".
- Bitte beachten Sie, dass die Parameter *DONE*, *ERROR* und *STATUS* immer nur für einen Bausteinanruf verfügbar sind. Zur weiteren Auswertung sollten Sie diese in einen freien Datenbereich kopieren.

**5.2.3 FB 61 - RECEIVE - Empfangen von System SLIO CP 040****Beschreibung**

Dieser FB dient zum Datenempfang vom System SLIO CP 040. Hierbei legen Sie über die Bezeichner *DB\_NO* und *DBB\_NO* den Empfangsbereich fest. Die Länge des eingelesenen Telegramms wird in *LEN* abgelegt.

## Parameter

Parameter	Deklaration	Datentyp	Beschreibung
EN_R	IN	BOOL	Freigabe zum Daten lesen.
R	IN	BOOL	Synchron Reset auslösen.
LADDR / HW_ID	IN	INT / HW_IO	<ul style="list-style-type: none"> <li>■ LADDR <ul style="list-style-type: none"> <li>– Logische Basisadresse des CP.</li> <li>– Bei Einsatz in CPUs von Yaskawa bzw. S7-300 CPUs von Siemens.</li> </ul> </li> <li>■ HW_ID <ul style="list-style-type: none"> <li>– <i>HW-Kennung</i> zur Adressierung des CP.</li> <li>– Bei Einsatz in S7-1500 CPUs von Siemens.</li> </ul> </li> </ul>
DB_NO	IN	INT	Datenbausteinnummer der Empfangsdaten.
DBB_NO	IN	INT	Datenbytenummer - Empfangsdaten ab Datenbyte.
IO_SIZE	IN	WORD	Parametrierte I/O Größe des Moduls.
LEN	OUT	INT	Länge des empfangenen Telegramms in Byte.
NDR <sup>1</sup>	OUT	BOOL	Empfangs-Auftrag fertig ohne Fehler.
ERROR <sup>1</sup>	OUT	BOOL	Empfangs-Auftrag fertig mit Fehler. Parameter <i>STATUS</i> enthält die Fehlerinformation.
STATUS <sup>1</sup>	OUT	WORD	Spezifikation des Fehlers bei <i>ERROR</i> = 1.
CONTROL	IN_OUT	BYTE	Geteiltes Byte mit SEND Hantierungsbaustein: SEND (Bit 0 ... 3), RECEIVE (Bit 4 ... 7).

1) Parameter steht bis zum nächsten Aufruf des FBs zur Verfügung.

## EN\_R

Enable Receive - Lesefreigabe:

- Mit Signalzustand "1" an *EN\_R* wird die Überprüfung, ob Daten vom CP zu lesen sind, freigegeben. Je nach Datenmenge kann eine Datenübertragung über mehrere Programmzyklen laufen.
- Sie können jederzeit mit dem Signalzustand "0" an *EN\_R* eine laufende Übertragung abbrechen. Hierbei wird der abgebrochene Empfangsauftrag mit einer Fehlermeldung (*STATUS*) beendet.
- Solange "0" an *EN\_R* ansteht, ist der Empfang ausgeschaltet.

## R

Synchron Reset:

- Für die Initialisierung ist RECEIVE im Anlauf-OB mit allen Parametern und mit gesetztem *R* einmalig aufzurufen.
- Sie können jederzeit mit dem Signalzustand "1" an *R* einen laufenden Auftrag abbrechen und den FB in den Grundzustand zurückversetzen.
- Steht statisch der Signalzustand "1" am Eingang *R* an, so ist der Empfang ausgeschaltet.

## LADDR

Peripherieadresse:

- Dieser Parameter ist nur bei Einsatz in CPUs von Yaskawa bzw. S7-300 CPUs von Siemens verfügbar.
- Mit *LADDR* geben Sie die Adresse des anzusprechenden CP an. Dies ist die Adresse, die Sie über die Hardware- Konfiguration für den CP vergeben haben.

**HW\_ID**

HW-Kennung:

- Dieser Parameter ist nur bei Einsatz in S7-1500 CPUs von Siemens verfügbar.
- Geben Sie unter *HW\_ID* die *HW-Kennung* an, mit deren Hilfe Ihr Modul entsprechend adressiert werden kann. → "*HW-Kennung - HW\_ID*"...Seite 10

**DB\_NO**

Datenbaustein-Nummer:

- Nummer des Datenbausteins, der die gelesenen Daten beinhaltet.
- Null ist nicht erlaubt.

**DBB\_NO**

Datenbyte-Nummer:

- Nummer des Datenbytes im Datenbaustein, ab dem die empfangenen Daten abgelegt werden sollen.

**IO\_SIZE**

Größe E/A-Bereich:

- Geben Sie hier die Größe des E/A-Bereichs an. Abhängig vom übergeordneten System belegt der CP für Ein- und Ausgabe jeweils folgende Anzahl an Bytes im Adress-Bereich:
  - PROFIBUS: 8Byte, 20Byte oder 60Byte wählbar
  - PROFINET: 20Byte oder 60Byte wählbar
  - CANopen: 8Byte
  - EtherCAT: 60Byte
  - DeviceNET: 60Byte
  - ModbusTCP: 60Byte

**LEN**

Länge:

- Länge der Nutzdaten, welche zu übertragen sind.
- Es gilt:  $1 \leq LEN \leq 1024$ .

**NDR**

New data ready:

- Neu empfangene Daten stehen für die CPU im CP bereit.

**ERROR**

ERROR:

- wird gesetzt bei Auftrag fertig mit Fehler. Hierbei enthält *STATUS* die entsprechende Fehlerinformation.

**STATUS**

Bei fehlerfreier Funktion *STATUS* = 0000h oder 8181h. Im Fehlerfall finden Sie hier den entsprechenden Fehlercode. Solange *ERROR* gesetzt ist bleibt der Wert in *STATUS* bestehen. Folgende Statusmeldungen sind möglich:

STATUS	Beschreibung
0000h	Kein Fehler vorhanden.
0202h	Mögliche Fehlerquellen: <ul style="list-style-type: none"> <li>■ Hantierungsbaustein und CP sind nicht synchron (Abhilfe: Synchron Reset auslösen).</li> <li>■ <i>IO_SIZE</i> ist ungültig (<math>IO\_SIZE = 0</math> oder <math>IO\_SIZE &gt; 60</math>).</li> </ul>

STATUS	Beschreibung
0301h	DB ist ungültig.
070Ah	Übertragung fehlgeschlagen, Partner antwortet nicht, oder hat den Auftrag negativ quittiert.
8090h	HW_ID ist unbekannt.
80A0h	Beim Zugriff auf die Peripherie wurde ein Zugriffsfehler erkannt.
80A1h	
8181h	Auftrag läuft (Status und keine Fehlermeldung).
8323h	Empfangs-DB vorhanden, aber zu kurz.
833Ah	Empfangs-DB nicht beschreibbar (DB nicht vorhanden oder DB im optimierten Bausteinzugriff). → <a href="#">"Kein optimierter Bausteinzugriff" ...Seite 7</a>

## CONTROL

- Die Hantierungsbausteine SEND und RECEIVE verwenden für den Handshake den gemeinsamen Parameter CONTROL.
- Weisen Sie diesem Parameter ein gemeinsames Merker-Byte zu.

## Fehleranzeige

- Der Ausgang *NDR* zeigt "Auftrag fertig ohne Fehler/Daten übernommen" an. Bei *ERROR* wird die entsprechende Ereignisnummer in *STATUS* angezeigt. Ist kein Fehler aufgetreten, hat *STATUS* den Wert "0".
- *NDR*, *ERROR* und *STATUS* werden auch bei RESET des FBs ausgegeben. Bei einem aufgetretenen Fehler wird das Binärergebnis BIE zurückgesetzt. Wird der Baustein ohne Fehler beendet, hat BIE den Zustand "1".
- Bitte beachten Sie, dass die Parameter *NDR*, *ERROR* und *STATUS* immer nur für einen Bausteinaufruf verfügbar sind. Zur weiteren Auswertung sollten Sie diese in einen freien Datenbereich kopieren.

## 5.2.4 FB 65 - CP040\_COM - Kommunikation SLIO CP 040

### Beschreibung

Dieser FB dient zur Datenübergabe von der System SLIO CPU an den CP 040. Hierbei legen Sie über die Bezeichner *DB\_NO\_SEND*, *DB\_NO\_RECV* den Sende-/Empfangsbereich fest. Über eine positive Flanke an *REQ\_SEND* wird das Senden angestoßen und die Daten werden gesendet. Über *EN\_RECV* wird die Freigabe zum Daten empfangen gegeben.

### Parameter bei Einsatz von CPUs von Yaskawa bzw. S7-300 CPUs von Siemens

Name	Deklaration	Typ	Beschreibung
REQ_SEND	IN	BOOL	Anstoß Sende-Auftrag bei positiver Flanke.
EN_RECV	IN	BOOL	Freigabe für Daten empfangen.
RESET	IN	BOOL	Synchron Reset auslösen.
ADDR_OUT	IN	INT	Ausgangsadresse des CP aus der HW-Konfiguration.
ADDR_IN	IN	INT	Eingangsadresse des CP aus der HW-Konfiguration.
IO_SIZE	IN	WORD	Parametrierte IO Größe des Moduls.
DB_NO_SEND	IN	INT	Datenbaustein-Nummer - Nummer des Sende-DB. Null ist nicht erlaubt.

Name	Deklaration	Typ	Beschreibung
DBB_NO_SEND	IN	INT	Datenbyte-Nummer - Sendedaten ab Datenbyte.
LEN_SEND	IN	INT	Länge des zu sendenden Telegramms in Byte. $1 \leq LEN\_SEND \leq 1024$
DB_NO_RECV	IN	INT	Datenbaustein-Nummer - Nummer des Empfang-DB. Null ist nicht erlaubt.
DBB_NO_RECV	IN	INT	Datenbyte-Nummer - Empfangsdaten ab Datenbyte.
DONE_SEND <sup>1</sup>	OUT	BOOL	Sende-Auftrag fertig ohne Fehler. Daten gesendet: Parameter <i>STATUS_SEND</i> = 0000h.
ERROR_SEND <sup>1</sup>	OUT	BOOL	Sende-Auftrag fertig mit Fehler. Parameter <i>STATUS_SEND</i> enthält die Fehlerinformation.
NDR_RCV <sup>1</sup>	OUT	BOOL	Empfangs-Auftrag fertig ohne Fehler. Daten gesendet: Parameter <i>STATUS_RCV</i> = 0000h. Bleibt für einen Zyklus stehen.
ERROR_RCV <sup>1</sup>	OUT	BOOL	Empfangs-Auftrag fertig mit Fehler. Parameter <i>STATUS_RCV</i> enthält die Fehlerinformation.
STATUS_SEND <sup>1</sup>	OUT	WORD	Spezifikation des Fehlers beim Senden bei <i>ERROR_SEND</i> = 1
LEN_RCV	OUT	INT	Länge des empfangenen Telegramms in Byte. $1 \leq LEN\_RCV \leq 1024$
STATUS_RCV <sup>1</sup>	OUT	WORD	Spezifikation des Fehlers beim Empfangen bei <i>ERROR_RCV</i> = 1

1) Parameter steht bis zum nächsten Aufruf des FBs zur Verfügung.

#### Parameter bei Einsatz von S7-1500 CPUs von Siemens

Name	Deklaration	Typ	Beschreibung
REQ_SEND	IN	BOOL	Anstoß Sende-Auftrag bei positiver Flanke.
EN_RECV	IN	BOOL	Freigabe für Daten empfangen.
RESET	IN	BOOL	Synchron Reset auslösen.
HW_ID	IN	HW_IO	<i>HW-Kennung</i> zur Adressierung des CP.
IO_SIZE	IN	WORD	Parametrierte IO Größe des Moduls.
DB_NO_SEND	IN	INT	Datenbaustein-Nummer - Nummer des Sende-DB. Null ist nicht erlaubt.
DBB_NO_SEND	IN	INT	Datenbyte-Nummer - Sendedaten ab Datenbyte.
LEN_SEND	IN	INT	Länge des zu sendenden Telegramms in Byte. $1 \leq LEN\_SEND \leq 1024$
DB_NO_RECV	IN	INT	Datenbaustein-Nummer - Nummer des Empfang-DB. Null ist nicht erlaubt.
DBB_NO_RECV	IN	INT	Datenbyte-Nummer - Empfangsdaten ab Datenbyte.
DONE_SEND <sup>1</sup>	OUT	BOOL	Sende-Auftrag fertig ohne Fehler. Daten gesendet: Parameter <i>STATUS_SEND</i> = 0000h.

Name	Deklaration	Typ	Beschreibung
ERROR_SEND <sup>1</sup>	OUT	BOOL	Sende-Auftrag fertig mit Fehler. Parameter <i>STATUS_SEND</i> enthält die Fehlerinformation.
NDR_RCV <sup>1</sup>	OUT	BOOL	Empfangs-Auftrag fertig ohne Fehler. Daten gesendet: Parameter <i>STATUS_RCV</i> = 0000h. Bleibt für einen Zyklus stehen.
ERROR_RCV <sup>1</sup>	OUT	BOOL	Empfangs-Auftrag fertig mit Fehler. Parameter <i>STATUS_RCV</i> enthält die Fehlerinformation.
STATUS_SEND <sup>1</sup>	OUT	WORD	Spezifikation des Fehlers beim Senden bei <i>ERROR_SEND</i> = 1
LEN_RCV	OUT	INT	Länge des empfangenen Telegramms in Byte. $1 \leq LEN\_RCV \leq 1024$
STATUS_RCV <sup>1</sup>	OUT	WORD	Spezifikation des Fehlers beim Empfangen bei <i>ERROR_RCV</i> = 1

1) Parameter steht bis zum nächsten Aufruf des FBs zur Verfügung.

## REQ\_SEND

Request - Sendefreigabe:

- Mit einer positiven Flanke am Eingang *REQ\_SEND* wird die Übertragung der Daten angestoßen.
- Je nach Datenmenge kann eine Datenübertragung über mehrere Programmzyklen laufen.

## EN\_RECV

Freigabe für Daten empfangen.

## RESET

Synchron Reset:

- Für die Initialisierung ist der FB 65 im Anlauf-OB mit allen Parametern und mit gesetztem *RESET* einmalig aufzurufen.
- Sie können jederzeit mit dem Signalzustand "1" an *RESET* einen laufenden Auftrag abbrechen und den FB in den Grundzustand zurückversetzen. Bitte beachten Sie, dass hierbei die Daten, die der CP bereits erhalten hat, noch an den Kommunikationspartner gesendet werden.
- Steht statisch der Signalzustand "1" am Eingang *RESET* an, so ist das Senden ausgeschaltet.

## ADDR\_IN

Eingangs-Peripherieadresse:

- Dieser Parameter ist nur bei Einsatz in CPUs von Yaskawa bzw. S7-300 CPUs von Siemens verfügbar.
- Mit *ADDR\_IN* geben Sie die Eingangs-Adresse des anzusprechenden CP an. Dies ist die Adresse, die Sie über die Hardware-Konfiguration für den CP vergeben haben.

## ADDR\_OUT

Ausgangs-Peripherieadresse:

- Dieser Parameter ist nur bei Einsatz in CPUs von Yaskawa bzw. S7-300 CPUs von Siemens verfügbar.
- Mit *ADDR\_OUT* geben Sie die Ausgangs-Adresse des anzusprechenden CP an. Dies ist die Adresse, die Sie über die Hardware-Konfiguration für den CP vergeben haben.

HW_ID	<p>HW-Kennung:</p> <ul style="list-style-type: none"><li>■ Dieser Parameter ist nur bei Einsatz in S7-1500 CPUs von Siemens verfügbar.</li><li>■ Geben Sie unter <i>HW_ID</i> die <i>HW-Kennung</i> an, mit deren Hilfe Ihr Modul entsprechend adressiert werden kann. → "<i>HW-Kennung - HW_ID</i>"...Seite 10</li></ul>
DB_NO_SEND	<p>Datenbaustein-Nummer SEND:</p> <ul style="list-style-type: none"><li>■ Nummer des Datenbausteins, der die zu sendenden Daten beinhaltet.</li><li>■ Null ist nicht erlaubt.</li></ul>
DBB_NO_SEND	<p>Datenbyte-Nummer SEND:</p> <ul style="list-style-type: none"><li>■ Nummer des Datenbytes im Datenbaustein, ab dem die Sendedaten abgelegt sind.</li></ul>
LEN_SEND	<p>Länge SEND:</p> <ul style="list-style-type: none"><li>■ Länge der Nutzdaten, welche zu übertragen sind.</li><li>■ Es gilt: <math>1 \leq LEN\_SEND \leq 1024</math>.</li></ul>
DB_NO_RECV	<p>Datenbaustein-Nummer RECV:</p> <ul style="list-style-type: none"><li>■ Nummer des Empfang-DBs.</li><li>■ Null ist nicht erlaubt.</li></ul>
DBB_NO_RECV	<p>Datenbyte-Nummer RECV:</p> <ul style="list-style-type: none"><li>■ Nummer des Datenbytes im Datenbaustein, ab dem die Empfangsdaten abgelegt sind.</li></ul>
IO_SIZE	<p>Größe E/A-Bereich:</p> <ul style="list-style-type: none"><li>■ Geben Sie hier die Größe des E/A-Bereichs an. Abhängig vom übergeordneten System belegt der CP für Ein- und Ausgabe jeweils folgende Anzahl an Bytes im Adress-Bereich:<ul style="list-style-type: none"><li>– SLIO CPU: 8Byte, 20Byte oder 60Byte wählbar</li><li>– PROFIBUS: 8Byte, 20Byte oder 60Byte wählbar</li><li>– PROFINET: 20Byte oder 60Byte wählbar</li><li>– CANopen: 8Byte</li><li>– EtherCAT: 60Byte</li><li>– DeviceNET: 60Byte</li><li>– ModbusTCP: 60Byte</li></ul></li></ul>
DONE_SEND	<p>Sende-Auftrag fertig ohne Fehler, Daten gesendet. Parameter <i>STATUS_SEND</i> = 0000h.</p>
ERROR_SEND	<p><i>ERROR_SEND</i> wird gesetzt bei Auftrag fertig mit Fehler. Hierbei enthält <i>STATUS_SEND</i> die entsprechende Fehlerinformation.</p>
STATUS_SEND	<p>Bei fehlerfreier Funktion ist <i>STATUS_SEND</i> 0000h oder 8181h. Im Fehlerfall finden Sie hier den entsprechenden Fehlercode. Solange <i>ERROR_SEND</i> gesetzt ist bleibt der Wert in <i>STATUS_SEND</i> bestehen. Folgende Statusmeldungen sind möglich:</p>

STATUS	Beschreibung
0000h	Kein Fehler vorhanden.
0202h	<i>IO_SIZE</i> ist ungültig ( <i>IO_SIZE</i> = 0 oder <i>IO_SIZE</i> > 60).
0301h	DB ist ungültig.
070Ah	Übertragung fehlgeschlagen, Partner antwortet nicht, oder hat den Auftrag negativ quittiert
0517h	<i>LEN</i> ist ungültig ( <i>LEN</i> = 0 oder <i>LEN</i> > 1024).
8090h	<i>HW_ID</i> ist unbekannt.
80A0h	Beim Zugriff auf die Peripherie wurde ein Zugriffsfehler erkannt.
80A1h	
8181h	Auftrag läuft (Status und keine Fehlermeldung).
8323h	Sende-DB vorhanden, aber zu kurz.
833Ah	Sende-DB nicht lesbar (DB nicht vorhanden oder DB im optimierten Bausteinzugriff. → "Kein optimierter Bausteinzugriff"...Seite 7)

#### LEN\_RCV

Länge Receive:

- Länge des empfangenen Telegramms in Byte.
- $1 \leq \text{LEN\_RCV} \leq 1024$

#### NDR\_RCV

New data ready:

- Neu empfangene Daten stehen im Empfangs-DB zur Verfügung. *NDR\_RCV* bleibt für einen Zyklus stehen.
- Daten ohne Fehler empfangen: Parameter *STATUS\_RCV* = 0000h.

#### ERROR\_RCV

*ERROR\_RCV* wird gesetzt bei Auftrag fertig mit Fehler. Hierbei enthält *STATUS\_REC* die entsprechende Fehlerinformation.

#### STATUS\_RCV

Bei fehlerfreier Funktion ist *STATUS\_RCV* 0000h oder 8181h. Im Fehlerfall finden Sie hier den entsprechenden Fehlercode. Solange *ERROR\_RCV* gesetzt ist bleibt der Wert in *STATUS\_RCV* bestehen. Folgende Statusmeldungen sind möglich:

STATUS	Beschreibung
0000h	Kein Fehler vorhanden.
0202h	<i>IO_SIZE</i> ist ungültig ( <i>IO_SIZE</i> = 0 oder <i>IO_SIZE</i> > 60).
0301h	DB ist ungültig.
070Ah	Übertragung fehlgeschlagen, Partner antwortet nicht, oder hat den Auftrag negativ quittiert.
080Ah	Ein freier Empfangspuffer ist nicht vorhanden.
080Ch	Fehlerhaftes Zeichen empfangen (Zeichenrahmen- oder Paritätsfehler)
8090h	<i>HW_ID</i> ist unbekannt.
80A0h	Beim Zugriff auf die Peripherie wurde ein Zugriffsfehler erkannt.



STATUS	Beschreibung
80A1h	
8181h	Auftrag läuft (Status und keine Fehlermeldung).
8323h	Empfangs-DB vorhanden, aber zu kurz.
833Ah	Empfangs-DB nicht beschreibbar (DB nicht vorhanden oder DB im optimierten Bausteinzugriff). → "Kein optimierter Bausteinzugriff" ...Seite 7

### Fehleranzeige

- Der Ausgang *DONE\_SEND* zeigt "Auftragsende ohne Fehler" an.
- Der Ausgang *NDR\_RCV* zeigt "Auftragsempfang ohne Fehler" an.
- Bei *ERROR\_SEND* oder *ERROR\_RCV* wird die entsprechende Ereignisnummer in *STATUS\_SEND* oder *STATUS\_RCV* angezeigt. Ist kein Fehler aufgetreten, hat *STATUS\_SEND* und *STATUS\_RCV* den Wert 0000h.
- *DONE\_SEND*, *NDR\_RCV*, *ERROR\_SEND*, *ERROR\_RCV* und *STATUS\_SEND*, *STATUS\_RCV* werden auch bei RESET des FBs ausgegeben. Bei einem aufgetretenen Fehler wird das Binärergebnis BIE zurückgesetzt. Wird der Baustein ohne Fehler beendet, hat BIE den Zustand "1".
- Bitte beachten Sie, dass die Parameter *DONE\_SEND*, *NDR\_RCV*, *ERROR\_SEND*, *ERROR\_RCV* und *STATUS\_SEND*, *STATUS\_RCV* immer nur für einen Baustein aufrufbar sind. Zur weiteren Auswertung sollten Sie diese in einen freien Datenbereich kopieren.

## 5.3 CP240

### 5.3.1 FC 0 - SEND\_ASCII\_STX\_3964 - Senden an CP 240

#### Beschreibung

Dieser FC dient zur Datenausgabe von der CPU an den CP 240. Hierbei legen Sie über die Bezeichner *\_DB*, *ABD* und *ANZ* den Sendebereich fest. Über das Bit *FRG* wird der Sendeanstoß gesetzt und die Daten werden gesendet. Nach dem Übertragen der Daten setzt der Hantierungsbaustein das Bit *FRG* wieder zurück.

#### Parameter

Name	Deklaration	Typ	Beschreibung
ADR	IN	INT	Peripherieadresse
_DB	IN	BLOCK_DB / DB_ANY	_DB <ul style="list-style-type: none"> <li>■ DB-Nummer vom Typ BLOCK_DB mit den Sendedaten bei Einsatz in CPUs von Yaskawa bzw. in S7-300 CPUs von Siemens.</li> <li>■ Zeiger vom Typ DB_ANY auf den DB mit den Sendedaten bei Einsatz in S7-1500 CPUs von Siemens.</li> </ul>
ABD	IN	WORD	Nummer des 1. Datenworts.
ANZ	IN	WORD	Anzahl der Bytes.
PAFE	OUT / IN_OUT	BYTE	PAFE (0 = OK) <ul style="list-style-type: none"> <li>■ Parametrierfehler Code vom Typ BYTE als OUT-Parameter bei Einsatz in CPUs von Yaskawa bzw. in S7-300 CPUs von Siemens.</li> <li>■ Parametrierfehler Code vom Typ BYTE als IN_OUT-Parameter bei Einsatz in S7-1500 CPUs von Siemens.</li> </ul>

Name	Deklaration	Typ	Beschreibung
FRG	IN_OUT	BOOL	Auftrag anstoßen.
GESE	IN_OUT	WORD	Wird intern verwendet.
ANZ_INT	IN_OUT	WORD	Wird intern verwendet.
ENDE_KOM	IN_OUT	BOOL	Wird intern verwendet.
LETZTER_BLOCK	IN_OUT	BOOL	Wird intern verwendet.
SENDEN_LAEUFT	IN_OUT	BOOL	Status der Funktion.
FEHLER_KOM	IN_OUT	BOOL	Wird intern verwendet.

**ADR** Peripherieadresse unter welcher der CP 240 anzusprechen ist. Über die Hardware-Konfiguration bestimmen Sie die Peripherieadresse.

**\_DB**

- Nummer vom *Typ* BLOCK\_DB des Datenbausteins, welcher die Sendedaten für den CP beinhaltet, bei Einsatz in CPUs von Yaskawa bzw. in S7-300 CPUs von Siemens.
- Zeiger vom *Typ* DB\_ANY auf den Datenbaustein, welcher die Sendedaten für den CP beinhaltet, bei Einsatz in S7-1500 CPUs von Siemens.

**ABD** Wortvariable, welche die Nummer des Datenworts enthält, ab dem die auszugebenden Zeichen abgelegt sind.

**ANZ** Anzahl der Bytes, die zu übertragen sind.

**PAFE** Alle Bits dieses Merker-Bytes sind bei richtiger Funktion "0". Bei Fehlfunktion wird ein Fehlercode eingetragen. Die Fehlerangabe ist selbstquittierend, d.h. nach Beseitigung der Fehlerursache wird das Byte wieder auf "0" gesetzt. Folgende Fehler sind möglich:

- 1 = Datenbaustein nicht vorhanden
- 2 = Datenbaustein zu kurz
- 3 = Datenbausteinnummer nicht im gültigen Bereich

**FRG Sendefreigabe** Bei *FRG* = "1" werden die über *\_DB*, *ADB* und *ANZ* definieren Daten einmalig an den über *ADR* adressierten CP übertragen. Nach der Übertragung wird *FRG* wieder zurückgesetzt. Ist beim Aufruf *FRG* = "0", wird der Baustein sofort wieder verlassen!

**GESE, ANZ\_INT, ENDE\_KOM, LETZTER\_BLOCK, SENDEN\_LAEUFT, FEHLER\_KOM** Diese Parameter werden intern verwendet. Sie dienen dem Informationsaustausch zwischen den Hantierungsbausteinen. Für den Einsatz des SYNCHRON\_RESET (FC9) sind die Steuerbits *FRG*, *ENDE\_KOM*, *LETZTER\_BLOCK*, *SENDEN\_LAEUFT* und *FEHLER\_KOM* immer in einem Merker-Byte abzulegen.

### 5.3.2 FC 1 - RECEIVE\_ASCII\_STX\_3964 - Empfangen von CP 240

**Beschreibung** Dieser FC dient zum Datenempfang vom CP 240. Hierbei legen Sie über die Bezeichner *\_DB* und *ABD* den Empfangsbereich fest. Ist der Ausgang *EMFR* gesetzt, so ist ein neues Telegramm komplett eingelesen worden. Die Länge des eingelesenen Telegramms wird in *ANZ* abgelegt. Nach der Auswertung des Telegramms ist dieses Bit vom Anwender zurückzusetzen, da ansonsten kein weiteres Telegramm in der CPU übernommen werden kann.

## Parameter

Name	Deklaration	Typ	Beschreibung
ADR	IN	INT	Peripherieadresse
_DB	IN	BLOCK_DB / DB_ANY	_DB <ul style="list-style-type: none"> <li>■ DB-Nummer vom <i>Typ</i> BLOCK_DB mit den Empfangsdaten bei Einsatz in CPUs von Yaskawa bzw. in S7-300 CPUs von Siemens.</li> <li>■ Zeiger vom <i>Typ</i> DB_ANY auf den DB mit den Empfangsdaten bei Einsatz in S7-1500 CPUs von Siemens.</li> </ul>
ABD	IN	WORD	Nummer des 1. Datenworts.
ANZ	OUT / IN_OUT	WORD	ANZ <ul style="list-style-type: none"> <li>■ Anzahl der empfangenen Bytes vom <i>Typ</i> WORD als OUT-Parameter bei Einsatz in CPUs von Yaskawa bzw. in S7-300 CPUs von Siemens.</li> <li>■ Anzahl der empfangenen Bytes vom <i>Typ</i> WORD als IN_OUT-Parameter bei Einsatz in S7-1500 CPUs von Siemens.</li> </ul>
PAFE	OUT / IN_OUT	BYTE	PAFE (0 = OK) <ul style="list-style-type: none"> <li>■ Parametrierfehler Code vom <i>Typ</i> BYTE als OUT-Parameter bei Einsatz in CPUs von Yaskawa bzw. in S7-300 CPUs von Siemens.</li> <li>■ Parametrierfehler Code vom <i>Typ</i> BYTE als IN_OUT-Parameter bei Einsatz in S7-1500 CPUs von Siemens.</li> </ul>
EMFR	IN_OUT	BOOL	Empfangsbestätigung
GEEM	IN_OUT	WORD	Wird intern verwendet.
ANZ_INT	IN_OUT	WORD	Wird intern verwendet.
EMPF_LAEUFT	IN_OUT	BOOL	Status der Funktion.
LETZTER_BLOCK	IN_OUT	BOOL	Wird intern verwendet.
FEHLER_EMPF	IN_OUT	BOOL	Wird intern verwendet.
OFFSET	IN_OUT	WORD	Wird intern verwendet.

**ADR** Peripherieadresse unter welcher der CP 240 anzusprechen ist. Über die Hardware-Konfiguration bestimmen Sie die Peripherieadresse.

**\_DB**

- Nummer vom *Typ* BLOCK\_DB des Datenbausteins, welcher die Empfangsdaten des CP beinhaltet, bei Einsatz in CPUs von Yaskawa bzw. in S7-300 CPUs von Siemens.
- Zeiger vom *Typ* DB\_ANY auf den Datenbaustein, welcher die Empfangsdaten des CP beinhaltet, bei Einsatz in S7-1500 CPUs von Siemens.

**ABD** Wortvariable, welche die Nummer des Datenworts enthält, ab dem die auszugebenden Zeichen abgelegt sind.

**ANZ** Wort-Variable, welche die Anzahl der Bytes enthält, die empfangen wurden.

<b>PAFE</b>	<p>Alle Bits dieses Merker-Bytes sind bei richtiger Funktion "0". Bei Fehlfunktion wird ein Fehlercode eingetragen. Die Fehlerangabe ist selbstquittierend, d.h. nach Beseitigung der Fehlerursache wird das Byte wieder auf "0" gesetzt. Folgende Fehler sind möglich:</p> <ul style="list-style-type: none"> <li>■ 1 = Datenbaustein nicht vorhanden</li> <li>■ 2 = Datenbaustein zu kurz</li> <li>■ 3 = Datenbausteinnummer nicht im gültigen Bereich</li> </ul>
<b>EMFR</b>	<p>Durch Setzen des <i>EMFR</i> zeigt der Hantierungsbaustein an, dass Daten empfangen wurden. Erst durch Rücksetzen von <i>EMFR</i> im Anwenderprogramm können weitere Daten empfangen werden.</p>
<b>GEEM, ANZ_INT, LETZTER_BLOCK, EMPF_LAEUFT, FEHLER_EMPF, OFFSET</b>	<p>Diese Parameter werden intern verwendet. Sie dienen dem Informationsaustausch zwischen den Hantierungsbausteinen. Für den Einsatz des SYNCHRON_RESET (FC9) sind die Steuerbits EMFR, LETZTER_BLOCK, EMPF_LAEUFT und FEHLER_EMPF immer in einem Merker-Byte abzulegen.</p>

### 5.3.3 FC 8 - STEUERBIT - Modemfunktionalität CP 240

**Beschreibung** Mit diesem Baustein haben Sie folgenden Zugriff auf die seriellen Modemleitungen:

Lesen:	DTR, RTS, DSR, RI, CTS, CD
Schreiben:	DTR, RTS

#### Parameter

Name	Deklaration	Typ	Beschreibung
ADR	IN	INT	Peripherieadresse
RTS	IN	BOOL	Neuer Zustand RTS
DTR	IN	BOOL	Neuer Zustand DTR
MASKE_RTS	IN	BOOL	<ul style="list-style-type: none"> <li>■ 0: nichts ändern</li> <li>■ 1: Zustand an RTS übergeben</li> </ul>
MASKE_DTR	IN	BOOL	<ul style="list-style-type: none"> <li>■ 0: nichts ändern</li> <li>■ 1: Zustand an DTR übergeben</li> </ul>
RET_VAL	OUT	WORD	Rückgabewert (0 = OK)
STATUS	OUT / IN_OUT	BYTE	<b>STATUS</b> <ul style="list-style-type: none"> <li>■ Status vom <i>Typ</i> BYTE als OUT-Parameter bei Einsatz in CPUs von Yaskawa bzw. in S7-300 CPUs von Siemens.</li> <li>■ Status vom <i>Typ</i> BYTE als IN_OUT-Parameter bei Einsatz in S7-1500 CPUs von Siemens.</li> </ul>
DELTA_STATUS	OUT / IN_OUT	BYTE	<b>DELTA_STATUS</b> <ul style="list-style-type: none"> <li>■ Statusänderungen seit dem letzten Zugriff vom <i>Typ</i> BYTE als OUT-Parameter bei Einsatz in CPUs von Yaskawa bzw. in S7-300 CPUs von Siemens.</li> <li>■ Statusänderungen seit dem letzten Zugriff vom <i>Typ</i> BYTE als IN_OUT-Parameter bei Einsatz in S7-1500 CPUs von Siemens.</li> </ul>
START	IN_OUT	BOOL	Auftrag anstoßen

Name	Deklaration	Typ	Beschreibung
AUFTRAG_LAEUFT	IN_OUT	BOOL	Status der Funktion



*Dieser Baustein darf nicht aufgerufen werden, solange ein Sendeauftrag läuft, ansonsten kann dies zu Datenverlust führen.*

**ADR** Peripherieadresse unter welcher der CP 240 anzusprechen ist. Über die Hardware-Konfiguration bestimmen Sie die Peripherieadresse.

**RTS, DTR** Mit diesem Parameter geben Sie den Status für *RTS* bzw. *DTR* vor, den Sie über *MASK\_RTS* bzw. *MASK\_DTR* aktivieren können.

**MASK\_RTS, MASK\_DTR** Hier wird mit 1 der Status des entsprechenden Parameters übernommen, sobald Sie *START* auf 1 setzen.

**RET\_VAL** Dieser Parameter liefert zur Zeit immer 00h zurück und dient zukünftigen Fehlermeldungen.

**STATUS, DELTA\_STATUS** *STATUS* liefert den aktuellen Status der Modem-Leitungen zurück. *DELTA\_STATUS* liefert den Status der Modem-Leitungen zurück, die sich seit dem letzten Zugriff geändert haben. Die Bytes haben folgenden Aufbau:

Bit-Nr.	7	6	5	4	3	2	1	0
STATUS	x	x	RTS	DTR	CD	RI	DSR	CTS
DELTA_STATUS	x	x	x	x	CD	RI	DSR	CTS

**START** Durch Setzen von *START* wird der über die Maske aktivierte Status übernommen.

**AUFTRAG\_LAEUFT** Solange die Funktion abgearbeitet wird, bleibt dieses Bit gesetzt.

### 5.3.4 FC 9 - SYNCHRON\_RESET - Synchronisation CPU und CP 240

**Beschreibung** Der Baustein ist im zyklischen Programmteil aufzurufen. Mit dieser Funktion wird die Anlaufkennung des CP 240 quittiert, und so die Synchronisation zwischen CPU und CP hergestellt. Weiterhin kann bei einer Kommunikationsunterbrechung der CP rückgesetzt werden und so ein synchroner Anlauf erfolgen.



*Eine Kommunikation mit SEND- und RECEIVE-Bausteinen ist nur möglich, wenn zuvor im Anlauf-OB der Parameter ANL des SYNCHRON-Bausteins gesetzt wurde.*

#### Parameter

Name	Deklaration	Typ	Beschreibung
ADR	IN	INT	Peripherieadresse
TIMER_NR	IN	TIMER	Timer
ANL	IN_OUT	BOOL	CPU-Neustart erfolgt
ZERO	IN_OUT	BOOL	wird intern verwendet
RESET	IN_OUT	BOOL	Reset an CP
TIME_AN	IN_OUT	BOOL	wird intern verwendet
STEUERB_S	IN_OUT	BYTE	wird intern verwendet
STEUERB_R	IN_OUT	BYTE	wird intern verwendet

<b>ADR</b>	Peripherieadresse unter der der CP 240 anzusprechen ist. Über die Hardware-Konfiguration bestimmen Sie die Peripherieadresse.
<b>TIMER_NR</b>	Timer für die Wartezeit.
<b>ANL</b>	Mit <i>ANL</i> = 1 wird dem Hantierungsbaustein mitgeteilt, dass an der CPU STOP/START bzw. NETZ-AUS/NETZ-EIN erfolgt ist und nun eine Synchronisation erfolgen muss. Nach der Synchronisation wird <i>ANL</i> automatisch zurückgesetzt.
<b>ZERO</b>	Parameter wird intern verwendet.
<b>RESET</b>	Mit <i>RESET</i> = 1 können Sie den CP aus Ihrem Anwenderprogramm zurücksetzen.
<b>TIME_AN</b>	Parameter wird intern verwendet.
<b>STEUERB_S</b>	Hier ist das Merkerbyte anzugeben, in dem die Steuerbits FRG, ENDE_KOM, LETZTER_BLOCK, SENDEN_LAEUFT und FEHLER_KOM für den SEND-FC abgelegt sind.
<b>STEUERB_R</b>	Hier ist das Merkerbyte anzugeben, in dem die Steuerbits EMFR, LETZTER_BLOCK, EMPF_LAEUFT und FEHLER_EMPF für den RECEIVE-FC abgelegt sind.

### 5.3.5 FC 11 - ASCII\_FRAGMENT - Fragment Datenempfang CP 240

#### Beschreibung



*Bitte beachten Sie, dass dieser Baustein ausschließlich für den Einsatz in CPUs von Yaskawa bzw. in S7-300 CPUs von Siemens geeignet ist!*

Dieser FC dient zum fragmentierten ASCII-Datenempfang. Hiermit haben Sie die Möglichkeit große Telegramme in 12Byte-Blöcken direkt nach dem Erhalt an die CPU weiterzureichen. Hierbei wartet der CP nicht, bis das komplette Telegramm empfangen wurde. Der Einsatz des FC 11 setzt voraus, dass Sie beim Empfänger "ASCII-fragmentiert" parametrieren. Im FC 11 legen Sie über die Bezeichner **\_DB** und **ABD** den Empfangsbereich fest. Ist der Ausgang **EMFR** gesetzt, so ist ein neues Telegramm komplett eingelesen worden. Die Länge des eingelesenen Telegramms wird in **ANZ** abgelegt. Nach der Auswertung des Telegramms ist dieses Bit vom Anwender zurückzusetzen, da ansonsten kein weiteres Telegramm in der CPU übernommen werden kann.

#### Parameter

Name	Deklaration	Typ	Beschreibung
ADR	IN	INT	Peripherieadresse
_DB	IN	BLOCK_DB	DB-Nummer mit den Empfangsdaten
ABD	IN	WORD	Nummer des 1. Datenworts
ANZ	OUT	WORD	Anzahl der empfangenen Bytes
EMFR	IN_OUT	BOOL	Empfangsbestätigung
GEEM	IN_OUT	WORD	wird intern verwendet
ANZ_INT	IN_OUT	WORD	wird intern verwendet
EMPF_LAEUFT	IN_OUT	BOOL	wird intern verwendet
LETZTER_BLOCK	IN_OUT	BOOL	wird intern verwendet
FEHLER_EMFP	IN_OUT	BOOL	wird intern verwendet
PAFE	OUT	BYTE	Parametrierfehler (0 = OK)

<b>ADR</b>	Peripherieadresse unter welcher der CP 240 anzusprechen ist. Über die Hardware-Konfiguration bestimmen Sie die Peripherieadresse.
<b>_DB</b>	Nummer des Datenbausteins, der die empfangenen Daten beinhaltet.
<b>ABD</b>	Wortvariable, welche die Nummer des Datenworts enthält, ab dem die empfangenen Zeichen abgelegt sind.
<b>ANZ</b>	Wort-Variable, die die Anzahl der Bytes enthält, die empfangen wurden.
<b>EMFR</b>	Durch Setzen des <b>EMFR</b> zeigt der Hantierungsbaustein an, dass Daten empfangen wurden. Erst durch Rücksetzen von <b>EMFR</b> im Anwenderprogramm können weitere Daten empfangen werden.

<b>PAFE</b>	<p>Alle Bits dieses Merker-Bytes sind bei richtiger Funktion "0". Bei Fehlfunktion wird ein Fehlercode eingetragen. Die Fehlerangabe ist selbstquittierend, d.h. nach Beseitigung der Fehlerursache wird das Byte wieder "0" gesetzt. Folgende Fehler sind möglich:</p> <ul style="list-style-type: none"><li>■ 1 = Datenbaustein nicht vorhanden</li><li>■ 2 = Datenbaustein zu kurz</li><li>■ 3 = Datenbausteinnummer nicht im gültigen Bereich</li></ul>
<b>GEEM, ANZ_INT, LETZTER_BLOCK, EMPF_LAEUFT, FEHLER_EMPF</b>	<p>Diese Parameter werden intern verwendet. Sie dienen dem Informationsaustausch zwischen den Hantierungsbausteinen. Für den Einsatz des SYNCHRON_RESET sind die Steuerbits LETZTER_BLOCK, EMPF_LAEUFT und FEHLER_EMPF immer in einem Merker-Byte abzulegen.</p>