

VIPA SPEED7 Library

OPL_SP7-LIB | SW90GS0MA V10.002 | Handbuch

HB00 | OPL_SP7-LIB | SW90GS0MA V10.002 | de | 18-04

Baustein Bibliothek - Serial Communication



VIPA GmbH
Ohmstr. 4
91074 Herzogenaurach
Telefon: +49 9132 744-0
Telefax: +49 9132 744-1864
E-Mail: info@vipa.com
Internet: www.vipa.com

Inhaltsverzeichnis

1	Allgemeines	4
1.1	Copyright © VIPA GmbH	4
1.2	Über dieses Handbuch	5
2	Wichtige Hinweise	6
2.1	Intern verwendete Bausteine	6
3	Bibliothek einbinden	7
3.1	Einbinden in Siemens SIMATIC Manager	7
3.2	Einbinden in Siemens TIA Portal	8
4	Serielle Kommunikation - "Serial Communication"	9
4.1	Serielle Kommunikation - "Serial Communication"	9
4.1.1	SFC 207 - SER_CTRL - Modemfunktionalität PtP	9
4.1.2	FC/SFC 216 - SER_CFG - Parametrierung PtP	10
4.1.3	FC/SFC 217 - SER_SND - Senden an PtP	14
4.1.4	FC/SFC 218 - SER_RCV - Empfangen von PtP	19
4.1.5	FB 1 - RECEIVE_ASCII - Empfangen mit definierter Länge von PtP	21
4.1.6	FB 7 - P_RCV_RK - Empfangen von CP 341	22
4.1.7	FB 8 - P_SND_RK - Senden an CP 341	23
4.2	CP040	25
4.2.1	FB 60 - SEND - Senden an System SLIO CP 040	25
4.2.2	FB 61 - RECEIVE - Empfangen von System SLIO CP 040	27
4.2.3	FB 65 - CP040_COM - Kommunikation SLIO CP 040	30
4.3	CP240	34
4.3.1	FC 0 - SEND - Senden an CP 240	34
4.3.2	FC 1 - RECEIVE - Empfangen von CP 240	35
4.3.3	FC 8 - STEUERBIT - Modemfunktionalität CP 240	36
4.3.4	FC 9 - SYNCHRON_RESET - Synchronisation CPU und CP 240	37
4.3.5	FC 11 - ASCII_FRAGMENT - Fragment Datenempfang CP 240	38

1 Allgemeines

1.1 Copyright © VIPA GmbH

All Rights Reserved

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

Dieses Material ist durch Urheberrechtsgesetze geschützt. Ohne schriftliches Einverständnis von VIPA und dem Besitzer dieses Materials darf dieses Material weder reproduziert, verteilt, noch in keiner Form von keiner Einheit (sowohl VIPA-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: VIPA, Gesellschaft für Visualisierung und Prozessautomatisierung mbH Ohmstraße 4, D-91074 Herzogenaurach, Germany

Tel.: +49 9132 744 -0

Fax.: +49 9132 744-1864

E-Mail: info@vipa.de

<http://www.vipa.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 VIPA 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 VIPA GmbH.

Warenzeichen

VIPA, SLIO, System 100V, System 200V, System 300V, System 300S, System 400V, System 500S und Commander Compact sind eingetragene Warenzeichen der VIPA Gesellschaft für Visualisierung und Prozessautomatisierung mbH.

SPEED7 ist ein eingetragenes Warenzeichen der profichip GmbH.

SIMATIC, STEP, SINEC, TIA Portal, S7-300 und S7-400 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 VIPA GmbH, wenn Sie Fehler anzeigen oder inhaltliche Fragen zu diesem Dokument stellen möchten. Ist eine solche Stelle nicht erreichbar, können Sie VIPA über folgenden Kontakt erreichen:

VIPA GmbH, Ohmstraße 4, 91074 Herzogenaurach, Germany

Telefax: +49 9132 744-1204

E-Mail: documentation@vipa.de

Technischer Support

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

VIPA GmbH, Ohmstraße 4, 91074 Herzogenaurach, Germany

Telefon: +49 9132 744-1150 (Hotline)

E-Mail: support@vipa.de

1.2 Über dieses Handbuch

Zielsetzung und Inhalt

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

- Beschrieben wird Aufbau, Projektierung und Anwendung in verschiedenen Programmiersystemen.
- Das Handbuch ist geschrieben für Anwender mit Grundkenntnissen in der Automatisierungstechnik.
- Das Handbuch ist in elektronischer Form als PDF-Datei verfügbar. Hierzu ist der Adobe Acrobat Reader erforderlich.
- 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

Besonders wichtige Textteile sind mit folgenden Piktogrammen und Signalworten ausgezeichnet:

**GEFAHR!**

Unmittelbar drohende oder mögliche Gefahr. Personenschäden sind möglich.

**VORSICHT!**

Bei Nichtbefolgen sind Sachschäden möglich.



Zusätzliche Informationen und nützliche Tipps.

2 Wichtige Hinweise



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

2.1 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 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 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

3 Bibliothek einbinden

Baustein-Bibliothek "Serial Communication"

Die Baustein-Bibliothek finden Sie im "Service/Support"-Bereich auf www.vipa.com unter "Downloads → VIPA Lib" 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.

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 über- tragen

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 VIPA-spezifischen Bausteine.



Werden anstelle der SFCs FCs verwendet, so werden diese von den VIPA CPUs 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 öffnen und Bausteine in Projekt übertragen

ZIP-Datei laden

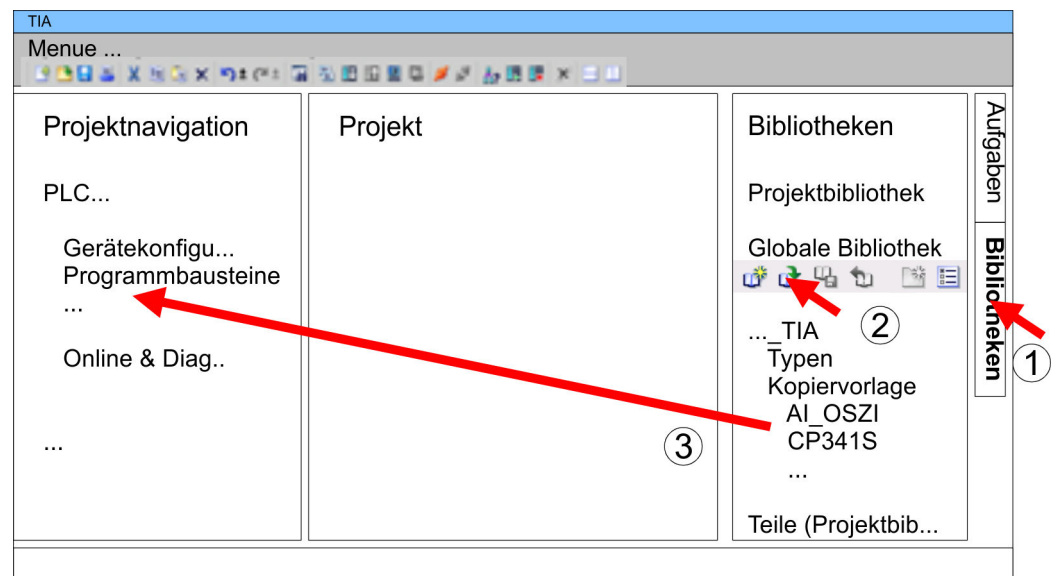
1. Navigieren Sie auf der Webseite zu der ZIP-Datei, welche zu Ihrer Programmversion passt.
2. Laden und speichern Sie diese in Ihrem Arbeitsverzeichnis.

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 VIPA-spezifischen Bausteine.

4 Serielle Kommunikation - "Serial Communication"

4.1 Serielle Kommunikation - "Serial Communication"

4.1.1 SFC 207 - SER_CTRL - Modemfunktionalität PtP

Beschreibung



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

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>0:</i>	DTR, RTS, DSR, RI, CTS, CD	nicht möglich

Parameter

Name	Deklaration	Typ	Beschreibung
WRITE	IN	BYTE	<ul style="list-style-type: none"> Bit 0: Neuer Zustand DTR Bit 1: Neuer Zustand RTS
MASKWRITE	IN	BYTE	<ul style="list-style-type: none"> Bit 0: Zustand an DTR übergeben Bit 1: Zustand an RTS übergeben
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"> 1: Fehler in <i>WRITE</i> 2: Fehler in <i>MASKWRITE</i> 3: Fehler in <i>READ</i> 4: Fehler in <i>READDELTA</i>
809Ah	Schnittstelle ist nicht vorhanden
809Bh	Schnittstelle ist nicht konfiguriert (SFC 216)

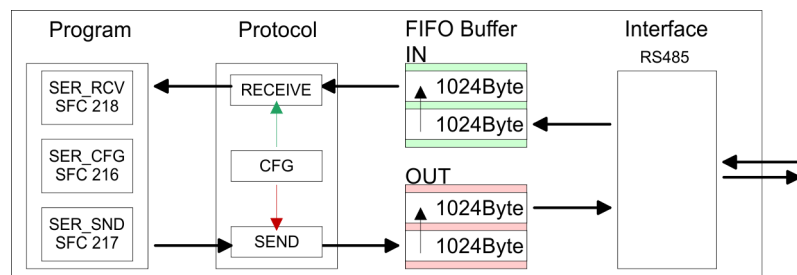
4.1.2 FC/SFC 216 - SER_CFG - Parametrierung PtP

Beschreibung

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*

3: 2Bit

*) 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 2.8.1 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 ...

4.1.3 FC/SFC 217 - SER_SND - Senden an PtP

Beschreibung

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:

Fehler-code	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)
8x24h	Fehler in FC/SFC-Parameter x, mit x: 1: Fehler in <i>DATAPTR</i> 2: Fehler in <i>DATALEN</i>

Fehler-code	Beschreibung
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

Fehler-code	Beschreibung
2000h	Senden fertig ohne Fehler
8080h	Empfangspuffer voll (kein Platz für Quittung)
8090h	Quittungsverzugszeit überschritten
80F0h	Falsche Checksumme in Rückantwort
80FEh	Falsches Startzeichen in der Rückantwort

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

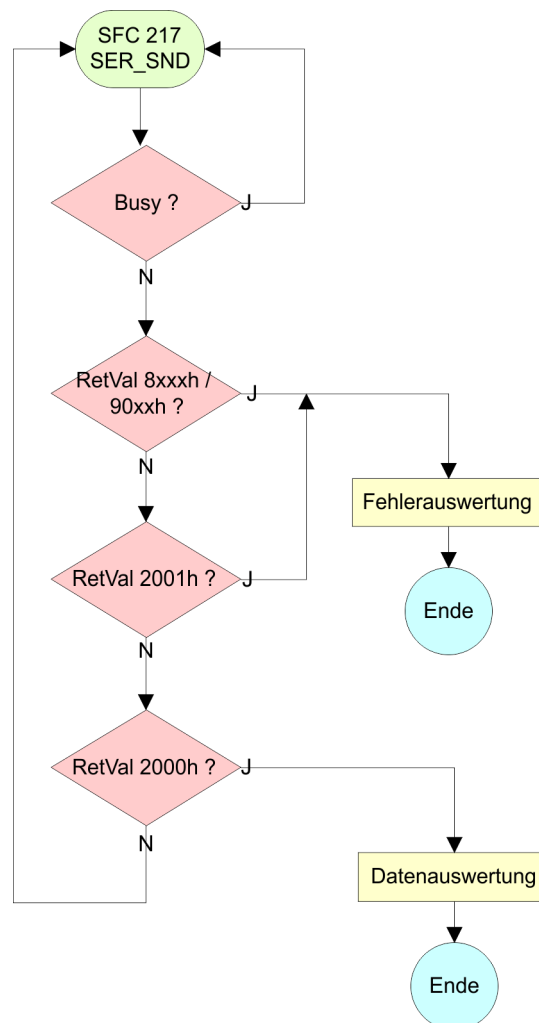
Modbus RTU/ASCII Master

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)

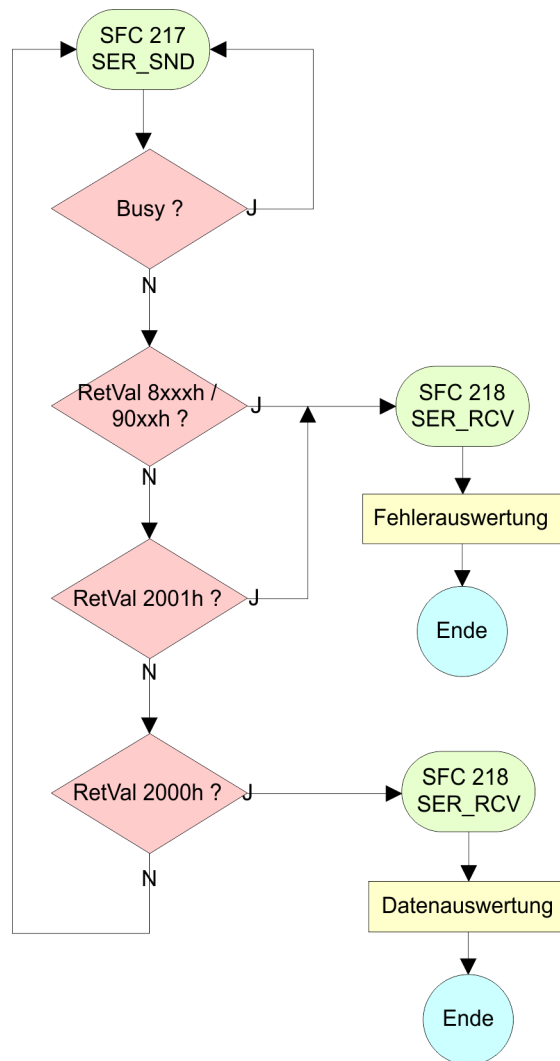
Prinzip der Programmierung

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

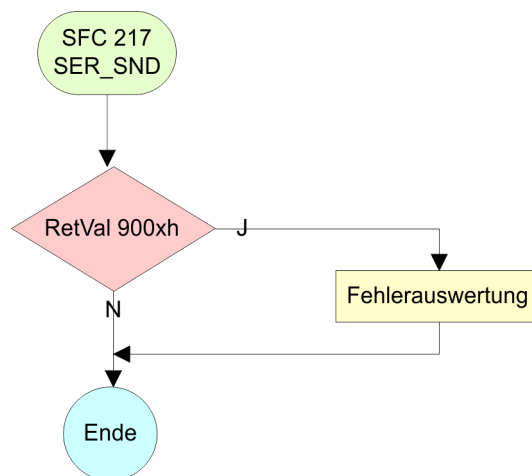
3964R



USS / Modbus



ASCII / STX/ETX



4.1.4 FC/SFC 218 - SER_RCV - Empfangen von PtP

Beschreibung

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

3964R / Modbus RTU/ASCII Master

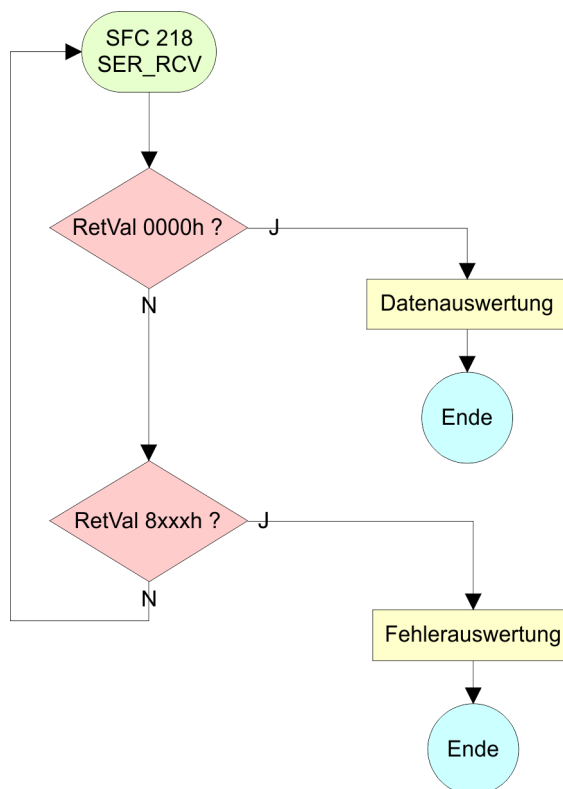
Bit	Fehler	Beschreibung
0	overflow	Das empfangene Telegramm übersteigt die Größe des Empfangspuffers.

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.

**4.1.5 FB 1 - RECEIVE_ASCII - Empfangen mit definierter Länge von PtP****Beschreibung**

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.

4.1.6 FB 7 - P_RCV_RK - Empfangen von CP 341

Beschreibung

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.
NDR*	OUT	BOOL	Auftrag fertig ohne Fehler, Daten übernommen Parameter <i>STATUS</i> = 00h
ERROR*	OUT	BOOL	Auftrag fertig mit Fehler Parameter <i>STATUS</i> enthält die Fehlerinformation
LEN*	OUT	BOOL	Länge des empfangenen Telegramms in Byte $1 \leq LEN \leq 1024$
STATUS*	OUT	WORD	Spezifikation des Fehlers bei <i>ERROR</i> = 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 Anlaufsynchronisation 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 *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.

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.

4.1.7 FB 8 - P_SND_RK - Senden an CP 341**Beschreibung**

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.
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*	OUT	BOOL	Auftrag fertig ohne Fehler, Daten gesendet Parameter <i>STATUS</i> = 00h
ERROR*	OUT	BOOL	Auftrag fertig mit Fehler Parameter <i>STATUS</i> enthält die Fehlerinformation
STATUS*	OUT	WORD	Spezifikation des Fehlers bei <i>ERROR</i> = 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 Anlaufsynchroisation 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.

4.2 CP040**4.2.1 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	IN	INT	Logische Basisadresse des CP.
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*	OUT	BOOL	Sende-Auftrag fertig ohne Fehler.
ERROR*	OUT	BOOL	Sende-Auftrag fertig mit Fehler. Parameter <i>STATUS</i> enthält die Fehlerinformation.
STATUS*	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).

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

REQ	<p>Request - Sendefreigabe:</p> <ul style="list-style-type: none"> ■ Mit einer positiven Flanke am Eingang <i>REQ</i> wird die Übertragung der Daten angestoßen. ■ Je nach Datenmenge kann eine Datenübertragung über mehrere Programmzyklen laufen.
R	<p>Synchron Reset:</p> <ul style="list-style-type: none"> ■ Für die Initialisierung ist SEND im Anlauf-OB mit allen Parametern und mit gesetztem <i>R</i> einmalig aufzurufen. ■ 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. ■ Steht statisch der Signalzustand "1" am Eingang <i>R</i> an, so ist das Senden ausgeschaltet.
LADDR	<p>Peripherieadresse:</p> <ul style="list-style-type: none"> ■ 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.
DB_NO	<p>Datenbaustein-Nummer:</p> <ul style="list-style-type: none"> ■ Nummer des Datenbausteins, der die zu sendenden Daten beinhaltet. ■ Null ist nicht erlaubt.
DBB_NO	<p>Datenbyte-Nummer:</p> <ul style="list-style-type: none"> ■ Nummer des Datenbytes im Datenbaustein, ab dem die Sendedaten abgelegt sind.
LEN	<p>Länge:</p> <ul style="list-style-type: none"> ■ Länge der Nutzdaten, welche zu übertragen sind. ■ Es gilt: $1 \leq LEN \leq 1024$.
IO_SIZE	<p>Größe E/A-Bereich:</p> <ul style="list-style-type: none"> ■ 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"> – PROFIBUS: 8Byte, 20Byte oder 60Byte wählbar – PROFINET: 20Byte oder 60Byte wählbar – CANopen: 8Byte – EtherCAT: 60Byte – DeviceNET: 60Byte – ModbusTCP: 60Byte
DONE	<p><i>DONE</i>:</p> <ul style="list-style-type: none"> ■ wird gesetzt bei Auftrag fertig ohne Fehler und <i>STATUS</i> = 0000h.

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	Hantierungsbaustein und CP sind nicht synchron (Abhilfe: Synchron Reset auslösen)
0301h	DB ist nicht gültig
070Ah	Übertragung fehlgeschlagen, Partner antwortet nicht, oder hat den Auftrag negativ quittiert
0816h	Ungültiger LEN-Parameter (LEN = 0 oder LEN > 1024)
8181h	Auftrag läuft (Status und keine Fehlermeldung)

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 Binäresultat 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.

4.2.2 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	IN	INT	Logische Basisadresse des CP.
DB_NO	IN	INT	Datenbausteinnummer der Empfangsdaten.

Parameter	Deklaration	Datentyp	Beschreibung
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*	OUT	BOOL	Empfangs-Auftrag fertig ohne Fehler.
ERROR*	OUT	BOOL	Empfangs-Auftrag fertig mit Fehler. Parameter <i>STATUS</i> enthält die Fehlerinformation.
STATUS*	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).

*) 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:
- 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.

- 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	Hantierungsbaustein und CP sind nicht synchron (Abhilfe: Synchron Reset auslösen)
0301h	Ungültiger DB ausgewählt
070Ah	Übertragung fehlgeschlagen, Partner antwortet nicht, oder hat den Auftrag negativ quittiert
0816h	Ungültiger <i>LEN</i> -Parameter (<i>LEN</i> = 0 oder <i>LEN</i> > 1024)
080Ah	Ein freier Empfangspuffer ist nicht vorhanden
080Ch	Fehlerhaftes Zeichen empfangen (Zeichenrahmen- oder Paritätsfehler)
8181h	Auftrag läuft (Status und keine Fehlermeldung)

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.

4.2.3 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

Name	Deklara- tion	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_IN	IN	INT	Eingangsadresse des CP aus der HW-Konfiguration.
ADDR_OUT	IN	INT	Ausgangsadresse des CP aus der HW-Konfiguration.
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.
IO_SIZE	IN	WORD	Parametrierte IO Größe des Moduls.
DONE_SEND*	OUT	BOOL	Sende-Auftrag fertig ohne Fehler. Daten gesendet: Parameter <i>STATUS_SEND</i> = 0000h.
ERROR_SEND*	OUT	BOOL	Sende-Auftrag fertig mit Fehler. Parameter <i>STATUS_SEND</i> enthält die Fehlerinformation.
STATUS_SEND*	OUT	WORD	Spezifikation des Fehlers beim Senden bei <i>ERROR_SEND</i> = 1
LEN_RCV	OUT	INT	Länge des empfangenen Telegramms in Byte. $1 \leq LEN_RCV \leq 1024$

Name	Deklara- tion	Typ	Beschreibung
NDR_RCV*	OUT	BOOL	Empfangs-Auftrag fertig ohne Fehler. Daten gesendet: Parameter <i>STATUS_RCV</i> = 0000h. Bleibt für einen Zyklus stehen.
ERROR_RCV*	OUT	BOOL	Empfangs-Auftrag fertig mit Fehler. Parameter <i>STATUS_RCV</i> enthält die Fehlerinformation.
STATUS_RCV*	OUT	WORD	Spezifikation des Fehlers beim Empfangen bei <i>ERROR_RCV</i> = 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:

- 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:

- 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.

DB_NO_SEND

Datenbaustein-Nummer SEND:

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

DBB_NO_SEND

Datenbyte-Nummer SEND:

- Nummer des Datenbytes im Datenbaustein, ab dem die Sendedaten abgelegt sind.

LEN_SEND

Länge SEND:

- Länge der Nutzdaten, welche zu übertragen sind.
- Es gilt: $1 \leq \text{LEN_SEND} \leq 1024$.

DB_NO_RECV

Datenbaustein-Nummer RECV:

- Nummer des Empfang-DBs.
- Null ist nicht erlaubt.

DBB_NO_RECV

Datenbyte-Nummer RECV:

- Nummer des Datenbytes im Datenbaustein, ab dem die Empfangsdaten abgelegt sind.

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:
 - SLIO CPU: 8Byte, 20Byte oder 60Byte wählbar
 - PROFIBUS: 8Byte, 20Byte oder 60Byte wählbar
 - PROFINET: 20Byte oder 60Byte wählbar
 - CANopen: 8Byte
 - EtherCAT: 60Byte
 - DeviceNET: 60Byte
 - ModbusTCP: 60Byte

DONE_SEND

Sende-Auftrag fertig ohne Fehler, Daten gesendet. Parameter *STATUS_SEND* = 0000h.

ERROR_SEND

ERROR_SEND wird gesetzt bei Auftrag fertig mit Fehler. Hierbei enthält *STATUS_SEND* die entsprechende Fehlerinformation.

STATUS_SEND

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

STATUS	Beschreibung
0000h	Kein Fehler vorhanden
0202h	<i>IO_SIZE</i> = 0 oder <i>IO_SIZE</i> > 60
0301h	DB ist nicht gültig
070Ah	Übertragung fehlgeschlagen, Partner antwortet nicht, oder hat den Auftrag negativ quittiert
0517h	Ungültiger Parameter <i>LEN_SEND</i> (<i>LEN_SEND</i> = 0 oder <i>LEN_SEND</i> > 1024)
8181h	Auftrag läuft (Status und keine Fehlermeldung)

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> = 0 oder <i>IO_SIZE</i> > 60
0301h	Ungültiger DB ausgewählt
070Ah	Übertragung fehlgeschlagen, Partner antwortet nicht, oder hat den Auftrag negativ quittiert
0816h	Ungültiger Parameter <i>LEN_RCV</i> (<i>LEN_RCV</i> = 0 oder <i>LEN_RCV</i> > 1024)
080Ah	Ein freier Empfangspuffer ist nicht vorhanden
080Ch	Fehlerhaftes Zeichen empfangen (Zeichenrahmen- oder Paritätsfehler)
8181h	Auftrag läuft (Status und keine Fehlermeldung)

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 aufruf verfügbar sind. Zur weiteren Auswertung sollten Sie diese in einen freien Datenbereich kopieren.

4.3 CP240

4.3.1 FC 0 - SEND - 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-Nummer mit den Sendedaten
ABD	IN	WORD	Nummer des 1. Datenworts
ANZ	IN	WORD	Anzahl der Bytes
FRG	IN_OUT	BOOL	Auftrag anstoßen
GESE	IN_OUT	WORD	wird intern verwendet
ANZ_INT	IN_OUT	WORD	wird intern verwendet
ENDE_KOMM	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
PAFE	OUT	BYTE	Parametrierfehler Code (0 = OK)

ADR	Peripherieadresse unter welcher der CP 240 anzusprechen ist. Über die Hardware-Konfiguration bestimmen Sie die Peripherieadresse.
_DB	Nummer des Datenbausteins, der die zu sendenden Daten beinhaltet.
ABD	Wortvariable, welche die Nummer des Datenworts enthält, ab dem die auszugebenden Zeichen abgelegt sind.
ANZ	Anzahl der Bytes, die zu übertragen sind.
FRG Sendefreigabe	Bei <i>FRG</i> = "1" werden die über <i>_DB</i> , <i>ADB</i> und <i>ANZ</i> definieren Daten einmalig an den über <i>ADR</i> adressierten CP übertragen. Nach der Übertragung wird <i>FRG</i> wieder zurückgesetzt. Ist beim Aufruf <i>FRG</i> = "0", wird der Baustein sofort wieder verlassen!
PAFE	<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"> ■ 1 = Datenbaustein nicht vorhanden ■ 2 = Datenbaustein zu kurz ■ 3 = Datenbausteinnummer nicht im gültigen Bereich

**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 ENDE_KOM, LETZTER_BLOCK, SENDEN_LAEUFT und FEHLER_KOM immer in einem Merker-Byte abzulegen.

4.3.2 FC 1 - RECEIVE - 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-Nummer mit den Empfangsdaten
ABD	IN	WORD	Nummer des 1. Datenworts
ANZ	OUT	WORD	Anzahl der empfangenen Bytes
EMFR	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
PAFE	OUT	BYTE	Parametrierfehler Code (0 = OK)
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 des Datenbausteins, der die zu sendenden Daten beinhaltet.

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.

EMFR Durch Setzen des *EMFR* zeigt der Hantierungsbaustein an, dass Daten empfangen wurden. Erst durch Rücksetzen von *EMFR* im Anwenderprogramm können weitere Daten empfangen werden.

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

**GEEM, ANZ_INT
LETZTER_BLOCK
EMPF_LAEUFT
FEHLER_EMPF OFFSET**

Diese Parameter werden intern verwendet. Sie dienen dem Informationsaustausch zwischen den Hantierungsbausteinen. Für den Einsatz des SYNCHRON_RESET (FC9) sind die Steuerbits LETZTER_BLOCK, EMPF_LAEUFT und FEHLER_EMPF immer in einem Merker-Byte abzulegen.

4.3.3 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"> ■ 0: nichts ändern ■ 1: Zustand an RTS übergeben
MASKE_DTR	IN	BOOL	<ul style="list-style-type: none"> ■ 0: nichts ändern ■ 1: Zustand an DTR übergeben
STATUS	OUT	BYTE	Status
DELTA_STATUS	OUT	BYTE	Statusänderungen seit dem letzten Zugriff
START	IN_OUT	BOOL	Auftrag anstoßen
AUFTRAG_LAEU	IN_OUT	BOOL	Status der Funktion
RET_VAL	OUT	WORD	Rückgabewert (0 = OK)



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.
- 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_LAEU** Solange die Funktion abgearbeitet wird, bleibt dieses Bit gesetzt.
- RET_VAL** Dieser Parameter liefert zur Zeit immer 00h zurück und dient zukünftigen Fehlermeldungen.

4.3.4 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	WORD	Timer-Nummer
ANL	IN_OUT	BOOL	CPU-Neustart erfolgt
NULL	IN_OUT	BOOL	wird intern verwendet
RESET	IN_OUT	BOOL	Reset an CP
STEUERB_S	IN_OUT	BYTE	wird intern verwendet
STEUERB_R	IN_OUT	BYTE	wird intern verwendet

CP240 > FC 11 - ASCII_FRAGMENT - Fragment Datenempfang CP 240

ADR	Peripherieadresse unter der der CP 240 anzusprechen ist. Über die Hardware-Konfiguration bestimmen Sie die Peripherieadresse.
TIMER_NR	Nummer des Timers für die Wartezeit.
ANL	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.
NULL	Parameter wird intern verwendet.
RESET	Mit <i>RESET</i> = 1 können Sie den CP aus Ihrem Anwenderprogramm zurücksetzen.
STEUERB_S	Hier ist das Merkerbyte anzugeben, in dem die Steuerbits ENDE_KOM, LETZTER_BLOCK, SENDEN_LAEUFT und FEHLER_KOM für den SEND-FC abgelegt sind.
STEUERB_R	Hier ist das Merkerbyte anzugeben, in dem die Steuerbits LETZTER_BLOCK, EMPF_LAEUFT und FEHLER_EMPF für den RECEIVE-FC abgelegt sind.

4.3.5 FC 11 - ASCII_FRAGMENT - Fragment Datenempfang CP 240

Beschreibung	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" parametrisiert haben. Im FC 11 legen Sie über die Bezeichner <i>_DB</i> und <i>ABD</i> den Empfangsbereich fest. Ist der Ausgang <i>EMFR</i> gesetzt, so ist ein neues Telegramm komplett eingelesen worden. Die Länge des eingelesenen Telegramms wird in <i>ANZ</i> 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

Name	Deklaration	Typ	Beschreibung
FEHLER_EMPF	IN_OUT	BOOL	wird intern verwendet
PAFE	OUT	BYTE	Parametrierfehler (0 = OK)

ADR	Peripherieadresse unter der der CP 240 anzusprechen ist. Über die Hardware-Konfiguration bestimmen Sie die Peripherieadresse.
_DB	Nummer des Datenbausteins, der die empfangenen Daten beinhaltet.
ABD	Wortvariable, die die Nummer des Datenworts enthält, ab dem die empfangenen Zeichen abgelegt sind.
ANZ	Wort-Variable, die die Anzahl der Bytes enthält, die empfangen wurden.
EMFR	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.
PAFE	<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"> ■ 1 = Datenbaustein nicht vorhanden ■ 2 = Datenbaustein zu kurz ■ 3 = Datenbausteinnummer nicht im gültigen Bereich
GEEM, ANZ_INT LETZTER_BLOCK EMPF_LAEUFT FEHLER_EMPF	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.