YASKAWA

VIPA SPEED7 Library

OPL_SP7-LIB | SW90KS0MA V10.002 | Handbuch

HB00 | OPL_SP7-LIB | SW90KS0MA V10.002 | de | 18-28

Baustein Bibliothek - System Blocks

www.vipa.com/de/service-support/handbuch

VIPA CONTROLS

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

1.1 Copyright © VIPA GmbH

All Rights Reserved

Tel.: +49 9132 744 -0

Fax.: +49 9132 744-1864

EMail: 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

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

EMail: 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)

EMail: support@vipa.de

1.2 Über dieses Handbuch

Zielsetzung und Inhalt

Das Handbuch beschreibt die Baustein-Bibliothek *"System Blocks"* 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

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 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 *"System Blocks"*

Die Baustein-Bibliothek finden Sie im *"Service/Support"*-Bereich auf www.vipa.com unter *"Downloads ➔ VIPA Lib"* als *"Baustein-Bibliothek System Blocks - SW90KS0MA"* zum Download. Die Bibliothek liegt als gepackte zip-Dateien vor. Sobald Sie die Bausteine verwenden möchten, müssen Sie diese in Ihr Projekt importieren.

Folgende Bausteinbibliotheken stehen zur Verfügung

Datei	Beschreibung
SystemBlocks_S7_V0003.zip	■ Bausteinbibliothek für Siemens SIMATIC Manager. ■ Für den Einsatz in CPUs von VIPA bzw. S7-300 CPUs von Siemens.

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 VIPA-spezifischen Bausteine.

Werden anstelle der SFCs FCs verwendet, so werden diese von den VIPA CPUs ab Firmware 3.6.0 unterstützt.

4 Bausteinparameter

4.1 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": **Spezifischer Fehlercode** Den spezifischen Fehlercode finden Sie in der Beschreibung der einzelnen SFCs. Bit 14 ... 8 > "0": **Allgemeiner Fehlercode** Die möglichen allgemeinen Fehlercodesfinden 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).

Allgemeine und spezifische Fehlercodes RET_VAL

Fehlercode	Beschreibung
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 Systembausteine - *"System Blocks"*

5.1 Fetch/Write - *"Fetch/Write Communication"*

5.1.1 SFC 228 - RW_KACHEL - Kacheldirektzugriff

Beschreibung

Über diesen SFC haben Sie direkten Zugriff auf den 4kByte großen Kachelbereich der CPU. Der Kachelbereich verteilt sich auf 4 Kacheln mit einer Größe von jeweils 1kByte. Durch Angabe von Kachel-Nr., -Offset und Datenbreite haben Sie über den SFC 228 schreibenden und lesenden Zugriff auf einen gewünschten Kachelbereich.

Dieser SFC wurde zu Testzwecken und zum Aufbau proprietärer Kommunikationssysteme entwickelt, und steht dem Anwender uneingeschränkt zur Verfügung. Bitte beachten, dass Sie durch einen schreibenden Zugriff auf einen Kachelbereich direkt in eine Kommunikation eingreifen können!

Parameter

Name	Deklaration	Typ	Beschreibung
K_NR	IN	INT	Kachelnummer
OFFSET	IN	INT	Kacheloffset
R_W	IN	INT	Zugriff
SIZE	IN	INT	Datenbreite
RET_VAL	OUT	BYTE	Rückgabewert (0 = OK)
VALUE	IN_ OUT	ANY	Zeiger auf Bereich für Datentransfer

K_NR

Kachel-Nr.

- Geben Sie hier die Kachel-Nr. an, auf die Sie zugreifen möchten.
  - Wertebereich: 0 ... 3

OFFSET

Kachel-Offset

- Geben Sie hier einen Offset innerhalb der spezifizierten Kachel an.
  - Wertebereich: 0 ... 1023

R_W

Read/Write

- Über diesen Parameter spezifizieren Sie einen Lese- bzw. Schreibzugriff.
  - 0 = Lesezugriff
  - 1 = Schreibzugriff

SIZE

Größe

- Hiermit bestimmen Sie die Breite des Datenfelds, das Sie über *K_NR* und *OFFSET* definiert haben. Sie können die Werte 1, 2 und 4Byte einstellen.

RET_VAL (Rückgabewert)

Byte, in das eine Fehlermeldung zurückgeliefert wird.

VALUE

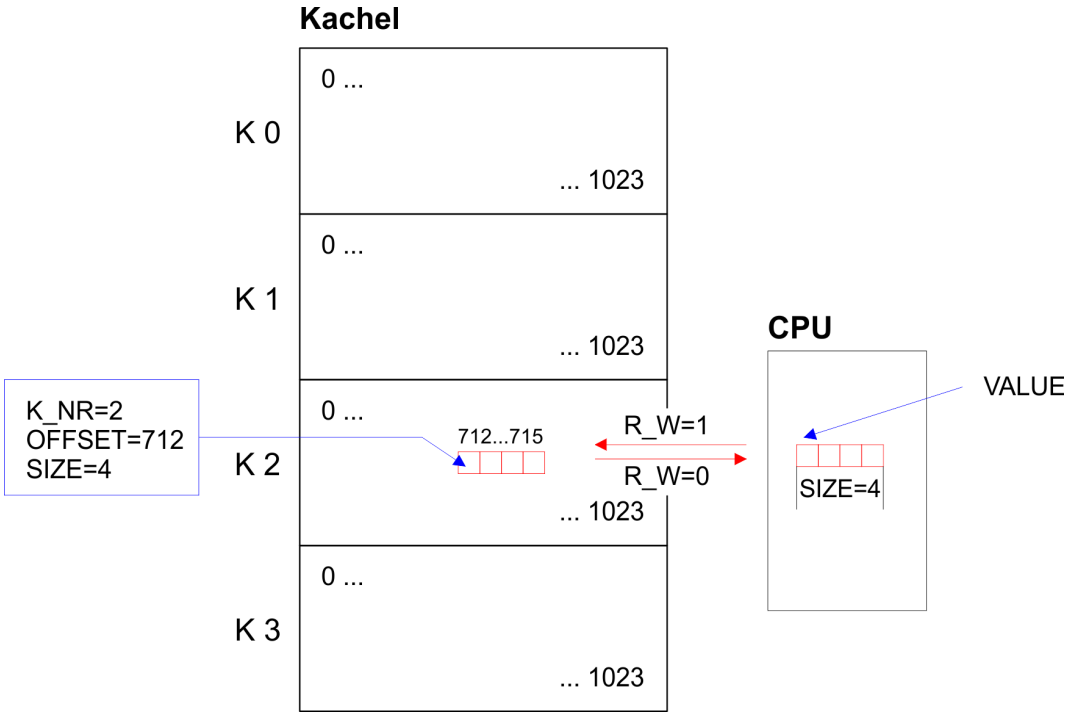
Ein-/Ausgabe-Bereich

- Mit diesem Parameter spezifizieren Sie den Ein- bzw. Ausgabebereich für den Datentransfer.
- Bei einem Lesezugriff finden in dem bis zu 4Byte breiten Bereich die Daten, die aus dem Kachelbereich gelesen werden.
- Bei einem Schreibzugriff werden aus diesem Bereich die bis zu 4Byte breiten Daten in den Kachelbereich übertragen.
  - Parameterart: Zeiger

Beispiel

Das nachfolgende Beispiel zeigt den lesenden Zugriff auf 4Bytes ab Byte 712 in Kachel 2. Die gelesenen 4Byte werden in DB10 ab Byte 2 abgelegt. Hierzu ist folgender Aufruf erforderlich:

```
CALL SFC 228
K_NR    :=2
OFFSET  :=712
R_W     :=0
SIZE    :=4
RET_VAL :=MB10
VALUE   :=P#DB10.DBX 2.0 Byte 4
```



Fehlermeldungen

Wert	Bedeutung
00h	kein Fehler aufgetreten
01h ... 05h	Interner Fehler: Für einen Parameter konnte keine gültige Adresse zugeordnet werden.
06h	die angegebene Kachel ist nicht vorhanden
07h	der Parameter SIZE ≠ 1, 2 oder 4 beim Lesezugriff

Wert	Bedeutung
08h	der Parameter SIZE ≠ 1, 2 oder 4 beim Schreibzugriff
09h	der Parameter R_W ist ≠ 0 oder 1

5.1.2 SFC 230 ... 238 - Kachelkommunikation

5.1.2.1 Paramenterbeschreibung

Übersicht

Durch die Hantierungsbausteine wird der Einsatz von Kommunikationsprozessoren in den CPUs von VIPA ermöglicht. Die Hantierungsbausteine steuern den gesamten Datenaustausch zwischen der CPU und den CPs. Vorteile der Hantierungsbausteine:

- wenig Anwenderprogrammspeicherplatz geht verloren
- kurze Laufzeiten der Bausteine

Die Hantierungsbausteine benötigen keine:

- Merkerbereiche
- Zeitbereiche
- Zählerbereiche

Alle nachfolgend behandelten Hantierungsbausteine haben eine einheitliche Schnittstelle zum Anwenderprogramm die folgende Parameter verwenden:

SSNR	-	Schnittstellennummer
ANR	-	Auftragsnummer
ANZW	-	Anzeigenwort (Doppelwort)
IND	-	Indirekte Angabe der relativen Anfangsadresse der Datenquelle bzw. des Datenziels
QANF/ZANF	-	Relative Anfangsadresse innerhalb des Typs
PAFE	-	Parametrierungsfehler
BLGR	-	Blockgröße

SSNR

Schnittstellen-Nummer

- Nummer der logischen Schnittstelle (Kacheladresse) auf die sich der betreffende Auftrag bezieht.
  - Parameterart: Integer
  - Sinnvoller Bereich: 0 ... 255

ANR

Auftragsnummer

- Angesprochene Auftragsnummer für die logische Schnittstelle.
  - Parameterart: Integer
  - Sinnvoller Bereich: 1 ... 223

ANZW

Anzeigenwort (Doppelwort)

- Adresse des Anzeigendoppelwortes im Anwenderspeicher, in dem die Abarbeitung des unter ANR angegebenen Auftrages angezeigt wird.
  - Parameterart: Doppelwort
  - Erlaubter Bereich: DW oder MW; belegt wird jeweils DW und DW+1 oder MW und MW+2
    Die Angabe DW bezieht sich auf den vor dem Aufruf aufgeschlagenen Datenbaustein oder auf den direkt angegebenen DB.

IND

Art der Parametrierung (direkt, indirekt)

- Über diesen Parameter bestimmen Sie die Art der Daten, auf die der Zeiger *QANF* zeigt.
  - 0: *QANF* zeigt direkt auf den Datenanfang der Quell- bzw. Zieldaten.
  - 1: der Zeiger *QANF/ZANF* zeigt auf eine Speicherzelle, ab der die Quell- bzw. Zieldaten definiert sind (indirekt).
  - 2: der Zeiger *QANF/ZANF* zeigt auf einen Speicherbereich in dem sich die Quell- und Zielangaben befinden (indirekt).
  - 5: der Zeiger *QANF/ZANF* zeigt auf eine Speicherzelle, ab der die Quell- bzw. Zielparameter und Parameter zum Anzeigenwort definiert sind (indirekt).
  - 6: der Zeiger *QANF/ZANF* zeigt auf einen Speicherbereich in dem die Quell- bzw. Zielparameter und Parameter zum Anzeigenwort definiert sind (indirekt).
  - Parameterart: Integer
  - Erlaubte Zahlen: 0, 1, 2, 5, 6

Bitte beachten Sie, dass bei IND = 5 bzw. IND = 6 der Parameter ANZW ignoriert wird!

QANF/ZANF

Relative Anfangsadresse der Datenquelle bzw. des Datenziels und bei *IND* = 5 bzw. *IND* = 6 des Anzeigenworts.

- Über diesen Parameter vom Typ Zeiger (Any-Pointer) können Sie die Relative Anfangsadresse und den Typ der Datenquelle (bei SEND) bzw. des Datenziels (bei RECEIVE) angeben.
- Bei *IND* = 5 bzw. *IND* = 6 befinden sich in der Datenquelle auch die Parameter zum Anzeigenwort.
  - Parameterart: Zeiger
  - Sinnvoller Bereich: DB, M, A, E

Beispiel:

```
P#DB10.DBX0.0 BYTE 16
P#M0.0 BYTE 10
P#E 0.0 BYTE 8
P#A 0.0 BYTE 10
```

BLGR

Blockgröße

- Bei Neustart wird mit Hilfe von "SYNCHRON" die Blockgröße (Größe der Datenblöcke) zwischen den Stationen ausgehandelt.
- Hierbei bedeutet große Bockgröße = hoher Datendurchsatz aber auch lange Laufzeit und damit hohe Zykluszeitbelastung.
- Kleine Blockgröße = kleiner Datendurchsatz aber auch kurze Laufzeiten der Bausteine.

Als Blockgröße kann eingestellt werden:

Wert	Blockgröße	Wert	Blockgröße
0	Default (64Byte)	4	128Byte
1	16Byte	5	256Byte
2	32Byte	6	512Byte
3	64Byte	255	512Byte

- Parameterart: Integer
- Möglicher Bereich: 0 ... 255

PAFE

Fehleranzeige bei Parametrierungsfehler

- Das hier angegebene "BYTE" (Ausgang, Merker) wird gesetzt, wenn der Baustein einen "Parametrierungsfehler" erkennt z.B. Schnittstelle (Anschaltung) nicht vorhanden oder unzulässige Parametrierung von QANF/ZANF erfolgte.
  - Parameterart: Byte
  - Sinnvoller Bereich: AB 0 ... AB127, MB 0...MB 255

5.1.2.2 Parameterübergabe

direkte/indirekte Parametrierung

Ein Hantierungsbaustein kann direkt oder indirekt parametriert werden. Nur der Parameter "*PAFE*" muss immer direkt angegeben werden. Bei der direkten Parametrierung verarbeitet der Hantierungsbaustein die beim Bausteinaufruf angegebenen Parameter unmittelbar. Bei der indirekten Parametrierung werden dem Hantierungsbaustein per Bausteinparameter Zeiger, die auf Parameterfelder (Datenbausteine bzw. Datenworte) zeigen, übergeben. Die Parameter *SSNR*, *ANR*, *IND* und *BLGR* sind von Typ "Integer" und können somit indirekt parametriert werden.

Beispiel

Direkte Parameterübergabe

```
CALL  SFC  230
       SSNR:=0
       ANR :=3
       IND :=0
       QANF:=P#A 0.0 BYTE 16
       PAFE:=MB79
       ANZW:=MD44
```

Indirekte Parameterübergabe

```
CALL  SFC  230
       SSNR:=MW10
       ANR :=MW12
       IND :=MW14
       QANF:=P#DB10.DBX0.0 BYTE 16
       PAFE:=MB80
       ANZW:=MD48
```

Bitte beachten Sie, dass die Merkerworte zuvor mit entsprechenden Werten zu laden sind.

5.1.2.3 Quell- bzw. Zielangaben

Übersicht

Sie haben die Möglichkeit die Angaben für Quelle, Ziel und für *ANZW* direkt anzugeben oder indirekt in einem Baustein abzulegen, auf den der Zeiger *QANF / ZANF* bzw. *ANZW* zeigt. Der Parameter *IND* dient als Umschaltkriterium für die direkte und indirekte Parametrierung.

Direkte Parametrierung der Quell- und Zielangaben (*IND* = 0)

Mit *IND* = 0 geben Sie an, dass der Zeiger *QANF / ZANF* direkt auf die Quell- bzw. Zieldaten zeigt. Nachfolgend sehen Sie eine Tabelle über mögliche *QANF / ZANF*-Parameter bei der direkten Parametrierung:

QTYP/ZTYP	Daten in DB	Daten in MB	Daten in AB Prozessabbild der Ausgänge	Daten in EB Prozessabbild der Eingänge
Zeiger: Beispiel:	`P#DBa.DBX b.0 BYTE CP#DB10.DBX 0.0 BYTE 8`	`P#M b.0 BYTE cP#M 5.0 BYTE 10`	`P#A b.0 BYTE cP#A 0.0 BYTE 2`	`P#E b.0 BYTE cP#E 20.0 BYTE 1`
DB, MB, AB, EB Bedeutung	`P#DBa` "a" steht für die DB-Nr., aus dem die Quelldaten entnommen werden oder in den die Zieldaten transferiert werden.	`P#M` kennzeichnet, dass die Daten in einem MB abgelegt sind.	`P#A` kennzeichnet, dass die Daten im Ausgangsbyte abgelegt sind.	`P#E` kennzeichnet, dass die Daten im Eingangsbyte abgelegt sind.
erlaubter Bereich für "a"	0 ... 32767	irrelevant	irrelevant	irrelevant
Daten-/Merker-Byte, AB, EB Bedeutung	DW-Nr., ab der die Daten entnommen oder geschrieben werden.	Merkerbyte-Nr., ab der die Daten entnommen oder geschrieben werden.	Ausgangs-Byte-Nr., ab der die Daten entnommen oder geschrieben werden.	Eingangsbyte-Nr., ab der die Daten entnommen oder geschrieben werden.
erlaubter Bereich für "b"	0.0 ... 2047.0	0 ... 255	0 ... 127	0 ... 127
`BYTE c` Bedeutung erlaubter Bereich für "c"	Länge des Quell-/ Ziel-Datenblocks in Worten. 1 ... 2048	Länge des Quell-/ Ziel-Datenblocks in Bytes. 1 ... 255	Länge des Quell-/ Ziel-Datenblocks in Bytes. 1 ... 128	Länge des Quell-/ Ziel-Datenblocks in Bytes. 1 ... 128

Indirekte Parametrierung der Quell- und Zielangaben (*IND* = 1 oder *IND* = 2)

Bei der indirekten Adressierung zeigt *QANF / ZANF* auf einen Speicherbereich, in dem die Adressen der Quell- bzw. Ziel-Bereiche hinterlegt sind. Hierbei können Sie entweder für Datenquelle und Datenziel einen Bereich angeben (*IND* = 1) oder für Datenquelle und Datenziel jeweils einen Bereich bestimmen (*IND* = 2). In der nachfolgenden Tabelle finden Sie mögliche QANF / ZANF-Parameter bei der indirekten Parametrierung:

QTYP/ZTYP	IND = 1		IND = 2		
Bedeutung	Indirekte Adressierung für Quell- **oder** Zielparameter. Die Quell- oder Zielparameter werden in einem DB hinterlegt. *QANF/ZANF:*		Indirekte Adressierung für Quell- **und** Zielparameter. Die Quell- und Zielparameter werden hintereinander in einem DB hinterlegt. *QANF/ZANF:*		
	DW +0	Datentyp Quelle	DW +0	Datentyp Quelle	Beschreibung Datenquelle
	+2	DB-Nr. bei Typ "DB" ansonsten irrelevant	+2	DB-Nr. bei Typ "DB" ansonsten irrelevant	
	+4	Anfangsadresse	+4	Anfangsadresse	
	+6	Länge in Byte	+6	Länge in Byte	
			+8	Datentyp Ziel	Beschreibung Datenziel
			+10	DB-Nr. bei Typ "DB" ansonsten irrelevant	
			+12	Anfangsadresse	
			+14	Länge in Byte	
erlaubte DB-Nr.	0 ... 32767		0 ... 32767		

QTYP/ZTYP	IND = 1	IND = 2
Daten-Wort Bedeutung	DW-Nr., ab der die Daten hinterlegt sind	DW-Nr., ab der die Daten hinterlegt sind
erlaubter Bereich	0.0 ... 2047.0	0.0 ... 2047.0
Länge Bedeutung	Länge des DBs in Byte	Länge des DBs in Byte
erlaubter Bereich	8 fix	16 fix

Indirekte Parametrierung von Quell- Zielangaben und *ANZW* (*IND* = 5 oder *IND* = 6)

Bei der indirekten Adressierung zeigt *QANF / ZANF* auf einen Speicherbereich, in dem die Adressen der Quell- bzw. Ziel-Bereiche und des Anzeigeworts hinterlegt sind. Hierbei können Sie entweder für Datenquelle oder -ziel und Anzeigenwort einen Bereich angeben (*IND* = 5) oder für Datenquelle, Datenziel und Anzeigenwort jeweils getrennte Bereiche bestimmen (*IND* = 6). In der nachfolgenden Tabelle finden Sie mögliche *QANF / ZANF*-Parameter bei der indirekten Parametrierung:

QTYP/ZTYP	IND = 5			IND = 6		
Bedeutung	Indirekte Adressierung für Quell- oder Zielparameter und Anzeigenwort.Die Quell- oder Ziel und *ANZW*-Parameter werden hintereinander in einem DB hinterlegt. *QANF/ZANF*			Indirekte Adressierung für Quell- und Zielparameter und Anzeigenwort.Die Quell-, Ziel- und *ANZW*-Parameter werden hintereinander in einem DB hinterlegt. *QANF/ZANF*		
	DW +0	Datentyp Quelle	Beschreibung Datenquelle/- ziel	DW +0	Datentyp Quelle	Beschreibung Datenquelle
	+2	DB-Nr. bei Typ "DB" ansonsten irrelevant		+2	DB-Nr. bei Typ "DB" ansonsten irrelevant	
	+4	Anfangsadresse		+4	Anfangsadresse	
	+6	Länge in Byte		+6	Länge in Byte	
	+8	Datentyp Quelle	Beschreibung Anzeigenwort	+8	Datentyp Ziel	Beschreibung Datenziel
	+10	DB-Nr. bei Typ "DB" ansonsten irrelevant		+10	DB-Nr. bei Typ "DB" ansonsten irrelevant	
	+12	Anfangsadresse		+12	Anfangsadresse	
				+14	Länge in Byte	
				+16	Datentyp Quelle	Beschreibung Anzeigenwort
				+18	DB-Nr. bei Typ "DB" ansonsten irrelevant	
				+20	Anfangsadresse	
erlaubte DB-Nr.	0 ... 32767			0 ... 32767		
Daten-Wort Bedeutung	DW-Nr., ab der die Daten hinterlegt sind			DW-Nr., ab der die Daten hinterlegt sind		
erlaubter Bereich	0.0 ... 2047.0			0.0 ... 2047.0		

QTYP/ZTYP	IND = 5	IND = 6
Länge Bedeutung	Länge des DBs in Byte	Länge des DBs in Byte
erlaubter Bereich	14 fix	22 fix

5.1.2.4 Anzeigenwort *ANZW*

Status- und Fehleranzeigen

Status und Fehleranzeigen liefern die Hantierungsbausteine:

- über das Anzeigenwort *ANZW* (Informationen zur Auftragsbearbeitung).
- über das Parametrierfehlerbyte *PAFE* (Anzeige einer fehlerhaften Auftragsparametrierung).

Inhalt und Aufbau Anzeigenwort *ANZW*

Das "Anzeigenwort" zeigt den Zustand für einen bestimmten Auftrag auf einem CP an. Im SPS-Programm sollte für jeden Auftrag ein eigenes "Anzeigenwort" für jeden definierten Auftrag bereitgestellt werden. Das Anzeigenwort hat den folgenden prinzipiellen Aufbau:

Byte	Bit 7 ... Bit 0
0	■ Bit 3 ... Bit 0: Fehlerverwaltung CPU – 0: kein Fehler – 1 ... 5: CPU-Fehler – 6 ... 15: CP-Fehler ■ Bit 7 ... Bit 4: reserviert
1	Statusverwaltung CPU ■ Bit 0: Handshake sinnvoll (Daten vorhanden) – 0: RECEIVE gesperrt – 1: RECEIVE freigegeben ■ Bit 1: Auftrag läuft – 0: SEND/FETCH freigegeben – 1: SEND/FETCH gesperrt ■ Bit 2: Auftrag fertig ohne Fehler ■ Bit 3: Auftrag fertig mit Fehler Datenverwaltung Hantierungsbaustein ■ Bit 4: Datenübernahme/-übergabe läuft ■ Bit 5: Datenübergabe erfolgt ■ Bit 6: Datenübernahme erfolgt ■ Bit 7: Disable/Enable Datenblock – 1: gesperrt – 0: freigegeben
2 ... 3	Längenwort Hantierungsbaustein

Im "Längenwort" hinterlegen die Hantierungsbausteine (SEND, RECEIVE) die für den entsprechenden Auftrag bereits transferierten Daten; empfangene Daten in Empfangsaufträgen; bereits gesendete Daten in Sendeaufträgen. Die Anzeige im "Längenwort" erfolgt immer in Bytes und absolut.

Fehlerverwaltung Byte 0, Bit 0 ... Bit 3	In diesen Bits werden die Fehleranzeigen des Auftrags angezeigt. Diese Fehleranzeigen sind nur gültig, wenn auch gleichzeitig das Bit "Auftrag fertig mit Fehler" im Statusbit gesetzt ist.

Folgende Fehlermeldungen können ausgegeben werden:

0 - **kein Fehler**

Sollte das Bit "Auftrag fertig mit Fehler" gesetzt sein, so hat der CP die Verbindung neu aufbauen müssen, wie z.B. nach einem Neustart oder RESET.

1 - **falscher Q/ZTYP am HTB**

Auftrag wurde mit falscher TYP-Kennung parametriert.

2 - **Bereich im AG nicht vorhanden**

Beim Anstoß des Auftrags wurde eine falsche DB-NR parametriert.

3 - **Bereich im AG zu klein**

Die Summe aus Q/ZANF und Q/ZLAE überschreitet die Bereichsgrenzen. Die Bereichsgrenze wird bei Datenbausteinen durch die Bausteingröße bestimmt. Bei Merkern, Zeiten, Zählern usw. ist die Bereichsgröße AG-abhängig.

4 - **QVZ-Fehler im AG**

Mit dem Quell- bzw. Zielparameter wurde ein Bereich im AG angegeben, dessen Speicher defekt oder nicht bestückt ist. Der QVZ-Fehler kann nur bei Q/ZTYP AS, PB, QB oder bei Speicherdefekten auftreten.

5 - **Fehler beim Anzeigenwort**

Das parametrierte Anzeigenwort kann nicht bearbeitet werden. Dieser Fehler tritt auf, wenn mit *ANZW* ein Datenwort bzw. Doppelwort angegeben wurde, das sich nicht oder nicht mehr in dem spezifizierten Datenbaustein befindet d.h. DB zu klein oder nicht vorhanden.

6 - **kein gültiges ORG-Format**

Das Datenziel bzw. die Datenquelle ist weder beim Hantierungsbaustein (Q/TYP="NN") noch im Verbindungsbaustein angegeben.

7 - **Reserviert**

8 - **keine freien Transportverbindungen**

Die Transportverbindungskapazitäten sind überschritten. Löschen Sie unnötige Verbindungen.

9 - **Remote-Fehler**

Bei einem RAD/WRITE-Auftrag ist ein Fehler im Kommunikationspartner aufgetreten.

A - **Verbindungsfehler**

Die Verbindung für einen Auftrag ist nicht bzw. noch nicht aufgebaut. Der Fehler verschwindet, sobald eine Verbindung aufgebaut werden kann. Sind alle Verbindungen des CPs unterbrochen, so deutet dies auf einen Defekt der Baugruppe oder des Buskabel hin. Der Fehler kann auch durch eine fehlerhafte Parametrierung ausgelöst werden, wie z.B. fehlerhafte Adressierung.

B - **Handshakefehler**

Dies kann ein Systemfehler sein oder die Datenblockgröße ist zu groß gewählt.

C - **Anstoßfehler**

Zum Anstoß des Auftrags wurde ein falscher Hantierungsbaustein benutzt oder ein zu großer Datenblock übergeben.

D - **Abbruch nach RESET**

Hier handelt es sich um eine Betriebsmeldung. Bei Priorität 1 und 2 ist die Verbindung unterbrochen und wird neu aufgebaut, sobald sich der Kommunikationspartner auf eine neue Verbindung eingestellt hat. Bei Priorität 3 Verbindungen ist die Verbindung gelöscht, ein neuer Anstoß ist möglich.

E - **Auftrag mit Urladefunktion**

Dies ist eine Betriebsmeldung. Der Auftrag ist ein RAD/WRITE-PASSIV und kann vom AG aus nicht gestartet werden.

F - **Auftrag nicht vorhanden**Der angesprochene Auftrag ist nicht auf dem CP parametriert. Dieser Fehler kann auftreten, wenn SSNR/A-NR Kombination im Hantierungsbaustein falsch oder kein Verbindungsbaustein eingetragen ist.

Die Bits 4 bis 7 von Byte 2 sind für Erweiterungen reserviert.

Statusverwaltung Byte 1, Bit 0 ... Bit 3

Hier können Sie erkennen, ob ein Auftrag bereits gestartet ist, ob hierbei Fehler aufgetreten sind oder ob der Auftrag gesperrt ist, dass beispielsweise eine virtuelle Verbindung nicht mehr besteht.

- **Bit 0 - Handshake sinnvoll**
  - Setzen:
    Durch die Anschaltung entsprechend der "Löschen"-Anzeige im Auftragsstatus-Bit. Handshake sinnvoll (= 1) wird beim RECEIVE-Baustein genutzt. (Telegramm vorhanden bei PRO 1 oder RECEIVE-Anstoß möglich bei PRO 2/3).
  - Auswerten:
    Durch den RECEIVE-Baustein: Nur wenn das Bit gesetzt ist, leitet der RECEIVE den Handshake mit dem CP ein. Durch die Anwendung: Für RECEIVE-Anfrage (Abfrage, ob Telegramm vorhanden bei PRO 1).
- **Bit 1 - Auftrag läuft**
  - Setzen:
    Durch die Anschaltung, wenn Auftrag an CP erteilt ist.
  - Löschen:
    Durch die Anschaltung, wenn ein Auftrag abgearbeitet ist (z.B. Quittung eingetroffen).
  - Auswerten:
    Durch die Hantierungsbausteine: Ein neuer Auftrag wird nur erteilt, wenn der "alte" Auftrag abgearbeitet ist. Durch den Anwender: um zu erfahren, ob das Triggern eines neuen Auftrags sinnvoll ist.
- **Bit 2 - Auftrag fertig ohne Fehler**
  - Setzen:
    Durch die Anschaltung, wenn der entsprechende Auftrag ohne Fehler abgeschlossen wurde.
  - Löschen:
    Durch die Anschaltung, wenn der Auftrag erneut ausgelöst wird.
  - Auswerten:
    Durch den Anwender zur Prüfung, ob der Auftrag fehlerlos abgeschlossen wurde.
- **Bit 3 - Auftrag fertig mit Fehler**
  - Setzen:
    Durch die Anschaltung, wenn der entsprechende Auftrag mit Fehler abgeschlossen wurde. Die Fehlerursache ist dann im High-Teil des Anzeigenwortes verschlüsselt.
  - Löschen:
    Durch die Anschaltung, wenn der Auftrag erneut ausgelöst wird.
  - Auswerten:
    Durch den Anwender: Zur Prüfung, ob der Auftrag mit Fehler abgeschlossen, wurde. Ist die Kennung "Auftrag fertig mit Fehler" gesetzt, steht im High-Byte des Anzeigenwortes die Fehlerursache.

Datenverwaltung Byte 1, Bit 4 ... Bit 7

Hier ist verschlüsselt, ob der Datentransfer für den Auftrag noch läuft oder ob die Datenübergabe bzw. Datenübernahme bereits abgeschlossen ist. Mit dem Bit "Enable / Disable" kann der Datentransfer für den Auftrag gesperrt werden. (Disable = 1; Enable = 0).

- **Bit 4 - Datenübernahme / Datenübergabe läuft**
  - Setzen:
    Durch die Hantierungsbausteine SEND, RECEIVE, wenn die Übergabe/Übernahme für einen Auftrag begonnen wurde, z.B. wenn Daten über die ALL-Funktion (DBA-Ersatz) ausgetauscht werden, der Anstoß jedoch mit SEND-DIREKT erfolgte.
  - Löschen:
    Durch die Hantierungsbausteine SEND, RECEIVE, wenn der Datenaustausch für einen Auftrag beendet ist (letzter Teilblock übertragen).
  - Auswerten:
    Durch den Anwender: Während der Datenübertragung CP <<->> AG darf der Anwender den Datensatz eines Auftrags nicht mehr verändern. Bei PRO 0/1 Aufträgen ist dies unkritisch, da hierbei der Datenaustausch in einem Baustein-Durchlauf erledigt werden kann. Größere Datenmengen können jedoch nur in Blöcken übertragen werden, wobei diese Blockung über mehrere AG-Zyklen verteilt wird. Zur Wahrung der Datenkonsistenz ist zu prüfen ob der Datenblock gerade übertragen wird, bevor dessen Inhalt geändert wird.
- **Bit 5 - Datenübergabe erfolgt**
  - Setzen:
    Durch den Hantierungsbaustein SEND, wenn die Datenübergabe für einen Auftrag erfolgt ist.
  - Löschen:
    Durch den Hantierungsbaustein SEND, wenn für einen neuen Auftrag (neue TRIGGERN) mit dem Transfer der Daten begonnen wurde. Durch den Anwender: Wenn die Auswertung erfolgte (Flankenbildung).
  - Auswerten:
    Durch den Anwender: Mit diesem Bit ist zu ermitteln, ob der Datensatz für einen Auftrag schon auf den CP übertragen wurde bzw. wann ein neuer Datensatz für einen laufenden Auftrag (z.B. zyklische Übertragung) bereitgestellt werden kann.
- **Bit 6 - Datenübernahme erfolgt**
  - Setzen:
    Durch RECEIVE, wenn die Übernahme von Daten für einen Auftrag abgeschlossen wurde.
  - Löschen:
    Durch RECEIVE, wenn für einen neuen Auftrag (neue TRIGGERN) mit dem Transfer der Daten ins AG begonnen wurde. Durch den Anwender, wenn die Auswertung erfolgt (Flankenbildung).
  - Auswerten:
    Durch den Anwender: Mit diesem Bit kann der Anwender ermitteln, ob der Datensatz eines Auftrags schon auf das AG übertragen wurde bzw. wann ein neuer Datensatz für einen laufenden Auftrag ins AG transferiert wurde.
- **Bit 7 - Disable / Enable Datenblock**
  - Setzen:
    Durch den Anwender, um das Beschreiben eines Bereichs durch den RECEIVE-Baustein bzw. das Auslesen aus einem Bereich durch den SEND-Baustein zu verhindern (nur beim 1. Datenblock).
  - Löschen:
    Durch den Anwender, um den zugehörigen Datenbereich freizugeben.
  - Auswerten:
    Durch die Hantierungsbausteine SEND und RECEIVE. Ist das Bit 7 gesetzt, führen die Bausteine keinen Daten-verkehr durch, sondern melden dem CP den Fehler.

Längenwort Byte 2 und Byte 3

Im Längenwort hinterlegen die Hantierungsbausteine (SEND, RECEIVE) die Menge für den entsprechenden Auftrag bereits transferierten Daten, d.h. bei Empfangsaufträgen die bereits empfangene Datenmenge, bei Sendeaufträgen die bereits gesendete Datenmenge.

Beschreiben: - Durch SEND, RECEIVE während des Datenaustausches. Das "Längen-Wort" wird errechnet aus: aktuelle Übertragungsanzahl + Anzahl bereits ausgetauschter Daten

Löschen: - Durch Überschreiben bzw. mit jedem neuen SEND, RECEIVE, FETCH. Wenn das Bit "Auftrag fertig ohne Fehler" bzw. "Datenübergabe/-Übernahme erfolgt" gesetzt ist, steht im "Längen-Wort" die aktuelle Quell- bzw. Ziellänge. Wenn das Bit "Auftrag fertig mit Fehler" gesetzt ist, beinhaltet das Längenwort die bis zum Fehlerfall übertragene Datenanzahl.

Status- und Fehleranzeigen

Im Folgenden sind wichtige Status- und Fehlermeldungen der CPU aufgeführt, die im "Anzeigenwort" erscheinen können. Die Darstellung hierbei erfolgt in "HEX"-Mustern. Das Zeichen X steht für "nicht bestimmt" bzw. für "irrelevant"; Nr. ist die Fehlernummer.

X F X A - Die Fehlerkennung "F" besagt, dass der entsprechende Auftrag auf dem CP nicht definiert ist. Die Statuskennung A bewirkt, dass der Auftrag gesperrt ist (für SEND / FETCH und RECEIVE).

X A X A - Die Fehlerkennung "A" zeigt an, dass die Verbindung des Kommuni-kations-auftrags nicht bzw. noch nicht aufgebaut ist. Mit der Statuskennung "A" ist sowohl der SEND als auch der RECEIVE und FETCH gesperrt.

X 0 X 8 - Die Verbindung ist neu aufgebaut (z.B. nach einem CP-Neuanlauf), der SEND ist freigegeben (SEND-Kommunikationsauftrag).

X 0 X 9 - Die Verbindung ist neu aufgebaut, der RECEIVE ist freigegeben (RECEIVE-Kommunikationsauftrag).

X 0 2 4 - Der SEND ist ohne Fehler abgearbeitet worden, die Daten wurden übertragen.

X 0 4 5 - Der RECEIVE ist ohne Fehler abgearbeitet worden, die Daten sind auf dem AG angekommen.

X 0 X 2 - Der SEND-, RECEIVE-, READ- bzw. WRITE-Auftrag läuft. Bei SEND hat sich der Partner noch nicht auf den RECEIVE eingestellt. Bei RECEIVE hat der Partner noch kein SEND abgesetzt.

Wichtige Anzeigenwortzustände

Anzeigen bei SEND

Zustand unter H1	**Prio 0/1**	**Prio 2**	**Prio 3/4**
Zustand unter TCP/IP	**Prio 1**	**Prio 2**	**Prio 3**
nach Neustart	0 A 0 A	0 A 0 A	0 0 0 8
nach Verbindungsaufbau	X 0 X 8	X 0 X 8	.....
nach Anstoß	X 0 X 2	X 0 X 2	X 0 X 2
fertig ohne Fehler	X 0 2 4	X 0 2 4	X 0 2 4
fertig mit Fehler	X Nr X 8	X Nr X 8	X Nr X 8
nach RESET	X D X A	X D X A	X D X 8

Anzeigen bei RECEIVE

Zustand unter H1	Prio 0/1	Prio 2	Prio 3/4
Zustand unter TCP/IP	**Prio 1**	**Prio 2**	**Prio 3**
nach Neustart	0 A 0 A	0 A 0 A	0 0 0 1
nach Verbindungsaufbau	X 0 X 4	X 0 0 9	.....
nach Anstoß	X 0 X 2	X 0 X 2	X 0 X 2
Telegramm da	X 0 X 1	.....	.....
fertig ohne Fehler	X 0 4 1	X 0 4 5	X 0 4 5
fertig mit Fehler	X Nr X 8	X Nr X 9	X Nr X 9
nach RESET	X D X A	X D X A	X D X 9

Anzeigen bei READ/WRITE-AKTIV

Zustand unter H1	Prio 0/1	Prio 2	Prio 3/4
Zustand unter TCP/IP	**Prio 1**	**Prio 2**	**Prio 3**
nach Neustart		0 A 0 A	
nach Verbindungsaufbau		X 0 0 8	
nach Anstoß		X 0 X 2	
READ fertig		X 0 4 4	
WRITE fertig		X 0 2 4	
fertig mit Fehler		X Nr X 8	
nach RESET		X D X A	

5.1.2.5 Parametrierfehler *PAFE*

PAFE wird gesetzt (Ausgang oder Merker), wenn der Baustein einen "Parametrierungsfehler" erkennt, z.B. Schnittstelle nicht vorhanden oder unzulässige Parametrierung von *QANF* / *ZANF* erfolgte. *PAFE* hat folgenden Aufbau:

Byte	Bit 7 ... Bit 0
0	■ Bit 0: Fehler – 0: kein Fehler – 1: Fehler vorhanden, Fehler-Nr. in Bit 4 ... Bit 7 ■ Bit 3 ... Bit 1: reserviert ■ Bit 7 ... Bit 4: Fehler-Nr. – 0: kein Fehler – 1: falsches ORG-Format – 2: Bereich nicht vorhanden (DB nicht vorhanden) – 3: Bereich zu klein – 4: QVZ-Fehler – 5: falsches Anzeigenwort – 6: keine Quell-/Zielparameter bei SEND/RECEIVE ALL – 7: Schnittstelle nicht vorhanden – 8: Schnittstelle unklar – 9: Schnittstelle überlastet – A: reserviert – B: unzulässige Auftrags-Nr. – C: Schnittstelle des CPs quittiert nicht oder negativ – D: Parameter *BLGR* nicht zulässig – E: reserviert – F: reserviert

5.1.3 SFC 230 - SEND - Senden an Kachel

Beschreibung

Der SEND-Baustein dient zum Auslösen eines Sende-Auftrags zu einem CP. SEND wird im Normalfall im zyklischen Teil des Anwenderprogramms aufgerufen. Die Einbindung des Bausteins im Interrupt oder Weck-Programmteil ist zwar möglich, das Anzeigenwort (*ANZW*) kann hierbei jedoch nicht zyklisch aktualisiert werden, dies sollte durch den CONTROL-Baustein übernommen werden.

Der Verbindungsaufbau mit dem CP wird für die Datenübergabe und für die Aktivierung eines Send-Anstoßes nur dann aufgenommen, wenn:

- dem FB VKE (Verknüpfungsergebnis) "1" übergeben wurde.
- der CP den Auftrag freigegeben hat.
  (Bit "Auftrag läuft" im *ANZW* = 0).

Im Leerlauf des Bausteins wird nur das Anzeigenwort aktualisiert.

Parameter

Name	Deklaration	Typ	Beschreibung
SSNR	IN	INT	Schnittstellennummer
ANR	IN	INT	Auftragsnummer
IND	IN	INT	Adressierungsmodus
QANF	IN	ANY	Zeiger auf Datenquelle
PAFE	OUT	BYTE	Parametrierungsfehler
ANZW	IN_OUT	DWORD	Anzeigenwort

SEND_ALL zur Datenübergabe

Kann der CP die Daten direkt übernehmen, überträgt der SEND-Baustein die angeforderten Daten in einem Zug zum CP. Signalisiert der CP jedoch, dass er nur die Parameter des Auftrages wünscht oder ist die Anzahl der zu übergebenden Daten zu groß, werden dem CP nur die Sende-Parameter bzw. die Parameter mit dem ersten Datenblock übergeben. Die Daten oder der Folgeblock zu diesen Aufträgen fordert der CP über SEND_ALL bei der CPU an. Hierzu ist es jedoch erforderlich, dass mindestens einmal im Zyklus der Baustein SEND_ALL aufgerufen wird. Die Bedienoberfläche ist in allen "Anstoßarten" für den Anwender der Bausteine gleich, nur der Zeitpunkt der Datenübergabe ist bei den zuletzt genannten Fällen um mindestens einen CPU-Zyklus verschoben.

5.1.4 SFC 231 - RECEIVE - Empfangen von Kachel

Beschreibung

Der RECEIVE-Baustein dient zum Empfangen von Daten von einem CP. Im Normalfall wird der RECEIVE-Baustein im zyklischen Teil des Anwenderprogramms aufgerufen. Die Einbindung des Bausteins im Interrupt oder Weck-Programmteil ist ebenso möglich, dabei wird jedoch das Anzeigenwort nicht zyklisch aktualisiert. Diese Funktion muss dann der CONTROL-Baustein übernehmen.

Der Quittungsverkehr mit dem CP (Auftragsanstoß) wird vom RECEIVE-Baustein nur aufgenommen wenn:

- dem FB VKE "1" übergeben wurde und
- der CP den Auftrag freigeben hat (Bit "Handshake sinnvoll" = 1).

Parameter

Name	Deklaration	Typ	Beschreibung
SSNR	IN	INT	Schnittstellenummer
ANR	IN	INT	Auftragsnummer
IND	IN	INT	Adressierungsmodus
ZANF	IN	ANY	Zeiger auf Datenziel
PAFE	OUT	BYTE	Parametrierungsfehler
ANZW	IN_OUT	DWORD	Anzeigenwort

Im "Leerlauf" des Bausteins wird nur das Anzeigenwort aktualisiert. Der RECEIVE-Baustein verhält sich unterschiedlich je nach Art der Versorgung und der CP-Reaktion:

- Wird vom CP ein Parametersatz geliefert, obwohl der RECEIVE-Baustein selbst mit den Zielparametern versorgt wurde, haben die Parameterangaben am Baustein Priorität gegenüber dem Parametersatz vom CP.
- Große Datenmengen können nur in Blöcken übernommen werden. Hierzu ist es erforderlich, solche Folgeblöcke mit RECEIVE_ALL in die CPU zu übertragen. Der Aufruf des RECEIVE_ALL mindestens einmal im zyklischen Programmablauf pro CP-Schnittstelle ist daher immer dann erforderlich, wenn mit einem CP größere Datenblöcke ausgetauscht werden sollen. Ebenso ist die zyklische Einbindung des RECEIVE_ALL erforderlich, wenn der CP den RECEIVE nur zur Freigabe eines Empfangtelegramms benutzt und die Daten über die "Hintergrundkommunikation" der CPU übergibt.

5.1.5 SFC 232 - FETCH - Anfordern von Kachel

Beschreibung

Der FETCH-Baustein dient dem Auslösen eines "Holauftrags" auf einer Gegenstation. Mit dem FETCH-Auftrag werden Daten-Quelle und -Ziel definiert und die Datenquelle an die Gegenstation übertragen. Bei der CPU von VIPA erfolgt die Angabe von Quelle und Ziel über einen Zeiger-Parameter. Die Gegenstation stellt die Daten aus der Quelle bereit und schickt diese über SEND_ALL an die anfordernde Station zurück. Über RECEIVE_ALL werden die Daten empfangen und im Ziel abgelegt. Die Aktualisierung des Anzeigenworts erfolgt über FETCH bzw. CONTROL.

Der Quittungsverkehr für den Anstoß des FETCH wird nur aufgenommen, wenn:

- dem Baustein VKE "1" übergeben
- im entsprechenden CP-Anzeigenwort die Funktion freigegeben wurde (Auftrag läuft = 0).

Parameter

Name	Deklaration	Typ	Beschreibung
SSNR	IN	INT	Schnittstellennummer
ANR	IN	INT	Auftragsnummer
IND	IN	INT	Adressierungsmodus
ZANF	IN	ANY	Zeiger auf Datenziel
PAFE	OUT	BYTE	Parametrierungsfehler
ANZW	IN_OUT	DWORD	Anzeigenwort

Angaben zur indirekten Parametrierung ↬ Kapitel 5.1.2.3 "Quell- bzw. Zielangaben" auf Seite 15

5.1.6 SFC 233 - CONTROL - Control Kachel

Beschreibung

Der CONTROL-Baustein hat folgende Aufgaben:

- Aktualisierung des Anzeigenworts
- Abfrage, ob ein bestimmter Auftrag des CP zur Zeit "tätig" ist, z.B. Nachfrage nach einem Empfangstelegramm
- Abfrage des CP, welcher Auftrag zur Zeit bearbeitet wird

Der CONTROL-Baustein nimmt keinen Quittungsverkehr mit dem CP auf, sondern überträgt nur die Anzeigen aus dem "Auftragsstatus" zum parametrierten Anzeigenwort. Der Baustein ist nicht VKE abhängig und sollte im zyklischen Teil des Programms aufgerufen werden.

Parameter

Name	Deklaration	Typ	Beschreibung
SSNR	IN	INT	Schnittstellennummer
ANR	IN	INT	Auftragsnummer
PAFE	OUT	BYTE	Parametrierungsfehler
ANZW	IN_OUT	DWORD	Anzeigenwort

ANR

Bei einer $ANR \neq 0$ wird das Anzeigenwort in der gleichen Weise aufgebaut und bearbeitet wie bei allen anderen Hantierungsbausteinen. Wird der Parameter *ANR* mit 0 versorgt, überträgt der CONTROL-Befehl den Inhalt der Auftragsstatuszelle 0 zum LOW-Teil des Anzeigenworts. In die Auftragsstatuszelle 0 schreibt der CP die Nummer des aktuellen Auftrags, d.h. des Auftrags, der gerade bearbeitet wird, wie z.B. die Auftragsnummer eines Telegramms.

5.1.7 SFC 234 - RESET - Rücksetzen Kachel

Beschreibung

Die RESET ALL-Funktion wird mit der Auftragsnummer 0 angewählt. Sie setzt alle Aufträge dieser logischen Schnittstelle zurück; z.B. löscht sie alle Auftragsdaten und bricht alle laufenden Aufträge ab. Mit einer "direkten" Funktion ($ANR \neq 0$) wird nur der angegebene Auftrag auf der logischen Schnittstelle zurückgesetzt. Der Baustein arbeitet VKE-abhängig und kann von zyklischen, zeitgesteuerten oder alarmgesteuerten Programmteilen aus aufgerufen werden.

Parameter

Name	Deklaration	Typ	Beschreibung
SSNR	IN	INT	Schnittstellennummer
ANR	IN	INT	Auftragsnummer
PAFE	OUT	BYTE	Parametrierfehler

Betriebsarten

Der Baustein kennt folgende beiden Betriebsarten:

- RESET ALL
- RESET DIREKT

5.1.8 SFC 235 - SYNCHRON - Synchronisieren Kachel

Beschreibung

Der Baustein stellt im CPU-Anlauf die Synchronisation zwischen CPU und CP her und ist daher in den Anlauf-OBs aufzurufen. Gleichzeitig wird der Übergabebereich der Schnittstelle gelöscht und voreingestellt, sowie die Blockgröße zwischen CP und CPU ausgehandelt.

Parameter

Name	Deklaration	Typ	Beschreibung
SSNR	IN	INT	Schnittstellenummer
BLGR	IN	INT	Blockgröße
PAFE	OUT	BYTE	Parametrierfehler

Blockgröße

Zur Vermeidung von langen Zykluszeiten ist es sinnvoll große Datenmengen in kleinen Blöcken zwischen CPU und CP zu übertragen. Die Größe dieser Blöcke stellen Sie über die "Blockgröße" ein. Hierbei bedeutet große Bockgröße = hoher Datendurchsatz aber auch lange Laufzeit und damit hohe Zykluszeitbelastung. Kleine Blockgröße = kleiner Datendurchsatz aber auch kleine Laufzeiten der Bausteine. Als Blockgröße kann eingestellt werden:

Wert	Blockgröße	Wert	Blockgröße
0	Default (64Byte)	4	128Byte
1	16Byte	5	256Byte
2	32Byte	6	512Byte
3	64Byte	255	512Byte

Parameterart:	Integer
Möglicher Bereich:	0 ... 255

5.1.9 SFC 236 - SEND_ALL - Alles senden an Kachel

Beschreibung

Mit dem SEND_ALL-Baustein werden die Daten von der CPU an den CP unter Verwendung der eingestellten Blockgröße übermittelt. Die Lage und Größe des Datenbereichs, der mit SEND_ALL zu übermitteln ist, muss zuvor über einen SEND bzw. FETCH-Aufruf definiert werden. Im Anzeigenwort, das dem betreffenden Auftrag zugeordnet ist, werden die Bits "Enable/Disable", "Datenübergabe erfolgt" sowie "Datenübergabe läuft" ausgewertet oder beeinflusst.

Parameter

Name	Deklaration	Typ	Beschreibung
SSNR	IN	INT	Schnittstellennummer
PAFE	OUT	BYTE	Parametrierfehler
ANZW	IN_OUT	DWORD	Anzeigenwort

ANZW

Im Baustein-Anzeigenwort, das im SEND_ALL-Baustein parametriert ist, wird die aktuelle Auftragsnummer hinterlegt (0 bedeutet Leerdurchlauf). Die Anzahl der übertragenen Daten zu einem Auftrag zeigt SEND_ALL in dem Datenwort an, das dem Anzeigenwort folgt.

In folgenden Fällen ist mindestens einmal SEND_ALL im Zyklus-Baustein OB 1 aufzurufen:

- *wenn der CP selbständig Daten von der CPU anfordern kann.*
- *wenn ein CP-Auftrag mit einem SEND angestoßen wird, der CP die Daten zu diesem Auftrag jedoch erst über die "Hintergrundkommunikation" bei der CPU anfordert.*
- *wenn die Anzahl der Daten, die mit einem SEND dem CP übergeben werden sollen, größer als die eingestellte Blockgröße ist.*

5.1.10 SFC 237 - RECEIVE_ALL - Alles empfangen von Kachel

Beschreibung

Mit dem RECEIVE_ALL-Baustein werden die Daten, die vom CP empfangen werden, vom CP an die CPU unter Verwendung der eingestellten Blockgröße übermittelt. Die Lage und Größe des Datenbereichs, der mit RECEIVE_ALL zu übermitteln ist, muss zuvor über einen RECEIVE-Aufruf definiert werden. Im Anzeigenwort, das dem zu bearbeitenden Auftrag zugeordnet ist, werden die Bits "Enable/Disable", "Datenübernahme erfolgt" sowie "Datenübernahme/-übergabe läuft" ausgewertet oder beeinflusst und im Folgewort die "Empfangslänge" angezeigt.

Parameter

Name	Deklaration	Typ	Beschreibung
SSNR	IN	INT	Schnittstellennummer
PAFE	OUT	BYTE	Parametrierfehler
ANZW	IN_OUT	DWORD	Anzeigenwort

ANZW

Im Baustein-Anzeigenwort, das im RECEIVE_ALL-Baustein parametriert ist, wird die aktuelle Auftragsnummer hinterlegt, für den RECEIVE_ALL aktiv war. Im Leerlauf des RECEIVE_ALL ist das Baustein-Anzeigenwort gelöscht.

In folgenden Fällen ist mindestens einmal RECEIVE_ALL im Zyklus-Baustein OB 1 aufzurufen:

- *wenn der CP selbständig Daten an die CPU senden soll.*
- *wenn ein CP-Auftrag mit RECEIVE angestoßen wird, der CP die Daten zu diesem Auftrag jedoch erst über die "Hintergrundkommunikation" an die CPU weitergeben kann.*
- *wenn die Anzahl der Daten, die mit einem RECEIVE an die CPU übergeben werden sollen, größer als die eingestellte Blockgröße ist.*

5.1.11 SFC 238 - CTRL1 - Control1 Kachel

Beschreibung

Dieser Baustein ist identisch mit dem CONTROL-Baustein SFC 233 mit der Ausnahme, dass das Anzeigenwort vom Typ Pointer ist und noch *IND* als weiterer Parameter eingefügt wurde. Der Parameter *IND* ist für zukünftige Erweiterungen reserviert. Der CONTROL-Baustein hat folgende Aufgaben:

- Aktualisierung des Anzeigenworts
- Abfrage, ob ein bestimmter Auftrag des CP zur Zeit "tätig" ist, z.B. Nachfrage nach einem Empfangstelegramm
- Abfrage des CP, welcher Auftrag zur Zeit bearbeitet wird

Der CONTROL-Baustein nimmt keinen Quittungsverkehr mit dem CP auf, sondern überträgt nur die Anzeigen aus dem "Auftragsstatus" zum parametrierten Anzeigenwort. Der Baustein ist nicht VKE abhängig und sollte im zyklischen Teil des Programms aufgerufen werden.

Parameter

Name	Deklaration	Typ	Beschreibung
SSNR	IN	INT	Schnittstellennummer
ANR	IN	INT	Auftragsnummer
IND	IN	INT	reserviert
PAFE	OUT	BYTE	Parametrierfehler
ANZW	IN_OUT	DWORD	Anzeigenwort

ANR

Bei einer *ANR* ≠ 0 wird das Anzeigenwort in der gleichen Weise aufgebaut und bearbeitet wie bei allen anderen "Hantierungsbausteinen". Wird der Parameter *ANR* mit 0 versorgt, überträgt der CTRL1-Befehl den Inhalt der Auftragsstatuszelle 0 zum LOW-Teil des Anzeigenworts. In die Auftragsstatuszelle 0 schreibt der CP die Nummer des aktuellen Auftrags, d.h. des Auftrags, der gerade bearbeitet wird, wie z.B. die Auftragsnummer eines Telegramms.

IND

Der Parameter *IND* hat zur Zeit keine Funktion und ist für zukünftige Erweiterungen reserviert.

ANZW

Das Anzeigenwort *ANZW* ist vom Typ Pointer. Somit haben Sie auch die Möglichkeit das Anzeigenwort in einem Datenbaustein abzulegen.

5.2 MMC-Funktionen Standard-CPUs - *"MMC Functions standard CPUs"*

5.2.1 SFC 220 ... 222 - MMC-Zugriff

Übersicht

Mit den hier aufgeführten SFCs haben Sie die Möglichkeit den Zugriff auf eine MMC in Ihr Anwenderprogramm einzubinden. Hierbei können Sie bei einer gesteckten MMC eine neue Datei anlegen bzw. eine bestehende Datei für den Zugriff öffnen. Solange Sie keine neue Datei öffnen haben Sie über Lese-/Schreib-Befehle Zugriff auf diese Datei.

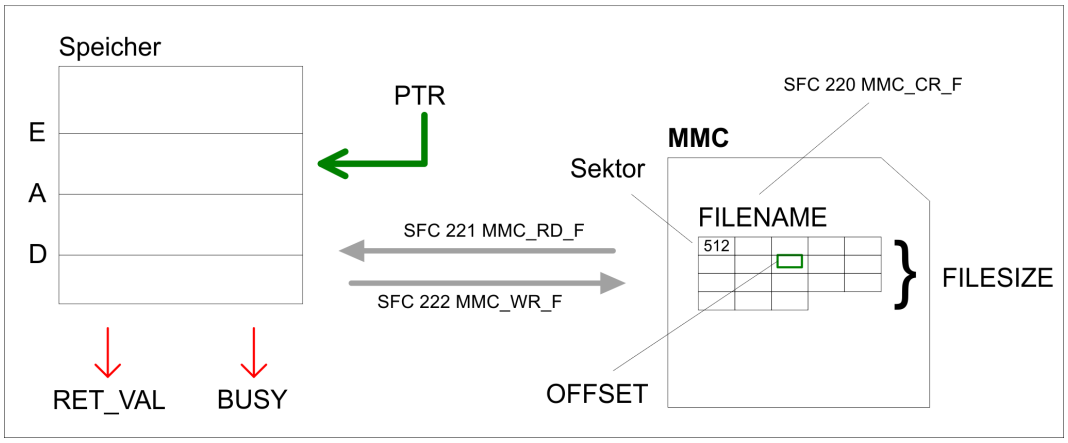
Einschränkungen

Für den Einsatz der SFCs 220, 221 und 222 sind folgende Einschränkungen zu beachten:

- Ein lesender bzw. schreibender Zugriff auf die MMC kann nur dann erfolgen, wenn die Datei zuvor mit dem SFC 220 angelegt bzw. geöffnet wurde.
- Es ist darauf zu achten, dass die Daten immer unfragmentiert auf der MMC abliegen, da nur zusammenhängende Datenblöcke gelesen bzw. geschrieben werden können.
- Werden Daten auf die MMC mit einem externen MMC-Kartenleser übertragen, so können diese fragmentiert sein d.h. die Daten werden in Blöcke aufgeteilt. Dies können Sie vermeiden, indem Sie die MMC vor dem Schreibzugriff formatieren.
- Bei einem Schreibzugriff von der CPU auf die MMC werden die Daten immer unfragmentiert auf der MMC abgelegt.
- Beim Öffnen einer schon bestehenden Datei sind für *FILENAME* und *FILESIZE* immer die Angaben zu verwenden, die Sie beim Anlegen der Datei verwendet haben.
- Eine MMC ist eingeteilt in Sektoren. Jeder Sektor hat eine Größe von 512Byte. Sektorübergreifendes Lesen bzw. Schreiben ist nicht möglich. Ein Zugriff auf sektorübergreifende Daten kann nur dann erfolgen, wenn Sie für jeden Sektor einen Schreib- bzw. Lesebefehl verwenden. Mit der Offset-Angabe bestimmen Sie den jeweiligen Sektor.

Die nachfolgende Abbildung zeigt die Verwendung der einzelnen SFCs und deren Variablen:

CPU



Für Lese- und Schreibzugriffe auf die MMC muss zuvor mit dem SFC 220 die Datei geöffnet werden!

5.2.2 SFC 220 - MMC_CR_F - MMC-Datei erstellen oder öffnen

Beschreibung

Unter Einsatz dieses Bausteins können Sie bei einer gesteckten MMC eine neue Datei anlegen bzw. eine bestehende Datei für den Zugriff öffnen. Solange Sie keine neue Datei öffnen, können Sie über Lese-/Schreib-Befehle auf diese Datei zugreifen. Näheres hierzu und zu den Einschränkungen ↳ *Kapitel 5.2.1 "SFC 220 ... 222 - MMC-Zugriff" auf Seite 35*.

Da der Aufruf des SFC im OB 1 zur Zykluszeit-Überschreitung führen kann, ist der SFC stattdessen im OB 100 aufzurufen.

Parameter

Name	Deklaration	Typ	Beschreibung
FILENAME	IN	STRING[254]	Dateiname
FILESIZE	IN	DWORD	Dateigröße
RET_VAL	OUT	WORD	Rückgabewert (0 = OK)

FILENAME

Geben Sie hier den Dateinamen an, unter dem Ihre Daten auf der MMC abzulegen sind bzw. abliegen. Der Dateiname mit Endekennung 00h darf eine maximale Länge von 13 Zeichen nicht überschreiten:

- 8 Zeichen für Name
- 1 Zeichen für "."
- 3 Zeichen für Dateierweiterung
- 1 Zeichen 00h als Endekennung

Aus softwaretechnischen Gründen müssen Sie das nächste Byte hinter dem Dateinamen mit 00h beschreiben (Endekennung Dateiname).

FILESIZE

Unter *FILESIZE* bestimmen Sie die Größe der Nutzdaten in Byte. Bei Zugriff auf eine schon bestehende Datei ist neben dem *FILENAME* die Angabe der vorgegebenen *FILESIZE* zwingend erforderlich. Die Angabe einer "Joker"-Länge wird zur Zeit nicht unterstützt.

Struktur

Byte 0	Byte 1	Byte 2	Byte 3	...	Byte 255
Max. Länge	belegte Länge	ASCII-Wert 1	ASCII-Wert 2	...	ASCII-Wert 254

RET_VAL (Rückgabewert)

Wort, in das eine Diagnose-/Fehlermeldung zurückgeliefert wird. 0 bedeutet, dass alles OK ist.

Wert	Beschreibung
Diagnosemeldungen	
0000h	keine Fehler (tritt nur beim Erzeugen einer neuen Datei auf).
0001h	Datei existiert schon, ist unfragmentiert und die Längenangabe *FILESIZE* ist identisch oder kleiner als die reale Dateigröße.
8001h	Es ist keine oder eine vom Typ unbekannte MMC gesteckt.
Fehlermeldungen	
8002h	Es befindet sich keine FAT auf der MMC.
A001h	Es wurde kein Dateiname angegeben. Diese Meldung kommt nur dann, wenn der Dateiname sich beispielsweise in einem nicht geladenen DB befindet.
A002h	Der angegebene Dateiname ist falsch (nicht 8.3 oder leer).
A003h	Die Datei existiert schon, aber die unter *FILESIZE* angegebene Größe ist größer als die existierende Datei.
A004h	Die Datei existiert schon, ist aber fragmentiert und kann nicht geöffnet werden.
A005h	Es ist kein ausreichender Speicherplatz auf der MMC vorhanden.
A006h	Es existiert kein freier Eintrag im Root-Verzeichnis. Abhängig von der eingesetzten MMC dürfen sich mindestens 16 bis maximal 512 Einträge im Root-Verzeichnis befinden.
B000h	Es ist ein interner Fehler aufgetreten.

5.2.3 SFC 221 - MMC_RD_F - MMC-Datei lesen

Beschreibung

Über den SFC 221 können Sie von einer gesteckten MMC lesen. Bitte beachten Sie, dass die Datei zuvor mit dem SFC 220 für den Zugriff zu öffnen ist und die Datei unfragmentiert vorzuliegen hat. Näheres hierzu und zu den Einschränkungen ↳ *Kapitel 5.2.1 "SFC 220 ... 222 - MMC-Zugriff" auf Seite 35*.

Parameter

Name	Deklaration	Typ	Beschreibung
PTR	IN	ANY	Zeiger auf Datenbereich für Lesedaten
OFFSET	IN	DWORD	Offset der Daten innerhalb der Datei
BUSY	OUT	BOOL	Auftragsstatus
RET_VAL	OUT	WORD	Rückgabewert (0 = OK)

PTR

Diese Variable vom Typ Pointer zeigt auf einen Datenbereich in der CPU, der mit dem Inhalt der MMC zu beschreiben ist.

OFFSET

Hiermit bestimmen Sie auf der MMC innerhalb des Files den Anfang der Daten, die in die CPU zu übertragen sind.

BUSY

Während der Datenübertragung bleibt dieses Bit gesetzt. Ist der Datentransfer abgeschlossen wird das Bit zurückgesetzt.

RET_VAL (Rückgabewert) Wort, in das eine Diagnose-/Fehlermeldung zurückgeliefert wird. 0 bedeutet, dass alles OK ist.

Wert	Bedeutung
0000h	keine Fehler (Daten wurden gelesen)
8001h	Es ist keine oder eine vom Typ unbekannte MMC gesteckt.
8002h	Es befindet sich keine FAT auf der MMC.
9000h	Es wurde versucht ein Bit zu lesen (Boolean-Variable). Das bitweise Lesen ist nicht möglich.
9001h	Pointerangabe ist fehlerhaft (zeigt z.B. außerhalb eines DBs)
9002h	Die Dateilänge wurde überschritten.
9003h	Es wurde versucht die Sektorgrenze von 512 zu überschreiten. Sektorübergreifendes Lesen ist nicht möglich.
B000h	Es ist ein interner Fehler aufgetreten.

5.2.4 SFC 222 - MMC_WR_F - MMC-Datei schreiben

Beschreibung Über den SFC 222 können Sie auf eine gesteckte MMC schreiben. Bitte beachten Sie, dass die Datei zuvor mit dem SFC 220 für den Zugriff zu öffnen ist und die Datei unfragmentiert vorzuliegen hat. Näheres hierzu und zu den Einschränkungen ↯ *Kapitel 5.2.1 "SFC 220 ... 222 - MMC-Zugriff" auf Seite 35*.

Parameter

Name	Deklaration	Typ	Beschreibung
PTR	IN	ANY	Zeiger auf Datenbereich für Schreibdaten
OFFSET	IN	DWORD	Offset der Daten innerhalb der Datei
BUSY	OUT	BOOL	Auftragsstatus
RET_VAL	OUT	WORD	Rückgabewert (0 = OK)

PTR Diese Variable vom Typ Pointer zeigt auf einen Datenbereich in der CPU, der die Daten beinhaltet, die auf die MMC zu schreiben sind.

OFFSET Hiermit bestimmen Sie auf der MMC innerhalb der Datei den Anfang der Daten, ab dem die Daten geschrieben werden.

BUSY Während der Datenübertragung bleibt dieses Bit gesetzt. Ist der Datentransfer abgeschlossen wird das Bit zurückgesetzt.

RET_VAL (Rückgabewert) Wort, in das eine Diagnose-/Fehlermeldung zurückgeliefert wird. 0 bedeutet, dass alles OK ist.

Wert	Bedeutung
0000h	keine Fehler
8001h	Es ist keine oder eine falsche MMC gesteckt.
8002h	Es befindet sich keine FAT auf der MMC.
9000h	Es wurde versucht ein Bit zu schreiben (Boolean-Variable). Das bitweise Schreiben ist nicht möglich.
9001h	Pointerangabe ist fehlerhaft (zeigt z.B. außerhalb eines DBs).
9002h	Die Dateilänge wurde überschritten.
9003h	Es wurde versucht die Sektorgrenze von 512 zu überschreiten. Sektorübergreifendes Lesen ist nicht möglich.
B000h	Es ist ein interner Fehler aufgetreten.

5.3 Datei-Funktionen SPEED7-CPUs - *"File Functions SPEED7 CPUs"*

5.3.1 FC/SFC 195 und FC/SFC 208...215 - Speicherkarten-Zugriff

Übersicht

Mit den FC/SFC 195 und FC/SFC 208 ... FC/SFC 215 haben Sie die Möglichkeit den Speicherkarten-Zugriff in Ihr Anwenderprogramm einzubinden. Folgende Parameter sind für den Einsatz der FC/SFCs erforderlich:

HANDLE, FILENAME

Der Zugriff erfolgt über eine *HANDLE*-Nr., die Sie durch Aufruf des FC/SFC 208 FILE_OPN bzw. FC/SFC 209 FILE_CRE einem *FILENAME* zuordnen können. Gleichzeitig dürfen maximal 4 *HANDLE* belegt sein (0 ... 3). Durch Schließen mit FC/SFC 210 FILE_CLO wird eine geöffnete Datei geschlossen und der *HANDLE* wieder freigegeben.

MEDIA

Geben Sie als Media-Format für die MMC eine 0 an. Andere Formate werden zur Zeit nicht unterstützt.

ORIGIN, OFFSET

Das Lesen und Schreiben erfolgt ab der Position einer Schreib/Lesemarke. Nach dem Öffnen bzw. neu Anlegen einer Datei befindet sich die Schreib/Lesemarke auf Position 0. Mit dem FC/SFC 213 FILE_SEK können Sie die Schreib/Lesemarke ab einer *ORIGIN*-Position um einen *OFFSET* (Anzahl Bytes) verschieben.

REQ, BUSY

- Mit *REQ* = 1 aktivieren Sie die entsprechende Funktion.
- Ist *REQ* = 0 erhalten Sie den aktuellen Status einer Funktion über *RETVAL* zurückgeliefert.
- *BUSY* = 1 zeigt an, dass die entsprechende Funktion bearbeitet wird.

RETVAL Nach Abarbeitung einer Funktion liefert *RETVAL* einen Zahlencode zurück:

RETVAL = 0:	Funktion wurde fehlerfrei ausgeführt.
0 < RETVAL < 7000h:	*RETVAL* = Länge der transferierten Daten (nur FC/SFC 211 und FC/SFC 212).
7000h ≤ RETVAL < 8000h:	Zeigt den Bearbeitungs-Status der Funktion.
RETVAL ≥ 8000h:	Kennzeichnet einen Fehler, der bei dem entsprechenden FC/SFC näher beschrieben ist.

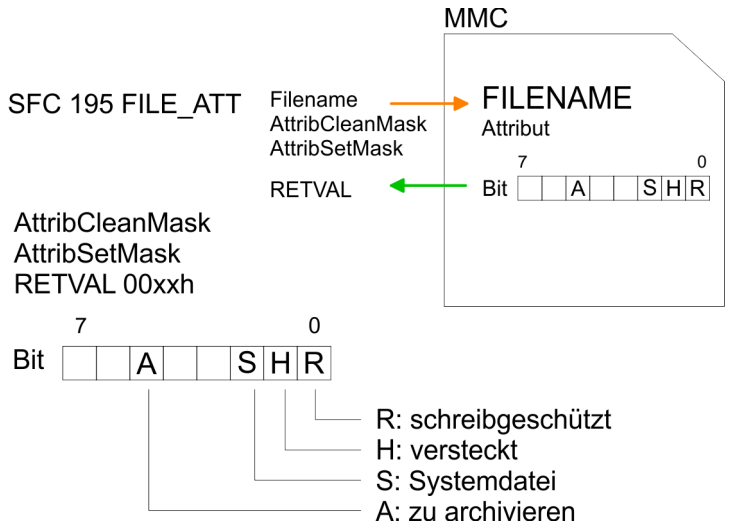
VORSICHT!
Für den Zugriff auf Speicherkarte sind folgende Hinweise zu berücksichtigen, deren Nichtbeachtung zu Datenverlust auf der Speicherkarte führen kann:

- Es dürfen maximal 4 Handle (0 ... 3) gleichzeitig belegt sein!
- Dateinamen müssen dem 8.3 Format oder Sonderzeichen entsprechen!
- Mit diesen FC/SFCs haben Sie ausschließlich Zugriff auf die oberste Verzeichnis-Ebene (Root-Verzeichnis) der Speicherkarte!
- Sie dürfen ausschließlich Dateien umbenennen bzw. löschen, die Sie zuvor mit FC/SFCs 210 FILE_CLO geschlossen haben!

5.3.2 FC/SFC 195 - FILE_ATT - Datei-Attribute ändern

Beschreibung

Unter Einsatz von FILE_ATT können Sie die Datei-Attribute einer Datei im Root-Verzeichnis der Speicherkarte ändern. Geben Sie hierzu einen Dateinamen an. Durch Vorgabe eines Bitmusters können Sie mit *ATTRIBCLEANMASK* das entsprechende Attribut rücksetzen bzw. mit *ATTRIBSETMASK* setzen. Bitte beachten Sie, dass hierbei das Setzen Vorrang gegenüber dem Rücksetzen hat. Über *RETVAL* 00xxh bekommen Sie den aktuellen Zustand der Dateiattribute nach Befehlsausführung zurückgeliefert. Wenn Sie *ATTRIBCLEANMASK* und *ATTRIBSETMASK* den Wert 00h übergeben, können Sie über *RETVAL* den aktuellen Status der Dateiattribute ermitteln.



Parameter

Parameter	Deklaration	Datentyp	Beschreibung
REQ	IN	BOOL	Funktion aktivieren
MEDIA	IN	INT	0 = MMC
FILENAME	IN	STRING[254]	Dateiname (muss im 8.3-Format sein)
ATTRIBCLEANMASK	IN	BYTE	Bit-Maske Datei-Attribute zurücksetzen
ATTRIBSETMASK	IN	BYTE	Bit-Maske Datei-Attribute setzen
RETVAL	OUT	WORD	Rückgabewert (00xxh=OK mit xx: Attribute)
BUSY	OUT	BOOL	Funktion wird bearbeitet

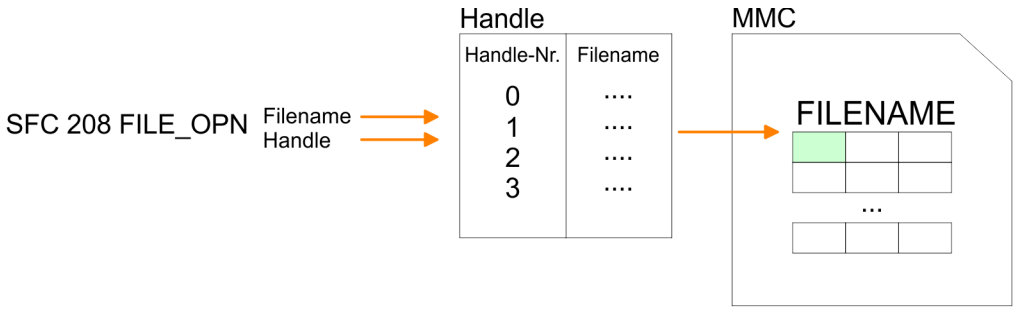
RETVAL (Rückgabewert) Codes, die *RETVAL* zurück liefert:

Code	Beschreibung
00xxh	OK, Attribute wurden geändert mit xx: Attribute
7000h	*REQ* = 0, *BUSY* = 0 (nichts ist zu tun)
7001h	*REQ* = 1, 1. Aufruf
7002h	Baustein wird bearbeitet
A001h	Der angegebene *MEDIA*-Typ ist falsch
A002h	Fehler im Parameter *ATTRIBSETMASK*
A004h	Datei *FILENAME* existiert nicht
A005h	*FILENAME* ist ein Verzeichnis
A006h	Datei ist geöffnet
A007h	Speicherkarte schreibgeschützt
A010h	Dateifehler FILENAME
A100h	Allgemeiner Filesystem-Fehler (z.B. keine Speicherkarte gesteckt)

5.3.3 FC/SFC 208 - FILE_OPN - Datei öffnen

Beschreibung

Eine Datei auf der Speicherkarte können Sie mit dem FC/SFC 208 öffnen. Hierbei wird ein *HANDLE* mit dem entsprechenden *FILENAME* verknüpft. Durch Angabe des *HANDLE* haben Sie jetzt solange lesenden und schreibenden Zugriff auf die Datei, bis die Datei mit FC/SFC 210 FILE_CLO wieder geschlossen wird. *REQ* = 1 löst die Funktion aus. Nach dem Öffnen steht die Schreib/Lesemarke auf 0.



Parameter

Parameter	Deklaration	Datentyp	Beschreibung
REQ	IN	BOOL	Funktion aktivieren
MEDIA	IN	INT	0 = MMC
FILENAME	IN	STRING[254]	Dateiname (muss im 8.3-Format sein)
HANDLE	IN	INT	Index der Datei 0 ... 3
RETVAL	OUT	WORD	Rückgabewert (0 = OK)
BUSY	OUT	BOOL	Funktion wird bearbeitet

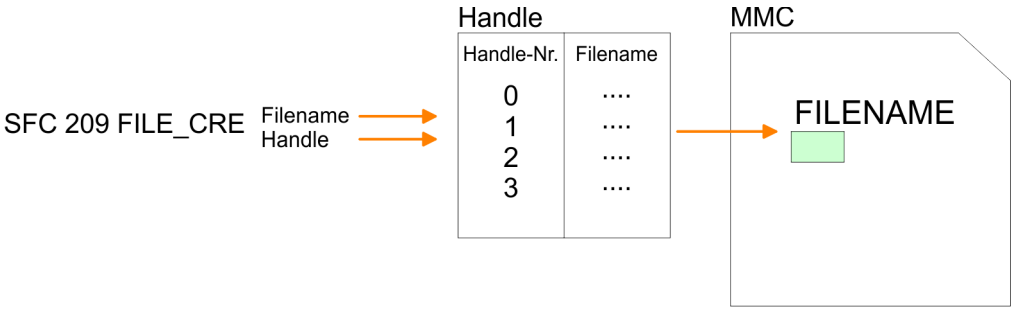
RETVAL (Rückgabewert) Codes, die RETVAL zurück liefert:

Code	Beschreibung
0000h	OK
7000h	*REQ* = 0, *BUSY* = 0 (nichts ist zu tun)
7001h	*REQ* = 1, 1. Aufruf
7002h	Baustein wird bearbeitet
8010h	Parameter *FILENAME* ist nicht verfügbar (z.B. DB nicht geladen)
8011h	*FILENAME* fehlerhaft (entspricht nicht dem Format 8.3 oder Sonderzeichen)
8100h	Der angegebene *HANDLE* ist ungültig
9001h	*HANDLE* ist bereits anderer Datei zugeordnet
9002h	Eine andere Funktion wurde über den *HANDLE* aufgerufen und ist fertig
9003h	Eine andere Funktion wurde über den *HANDLE* aufgerufen und ist nicht fertig
A000h	Systeminterner Fehler aufgetreten
A001h	Der angegebene *MEDIA*-Typ ist falsch
A003h	Es ist ein allgemeiner Fehler im Filesystem aufgetreten
A004h	Die unter *FILENAME* angegebene Datei existiert nicht bzw. ist ein Verzeichnis
A100h	Allgemeiner Filesystem-Fehler (z.B. keine Speicherkarte gesteckt)

5.3.4 FC/SFC 209 - FILE_CRE - Datei anlegen

Beschreibung

Durch Einsatz dieses Bausteins können Sie bei einer gesteckten Speicherkarte eine neue Datei mit dem entsprechenden Dateinamen anlegen und für den Lese-/Schreib-Zugriff öffnen. Bitte beachten Sie, dass ausschließlich Dateien auf der obersten Verzeichnis-Ebene erzeugt werden können. *REQ* = 1 löst die Funktion aus. Nach dem Öffnen steht die Schreib/Lesemarke auf 0.



Parameter

Parameter	Deklaration	Datentyp	Beschreibung
REQ	IN	BOOL	Funktion aktivieren
MEDIA	IN	INT	0 = MMC
FILENAME	IN	STRING[254]	Dateiname (muss im 8.3-Format sein)
HANDLE	IN	INT	Index der Datei 0 ... 3
RETVAL	OUT	WORD	Rückgabewert (0 = OK)
BUSY	OUT	BOOL	Funktion wird bearbeitet

RETVAL (Rückgabewert) Codes, die RETVAL zurück liefert:

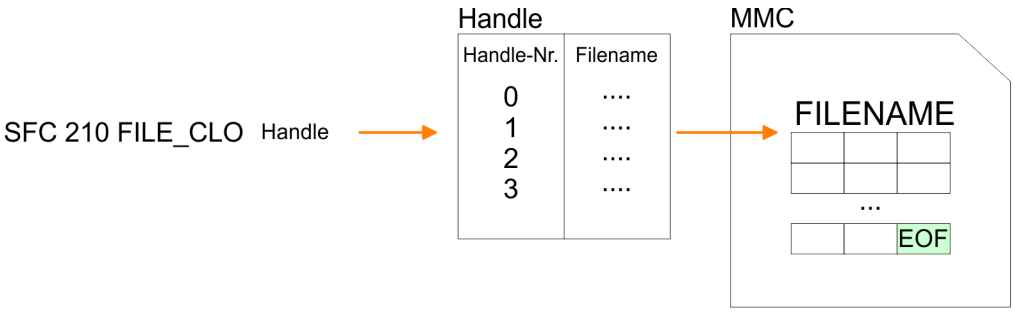
Code	Beschreibung
0000h	OK
7000h	*REQ* = 0, *BUSY* = 0 (nichts ist zu tun)
7001h	*REQ* = 1, 1. Aufruf
7002h	Baustein wird bearbeitet
8010h	Parameter *FILENAME* ist nicht verfügbar (z.B. DB nicht geladen)
8011h	*FILENAME* fehlerhaft (entspricht nicht dem Format 8.3 oder Sonderzeichen)
8100h	Der angegebene *HANDLE* ist ungültig
9001h	HANDLE ist bereits anderer Datei zugeordnet
9002h	Eine andere Funktion wurde über den *HANDLE* aufgerufen und ist fertig
9003h	Eine andere Funktion wurde über den *HANDLE* aufgerufen und ist nicht fertig
A000h	Systeminterner Fehler aufgetreten
A001h	Der angegebene *MEDIA*-Typ ist falsch
A003h	Es ist ein allgemeiner Fehler im Filesystem aufgetreten

Code	Beschreibung
A004h	Es ist kein Root-Eintrag im Verzeichnis verfügbar
A005h	Speicherkarte ist schreibgeschützt
A100h	Allgemeiner Filesystem-Fehler (z.B. keine Speicherkarte gesteckt)

5.3.5 FC/SFC 210 - FILE_CLO - Datei schließen

Beschreibung

Mit diesem Baustein können Sie eine geöffnete Datei schließen. Hierbei wird ein EOF (End of File) angefügt, die Datei geschlossen und der *HANDLE* wieder freigegeben. *REQ* = 1 löst die Funktion aus.



Parameter

Parameter	Deklaration	Datentyp	Beschreibung
REQ	IN	BOOL	Funktion aktivieren
HANDLE	IN	INT	Index der Datei 0 ... 3
RETVAL	OUT	WORD	Rückgabewert (0 = OK)
BUSY	OUT	BOOL	Funktion wird bearbeitet

RETVAL (Rückgabewert)

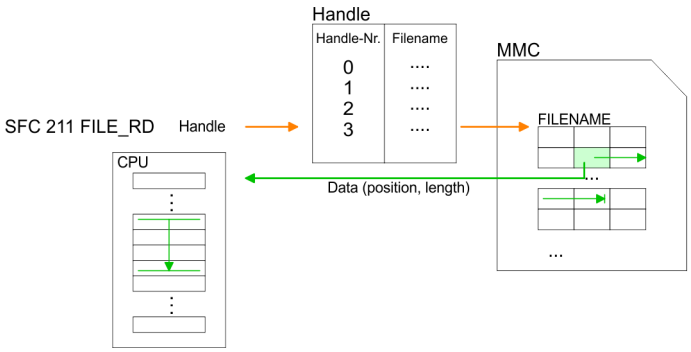
Codes, die RETVAL zurück liefert:

Code	Beschreibung
0000h	OK
7000h	*REQ* = 0, *BUSY* = 0 (nichts ist zu tun)
7001h	*REQ* = 1, 1. Aufruf
7002h	Baustein wird bearbeitet
8100h	Der angegebene *HANDLE* ist ungültig
9001h	Dem *HANDLE* ist kein Dateiname zugeordnet
9002h	Eine andere Funktion wurde über den *HANDLE* aufgerufen und ist fertig
9003h	Eine andere Funktion wurde über den *HANDLE* aufgerufen und ist nicht fertig
A000h	Systeminterner Fehler aufgetreten
A100h	Allgemeiner Filesystem-Fehler (z.B. keine Speicherkarte gesteckt)

5.3.6 FC/SFC 211 - FILE_RD - Datei lesen

Beschreibung

Hiermit können Sie ab einer ORIGIN-Position (Position der Schreib-/Lesemarke) von der Speicherkarte über den geöffneten Handle Daten in die CPU übertragen. Pro Aufruf können maximal 512Byte übertragen werden. Durch Angabe von *DATA* bestimmen Sie Speicherort und Länge des Schreib-Bereichs in Ihrer CPU. *REQ* = 1 löst die Funktion aus.



Parameter

Parameter	Deklaration	Datentyp	Beschreibung
REQ	IN	BOOL	Funktion aktivieren
HANDLE	IN	INT	Index der Datei 0 ... 3
DATA	IN	ANY	Zeiger auf Speicherort und Länge des Schreibbereichs
RETVAL	OUT	WORD	Rückgabewert (0 = OK)
BUSY	OUT	BOOL	Funktion wird bearbeitet

RETVAL (Rückgabewert) Codes, die RETVAL zurück liefert:

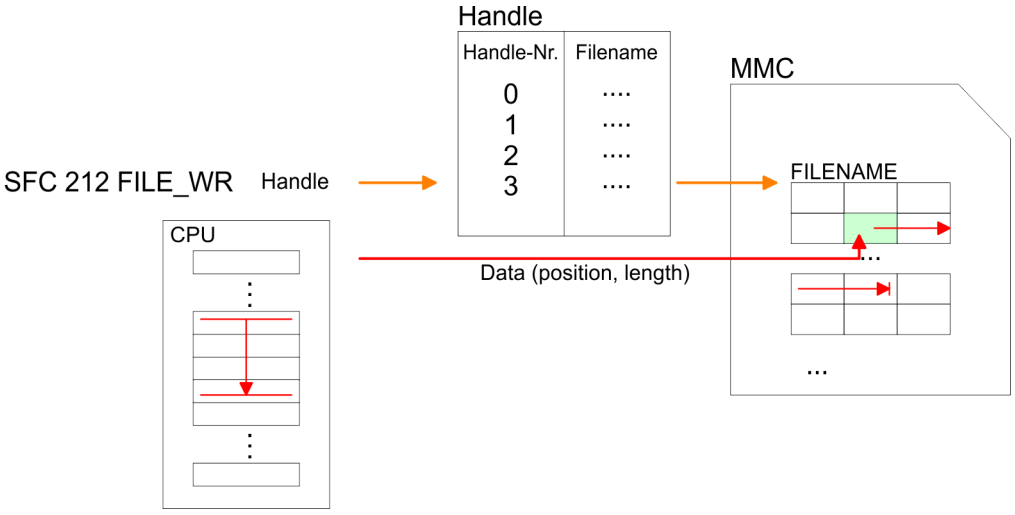
Code	Beschreibung
0xxxh	0 = OK, 0xxx = Länge gelesener Daten
7000h	*REQ* = 0, *BUSY* = 0 (nichts ist zu tun)
7001h	*REQ* = 1, 1. Aufruf
7002h	Baustein wird bearbeitet
8010h	Pointer in *DATA* ist vom Typ BOOL
8011h	Pointer in *DATA* kann nicht dekodiert werden (z.B. DB nicht geladen)
8012h	Datenlänge ist größer als 512Byte
8013h	Es wurde versucht auf einen schreibgeschützten DB zuzugreifen
8100h	Der angegebene *HANDLE* ist ungültig
9001h	Für diesen *HANDLE* ist keine Datei geöffnet
9002h	Eine andere Funktion wurde über den *HANDLE* aufgerufen und ist fertig
9003h	Eine andere Funktion wurde über den *HANDLE* aufgerufen und ist nicht fertig
A000h	Systeminterner Fehler aufgetreten

Code	Beschreibung
A003h	interner Fehler
A100h	Allgemeiner Filesystem-Fehler (z.B. keine Speicherkarte gesteckt)

5.3.7 FC/SFC 212 - FILE_WR - Datei schreiben

Beschreibung

Für Schreibzugriffe auf die Speicherkarte ist dieser Baustein zu verwenden. Hierbei werden Daten von der unter *DATA* angegebenen Position und Länge in der CPU über den entsprechenden *HANDLE* ab der Schreib-/Lese-Position auf die Speicherkarte geschrieben. Pro Aufruf können maximal 512Byte übertragen werden. *REQ* = 1 löst die Funktion aus.



Parameter

Parameter	Deklaration	Datentyp	Beschreibung
REQ	IN	BOOL	Funktion aktivieren
HANDLE	IN	INT	Index der Datei 0 ... 3
DATA	IN	ANY	Zeiger auf Speicherort und Länge des Schreibbereichs
RETVAL	OUT	WORD	Rückgabewert
BUSY	OUT	BOOL	Funktion wird bearbeitet

Der Parameter *RETVAL* liefert die Länge der geschriebenen Daten zurück. Der Baustein liefert keine Fehlermeldung, wenn die Speicherkarte voll ist. Der Anwender muss überprüfen, dass die Anzahl der geforderten zu schreibenden Bytes der in *RETVAL* zurück gelieferten geschriebenen Bytes entspricht.

RETVAL (Rückgabewert)

Codes, die RETVAL zurück liefert:

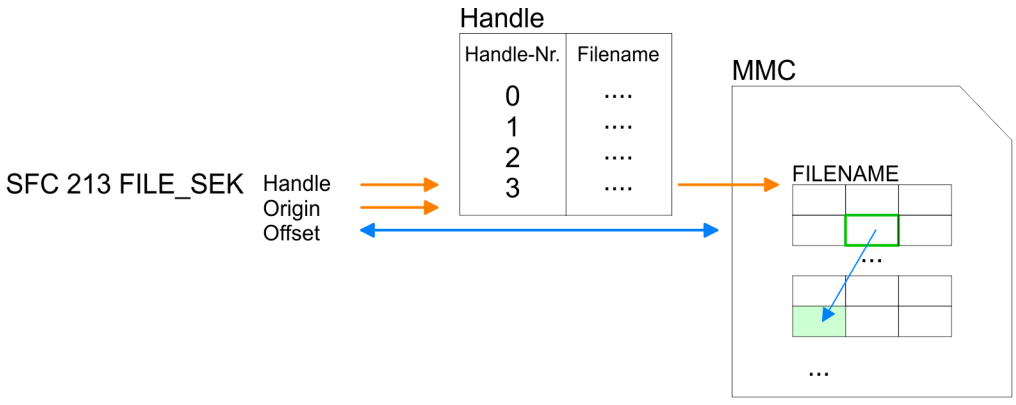
Code	Beschreibung
0xxxh	0 = OK, 0xxx = Länge geschriebener Daten
7000h	*REQ* = 0, *BUSY* = 0 (nichts ist zu tun)
7001h	*REQ* = 1, 1. Aufruf

Code	Beschreibung
7002h	Baustein wird bearbeitet
8010h	Pointer in *DATA* ist vom Typ BOOL
8011h	Pointer in *DATA* kann nicht dekodiert werden (z.B. DB nicht geladen)
8012h	Datenlänge ist größer als 512Byte
8100h	Der angegebene *HANDLE* ist ungültig
9001h	Für diesen *HANDLE* ist keine Datei geöffnet
9002h	Eine andere Funktion wurde über den *HANDLE* aufgerufen und ist fertig
9003h	Eine andere Funktion wurde über den *HANDLE* aufgerufen und ist nicht fertig
A000h	Systeminterner Fehler aufgetreten
A002h	Die Datei ist schreibgeschützt
A003h	Interner Fehler
A004h	Speicherkarte ist schreibgeschützt
A100h	Allgemeiner Filesystem-Fehler (z.B. keine Speicherkarte gesteckt)

5.3.8 FC/SFC 213 - FILE_SEK - Position Schreib-/Lesemarke

Beschreibung

Mit FILE_SEK können Sie die Position der Schreib-/Lesemarke für den entsprechenden *HANDLE* ändern bzw. ermitteln. Durch Angabe von *ORIGIN* als Startposition und einem *OFFSET* können Sie für den entsprechenden *HANDLE* die Schreib-/Lesemarke platzieren. *REQ* = 1 startet Funktion.



Parameter

Parameter	Deklaration	Datentyp	Beschreibung
REQ	IN	BOOL	Funktion aktivieren
HANDLE	IN	INT	Index der Datei 0 ... 3
ORIGIN	IN	INT	0 = Datei-Anfang, 1 = aktuelle Position, 2 = Dateiende
RETVAL	OUT	WORD	Rückgabewert (0 = OK)
BUSY	OUT	BOOL	Funktion wird bearbeitet
OFFSET	INOUT	DINT	Offset Schreib-/Lesemarke

RETVAL (Rückgabewert) Codes, die *RETVAL* zurück liefert:

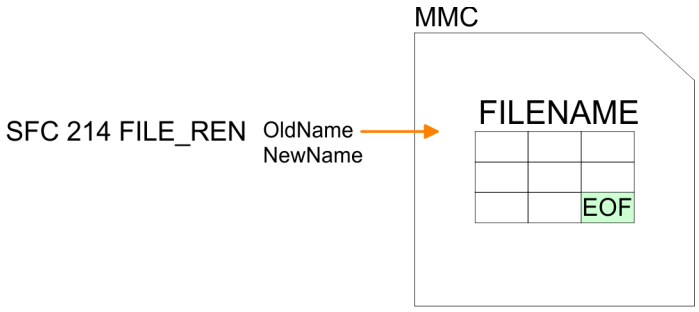
Code	Beschreibung
0000h	OK, OFFSET beinhaltet die Aktuelle Schreib-/Lese-Position
7000h	*REQ* = 0, *BUSY* = 0 (nichts ist zu tun)
7001h	*REQ* = 1, 1. Aufruf
7002h	Baustein wird bearbeitet
8100h	Der angegebene HANDLE ist ungültig
9001h	Für diesen HANDLE ist keine Datei geöffnet
9002h	Eine andere Funktion wurde über den *HANDLE* aufgerufen und ist fertig
9003h	Eine andere Funktion wurde über den *HANDLE* aufgerufen und ist nicht fertig
A000h	Systeminterner Fehler aufgetreten
A004h	*ORIGIN*-Parameter ist fehlerhaft
A100h	Allgemeiner Filesystem-Fehler (z.B. keine Speicherkarte gesteckt)

5.3.9 FC/SFC 214 - FILE_REN - Datei umbenennen

Beschreibung

Unter Einsatz von FILE_REN können Sie den unter *OLDNAME* angegebenen Dateinamen ändern in *NEWNAME*.

VORSICHT!
Bitte beachten Sie, dass Sie nur Dateien umbenennen dürfen, die zuvor mit File_CLO geschlossen wurden. Ansonsten könnte dies zu Datenverlust auf der Speicherkarte führen!



Parameter

Parameter	Deklaration	Datentyp	Beschreibung
REQ	IN	BOOL	Funktion aktivieren
MEDIA	IN	INT	0 = MMC
OLDNAME	IN	STRING[254]	Alter Dateiname (muss im 8.3-Format sein)
NEWNAME	IN	STRING[254]	Neuer Dateiname (muss im 8.3-Format sein)

Parameter	Deklaration	Datentyp	Beschreibung
RETVAL	OUT	WORD	Rückgabewert (0 = OK)
BUSY	OUT	BOOL	Funktion wird bearbeitet

RETVAL (Rückgabewert) Codes, die RETVAL zurück liefert:

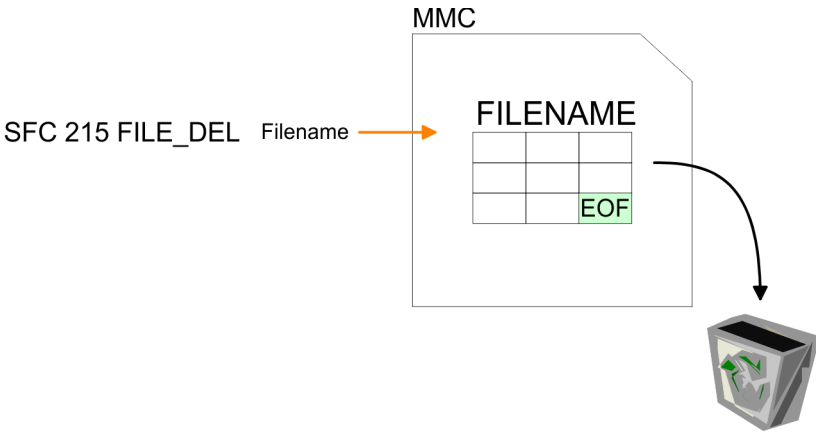
Code	Beschreibung
0000h	OK, Datei wurde umbenannt
7000h	*REQ* = 0, *BUSY* = 0 (nichts ist zu tun)
7001h	*REQ* = 1, 1. Aufruf
7002h	Baustein wird bearbeitet
8010h	Parameter *OLDNAME* ist nicht verfügbar (z.B. DB nicht geladen)
8011h	*OLDNAME* fehlerhaft (entspricht nicht dem 8.3 Format oder Sonderzeichen)
8020h	Parameter *NEWNAME* ist nicht verfügbar (z.B. DB nicht geladen)
8021h	*NEWNAME* fehlerhaft (entspricht nicht dem 8.3 Format oder Sonderzeichen)
A000h	Systeminterner Fehler aufgetreten
A001h	Der angegebene MEDIA-Typ ist falsch
A003h	Der neue Dateiname *NEWNAME* existiert schon
A004h	Datei *OLDNAME* existiert nicht
A006h	Datei *OLDNAME* ist geöffnet
A007h	Speicherkarte schreibgeschützt
A100h	Das Filesystem liefert einen Fehler beim Anlegen der Datei (z.B. keine Speicherkarte gesteckt)

5.3.10 FC/SFC 215 - FILE_DEL - Datei löschen

Beschreibung

Mit diesem Baustein können Sie eine Datei auf der Speicherkarte löschen. Geben Sie hierzu unter *FILENAME* den Namen der zu löschenden Datei an.

VORSICHT!
Bitte beachten Sie, dass Sie nur Dateien löschen dürfen, die zuvor mit File_CLO geschlossen wurden. Ansonsten könnte dies zu Datenverlust auf der Speicherkarte führen!



Parameter

Parameter	Deklaration	Datentyp	Beschreibung
REQ	IN	BOOL	Funktion aktivieren
MEDIA	IN	INT	0 = MMC
FILENAME	IN	STRING[254]	Dateiname (muss im 8.3-Format sein)
RETVAL	OUT	WORD	Rückgabewert (0 = OK)
BUSY	OUT	BOOL	Funktion wird bearbeitet.

RETVAL (Rückgabewert) Codes, die *RETVAL* zurück liefert:

Code	Beschreibung
0000h	OK, Datei wurde gelöscht
7000h	*REQ* = 0, *BUSY* = 0 (nichts ist zu tun)
7001h	*REQ* = 1, 1. Aufruf
7002h	Baustein wird bearbeitet
8010h	Parameter *FILENAME* ist nicht verfügbar (z.B. DB nicht geladen)
8011h	*FILENAME* ist fehlerhaft (z.B. entspricht nicht dem 8.3 Format oder Sonderzeichen)
A000h	Systeminterner Fehler aufgetreten
A001h	Der angegebene *MEDIA*-Typ ist falsch
A002h	Die Datei ist schreibgeschützt
A004h	Datei *FILENAME* existiert nicht
A005h	*FILENAME* ist ein Verzeichnis - nicht löschbar
A006h	Datei ist geöffnet
A007h	Speicherkarte schreibgeschützt.
A100h	Allgemeiner Filesystem-Fehler (z.B. keine Speicherkarte gesteckt)

5.4 Systemfunktionen - *"System Functions"*

5.4.1 SFC 75 - SET_ADDR - PROFIBUS MAC-Adresse setzen

Beschreibung

Mit diesem SFC können Sie die MAC-Adresse der integrierten PROFIBUS-Schnittstelle einer CPU ändern. Die Funktion ist nur in der Betriebart passiver DP-Slave möglich. Zur Identifikation dient die Diagnoseadresse. Der SFC arbeitet asynchron und kann nur auf eine Schnittstelle angewendet werden. Bei STOP und anschließendem Warmstart bleibt die eingestellte Netzadresse erhalten. Bei PowerOFF-PowerON und bei Urlöschen erhält die Schnittstelle wieder ihre projektierte Teilnehmernummer. Der DP-Slave nimmt konsequent die Identität des DP-Slaves mit der neuen Adresse an. Gegenüber dem DP-Master fällt der DP-Slave mit der alten Adresse aus und ein DP-Slave mit der neuen Adresse kehrt wieder. Wird eine Adresse gewählt, die schon ein anderer Teilnehmer am DP-Strang besitzt, fallen beide Slaves bezogen auf die DP-Kommunikation aus.

Parameter

Parameter	Deklaration	Datentyp	Speicherbereich	Beschreibung
REQ	INPUT	BOOL	E, A, M, D, L	Funktionsanstoß mit *REQ* = 1
LADDR	INPUT	WORD	E, A, M, D, L	Identifikation der Schnittstelle
ADDR	INPUT	BYTE	E, A, M, D, L	Neue Teilnehmeradresse
RET_VAL	OUTPUT	INT	E, A, M, D, L	Fehlercode
BUSY	OUTPUT	BOOL	E, A, M, D, L	*BUSY* = 1: In Bearbeitung

RET_VAL (Rückgabewert)

Wert	Beschreibung
0000h	Der Auftrag wurde fehlerfrei durchgeführt
7000h	Aufruf mit *REQ* = 0 (Aufruf ohne Bearbeitung), *BUSY* hat den Wert 0, es ist keine Datenübertragung aktiv
7001h	Erstaufruf mit *REQ* = 1: Datenübertragung angestoßen; *BUSY* hat den Wert 1
7002h	Zwischenaufruf (*REQ* irrelevant): Datenübertragung bereits aktiv; *BUSY* hat den Wert 1
8xyyh	Allgemeine Fehlerinformation ↳ *Kapitel 4.1 "Allgemeine und spezifische Fehlercodes RET_VAL" auf Seite 8*
8090h	Identifikation der Schnittstellen: Logische Adresse ist ungültig
8091h	Neue Teilnehmeradresse ist ungültig
8093h	Identifikation der Schnittstellen: Logische Adresse ist keine Schnittstelle
809Bh	Funktion nicht durchführbar (z.B. Schnittstelle ist kein DP-Slave oder aktiv)
80C3h	Ressourcenmangel (z.B. Mehrfachaufruf des SFC)

5.4.2 FC/SFC 193 - AI_OSZI - Oszilloskop-/FIFO-Funktion

Beschreibung

Der FC/SFC 193 dient der Ansteuerung der Oszilloskop-/FIFO-Funktion von analogen Eingabe-Kanälen, welche diese Funktionalität besitzen. Er ermöglicht das Starten der Aufzeichnung und das Auslesen der aufgezeichneten Daten. Je nach Parametrierung ergeben sich folgende Möglichkeiten:

Oszilloskop-Betrieb

- Je nach Trigger-Bedingung können Sie bei Flankenauswertung die Überwachung des eingestellten Kanals starten bzw. im manuellen Betrieb die Aufzeichnung starten.
- Sobald der Speicher voll ist haben Sie mit dem FC/SFC 193 Zugriff auf die aufgezeichneten Messwerte.

FIFO-Betrieb

- Die Aufzeichnung starten.
- Jederzeit den Puffer lesen.

Hinweis!

Der Aufruf des FC/SFC darf nur aus einer Prioritätsebene erfolgen, zum Beispiel nur aus OB 1 oder nur aus OB 35.

Das Modul muss zuvor parametriert werden.

Zum Starten und zum Auslesen ist jeweils ein Aufruf des FC/SFC 193 erforderlich. Die Unterscheidung der beiden Aufruf-Varianten erfolgt im Parameter MODE.

Parameter

Parameter	Deklaration	Datentyp	Funktion in Abhängig von MODE
REQ	IN	BOOL	Funktion ausführen (Starten/Auslesen)
LADR	IN	WORD	Basisadresse des Moduls
MODE	IN	WORD	Modus (Starten/Auslesen)
CHANNEL	IN	BYTE	Kanal, der ausgelesen werden soll
OFFSET	IN	DWORD	Adress-Offset beim Auslesen (nicht im FIFO-Betrieb)
RECORD	IN	ANY	Bereich für die ausgelesenen Daten
RETVAL	OUT	WORD	Rückgabewert (0 = OK)
BUSY	OUT	BOOL	Funktion wird bearbeitet
TIMESTAMP	OUT	DWORD	Zeitstempel (nur bei Flankenauswertung)
LEN	INOUT	DWORD	Anzahl der Werte, die pro Kanal zu bearbeiten sind

REQ

- Abhängig vom eingestellten *MODE* lässt sich durch Setzen dieses Bits die Aufzeichnung starten bzw. das Auslesen beginnen.
- Je nach Trigger-Bedingung wird bei Flankenauswertung die Überwachung des eingestellten Kanals oder im manuellen Betrieb die Aufzeichnung gestartet.
- Ist unter *MODE* der Befehl "Auslesen" eingestellt, werden die Daten aus dem Modul gelesen.

LADR

Logische Basisadresse des Moduls.

MODE

Den FC/SFC 193 können Sie in 3 verschiedenen Modi aufrufen. Den entsprechenden Modus geben Sie über *MODE* vor. Durch Setzen von *REQ* wird der entsprechende Modus ausgeführt. Folgende Werte werden unterstützt:

- 01h: Je nach Parametrierung Aufzeichnung starten bzw. Flankenüberwachung starten
- 00h: Daten über mehrere Zyklen lesen bis *BUSY* = 0 erfolgt.
- 80h: Daten in einem Zugriff lesen

CHANNEL

Hier wird der Kanal angegeben, der ausgelesen werden soll. Mit jedem Aufruf kann nur jeweils ein Kanal ausgelesen werden. Für Start-Aufrufe mit *MODE* = 01h ist dieser Parameter irrelevant.

OFFSET

- Der Adress-Offset gibt einen Offset-Wert der Adresse beim Auslesen an. Dies ermöglicht den Zugriff auf Teilbereiche der aufgezeichneten Daten.
- Der Wert für den maximale Offset-Wert hängt von der Anzahl der pro Kanal aufgezeichneten Werte ab.
- Im FIFO-Betrieb wird *OFFSET* nicht unterstützt und deshalb dieser Parameter ignoriert.

RECORD

- Hier können Sie einen Bereich definieren, in dem die gelesenen Werte zu speichern sind.
- Im FIFO-Betrieb werden hier alle Werte des eingestellten Kanals ausgelesen, die zum Zeitpunkt des Auslesens aufgezeichnet wurden.
- Bitte tragen Sie hierfür Sorge, dass der Puffer eine ausreichende Größe zur Aufnahme der Daten besitzt, ansonsten erhalten Sie eine Fehlermeldung.

BUSY

- *BUSY* = 1 zeigt an, dass die entsprechende Funktion bearbeitet wird.
- Mit *BUSY* = 0 ist die Bearbeitung der Funktion abgeschlossen.

TIMESTAMP

- In jedem SPEED-Bus-Modul läuft eine interne Uhr mit der Auflösung von 1µs mit.
- Der Rückgabewert entspricht der Uhrzeit auf dem SPEED-Bus- Modul, bei der das Trigger-Ereignis eingetreten ist.
- *TIMESTAMP* ist ausschließlich im flankengesteuerten Oszilloskop-Betrieb verfügbar.
- Er ist gültig solange der Auftrag läuft (*RETVAL* = 7xxxh) bzw. wenn dieser ohne Fehler beendet wurde (*RETVAL* = 0000h).

LEN

Der als IN/OUT realisierte Längenparameter wird beim Funktionsaufruf in den unterschiedlichen Modi verschieden interpretiert.

Modus: starten (*MODE*: = 01h)

Unter *MODE* = 01h kommt dieser Parameter ausschließlich bei manuellem Oszilloskop-Start zum Einsatz. Hier übergeben Sie die gewünschte Anzahl der Werte, die pro Kanal aufzuzeichnen sind. In diesem Modus liefert *LEN* keinen Wert zurück.

Modus: auslesen (*MODE*: = 00h oder 80h)

Bei *MODE* = 00h bzw. 80h geben Sie hier die Anzahl der Werte an, die auszulesen sind. Im FIFO-Betrieb wird dieser Parameter beim Aufruf nicht berücksichtigt. *LEN* liefert als Rückgabewert die Anzahl der Werte, die ausgelesen wurden.

RETVAL (Rückgabewert)

Zusätzlich zu den hier aufgeführten modulspezifischen Fehlercodes sind auch noch die allgemeingültigen Fehlercodes für FC/SFCs als Rückgabewert möglich.

RETVAL	Beschreibung in Abhängigkeit vom *BUSY*-Bit	BUSY
Byte		
0	Bit 1, 0:	
	00: Aufruf mit Request: = 0 (Leerlauf, warte auf *REQ* = 1).	0
	01: Erstaufruf mit *REQ*: = 1	1
	10: Folgeaufruf mit *REQ*: = 1	1
	11: Oszilloskop zeichnet gerade auf.	1
	Bit 2: Request: = 1, aber Aufzeichnung wurde noch nicht gestartet (*MODE*: = 00h oder *MODE*: = 80h)	0
	Bit 3: reserviert	-
	Bit 4: Trigger-Ereignis eingetreten und Aufzeichnung läuft	1
	Bit 5: Warte auf Trigger-Ereignis	1
	Bit 7…6: reserviert	-
1	Bit 0: reserviert	-
	Bit 1: Die Anzahl der aufgezeichneten Werte ist größer als die Länge des durch *RECORD* aufgespannten Zielbereichs (in Worten).	0
	Bit 2: Die Anzahl der aufgezeichneten Werte ist größer als die übergebene Länge am Parameter *LEN* und dem *OFFSET*.	0
	Bit 3: Im FIFO-Betrieb ist der Puffer übergelaufen.	0
	Bit 7...4:	
	0000: Auftrag beendet ohne Fehler	0
	0111: Auftrag läuft	1
	1000: Auftrag beendet mit Fehler	0

Auftrag beendet ohne Fehler

RETVAL	Beschreibung in Abhängigkeit vom *BUSY*-Bit	BUSY
0000h	Auftrag wurde ohne Fehler durchgeführt	0

Auftrag beendet mit Fehler

RETVAL	Beschreibung in Abhängigkeit vom *BUSY*-Bit	BUSY
8002h:	Die Oszilloskop-/FIFO-Funktion ist nicht projektiert.	0
8003h:	Es ist ein interner Fehler aufgetreten - kontaktieren Sie VIPA.	0
8005h:	Der angegebene Kanal kann nicht ausgelesen werden - falsche Kanal-Nummer.	0
8007h:	Der Wert unter *OFFSET* ist größer als die Anzahl der aufgezeichneten Werte.	0
8090h:	Es ist kein SPEED-Bus-Modul unter dieser Adresse verfügbar.	0
80D2h:	*LADR* liegt außerhalb des Peripherieadressbereichs.	0

5.4.3 FC/SFC 194 - DP_EXCH - Datenaustausch mit CP342S

Beschreibung

Mit dem FC/SFC 194 können Sie Daten zwischen Ihrer CPU und einem über SPEED-Bus angebunden PROFIBUS-DP-Master austauschen. Normalerweise blendet jeder PROFIBUS-DP-Master seinen E/A-Bereich im Peripherie-Bereich der CPU ein. Hierbei können Sie über die Hardware- Konfiguration einen Peripherie-Bereich von 0 ... 2047 adressieren. Da dies die maximale Anzahl an PROFIBUS-DP-Master-Modulen am SPEED-Bus einschränkt, haben Sie die Möglichkeit das Mapping an dem entsprechenden DP-Master zu deaktivieren und statt dessen den Zugriff über Hantierungsbaustein zu aktivieren. Hierbei können Sie mit dem FC/SFC 194 Daten von der CPU in einen definierten Bereich des DP-Master schreiben und Daten aus einem definierten Bereich des DP Master lesen.

Parameter

Parameter	Deklaration	Datentyp	Funktion in Abhängig von MODE
LADR	IN	WORD	Basisadresse des DP-Master-Moduls am SPEED-Bus
MODE	IN	WORD	Modus (0 = lesen / 1 = schreiben)
LEN	IN	WORD	Länge des Datenbereichs im DP-Master
OFFSET	IN	DWORD	Beginn des Datenbereichs im DP-Master
RETVAL	OUT	WORD	Rückgabewert (0 = OK)
DATA	IN OUT	ANY	Zeiger auf Datenbereich in der CPU

LADR

Logische Basisadresse des Moduls.

MODE

Den FC/SFC 194 können Sie mit folgenden Modi aufrufen:

- 0000 = Daten transferieren von DP-Master in die CPU.
- 0001 = Daten transferieren von der CPU in den DP-Master.

LEN

Hier definieren Sie die Länge des Datenbereichs im DP-Master.

OFFSET

Definieren Sie hier den Beginn des Datenbereichs im DP-Master. Bitte beachten Sie, dass der über *OFFSET* und *LEN* definierte Bereich den über die Hardware-Konfiguration parametrierten Bereich im DP-Master nicht überschreitet.

RETVAL (Rückgabewert)

Zusätzlich zu den hier aufgeführten modulspezifischen Fehlercodes sind auch noch die allgemeingültigen Fehlercodes für FC/SFCs als Rückgabewert möglich. ↳ *Kapitel 4.1 "Allgemeine und spezifische Fehlercodes RET_VAL" auf Seite 8*

RETVAL	Beschreibung
0000h	Kein Fehler
8001h	*LADR* konnte keinem DP-Master am SPEED-Bus zugeordnet werden.
8002h	Wert des Parameters *MODE* ist außerhalb der Grenzen.
8003h	Wert des Parameters *LEN* ist 0.
8004h	Wert des Parameters *LEN* ist größer als der unter DATA definierte Datenbereich.

RETVAL	Beschreibung
8005h	Der über *OFFSET* und *LEN* definierte Bereich liegt außerhalb 0 …2047.
8006h	Der über *LADR* definierte DP-Master ist nicht für den Zugriff über Hantierungsbaustein parametriert. Aktivieren Sie in den Eigenschaften des DP-Master "IO-Mode HTB".
8008h	Lücke(n) im Eingangsbereich vorhanden.
8009h	Lücke(n) im Ausgangsbereich vorhanden.
8010h	Fehler beim Zugriff auf Eingabebereich (z.B. DP-Master ist nicht erreichbar)
8011h	Fehler beim Zugriff auf Ausgabebereich (z.B. DP-Master ist nicht erreichbar)
8Fxxh	DATA fehlerhaft (xx) ↳ *Kapitel 4.1 "Allgemeine und spezifische Fehlercodes RET_VAL" auf Seite 8*

5.4.4 FC/SFC 219 - CAN_TLGR - CANopen-Kommunikation

FC/SFC 219 CAN_TLGR SDO-Anforderung an CAN-Master

Jede SPEED7-CPU hat den FC/SFC 219 integriert. Hiermit können Sie von Ihrem SPS-Programm auf Ihrem CAN-Master einen SDO- Lese- oder Schreibzugriff auslösen. Hierbei adressieren Sie den Master über die Steckplatz-Nr. und den Ziel- Slave über seine CAN-Adresse. Die Prozessdaten bestimmen Sie durch Angabe von *INDEX* und *SUBINDEX*. Über SDO kann pro Zugriff maximal ein Datenwort Prozessdaten übertragen werden.

Parameter

Parameter	Deklaration	Datentyp	Beschreibung
REQUEST	IN	BOOL	Funktion aktivieren
SLOT_MASTER	IN	BYTE	SPEED-Bus Steckplatz (101 ... 116)
NODEID	IN	BYTE	CAN-Adresse (1 ... 127)
TRANSFERTYP	IN	BYTE	Transfertyp
INDEX	IN	DWORD	CANopen Index
SUBINDEX	IN	DWORD	CANopen Subindex
CANOPENERROR	OUT	DWORD	CANopen Fehler
RETVAL	OUT	WORD	Rückgabewert (0 = OK)
BUSY	OUT	BOOL	Funktion wird bearbeitet
DATABUFFER	INOUT	ANY	Datenpuffer für FC/SFC-Kommunikation

REQUEST Steuerparameter: 1: Anstoß des Auftrags

SLOT_MASTER 101...116: Steckplatz 1 ... 16 von Master auf SPEED-Bus

NODELD Adresse des CANopen Knotens (1...127)

TRANSFERTYPE

40h: Lesen SDO	23h: Schreiben SDO (1 DWORD)
	2Bh: Schreiben SDO (1 WORD)
	2Fh: Schreiben SDO (1 BYTE)

INDEX

CANopen Index

SUBINDEX

CANopen Subindex

SLOT_MASTER

0:	System 200 CPU 21xCAN
1...32:	System 200 IM 208CAN
101...115:	System 300S 342-1CA70

CANOPENERROR

Liegt kein Fehler vor, so liefert *CANOPENERROR* eine 0 zurück. Im Fehlerfall beinhaltet *CANOPENERROR* eine der nachfolgend aufgeführten Fehlermeldungen, die vom CAN-Master generiert wird:

Code	Beschreibung
0503 0000h	Toggle-Bit nicht geändert
0504 0000h	SDO Protokoll Time-out
0504 0001h	Client/Server Befehlsspezifizierung nicht gültig oder unbekannt
0504 0002h	Ungültige Blockgröße (nur Block-Modus)
0504 0003h	Ungültige Sequenznummer (nur Block-Modus)
0504 0004h	CRC Fehler (nur Block-Modus)
0504 0005h	Unzureichender Speicher
0601 0000h	Lesezugriff auf ein Nur-Schreiben-Objekt
0601 0001h	Schreibzugriff auf ein Nur-Lesen-Objekt
0602 0000h	Objekt nicht im Objektverzeichnis vorhanden
0604 0041h	Objekt kann nicht ins PDO gemappt werden
0604 0042h	Anzahl und Länge der zu mappenden Objekte überschreitet PDO-Länge
0604 0043h	Generelle Parameterinkompatibilität
0604 0047h	Generelle interne Inkompatibilität im Gerät
0606 0000h	Zugriffsfehler wegen Hardwareausfall
0607 0010h	Datentyp nicht korrekt, Länge der Serviceparameter nicht korrekt
0607 0012h	Datentyp nicht korrekt, Serviceparameter zu lang
0607 0013h	Datentyp nicht korrekt, Serviceparameter zu kurz
0609 0011h	Subindex existiert nicht
0609 0030h	Wertebereich der Parameter überschritten (nur für Schreibzugriff)
0609 0031h	Zu schreibender Parameterwert ist zu hoch

Code	Beschreibung
0609 0032h	Zu schreibender Parameterwert ist zu niedrig
0609 0036h	Maximumwert ist kleiner als Minimumwert
0800 0000h	Genereller Fehler
0800 0020h	Die Daten können entweder nicht transferiert oder nicht in der SPS gespeichert werden.
0800 0021h	Die Daten können wegen lokaler Kontrollen entweder nicht transferiert oder nicht in der SPS gespeichert werden.
0800 0022h	Die Daten können wegen aktuellem Modulstatus entweder nicht transferiert oder nicht in der SPS gespeichert werden.
0800 0023h	Dynamische Objektverzeichnisgenerierung fehlgeschlagen oder kein Objektverzeichnis gefunden (z.B. Objektverzeichnis wird aus Datei generiert und ein Dateifehler ist aufgetreten).

RETVAL

Wird die Funktion fehlerfrei ausgeführt, enthält der Rückgabewert die gültige Länge der Antwortdaten: 1: Byte, 2: Wort, 4: Doppelwort Tritt während der Bearbeitung der Funktion ein Fehler auf, enthält der Rückgabewert einen der nachfolgend aufgeführten Fehler-codes.

Code	Beschreibung
F021h	Ungültige Slave-Adresse (Aufrufparameter gleich 0 oder größer 127)
F022h	Ungültiger Transfertyp (Wert ungleich 40h, 23h, 2Bh, 2Fh)
F023h	Ungültige Datenlänge (der Datenpuffer ist zu klein, beim SDO-Lesezugriff sollte dieser mindestens 4Byte groß sein, beim SDO-Schreibzugriff sollte dieser 1Byte, 2Byte oder 4Byte groß sein)
F024h	Der FC/SFC wird nicht unterstützt.
F025h	Schreibpuffer im CANopen-Master ist voll, Service kann zur Zeit nicht bearbeitet werden.
F026h	Lesepuffer im CANopen-Master ist voll, Service kann zur Zeit nicht bearbeitet werden.
F027h	Der SDO-Lese- oder Schreibzugriff wurde fehlerhaft beantwortet ↳ *"CANOPENERROR" auf Seite 57*.
F028h	SDO-Timeout (es wurde kein CANopen-Teilnehmer mit der Node-ID gefunden).

BUSY

Solange *BUSY* = 1 ist der aktuelle Auftrag ist noch nicht beendet.

DATABUFFER

- Datenbereich, über den der FC/SFC kommuniziert. Geben Sie hier einen ANYPointer vom Typ Byte an.
- SDO-Lesezugriff: Zielbereich für die gelesenen Nutzdaten.
- SDO-Schreibzugriff: Quellbereich für die zu schreibenden Nutzdaten.

Sofern eine SDO-Anforderung fehlerfrei abgearbeitet wurde, enthält RETVAL die Länge der gültigen Antwortdaten in (1, 2 oder 4Byte) und CANOPENERROR den Wert 0.

5.4.5 FC/SFC 254 - RW_SBUS - IBS-Kommunikation

Beschreibung

Dieser Baustein dient den INTERBUS-FCs 20x als Kommunikationsbaustein zwischen INTERBUS-Master und CPU. Für den Einsatz der INTERBUS-FCs 20x ist der FC/SFC 254 als Baustein in Ihr Projekt einzubinden.

Parameter

Parameter	Deklaration	Datentyp	Beschreibung
READ/WRITE	IN	Byte	0 = Lesen, 1 = Schreiben
LADDR	IN	WORD	Logical Adresse INTERBUS-Master
IBS_ADDR	IN	WORD	Adresse INTERBUS-Master
DATAPOINTER	IN	ANY	Zeiger auf Datenbereich in der CPU
RETVAL	OUT	WORD	Rückgabewert (0 = OK)

READ/WRITE

Hiermit bestimmen Sie die Transferrichtung aus CPU-Sicht. Mit *READ* lesen Sie Daten aus dem Dual-port-memory des INTERBUS-Master.

LADDR

Geben Sie hier die Adresse (**L**ogical **Addr**ess) an, ab der das Register des Masters in der CPU eingeblendet wird. Beim Hochlauf der CPU werden, sofern keine Hardware-Konfiguration vorliegt, die INTERBUS-Master nach folgender Formel im E/A-Adress-Bereich der CPU abgelegt:

Anfangsadresse = 256 = (Steckplatz-101)+2048

Die Steckplatz-Nummerierung am SPEED-Bus beginnt bei 101 links der CPU und geht von rechts nach links. Beispielsweise hat der 1. Steckplatz die Adresse 2048, der 2. den Steckplatz 2304 usw.

IBS_ADDR

Adresse im Adressraum des INTERBUS-Master.

DATAPOINTER

Zeiger auf Datenbereich in der CPU.

RETVAL

Wert, den die Funktion zurück liefert. Bei 0 ist alles OK.

5.5 Systemfunktions-Blöcke - *"System Function Blocks"*

5.5.1 SFB 7 - TIMEMESS - Zeitmessung

Im Gegensatz zum FC/SFC 53 liefert der SFB 7 die Differenz zwischen zwei Aufrufen in µs zurück. Mit *RESET* = 1 wird der aktuelle µs Zählerstand im InstDB gespeichert. Ein erneuter Aufruf mit *RESET* = 0 liefert über *VALUE* den Differenzwert zum ersten Aufruf in µs.

Parameter

Parameter	Deklaration	Datentyp	Beschreibung
RESET	IN	BOOL	*RESET* = 1 startet Zähler
VALUE	OUT	DWORD	Differenz in µs

RESET

Mit *RESET* = 1 wird der aktuelle Zählerstand im InstDB gespeichert. Der Wert in *VALUE* wird hierbei nicht beeinflusst.

VALUE

Nach einem Aufruf mit *RESET* = 0 liefert *VALUE* die zeitliche Differenz zwischen den zwei SFB 7 Aufrufen zurück.