

VIPA

Standard | Operationsliste | Handbuch

HB00D_OPL_STD | Rev. 14/22

Mai 2014

Copyright © VIPA GmbH. All Rights Reserved.

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

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

Zur Genehmigung von Vervielfältigung oder Verteilung wenden Sie sich bitte an:

VIPA, Gesellschaft für Visualisierung und Prozessautomatisierung mbH

Ohmstraße 4, D-91074 Herzogenaurach, Germany

Tel.: +49 (91 32) 744 -0

Fax.: +49 9132 744 1864

E-Mail: info@vipa.de

<http://www.vipa.com>

Hinweis

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

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

EG-Konformitätserklärung

Hiermit erklärt VIPA GmbH, dass die Produkte und Systeme mit den grundlegenden Anforderungen und den anderen relevanten Vorschriften übereinstimmen.

Die Übereinstimmung ist durch CE-Zeichen gekennzeichnet.

Informationen zur Konformitätserklärung

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

Warenzeichen

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

SPEED7 ist ein eingetragenes Warenzeichen der profichip GmbH.

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

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

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

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

Dokument-Support

Wenden Sie sich an Ihre Landesvertretung der VIPA GmbH, wenn Sie Fehler anzeigen oder inhaltliche Fragen zu diesem Dokument stellen möchten. Ist eine solche Stelle nicht erreichbar, können Sie VIPA über folgenden Kontakt erreichen:

VIPA GmbH, Ohmstraße 4, 91074 Herzogenaurach, Germany

Telefax: +49 9132 744 1204

E-Mail: documentation@vipa.de

Technischer Support

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

VIPA GmbH, Ohmstraße 4, 91074 Herzogenaurach, Germany

Telefon: +49 9132 744 1150 (Hotline)

E-Mail: support@vipa.de

Inhaltsverzeichnis

Über dieses Handbuch	1
Teil 1 Befehlsliste	1-1
Alphabetisches Verzeichnis der Operationen	1-2
Abkürzungen	1-4
Register.....	1-6
Adressierungsbeispiele	1-7
Arithmetische Operationen	1-9
Baustein-Operationen	1-11
Bild- und Null-Operationen	1-12
Flanken-Operationen.....	1-12
Lade-Operationen	1-13
Schiebe-Operationen	1-16
Setzen/Rücksetzen von Bitoperanden.....	1-17
Sprung-Operationen.....	1-19
Transfer-Operationen	1-20
Umwandlungs-Operationen.....	1-23
Vergleichs-Operationen.....	1-24
Verknüpfungs-Operationen (Bit).....	1-25
Verknüpfungs-Operationen (Wort)	1-31
Zeit-Operationen	1-31
Zähl-Operationen	1-32
Teil 2 Organisationsbausteine	2-1
Übersicht.....	2-2
OB 1 - Zyklische Programmbearbeitung.....	2-3
OB 10 - Uhrzeitalarm.....	2-5
OB 20 - Verzögerungsalarm.....	2-7
OB 35 - Weckalarm.....	2-8
OB 40 - Prozessalarm	2-9
OB 80 - Zeitfehler.....	2-11
OB 81 - Stromversorgungsfehler.....	2-14
OB 82 - Diagnosealarm.....	2-15
OB 85 - Programmablauffehler	2-17
OB 86 - Slaveausfall / -wiederkehr	2-21
OB 100 - Neustart	2-23
OB 121 - Programmierfehler (Synchronfehler)	2-25
OB 122 - Peripheriezugriffsfehler	2-28
Teil 3 Integrierte SFBs	3-1
Übersicht.....	3-2
SFB 0 - CTU - Vorwärtszählen	3-3
SFB 1 - CTD - Rückwärtszählen	3-4
SFB 2 - CTUD - Vorwärts-/Rückwärtszählen	3-5
SFB 3 - TP - Impuls erzeugen.....	3-7
SFB 4 - TON - Einschaltverzögerung	3-9
SFB 5 - TOF - Ausschaltverzögerung.....	3-11
SFB 32 - DRUM - Schrittschaltwerk	3-13

SFB 52 - RDREC - Datensatz aus einem DP-V1-Slave lesen	3-18
SFB 53 - WRREC - Datensatz in einen DP-V1-Slave schreiben	3-20
SFB 54 - RALRM - Alarm von einem DP-V1-Slave empfangen	3-22
Teil 4 Integrierte Standard-SFCs	4-1
Übersicht Integrierte Standard-SFCs.....	4-3
Allgemeine und spezifische Fehlercodes RET_VAL.....	4-5
SFC 0 - SET_CLK - Uhrzeit stellen	4-8
SFC 1 - READ_CLK - Uhrzeit lesen	4-9
SFC 2 ... 4 - Betriebsstundenzähler.....	4-10
SFC 2 - SET_RTM - Betriebsstundenzähler setzen	4-11
SFC 3 - CTRL_RTM - Betriebsstundenzähler starten/stoppen	4-12
SFC 4 - READ_RTM - Betriebsstundenzähler auslesen.....	4-13
SFC 5 - GADR_LGC - Logische Adresse eines Kanals ermitteln	4-14
SFC 6 - RD_SINFO - Startinformation auslesen	4-16
SFC 12 - D_ACT_DP - DP-Slave aktivieren und deaktivieren	4-18
SFC 13 - DPNRM_DG - Slave-Diagnosedaten lesen	4-23
SFC 14 - DPRD_DAT - Konsistente Nutzdaten lesen	4-26
SFC 15 - DPWR_DAT - Konsistente Nutzdaten schreiben.....	4-28
SFC 17 - ALARM_SQ und SFC 18 - ALARM_S	4-30
SFC 19 - ALARM_SC - Quittierzustand der letzten Meldung.....	4-33
SFC 20 - BLKMOV - Variable kopieren	4-34
SFC 21 - FILL - Feld vorbesetzen	4-36
SFC 22 - CREAT_DB - Datenbaustein erzeugen	4-38
SFC 23 - DEL_DB - Datenbaustein löschen.....	4-40
SFC 24 - TEST_DB - Datenbaustein testen	4-41
SFC 28 ... SFC 31 - Uhrzeitalarm.....	4-42
SFC 32 - SRT_DINT - Verzögerungsalarm starten	4-46
SFC 33 - CAN_DINT - Verzögerungsalarm stornieren	4-47
SFC 34 - QRY_DINT - Verzögerungsalarm Status abfragen.....	4-48
SFC 36 - MSK_FLT - Synchronfehlerereignisse maskieren	4-49
SFC 37 - DMSK_FLT - Synchronfehlerereignisse demaskieren	4-50
SFC 38 - READ_ERR - Ereignisstatusregister lesen.....	4-51
SFC 39 - DIS_IRT - Alarmereignisse sperren.....	4-52
SFC 40 - EN_IRT - Gesperrte Alarmereignisse freigeben.....	4-54
SFC 41 - DIS_AIRT - Alarmereignisse verzögern	4-55
SFC 42 - EN_AIRT - Verzögerte Alarmereignissen freigeben	4-56
SFC 43 - RE_TRIGR - Zykluszeitüberwachung neu starten	4-56
SFC 44 - REPL_VAL - Ersatzwert in AKKU1 übertragen.....	4-57
SFC 46 - STP - CPU in STOP überführen.....	4-57
SFC 47 - WAIT - Verzögern des Anwenderprogramms.....	4-58
SFC 49 - LGC_GADR - Steckplatz ermitteln	4-59
SFC 50 - RD_LGADR - Alle Adressen eines Moduls lesen	4-60
SFC 51 - RDSYSST - Auslesen der Informationen der SZL	4-61
SFC 52 - WR_USMSG - Eintrag in Diagnosepuffer schreiben	4-63
SFC 54 - RD_DPARM - Vordefinierte Parameter lesen.....	4-67
SFC 55 - WR_PARM - Dynamische Parameter schreiben	4-69
SFC 56 - WR_DPARM - Vordefinierte Parameter schreiben.....	4-72
SFC 57 - PARM_MOD - Modul parametrieren.....	4-74

SFC 58 - WR_REC - Datensatz schreiben	4-76
SFC 59 - RD_REC - Datensatz lesen.....	4-79
SFC 64 - TIME_TCK - Systemzeit lesen	4-82
SFC 65 - X_SEND - Daten senden	4-83
SFC 66 - X_RCV - Daten empfangen	4-86
SFC 67 - X_GET - Daten lesen.....	4-91
SFC 68 - X_PUT - Daten schreiben	4-95
SFC 69 - X_ABORT - Verbindung abbrechen	4-98
SFC 81 - UBLKMOV - Variable ununterbrechbar kopieren.....	4-101
Teil 5 VIPA-spezifische Bausteine.....	5-1
Übersicht.....	5-2
VIPA-Bibliothek einbinden.....	5-4
FB 55 - IP_CONFIG - Programmierte Kommunikationsverbindung.....	5-5
FC 0 - SEND - Senden an CP 240	5-10
FC 1 - RECEIVE - Empfangen von CP 240.....	5-11
FC 5 - AG_SEND / FC 6 - AG_RECV - CP 243 Kommunikation	5-12
FC 8 - STEUERBIT - Modemfunktionalität CP 240	5-17
FC 9 - SYNCHRON_RESET - Synchronisation CPU und CP 240.....	5-18
FC 11 - ASCII_FRAGMENT - Fragment Datenempfang CP 240.....	5-19
Serielle Kommunikation - SFC 207 und SFC 216...218.....	5-20
SFC 207 - SER_CTRL	5-21
SFC 216 - SER_CFG	5-22
SFC 217 - SER_SND	5-26
SFC 218 - SER_RCV	5-29
SFC 219 - CAN_TLGR - CAN-Telegramm senden	5-30
MMC-Zugriff - SFC 220...222	5-33
SFC 220 - MMC_CR_F	5-34
SFC 221 - MMC_RD_F	5-36
SFC 222 - MMC_WR_F	5-37
SFC 223 - PWM - Pulsweitenmodulation	5-38
SFC 224 - HSC - High-speed-Counter	5-40
SFC 225 - HF_PWM - HF Pulsweitenmodulation	5-42
SFC 227 - TD_PRM - TD200-Kommunikation.....	5-44
SFC 228 - RW_KACHEL - Kacheldirektzugriff	5-46
Kachelkommunikation - SFC 230 ... 238	5-48
Kachelkommunikation - Parameterübergabe	5-51
Kachelkommunikation - Quell- bzw. Zielangaben.....	5-52
Kachelkommunikation - Anzeigenwort ANZW	5-55
Kachelkommunikation - Parametrierfehler PAFE	5-62
SFC 230 - SEND.....	5-63
SFC 231 - RECEIVE	5-64
SFC 232 - FETCH.....	5-65
SFC 233 - CONTROL	5-66
SFC 234 - RESET	5-67
SFC 235 - SYNCHRON.....	5-68
SFC 236 - SEND_ALL.....	5-69
SFC 237 - RECEIVE_ALL	5-70
SFC 238 - CTRL1	5-71

Über dieses Handbuch

Dieses Handbuch gibt Ihnen einen vollständigen Überblick über alle Bausteine, die in den VIPA-Standard-CPU's der Systeme 100V, 200V, 300V und 500V integriert sind.

Beschrieben sind Befehlsliste, integrierte OBs, SFBs, SFCs und alle VIPA-spezifischen Bausteine.

Überblick

Teil 1: Befehlsliste

Das Kapitel beinhaltet die Befehlsliste für die Standard-CPU's der Systeme 100V, 200V, 300V und 500V von VIPA. Die Befehlsliste soll Ihnen einen schnellen Überblick der Befehle und deren Syntax geben. Die Befehle sind in Themengruppen zusammengefasst, die in alphabetisch sortierter Reihenfolge aufgelistet sind.

Teil 2: Organisationsbausteine

In diesem Kapitel finden Sie eine Beschreibung der integrierten Organisationsbausteine der VIPA-Standard-CPU's der Systeme 100V, 200V, 300V und 500V.

Teil 3: Integrierte SFBs

Die Beschreibung der integrierten System-Funktionsbausteine der VIPA-Standard CPU's der Systeme 100V, 200V, 300V und 500V finden Sie hier.

Teil 4: Integrierte Standard SFCs

Hier finden Sie die Beschreibung der integrierten Standard-SFCs der VIPA Standard-CPU's der Systeme 100V, 200V, 300V und 500V.

Teil 5: VIPA-spezifische Bausteine

Hier finden Sie die Beschreibung der VIPA spezifischen Bausteine, die ausschließlich in den Standard-CPU's der Systeme 100V, 200V, 300V und 500V der VIPA Verwendung finden. Bitte beachten Sie, dass manche der hier aufgeführten Bausteine nicht in allen System-CPU's integriert sind. Die Zuordnung der entsprechenden Bausteine zu den Systemen finden Sie im Kapitel "Übersicht" in einer Tabelle.

Zielsetzung und Inhalt	Dieses Handbuch beinhaltet eine Befehlsliste und die Beschreibung der integrierten Bausteine, die ausschließlich in den Standard-CPU's der Systeme 100V, 200V, 300V und 500V der VIPA Verwendung finden.
Zielgruppe	Das Handbuch ist geschrieben für Anwender mit Grundkenntnissen in der Automatisierungstechnik.
Aufbau des Handbuchs	Das Handbuch ist in Kapitel gegliedert. Jedes Kapitel beschreibt eine abgeschlossene Thematik.
Orientierung im Dokument	Als Orientierungshilfe stehen im Handbuch zur Verfügung: <ul style="list-style-type: none">• Gesamt-Inhaltsverzeichnis am Anfang des Handbuchs• Übersicht der beschriebenen Themen am Anfang jedes Kapitels• Stichwortverzeichnis (Index) am Ende des Handbuchs
Verfügbarkeit	Das Handbuch ist verfügbar in: <ul style="list-style-type: none">• gedruckter Form auf Papier• in elektronischer Form als PDF-Datei (Adobe Acrobat Reader)
Piktogramme Signalwörter	Besonders wichtige Textteile sind mit folgenden Piktogrammen und Signalworten ausgezeichnet:

**Gefahr!**

Unmittelbar drohende oder mögliche Gefahr.
Personenschäden sind möglich.

**Achtung!**

Bei Nichtbefolgen sind Sachschäden möglich.

**Hinweis!**

Zusätzliche Informationen und nützliche Tipps

Teil 1 Befehlsliste

Überblick

Das folgende Kapitel beinhaltet die Befehlsliste für die Standard-CPU's der Systeme 100V, 200V, 300V und 500V von VIPA. Die Befehlsliste soll Ihnen einen schnellen Überblick der Befehle und deren Syntax geben. Die Befehle sind in Themengruppen zusammengefasst, die in alphabetisch sortierter Reihenfolge aufgelistet sind.

Über das Inhaltsverzeichnis gelangen Sie zu den Themengruppen.

Die alphabetisch geordnete Operationen-Liste bietet Ihnen direkten Zugriff auf die Operationen.

Da die Parameter in die Befehlsliste integriert sind, wurde auf eine gesonderte Parameterliste verzichtet.

Inhalt

Thema	Seite
Teil 1 Befehlsliste	1-1
Alphabetisches Verzeichnis der Operationen	1-2
Abkürzungen	1-4
Register	1-6
Adressierungsbeispiele	1-7
Arithmetische Operationen	1-9
Baustein-Operationen	1-11
Bild- und Null-Operationen	1-12
Flanken-Operationen	1-12
Lade-Operationen	1-13
Schiebe-Operationen	1-16
Setzen/Rücksetzen von Bitoperanden	1-17
Sprung-Operationen	1-19
Transfer-Operationen	1-20
Umwandlungs-Operationen	1-23
Vergleichs-Operationen	1-24
Verknüpfungs-Operationen (Bit)	1-25
Verknüpfungs-Operationen (Wort)	1-31
Zeit-Operationen	1-31
Zähl-Operationen	1-32

Alphabetisches Verzeichnis der Operationen

Operation	Seite
)	1-27
+	1-10
+AR1	1-10
+AR2	1-10
+I	1-9
+D	1-9
+R	1-9
-D	1-9
-I	1-9
-R	1-9
*D	1-9
*I	1-9
*R	1-9
/D	1-9
/I	1-9
/R	1-9
=	1-17
==D	1-24
==I	1-24
==R	1-24
<=D	1-24
<=I	1-24
<=R	1-24
<D	1-24
<I	1-24
<R	1-24
<>D	1-24
<>I	1-24
<>R	1-24
>=D	1-24
>=I	1-24
>=R	1-24
>D	1-24
>I	1-24
>R	1-24
ABS	1-9
ACOS	1-10
ASIN	1-10
ATAN	1-10
AUF	1-11

Operation	Seite
BE	1-11
BEA	1-11
BEB	1-11
BLD	1-12
BTD	1-23
BTI	1-23
CALL	1-11
CC	1-11
CLR	1-18
COS	1-10
DEC	1-22
DTB	1-23
DTR	1-23
EXP	1-10
FN	1-12
FP	1-12
FR	1-31, 1-32
INC	1-22
INVD	1-23
INVI	1-23
ITB	1-23
ITD	1-23
L	1-13, 1-14, 1-15, 1-22
LAR1	1-21
LAR2	1-21
LC	1-15
LN	1-10
LOOP	1-19
MOD	1-9
NEGD	1-23
NEGI	1-23
NEGR	1-9
NOP	1-12
NOT	1-18
O	1-25, 1-27, 1-28, 1-29
O(1-27
OD	1-31
ON	1-26, 1-28, 1-30
ON(1-27

OW	1-31
POP	1-22
PUSH	1-22
R	1-17, 1-31, 1-32
RLD	1-16
RLDA	1-16
RND	1-23
RND+	1-23
RND-	1-23
RRD	1-16
RRDA	1-16
S	1-17, 1-32
SA	1-31
SAVE	1-18
SE	1-31
SET	1-18
SI	1-31
SIN	1-10
SLD	1-16
SLW	1-16
SPA	1-19
SPB	1-19
SPBB	1-19
SPBI	1-19
SPBIN	1-19
SPBN	1-19
SPBNB	1-19
SPL	1-19
SPM	1-19
SPMZ	1-19
SPN	1-19
SPO	1-19
SPP	1-19
SPPZ	1-19
SPS	1-19
SPU	1-19
SPZ	1-19
SQR	1-10
SQRT	1-10
SRD	1-16
SRW	1-16
SS	1-31
SSD	1-16

SSI	1-16
SV	1-31
T	1-20, 1-21, 1-22
TAD	1-22
TAK	1-22
TAN	1-10
TAR	1-22
TAR1	1-21
TAR2	1-22
TAW	1-22
TDB	1-11
TRUNC	1-23
U	1-25, 1-28, 1-29
U(1-27
UC	1-11
UD	1-31
UN	1-25, 1-28, 1-29
UN(1-27
UW	1-31
X	1-26, 1-28, 1-30
X(1-27
XN	1-26, 1-28, 1-30
XN(1-27
XOD	8-30
XOW	1-31
ZR	1-32
ZV	1-32

Abkürzungen

Abkürzung	Erläuterung
2#	Binärkonstante
a	Byteadresse
A	Ausgang (im PAA)
A0	Ergebnisanzeige
A1	Ergebnisanzeige
AB	Ausgangsbyte (im PAA)
AD	Ausgangsdoppelwort (im PAA)
AKKU	Register für die Verarbeitung von Bytes, Worten und Doppelworten.
AR	Adressregister, enthalten die bereichsinternen oder bereichsübergreifenden Adressen für die registerindirekt adressierten Operationen
AW	Ausgangswort (im PAA)
b	Bitadresse
B	bereichsübergreifend, registerindirekt adressiertes Byte
B (b1,b2)	Konstante, 2Byte
B (b1,b2,b3,b4)	Konstante, 4Byte
BIE	Binäresultat
B#16#	Byte hexadezimal
c	Operandenbereich
C#	Zählerkonstante (BCD-codiert)
D	bereichsübergreifend, registerindirekt adressiertes Doppelwort
D#	IEC Datumskonstante
DB	Datenbaustein
DBB	Datenbyte im Datenbaustein
DBD	Datendoppelwort im Datenbaustein
DBW	Datenwort im Datenbaustein
DBX	Datenbit im Datenbaustein
DI	Instanz-Datenbaustein
DIB	Datenbyte im Instanz-DB
DID	Datendoppelwort im Instanz-DB
DIW	Datenwort im Instanz-DB
DIX	Datenbit im Instanz-DB
DW#16#	Doppelwort hexadezimal
E	Eingang (im PAE)
EB	Eingangsbyte (im PAE)
ED	Eingangsdoppelwort (im PAE)
/ER	Erstabfrage
EW	Eingangswort (im PAE)
f	Timer-/Zähler-Nr.
FB	Funktionsbaustein
FC	Funktionen
g	Operandenbereich
h	Operandenbereich
i	Operandenbereich
i8	Ganzzahl (8Bit)
i16	Ganzzahl (16Bit)

Fortsetzung ...

... Fortsetzung

Abkürzung	Erläuterung
i32	Ganzzahl (32Bit)
k8	Konstante (8Bit)
k16	Konstante (16Bit)
k32	Konstante (32Bit)
L	Lokaldaten
L#	Ganzzahlkonstante (32Bit)
LB	Lokaldatenbyte
LD	Lokaldaten-Doppelwort
LW	Lokaldatenwort
m	Pointer-Konstante
M	Merker
MARKE	Symbolische Sprungadresse (max. 4 Buchstaben)
MB	Merkerbyte
MD	Merkerdoppelwort
MW	Merkerwort
n	Binärkonstante
OB	Operationsbaustein
OR	Oder
OS	Überlaufspeichernd
OV	Überlauf
p	Hexadezimalkonstante
P#	Pointerkonstante
PAA	Prozessabbild der Ausgänge
PAB	Peripherieausgangsbyte (direkter Peripheriezugriff)
PAD	Peripherieausgangsdoppelwort (direkter Peripheriezugriff)
PAE	Prozessabbild der Eingänge
PAW	Peripherieausgangswort (direkter Peripheriezugriff)
PEB	Peripherieeingangsbyte (direkter Peripheriezugriff)
PED	Peripherieeingangsdoppelwort (direkter Peripheriezugriff)
PEW	Peripherieeingangswort (direkter Peripheriezugriff)
q	Realzahl (32-Gleitpunktzahl)
r	Baustein-Nr.
S5T#	S5-Zeitkonstante (16Bit), dient zum Laden der S5-Timer
SFB	Systemfunktionsbausteine
SFC	Systemfunktionen
STA	Status
T	Timer (Zeiten)
T#	Zeitkonstante (16/32Bit)
TOD#	IEC-Zeitkonstante
VKE	Verknüpfungsergebnis
W	bereichsübergreifend, registerindirekt adressiertes Wort
W#16#	Wort hexadezimal
Z	Zähler

Register

AKKU1 und AKKU2 (32Bit)

Die AKKUs sind Register für die Verarbeitung von Bytes, Worten oder Doppelworten. Hierzu werden die Operanden in die AKKUs geladen und dort verknüpft. Das Ergebnis der Operation steht immer im AKKU1.

AKKU	Bit
AKKUx (x=1 bis 2)	Bit 0 ... Bit 31
AKKUx-L	Bit 0 ... Bit 15
AKKUx-H	Bit 16 ... Bit 31
AKKUx-LL	Bit 0 ... Bit 7
AKKUx-LH	Bit 8 ... Bit 15
AKKUx-HL	Bit 16 ... Bit 23
AKKUx-HH	Bit 24 ... Bit 31

Adressregister AR1 und AR2 (32Bit)

Die Adressregister enthalten die bereichsinternen oder bereichsübergreifenden Adressen für die registerindirekt adressierenden Operationen. Die Adressregister sind 32Bit breit.

Die bereichsinternen bzw. bereichsübergreifenden Adressen haben folgenden Aufbau:

bereichsinterne Adresse:

00000000 00000bbb bbbbbb bbbbx

bereichsübergreifende Adresse:

10000yyy 00000bbb bbbbbb bbbbx

Legende: b Byteadresse
 x Bitnummer
 Y Bereichskennung
 (siehe Kapitel "Adressierungsbeispiele")

Statuswort (16Bit)

Die Anzeigen werden durch die Operationen ausgewertet oder gesetzt. Das Statuswort ist 16Bit breit.

Bit	Belegung	Bedeutung
0	/ER	Erstabfrage*
1	VKE	Verknüpfungsergebnis
2	STA	Status*
3	OR	Oder*
4	OS	Überlauf speichernd
5	OV	Überlauf
6	A0	Ergebnisanzeige
7	A1	Ergebnisanzeige
8	BIE	Binärerergebnis
9 ... 15	nicht belegt	-

* Bit kann im Anwenderprogramm mit Operation L STW nicht ausgewertet werden, da das Bit zur Programmlaufzeit nicht aktualisiert wird.

Adressierungsbeispiele

Adressierungsbeispiel	Beschreibung
Unmittelbare Adressierung	
L +27	Lade 16-Bit-Ganzzahlkonstante "27" in AKKU1
L L#-1	Lade 32-Bit-Ganzzahlkonstante "-1" in AKKU1
L 2#1010101010101010	Lade Binärkonstante in AKKU1
L DW#16#A0F0_BCFD	Lade Hexadezimalkonstante in AKKU1
L 'Ende'	Lade ASCII-Zeichen in AKKU1
L T#500ms	Lade Zeitwert in AKKU1
L C#100	Lade Zählerwert in AKKU1
L B#(100,12)	Lade Konstante als 2Byte
L B#(100,12,50,8)	Lade Konstante als 4Byte
L P#10.0	Lade bereichsinterne Pointer in AKKU1
L P#E20.6	Lade bereichsübergreifende Pointer in AKKU1
L -2.5	Lade Realzahl in AKKU1
L D#1995-01-20	Lade Datum
L TOD#13:20:33.125	Lade Uhrzeit
Direkte Adressierung	
U E 0.0	UND-Verknüpfung des Eingangsbits 0.0
L EB 1	Lade Eingangsbyte 1 in AKKU1
L EW 0	Lade Eingangswort 0 in AKKU1
L ED 0	Lade Eingangsdoublewort 0 in AKKU1
Indirekte Adressierung Timer/Zähler	
SI T [LW 8]	Starte Timer; die Timer-Nr. steht im Lokaldatenwort 8
ZV Z [LW 10]	Starte Zähler; die Zähler-Nr. steht im Lokaldatenwort 10
Speicherindirekte, bereichsinterne Adressierung	
U E [LD 12] Bsp.: LP#22.2 T LD 12 U E [LD 12]	UND-Operation; die Adresse des Eingangs steht als Pointer im Lokaldaten-Doppelwort 12
U E [DBD 1]	UND-Operation; die Adresse des Eingangs steht als Pointer im Datendoppelwort 1 des DB
U A [DID 12]	UND-Operation; die Adresse des Ausgangs steht als Pointer im Datendoppelwort 12 des Instanz-DB
U A [MD 12]	UND-Operation; die Adresse des Ausgangs steht als Pointer im Merkerdoppelwort 12

Fortsetzung ...

... Fortsetzung

Registerindirekte, bereichsinterne Adressierung			
U E [AR1,P#12.2]		UND-Operation; die Adresse des Eingangs errechnet sich zu "Pointerwert im Adressregister 1 + Pointer P#12.2"	
Registerindirekte, bereichsübergreifende Adressierung			
Für die bereichsübergreifende, registerindirekte Adressierung muss die Adresse zusätzlich eine Bereichskennung in den Bits 24-26 enthalten. Die Adresse steht im Adressregister.			
Bereichskennung	Codierung binär	hex.	Bereich
P	1000 0000	80	Peripheriebereich
E	1000 0001	81	Eingangsbereich
A	1000 0010	82	Ausgangsbereich
M	1000 0011	83	Merkerbereich
DB	1000 0100	84	Datenbereich
DI	1000 0101	85	Instanz-Datenbereich
L	1000 0110	86	Lokaldatenbereich
VL	1000 0111	87	Vorgänger-Lokaldatenbereich (Zugriff auf Lokaldaten des aufrufenden Bausteins)
L B [AR1,P#8.0]		Lade Byte in AKKU1; die Adresse errechnet sich aus "Pointerwert im Adressregister 1 + Pointer P#8.0"	
U [AR1,P#32.3]		UND-Operation; die Adresse des Operanden errechnet sich aus "Pointerwert im Adressregister 1 + Pointer P#32.3"	
Adressierung über Parameter			
U Parameter		Der Operand wird über den Parameter adressiert	

Beispiel zur Pointerberechnung

Beispiel bei Summe der Bitadressen ≤ 7:

LAR1 P#8.2
U E [AR1,P#10.2]

Ergebnis: Adressiert wird Eingang 18.4
(durch jeweilige Addition der Byte- und Bitadressen)

Beispiel bei Summe der Bitadressen > 7:

L MD 0 beliebig berechneter Pointer, z.B. P#10.5
LAR1
U E [AR1,P#10.7]

Ergebnis: Adressiert wird Eingang 21.4
(durch Addition der Byte- und Bitadressen mit Übertrag)

Befehl	Operand	Parameter	Statuswort										Funktionsbezeichnung	Länge in Worten		
			BIE	A1	A0	OV	OS	OR	STA	VKE	ER					
															: Operation hängt ab von	
															: Operation beeinflusst	

Arithmetische Operationen

Festpunktarithmetik (16Bit)			Statuswort										Arithmetische Operationen zweier 16-Bit-Zahlen. Das Ergebnis steht im AKKU1 bzw. AKKU1-L.	
+I	-		BIE	A1	A0	OV	OS	OR	STA	VKE	ER		Addiere zwei Ganzzahlen (16Bit) (AKKU1-L)=(AKKU1-L)+(AKKU2-L)	1
			-	-	-	-	-	-	-	-	-			
-I	-		-	J	J	J	J	-	-	-	-		Subtrahiere zwei Ganzzahlen (16Bit) (AKKU1-L)=(AKKU2-L)-(AKKU1-L)	1
I	-												Multipliziere zwei Ganzzahlen (16Bit) (AKKU1-L)=(AKKU2-L)(AKKU1-L)	1
/I	-												Dividiere zwei Ganzzahlen (16Bit) (AKKU1-L)=(AKKU2-L):(AKKU1-L) Im AKKU1-H steht der Rest der Division	1
Festpunktarithmetik (32Bit)			Statuswort										Arithmetische Operationen zweier 32-Bit-Zahlen. Das Ergebnis steht im AKKU1.	
+D	-		BIE	A1	A0	OV	OS	OR	STA	VKE	ER		Addiere zwei Ganzzahlen (32Bit) (AKKU1)=(AKKU2)+(AKKU1)	1
			-	-	-	-	-	-	-	-	-			
-D	-		-	J	J	J	J	-	-	-	-		Subtrahiere zwei Ganzzahlen (32Bit) (AKKU1)=(AKKU2)-(AKKU1)	1
D	-												Multipliziere zwei Ganzzahlen (32Bit) (AKKU1)=(AKKU2)(AKKU1)	1
/D	-												Dividiere zwei Ganzzahlen (32Bit) (AKKU1)=(AKKU2):(AKKU1)	1
MOD	-												Dividiere zwei Ganzzahlen (32Bit) und lade den Rest der Division in AKKU1 (AKKU1)=Rest von [(AKKU2):(AKKU1)]	1
Gleitpunktarithmetik (32Bit)			Statuswort										Das Ergebnis der arithmetische Operationen steht im AKKU1. Die Ausführungszeit der Operation hängt vom Wert ab, der berechnet werden soll.	
+R	-		BIE	A1	A0	OV	OS	OR	STA	VKE	ER		Addiere zwei Realzahlen (32Bit) (AKKU1)=(AKKU2)+(AKKU1)	1
			-	-	-	-	-	-	-	-	-			
-R	-		-	J	J	J	J	-	-	-	-		Subtrahiere zwei Realzahlen (32Bit) (AKKU1)=(AKKU2)-(AKKU1)	1
R	-												Multipliziere zwei Realzahlen (32Bit) (AKKU1)=(AKKU2)(AKKU1)	1
/R	-												Dividiere zwei Realzahlen (32Bit) (AKKU1)=(AKKU2):(AKKU1)	1
NEGR	-		BIE	A1	A0	OV	OS	OR	STA	VKE	ER		Negiere Realzahl im AKKU1	1
			-	-	-	-	-	-	-	-	-			
ABS	-		-	-	-	-	-	-	-	-	-		Bilde Betrag der Realzahl im AKKU1	1

Befehl	Operand	Parameter	Statuswort										Funktionsbezeichnung	Länge in Worten		
			BIE	A1	A0	OV	OS	OR	STA	VKE	ER					
															: Operation hängt ab von	
															: Operation beeinflusst	

Quadratwurzel, Quadrat (32Bit)			<i>Statuswort</i>										Das Ergebnis der Operationen steht im AKKU1. Operationen sind durch Alarme unterbrechbar.	
SQRT	-		BIE	A1	A0	OV	OS	OR	STA	VKE	ER	Berechne die Quadratwurzel einer Realzahl in AKKU1	1	
			-	-	-	-	-	-	-	-	-			
SQR	-		-	J	J	J	J	-	-	-	-	Quadriere die Restzahl in AKKU1	1	
Logarithmusfunktion (32Bit)			<i>Statuswort</i>										Das Ergebnis der Logarithmusfunktion steht im AKKU1. Die Operationen sind durch Alarme unterbrechbar.	
LN	-		BIE	A1	A0	OV	OS	OR	STA	VKE	ER	Bilde den natürlichen Logarithmus einer Realzahl in AKKU1	1	
			-	-	-	-	-	-	-	-	-			
EXP	-		-	J	J	J	J	-	-	-	-	Berechne den Exponentialwert einer Realzahl in AKKU1 zur Basis e (=2,71828)	1	
Trigonometrische Funktionen (32Bit)			<i>Statuswort</i>										Das Ergebnis der Operationen steht im AKKU1. Operationen sind durch Alarme unterbrechbar.	
SIN ¹	-		BIE	A1	A0	OV	OS	OR	STA	VKE	ER	Berechne den Sinus der Realzahl	1	
			-	-	-	-	-	-	-	-	-			
ASIN ²	-		-	J	J	J	J	-	-	-	-	Berechne den Arcussinus der Realzahl	1	
COS ¹	-											Berechne den Cosinus der Realzahl	1	
ACOS ²	-											Berechne den Arcuscossinus der Realzahl	1	
TAN ¹	-											Berechne den Tangens der Realzahl	1	
ATAN ²	-											Berechne den Arcustangens der Realzahl	1	
Addition von Konstanten													Addition von Ganzzahl-Konstanten zum AKKU1. Die Anzeigen werden nicht beeinflusst.	
+	i8												Addiere eine 8-Bit Integer-Konstante	1
+	i16												Addiere eine 16-Bit Integer-Konstante	2
+	i32												Addiere eine 32-Bit Integer-Konstante	3
Addition über Adressregister													Addition einer Ganzzahl (16Bit) zum Inhalt des Adressregisters. Der Wert steht in der Operation oder im AKKU 1-L. Die Anzeigen werden nicht beeinflusst.	
+AR1	-												Addiere Inhalt von AKKU1-L zum AR1	1
+AR1	m												Addiere Pointer-Konstante zum AR1	2
+AR2	-												Addiere Inhalt von AKKU1-L zum AR2	1
+AR2	m												Addiere Pointer-Konstante zum AR2	2

1 Geben Sie den Winkel in Grad an. Der Winkel muss als Gleitpunktzahlenwert in AKKU1 angegeben werden.

2 Das Ergebnis ist ein Winkel in Grad

Befehl	Operand	Parameter	Statuswort										Funktionsbezeichnung	Länge in Worten		
			BIE	A1	A0	OV	OS	OR	STA	VKE	/ER					
															: Operation hängt ab von	
															: Operation beeinflusst	

Baustein-Operationen

Baustein-Aufrufoperationen			Statuswort											
CALL	FB p,DB r		BIE	A1	A0	OV	OS	OR	STA	VKE	/ER		Unbedingter Aufruf eines FB, mit Parameterübergabe	1
			-	-	-	-	-	-	-	-	-			
CALL	SFB p,DB r		-	-	-	-	0	0	1	-	0		Unbedingter Aufruf eines SFB, mit Parameterübergabe	2
CALL	FC p												Unbedingter Aufruf einer Funktion, mit Parameterübergabe	1
CALL	SFC p												Unbedingter Aufruf einer SFC, mit Parameterübergabe	2
UC	FB q FC q Parameter												Unbedingter Aufruf von Bausteinen, ohne Parameter- übergabe FB/FC-Aufruf über Parameter	1
CC	FB q FC q Parameter		BIE	A1	A0	OV	OS	OR	STA	VKE	/ER		Bedingter Aufruf von Bausteinen, ohne Parameter- übergabe FB/FC-Aufruf über Parameter	1
			-	-	-	-	-	-	-	J	-			
			-	-	-	-	0	0	1	-	0			
AUF	DB p DI p Parameter		BIE	A1	A0	OV	OS	OR	STA	VKE	/ER		Aufschlagen eines Datenbausteins	1/2
			-	-	-	-	-	-	-	-	-		Aufschlagen eines Instanz-Datenbausteins	2
			-	-	-	-	-	-	-	-	-		Aufschlagen eines Datenbausteins über Parameter	2
Baustein-Endeoperationen			Statuswort											
BE			BIE	A1	A0	OV	OS	OR	STA	VKE	/ER		Beende Baustein	1
			-	-	-	-	-	-	-	-	-			
BEA			-	-	-	-	0	0	1	-	0		Beende Baustein absolut	1
BEB			BIE	A1	A0	OV	OS	OR	STA	VKE	/ER		Beende Baustein bei VKE="1"	
			-	-	-	-	-	-	-	J	-			
			-	-	-	-	J	0	1	1	0			
Tausche Datenbausteine													Tauschen der beiden aktuellen Datenbau- steine. Der aktuelle Datenbaustein wird zum aktuellen Instanz-Datenbaustein und um- gekehrt. Die Anzeigen werden nicht beeinflusst.	
TDB													Tausche Datenbausteine	1

Befehl	Operand	Parameter	Statuswort										Funktionsbezeichnung	Länge in Worten		
			BIE	A1	A0	OV	OS	OR	STA	VKE	/ER					
															: Operation hängt ab von	
															: Operation beeinflusst	

Bild- und Null-Operationen

Bild- und Null-Operation			Statuswort										Funktionsbezeichnung	Länge in Worten		
BLD	0 ... 255														Das Statuswort wird nicht beeinflusst.	
															Bildaufbau-Operation; wird von der CPU wie eine Null-Operation behandelt	1
NOP	0 1														Nulloperation	1

Flanken-Operationen

Flanken-Operationen			Statuswort										Funktionsbezeichnung	Länge in Worten		
															Erkennen eines Flankenwechsels. Der aktuelle Signalzustand im VKE wird verglichen mit dem Signalzustand im Operanden, dem "Flankenmerker". FP erkennt einen Flankenwechsel von "0" nach "1". FN erkennt einen Flankenwechsel von "1" nach "0".	
FP	E/A a.b	0.0 ... 127.7	BIE	A1	A0	OV	OS	OR	STA	VKE	/ER				Anzeigen der steigenden Flanke im VKE, Flankenhilfs-	2
	M a.b	0.0 ... 1023.7	-	-	-	-	-	-	-	J	-				merker ist das in der Operation adressierte Bit	2
	L a.b	0.0 ... 1043.7	-	-	-	-	-	0	J	J	1					2
	DBX a.b	0.0 ... 8191.7														2
	DIX a.b	0.0 ... 8191.7														2
	c [AR1,m] c [AR2,m] [AR1,m] [AR2,m] Parameter															2 2 2 2 2
FN	E/A a.b	0.0 ... 127.7	BIE	A1	A0	OV	OS	OR	STA	VKE	/ER				Anzeigen der fallenden Flanke im VKE, Flankenhilfs-	2
	M a.b	0.0 ... 1023.7	-	-	-	-	-	-	-	J	-				merker ist das in der Operation adressierte Bit	2
	L a.b	0.0 ... 1043.7	-	-	-	-	-	0	J	J	1					2
	DBX a.b	0.0 ... 8191.7														2
	DIX a.b	0.0 ... 8191.7														2
	c [AR1,m] c [AR2,m] [AR1,m] [AR2,m] Parameter															2 2 2 2 2

Befehl	Operand	Parameter	Statuswort								Funktionsbezeichnung	Länge in Worten	
			BIE	A1	A0	OV	OS	OR	STA	VKE/ER			
												: Operation hängt ab von	
												: Operation beeinflusst	

Lade-Operationen

Lade-Operationen				Laden der Operanden in AKKU1, zuvor wird der alte Inhalt von AKKU1 in AKKU2 gerettet. Das Statuswort wird nicht beeinflusst.		
L	EB	a	0 ... 127		Lade ... Eingangsbyte	1/2
	AB	a	0 ... 127		Ausgangsbyte	1/2
	PEB	a	0 ... 1023		Peripherie-Eingangsbyte	2
	MB	a	0 ... 1023		Merkerbyte	1/2
	LB	a	0 ... 1043		Lokaldatenbyte	2
	DBB	a	0 ... 8191		Datenbyte	2
	DIB	a	0 ... 8191		Instanz-Datenbyte ... in AKKU1	2
	g [AR1,m]				registerindirekt, bereichsintern (AR1)	2
	g [AR2,m]				registerindirekt, bereichsintern (AR2)	2
	B [AR1,m]				bereichsübergreifend (AR1)	2
B [AR2,m]				bereichsübergreifend (AR2)	2	
Parameter				über Parameter	2	
L	EW	a	0 ... 126		Lade ... Eingangswort	1/2
	AW	a	0 ... 126		Ausgangswort	1/2
	PEW	a	0 ... 1022		Peripherie-Eingangswort	
	MW	a	0 ... 1022		Merkerwort	1/2
	LW	a	0 ... 1042		Lokaldatenwort	2
	DBW	a	0 ... 8190		Datenwort	1/2
	DIW	a	0 ... 8190		Instanz-Datenwort ... in AKKU1-L	1/2
	h [AR1,m]				registerindirekt, bereichsintern (AR1)	2
	h [AR2,m]				registerindirekt, bereichsintern (AR2)	2
	W [AR1,m]				bereichsübergreifend (AR1)	2
W [AR2,m]				bereichsübergreifend (AR2)	2	
Parameter				über Parameter	2	

Befehl	Operand	Parameter	Statuswort										Funktionsbezeichnung	Länge in Worten		
			BIE	A1	AD	OV	OS	OR	STA	VKE	/ER					
															: Operation hängt ab von	
															: Operation beeinflusst	

Lade-Operationen					
				Laden der Operanden in AKKU1, zuvor wird der alte Inhalt von AKKU1 in AKKU2 gerettet. Das Statuswort wird nicht beeinflusst.	
L	ED a	0 ... 124		Lade ... Eingangsdoppelwort	1/2
	AD a	0 ... 124		Ausgangsdoppelwort	1/2
	PED a	0 ... 1020		Peripherie-Eingangsdoppelwort	2
	MD a	0 ... 1020		Merkerdoppelwort	1/2
	LD a	0 ... 1040		Lokaldatendoppelwort	2
	DBD a	0 ... 8188		Datendoppelwort	2
	DID a	0 ... 8188		Instanz-Datendoppelwort ... in AKKU1-L	2
	i [AR1,m]			registerindirekt, bereichsintern (AR1)	2
	i [AR2,m]			registerindirekt, bereichsintern (AR2)	2
	D [AR1,m]			bereichsübergreifend (AR1)	2
	D [AR2,m]			bereichsübergreifend (AR2)	2
	Parameter			über Parameter	2
L	k8			Lade ... 8-Bit-Konstante in AKKU1-LL	1
	k16			16-Bit-Konstante in AKKU1-L	2
	k32			32-Bit-Konstante in AKKU1	3
	Parameter			Lade Konstante in AKKU1 (über Parameter adressiert).	2
L	2#n			Lade 16-Bit-Binärkonstante in AKKU1-L	2
				Lade 32-Bit-Binärkonstante in AKKU1	3
L	B#8#p			Lade 8-Bit-Hexadezimalkonstante in AKKU1-LL	1
	W#16#p			Lade 16-Bit-Hexadezimalkonstante in AKKU1-L	2
	DW#16#p			Lade 32-Bit-Hexadezimalkonstante in AKKU1	3
L	x			Lade ein Zeichen	
L	xx			Lade zwei Zeichen	2
L	xxx			Lade drei Zeichen	
L	xxxx			Lade vier Zeichen	3
L	D# Datum			Lade IEC-Datum (BCD-codiert)	3
L	S5T# Zeitwert			Lade Zeitkonstante (16-Bit)	2
L	TOD# Zeitwert			Lade 32-Bit-Zeitkonstante (IEC-Tageszeit)	3
L	T# Zeitwert			Lade 16-Bit-Zeitkonstante	2
				Lade 32-Bit-Zeitkonstante	3
L	C# Zählwert			Lade 16-Bit-Zählerkonstante	2
L	P# Bitpointer			Lade Bitpointer	3
L	L# Integer			Lade 32-Bit-Ganzzahlkonstante	3
L	Realzahl			Lade Realzahl	3

Befehl	Operand	Parameter	Statuswort									Funktionsbezeichnung	Länge in Worten		
			BIE	A1	A0	OV	OS	OR	STA	VKE	ER				
														: Operation hängt ab von	
														: Operation beeinflusst	

Lade-Operationen für Timer und Zähler					
				Laden eines Zeitwertes oder Zählwertes in AKKU1. Zuvor wird der alte Inhalt von AKKU1 in AKKU2 gerettet. Die Anzeigen werden nicht beeinflusst.	
L	T f Timerpara.	0 ... 255		Lade Zeitwert Lade Zeitwert (über Parameter adressiert)	1/2 2
L	Z f Zählerpara.	0 ... 255		Lade Zählwert Lade Zählwert (über Parameter adressiert)	1/2 2
LC	T f Timerpara.	0 ... 255		Lade Zeitwert BCD-codiert Lade Zeitwert BCD-codiert (über Parameter adressiert)	1/2 2
LC	Z f Zählerpara.	0 ... 255		Lade Zählwert BCD-codiert Lade Zählwert BCD-codiert (über Parameter adressiert)	1/2 2

Befehl	Operand	Parameter	Statuswort										Funktionsbezeichnung	Länge in Worten		
			BIE	A1	A0	OV	OS	OR	STA	VKE	/ER					
															: Operation hängt ab von	
															: Operation beeinflusst	

Schiebe-Operationen

Schiebe-Operationen			Statuswort													
															Schiebe Inhalt von AKKU1 oder AKKU1-L um die angegebene Anzahl von Stellen nach links/rechts. Ist kein Operand angegeben, schiebe Anzahl in AKKU2-LL. Freierwerdende Stellen werden mit Nullen bzw. mit dem Vorzeichen aufgefüllt. Zuletzt geschobenes Bit steht im Anzeigenbit A1.	
SLW	-		BIE	A1	A0	OV	OS	OR	STA	VKE	/ER		Schiebe Inhalt von AKKU1-L nach links, freierwerdende Stellen werden mit Nullen aufgefüllt	1		
SLW	0 ... 15		-	-	-	-	-	-	-	-	-					
SLD	-		-	J	J	J	-	-	-	-	-		Schiebe Inhalt von AKKU1 nach links, freierwerdende Stellen werden mit Nullen aufgefüllt	1		
SLD	0 ... 32															
SRW	-												Schiebe Inhalt von AKKU1-L nach rechts, freierwerdende Stellen werden mit Nullen aufgefüllt	1		
SRW	0 ... 15															
SRD	-												Schiebe Inhalt von AKKU1 nach rechts, freierwerdende Stellen werden mit Nullen aufgefüllt	1		
SRD	0 ... 32															
SSI	-												Schiebe den Inhalt von AKKU1-L mit Vorzeichen nach rechts, freierwerdende Stellen werden mit den Vorzeichen (Bit 15) aufgefüllt	1		
SSI	0 ... 15															
SSD	-												Schiebe den Inhalt von AKKU1 mit Vorzeichen nach rechts	1		
SSD	0 ... 32															
Rotier-Operationen			Statuswort													
														Rotiere Inhalt von AKKU1 um die angegebene Anzahl von Stellen nach links/rechts. Ist kein Operand angegeben, rotiere Anzahl in AKKU2-LL.		
RLD	-		BIE	A1	A0	OV	OS	OR	STA	VKE	/ER		Rotiere Inhalt von AKKU1 nach links	1		
RLD	0 ... 32		-	-	-	-	-	-	-	-	-					
RRD	-		-	J	J	J	-	-	-	-	-		Rotiere Inhalt von AKKU1 nach rechts	1		
RRD	0 ... 32															
RLDA	-		BIE	A1	A0	OV	OS	OR	STA	VKE	/ER		Rotiere Inhalt von AKKU1 um eine Bitposition nach links über Anzeigebit A1			
RLDA			-	-	-	-	-	-	-	-	-					
RRDA	-		-	J	0	0	-	-	-	-	-		Rotiere Inhalt von AKKU1 um eine Bitposition nach rechts über Anzeigebit A1			
RRDA																

Befehl	Operand	Parameter	Statuswort										Funktionsbezeichnung	Länge in Worten		
			BIE	A1	A0	OV	OS	OR	STA	VKE	/ER					
															: Operation hängt ab von	
															: Operation beeinflusst	

Setzen/Rücksetzen von Bitoperanden

Setzen/Rücksetzen von Bitoperanden			Statuswort										Zuweisen des Wertes "1" oder "0" bzw. des VKE an den adressierten Operanden.		
			BIE	A1	A0	OV	OS	OR	STA	VKE	/ER				
S	E/A	a.b	0.0 ... 2047.7	-	-	-	-	-	-	-	J	-	Setze ...		
				-	-	-	-	-	-	-	J	-	Eingang/Ausgang auf "1"	1/2	
	M	a.b	0.0 ... 8191.7	-	-	-	-	0	J	-	0	Merker auf "1"	1/2		
	L	a.b	parametrierbar											Lokaldatenbit auf "1"	2
	DBX	a.b	0.0 ... 65535.7											Datenbit auf "1"	2
	DIX	a.b	0.0 ... 65535.7											Instanz-Datenbit auf "1"	2
	c	[AR1,m]												registerindirekt, bereichsintern (AR1)	2
	c	[AR2,m]												registerindirekt, bereichsintern (AR2)	2
		[AR1,m]												bereichsübergreifend (AR1)	2
		[AR2,m]												bereichsübergreifend (AR2)	2
	Parameter												über Parameter	2	
R	E/A	a.b	0.0 ... 2047.7	-	-	-	-	-	-	-	J	-	Rücksetze ...		
				-	-	-	-	-	-	-	J	-	Eingang/Ausgang auf "0"	1/2	
	M	a.b	0.0 ... 8191.7	-	-	-	-	0	J	-	0	Merker auf "0"	1/2		
	L	a.b	parametrierbar											Lokaldatenbit auf "0"	2
	DBX	a.b	0.0 ... 65535.7											Datenbit auf "0"	2
	DIX	a.b	0.0 ... 65535.7											Instanz-Datenbit auf "0"	2
	h	[AR1,m]												registerindirekt, bereichsintern (AR1)	2
	h	[AR2,m]												registerindirekt, bereichsintern (AR2)	2
	W	[AR1,m]												bereichsübergreifend (AR1)	2
	W	[AR2,m]												bereichsübergreifend (AR2)	2
	Parameter												über Parameter	2	
=	E/A	a.b	0.0 ... 2047.7	-	-	-	-	-	-	-	J	-	Zuweisen ...		
				-	-	-	-	-	-	-	J	-	des VKE an Eingang/Ausgang	1/2	
	M	a.b	0.0 ... 8191.7	-	-	-	-	0	J	-	0	des VKE an Merker	1/2		
	L	a.b	parametrierbar											des VKE an Lokaldatenbit	2
	DBX	a.b	0.0 ... 65535.7											des VKE an Datenbit	2
	DIX	a.b	0.0 ... 65535.7											des VKE an Instanz-Datenbit	2
	c	[AR1,m]												registerindirekt, bereichsintern (AR1)	2
	c	[AR2,m]												registerindirekt, bereichsintern (AR2)	2
		[AR1,m]												bereichsübergreifend (AR1)	2
		[AR2,m]												bereichsübergreifend (AR2)	2
	Parameter												über Parameter	2	

Befehl	Operand	Parameter	Statuswort									Funktionsbezeichnung	Länge in Worten		
			BIE	A1	A0	OV	OS	OR	STA	VKE	/ER				
														: Operation hängt ab von	
														: Operation beeinflusst	

VKE direkt beeinflussende Operationen			Statuswort									Die folgenden Operationen bearbeiten direkt das VKE.	
CLR			BIE	A1	A0	OV	OS	OR	STA	VKE	/ER	Setze VKE auf "0"	1
			-	-	-	-	-	-	-	-	-		
			-	-	-	-	0	0	0	0			
SET			BIE	A1	A0	OV	OS	OR	STA	VKE	/ER	Setze VKE auf "1"	1
			-	-	-	-	-	-	-	-	-		
			-	-	-	-	0	1	1	0			
NOT			BIE	A1	A0	OV	OS	OR	STA	VKE	/ER	Negiere das VKE	1
			-	-	-	-	-	J	-	J	-		
			-	-	-	-	-	1	J	-			
SAVE			BIE	A1	A0	OV	OS	OR	STA	VKE	/ER	Rette das VKE in das BIE-Bit	1
			-	-	-	-	-	-	-	J	-		
			J	-	-	-	-	-	-	-	-		

Befehl	Operand	Parameter	Statuswort									Funktionsbezeichnung	Länge in Worten	
			BIE	A1	A0	OV	OS	OR	STA	VKE	/ER			
													: Operation hängt ab von	
													: Operation beeinflusst	

Sprung-Operationen

Sprung-Operationen			Statuswort									Sprung, abhängig von der Bedingung. Bei 8-Bit-Operanden liegt die Sprungweite zw. (-128...+127), bei 16-Bit-Operanden zwischen (-32768...-129) oder (+128...+32767)		
SPA	MARKE		BIE	A1	A0	OV	OS	OR	STA	VKE	/ER	Sprunge unbedingt	1/2	
			-	-	-	-	-	-	-	-	-			
			-	-	-	-	-	-	-	-	-			
SPB	MARKE		BIE	A1	A0	OV	OS	OR	STA	VKE	/ER	Sprunge bedingt bei VKE="1"	1/2	
SPBN	MARKE		-	-	-	-	-	-	-	J	-	Sprunge bedingt bei VKE="0"	2	
			-	-	-	-	0	1	1	0				
SPBB	MARKE		BIE	A1	A0	OV	OS	OR	STA	VKE	/ER	Sprunge bedingt bei VKE="1"	2	
			-	-	-	-	-	-	-	J	-	Retten des VKE in das BIE-Bit		
SPBNB	MARKE		J	-	-	-	-	0	1	1	0	Sprunge bedingt bei VKE="0"	2	
												Retten des VKE in das BIE-Bit		
SPBI	MARKE		BIE	A1	A0	OV	OS	OR	STA	VKE	/ER	Sprunge bedingt bei BIE="1"	2	
SPBIN	MARKE		J	-	-	-	-	-	-	-	-	Sprunge bedingt bei BIE="0"	2	
			-	-	-	-	0	1	-	0				
SPO	MARKE		BIE	A1	A0	OV	OS	OR	STA	VKE	/ER	Sprunge bedingt bei Überlauf (OV="1")	1/2	
			-	-	-	J	-	-	-	-	-			
			-	-	-	-	-	-	-	-	-			
SPS	MARKE		BIE	A1	A0	OV	OS	OR	STA	VKE	/ER	Sprunge bedingt bei Überlauf speichernd (OS="1")	2	
			-	-	-	-	J	-	-	-	-			
			-	-	-	0	-	-	-	-	-			
SPU	MARKE		BIE	A1	A0	OV	OS	OR	STA	VKE	/ER	Sprunge bei "Unzulässiger Operation" (A1=1 und A0=1)	2	
SPZ	MARKE		-	J	J	-	-	-	-	-	-	Sprung bedingt bei Ergebnis=0 (A1=0 und A0=0)	1/2	
SPP	MARKE		-	-	-	-	-	-	-	-	-	Sprung bedingt bei Ergebnis>0 (A1=1 und A0=0)	1/2	
SPM	MARKE											Sprung bedingt bei Ergebnis<0 (A1=0 und A0=1)	1/2	
SPN	MARKE											Sprung bedingt bei Ergebnis≠0 (A1=1 und A0=0) oder (A1=0) und (A0=1)	1/2	
SPMZ	MARKE											Sprunge bedingt bei Ergebnis≤0 (A1=0 und A0=1) oder (A1=0 und A0=0)	2	
SPPZ	MARKE											Sprunge bedingt bei Ergebnis≥0 (A1=1 und A0=0) oder (A1=0 und A0=0)	2	
SPL	MARKE		BIE	A1	A0	OV	OS	OR	STA	VKE	/ER	Sprungverteiler	2	
			-	-	-	-	-	-	-	-	-	Der Operation folgt eine Liste von Sprungoperationen. Der		
			-	-	-	-	-	-	-	-	-	Operand ist eine Sprungmarke auf die der Liste folgenden		
												Operationen. AKKU1-L enthält die Nr. der Sprung-		
												operation, die ausgeführt werden soll.		
LOOP	MARKE											Dekrementiere AKKU1-L und springe bei AKKU1-L≠0 (Schleifenprogrammierung)	2	

Befehl	Operand	Parameter	Statuswort										Funktionsbezeichnung	Länge in Worten		
			BIE	A1	AD	OV	OS	OR	STA	VKE	/ER					
															: Operation hängt ab von	
															: Operation beeinflusst	

Transfer-Operationen

Transfer-Operationen				Transferieren des Inhalts von AKKU1 in den adressierten Operanden. Das Statuswort wird nicht beeinflusst.		
T	EB	a	0 ... 127		Transferiere Inhalt von AKKU1-LL zum... Eingangsbyte	1/2
	AB	a	0 ... 127		Ausgangsbyte	1/2
	PAB	a	0 ... 1023		Peripherie-Ausgangsbyte	1/2
	MB	a	0 ... 1023		Merkerbyte	1/2
	LB	a	0 ... 1043		Lokaldatenbyte	2
	DBB	a	0 ... 8191		Datenbyte	2
	DIB	a	0 ... 8191		Instanz-Datenbyte	2
	g [AR1,m]				registerindirekt, bereichsintern (AR1)	2
	g [AR2,m]				registerindirekt, bereichsintern (AR2)	2
	B [AR1,m]				bereichsübergreifend (AR1)	2
	B [AR2,m]				bereichsübergreifend (AR2)	2
Parameter				über Parameter	2	
T	EW		0 ... 126		Transferiere Inhalt von AKKU1-L zum... Eingangswort	1/2
	AW		0 ... 126		Ausgangswort	1/2
	PAW		0 ... 1022		Peripherie-Ausgangswort	1/2
	MW		0 ... 1022		Merkerwort	1/2
	LW		0 ... 1042		Lokaldatenwort	2
	DBW		0 ... 8190		Datenwort	2
	DIW		0 ... 8190		Instanz-Datenwort	2
	h [AR1,m]				registerindirekt, bereichsintern (AR1)	2
	h [AR2,m]				registerindirekt, bereichsintern (AR2)	2
	W [AR1,m]				bereichsübergreifend (AR1)	2
	W [AR2,m]				bereichsübergreifend (AR2)	2
Parameter				über Parameter	2	

Befehl	Operand	Parameter	Statuswort									Funktionsbezeichnung	Länge in Worten	
			BIE	A1	A0	OV	OS	OR	STA	VKE	/ER			
													: Operation hängt ab von	
													: Operation beeinflusst	

Transfer-Operationen			Transferieren des Inhalts von AKKU1 in den adressierten Operanden. Das Statuswort wird nicht beeinflusst.		
T	ED	0 ... 124		Transferiere Inhalt von AKKU1 zum...	1/2
	AD	0 ... 124		Eingangsdoppelwort	1/2
	PAD	0 ... 1020		Ausgangsdoppelwort	1/2
	MD	0 ... 1020		Peripherie-Ausgangsdoppelwort	1/2
	LD	0 ... 1040		Merkerdoppelwort	2
	DBD	0 ... 8188		Lokaldatendoppelwort	2
	DID	0 ... 8188		Datendoppelwort	2
				Instanz-Datendoppelwort	2
	i [AR1,m]			registerindirekt, bereichsintern (AR1)	2
	i [AR2,m]			registerindirekt, bereichsintern (AR2)	2
	D [AR1,m]			bereichsübergreifend (AR1)	2
	D [AR2,m]			bereichsübergreifend (AR2)	2
	Parameter			über Parameter	2
Lade- und Transfer-Operationen für Adressregister			Laden eines Doppelwortes aus einem Speicher oder einem Register in AR1 oder AR2.		
LAR1	-			Lade Inhalt aus ...	1
	AR2			AKKU1	1
	DBD a	0 ... 8188		Adressregister 2	1
	DID a	0 ... 8188		Datendoppelwort	2
	m			Instanz-Datendoppelwort	2
	LD a	0 ... 1040		32-Bit-Konstante als Pointer	3
	MD a	0 ... 1020		Lokaldatendoppelwort	2
				Merkerdoppelwort	2
				... in AR1	
LAR2	-			Lade Inhalt aus ...	1
	DBD a	0 ... 8188		AKKU1	1
	DID a	0 ... 8188		Datendoppelwort	2
	m			Instanz-Datendoppelwort	2
	LD a	0 ... 1040		32-Bit-Konstante als Pointer	3
	MD a	0 ... 1020		Lokaldatendoppelwort	2
				Merkerdoppelwort	2
				... in AR2	
TAR1	-			Transferiere Inhalt aus AR1 in ...	1
	AR2			AKKU1	1
	DBD a	0 ... 8188		Adressregister 2	1
	DID a	0 ... 8188		Datendoppelwort	2
	LD a	0 ... 1040		Instanz-Datendoppelwort	2
	MD a	0 ... 1020		Lokaldatendoppelwort	2
				Merkerdoppelwort	2

Befehl	Operand	Parameter	Statuswort										Funktionsbezeichnung	Länge in Worten		
			BIE	A1	A0	OV	OS	OR	STA	VKE	/ER					
															: Operation hängt ab von	
															: Operation beeinflusst	

TAR2	-														Transferiere Inhalt aus AR2 in ... AKKU1	1
	DBD	a	0 ... 8188												Datendoppelwort	2
	DID	a	0 ... 8188												Instanz-Datendoppelwort	2
	LD	a	0 ... 1040												Lokaldatendoppelwort	2
	MD	a	0 ... 1020												Merkerdoppelwort	2
TAR															Tausche die Inhalte von AR1 und AR2	1
Lade- und Transfer-Operationen für das Statuswort			<i>Statuswort</i>													
L	STW														Lade Statuswort in AKKU1	
	-														J J J J J 0 0 J 0	
															- - - - - - - - -	
T	STW														Transferiere AKKU1 (Bits 8 ... 0) in das Statuswort	
	-														- - - - - - - - -	
															J J J J J - - J -	
Lade-Operationen für DB-Nummer und DB-Länge													Laden der Nummer/Länge eines Datenbausteins in AKKU1. Der alte Inhalt von AKKU1 wird in AKKU2 gerettet. Die Anzeigen werden nicht beeinflusst.			
L	DBNO														Lade Nummer des Datenbausteins	1
L	DINO														Lade Nummer des Instanz-Datenbausteins	1
L	DBLG														Lade Länge des Datenbausteins in Byte	1
L	DILG														Lade Länge des Instanz-Datenbausteins in Byte	1
AKKU-Transferoperationen, Inkrementieren, Dekrementieren													Das Statuswort wird nicht beeinflusst.			
TAW	-														Umkehr der Reihenfolge der Bytes im AKKU1-L Aus LL, LH wird LH, LL	1
TAD	-														Umkehr der Reihenfolge der Bytes im AKKU1 Aus LL, LH, HL, HH wird HH, HL, LH, LL	1
TAK	-														Tausche Inhalte von AKKU1 und AKKU2	1
PUSH	-														Der Inhalt von AKKU1 wird in AKKU2 übertragen	1
POP	-														Der Inhalt von AKKU2 wird in AKKU1 übertragen	1
INC	0 ... 255														Inkrementiere AKKU1-LL	1
DEC	0 ... 255														Dekrementiere AKKU1-LL	1

Befehl	Operand	Parameter	Statuswort										Funktionsbezeichnung	Länge in Worten		
			BIE	A1	A0	OV	OS	OR	STA	VKE	/ER					
															: Operation hängt ab von	
															: Operation beeinflusst	

Umwandlungs-Operationen

Datentyp- Umwandlungsoperationen		Statuswort										Die Ergebnisse der Wandlung stehen im AKKU1. Bei der Wandlung von Realzahlen ist die Ausführungszeit abhängig vom Wert.	
BTI	-		BIE	A1	A0	OV	OS	OR	STA	VKE	/ER	Konvertiere AKKU1 von BCD nach Ganzzahl (16Bit) (BCD To Int.)	1
BTD	-		-	-	-	-	-	-	-	-	-	Konvertiere AKKU1 von BCD nach Ganzzahl (32Bit) (BCD To Doubleint.)	1
DTR	-											Konvertiere AKKU1 von Ganzzahl (32Bit) nach Realzahl (32Bit) (Doubleint. To Real)	1
ITD	-											Konvertiere AKKU1 von Ganzzahl (16Bit) nach Ganzzahl (32Bit) (Int. To Doubleint.)	1
ITB	-		BIE	A1	A0	OV	OS	OR	STA	VKE	/ER	Konvertiere AKKU1 von Ganzzahl (16Bit) nach BCD 0 ... +/-999 (Int. To BCD)	1
DTB	-		-	-	-	J	J	-	-	-	-	Konvertiere AKKU1 von Ganzzahl (32Bit) nach BCD 0 ... +/-9 999 999 (Doubleint. To BCD)	1
RND	-		BIE	A1	A0	OV	OS	OR	STA	VKE	/ER	Wandle Realzahl in 32-Bit-Ganzzahl um	1
RND-	-		-	-	-	-	-	-	-	-	-	Wandle Realzahl in 32-Bit-Ganzzahl um	1
RND+	-		-	-	-	J	J	-	-	-	-	Es wird abgerundet zur nächsten ganzen Zahl	1
TRUNC	-											Wandle Realzahl in 32-Bit-Integerzahl um Es wird aufgerundet zur nächsten ganzen Zahl Wandle Realzahl in 32-Bit-Integerzahl um Es werden die Nachkommastellen abgeschnitten	1
Komplementbildung		<i>Statuswort</i>											
INVI	-		BIE	A1	A0	OV	OS	OR	STA	VKE	/ER	Bilde 1er-Komplement von AKKU1-L	1
INVD	-		-	-	-	-	-	-	-	-	-	Bilde 1er-Komplement von AKKU1	1
NEGI	-		BIE	A1	A0	OV	OS	OR	STA	VKE	/ER	Bilde 2er-Komplement von AKKU1-L (Integerzahl)	1
NEGD	-		-	-	-	-	-	-	-	-	-	Bilde 2er-Komplement von AKKU1 (Double-Integerzahl)	1
			-	J	J	J	J	-	-	-	-		

Befehl	Operand	Parameter	Statuswort										Funktionsbezeichnung	Länge in Worten		
			BIE	A1	A0	OV	OS	OR	STA	VKE	/ER					
															: Operation hängt ab von	
															: Operation beeinflusst	

Vergleichs-Operationen

Vergleichs-Operationen mit Ganzzahl (16Bit)			Statuswort										Vergleich der Ganzzahl (16Bit) in AKKU1-L und AKKU2-L. VKE=1, wenn Bedingung erfüllt.	
==I	-		BIE	A1	A0	OV	OS	OR	STA	VKE	/ER	AKKU2-L=AKKU1-L	1	
<>I	-		-	-	-	-	-	-	-	-	-	AKKU2-L≠AKKU1-L	1	
<I	-		-	J	J	0	-	0	J	J	1	AKKU2-L<AKKU1-L	1	
<=I	-											AKKU2-L<=AKKU1-L	1	
>I	-											AKKU2-L>AKKU1-L	1	
>=I	-											AKKU2-L>=AKKU1-L	1	
Vergleichs-Operationen mit Ganzzahl (32Bit)			Statuswort										Vergleich der Ganzzahl (32Bit) in AKKU1 und AKKU2. VKE=1, wenn Bedingung erfüllt.	
==D	-		BIE	A1	A0	OV	OS	OR	STA	VKE	/ER	AKKU2=AKKU1	1	
<>D	-		-	-	-	-	-	-	-	-	-	AKKU2≠AKKU1	1	
<D	-		-	J	J	0	-	0	J	J	1	AKKU2<AKKU1	1	
<=D	-											AKKU2<=AKKU1	1	
>D	-											AKKU2>AKKU1	1	
>=D	-											AKKU2>=AKKU1	1	
Vergleichs-Operationen mit (32-Bit-Realzahl)			Statuswort										Vergleich der 32-Bit-Realzahlen in AKKU1 und AKKU2. VKE=1, wenn Bedingung erfüllt. Die Ausführungszeit der Operation hängt vom Wert ab, der verglichen werden soll.	
==R	-		BIE	A1	A0	OV	OS	OR	STA	VKE	/ER	AKKU2=AKKU1	1	
<>R	-		-	-	-	-	-	-	-	-	-	AKKU2≠AKKU1	1	
<R	-		-	J	J	J	J	0	J	J	1	AKKU2<AKKU1	1	
<=R	-											AKKU2<=AKKU1	1	
>R	-											AKKU2>AKKU1	1	
>=R	-											AKKU2>=AKKU1	1	

Befehl	Operand	Parameter	Statuswort										Funktionsbezeichnung	Länge in Worten		
			BIE	A1	A0	OV	OS	OR	STA	VKE	/ER					
															: Operation hängt ab von	
															: Operation beeinflusst	

Verknüpfungs-Operationen (Bit)

Verknüpfungs-Operationen mit Bitoperanden			Statuswort										Abfrage des adressierten Operanden auf seinen Signalzustand und Verknüpfung des Ergebnisses mit dem VKE nach der entsprechenden Funktion.		
U	E/A	a.b	0.0 ... 127.7	BIE	A1	A0	OV	OS	OR	STA	VKE	/ER	UND-Verknüpfung auf Signalzustand "1"		
				-	-	-	-	-	J	-	J	J	Eingang/Ausgang	1/2	
	M	a.b	0.0 ... 1023.7	-	-	-	-	-	J	J	J	1	Merker	1/2	
	L	a.b	0.0 ... 1043.7											Lokaldatenbit	2
	DBX	a.b	0.0 ... 8191.7											Datenbit	2
	DIX	a.b	0.0 ... 8191.7											Instanzen-Datenbit	2
	c [AR1,m]													registerindirekt, bereichsintern (AR1)	2
c [AR2,m]													registerindirekt, bereichsintern (AR2)	2	
[AR1,m]													bereichsübergreifend (AR1)	2	
[AR2,m]													bereichsübergreifend (AR2)	2	
Parameter													über Parameter	2	
UN	E/A	a.b	0.0 ... 127.7	BIE	A1	A0	OV	OS	OR	STA	VKE	/ER	UND-Verknüpfung auf Signalzustand "0"		
				-	-	-	-	-	J	-	J	J	Eingang/Ausgang	1/2	
	M	a.b	0.0 ... 1023.7	-	-	-	-	-	J	J	J	1	Merker	1/2	
	L	a.b	0.0 ... 1043.7											Lokaldatenbit	2
	DBX	a.b	0.0 ... 8191.7											Datenbit	2
	DIX	a.b	0.0 ... 8191.7											Instanzen-Datenbit	2
	c [AR1,m]													registerindirekt, bereichsintern (AR1)	2
c [AR2,m]													registerindirekt, bereichsintern (AR2)	2	
[AR1,m]													bereichsübergreifend (AR1)	2	
[AR2,m]													bereichsübergreifend (AR2)	2	
Parameter													über Parameter	2	
O	E/A	a.b	0.0 ... 127.7	BIE	A1	A0	OV	OS	OR	STA	VKE	/ER	ODER-Verknüpfung auf Signalzustand "1"		
				-	-	-	-	-	-	-	J	J	Eingang/Ausgang	1/2	
	M	a.b	0.0 ... 1023.7	-	-	-	-	-	0	J	J	1	Merker	1/2	
	L	a.b	0.0 ... 1043.7											Lokaldatenbit	2
	DBX	a.b	0.0 ... 8191.7											Datenbit	2
	DIX	a.b	0.0 ... 8191.7											Instanzen-Datenbit	2
	c [AR1,m]													registerindirekt, bereichsintern (AR1)	2
c [AR2,m]													registerindirekt, bereichsintern (AR2)	2	
[AR1,m]													bereichsübergreifend (AR1)	2	
[AR2,m]													bereichsübergreifend (AR2)	2	
Parameter													über Parameter	2	

Befehl	Operand	Parameter	Statuswort										Funktionsbezeichnung	Länge in Worten		
			BIE	A1	A0	OV	OS	OR	STA	VKE	/ER					
															: Operation hängt ab von	
															: Operation beeinflusst	

ON	E/A	a.b	0.0 ... 127.7	-	-	-	-	-	-	-	-	J	J	ODER-Verknüpfung auf Signalzustand "0"	Eingang/Ausgang	1/2
	M	a.b	0.0 ... 1023.7	-	-	-	-	-	0	J	J	J	1	Merker		1/2
	L	a.b	0.0 ... 1043.7											Lokaldatenbit		2
	DBX	a.b	0.0 ... 8191.7											Datenbit		2
	DIX	a.b	0.0 ... 8191.7											Instanz-Datenbit		2
	c [AR1,m]													registerindirekt, bereichsintern (AR1)		2
	c [AR2,m]													registerindirekt, bereichsintern (AR2)		2
[AR1,m]													bereichsübergreifend (AR1)		2	
[AR2,m]													bereichsübergreifend (AR2)		2	
Parameter													über Parameter		2	
X	E/A	a.b	0.0 ... 127.7	-	-	-	-	-	-	-	-	J	J	EXKLUSIV-ODER-Verknüpfung auf Signalzustand "1"	Eingang/Ausgang	2
	M	a.b	0.0 ... 1023.7	-	-	-	-	-	0	J	J	J	1	Merker		2
	L	a.b	0.0 ... 1043.7											Lokaldatenbit		2
	DBX	a.b	0.0 ... 8191.7											Datenbit		2
	DIX	a.b	0.0 ... 8191.7											Instanz-Datenbit		2
	c [AR1,m]													registerindirekt, bereichsintern (AR1)		2
	c [AR2,m]													registerindirekt, bereichsintern (AR2)		2
[AR1,m]													bereichsübergreifend (AR1)		2	
[AR2,m]													bereichsübergreifend (AR2)		2	
Parameter													über Parameter		2	
XN	E/A	a.b	0.0 ... 127.7	-	-	-	-	-	-	-	-	J	J	EXKLUSIV-ODER-Verknüpfung auf Signalzustand "0"	Eingang/Ausgang	2
	M	a.b	0.0 ... 1023.7	-	-	-	-	-	0	J	J	J	1	Merker		2
	L	a.b	0.0 ... 1043.7											Lokaldatenbit		2
	DBX	a.b	0.0 ... 8191.7											Datenbit		2
	DIX	a.b	0.0 ... 8191.7											Instanz-Datenbit		2
	c [AR1,m]													registerindirekt, bereichsintern (AR1)		2
	c [AR2,m]													registerindirekt, bereichsintern (AR2)		2
[AR1,m]													bereichsübergreifend (AR1)		2	
[AR2,m]													bereichsübergreifend (AR2)		2	
Parameter													über Parameter		2	

Befehl	Operand	Parameter	Statuswort										Funktionsbezeichnung	Länge in Worten		
			BIE	A1	A0	OV	OS	OR	STA	VKE	/ER					
															: Operation hängt ab von	
															: Operation beeinflusst	

Verknüpfungs-Operationen von Klammersausdrücken			Statuswort													
															Retten der Bits BIE, VKE, OR und einer Funktionskennung (U, UN, ...) auf den Klammerstack. Pro Baustein sind 7 Klammerebenen möglich.	
U(BIE	A1	A0	OV	OS	OR	STA	VKE	/ER				UND-Klammer auf	1
UN(J	-	-	-	-	J	-	J	J				UND-NICHT-Klammer auf	1
O(-	-	-	-	-	0	1	-	0				ODER-Klammer auf	1
ON(ODER-NICHT-Klammer auf	1		
X(EXCLUSIV-ODER-Klammer auf	1		
XN(EXKLUSIV-ODER-NICHT-Klammer auf	1		
)			BIE	A1	A0	OV	OS	OR	STA	VKE	/ER				Klammer zu	1
			-	-	-	-	-	-	-	J	-				Entfernen eines Eintrags vom Klammerstack	
			J	-	-	-	-	J	1	J	1				Verknüpfen des VKE mit dem aktuellen VKE im Prozessor	
ODER-Verknüpfung von UND-Funktionen			Statuswort													
															Es erfolgt die ODER-Verknüpfung von UND-Funktionen nach der Regel: UND vor ODER	
O			BIE	A1	A0	OV	OS	OR	STA	VKE	/ER				ODER-Verknüpfung von UND-Funktionen nach der Regel: UND vor ODER	1
			-	-	-	-	-	J	-	J	J					
			-	-	-	-	-	J	1	-	J					

Befehl	Operand	Parameter	Statuswort										Funktionsbezeichnung	Länge in Worten		
			BIE	A1	A0	OV	OS	OR	STA	VKE	/ER					
															: Operation hängt ab von	
															: Operation beeinflusst	

Verknüpfungs-Operationen mit Timern und Zählern			Statuswort										Abfrage des adressierten Timers/Zählers auf seinen Signalzustand und Verknüpfen des Ergebnisses mit dem VKE nach der entspr. Funktion	
U	T f	0 bis 255	BIE	A1	A0	OV	OS	OR	STA	VKE	/ER	UND-Verknüpfung auf Signalzustand		
			-	-	-	-	-	J	-	J	J	Timer	1/2	
	Z f	0 bis 255	-	-	-	-	-	J	J	J	1	Zähler	1/2	
	Timerpara. Zählerpara.											Timer über Parameter adressiert Zähler über Parameter adressiert	2	
UN	T f	0 bis 255	BIE	A1	A0	OV	OS	OR	STA	VKE	/ER	UND-Verknüpfung auf Signalzustand		
			-	-	-	-	-	J	-	J	J	Timer	1/2	
	Z f	0 bis 255	-	-	-	-	-	J	J	J	1	Zähler	1/2	
	Timerpara. Zählerpara.											Timer über Parameter adressiert Zähler über Parameter adressiert	2	
O	T f	0 bis 255	BIE	A1	A0	OV	OS	OR	STA	VKE	/ER	ODER-Verknüpfung auf Signalzustand		
			-	-	-	-	-	-	-	J	J	Timer	1/2	
	Z f	0 bis 255	-	-	-	-	-	0	J	J	1	Zähler	1/2	
	Timerpara. Zählerpara.											Timer über Parameter adressiert Zähler über Parameter adressiert	2	
ON	T f	0 bis 255	BIE	A1	A0	OV	OS	OR	STA	VKE	/ER	ODER-Verknüpfung auf Signalzustand		
			-	-	-	-	-	-	-	J	J	Timer	1/2	
	Z f	0 bis 255	-	-	-	-	-	0	J	J	1	Zähler	1/2	
	Timerpara. Zählerpara.											Timer über Parameter adressiert Zähler über Parameter adressiert	2	
X	T f	0 bis 255	BIE	A1	A0	OV	OS	OR	STA	VKE	/ER	EXKLUSIV-ODER-Verknüpfung auf Signalzustand		
			-	-	-	-	-	-	-	J	J	Timer	2	
	Z f	0 bis 255	-	-	-	-	-	0	J	J	1	Zähler	2	
	Timerpara. Zählerpara.											Timer über Parameter adressiert Zähler über Parameter adressiert	2	
XN	T f	0 bis 255	BIE	A1	A0	OV	OS	OR	STA	VKE	/ER	EXKLUSIV-ODER-Verknüpfung auf Signalzustand		
			-	-	-	-	-	-	-	J	J	Timer	2	
	Z f	0 bis 255	-	-	-	-	-	0	J	J	1	Zähler	2	
	Timerpara. Zählerpara.											Timer über Parameter adressiert Zähler über Parameter adressiert	2	

Befehl	Operand	Parameter	Statuswort										Funktionsbezeichnung	Länge in Worten		
			BIE	A1	A0	OV	OS	OR	STA	VKE	/ER					
															: Operation hängt ab von	
															: Operation beeinflusst	

Verknüpfungs-Operationen mit Anzeigenbits			Statuswort										Abfrage der angegebenen Bedingungen auf ihren Signalzustand und Verknüpfen des Ergebnisses mit dem VKE nach der entspr. Funktion	
U	==0 >0 <0 <>0 <=0 >=0 UO OS BIE OV		BIE	A1	A0	OV	OS	OR	STA	VKE	/ER	UND-Verknüpfung auf Signalzustand "1"		
			J	J	J	J	J	J	-	J	J	Ergebnis=0 (A1=0) and (A0=0)	1	
			-	-	-	-	-	J	J	J	1	Ergebnis>0 (A1=1) and (A0=0)	1	
													Ergebnis<0 (A1=0) and (A0=1)	1
													Ergebnis≠0 ((A1=0) and (A0=1)) or ((A1=1) and (A0=0))	1
													Ergebnis<=0 ((A1=0) and (A0=1)) or ((A1=0) and (A0=0))	1
													Ergebnis>=0 ((A1=1) and (A0=0)) or ((A1=1) and (A0=0))	1
													unordered (A1=1) and (A0=1)	1
													OS=1	1
													BIE=1	1
										OV=1	1			
UN	==0 >0 <0 <>0 <=0 >=0 UO OS BIE OV		BIE	A1	A0	OV	OS	OR	STA	VKE	/ER	UND-Verknüpfung auf Signalzustand "0"		
			J	J	J	J	J	J	-	J	J	Ergebnis=0 (A1=0) and (A0=0)	1	
			-	-	-	-	-	J	J	J	1	Ergebnis>0 (A1=1) and (A0=0)	1	
													Ergebnis<0 (A1=0) and (A0=1)	1
													Ergebnis≠0 ((A1=0) and (A0=1)) or ((A1=1) and (A0=0))	1
													Ergebnis<=0 ((A1=0) and (A0=1)) or ((A1=0) and (A0=0))	1
													Ergebnis>=0 ((A1=1) and (A0=0)) or ((A1=1) and (A0=0))	1
													unordered (A1=1) and (A0=1)	1
													OS=0	1
													BIE=0	1
										OV=0	1			
O	==0 >0 <0 <>0 <=0 >=0 UO OS BIE OV		BIE	A1	A0	OV	OS	OR	STA	VKE	/ER	ODER-Verknüpfung auf Signalzustand "1"		
			J	J	J	J	J	-	-	J	J	Ergebnis=0 (A1=0) and (A0=0)	1	
			-	-	-	-	-	0	J	J	1	Ergebnis>0 (A1=1) and (A0=0)	1	
													Ergebnis<0 (A1=0) and (A0=1)	1
													Ergebnis≠0 ((A1=0) and (A0=1)) or ((A1=1) and (A0=0))	1
													Ergebnis<=0 ((A1=0) and (A0=1)) or ((A1=0) and (A0=0))	1
													Ergebnis>=0 ((A1=1) and (A0=0)) or ((A1=1) and (A0=0))	1
													unordered (A1=1) and (A0=1)	1
													OS=1	1
													BIE=1	1
										OV=1	1			

Befehl	Operand	Parameter	Statuswort										Funktionsbezeichnung	Länge in Worten		
			BIE	A1	A0	OV	OS	OR	STA	VKE	/ER					
															: Operation hängt ab von	
															: Operation beeinflusst	

ON	==0		J	J	J	J	J	-	-	J	J	ODER-Verknüpfung auf Signalzustand "0"	Ergebnis=0 (A1=0) and (A0=0)	1
	>0		-	-	-	-	-	0	J	J	1	Ergebnis>0 (A1=1) and (A0=0)	1	
	<0											Ergebnis<0 (A1=0) and (A0=1)	1	
	<>0											Ergebnis≠0 ((A1=0) and (A0=1)) or ((A1=1) and (A0=0))	1	
	<=0											Ergebnis<=0 ((A1=0) and (A0=1)) or ((A1=0) and (A0=0))	1	
	>=0											Ergebnis>=0 ((A1=1) and (A0=0)) or ((A1=1) and (A0=0))	1	
	UO											unordered (A1=1) and (A0=1)	1	
	OS											OS=0	1	
	BIE											BIE=0	1	
	OV											OV=0	1	
X	==0		J	J	J	J	J	-	-	J	J	EXKLUSIV-ODER-Verknüpfung auf Signalzustand "1"	Ergebnis=0 (A1=0) and (A0=0)	1
	>0		-	-	-	-	-	0	J	J	1	Ergebnis>0 (A1=1) and (A0=0)	1	
	<0											Ergebnis<0 (A1=0) and (A0=1)	1	
	<>0											Ergebnis≠0 ((A1=0) and (A0=1)) or ((A1=1) and (A0=0))	1	
	<=0											Ergebnis<=0 ((A1=0) and (A0=1)) or ((A1=0) and (A0=0))	1	
	>=0											Ergebnis>=0 ((A1=1) and (A0=0)) or ((A1=1) and (A0=0))	1	
	UO											unordered (A1=1) and (A0=1)	1	
	OS											OS=1	1	
	BIE											BIE=1	1	
	OV											OV=1	1	
XN	==0		J	J	J	J	J	-	-	J	J	EXKLUSIV-ODER-Verknüpfung auf Signalzustand "0"	Ergebnis=0 (A1=0) and (A0=0)	1
	>0		-	-	-	-	-	0	J	J	1	Ergebnis>0 (A1=1) and (A0=0)	1	
	<0											Ergebnis<0 (A1=0) and (A0=1)	1	
	<>0											Ergebnis≠0 ((A1=0) and (A0=1)) or ((A1=1) and (A0=0))	1	
	<=0											Ergebnis<=0 ((A1=0) and (A0=1)) or ((A1=0) and (A0=0))	1	
	>=0											Ergebnis>=0 ((A1=1) and (A0=0)) or ((A1=1) and (A0=0))	1	
	UO											unordered (A1=1) and (A0=1)	1	
	OS											OS=0	1	
	BIE											BIE=0	1	
	OV											OV=0	1	

Befehl	Operand	Parameter	Statuswort										Funktionsbezeichnung	Länge in Worten		
			BIE	A1	A0	OV	OS	OR	STA	VKE	/ER					
															: Operation hängt ab von	
															: Operation beeinflusst	

Verknüpfungs-Operationen (Wort)

Verknüpfungs-Operationen mit dem Inhalt von AKKU1			Statuswort												
			BIE	A1	A0	OV	OS	OR	STA	VKE	/ER				
UW														UND AKKU2-L	1
UW	k16		-	-	-	-	-	-	-	-	-	-	-	UND 16-Bit-Konstante	2
OW			-	J	0	0	-	-	-	-	-	-	-	ODER AKKU2-L	1
OW	k16													ODER 16-Bit-Konstante	2
XOW														EXKLUSIV ODER AKKU2-L	1
XOW	k16													EXKLUSIV ODER 16-Bit-Konstante	2
UD														UND AKKU2	1
UD	k32													UND 32-Bit-Konstante	3
OD														ODER AKKU2	1
OD	k32													ODER 32-Bit-Konstante	3
XOD														EXKLUSIV ODER AKKU2	1
XOD	k32													EXKLUSIV ODER 32-Bit-Konstante	3

Zeit-Operationen

Zeit-Operationen			Statuswort												
			BIE	A1	A0	OV	OS	OR	STA	VKE	/ER				
SI	T f Timerpara.	0 ... 255												Starte Zeit als Impuls bei Flankenwechsel von "0" nach "1"	1/2
			-	-	-	-	-	-	-	J	-	-	-	"1"	2
SV	T f Timerpara.	0 ... 255	-	-	-	-	-	0	-	-	-	0	-	Starte Zeit als verlängerten Impuls bei Flankenwechsel von "0" nach "1"	1/2
														2	
SE	T f Timerpara.	0 ... 255												Starte Zeit als Einschaltverzögerung bei Flankenwechsel von "0" nach "1"	1/2
														2	
SS	T f Timerpara.	0 ... 255												Starte Zeit als speichernde Einschaltverzögerung bei Flankenwechsel von "0" nach "1"	1/2
														2	
SA	T f Timerpara.	0 ... 255												Starte Zeit als Ausschaltverzögerung bei Flankenwechsel "1" nach "0"	1/2
														2	
FR	T f Timerpara.	0 ... 255												Freigabe eines Timers für das erneute Starten bei Flankenwechsel (Löschen des Flankenmerkers für das Starten der Zeit)	1/2
														2	
R	T f Timerpara.	0 ... 255												Rücksetzen einer Zeit	1/2
														2	

Befehl	Operand	Parameter	Statuswort										Funktionsbezeichnung	Länge in Worten		
			BIE	A1	A0	OV	OS	OR	STA	VKE	/ER					
															: Operation hängt ab von	
															: Operation beeinflusst	

Zähl-Operationen

Zähl-Operationen			Statuswort										Der Zählwert steht im AKKU1-L bzw. in der als Parameter übergebenen Adresse.		
S	Z f Zählerpara.	0 ... 255	BIE	A1	A0	OV	OS	OR	STA	VKE	/ER			Vorbelegen eines Zählers bei Flankenwechsel von "0"	1/2
			-	-	-	-	-	-	-	J	-			nach "1"	2
R	Z f Zählerpara.	0 ... 255	-	-	-	-	-	0	-	-	0			Rücksetzen des Zählers auf "0" bei Flankenwechsel von	1/2
														"0" nach "1"	2
ZV	Z f Zählerpara.	0 ... 255												Zähle um 1 vorwärts bei Flankenwechsel von "0" nach "1"	1/2
															2
ZR	Z f Zählerpara.	0 ... 255												Zähle um 1 rückwärts bei Flankenwechsel von "0" nach "1"	1/2
															2
FR	Z f Zählerpara.	0 ... 255												Freigabe eines Zählers bei Flankenwechsel "0" nach "1"	1/2
														(Löschen des Flankenmerkers für Vorwärts- und Rückwärtszählen eines Zählers)	2

Teil 2 Organisationsbausteine

Überblick Hier finden Sie eine Beschreibung der integrierten Organisationsbausteine der VIPA-Standard-CPU's der Systeme 100V, 200V, 300V und 500V.

Inhalt	Thema	Seite
	Teil 2 Organisationsbausteine	2-1
	Übersicht.....	2-2
	OB 1 - Zyklische Programmbearbeitung.....	2-3
	OB 10 - Uhrzeitalarm.....	2-5
	OB 20 - Verzögerungsalarm.....	2-7
	OB 35 - Weckalarm.....	2-8
	OB 40 - Prozessalarm.....	2-9
	OB 80 - Zeitfehler.....	2-11
	OB 81 - Stromversorgungsfehler.....	2-14
	OB 82 - Diagnosealarm.....	2-15
	OB 85 - Programmablauffehler.....	2-17
	OB 86 - Slaveausfall / -wiederkehr.....	2-21
	OB 100 - Neustart.....	2-23
	OB 121 - Programmierfehler (Synchronfehler).....	2-25
	OB 122 - Peripheriezugriffsfehler.....	2-28

Übersicht

Allgemeines

OBs (**O**rganisations**b**austeine) bilden die Schnittstelle zwischen Betriebssystem und Anwenderprogramm. Das Hauptprogramm ist immer in OB 1 abzulegen. Die anderen OBs haben entsprechend den Aufrufereignissen festgelegte Nummern. OBs werden entsprechend ihrer zugeordneten Priorität bearbeitet.

Das Betriebssystem der CPU ruft die OBs bei folgenden Ereignissen auf:

- Anlauf der CPU
- Zyklisch oder zeitlich getaktete Programmfolge
- Auftreten von Fehlern
- Auftreten von Prozessalarmen

Integrierte OBs

Folgende Organisationsbausteine (OBs) stehen zur Verfügung:

OB	Beschreibung
OB 1	Zyklische Programmbearbeitung
OB 10	Uhrzeitalarm
OB 20	Verzögerungsalarm
OB 35	Weckalarm
OB 40	Prozessalarm
OB 80	Zeitfehler (Zykluszeitüberschreitung oder Uhrzeitalarm abgelaufen)
OB 81	Stromversorgungsfehler
OB 82	Diagnosealarm
OB 85	Programmablauffehler (OB nicht vorhanden oder Peripheriezugriffsfehler beim Prozessabbild aktualisieren)
OB 86	Slaveausfall/-wiederkehr
OB 100	Neustart
OB 121	Programmierfehler (Synchrone Fehler)
OB 122	Peripheriezugriffsfehler

OB 1 - Zyklische Programmbearbeitung

Beschreibung	<p>Das Betriebssystem der CPU bearbeitet den OB 1 zyklisch. Nach dem Übergang von ANLAUF nach RUN startet die zyklische Bearbeitung des OB 1. Der OB 1 hat die niedrigste Priorität (Priorität 1) aller laufzeit-überwachten OBs.</p> <p>Innerhalb des OB 1 haben Sie die Möglichkeit Funktionsbausteine und Funktionen aufzurufen.</p>
Funktionsweise	<p>Wenn die Bearbeitung des OB 1 beendet ist, sendet das Betriebssystem Globaldaten. Bevor der OB 1 neu gestartet wird, schreibt das Betriebssystem das Prozessabbild der Ausgänge in die Ausgabe-Module, aktualisiert das Prozessabbild der Eingänge und empfängt Globaldaten für die CPU.</p>
Zykluszeit	<p>Die Zeit, die für die gesamte Bearbeitung des OB 1 erforderlich ist, nennt man <i>Zykluszeit</i>. Hierin eingeschlossen sind beispielsweise die Bearbeitungszeiten für höhere Prioritätsklassen, die das Hauptprogramm unterbrechen bzw. Kommunikationsprozesse des Betriebssystems. Hierzu zählen auch Systemsteuerung der zyklischen Bearbeitung, Aktualisierung des Prozessabbilds und Aktualisierung der Zeitfunktionen.</p> <p>Sofern Sie mit der CPU online sind, haben Sie die Möglichkeit die aktuelle Zykluszeit über den Siemens SIMATIC Manager zu ermitteln. Über Zielsystem > <i>Baugruppenzustand</i> > <i>Zykluszeit</i> können Sie die minimale, maximale und aktuelle Zykluszeit ausgeben lassen.</p>
Zykluszeit-überwachung	<p>Die CPU bietet eine Überwachung der <i>maximalen Zykluszeit</i>. Der Wert für die <i>maximale Zykluszeit</i> ist auf 150ms voreingestellt. Sie können diesen Wert umparametrieren oder aber mit dem SFC 43 (RE_TRIGR) an jeder Stelle Ihres Programms die Zeitüberwachung neu starten. Überschreitet Ihr Programm die <i>maximale Zykluszeit</i> für den OB 1, dann ruft das Betriebssystem den OB 80 (Zykluszeitüberschreitung) auf. Ist der OB 80 nicht programmiert, dann geht die CPU in den Betriebszustand STOP.</p> <p>Neben der Überwachung der <i>maximalen Zykluszeit</i> kann die Einhaltung einer <i>minimalen Zykluszeit</i> garantiert werden. Hierbei verzögert das Betriebssystem den Beginn eines neuen Zyklus (Schreiben des Prozessabbilds der Ausgänge), solange, bis die <i>minimale Zykluszeit</i> erreicht ist.</p>
Zugriff auf die Lokaldaten	<p>Das Betriebssystem der CPU übergibt dem OB 1, wie jedem OB, in den ersten 20Byte der temporären Lokaldaten eine Startinformation.</p> <p>Mit der Systemfunktion SFC 6 RD_SINFO können Sie auf diese Startinformation zugreifen. Bitte beachten Sie hierbei, dass Sie die Startinformationen eines OBs nur im OB selbst lesen können, da es sich hier um temporäre Daten handelt.</p> <p>Näheres hierzu finden Sie im Teil "Integrierte Standard SFCs".</p>

Lokaldaten

Die folgende Tabelle zeigt die Belegung der Startinformationen für den OB 1, die defaultmäßige symbolische Bezeichnung und die Datentypen:

Variable	Datentyp	Beschreibung
OB1_EV_CLASS	BYTE	Ereignisklasse und Kennung: 11h: OB 1 ist aktiv
OB1_SCAN_1	BYTE	01h: Abschluss des Neustart 03h: Abschluss des freien Zyklus
OB1_PRIORITY	BYTE	Prioritätsklasse: 1
OB1_OB_NUMBR	BYTE	OB-NR. (01)
OB1_RESERVED_1	BYTE	reserviert
OB1_RESERVED_2	BYTE	reserviert
OB1_PREV_CYCLE	INT	Laufzeit des vorherigen Zyklus (ms)
OB1_MIN_CYCLE	INT	Minimale Zykluszeit (ms) seit dem letzten Anlauf
OB1_MAX_CYCLE	INT	Maximale Zykluszeit (ms) seit dem letzten Anlauf
OB1_DATE_TIME	DATE_AND_TIME	Datum und Uhrzeit, zu denen der OB angefordert wurde

OB 10 - Uhrzeitalarm

Beschreibung Sie verwenden den Uhrzeitalarm, wenn Sie ein Programm zu einer bestimmten Uhrzeit einmalig oder periodisch bearbeiten lassen möchten. Sie können den Uhrzeitalarm in der Hardware-Konfiguration projektieren oder mit Systemfunktionen vom Programm aus zur Laufzeit steuern. Voraussetzung für die ordnungsgemäße Bearbeitung des Uhrzeitalarms ist eine richtig eingestellte Echtzeituhr auf der CPU.

Bei der Ausführung haben Sie folgende Auswahlmöglichkeiten:

- einmalig
- minütlich
- stündlich
- täglich
- wöchentlich
- monatlich
- jährlich
- am Monatsende



Hinweis!

Bei monatlicher Bearbeitung des Uhrzeitalarm-OBs sind beim Startdatum nur die Tage 1, 2, ... 28 möglich.

Aktivierung Um den Uhrzeitalarm zu starten, müssen Sie den Alarm erst einstellen und dann aktivieren. Hierzu gibt es folgende Möglichkeiten:

- Die Aktivierung des Uhrzeitalarms geschieht über die Hardware-Konfiguration. Sie öffnen die CPU mit **Bearbeiten** > *Objekteigenschaften* > *Uhrzeitalarme*. Hier können Sie den entsprechenden Uhrzeitalarm einstellen und aktivieren. Nach Übertragung in die CPU und dem Übergang von Anlauf in RUN wird die Uhrzeitalarmüberwachung automatisch gestartet.
- Sie stellen den Uhrzeitalarm wie oben gezeigt über die Hardware-Konfiguration und aktivieren ihn durch Aufrufen des SFC 30 ACT_TINT aus Ihrem Anwenderprogramm heraus.
- Sie stellen den Uhrzeitalarm durch Aufruf des SFC 28 SET_TINT und aktivieren ihn durch Aufruf des SFC 30 ACT_TINT.

Sie können den Uhrzeitalarm mit Hilfe des SFC 41 DIS_AIRT verzögern bzw. die Verzögerung mit dem SFC 42 EN_AIRT aufheben.

Fehlerverhalten Fehlt zum Zeitpunkt seines Aufrufs der Uhrzeitalarm-OB im Anwenderprogramm ruft das Betriebssystem den OB 85 auf. Ist dieser nicht vorhanden geht die CPU in STOP. Kommt es zu einem Fehler bei der Uhrzeitalarmbearbeitung z.B. Startzeitpunkt liegt in der Vergangenheit, wird der Zeitfehler-OB OB 80 aufgerufen und danach der Uhrzeitalarm-OB einmalig bearbeitet.

Möglichkeiten bei der Aktivierung

Die folgende Tabelle zeigt die prinzipiellen Möglichkeiten beim Aktivieren eines Uhrzeitalarms:

Intervall	Beschreibung
Nicht aktiviert	Der Uhrzeitalarm-OB wird nicht bearbeitet. Sie können den Uhrzeitalarm durch Aufruf des SFC 30 aktivieren.
Einmalig aktiviert	Der Uhrzeitalarm-OB wird automatisch storniert, nachdem er einmal wie angegeben bearbeitet wurde. Ihr Programm kann mit dem SFC 28 den Uhrzeitalarm erneut stellen und ihn mit dem SFC 30 wieder aktivieren.
Periodisch aktiviert	Die CPU ermittelt zum Zeitpunkt des Uhrzeitalarms aus der aktuellen Uhrzeit und der Periode den nächsten Startzeitpunkt des Uhrzeitalarms.

Lokaldaten des Uhrzeitalarm-OBs

Nachfolgend sehen Sie die Belegung der Startinformationen für OB 10, die defaultmäßige symbolische Bezeichnung und die Datentypen:

Variable	Datentyp	Beschreibung
OB10_EV_CLASS	BYTE	Ereignisklasse und Kennung: 11h: Alarm ist aktiv
OB10_STRT_INFO	BYTE	11h: Startanforderung für OB 10
OB10_PRIORITY	BYTE	Parametrierte Prioritätsklasse: Defaultwert 2
OB10_OB_NUMBR	BYTE	OB-NR. (10)
OB10_RESERVED_1	BYTE	reserviert
OB10_RESERVED_2	BYTE	reserviert
OB10_PERIOD_EXE	WORD	Der OB wird zu dem angegebenen Intervall bearbeitet: 0000h: einmalig 0201h: minütlich 0401h: stündlich 1001h: täglich 1201h: wöchentlich 1401h: monatlich 1801h: jährlich 2001h: am Monatsende
OB10_RESERVED_3	INT	reserviert
OB10_RESERVED_4	INT	reserviert
OB10_DATE_TIME	DATE_AND_TIME	Datum und Uhrzeit, zu denen der OB angefordert wurde

Informationen, wie Sie auf die Lokaldaten zugreifen können, finden Sie in der Beschreibung zum OB 1.

OB 20 - Verzögerungsalarm

Beschreibung Mit einem Verzögerungsalarm haben Sie die Möglichkeit, unabhängig von den Zeitfunktionen, eine Zeitverzögerung zu realisieren. Sie können die Verzögerungsalarmlarmer in der Hardware-Konfiguration projektieren bzw. mit Systemfunktionen vom Programm aus zur Laufzeit steuern.

Aktivierung Für die Aktivierung ist keine Hardware-Konfiguration erforderlich. Sie starten den Verzögerungsalarm durch Aufruf des SFC 32 SRT_DINT und durch Laden des entsprechenden OBs in die CPU. Übergeben Sie hierbei der Funktion OB-Nr., Verzögerungszeit und eine Kennzeichnung. Nach der angegebenen Verzögerungszeit wird vom Betriebssystem der zugehörige OB aufgerufen. Sie haben die Möglichkeit mit dem SFC 33 CAN_DINT einen Verzögerungsalarm, der noch nicht aktiviert ist, zu stornieren bzw. mit dem SFC 34 QRY_DINT den Zustand des Verzögerungsalarms zu ermitteln.

Näheres zum Einsatz der SFCs finden Sie im Teil "Integrierte Standard SFCs".

Mittels einer Hardware-Konfiguration können Sie die Priorität der entsprechenden OBs verändern. Hierzu öffnen sie die CPU mit **Bearbeiten** > *Objekteigenschaften* > *Alarmer*. Hier können Sie die entsprechenden Prioritäten einstellen.

Fehlerverhalten Fehlt zum Zeitpunkt seines Aufrufs der Verzögerungsalarm-OB im Anwenderprogramm, ruft das Betriebssystem den OB 85 auf. Ist dieser nicht vorhanden, geht die CPU in STOP. Kommt es zu einem Fehler bei der Verzögerungsalarmbearbeitung, z.B. Verzögerungszeit ist abgelaufen und der zugehörige OB ist noch in Bearbeitung, wird der Zeitfehler-OB OB 80 aufgerufen. Ist dieser nicht vorhanden, geht die CPU in STOP.

Lokaldaten Nachfolgend sehen Sie die Belegung der Startinformationen für OB 20 und die defaultmäßige symbolische Bezeichnung und die Datentypen:

Variable	Datentyp	Beschreibung
OB20_EV_CLASS	BYTE	Ereignisklasse und Kennung: 11h: Verzögerungsalarm ist aktiv
OB20_STRT_INF	BYTE	21h: Startanforderung für OB 20
OB20_PRIORITY	BYTE	Parametrierte Prioritätsklasse: Defaultwerte 3 (OB20)
OB20_OB_NUMBR	BYTE	OB-NR. (20)
OB20_RESERVED_1	BYTE	reserviert
OB20_RESERVED_2	BYTE	reserviert
OB20_SIGN	WORD	Anwenderkennung: Eingangsparameter SIGN aus dem Aufruf des SFC 32
OB20_DTIME	TIME	Abgelaufene Verzögerungszeit in ms
OB20_DATE_TIME	DATE_AND_TIME	Datum und Uhrzeit, zu denen der OB angefordert wurde

Informationen, wie Sie auf die Lokaldaten zugreifen können, finden Sie in der Beschreibung zum OB 1.

OB 35 - Weckalarm

Beschreibung Mit Hilfe des Weckalarm OBs können Sie die zyklische Programmbe-
arbeitung in äquidistanten Zeitabständen unterbrechen. Startzeitpunkt des
Zeittakts ist der Betriebszustandwechsel von ANLAUF in RUN nach
Abarbeitung des Anlauf-OB OB 100.

Weckalarm-OB	Defaultwert für Zeitraster	Defaultwert für die Prioritätsklasse
OB 35	100ms	12

Aktivierung Einen Weckalarm aktivieren Sie, indem Sie den entsprechenden OB in
Ihrer CPU programmieren.
Sie können den Weckalarm mit Hilfe des SFC 41 DIS_AIRT verzögern
bzw. die Verzögerung mit dem SFC 42 EN_AIRT aufheben.

Funktionsweise Nach dem Übergang von ANLAUF in RUN wird der aktivierte Weckalarm-
OB in den vorgegebenen äquidistanten Intervallen aufgerufen.
Durch Programmieren des OB 35 können Sie auf diese Weise
zeitgesteuert ein Unterprogramm ausführen.

Lokaldaten Nachfolgend sehen Sie die Belegung der Startinformationen für OB 35, die
defaultmäßige symbolische Bezeichnung und die Datentypen:

Variable	Datentyp	Beschreibung
OB35_EV_CLASS	BYTE	Ereignisklasse und Kennung: 11h: Weckalarm ist aktiv
OB35_STRT_INF	BYTE	36h: Startanforderung für OB 35
OB35_PRIORITY	BYTE	Parametrierte Prioritätsklasse; Default: 12 (OB35)
OB35_OB_NUMBR	BYTE	OB-NR. (35)
OB35_RESERVED_1	BYTE	reserviert
OB35_RESERVED_2	BYTE	reserviert
OB35_PHASE_OFFSET	WORD	Phasenverschiebung in ms
OB35_RESERVED_3	INT	reserviert
OB35_EXT_FREQ	INT	Zeittakt in ms
OB35_DATE_TIME	DATE_AND_TIME	Datum und Uhrzeit der OB-Anforderung

Informationen, wie Sie auf die Lokaldaten zugreifen können, finden Sie in
der Beschreibung zum OB 1.

OB 40 - Prozessalarm

Beschreibung	<p>Durch Parametrierung legen Sie für jedes Modul fest, welche Kanäle bei welcher Randbedingung einen Prozessalarm auslösen.</p> <p>Mit den SFC 55 WR_PARM, SFC 56 WR_DPARM und SFC 57 PARM_MOD können Sie die prozessalarmerfassenden Module auch während des laufenden Betriebs (um-)parametrieren.</p>
Aktivierung	<p>Die Prozessalarmverwaltung der CPU ist immer aktiv. Damit ein Modul einen Prozessalarm auslösen kann, müssen Sie auf dem entsprechenden Modul über eine Hardware-Konfiguration die Auslösung eines Prozessalarms freigeben. Hierbei können Sie wählen, ob der Prozessalarm bei kommendem, gehenden oder bei beiden Ereignis ausgelöst werden soll.</p>
Funktionsweise	<p>Nach Auslösen eines Prozessalarms durch das Modul identifiziert das Betriebssystem den Steckplatz und ruft den Prozessalarm-OB auf. Hat dieser eine höhere Priorität als die momentan aktive Prioritätsklasse, wird dieser gestartet. Nach der Bearbeitung des zum Prozessalarm gehörenden OB wird der Prozessalarm auf dem Modul quittiert.</p> <p>Tritt in der Zeit zwischen der Identifikation und der Quittung eines Prozessalarms auf demselben Modul erneut ein Ereignis auf, das einen Prozessalarm auslösen soll, gilt:</p> <ul style="list-style-type: none">• Tritt das Ereignis bei dem Kanal auf, der zuvor den Prozessalarm ausgelöst hat, so geht der zugehörige Alarm verloren.• Tritt das Ereignis bei einem anderen Kanal desselben Moduls auf, so kann momentan kein Prozessalarm ausgelöst werden. Steht nach der Quittierung des vorhergehenden Ereignisses das "neue" Alarmereignis noch an, wird ein Prozessalarm ausgelöst. Ansonsten geht der Alarm verloren.• Soll ein Prozessalarm ausgelöst werden, dessen OB aufgrund eines Prozessalarms eines anderen Moduls aktiv ist, so kann nur dann ein Prozessalarm ausgelöst werden, wenn nach der Quittierung das auslösende Ereignis noch ansteht. <p>Im Anlauf erzeugt die Baugruppe keine Prozessalarme. Die Alarmbearbeitung beginnt beim Übergang in den Betriebszustand RUN. Beim Übergang anstehende Prozessalarme gehen verloren.</p>
Fehlerverhalten	<p>Fehlt zum Zeitpunkt seines Aufrufs der Prozessalarm-OB im Anwenderprogramm, ruft das Betriebssystem den OB 85 auf. Der Prozessalarm wird quittiert. Ist der OB 85 nicht vorhanden, geht die CPU in STOP.</p>

Diagnosealarm Während der Bearbeitung eines Prozessalarms kann ein Diagnosealarm ausgelöst werden. Tritt in der Zeit vom Auslösen des Prozessalarms bis zu dessen Quittierung auf demselben Kanal ein weiterer Prozessalarm auf, wird über einen Diagnosealarm der Prozessalarmverlust zur Systemdiagnose gemeldet.

Lokaldaten Nachfolgend sehen Sie die Belegung der Startinformationen für OB 40, die defaultmäßige symbolische Bezeichnung und die Datentypen:

Variable	Datentyp	Beschreibung
OB40_EV_CLASS	BYTE	Ereignisklasse und Kennung: 11h: Alarm ist aktiv
OB40_STRT_INF	BYTE	Bei CPU 112: Die OB-Startinformation hängt von der Leitung ab, über die der Alarm ausgelöst wurde. 41h: E0.0 42h: E0.1 43h: E0.2 44h: E0.3 Andere CPUs 11x und CC03: 40h: Alarmeingänge E0.0 ... E0.3 41h: Interrupt-Quellen am V-Bus (ext. gesteckte Module) Bei System 200V und 300V: 41h: Alarm über Interrupt-Leitung 1
OB40_PRIORITY	BYTE	Parametrierte Prioritätsklasse: Defaultwert: 16 (OB 40)
OB40_OB_NUMBR	BYTE	OB-NR. (40)
OB40_RESERVED_1	BYTE	reserviert
OB40_IO_FLAG	BYTE	54h: Eingabe-Module 55h: Ausgabe-Module
OB40_MDL_ADDR	WORD	Logische Basisadresse des Moduls, das den Alarm auslöst
OB40_POINT_ADDR	DWORD	Bei Digital-Modulen: Bitfeld mit den Zuständen der Eingänge auf dem Modul (Bit 0 entspricht dem ersten Eintrag). Bei Analog-Modulen: Bitfeld mit Informationen welcher Kanal welche Grenzen erreicht hat. Bei CPs oder IMs: Alarmzustand des Moduls.
OB40_DATE_TIME	DATE_AND_TIME	Datum und Uhrzeit, zu denen der OB angefordert wurde.

Informationen, wie Sie auf die Lokaldaten zugreifen können finden Sie in der Beschreibung zum OB 1.

OB 80 - Zeitfehler

Beschreibung Das Betriebssystem der CPU ruft den OB 80 auf, wenn einer der folgenden Fehler auftritt:

- Überschreiten der Zyklusüberwachungszeit
- Quittungsfehler bei der Bearbeitung eines OBs d.h. der abgeforderte OB ist noch in Bearbeitung bzw. ein OB wird innerhalb einer Prioritätsklasse zu oft angefordert.
- Uhrzeitfehleralarm, d.h. abgelaufener Uhrzeitalarm durch Vorstellen der Uhrzeit oder nach Übergang in RUN.

Sie können den Zeitfehler-OB mit Hilfe der SFC 41 verzögern und mit der SFC 42 wieder freigeben.



Hinweis!

Haben Sie den OB 80 nicht programmiert, geht die CPU in den Betriebszustand STOP.

Wird der OB 80 in demselben Zyklus zweimal aufgrund der Zykluszeitüberschreitung aufgerufen, geht die CPU in STOP. Sie können dies durch Aufruf des SFC 43 RE_TRIGR an geeigneter Stelle verhindern.

Lokaldaten Nachfolgend sehen Sie die Belegung der Startinformationen für den OB 80, die defaultmäßige symbolische Bezeichnung und die Datentypen:

Variable	Datentyp	Beschreibung
OB80_EV_CLASS	BYTE	Ereignisklasse und Kennung: 35h
OB80_FLT_ID	BYTE	Fehlercode (mögliche Werte: 01h, 02h, 05h, 06h, 07h, 08h, 09h, 0Ah)
OB80_PRIORITY	BYTE	Prioritätsklasse: 26 (Betriebszustand RUN) 28 (Überlauf des OB-Anforderungspuffers)
OB80_OB_NUMBR	BYTE	OB-NR. (80)
OB80_RESERVED_1	BYTE	reserviert
OB80_RESERVED_2	BYTE	reserviert
OB80_ERROR_INFO	WORD	Fehlerinformation: abhängig vom Fehlercode
OB80_ERR_EV_CLASS	BYTE	Klasse des Ereignisses, das den Fehler ausgelöst hat
OB80_ERR_EV_NUM	BYTE	Nummer des Ereignisses, das den Fehler ausgelöst hat
OB80_OB_PRIORITY	BYTE	Fehlerinformation: abhängig vom Fehlercode
OB80_OB_NUM	BYTE	Fehlerinformation: abhängig vom Fehlercode
OB80_DATE_TIME	DATE_AND_TIME	Datum und Uhrzeit, zu denen der OB angefordert wurde

Informationen, wie Sie auf die Lokaldaten zugreifen können finden Sie in der Beschreibung zum OB 1.

Belegung der Variablen nach Fehlercode

Die vom Fehlercode abhängigen Variablen haben folgende Bedeutung:

Fehlercode	Variable	Bit	Beschreibung
01h	OB80_ERROR_INFO		<i>Zykluszeit überschritten</i> Laufzeit des letzten Zyklus (ms)
	OB80_ERR_EV_CLASS		Klasse des Ereignisses, das den Alarm ausgelöst hat
	OB80_ERR_EV_NUM		Nummer des Ereignisses, das den Alarm ausgelöst hat
	OB80_OB_PRIORITY		Prioritätsklasse des OBs, der bearbeitet wurde, als der Fehler auftrat
	OB80_OB_NUM		Nummer des OBs, der bearbeitet wurde, als der Fehler auftrat
02h	OB80_ERROR_INFO		<i>Der angeforderte OB ist noch in Bearbeitung</i> Die zugehörige temporäre Variable des angeforderten OB. Diese ist bestimmt durch:
	OB80_ERR_EV_CLASS		OB80_ERR_EV_CLASS und OB80_ERR_EV_NUM
	OB80_ERR_EV_NUM		Klasse des Ereignisses, das den Alarm ausgelöst hat
	OB80_OB_PRIORITY		Nummer des Ereignisses, das den Alarm ausgelöst hat
	OB80_OB_NUM		Prioritätsklasse des fehlerverursachenden OBs
			Nummer des fehlerverursachenden OBs
05h und 06h	OB80_ERROR_INFO	Bit 0 = "1"	<i>Abgelaufener Uhrzeitalarm durch Uhrzeitsprung</i> Abgelaufener Uhrzeitalarm bei Wiedereintritt in RUN nach HALT
		...	Für den Uhrzeitalarm 0 liegt der Startzeitpunkt in der Vergangenheit.
		Bit 7 = "1"	...
		Bit 15 ... 8	Für den Uhrzeitalarm 7 liegt der Startzeitpunkt in der Vergangenheit.
	OB80_ERR_EV_CLASS		nicht verwendet
OB80_ERR_EV_NUM		nicht verwendet	
OB80_OB_PRIORITY		nicht verwendet	
OB80_OB_NUM		nicht verwendet	

Fortsetzung ...

... Fortsetzung Fehlercode

Fehlercode	Variable	Bit	Beschreibung
07h	Bedeutung der Parameter siehe Fehlercode 02h		<i>Überlauf des OB-Anforderungspuffers für die aktuelle Prioritätsklasse</i> (jede OB-Startanforderung für eine Prioritätsklasse wird in den zugehörigen OB-Anforderungspuffer eingetragen; nach Beendigung des OBs wird der Eintrag wieder gelöscht. Falls für eine Prioritätsklasse mehr Startanforderungen vorliegen als die maximal mögliche Anzahl der Einträge im zugehörigen OB-Anforderungspuffer, wird der OB 80 mit dem Fehlercode 07h aufgerufen.)
08h			<i>Taktsynchronalarm-Zeitfehler</i>
09h			<i>Alarmverlust durch zu hohe Alarmlast</i>
0Ah	OB80_ERROR_INFO		<i>Wiedereintritt in RUN nach CiR (Configuration in RUN) CiR-Synchronisationszeit in ms</i>

OB 81 - Stromversorgungsfehler

Beschreibung

Das Betriebssystem der CPU ruft den OB 81 auf, wenn ein Ereignis auftritt, das durch einen Fehler in der Stromversorgung oder der Pufferung ausgelöst wird. Eine Überprüfung erfolgt ausschließlich nach Netz EIN.

Ist im Fehlerfall der OB 81 in der CPU nicht vorhanden, bleibt diese im RUN.

Sie können den Stromversorgungsfehler-OB mit Hilfe der SFCs 39 ... 42 sperren bzw. verzögern und wieder freigeben.

Lokaldaten

Nachfolgend sehen Sie die Belegung der Startinformationen für den OB 81, die defaultmäßige symbolische Bezeichnung und die Datentypen:

Variable	Datentyp	Beschreibung
OB81_EV_CLASS	BYTE	Ereignisklasse und Kennung: 39h: kommendes Ereignis
OB81_FLT_ID	BYTE	Fehlercode: 22h: Pufferspannung in CPU fehlt
OB81_PRIORITY	BYTE	Prioritätsklasse: 28 (Betriebszustand ANLAUF)
OB81_OB_NUMBR	BYTE	OB-NR. (81)
OB81_RESERVED_1	BYTE	reserviert
OB81_RESERVED_2	BYTE	reserviert
OB81_RACK_CPU	WORD	Bit 2 ... 0: 000 (Baugruppenträgernummer) Bit 3: 1 (Master-CPU) Bit 7 ... 4: 1111 (fix)
OB81_RESERVED_3	BYTE	reserviert
OB81_RESERVED_4	BYTE	reserviert
OB81_RESERVED_5	BYTE	reserviert
OB81_RESERVED_6	BYTE	reserviert
OB80_DATE_TIME	DATE_AND_TIME	Datum und Uhrzeit, zu denen der OB angefordert wurde.

Informationen, wie Sie auf die Lokaldaten zugreifen können finden Sie in der Beschreibung zum OB 1.

OB 82 - Diagnosealarm

Beschreibung	<p>Die Systemdiagnose ist die Erkennung, Auswertung und die Meldung von Fehlern, die innerhalb Ihres Automatisierungssystems auftreten. Dies können Fehler im Anwenderprogramm oder Ausfälle auf Modulen aber auch Drahtbrüche bei Signal-Modulen sein.</p> <p>Wenn ein diagnosefähiges Modul, bei dem Sie den Diagnosealarm freigegeben haben, einen Fehler erkennt, stellt dies eine Diagnosealarmanforderung sowohl bei kommendem als auch bei gehendem Ereignis an die CPU. Daraufhin ruft das Betriebssystem den OB 82 auf. Der OB 82 enthält in seinen lokalen Variablen die logische Basisadresse sowie eine vier Byte lange Diagnoseinformation des fehlerhaften Moduls. Haben Sie den OB 82 nicht programmiert, geht die CPU in den Betriebszustand STOP. Sie können den Diagnosealarm mit Hilfe des SFC 41 DIS_AIRT verzögern bzw. die Verzögerung mit dem SFC 42 EN_AIRT aufheben.</p>
Diagnose in Ringpuffer	<p>Alle an das Betriebssystem der CPU gemeldeten Diagnoseereignisse werden in einem Diagnosepuffer in der Reihenfolge ihres Auftretens mit Datum und Uhrzeit abgelegt. Dies ist ein gepufferter Speicherbereich der CPU, der seinen Inhalt auch nach Umräumen behält.</p> <p>Dieser Diagnosepuffer ist als Ringpuffer aufgebaut und bietet bei den CPUs von VIPA Platz für 100 Diagnoseeinträge. Ist der Diagnosepuffer voll, wird der jeweils älteste Eintrag vom aktuellen Diagnoseereignis überschrieben. Mittels der <i>Zielsystemfunktionen</i> des Siemens SIMATIC Manager können Sie jederzeit den Diagnosepuffer auslesen.</p> <p>Neben den Standardeinträgen im Diagnosepuffer gibt es in den CPUs der VIPA noch zusätzliche Einträge, die ausschließlich in Form einer Ereignis-ID angezeigt werden. Informationen hierzu finden Sie im Handbuch Ihrer CPU im Teil "Einsatz CPU ..." unter "VIPA-spezifische Diagnoseeinträge".</p>
Parametrierbare Diagnose	<p>Bei den parametrierbaren Diagnoseereignissen erfolgt eine Meldung nur dann, wenn Sie mittels Parametrierung die Diagnose freigegeben haben. Die nichtparametrierbaren Diagnoseereignisse werden unabhängig von der Diagnosefreigabe immer gemeldet.</p>
Diagnose mittels SFC eintragen	<p>Mit der Systemfunktion SFC 52 WR_USMSG können Sie einen Eintrag in den Diagnosepuffer schreiben. Näheres hierzu finden Sie im Teil "Integrierte Standard-SFCs".</p>
Diagnosedaten mittels SFC 59 auslesen	<p>Innerhalb des OB 82 können Sie mittels SFC 59 RD_REC (Datensatz lesen) auf detaillierte Fehlerinformationen zugreifen. Die Diagnosedaten sind bis zum Verlassen des OB 82 konsistent. Mit dem Verlassen des OB 82 wird der Diagnosealarm quittiert.</p> <p>Die Diagnosedaten befinden sich in Datensatz 0 (DS 0) und Datensatz 1 (DS 1). DS 0 besteht aus 4 Byte, die den aktuellen Zustand des Moduls beschreiben. Die Belegung dieser Byte entspricht der Belegung der Byte 8 ... 11 der Startinformationen des OB 82. DS 1 setzt sich aus den 4 Byte des DS 0 und den modulspezifischen Diagnosedaten zusammen.</p> <p>Näheres zu den modulspezifischen Diagnosedaten finden Sie bei der Beschreibung des entsprechenden Moduls.</p>

Lokaldaten

Nachfolgend sehen Sie die Belegung der Startinformationen für den OB 82, die defaultmäßige symbolische Bezeichnung und die Datentypen:

Variable	Datentyp	Beschreibung
OB82_EV_CLASS	BYTE	Ereignisklasse und Kennung: 38h: gehendes Ereignis 39h: kommendes Ereignis
OB82_FLT_ID	BYTE	Fehlercode (42h)
OB82_PRIORITY	BYTE	Prioritätsklasse: parametrierbar über Hardware-Konfigurator
OB82_OB_NUMBR	BYTE	OB-NR. (82)
OB82_RESERVED_1	BYTE	reserviert
OB82_IO_FLAG	BYTE	Eingabe-Module 54h Ausgabe-Module 55h
OB82_MDL_ADDR	INT	Logische Basisadresse des Moduls, in dem der Fehler aufgetreten ist
OB82_MDL_DEFECT	BOOL	Modulstörung
OB82_INT_FAULT	BOOL	Interner Fehler
OB82_EXT_FAULT	BOOL	Externer Fehler
OB82_PNT_INFO	BOOL	Kanalfehler vorhanden
OB82_EXT_VOLTAGE	BOOL	Externe Hilfsspannung fehlt
OB82_FLD_CONNCTR	BOOL	Frontstecker fehlt
OB82_NO_CONFIG	BOOL	Modul nicht parametriert
OB82_CONFIG_ERR	BOOL	Falsche Parameter im Modul
OB82_MDL_TYPE	BYTE	Bit 3 ... 0: Modulklasse Bit 4: Kanalinformation vorhanden Bit 5: Anwenderinformation vorhanden Bit 6: Diagnosealarm von Stellvertreter Bit 7: reserviert
OB82_SUB_MDL_ERR	BOOL	Anwendermodul falsch/fehlt
OB82_COMM_FAULT	BOOL	Kommunikationsstörung
OB82_MDL_STOP	BOOL	Betriebszustand (0: RUN, 1:STOP)
OB82_WTCH_DOG_FLT	BOOL	Zeitüberwachung hat angesprochen
OB82_INT_PS_FLT	BOOL	Modulinterne Versorgungsspannung ausgefallen
OB82_PRIM_BATT_FLT	BOOL	Batterie leer
OB82_BCKUP_BATT_FLT	BOOL	Gesamte Pufferung ausgefallen
OB82_RESERVED_2	BOOL	Reserviert
OB82_RACK_FLT	BOOL	Erweiterungsgeräteausfall
OB82_PROC_FLT	BOOL	Prozessorausfall
OB82_EPROM_FLT	BOOL	EPROM-Fehler
OB82_RAM_FLT	BOOL	RAM-Fehler
OB82_ADU_FLT	BOOL	ADU/DAU-Fehler
OB82_FUSE_FLT	BOOL	Sicherungsausfall
OB82_HW_INTR_FLT	BOOL	Prozessalarm verloren
OB82_RESERVED_3	BOOL	reserviert
OB82_DATE_TIME	DATE_AND_TIME	Datum und Uhrzeit, zu denen der OB angefordert wurde

Informationen, wie Sie auf die Lokaldaten zugreifen können finden Sie in der Beschreibung zum OB 1.

OB 85 - Programmablauffehler

Beschreibung

Das Betriebssystem der CPU ruft den OB 85 auf wenn:

- ein OB aufgerufen wird, der nicht in der CPU geladen wurde
- ein Fehler beim Zugriff des Betriebssystems auf einen Baustein auftritt
- ein Peripheriezugriffsfehler bei der systemseitigen Aktualisierung des Prozessabbilds auftritt

Sie können den OB 85 mit Hilfe des SFC 41 verzögern bzw. mit SFC 42 wieder freigeben.



Hinweis!

Haben Sie den OB 85 nicht programmiert, geht die CPU in den Betriebszustand STOP.

Lokaldaten

Nachfolgend sehen Sie die Belegung der Startinformationen für den OB 85, die defaultmäßige symbolische Bezeichnung und die Datentypen:

Variable	Datentyp	Beschreibung
OB85_EV_CLASS	BYTE	Ereignisklasse und Kennung: 35h 38h (nur bei Fehlercode B3h, B4h) 39h (nur bei Fehlercode B1h, B2h, B3h, B4h)
OB85_FLT_ID	BYTE	Fehlercode (mögliche Werte: A1h, A2h, A3h, A4h, B1h, B2h, B3h, B4h)
OB85_PRIORITY	BYTE	Prioritätsklasse: 26 (Defaultwert Betriebszustand RUN) 28 (Betriebszustand ANLAUF)
OB85_OB_NUMBR	BYTE	OB-NR. (85)
OB85_RESERVED_1	BYTE	reserviert
OB85_RESERVED_2	BYTE	reserviert
OB85_RESERVED_3	INT	reserviert
OB85_ERR_EV_CLASS	BYTE	Klasse des Ereignisses, das den Fehler ausgelöst hat.
OB85_ERR_EV_NUM	BYTE	Nummer des Ereignisses, das den Fehler ausgelöst hat.
OB85_OB_PRIOR	BYTE	Prioritätsklasse des OBs, der bearbeitet wurde, als der Fehler auftrat.
OB85_OB_NUM	BYTE	Nummer des OBs, der bearbeitet wurde, als der Fehler auftrat.
OB85_DATE_TIME	DATE_AND_TIME	Datum und Uhrzeit, zu denen der OB angefordert wurde.

Informationen, wie Sie auf die Lokaldaten zugreifen können finden Sie in der Beschreibung zum OB 1.

OB 85 in Abhängigkeit von Fehlercodes

Falls Sie den OB 85 in Abhängigkeit von den möglichen Fehlercodes programmieren wollen, wird empfohlen die lokalen Variablen wie folgt zu organisieren:

Variable	Datentyp
OB85_EV_CLASS	BYTE
OB85_FLT_ID	BYTE
OB85_PRIORITY	BYTE
OB85_OB_NUMBR	BYTE
OB85_DKZ23	BYTE
OB85_RESERVED_2	BYTE
OB85_Z1	WORD
OB85_Z23	DWORD
OB85_DATE_TIME	DATE_AND_TIME

Die nachfolgende Tabelle beinhaltet das Ereignis, das den OB 85 aufgerufen hat.

OB85_EV_CLASS	OB85_FLT_ID	Variable	Bedeutung
35h	A1h, A2h		Ihr Programm oder das Betriebssystem erzeugt ein Starterereignis für einen OB, der nicht in der CPU geladen ist.
	A1h, A2h	OB85_Z1	Die zugehörige temporäre Variable des angeforderten OBs. Dieser ist bestimmt durch OB85_Z23.
	A1h, A2h	OB85_Z23	High-Word: Klasse und Nummer des verursachenden Ereignisses Low-Word, High-Byte: Zum Fehlerzeitpunkt aktive Programm-ebene Low-Word, Low-Byte: Aktiver OB
35h	A3h	OB85_Z1	Fehler beim Zugriff des Betriebssystems auf einen Baustein Fehlererkennung des Betriebssystems High-Byte: 1: Integrierte Funktion 2: IEC-Timer Low-Byte: 0: keine Fehlerauflösung 1: Baustein nicht geladen 2: Bereichslängenfehler 3: Schreibschutzfehler

Fortsetzung ...

... Fortsetzung

OB85_EV_CLASS	OB85_FLT_ID	Variable	Bedeutung
		OB85_Z23	High-Word: Bausteinnummer Low-Word: Relativadresse des Fehler verursachenden MC7-Befehls. Der Bausteintyp ist OB85_DKZ23 zu entnehmen. (88h: OB, 8Ch: FC, 8Eh: FB, 8Ah: DB)
35h	A4h		PROFINET-DB nicht ansprechbar
34h	A4h		PROFINET-DB wieder ansprechbar
39h	B1h B2h B1h, B2h B1h, B2h B1h, B2h	OB85_DKZ23 OB85_Z1 OB85_Z23	Peripheriezugriffsfehler beim Aktualisieren des Prozessabbilds der Eingänge. Peripheriezugriffsfehler beim Aktualisieren des Prozessabbilds der Ausgänge zu den Ausgabe-Modulen. Kennung für die Art des Prozessabbildtransfers bei dem der Peripheriezugriffsfehler auftrat: 10h: Bytezugriff 20h: Wortzugriff 30h: Doppelwortzugriff 57h: Übertragung eines projektierten Konsistenzbereichs reserviert für interne Verwendung der CPU: logische Basisadresse des Moduls Falls OB85_RESERVED_2 den Wert 76h hat, enthält OB85_Z1 den Rückgabewert der betroffenen SFC Byte 0: Teilprozessabbild-Nummer Byte 1: Irrelevant, falls OB85_DKZ23=10, 20 oder 30 OB85_DKZ23=57: Länge des Konsistenzbereichs in Bytes Byte 2, 3 Nummer des PZF-verursachenden Peripheriebytes falls OB85_DKZ23=10, 20 oder 30 OB85_DKZ23=57: logische Anfangsadresse des Konsistenzbereichs
Sie erhalten die Fehlercodes B1h und B2h, falls Sie für die systemseitige Prozessabbildaktualisierung das wiederholte Melden von Peripheriezugriffsfehlern projektiert haben.			

Fortsetzung ...

... Fortsetzung

OB85_EV_CLASS	OB85_FLT_ID	Variable	Bedeutung
38h, 39h	B3h		Peripheriezugriffsfehler beim Aktualisieren des Prozessabbilds der Eingänge kommend/gehend.
38h, 39h	B4h		Peripheriezugriffsfehler beim Aktualisieren des Prozessabbilds der Ausgänge zu den Ausgabe-Modulen kommend/gehend.
	B3h, B4h	OB85_DKZ23	Kennung für die Art des Prozessabbildtransfers bei dem der Peripheriezugriffsfehler auftrat: 10h: Bytezugriff 20h: Wortzugriff 30h: Doppelwortzugriff 57h: Übertragung eines projektierten Konsistenzbereichs
	B3h, B4h	OB85_Z1	reserviert für interne Verwendung der CPU: logische Basisadresse des Moduls Falls OB85_RESERVED_2 den Wert 76h hat, enthält OB85_Z1 den Rückgabewert der betroffenen SFC
	B3h, B4h	OB85_Z23	Byte 0: Teilprozessabbild-Nummer Byte 1: Irrelevant, falls OB85_DKZ23=10, 20 oder 30 OB85_DKZ23=57: Länge des Konsistenzbereichs in Bytes Byte 2, 3 Nummer des PZF-verursachenden Peripheriebytes falls OB85_DKZ23=10, 20 oder 30 OB85_DKZ23=57: logische Anfangsadresse des Konsistenzbereichs
<p>Sie erhalten die Fehlercodes B3h und B4h, falls Sie für die systemseitige Prozessabbildaktualisierung das Melden von kommenden und gehenden Peripheriezugriffsfehlern projektiert haben. Nach einem Neustart werden bei der nächsten Prozessabbildaktualisierung alle Zugriffe auf nicht vorhandenen Ein- und Ausgänge als kommende Peripheriezugriffsfehler gemeldet.</p>			

OB 86 - Slaveausfall / -wiederkehr

Beschreibung Das Betriebssystem der CPU ruft den OB 86 auf, wenn der Ausfall eines Slaves erkannt wird (sowohl bei kommendem als auch bei gehendem Ereignis).



Hinweis!

Haben Sie den OB 86 nicht programmiert, geht die CPU in den Betriebszustand STOP.

Sie können den OB 86 mit Hilfe des SFC 41 verzögern bzw. mit SFC 42 wieder freigeben.

Lokaldaten Nachfolgend sehen Sie die Belegung der Startinformationen für den OB 86, die defaultmäßige symbolische Bezeichnung und die Datentypen:

Variable	Datentyp	Beschreibung
OB86_EV_CLASS	BYTE	Ereignisklasse und Kennung: 38h: gehendes Ereignis 39h: kommendes Ereignis
OB86_FLT_ID	BYTE	Fehlercode (mögliche Werte: C4h, C5h, C7h, C8h)
OB86_PRIORITY	BYTE	Prioritätsklasse: Wird über den Hardware-Konfigurator eingestellt
OB86_OB_NUMBR	BYTE	OB-NR. (86)
OB86_RESERVED_1	BYTE	reserviert
OB86_RESERVED_2	BYTE	reserviert
OB86_MDL_ADDR	WORD	Abhängig vom Fehlercode
OB86_RACKS_FLTD	ARRAY (0 ... 31) OF BOOL	Abhängig vom Fehlercode
OB86_DATE_TIME	DATE_AND_TIME	Datum und Uhrzeit, zu denen der OB angefordert wurde

Informationen, wie Sie auf die Lokaldaten zugreifen können finden Sie in der Beschreibung zum OB 1.

OB 86 in Abhängigkeit von Fehlercodes

Falls Sie den OB 86 in Abhängigkeit von den möglichen Fehlercodes programmieren wollen, wird empfohlen die lokalen Variablen wie folgt zu organisieren:

Variable	Datentyp
OB86_EV_CLASS	BYTE
OB86_FLT_ID	BYTE
OB86_PRIORITY	BYTE
OB86_OB_NUMBR	BYTE
OB86_RESERVED_1	BYTE
OB86_RESERVED_2	BYTE
OB86_MDL_ADDR	WORD
OB86_Z23	DWORD
OB86_DATE_TIME	DATE_AND_TIME

Die Variablen, deren Inhalt abhängig vom Fehlercode ist, haben folgende Bedeutung:

EV_CLASS	FLT_ID	Variable	Bit ...	Beschreibung
39h, 38h	C4h C5h	OB86_MDL_ADDR OB86_Z23	Bit 7 ... 0 Bit 15 ... 8 Bit 30 ... 16 Bit 31	Ausfall einer DP-Station Störung einer DP-Station Logische Basisadresse des DP-Masters Adresse des betroffenen DP-Slaves: Nummer der DP-Station DP-Mastersystem-ID Logische Basisadresse des DP-Slave I/O Kennung
	38h	C7h	OB86_MDL_ADDR OB86_Z23	Bit 7 ... 0 Bit 15 ... 8 Bit 30 ... 16 Bit 31
		C8h	OB86_MDL_ADDR OB86_Z23	Bit 7 ... 0 Bit 15 ... 8 Bit 30 ... 16 Bit 31

OB 100 - Neustart

Beschreibung

Bei einem Neustart bzw. Anlauf versetzt die CPU sich selbst und die angebundenen Module in den projektierten Grundzustand, löscht die nicht remanenten Daten im Systemspeicher, ruft den OB 100 auf und bearbeitet danach den OB 1 von Anfang an.

Hierbei bleiben das aktuelle Programm und die aktuellen Daten im Arbeitsspeicher erhalten, ebenso die per SFC erzeugten Datenbausteine.

Bei den VIPA CPUs wird in folgenden Fällen der OB 100 aufgerufen:

- nach NETZ EIN mit Betriebsartenschalter in RUN
- nach Umschalten des Betriebsartenschalters von STOP in RUN
- nach einer RUN/STOP-Anforderung durch eine Kommunikationsfunktion

Auch wenn kein OB 100 in die CPU geladen ist, geht diese ohne Fehlermeldung in RUN.

Lokaldaten

Nachfolgend sehen Sie die Belegung der Startinformationen für den OB 100, die defaultmäßige symbolische Bezeichnung und die Datentypen:

Variable	Datentyp	Beschreibung
OB100_EV_CLASS	BYTE	Ereignisklasse und Kennung: 13h: aktiv
OB100_STRTUP	BYTE	Anlaufanforderung 81h: Manuelle Neustart-Anforderung
OB100_PRIORITY	BYTE	Prioritätsklasse: 27
OB100_OB_NUMBR	BYTE	OB-NR. (100)
OB100_RESERVED_1	BYTE	reserviert
OB100_RESERVED_2	BYTE	reserviert
OB100_STOP	WORD	Nummer des Ereignisses, das die CPU in STOP versetzt hat
OB100_STRT_INFO	DWORD	Ergänzende Informationen zum aktuellen Anlauf (siehe Folgeseite)
OB100_DATE_TIME	DATE_AND_TIME	Datum und Uhrzeit, zu denen der OB angefordert wurde

Informationen, wie Sie auf die Lokaldaten zugreifen können finden Sie in der Beschreibung zum OB 1.

Belegung Die folgende Tabelle zeigt die Belegung der Variablen OB100_STR_INFO:
OB100_STR_INFO

Bit-Nr.	Bedeutung	mögliche Werte (binär)	Beschreibung
31 - 24	Anlaufinformation	xxxx xxx0 xxxx xxx1 xxxx 0xxx xxxx 1xxx	keine Soll-Istausbau-Differenz vorhanden Soll-Istausbau-Differenz vorhanden Uhr für Zeitstempel bei letztem NETZEIN nicht gepuffert Uhr für Zeitstempel bei letztem NETZEIN gepuffert
23 - 16	Soeben durchgeführte Anlaufart	0000 0011 0000 0100 0001 0000 0001 0011 0001 0100 0010 0000 0010 0011 0010 0100	Neustart über Betriebsartenschalter Neustart über MPI-Bedienung Automatischer Neustart nach gepuffertem Netzein Neustart über Betriebsartenschalter; letzter NETZEIN gepuffert Neustart über MPI-Bedienung; letzter NETZEIN gepuffert Automatischer Neustart nach ungepuffertem NETZEIN (mit systemseitigem Urlöschen) Neustart über Betriebsartenschalter; letzter NETZEIN ungepuffert Neustart über MPI-Bedienung; letzter NETZEIN ungepuffert
15 - 12	Zulässigkeit automatischer Anlaufarten	0000 0001 0111	Automatischer Anlauf unzulässig, Urlöschen angefordert Automatischer Anlauf unzulässig, Parameteränderung erforderlich Automatischer Neustart zulässig
11 - 8	Zulässigkeit manueller Anlaufarten	0000 0001 0111	Manueller Anlauf unzulässig, Urlöschen angefordert Manueller Anlauf unzulässig, Parameteränderung erforderlich Manueller Neustart zulässig
7 - 0	Letzte gültige Bedienung oder Einstellung der automatischen Anlaufart bei NETZEIN	0000 0000 0000 0011 0000 0100 0001 0000 0001 0011 0001 0100 0010 0000 0010 0011 1010 0000	Keine Anlaufart Neustart durch Betriebsartenschalter Neustart über MPI-Bedienung Automatischer Neustart nach gepuffertem NETZEIN Neustart über Betriebsartenschalter, letzter NETZEIN gepuffert Neustart über MPI-Bedienung, letzter NETZEIN gepuffert Automatischer Neustart nach ungepuffertem NETZEIN (mit systemseitigem Urlöschen) Neustart über Betriebsartenschalter, letzter NETZEIN ungepuffert Neustart über MPI-Bedienung, letzter NETZEIN ungepuffert

OB 121 - Programmierfehler (Synchronfehler)

Beschreibung Das Betriebssystem der CPU generiert ein Fehler-Ereignis, wenn in unmittelbarem Zusammenhang mit der Programmbearbeitung ein Fehler auftritt. Ist der OB 121 nicht programmiert, wechselt die CPU in den Betriebszustand STOP.

Wenn Sie beispielsweise in Ihrem Programm einen Baustein aufrufen, der nicht in die CPU geladen wurde, dann wird der OB 121 aufgerufen. Der OB 121 läuft in derselben Prioritätsklasse wie der unterbrochene Baustein. Sie können deshalb innerhalb des OB 121 auf die Register des unterbrochenen Bausteins zugreifen und diese ggf. mit geänderten Parametern an den OB zurückgeben.

Startereignisse maskieren Die CPU verfügt über die folgenden SFCs, mit denen Sie Startereignisse des OB 121 maskieren und demaskieren können, während ihr Programm bearbeitet wird:

- Die SFC 36 MSK_FLT maskiert bestimmte Fehlercodes.
- Die SFC 37 DMSK_FLT demaskiert die Fehlercodes, die von der SFC 36 maskiert wurden.
- Die SFC 38 READ_ERR liest das Ereignisstatusregister.

Lokaldaten Nachfolgend sehen Sie die Belegung der Startinformationen für den OB 121, die defaultmäßige symbolische Bezeichnung und die Datentypen:

Variable	Datentyp	Beschreibung
OB121_EV_CLASS	BYTE	Ereignisklasse und Kennung: 25h
OB121_SW_FLT	BYTE	Fehlercode (siehe Folgeseite)
OB121_PRIORITY	BYTE	Prioritätsklasse: Prioritätsklasse des OBs, in dem der Fehler aufgetreten ist.
OB121_OB_NUMBR	BYTE	OB-Nr. (121)
OB121_BLK_TYPE	BYTE	Art des Bausteins, in dem der Fehler aufgetreten ist 88h: OB, 8Ah: DB, 8Ch: FC, 8Eh: FB
OB121_RESEVED_1	BYTE	reserviert (Datenbereich und die Zugriffsart)
OB121_FLT_REG	WORD	Fehlerquelle (abhängig vom Fehlercode), z.B.: <ul style="list-style-type: none"> • Register, in dem der Konvertierungsfehler aufgetreten ist. • Fehlerhafte Adresse (Lese-/Schreibfehler) • Fehlerhafte Nummer eines Timers, eines Zählers oder eines Bausteins. • Fehlerhafter Speicherbereich
OB121_BLK_NUM	WORD	Nummer des Bausteins mit dem fehlerverursachenden Befehl
OB121_PRG_ADDR	WORD	Relativadresse des fehlerverursachenden Befehls
OB121_DATE_TIME	DATE_AND_TIME	Datum und Uhrzeit, zu denen der OB angefordert wurde.

Informationen, wie Sie auf die Lokaldaten zugreifen können finden Sie in der Beschreibung zum OB 1.

Fehlercodes Die vom Fehlercode abhängigen Variablen haben folgende Bedeutung:

Fehlercode	Variable	Beschreibung
21h	OB121_FLT_REG:	BCD-Konvertierungsfehler Kennung für das betroffene Register (0000h: Akku 1)
22h 23h 28h 29h	OB121_RESERVED_1	Bereichslängenfehler beim Lesen Bereichslängenfehler beim Schreiben lesender Zugriff auf ein Byte, Wort oder Doppelwort mit einem Pointer, dessen Bitadresse ungleich 0 ist. schreibender Zugriff auf ein Byte, Wort oder Doppelwort mit einem Pointer, dessen Bitadresse ungleich 0 ist. fehlerhafte Byteadresse Der Datenbereich und die Zugriffsart sind OB121_RESERVED_1 zu entnehmen Bit 3 ... 0 Speicherbereich: 0: Peripheriebereich 1: Prozessabbild der Eingänge 2: Prozessabbild der Ausgänge 3: Merker 4: Global-DB 5: Instanz-DB 6: Eigene Lokaldaten 7: Lokaldaten des Aufrufers Bit 7 ... 4 Zugriffsart: 0: Bitzugriff 1: Bytezugriff 2: Wortzugriff 3: Doppelwortzugriff
24h 25h	OB121_FLT_REG	Bereichsfehler beim Lesen Bereichsfehler beim Schreiben enthält im Low-Byte die Kennung des unzulässigen Bereichs (86h eigener Lokaldatenbereich)
26h 27h	OB121_FLT_REG	Fehler bei Timernummer Fehler bei Zählernummer unzulässige Nummer

Fortsetzung ...

... Fortsetzung

Fehlercode	Variable	Beschreibung
30h		Schreibender Zugriff auf einen schreibgeschützten Global-DB
31h		Schreibender Zugriff auf einen schreibgeschützten Instanz-DB
32h		DB-Nummernfehler beim Zugriff auf einen Global-DB
33h		DB-Nummernfehler beim Zugriff auf einen Instanz-DB
	OB121_FLT_REG	unzulässige DB-Nummer
34h		Nummernfehler beim FC-Aufruf
35h		FB-Nummernfehler beim FB-Aufruf
3Ah		Zugriff auf einen nicht geladenen DB; die DB-Nummer liegt im zulässigen Bereich
3Ch		Zugriff auf einen nicht geladenen FC; die FC-Nummer liegt im zulässigen Bereich
3Dh		Zugriff auf eine nicht geladenen SFC; die SFC-Nummer liegt im zulässigen Bereich
3Eh		Zugriff auf einen nicht geladenen FB; die FB-Nummer liegt im zulässigen Bereich
3Fh		Zugriff auf einen nicht geladenen SFB; die SFB-Nummer liegt im zulässigen Bereich
	OB121_FLT_REG	unzulässige Nummer

OB 122 - Peripheriezugriffsfehler

Beschreibung Das Betriebssystem der CPU ruft den OB 122 auf, wenn beim Zugreifen auf Daten eines Moduls ein Fehler auftritt. Wenn die CPU beispielsweise einen Lesefehler beim Zugriff auf Daten eines I/O-Moduls erkennt, dann ruft das Betriebssystem den OB 122 auf. Ist der OB 122 nicht programmiert, wechselt die CPU in den Betriebszustand STOP. Der OB 122 läuft in derselben Prioritätsklasse wie der unterbrochene Baustein. Sie können deshalb innerhalb des OB 122 auf die Register des unterbrochenen Bausteins zugreifen und diese ggf. mit geänderten Parametern an den OB zurückgeben.

Startereignisse maskieren Die CPU verfügt über die folgenden SFCs, mit denen Sie Startereignisse des OB 122 maskieren und demaskieren können, während ihr Programm bearbeitet wird:

- Die SFC 36 MSK_FLT maskiert bestimmte Fehlercodes.
- Die SFC 37 DMSK_FLT demaskiert die Fehlercodes, die von der SFC 36 maskiert wurden.
- Die SFC 38 READ_ERR liest das Ereignisstatusregister.

Lokaldaten Nachfolgend sehen Sie die Belegung der Startinformationen für den OB 122, die defaultmäßige symbolische Bezeichnung und die Datentypen:

Variable	Datentyp	Beschreibung
OB122_EV_CLASS	BYTE	Ereignisklasse und Kennung: 29h
OB122_SW_FLT	BYTE	Fehlercode: 42h: Peripheriezugriffsfehler - lesend 43h: Peripheriezugriffsfehler - schreibend
OB122_PRIORITY	BYTE	Prioritätsklasse: Prioritätsklasse des OBs, der Fehler verursachte
OB122_OB_NUMBR	BYTE	OB-NR. (122)
OB122_BLK_TYPE	BYTE	hier wird kein gültiger Wert eingetragen
OB122_MEM_AREA	BYTE	Speicherbereich und Zugriffsart: Bit 3 ... 0: Speicherbereich 0: Peripheriebereich; 1: Prozessabbild der Eingänge 2: Prozessabbild der Ausgänge Bit 7 ... 4: Zugriffsart: 0: Bitzugriff, 1: Bytezugriff, 2: Wortzugriff, 3: Doppelwortzugriff
OB122_MEM_ADDR	WORD	Adresse des Fehlers im Speicher
OB122_BLK_NUM	WORD	hier wird kein gültiger Wert eingetragen
OB122_PGR_ADDR	WORD	hier wird kein gültiger Wert eingetragen
OB122_DATE_TIME	DATE_AND_TIME	Datum und Uhrzeit, zu denen der OB angefordert wurde

Informationen, wie Sie auf die Lokaldaten zugreifen können finden Sie in der Beschreibung zum OB 1.

Teil 3 Integrierte SFBs

Überblick

Hier finden Sie die Beschreibung der integrierten System-Funktionsbausteine der VIPA-Standard CPUs der Systeme 100V, 200V, 300V und 500V.

Inhalt

Thema	Seite
Teil 3 Integrierte SFBs.....	3-1
Übersicht.....	3-2
SFB 0 - CTU - Vorwärtszählen.....	3-3
SFB 1 - CTD - Rückwärtszählen.....	3-4
SFB 2 - CTUD - Vorwärts-/Rückwärtszählen.....	3-5
SFB 3 - TP - Impuls erzeugen.....	3-7
SFB 4 - TON - Einschaltverzögerung.....	3-9
SFB 5 - TOF - Ausschaltverzögerung.....	3-11
SFB 32 - DRUM - Schrittschaltwerk.....	3-13
SFB 52 - RDREC - Datensatz aus einem DP-V1-Slave lesen.....	3-18
SFB 53 - WRREC - Datensatz in einen DP-V1-Slave schreiben.....	3-20
SFB 54 - RALRM - Alarm von einem DP-V1-Slave empfangen.....	3-22

Übersicht

Allgemeines

Das Systemprogramm der CPU bietet Ihnen Sonderfunktionen, die Sie durch Aufruf aus FBs, FCs oder OBs nutzen können. Diese Sonderfunktionen sind Bestandteil des Systemprogramms und belegen keinen Platz im Anwenderspeicher. Die Sonderfunktionen können zwar aufgerufen, aber weder gelesen noch geändert werden.

Der Aufruf einer Sonderfunktion aus FB, FC oder OB gilt als Bausteinwechsel und beeinflusst die Schachtelungstiefe für Bausteine.

Integrierte SFBs

Folgende System-Funktionsbausteine (SFBs) stehen zur Verfügung:

SFB	Bez.	Beschreibung
SFB 0	CTU	Vorwärtszählen
SFB 1	CTD	Rückwärtszählen
SFB 2	CTUD	Vorwärts- und Rückwärtszählen
SFB 3	TP	Impuls erzeugen
SFB 4	TON	Einschaltverzögerung erzeugen
SFB 5	TOF	Ausschaltverzögerung erzeugen
SFB 32	DRUM	Realisieren eines Schrittschaltwerks mit maximal 16 Schritten
SFB 52	RDREC	DP-V1-SFB Datensatz aus dem DP-Slave lesen
SFB 53	WRREC	DP-V1-SFB Datensatz in einen DP-Slave schreiben
SFB 54	RALRM	DP-V1-SFB Alarm von einem DP-Slave empfangen

SFB 0 - CTU - Vorwärtzählen

Beschreibung Mit dem SFB 0 CTU können Sie vorwärts zählen. Hierbei haben Sie folgendes Verhalten:

- Wechselt der Signalzustand am Vorwärtzählereingang *CU* von "0" auf "1" (positive Flanke), wird der aktuelle Zählerstand um 1 erhöht und am Ausgang *CV* angezeigt.
- Beim ersten Mal Aufrufen und bei $R=0$ entspricht der Zählwert dem Vorbesetztwert am Eingang *PV*.
- Erreicht der Zählwert die obere Grenze 32767, wird er nicht mehr erhöht und jede weitere steigende Flanke am Eingang *CU* bleibt ohne Wirkung.
- Der Zähler wird auf Null zurückgesetzt bzw. gehalten, solange *R* Signalzustand "1" führt.
- Der Ausgang *Q* führt Signalzustand "1", wenn $CV \geq PV$ ist.
- Falls die Instanzen des SFB 0 nach Neustart initialisiert werden sollen, müssen Sie im OB 100 die zu initialisierenden Instanzen mit $R = 1$ aufrufen.

Parameter

Parameter	Deklaration	Datentyp	Speicherbereich	Beschreibung
CU	INPUT	BOOL	E, A, M, D, L, Konstante	Zähleingang
R	INPUT	BOOL	E, A, M, D, L, Konstante	Rücksetzeingang. <i>R</i> dominiert gegenüber <i>CU</i> .
PV	INPUT	INT	E, A, M, D, L, Konstante	Vorbesetztwert. Zur Wirkung von <i>PV</i> siehe Parameter <i>Q</i> .
Q	OUTPUT	BOOL	E, A, M, D, L	Status des Zählers
CV	OUTPUT	INT	E, A, M, D, L	aktueller Zählwert

CU Zähleingang:
Mit einer steigenden Flanke (gegenüber dem letzten SFB-Aufruf) am Zähleingang *CU* wird der Zählwert um 1 erhöht.

R Rücksetzeingang:
Signal "1" am Eingang *R* bewirkt das Rücksetzen des Zählers auf den Wert 0, unabhängig davon welcher Wert an *CU* anliegt.

PV Vorbesetztwert (Vergleichswert):
Ist der aktuelle Zählerstand größer oder gleich diesem Vorbesetztwert wird der Ausgang *Q* gesetzt.

Q Status des Zählers:

- *Q* ist "1" wenn $CV \geq PV$ (aktueller Zählwert \geq Vorbesetztwert)
- *Q* ist sonst "0"

CV aktueller Zählerwert

- Mögliche Zählerwerte: 0 ... 32 767

SFB 1 - CTD - Rückwärtszählen

Beschreibung Mit dem SFB 1 können Sie rückwärts zählen. Hierbei haben Sie folgendes Verhalten:

- Wechselt der Signalzustand am Rückwärtszählereingang *CD* von "0" auf "1" (positive Flanke), wird der aktuelle Zählerstand um 1 erniedrigt und am Ausgang *CV* angezeigt.
- Beim ersten Mal Aufrufen und bei *LOAD* = "0" entspricht der Zählwert dem Vorbesetztwert am Eingang *PV*.
- Erreicht der Zählwert die untere Grenze -32768, wird er nicht mehr erniedrigt und jede weitere steigende Flanke am Eingang *CU* bleibt ohne Wirkung.
- Der Zähler wird auf den Vorbesetztwert *PV* gesetzt bzw. gehalten, solange *LOAD* Signalzustand "1" führt.
- Der Ausgang *Q* führt Signalzustand "1", wenn $CV \leq 0$ ist.
- Falls die Instanzen des SFB 1 nach Neustart initialisiert werden sollen, müssen Sie im OB 100 die zu initialisierenden Instanzen mit *LOAD* = 1 und *PV* = Anfangswert aufrufen.

Parameter

Parameter	Deklaration	Datentyp	Speicherbereich	Beschreibung
CD	INPUT	BOOL	E, A, M, D, L, Konst.	Zähleingang
LOAD	INPUT	BOOL	E, A, M, D, L, Konst.	Ladeeingang <i>LOAD</i> dominiert gegenüber <i>CD</i>
PV	INPUT	INT	E, A, M, D, L, Konst.	Vorbesetztwert
Q	OUTPUT	BOOL	E, A, M, D, L	Status des Zählers
CV	OUTPUT	INT	E, A, M, D, L	aktueller Zählwert

CD Zähleingang:
Mit einer steigenden Flanke (gegenüber dem letzten SFB-Aufruf) am Zähleingang *CD* wird der Zählwert um 1 erniedrigt.

LOAD Ladeeingang:
Signal "1" am Eingang *LOAD* bewirkt, dass der Zähler auf den Vorbesetztwert *PV* voreingestellt wird, unabhängig davon welcher Wert an *CD* anliegt.

PV Vorbesetztwert:
Der Zähler wird auf *PV* voreingestellt, wenn am Eingang *LOAD* Signal "1" ansteht.

Q Status des Zählers

- *Q* ist "1" wenn $0 \geq CV$ (aktueller Zählwert kleiner/gleich 0)
- *Q* ist sonst "0"

CV aktueller Zählerwert

- Mögliche Zählerwerte: -32 768 ... 32 767

SFB 2 - CTUD - Vorwärts-/Rückwärtszählen

Beschreibung

Mit dem SFB 2 können Sie vorwärts und rückwärts zählen. Hierbei haben Sie folgendes Verhalten:

- Wechselt der Signalzustand am Vorwärtszählereingang *CU* von "0" auf "1" (positive Flanke), wird der aktuelle Zählerstand um 1 erhöht und am Ausgang *CV* angezeigt.
- Wechselt der Signalzustand am Rückwärtszählereingang *CD* von "0" auf "1" (positive Flanke), wird der aktuelle Zählerstand um 1 erniedrigt und am Ausgang *CV* angezeigt.
- Zeigen beide Zählereingänge eine positive Flanke, ändert sich der aktuelle Zählwert nicht.
- Erreicht der Zählwert die obere Grenze 32767, wird er nicht mehr erhöht und jede weitere steigende Flanke am Eingang *CU* bleibt ohne Wirkung.
- Erreicht der Zählwert die untere Grenze -32768, wird er nicht mehr erniedrigt und jede weitere steigende Flanke am Eingang *CU* bleibt ohne Wirkung.
- Der Zähler wird auf den Vorbesetztwert *PV* gesetzt bzw. gehalten, solange *LOAD* Signalzustand "1" führt.
- Der Zähler wird auf Null zurückgesetzt bzw. gehalten, solange *R* Signalzustand "1" führt. Solange der Eingang *R* Signalzustand "1" führt, bleiben positive Signalfanken und den Zählereingängen und Signalzustand "1" am Eingang *LOAD* ohne Wirkung.
- Der Ausgang *QU* führt Signalzustand "1", wenn $CV \geq PV$ ist.
- Der Ausgang *QD* führt Signalzustand "1", wenn $CV \leq 0$ ist.
- Falls die Instanzen des SFB 2 nach Neustart initialisiert werden sollen, müssen Sie im OB 100 die zu initialisierenden Instanzen aufrufen:
 - bei Verwendung als Vorwärtszähler mit $R = "1"$
 - bei Verwendung als Rückwärtszähler mit $R = "0"$, $LOAD = "1"$ und $PV =$ Anfangswert aufrufen.

Parameter

Parameter	Deklaration	Datentyp	Speicherbereich	Beschreibung
CU	INPUT	BOOL	E, A, M, D, L, Konstante	Vorwärtszähleingang
CD	INPUT	BOOL	E, A, M, D, L, Konstante	Rückwärtszähleingang
R	INPUT	BOOL	E, A, M, D, L, Konstante	Rücksetzeingang, <i>R</i> dominiert gegenüber <i>LOAD</i>
LOAD	INPUT	BOOL	E, A, M, D, L, Konstante	Ladeingang, <i>LOAD</i> dominiert gegenüber <i>CU</i> und <i>CD</i>
PV	INPUT	INT	E, A, M, D, L, Konstante	Vorbesetztwert
QU	OUTPUT	BOOL	E, A, M, D, L	Status des Vorwärtszählers
QD	OUTPUT	BOOL	E, A, M, D, L	Status des Rückwärtszählers
CV	OUTPUT	INT	E, A, M, D, L	aktueller Zählwert

CU	<p>Vorwärts Zähleingang: Mit einer steigenden Flanke (gegenüber dem letzten SFB-Aufruf) am Zähleingang <i>CU</i> wird der Zählwert um 1 erhöht.</p>
CD	<p>Rückwärts Zähleingang: Mit einer steigenden Flanke (gegenüber dem letzten SFB-Aufruf) am Zähleingang <i>CD</i> wird der Zählwert um 1 erniedrigt.</p>
R	<p>Rücksetzeingang: Signal "1" am Eingang <i>R</i> bewirkt das Rücksetzen des Zählers auf den Wert 0, unabhängig davon welcher Wert an <i>CU</i>, <i>CD</i> oder <i>LOAD</i> anliegt.</p>
LOAD	<p>Ladeeingang: Signal "1" am Eingang <i>LOAD</i> bewirkt, dass der Zähler auf den Vorbesetzwert <i>PV</i> voreingestellt wird, unabhängig davon welche Werte an <i>CU</i> und <i>CD</i> anliegen.</p>
PV	<p>Vorbesetzwert: Der Zähler wird auf <i>PV</i> voreingestellt, wenn am Eingang <i>LOAD</i> Signal "1" ansteht.</p>
QU	<p>Status des Vorwärtzählers:</p> <ul style="list-style-type: none">• <i>QU</i> ist "1" wenn $CV \geq PV$ (aktueller Zählwert \geq Vorbesetzwert)• <i>QU</i> ist sonst "0"
QD	<p>Status des Rückwärtzählers</p> <ul style="list-style-type: none">• <i>QD</i> ist "1" wenn $0 \geq CV$ (aktueller Zählwert kleiner/gleich 0)• <i>QD</i> ist sonst "0"
CV	<p>aktueller Zählerwert</p> <ul style="list-style-type: none">• Mögliche Zählerwerte: -32 768 ... 32 767

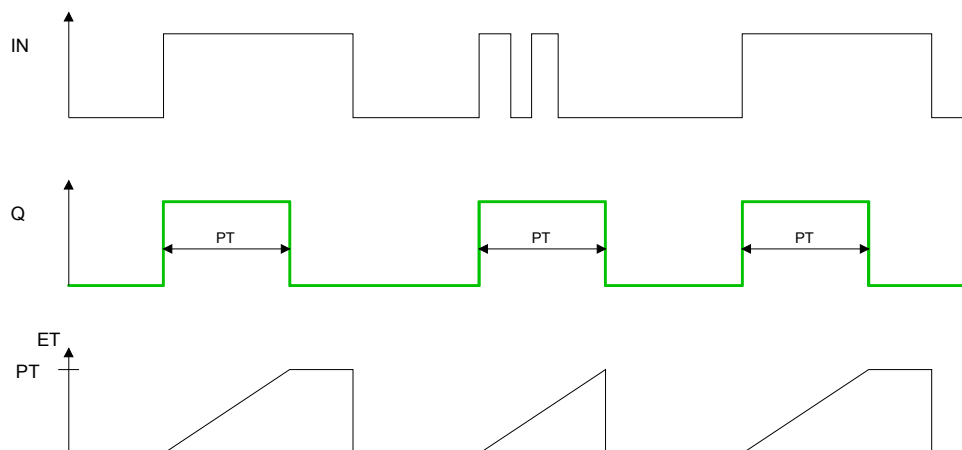
SFB 3 - TP - Impuls erzeugen

Beschreibung

Der SFB 3 erzeugt einen Impuls mit der Länge *PT*. Hierbei haben Sie folgendes Verhalten:

- Die Zeit läuft nur in den Betriebszuständen ANLAUF und RUN.
- Eine steigende Flanke am Eingang *IN* bewirkt den Start des Impulses.
- Der Ausgang *Q* bleibt für die Zeitdauer *PT* gesetzt unabhängig vom Eingangssignal.
- Der Ausgang *ET* liefert die Zeit während der der Ausgang *Q* bereits gesetzt ist. Er kann maximal den Wert des Eingangs *PT* annehmen. Er wird zurückgesetzt, wenn der Eingang *IN* nach "0" wechselt, jedoch frühestens nach Ablauf der Zeit *PT*.
- Falls die Instanzen des SFB 3 nach Neustart initialisiert werden sollen, müssen Sie im OB 100 die zu initialisierenden Instanzen mit *PT* = 0ms aufrufen.

Zeitdiagramm



Parameter

Parameter	Deklaration	Datentyp	Speicherbereich	Beschreibung
IN	INPUT	BOOL	E, A, M, D, L, Konstante	Starteingang
PT	INPUT	TIME	E, A, M, D, L, Konstante	Zeitdauer des Impulses
Q	OUTPUT	BOOL	E, A, M, D, L	Status der Zeit
ET	OUTPUT	TIME	E, A, M, D, L	abgelaufene Zeit

IN	Starteingang: Eine steigende Flanke am Eingang <i>IN</i> bewirkt den Start des Impulses.
PT	Zeitdauer: Die Zeitdauer muss positiv sein. Der Wertebereich ist durch den Datentyp TIME festgelegt.
Q	Ausgang Q: Der Ausgang Q bleibt für die Zeitdauer von <i>PT</i> gesetzt, unabhängig vom weiteren Verlauf des Eingangssignals.
ET	abgelaufene Zeit Der Ausgang <i>ET</i> liefert die Zeit während der Ausgang Q bereits gesetzt ist. Er kann maximal den Wert des Eingangs <i>PT</i> annehmen. Er wird zurückgesetzt, wenn der Eingang <i>IN</i> nach 0 wechselt, jedoch frühestens nach Ablauf der Zeit <i>PT</i> .

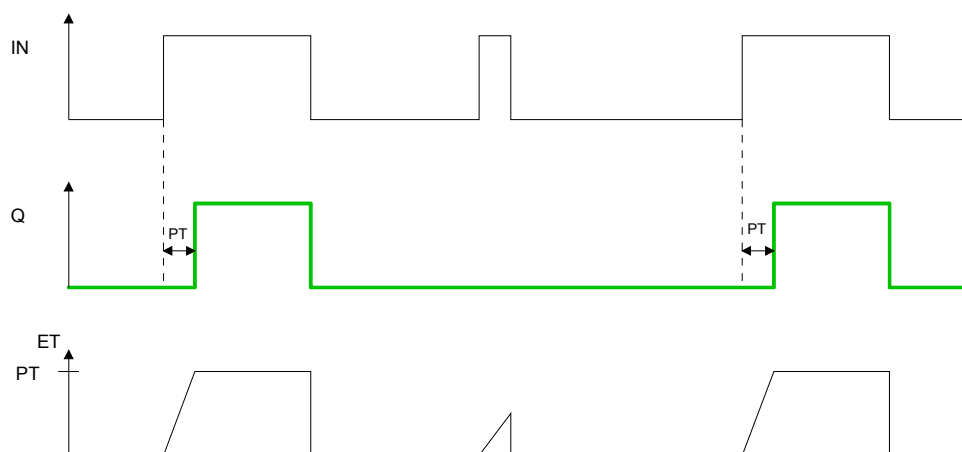
SFB 4 - TON - Einschaltverzögerung

Beschreibung

Der SFB 4 verzögert eine steigende Flanke um die Zeit *PT*. Hierbei haben Sie folgendes Verhalten:

- Die Zeit läuft nur in den Betriebszuständen ANLAUF und RUN.
- Eine steigende Flanke am Eingang *IN* hat nach Ablauf der Zeitdauer *PT* eine steigende Flanke am Ausgang *Q* zur Folge. *Q* bleibt so lange gesetzt, bis der Eingang *IN* nach "0" wechselt. Falls der Eingang *IN* nach "0" wechselt, bevor die Zeit *PT* abgelaufen ist, bleibt der Ausgang *Q* auf "0".
- Der Ausgang *ET* liefert die Zeit, die seit der letzten steigenden Flanke am Eingang *IN* vergangen ist, jedoch höchstens bis zum Wert des Eingangs *PT*. *ET* wird zurückgesetzt, wenn der Eingang *IN* nach "0" wechselt.
- Falls Instanzen dieses SFB nach Neustart initialisiert sein sollen, müssen Sie im OB 100 die zu initialisierenden Instanzen mit *PT* = 0 ms aufrufen.

Zeitdiagramm



Parameter

Parameter	Deklaration	Datentyp	Speicherbereich	Beschreibung
IN	INPUT	BOOL	E, A, M, D, L, Konstante	Starteringang
PT	INPUT	TIME	E, A, M, D, L, Konstante	Zeitdauer
Q	OUTPUT	BOOL	E, A, M, D, L	Status der Zeit
ET	OUTPUT	TIME	E, A, M, D, L	abgelaufene Zeit

IN	<p>Starteingang: Eine steigende Flanke am Eingang <i>IN</i> hat nach Ablauf der Zeitdauer <i>PT</i> eine steigende Flanke am Ausgang <i>Q</i> zur Folge.</p>
PT	<p>Zeitdauer: Zeitdauer, um die die steigende Flanke am Eingang <i>IN</i> verzögert wird. <i>PT</i> muss positiv sein. Der Wertebereich ist durch den Datentyp <i>TIME</i> festgelegt.</p>
Q	<p>Ausgang <i>Q</i>: Eine steigende Flanke am Eingang <i>IN</i> hat nach Ablauf der Zeitdauer <i>PT</i> eine steigende Flanke am Ausgang <i>Q</i> zur Folge. Falls der Eingang <i>IN</i> nach "0" wechselt bevor die Zeit <i>PT</i> abgelaufen ist, bleibt der Ausgang <i>Q</i> auf "0".</p>
ET	<p>abgelaufene Zeit: Der Ausgang <i>ET</i> liefert die Zeit, die seit der letzten steigenden Flanke am Eingang <i>IN</i> vergangen ist, jedoch höchstens bis zum Wert des Eingangs <i>PT</i>. <i>ET</i> wird zurückgesetzt, wenn der Eingang <i>IN</i> nach "0" wechselt.</p>

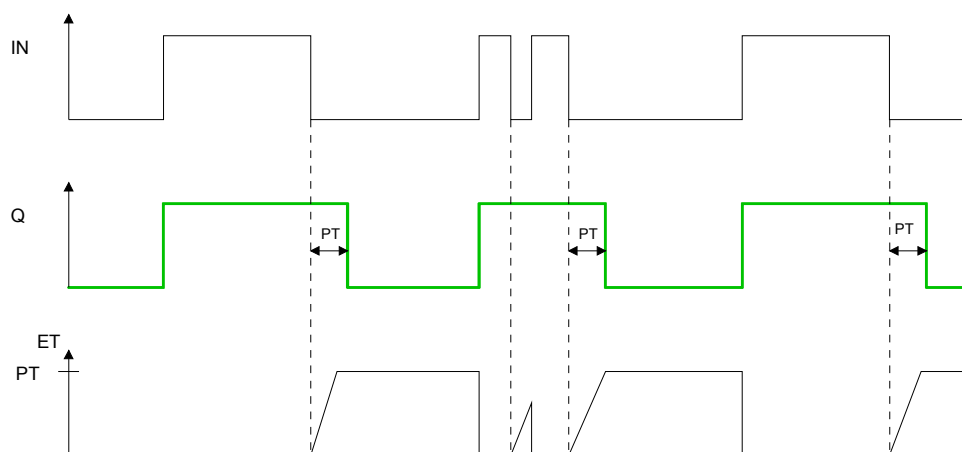
SFB 5 - TOF - Ausschaltverzögerung

Beschreibung

Der SFB 5 verzögert eine fallende Flanke um die Zeit *PT*. Hierbei haben Sie folgendes Verhalten:

- Die Zeit läuft nur in den Betriebszuständen ANLAUF und RUN.
- Eine steigende Flanke am Eingang *IN* bewirkt eine steigende Flanke am Ausgang *Q*. Eine fallende Flanke am Eingang *IN* hat nach Ablauf der Zeitdauer *PT* eine fallende Flanke am Ausgang *Q* zur Folge. Falls der Eingang *IN* wieder nach "1" wechselt, bevor die Zeit *PT* abgelaufen ist, bleibt der Ausgang *Q* auf "1".
- Der Ausgang *ET* liefert die Zeit, die seit der letzten fallenden Flanke am Eingang *IN* vergangen ist, jedoch höchstens bis zum Wert des Eingangs *PT*. *ET* wird zurückgesetzt, wenn der Eingang *IN* nach "1" wechselt.
- Falls die Instanzen des SFB 5 nach Neustart initialisiert werden sollen, müssen Sie im OB 100 die zu initialisierenden Instanzen mit *PT* = 0ms aufrufen.

Zeitdiagramm



Parameter

Parameter	Deklaration	Datentyp	Speicherbereich	Beschreibung
IN	INPUT	BOOL	E, A, M, D, L, Konstante	Starteingang
PT	INPUT	TIME	E, A, M, D, L, Konstante	Zeitdauer
Q	OUTPUT	BOOL	E, A, M, D, L	Status der Zeit
ET	OUTPUT	TIME	E, A, M, D, L	abgelaufene Zeit

- IN** Starteingang:
Eine steigende Flanke am Eingang *IN* bewirkt eine steigende Flanke am Ausgang *Q*. Eine fallende Flanke am Eingang *IN* hat nach Ablauf der Zeitdauer *PT* eine fallende Flanke am Ausgang *Q* zur Folge. Falls der Eingang *IN* wieder nach "1" wechselt, bevor die Zeit *PT* abgelaufen ist, bleibt der Ausgang *Q* auf "1".
- PT** Zeitdauer:
Zeitdauer, um die die fallende Flanke am Eingang *IN* verzögert wird. *PT* muss positiv sein. Der Wertebereich ist durch den Datentyp TIME festgelegt.
- Q** Ausgang *Q*:
Eine steigende Flanke am Eingang *IN* bewirkt eine steigende Flanke am Ausgang *Q*. Eine fallende Flanke am Eingang *IN* hat nach Ablauf der Zeitdauer *PT* eine fallende Flanke am Ausgang *Q* zur Folge. Falls der Eingang *IN* wieder nach "1" wechselt, bevor die Zeit *PT* abgelaufen ist, bleibt der Ausgang *Q* auf "1".
- ET** abgelaufene Zeit:
Der Ausgang *ET* liefert die Zeit, die seit der letzten fallenden Flanke am Eingang *IN* vergangen ist, jedoch höchstens bis zum Wert des Eingangs *PT*. *ET* wird zurückgesetzt, wenn der Eingang *IN* nach "1" wechselt.

SFB 32 - DRUM - Schrittschaltwerk

Beschreibung

Der SFB 32 realisiert ein Schrittschaltwerk mit maximal 16 Schritten.

Die Nummer des ersten Schrittes geben Sie mit dem Parameter *DSP* und die Nummer des letzten Schrittes mit dem Parameter *LST_STEP* vor.

In jedem Schritt werden alle 16 Ausgangsbits *OUT0 ... OUT15* und der Ausgangsparameter *OUT_WORD* (Ausgangsbits zusammengefasst) beschrieben.

Der SFB 32 schaltet in den nächsten Schritt, wenn am Eingang *JOG* gegenüber dem vorangegangenen SFB-Aufruf eine positive Flanke aufgetreten ist. Falls sich der SFB bereits im letzten Schritt befindet, werden bei positiver Flanke an *JOG* die Variablen *Q* und *EOD* gesetzt, *DCC* erhält den Wert 0, und der SFB verbleibt im letzten Schritt, bis Sie den Eingang *RESET* mit "1" belegen.

Zeitabhängiges Weiterschalten

Wenn Sie ein zeitabhängiges Weiterschalten in den nächsten Schritt möchten, belegen Sie den Parameter *DRUM_EN* mit "1".

Der Übergang in den nächsten Schritt erfolgt dann, wenn

- das zum aktuellen Schritt zugehörige Ereignisbit *EVENTi* gesetzt ist und
- die für den aktuellen Schritt vorgegebene Zeit abgelaufen ist.

Die Zeit ergibt sich aus dem Produkt von der Zeitbasis *DTBP* und dem für den aktuellen Schritt gültigen Zeitfaktor (Parameter *S_PRESET*).



Hinweis!

Die im aktuellen Schritt noch verbleibende Bearbeitungszeit *DCC* wird nur dann reduziert, wenn das zugehörige Ereignisbit *EVENTi* gesetzt ist.

Wenn beim Aufruf des SFB am Eingang *RESET* "1" anliegt, geht das Schrittschaltwerk in den Schritt, dessen Nummer Sie dem Eingang *DSP* zugewiesen haben.

**Hinweis!**

Falls Sie *DRUM_EN* zu "1" gewählt haben, erreichen Sie den Sonderfall:

- einer rein zeitgesteuerten Weiterschaltung der Schritte, indem Sie $EVENT_i = "1"$ wählen für $DSP \leq i \leq LST_STEP$.
- einer rein ereignisgesteuerten Weiterschaltung der Schritte über die Ereignisbits $EVENT_i$, indem Sie Zeitbasis $DTBP = "0"$ wählen.

Zusätzlich können Sie das Schrittschaltwerk jederzeit (auch bei $DRUM_EN = "1"$) über den Eingang *JOG* weiterschalten.

Beim Erstaufruf des Bausteins müssen Sie den Eingang *RESET* mit "1" belegen.

Wenn sich das Schaltwerk im letzten Schritt befindet (*DSC* hat den Wert *LST_STEP*) und die für diesen Schritt vorgegebene Bearbeitungszeit abgelaufen ist, werden die Ausgänge *Q* und *EOD* gesetzt, und der SFB verbleibt im letzten Schritt, bis Sie den Eingang *RESET* mit "1" belegen.

Der SFB 32 läuft nur in den Betriebszuständen ANLAUF und RUN.

Falls der SFB 32 nach Neustart initialisiert werden soll, müssen Sie ihn im OB 100 mit $RESET = "1"$ aufrufen.

Parameter

Parameter	Deklaration	Datentyp	Speicherbereich	Beschreibung
RESET	INPUT	BOOL	E, A, M, D, L, Konstante	Rücksetzen
JOG	INPUT	BOOL	E, A, M, D, L, Konstante	Weiterschalten in den nächsten Schritt
DRUM_EN	INPUT	BOOL	E, A, M, D, L, Konstante	Steuerparameter
LST_STEP	INPUT	BYTE	E, A, M, D, L, Konstante	Nummer des letzten Schritts
EVENT _i , 1 ≤ i ≤ 16	INPUT	BOOL	E, A, M, D, L, Konstante	Ereignisbit Nr. i (gehört zum Schritt i)
OUT _j , 0 ≤ j ≤ 15	OUTPUT	BOOL	E, A, M, D, L	Ausgangsbit Nr. j
Q	OUTPUT	BOOL	E, A, M, D, L	Zustandsparameter.
OUT_WORD	OUTPUT	WORD	E, A, M, D, L, P	Ausgangsbits
ERR_CODE	OUTPUT	WORD	E, A, M, D, L, P	Tritt während der Bearbeitung des SFB ein Fehler auf, enthält <i>ERR_CODE</i> die Fehlerinformation.
JOG_HIS	VAR	BOOL	E, A, M, D, L, Konstante	Nicht anwenderrelevant
EOD	VAR	BOOL	E, A, M, D, L, Konstante	Identisch zum Ausgangsparameter Q
DSP	VAR	BYTE	E, A, M, D, L, P, Konstante	Nummer des ersten Schritts
DSC	VAR	BYTE	E, A, M, D, L, P, Konstante	Nummer des aktuellen Schritts
DCC	VAR	DWORD	E, A, M, D, L, P, Konstante	noch verbleibende Bearbeitungszeit in ms
DTBP	VAR	WORD	E, A, M, D, L, P, Konstante	Zeitbasis in ms
PREV_TIME	VAR	DWORD	E, A, M, D, L, Konstante	Nicht anwenderrelevant
S_PRESET	VAR	ARRAY of WORD	E, A, M, D, L, Konstante	Zeitfaktoren für jeden Schritt.
OUT_VAL	VAR	ARRAY of BOOL	E, A, M, D, L, Konstante	in jedem Schritt auszugebenden Werte
S_MASK	VAR	ARRAY of BOOL	E, A, M, D, L, Konstante	Maskenbits für jeden Schritt

RESET	Rücksetzen: Der Pegel 1 bewirkt das Rücksetzen des Schrittschaltwerks. Beim Erstaufruf des Bausteins ist <i>RESET</i> mit "1" zu belegen.
JOG	Eine steigende Flanke (gegenüber dem letzten SFB-Aufruf) bewirkt das Weiterschalten in den nächsten Schritt, falls sich das Schaltwerk noch nicht im letzten Schritt befindet. Die Weiterschaltung erfolgt unabhängig davon, welchen Wert Sie <i>DRUM_EN</i> zugewiesen haben.
DRUM_EN	Steuerparameter, der festlegt, ob ein zeitabhängiges Weiterschalten in den nächsten Schritt möglich sein soll ("1": zeitabhängiges Weiterschalten möglich).
LST_STEP	Nummer des letzten Schrittes: <ul style="list-style-type: none">• mögliche Werte 1 ... 16
EVENT_i , 1 ≤ i ≤ 16	Ereignisbit Nr. i (gehört zum Schritt i)
OUT_j 0 ≤ j ≤ 15	Ausgangsbit NR. j (identisch mit dem Bit Nr. j von <i>OUT_WORD</i>)
Q	Zustandsparameter, der angibt, ob die von Ihnen vorgegebene Bearbeitungszeit des letzten Schrittes abgelaufen ist.
OUT_WORD	Ausgangsbits zusammengefasst in einer Variablen.
ERR_CODE	Tritt während der Bearbeitung des SFB ein Fehler auf, enthält <i>ERR_CODE</i> die Fehlerinformation (siehe Tabelle).
JOG_HIS	Nicht anwenderrelevant: Eingangsparameter <i>JOG</i> des vorhergehenden SFB-Aufruf.
EOD	Identisch zum Ausgangsparameter Q.
DSP	Nummer des ersten Schrittes: <ul style="list-style-type: none">• mögliche Werte 1 ... 16
DSC	Nummer des aktuellen Schrittes
DCC	Die im aktuellen Schritt noch verbleibende Bearbeitungszeit in ms (nur relevant, falls <i>DRUM_EN</i> = "1" und das zugehörige Ereignisbit = "1").
DTBP	Die für alle Schritte gültige Zeitbasis in ms.
PREV_TIME	Nicht anwenderrelevant: Systemzeit des vorhergehenden SFB-Aufrufs.
S_PRESET	Eindimensionales Feld mit den Zeitfaktoren für alle Schritte. <ul style="list-style-type: none">• Sinnvolle Wahl der Indizes: [1 ... 16]. In diesem Fall enthält <i>S_PRESET</i> [x] den Zeitfaktor des Schrittes x.

OUT_VAL Zweidimensionales Feld mit den in jedem Schritt auszugebenden Werten, falls Sie diese nicht über *S_MASK* ausgeblendet haben.

- Sinnvolle Wahl der Indizes: [1 ... 16, 0 ... 15].

In diesem Fall enthält *OUT_VAL* [x, y] den Wert, der dem Ausgangsbit *OUTy* im Schritt x zugewiesen wird.

S_MASK Zweidimensionales Feld mit den Maskenbits für jeden Schritt.

- Sinnvolle Wahl der Indizes: [1 ... 16, 0 ... 15].

In diesem Fall enthält *S_MASK* [x, y] das Maskenbit für den y-ten auszugebenden Wert im Schritt x.

Bedeutung der Maskenbits:

- 0: der entsprechende Wert des vorangegangenen Schrittes wird dem zugehörigen Ausgangsbit zugewiesen
- 1: der entsprechende Wert *aus OUT_VAL* wird dem zugehörigen Ausgangsbit zugewiesen.

Fehlerinformation *ERR_CODE*

Im Fehlerfall verbleibt der SFB 32 im aktuellen Zustand und folgender Fehlercode wird ausgegeben:

ERR_CODE	Beschreibung
0000h	kein Fehler
8081h	unzulässiger Wert für <i>LST_STEP</i>
8082h	unzulässiger Wert für <i>DSC</i>
8083h	unzulässiger Wert für <i>DSP</i>
8084h	Das Produkt $DCC = DTBP \times S_PRESET[DSC]$ überschreitet den Wert 2^{31-1} (ca. 24,86 Tage)

SFB 52 - RDREC - Datensatz aus einem DP-V1-Slave lesen



Hinweis!

Die Schnittstelle des SFB 52 ist identisch mit der des in der Norm "PROFIBUS Guideline PROFIBUS Communication and Proxy Function Blocks according to IEC 61131-3" definierten FB RDREC.

Beschreibung

Mit dem SFB 52 RDREC (read record) lesen Sie den Datensatz mit der Nummer *INDEX* von der mittels *ID* adressierten Komponente (Modul) eines DP-Slaves.

Mit *MLEN* geben Sie vor, wie viele Bytes Sie maximal lesen möchten. Den Zielbereich *RECORD* sollten Sie daher mindestens *MLEN* Bytes lang wählen.

Den Wert TRUE des Ausgangsparameters *VALID* zeigt an, dass der Datensatz erfolgreich in den Zielbereich *RECORD* übertragen wurde. In diesem Fall erhält der Ausgangsparameter *LEN* die Länge der gelesenen Daten in Bytes.

Falls bei der Datensatzübertragung ein Fehler auftrat, wird dies über den Ausgangsparameter *ERROR* angezeigt. Der Ausgangsparameter *STATUS* enthält in diesem Fall die Fehlerinformation.

Systembedingt ist dieser Baustein nicht unterbrechbar!



Hinweis!

Wenn ein DP-V1-Slave über GSD-Datei (GSD ab Rev. 3) projektiert ist und die DP-Schnittstelle des DP-Masters auf Siemens "S7-kompatibel" eingestellt ist, dürfen im Anwenderprogramm keine Datensätze mit SFB 52 von den E/A-Modulen gelesen werden. Der DP-Master adressiert in diesem Fall den falschen Steckplatz (projektiertes Steckplatz +3).

Abhilfe: Schnittstelle des DP-Masters auf "DP-V1" umstellen!

Funktion

Der SFB 52 RDREC ist ein asynchron arbeitender SFB, d.h. die Bearbeitung erstreckt sich über mehrere SFB-Aufrufe. Sie starten die Datensatzübertragung, indem Sie den SFB 52 mit *REQ* = 1 aufrufen.

Über die Ausgangsparameter *BUSY* und die Bytes 2 und 3 des Ausgangsparameters *STATUS* wird der Zustand des Auftrages angezeigt. Dabei entsprechen die Bytes 2 und 3 von *STATUS* dem Ausgangsparameter *RET_VAL* der asynchron arbeitenden SFCs (siehe auch Bedeutung von *REQ*, *RET_VAL* und *BUSY* bei asynchron arbeitenden SFCs).

Die Datensatzübertragung ist abgeschlossen, wenn der Ausgangsparameter *BUSY* den Wert FALSE angenommen hat.

Parameter

Parameter	Deklaration	Datentyp	Speicherbereich	Beschreibung
REQ	INPUT	BOOL	E, A, M, D, L, Konstante	<i>REQ</i> = 1: Datensatzübertragung durchführen
ID	INPUT	DWORD	E, A, M, D, L, Konstante	Logische Adresse der DP-Slave-Komponente (Modul) Bei einem Ausgabemodul muss Bit 15 gesetzt werden (z.B. für Adresse 5: <i>ID</i> = 8005h). Bei einem Mischmodul ist die kleinere der beiden Adressen anzugeben.
INDEX	INPUT	INT	E, A, M, D, L, Konstante	Datensatznummer
MLEN	INPUT	INT	E, A, M, D, L, Konstante	Maximale Länge der zu lesenden Datensatzinformation in Bytes
VALID	OUTPUT	BOOL	E, A, M, D, L	Neuer Datensatz wurde empfangen und ist gültig
BUSY	OUTPUT	BOOL	E, A, M, D, L	<i>BUSY</i> = 1: Der Lesevorgang ist noch nicht beendet.
ERROR	OUTPUT	BOOL	E, A, M, D, L	<i>ERROR</i> = 1: Beim Lesevorgang trat ein Fehler auf.
STATUS	OUTPUT	DWORD	E, A, M, D, L	Aufrufkennung (Bytes 2 und 3) bzw. Fehlercode
LEN	OUTPUT	INT	E, A, M, D, L	Länge der gelesenen Datensatzinformation
RECORD	IN_OUT	ANY	E, A, M, D, L	Zielbereich für den gelesenen Datensatz.

**Fehler-
informationen**

Siehe Alarm von einem DP-Slave empfangen mit dem SFB 54 RALRM.

SFB 53 - WRREC - Datensatz in einen DP-V1-Slave schreiben



Hinweis!

Die Schnittstelle des SFB 53 ist identisch mit der des in der Norm "PROFIBUS Guideline PROFIBUS Communication and Proxy Function Blocks according to IEC 61131-3" definierten FB WRREC.

Beschreibung

Mit dem SFB 53 WRREC (write record) übertragen Sie den Datensatz *RECORD* zu der mittels *ID* adressierten Komponente (Modul) eines DP-Slaves.

Mit *LEN* geben Sie die Länge des zu übertragenden Datensatzes in Bytes vor.

Den Wert TRUE des Ausgangsparameters *DONE* zeigt an, dass der Datensatz erfolgreich zum DP-Slave übertragen wurde.

Falls bei der Datensatzübertragung ein Fehler auftrat, wird dies über den Ausgangsparameter *ERROR* angezeigt. Der Ausgangsparameter *STATUS* enthält in diesem Fall die Fehlerinformation.

Systembedingt ist dieser Baustein nicht unterbrechbar!



Hinweis!

Wenn ein DP-V1-Slave über GSD-Datei (GSD ab Rev. 3) projektiert ist und die DP-Schnittstelle des DP-Masters auf Siemens "S7-kompatibel" eingestellt ist, dürfen im Anwenderprogramm keine Datensätze mit SFB 53 zu den E/A-Modulen geschrieben werden. Der DP-Master adressiert in diesem Fall den falschen Steckplatz (projektiertes Steckplatz +3).

Abhilfe: Schnittstelle des DP-Masters auf "DP-V1" umstellen!

Funktion

Der SFB 53 WRREC ist ein asynchron arbeitender SFB, d.h. die Bearbeitung erstreckt sich über mehrere SFB-Aufrufe. Sie starten die Datensatzübertragung, indem Sie den SFB 53 mit *REQ* = 1 aufrufen.

Über die Ausgangsparameter *BUSY* und die Bytes 2 und 3 des Ausgangsparameters *STATUS* wird der Zustand des Auftrages angezeigt. Dabei entsprechen die Bytes 2 und 3 von *STATUS* dem Ausgangsparameter *RET_VAL* der asynchron arbeitenden SFCs (siehe auch Bedeutung von *REQ*, *RET_VAL* und *BUSY* bei asynchron arbeitenden SFCs).

Beachten Sie, dass Sie dem Aktualparameter von *RECORD* bei allen zu ein und demselben Auftrag gehörenden Aufrufen des SFB 53 denselben Wert zuweisen. Dasselbe gilt für die Aktualparameter von *LEN*.

Die Datensatzübertragung ist abgeschlossen, wenn der Ausgangsparameter *BUSY* den Wert FALSE angenommen hat.

Parameter

Parameter	Deklaration	Datentyp	Speicherbereich	Beschreibung
REQ	INPUT	BOOL	E, A, M, D, L, Konstante	<i>REQ</i> = 1: Datensatzübertragung durchführen
ID	INPUT	DWORD	E, A, M, D, L, Konstante	Logische Adresse der DP-Slave-Komponente (Modul). Bei einem Ausgabemodul muss Bit 15 gesetzt werden (z.B. für Adresse 5: <i>ID</i> = 8005h). Bei einem Mischmodul ist die kleinere der beiden Adressen anzugeben.
INDEX	INPUT	INT	E, A, M, D, L, Konstante	Datensatznummer
LEN	INPUT	INT	E, A, M, D, L, Konstante	Maximale Länge des zu übertragenden Datensatzes in Bytes
DONE	OUTPUT	BOOL	E, A, M, D, L	Datensatz wurde übertragen
BUSY	OUTPUT	BOOL	E, A, M, D, L	<i>BUSY</i> = 1: Der Schreibvorgang ist noch nicht beendet.
ERROR	OUTPUT	BOOL	E, A, M, D, L	<i>ERROR</i> = 1: Beim Schreibvorgang trat ein Fehler auf.
STATUS	OUTPUT	DWORD	E, A, M, D, L	Aufrufkennung (Bytes 2 und 3) bzw. Fehlercode
LEN	OUTPUT	INT	E, A, M, D, L	Länge der gelesenen Datensatz-information
RECORD	IN_OUT	ANY	E, A, M, D, L	Datensatz

**Fehler-
informationen**

Siehe Alarm von einem DP-Slave empfangen mit dem SFB 54 RALRM.

SFB 54 - RALRM - Alarm von einem DP-V1-Slave empfangen



Hinweis!

Die Schnittstelle des SFB 54 ist identisch mit der des in der Norm "PROFIBUS Guideline PROFIBUS Communication and Proxy Function Blocks according to IEC 61131-3" definierten FB RALRM.

Beschreibung

Der SFB 54 RALRM empfängt einen Alarm samt der zugehörigen Information von einem Peripheriemodul bzw. einer Komponente eines DP-Slaves und stellt diese Information an seinen Ausgangsparametern zur Verfügung.

Die Information in den Ausgangsparametern enthält sowohl die Startinformation des aufgerufenen OB als auch Informationen aus der Alarmquelle.

Rufen Sie den SFB 54 nur innerhalb desjenigen Alarm-OBs auf, den das Betriebssystem der CPU aufgrund des zu untersuchenden Alarms aus der Peripherie gestartet hat.



Hinweis!

Wenn Sie den SFB 54 RALRM in einem OB aufrufen, dessen Starterereignis kein Alarm aus der Peripherie ist, stellt Ihnen der SFB an seinen Ausgängen entsprechend weniger Informationen zur Verfügung.

Achten Sie darauf, dass Sie beim Aufruf des SFB 54 in verschiedenen OBs unterschiedliche Instanz-DBs verwenden. Falls Sie die aus einem SFB 54 Aufruf resultierenden Daten außerhalb des zugehörigen Alarm-OBs auswerten, sollten Sie sogar pro OB-Starterereignis einen eigenen Instanz-DB benutzen.

Parameter

Parameter	Deklaration	Datentyp	Speicherbereich	Beschreibung
<i>MODE</i>	INPUT	INT	E, A, M, D, L, Konstante	Betriebsart
<i>F_ID</i>	INPUT	DWORD	E, A, M, D, L, Konstante	Logische Anfangsadresse der Komponente (Modul), von der Alarme empfangen werden sollen
<i>MLEN</i>	INPUT	INT	E, A, M, D, L, Konstante	Maximale Länge der zu empfangenen Alarminformation in Bytes
<i>NEW</i>	OUTPUT	BOOL	E, A, M, D, L	Ein Alarm wurde empfangen
<i>STATUS</i>	OUTPUT	DWORD	E, A, M, D, L	Fehlercode des SFB bzw. des DP-Masters
<i>ID</i>	OUTPUT	DWORD	E, A, M, D, L	Logische Anfangsadresse der Komponente (Modul), von der ein Alarm empfangen wurde Bit 15 enthält die E/A-Kennung: 0: bei einer Eingangsadresse 1: bei einer Ausgangsadresse
<i>LEN</i>	OUTPUT	INT	E, A, M, D, L	Länge der empfangenen Alarminformation
<i>TINFO</i>	IN_OUT	ANY	E, A, M, D, L	(Task Information) Zielbereich für OB-Startinformation und Verwaltungsinformation
<i>AINFO</i>	IN_OUT	ANY	E, A, M, D, L	(Alarm Information) Zielbereich für Kopfinformation und Alarmzusatzinformation. Für <i>AINFO</i> sollten Sie eine Länge von mindestens <i>MLEN</i> Bytes vorsehen.

MODE

Den SFB 54 können Sie in drei verschiedenen Betriebsarten (*MODE*) aufrufen:

- 0: zeigt die alarmlösende Komponente im Ausgangsparameter *ID* an und beschreibt den Ausgangsparameter *NEW* mit TRUE.
- 1: beschreibt sämtliche Ausgangsparameter unabhängig von der alarm-lösenden Komponente.
- 2: prüft, ob die im Eingangsparameter *F_ID* angegebene Komponente den Alarm ausgelöst hat.
 - falls nein, erhält *NEW* den Wert FALSE
 - falls ja, erhält *NEW* den Wert TRUE, und alle anderen Ausgangsparameter werden beschrieben.

**Hinweis!**

Falls Sie den Zielbereich *TINFO* bzw. *AINFO* zu kurz wählen, kann der SFC 54 nicht die gesamte Information eintragen.

TINFO Datenstruktur des Zielbereichs (Task Information):

Byte	Datentyp	Beschreibung				
0 ... 19		Startinformation des OB, in dem der SFC 54 aktuell aufgerufen wurde: Byte 0 ... 11: strukturiert wie der Parameter <i>TOP_SI</i> der SFC 6 RD_SINFO Byte 12 ... 19: Datum und Uhrzeit, zu denen der OB angefordert wurde				
20 ... 27		Verwaltungsinformation:				
20	Byte	zentral:	0			
		dezentral:	DP-Mastersystem- <i>ID</i> (mögliche Werte: 1 ... 255)			
21	Byte	zentral:	Baugruppenträgernummer (mögliche Werte: 0 ... 31)			
		dezentral:	Nummer der DP-Station (mögliche Werte: 0 ... 127)			
22	Byte	zentral:	0			
		dezentral:	Bit 3 ... 0	Slavetyp	0000:	DP
					0001:	DPS7
					0010:	DPS7 V1
					0011:	DP-V1
					ab 0100:	reserviert
		dezentral:	Bit 7 ... 4	Profiltyp	0000:	DP
ab 0001:	reserviert					
23	Byte	zentral:	0			
		dezentral:	Bit 3 ... 0	Alarminfotyp	0000:	Transparent (Alarm kommt von einem projektierten dezentralen Modul)
					0001:	Stellvertreter (Alarm eines nicht-DP-V1 Slaves oder eines nicht projektierten Steckplatzes)
					0010:	Erzeugter Alarm (in der CPU)
					ab 0011:	reserviert
					dezentral:	Bit 7 ... 4
		ab 0001:	reserviert			

Fortsetzung ...

... Fortsetzung TINFO

Byte	Datentyp	Beschreibung	
24	Byte	zentral: 0	
		dezentral: Flags der DP-Master-Anschaltung	
		Bit 0 = 0:	Alarm von einer integrierten DP-Anschaltung
		Bit 0 = 1:	Alarm von einer externen DP-Anschaltung
		Bit 7 ... 1:	reserviert
25	Byte	zentral: 0	
		dezentral: Flags der DP-Slave-Anschaltung	
		Bit 0:	EXT_DIAG_Bit aus dem Diagnose-telegramm bzw. 0, falls dieses Bit beim Alarm nicht vorliegt
		Bit 7 ... 1:	reserviert
26, 27	WORD	zentral: 0	
		dezentral: PROFIBUS Identnummer	

AINFO

Datenstruktur des Zielbereiches (Alarm Information):

Byte	Datentyp	Beschreibung
0 ... 3		Kopfinformation
0	Byte	Länge der empfangenen Alarminformation in Bytes
		zentral: 4 ... 224
		dezentral: 4 ... 63
1	Byte	zentral: reserviert
		dezentral: Kennung für den Alarmtyp
		1: Diagnosealarm
		2: Prozessalarm
		3: Ziehen-Alarm
		4: Stecken-Alarm
		5: Status-Alarm
		6: Update-Alarm
		31: Ausfall eines Erweiterungsgeräts, eines DP-Mastersystems oder einer DP-Station
		32 ... 126: herstellerspezifischer Alarm
2	Byte	Steckplatznummer der alarmauslösenden Komponente
3	Byte	zentral: reserviert
		dezentral: Specifier
		Bit 1, 0:
		00: keine weitere Information
		01: kommendes Ereignis, Steckplatz gestört
		10: gehendes Ereignis, Steckplatz nicht mehr gestört
		11: gehendes Ereignis, Steckplatz weiterhin gestört
		Bit 2: Add_Ack
		Bit 7 ... 3: Sequenznummer
4 ... 223		Alarmzusatzinformation: Modulspezifische Daten zum jeweiligen Alarm:
		zentral: ARRAY[0] ... ARRAY[220]
		dezentral: ARRAY[0] ... ARRAY[59]

TINFO und AINFO

Zielbereich:

Abhängig von dem jeweiligen OB, in dem der SFB 54 aufgerufen wird, werden die Zielbereiche *TINFO* und *AINFO* nur teilweise beschrieben. Welche Information jeweils eingetragen wird, entnehmen Sie der folgenden Tabelle.

Alarmtyp	OB	<i>TINFO</i> OB-Status- Information	<i>TINFO</i> Verwaltungs- Information	<i>AINFO</i> Kopf- Information	<i>AINFO</i> Alarmzusatz- Information
Prozessalarm	4x	ja	ja	ja	zentral: nein
					dezentral: wie vom DP-Slave geliefert
Statusalarm	55	ja	ja	ja	ja
Update-Alarm	56	ja	ja	ja	ja
Hersteller-spezifischer Alarm	57	ja	ja	ja	ja
Peripherie-Redundanzfehler	70	ja	ja	nein	nein
Diagnosealarm	82	ja	ja	ja	zentral: Datensatz 1
					dezentral: wie vom DP-Slave geliefert
Ziehen-Stecken-Alarm	83	ja	ja	ja	zentral: nein
					dezentral: wie vom DP-Slave geliefert
Baugruppen-trägerausfall/ Stationsausfall	86	ja	ja	nein	nein
...	alle anderen OBs	ja	nein	nein	nein

**Fehler-
informationen**

Der Ausgangsparameter *STATUS* enthält Fehlerinformationen. Wird er als *ARRAY[1...4] OF BYTE* interpretiert, hat die Fehlerinformation folgende Struktur:

Feldelement	Name	Beschreibung
STATUS[1]	Function_Num	00h: falls kein Fehler Funktionskennung aus DP-V1-PDU: im Fehlerfall ist 80h aufgeodert. Falls kein DP-V1-Protokollelement benutzt wird: C0h
STATUS[2]	Error Decode	Ort der Fehlerkennung
STATUS[3]	Error_1	Fehlerkennung
STATUS[4]	Error_2	herstellerspezifische Erweiterung der Fehlerkennung <i>STATUS[4]</i> wird bei DP-V1-Fehlern vom DP-Master an die CPU und den SFB durchgereicht. Ohne DP-V1-Fehler wird der Wert auf "0" gesetzt mit folgenden Ausnahmen beim SFB 52: <ul style="list-style-type: none"> • <i>STATUS[4]</i> enthält die Länge des Zielbereichs aus <i>RECORD</i>, falls <i>MLEN</i> > Länge des Zielbereichs aus <i>RECORD</i> • <i>STATUS[4]=MLEN</i>, falls die tatsächliche Datensatzlänge < <i>MLEN</i> < Länge des Zielbereichs aus <i>RECORD</i>

STATUS[2] (Ort der Fehlerkennung) kann folgende Werte annehmen:

Error Decode	Quelle	Beschreibung
00 ... 7Fh	CPU	kein Fehler oder keine Warnung
80h	DP-V1	Fehler nach IEC 61158-6
81h ... 8Fh	CPU	8xh zeigt einen Fehler im x-ten Aufrufparameter des SFB an
FEh, FFh	DP Profile	profilspezifischer Fehler

STATUS/3/ (Fehlerkennung) kann folgende Werte annehmen:

Error Decode	Error Code 1	Erläuterung laut DP-V1	Beschreibung
00h	00h		kein Fehler, keine Warnung
70h	00h	reserved, reject	Erstaufruf; keine Datensatzübertragung aktiv
	01h	reserved, reject	Erstaufruf; Datensatzübertragung angestoßen
	02h	reserved, reject	Zwischenaufruf; Datensatzübertragung ist bereits aktiv
80h	90h	reserved, pass	logische Anfangsadresse ungültig
	92h	reserved, pass	unzulässiger Typ bei ANY-Pointer
	93h	reserved, pass	Die mittels <i>ID</i> bzw. <i>F_ID</i> adressierte DP-Komponente ist nicht konfiguriert.
	A0h	read error	negative Quittung beim Lesen vom Modul
	A1h	write error	negative Quittung beim Schreiben zum Modul
	A2h	module failure	DP-Protokollfehler bei Layer 2
	A3h	reserved, pass	DP-Protokollfehler bei Direct-Data-Link-Mapper oder User-Interface/User
	A4h	reserved, pass	Kommunikation am Bus gestört
	A5h	reserved, pass	-
	A7h	reserved, pass	DP-Slave oder Modul ist beschäftigt (temporärer Fehler)
	A8h	version conflict	DP-Slave oder Modul meldet nicht passende Versionen
	A9h	feature not supported	Funktion wird vom DP-Slave oder Modul nicht unterstützt
	AA ... AFh	user specific	DP-Slave oder Modul meldet einen herstellerspezifischen Fehler seiner Anwendung. Bitte sehen Sie in der Dokumentation des Herstellers des DP-Slaves bzw. des Moduls nach.
	B0h	invalid index	Modul kennt den Datensatz nicht Datensatznummer ≥ 256 ist unzulässig
	B1h	write length error	Die Längenangabe im Parameter <i>RECORD</i> ist falsch; bei SFB 54: Längenfehler in <i>AINFO</i>
	B2h	invalid slot	Der projektierte Steckplatz ist nicht belegt.
	B3h	type conflict	Ist-Modultyp ungleich Soll-Modultyp
B4h	invalid area	DP-Slave oder Modul meldet einen Zugriff auf einen unzulässigen Bereich	
B5h	state conflict	DP-Slave oder Modul ist nicht bereit	
B6h	access denied	DP-Slave oder Modul verweigert den Zugriff	

Fortsetzung ...

... Fortsetzung STATUS[3]

Error Decode	Error Code 1	Erläuterung laut DP-V1	Beschreibung
80h	B7h	invalid range	DP-Slave oder Modul meldet einen unzulässigen Bereich eines Parameters oder eines Werts
	B8h	invalid parameter	DP-Slave oder Modul meldet einen unzulässigen Parameter
	B9h	invalid type	DP-Slave oder Modul meldet einen unzulässigen Typ
	BAh ... BFh	user specific	DP-Slave oder Modul meldet einen herstellerspezifischen Fehler beim Zugriff. Bitte sehen Sie in der Dokumentation des Herstellers des DP-Slaves oder Moduls nach.
	C0h	read constrain conflict	Das Modul führt den Datensatz, aber es sind noch keine Lesedaten da.
	C1h	write constrain conflict	Die Daten des auf dem Modul vorangegangenen Schreibauftrags für denselben Datensatz sind von dem Modul noch nicht verarbeitet.
	C2h	resource busy	Das Modul bearbeitet momentan das mögliche Maximum an Aufträgen für eine CPU.
	C3h	resource unavailable	Die benötigten Betriebsmittel sind momentan belegt.
	C4h		Interner temporärer Fehler. Auftrag konnte nicht ausgeführt werden. Wiederholen Sie den Auftrag. Bei häufigem Auftreten dieses Fehlers überprüfen Sie bitte Ihren Aufbau auf elektrische Störquellen.
	C5h		DP-Slave oder Modul nicht verfügbar
	C6h		Datensatzübertragung wurde abgebrochen wegen Prioritätsklassenabbruchs.
	C7h		Auftragsabbruch wegen Neustart des DP-Masters
	C8h ... CFh		DP-Slave oder Modul meldet einen herstellerspezifischen Fehler zu seinen Ressourcen. Bitte sehen Sie in der Dokumentation des Herstellers des DP-Slaves oder Moduls nach.
	Dxh	user specific	DP-Slave spezifisch, Siehe Beschreibung DP-Slaves.
81h	00h ... FFh		Fehler im 1. Aufrufparameter (bei SFB 54: <i>MODE</i>)
	00h		Unzulässige Betriebsart

Fortsetzung ...

... Fortsetzung STATUS[3]

Error Decode	Error Code 1	Erläuterung laut DP-V1	Beschreibung
82h	00h ... FFh		Fehler im 2. Aufrufparameter.
...
88h	00h ... FFh		Fehler im 8. Aufrufparameter (bei SFB 54: <i>TINFO</i>)
	01h		Syntaxkennung falsch
	23h		Überschreitung des Mengengerüsts oder Zielbereich zu klein
	24h		Bereichskennung falsch
	32h		DB/DI-Nr. außerhalb des Anwenderbereichs
	3Ah		DB/DI-Nr. ist NULL bei Bereichs- kennung DB/DI oder angegebener DB/DI nicht vorhanden.
89h	00h ... FFh		Fehler im 9. Aufrufparameter (bei SFB 54: <i>AINFO</i>)
	01h		Syntaxkennung falsch
	23h		Überschreitung des Mengengerüsts oder Zielbereich zu klein
	24h		Bereichskennung falsch
	32h		DB/DI-Nr. außerhalb des Anwenderbereichs
	3Ah		DB/DI-Nr. ist Null bei Bereichs- kennung DB/DI oder angegebener DB/DI nicht vorhanden
8Ah	00h ... FFh		Fehler im 10. Aufrufparameter
...
8Fh	00h ... FFh		Fehler im 15. Aufrufparameter
FEh, FFh			Profilspezifischer Fehler

Teil 4 Integrierte Standard-SFCs

Überblick

Hier finden Sie die Beschreibung der integrierten Standard-SFCs der VIPA Standard-CPU's der Systeme 100V, 200V, 300V und 500V.

Die Beschreibung der SFCs, die Bestandteil der VIPA-Bibliothek sind, finden Sie im Teil "VIPA -spezifische Bausteine".

Inhalt

Thema	Seite
Teil 4 Integrierte Standard-SFCs	4-1
Übersicht Integrierte Standard-SFCs.....	4-3
Allgemeine und spezifische Fehlercodes RET_VAL.....	4-5
SFC 0 - SET_CLK - Uhrzeit stellen	4-8
SFC 1 - READ_CLK - Uhrzeit lesen	4-9
SFC 2 ... 4 - Betriebsstundenzähler.....	4-10
SFC 2 - SET_RTM - Betriebsstundenzähler setzen	4-11
SFC 3 - CTRL_RTM - Betriebsstundenzähler starten/stoppen	4-12
SFC 4 - READ_RTM - Betriebsstundenzähler auslesen.....	4-13
SFC 5 - GADR_LGC - Logische Adresse eines Kanals ermitteln	4-14
SFC 6 - RD_SINFO - Startinformation auslesen	4-16
SFC 12 - D_ACT_DP - DP-Slave aktivieren und deaktivieren	4-18
SFC 13 - DPNRM_DG - Slave-Diagnosedaten lesen	4-23
SFC 14 - DPRD_DAT - Konsistente Nutzdaten lesen	4-26
SFC 15 - DPWR_DAT - Konsistente Nutzdaten schreiben.....	4-28
SFC 17 - ALARM_SQ und SFC 18 - ALARM_S	4-30
SFC 19 - ALARM_SC - Quittierzustand der letzten Meldung.....	4-33
SFC 20 - BLKMOV - Variable kopieren	4-34
SFC 21 - FILL - Feld vorbesetzen	4-36
SFC 22 - CREAT_DB - Datenbaustein erzeugen	4-38
SFC 23 - DEL_DB - Datenbaustein löschen.....	4-40
SFC 24 - TEST_DB - Datenbaustein testen	4-41
SFC 28 ... SFC 31 - Uhrzeitalarm.....	4-42
SFC 32 - SRT_DINT - Verzögerungsalarm starten	4-46
SFC 33 - CAN_DINT - Verzögerungsalarm stornieren	4-47
SFC 34 - QRY_DINT - Verzögerungsalarm Status abfragen.....	4-48
SFC 36 - MSK_FLT - Synchronfehlerereignisse maskieren	4-49
SFC 37 - DMSK_FLT - Synchronfehlerereignisse demaskieren	4-50
SFC 38 - READ_ERR - Ereignisstatusregister lesen.....	4-51
SFC 39 - DIS_IRT - Alarmereignisse sperren.....	4-52
SFC 40 - EN_IRT - Gesperrte Alarmereignisse freigeben	4-54
SFC 41 - DIS_AIRT - Alarmereignisse verzögern	4-55
SFC 42 - EN_AIRT - Verzögerte Alarmereignissen freigeben	4-56

SFC 43 - RE_TRIGR - Zykluszeitüberwachung neu starten	4-56
SFC 44 - REPL_VAL - Ersatzwert in AKKU1 übertragen.....	4-57
SFC 46 - STP - CPU in STOP überführen.....	4-57
SFC 47 - WAIT - Verzögern des Anwenderprogramms.....	4-58
SFC 49 - LGC_GADR - Steckplatz ermitteln	4-59
SFC 50 - RD_LGADR - Alle Adressen eines Moduls lesen	4-60
SFC 51 - RDSYSST - Auslesen der Informationen der SZL	4-61
SFC 52 - WR_USMSG - Eintrag in Diagnosepuffer schreiben	4-63
SFC 54 - RD_DPARM - Vordefinierte Parameter lesen.....	4-67
SFC 55 - WR_PARM - Dynamische Parameter schreiben	4-69
SFC 56 - WR_DPARM - Vordefinierte Parameter schreiben.....	4-72
SFC 57 - PARM_MOD - Modul parametrieren.....	4-74
SFC 58 - WR_REC - Datensatz schreiben.....	4-76
SFC 59 - RD_REC - Datensatz lesen.....	4-79
SFC 64 - TIME_TCK - Systemzeit lesen	4-82
SFC 65 - X_SEND - Daten senden	4-83
SFC 66 - X_RCV - Daten empfangen	4-86
SFC 67 - X_GET - Daten lesen.....	4-91
SFC 68 - X_PUT - Daten schreiben	4-95
SFC 69 - X_ABORT - Verbindung abbrechen	4-98
SFC 81 - UBLKMOV - Variable ununterbrechbar kopieren.....	4-101

Übersicht Integrierte Standard-SFCs

Standard-SFCs Folgende Standard-System-Funktionen (SFCs) stehen zur Verfügung:

SFC	Bezeichnung	Beschreibung
SFC 0	SET_CLK	Uhrzeit stellen
SFC 1	READ_CLK	Uhrzeit lesen
SFC 2	SET_RTM	Betriebsstundenzähler setzen
SFC 3	CTRL_RTM	Betriebsstundenzähler starten/stoppen
SFC 4	READ_RTM	Betriebsstundenzähler auslesen
SFC 5	GADR_LGC	Logische Adresse eines Kanals ermitteln (nur von Modulen in Rack 0)
SFC 6	RD_SINFO	Startinformation des aktuellen OB auslesen
SFC 12	D_ACT_DP	Aktivieren oder Deaktivieren von DP-Slaves
SFC 13	DPNRM_DG	Slave-Diagnosedaten lesen
SFC 14	DPRD_DAT	Konsistente Nutzdaten lesen (auch von DP-Slaves → DP-Master FW ≥ V3.00)
SFC 15	DPWR_DAT	Konsistente Nutzdaten schreiben (auch auf DP-Slaves → DP-M. FW ≥ V3.00)
SFC 17	ALARM_SQ	Quittierbare bausteinbezogene Meldungen erzeugen
SFC 18	ALARM_S	Nicht quittierbare bausteinbezogene Meldungen erzeugen
SFC 19	ALARM_SC	Quittierzustand der letzten Alarm SQ-gekommenen-Meldung
SFC 20 ¹⁾	BLKMOV	Variable kopieren innerhalb des Arbeitsspeichers
SFC 21 ¹⁾	FILL	Feld vorbesetzen innerhalb des Arbeitsspeichers
SFC 22	CREAT_DB	Datenbaustein erzeugen
SFC 23	DEL_DB	Datenbaustein löschen
SFC 24	TEST_DB	Datenbaustein testen
SFC 28	SET_TINT	Uhrzeitalarm stellen
SFC 29	CAN_TINT	Uhrzeitalarm stornieren
SFC 30	ACT_TINT	Uhrzeitalarm aktivieren
SFC 31	QRY_TINT	Uhrzeitalarm abfragen
SFC 32	SRT_DINT	Verzögerungsalarm starten
SFC 33	CAN_DINT	Verzögerungsalarm stornieren
SFC 34	QRY_DINT	Verzögerungsalarm abfragen
SFC 36	MSK_FLT	Synchronfehlerereignisse maskieren
SFC 37	DMSK_FLT	Synchronfehlerereignisse demaskieren
SFC 38	READ_ERR	Ereignisstatusregister lesen
SFC 39	DIS_IRT	Alarmereignisse sperren
SFC 40	EN_IRT	Gesperrte Alarmereignisse freigeben
SFC 41	DIS_AIRT	Verzögern von Alarmereignissen
SFC 42	EN_AIRT	Verzögern von Alarmereignissen aufheben
SFC 43	RE_TRIGR	Zykluszeitüberwachung nachtriggern
SFC 44	REPL_VAL	Ersatzwert in AKKU1 übertragen
SFC 46	STP	CPU in STOP überführen
SFC 47 ¹⁾	WAIT	Programmbearbeitung zusätzlich zur Wartezeit verzögern
SFC 49	LGC_GADR	Den zu einer logischen Adresse gehörenden Steckplatz ermitteln
SFC 50	RD_LGADR	Sämtliche logische Adressen eines Moduls ermitteln

Fortsetzung ...

... Fortsetzung Standard-SFCs

SFC	Bezeichnung	Beschreibung
SFC 51	RDSYSST	Auslesen der Informationen aus der Systemzustandsliste
SFC 52	WR_USMSG	Anwendereintrag in Diagnosebuffer schreiben (über MPI senden in Vorbereitung)
SFC 54	RD_DPARM	Vordefinierte Parameter lesen
SFC 55	WR_PARM	Dynamische Parameter schreiben (nur mit Analog-, Digitalmodulen, FMs, CPs und über PROFIBUS DP-V1 möglich)
SFC 56	WR_DPARM	Vordefinierte Parameter schreiben (nur mit Analog-, Digitalmodulen, FMs, CPs und über PROFIBUS DP-V1 möglich)
SFC 57	PARM_MOD	Modul parametrieren (nur mit Analog-, Digitalmodulen, FMs, CPs und über PROFIBUS DP-V1 möglich)
SFC 58	WR_REC	Datensatz schreiben (nur mit Analog-, Digitalmodulen, FMs, CPs und über PROFIBUS DP-V1 möglich)
SFC 59	RD_REC	Datensatz lesen (nur mit Analog-, Digitalmodulen, FMs, CPs und über PROFIBUS DP-V1 möglich)
SFC 64 ¹⁾	TIME_TCK	Millisekundentimer auslesen
SFC 65	X_SEND	Daten an externen Partner senden
SFC 66	X_RCV	Daten von externem Partner empfangen
SFC 67	X_GET	Daten aus externem Partner lesen
SFC 68	X_PUT	Daten in externen Partner schreiben
SFC 69	X_ABORT	Verbindung zu externem Partner abbrechen
SFC 81	UBLKMOV	Variable ununterbrechbar kopieren

¹⁾ Dieser Baustein ist unterbrechbar und beeinflusst somit nicht die Alarm-Reaktionszeit.

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 Fehlercodes finden Sie auf der folgenden Seite.
15	Bit 15 = "1": zeigt an, dass ein Fehler aufgetreten ist.

Spezifischer Fehlercode

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

Ein spezifischer Fehlercode besteht aus:

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

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

Allgemeine Fehlercodes RET_VAL

Der Parameter *RET_VAL* verschiedener SFCs liefert keine spezifischen, sondern nur allgemeine Fehlerinformationen zurück.

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

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

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

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.

Allgemeine Fehlercodes

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.
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.

Fortsetzung ...

... Fortsetzung

Fehlercode	Beschreibung
8x28h	Ausrichtungsfehler beim Lesen eines Parameters.
8x29h	Ausrichtungsfehler beim Schreiben eines Parameters. Dieser Fehlercode zeigt an, dass der Verweis auf den Parameter x ein Operand ist, dessen Bitadresse ungleich 0 ist.
8x30h	Der Parameter befindet sich in dem schreibgeschützten Global-DB.
8x31h	Der Parameter befindet sich in dem schreibgeschützten Instanz-DB. Dieser Fehlercode zeigt an, dass der Parameter x sich in einem schreibgeschützten Datenbaustein befindet. Wenn der Datenbaustein von der Systemfunktion selbst geöffnet wurde, gibt die Systemfunktion immer den Wert 8x30h aus.
8x32h	Der Parameter enthält eine zu große DB-Nummer (Nummernfehler des DBs).
8x34h	Der Parameter enthält eine zu große FC-Nummer (Nummernfehler des FCs).
8x35h	Der Parameter enthält eine zu große FB-Nummer (Nummernfehler des FBs). Dieser Fehlercode zeigt an, dass der Parameter x eine Bausteinnummer enthält, die größer ist als die maximal zulässige Bausteinnummer.
8x3Ah	Der Parameter enthält die Nummer eines DBs, der nicht geladen ist.
8x3Ch	Der Parameter enthält die Nummer eines FCs, der nicht geladen ist.
8x3Eh	Der Parameter enthält die Nummer eines FBs, der nicht geladen ist.
8x42h	Es ist ein Zugriffsfehler aufgetreten, während das System einen Parameter aus dem Peripheriebereich der Eingänge auslesen wollte.
8x43h	Es ist ein Zugriffsfehler aufgetreten, während das System einen Parameter in den Peripheriebereich der Ausgänge schreiben wollte.
8x44h	Fehler beim n-ten ($n > 1$) Lesezugriff nach Auftreten eines Fehlers.
8x45h	Fehler beim n-ten ($n > 1$) Schreibzugriff nach Auftreten eines Fehlers. Dieser Fehlercode zeigt an, dass der Zugriff auf den gewünschten Parameter verweigert wird.

SFC 0 - SET_CLK - Uhrzeit stellen

Beschreibung Mit dem SFC 0 SET_CLK (set system clock) werden Uhrzeit und Datum der CPU-Uhr gestellt, wobei die Uhr dann ab der eingestellten Uhrzeit und dem eingestellten Datum läuft.
Handelt es sich um eine Master-Uhr, dann wird beim Aufruf des SFC 0 zusätzlich die Synchronisation der Uhrzeit gestartet. Die Synchronisationsintervalle werden in der Hardwarekonfiguration eingestellt.

Parameter

Parameter	Deklaration	Datentyp	Speicherbereich	Beschreibung
PDT	INPUT	DT	D, L	Über <i>PDT</i> geben Sie das Datum und die Uhrzeit ein, die Sie einstellen möchten.
RET_VAL	OUTPUT	INT	E, A, M, D, L	Tritt während der Bearbeitung der Funktion ein Fehler auf, enthält der Rückgabewert einen Fehlercode.

PDT Datum und Uhrzeit sind als Datentyp DT einzugeben.

Beispiel:

Datum: 27.04.2006, Uhrzeit: 14:15:55 → DT#2006-04-27-14:15:55

Die Uhrzeit kann nur sekundengranular eingegeben werden. Der Wochentag wird vom SFC 0 automatisch aus dem Datum errechnet.

Beachten Sie, dass Sie den Datentyp DT mit dem FC 3 D_TOD_DT erst bilden müssen, bevor Sie ihn dem Eingangsparameter übergeben können (siehe Uhrzeitfunktionen; FC 3, FC 6, FC 7, FC 8, FC 33, FC 40, FC 1, FC 35, FC 34).

RET_VAL (Rückgabewert)

Wert	Beschreibung
0000h	kein Fehler
8080h	Fehler im Datum
8081h	Fehler in der Uhrzeit

SFC 1 - READ_CLK - Uhrzeit lesen

Beschreibung Mit dem SFC 1 READ_CLK (read system clock) wird die Uhr in der CPU ausgelesen. Dadurch erhalten Sie das aktuelle Datum und die Uhrzeit.

Parameter

Parameter	Deklaration	Datentyp	Speicherbereich	Beschreibung
RET_VAL	OUTPUT	INT	E, A, M, D, L	Tritt während der Bearbeitung der Funktion ein Fehler auf, enthält der Rückgabewert einen Fehlercode.
CDT	OUTPUT	DT	D, L	Am Ausgang <i>CDT</i> werden das aktuelle Datum und die aktuelle Uhrzeit ausgegeben.

RET_VAL
(Rückgabewert) Der SFC 1 liefert keine spezifischen Fehlerinformationen aus.

CDT Am Ausgang *CDT* wird das aktuelle Datum und die aktuelle Uhrzeit im Format DT ausgegeben.

SFC 2 ... 4 - Betriebsstundenzähler

Beschreibung	VIPAs-CPU's verfügen über 8 Betriebsstundenzähler.		
	Sie können über:		
	SFC 2	SET_RTM	Betriebsstundenzähler setzen
	SFC 3	CTRL_RTM	Betriebsstundenzähler starten/stoppen
	SFC 4	READ_RTM	Betriebsstundenzähler auslesen
	Über einen Betriebsstundenzähler können Sie:		
	<ul style="list-style-type: none">• die Betriebsdauer der CPU berechnen.• die Betriebsdauer von angesteuerten Betriebsmitteln berechnen.		
Eigenschaften	Mit dem Start beginnt der Betriebsstundenzähler immer ab dem letzten Zählerstand zu zählen. Soll er ab einem anderen Anfangswert beginnen, dann müssen Sie diesen Wert mit dem SFC 2 zuweisen. Geht die CPU in STOP oder Sie stoppen den Betriebsstundenzähler, dann merkt sich die CPU den aktuellen Wert. Bei Neustart der CPU muss der Betriebsstundenzähler erneut mit dem SFC 3 gestartet werden.		
Wertebereich	Der Betriebsstundenzähler hat einen Wertebereich von 0 ... 32767 Stunden.		

SFC 2 - SET_RTM - Betriebsstundenzähler setzen

Beschreibung Mit dem SFC 2 SET_RTM (set run-time meter) wird ein Betriebsstundenzähler der CPU auf einen vorgegebenen Wert gestellt. Bei VIPA-CPU's sind maximal 8 Betriebsstundenzähler verfügbar.

Parameter

Parameter	Deklaration	Datentyp	Speicherbereich	Beschreibung
NR	INPUT	BYTE	E, A, M, D, L, Konstante	Eingang <i>NR</i> enthält die Nummer des Betriebsstundenzählers, den Sie einstellen möchten. Mögliche Werte: 0 ... 7
PV	INPUT	INT	E, A, M, D, L, Konstante	Eingang <i>PV</i> enthält die Einstellung für den Betriebsstundenzähler.
RET_VAL	OUTPUT	INT	E, A, M, D, L	Tritt während der Bearbeitung der Funktion ein Fehler auf, enthält der Rückgabewert einen Fehlercode.

RET_VAL (Rückgabewert)

Wert	Beschreibung
0000h	kein Fehler
8080h	Falsche Nummer des Betriebsstundenzählers
8081h	Ein negativer Wert wurde dem Parameter <i>PV</i> übergeben.

SFC 3 - CTRL_RTM - Betriebsstundenzähler starten/stoppen

Beschreibung Mit dem SFC 3 CTRL_RTM (control run-time meter) wird ein Betriebsstundenzähler in Abhängigkeit des Zustandes von Eingang S gestartet oder gestoppt.

Parameter

Parameter	Deklaration	Datentyp	Speicherbereich	Beschreibung
NR	INPUT	BYTE	E, A, M, D, L, Konstante	Eingang NR enthält die Nummer des Betriebsstundenzählers, den Sie starten bzw. stoppen möchten. Mögliche Werte: 0 ... 7
S	INPUT	BOOL	E, A, M, D, L, Konstante	Eingang S startet bzw. stoppt den Betriebsstundenzähler. Setzen Sie den Signalzustand auf "0", wenn Sie den Zähler stoppen möchten. Setzen Sie den Signalzustand auf "1", wenn Sie den Zähler starten möchten.
RET_VAL	OUTPUT	INT	E, A, M, D, L	Tritt während der Bearbeitung der Funktion ein Fehler auf, enthält der Rückgabewert einen Fehlercode.

RET_VAL (Rückgabewert)

Wert	Beschreibung
0000h	kein Fehler
8080h	Falsche Nummer des Betriebsstundenzählers

SFC 4 - READ_RTM - Betriebsstundenzähler auslesen

Beschreibung Mit dem SFC 4 READ_RTM (read run-time meter) wird ein Betriebsstundenzähler ausgelesen. Als Ausgangsdaten werden die aktuelle Betriebsstundenzahl und der Status des Zählers ("gestoppt" bzw. "zählt") ausgegeben.
Wenn der Betriebsstundenzähler länger als 32767 Stunden läuft, dann bleibt er bei diesem Wert stehen und der Rückgabewert *RET_VAL* enthält die Fehlermeldung "8081h: Überlauf".

Parameter

Parameter	Deklaration	Datentyp	Speicherbereich	Beschreibung
NR	INPUT	BYTE	E, A, M, D, L, Konstante	Eingang <i>NR</i> enthält die Nummer des Betriebsstundenzählers, den Sie auslesen möchten. Mögliche Werte: 0 ... 7
RET_VAL	OUTPUT	INT	E, A, M, D, L	Tritt während der Bearbeitung der Funktion ein Fehler auf, enthält der Rückgabewert einen Fehlercode.
CQ	OUTPUT	BOOL	E, A, M, D, L	Ausgang <i>CQ</i> gibt an, ob der Betriebsstundenzähler läuft oder angehalten ist. <ul style="list-style-type: none"> "0": Betriebsstundenzähler gestoppt "1": Betriebsstundenzähler läuft
CV	OUTPUT	INT	E, A, M, D, L	Ausgang <i>CV</i> gibt den aktuellen Wert des Betriebsstundenzählers an.

RET_VAL (Rückgabewert)

Wert	Beschreibung
0000h	kein Fehler
8080h	Falsche Nummer des Betriebsstundenzählers
8081h	Überlauf des Betriebsstundenzählers

SFC 5 - GADR_LGC - Logische Adresse eines Kanals ermitteln

Beschreibung Mit dem SFC 5 GADR_LGC (convert geographical address to logical address) ermitteln Sie die logische Adresse des Moduls.

Parameter

Parameter	Deklaration	Datentyp	Speicherbereich	Beschreibung
SUBNETID	INPUT	BYTE	E, A, M, D, L, Konstante	Bereichskennung
RACK	INPUT	WORD	E, A, M, D, L, Konstante	Nr. des Racks
SLOT	INPUT	WORD	E, A, M, D, L, Konstante	Steckplatz-Nummer
SUBSLOT	INPUT	BYTE	E, A, M, D, L, Konstante	Submodulsteckplatz
SUBADDR	INPUT	WORD	E, A, M, D, L, Konstante	Offset im Nutzdatenadressraum des Moduls
RET_VAL	OUTPUT	INT	E, A, M, D, L	Tritt während der Bearbeitung der Funktion ein Fehler auf, enthält der Rückgabewert einen Fehlercode.
IOID	OUTPUT	BYTE	E, A, M, D, L	Bereichskennung: siehe unten
LADDR	OUTPUT	WORD	E, A, M, D, L	Logische Basisadresse des Moduls

- SUBNETID** Bereichskennung:
- "0": falls das Modul lokal gesteckt ist (inklusive Zeilenanschlaltung)
 - DP-Mastersystem-ID des zugehörigen dezentralen Peripheriesystems, falls sich der Steckplatz in einem dezentralen Peripheriegerät befindet.
- Rack** Nr. des Racks, falls Bereichskennung 0
Stations-Nr. des dezentralen Peripheriegeräts, falls Bereichskennung > 0.
- SLOT** Steckplatz-Nummer
- SUBSLOT** Submodulsteckplatz
(falls kein Submodul gesteckt werden kann, ist hier 0 anzugeben).
- SUBADDR** Offset im Nutzdatenadressraum des Moduls.

RET_VAL (Rückgabewert) Tritt während der Bearbeitung der Funktion ein Fehler auf, enthält der Rückgabewert einen Fehlercode.

Wert	Beschreibung
0000h	kein Fehler
8094h	Es wurde kein Subnetz mit der angegebenen <i>SUBNETID</i> konfiguriert.
8095h	Unzulässiger Wert beim Parameter <i>RACK</i>
8096h	Unzulässiger Wert beim Parameter <i>SLOT</i>
8097h	Unzulässiger Wert beim Parameter <i>SUBSLOT</i>
8098h	Unzulässiger Wert beim Parameter <i>SUBADDR</i>
8099h	Der Steckplatz ist nicht projektiert.
809Ah	Die Subadresse für den ausgewählten Steckplatz ist nicht projektiert.

IOID Bereichskennung:

- 54h: Peripherie-Eingang (PE)
- 55h: Peripherie-Ausgang (PA)

Im Fall eines Mischmoduls liefert der SFC die Bereichskennung der niedrigeren Adresse. Bei gleichen Adressen liefert der SFC die Kennung 54h.

LADDR Logische Basisadresse des Moduls.

SFC 6 - RD_SINFO - Startinformation auslesen

Beschreibung Mit dem SFC 6 RD_SINFO (read start information) werden die Startinformationen des zuletzt aufgerufenen OBs, der noch nicht vollständig abgearbeitet wurde, und des zuletzt gestarteten Anlauf-OBs ausgelesen. Beide Startinformationen enthalten keinen Zeitstempel. Erfolgt der Aufruf im OB 100, dann werden zwei identische Startinformationen zurückgeliefert.

Parameter

Parameter	Deklaration	Datentyp	Speicherbereich	Beschreibung
RET_VAL	OUTPUT	INT	E, A, M, D, L	Tritt während der Bearbeitung der Funktion ein Fehler auf, enthält der Rückgabewert einen Fehlercode.
TOP_SI	OUTPUT	STRUCT	D, L	Startinformation des aktuellen OBs
START_UP_SI	OUTPUT	STRUCT	D, L	Startinformation des zuletzt gestarteten Anlauf-OBs

TOP_SI und START_UP_SI Hierbei handelt es sich um zwei identisch aufgebaute Strukturen, deren Aufbau nachfolgend dargestellt ist.

Strukturelement	Datentyp	Beschreibung
EV_CLASS	BYTE	Bits 3 ... 0: Ereigniskennung Bits 7 ... 4: Ereignisklasse: 1: Startereignisse von Standard-OBs 2: Startereignisse von Synchronfehler-OBs 3: Startereignisse von Asynchronfehler-OBs
EV_NUM	BYTE	Ereignisnummer
PRIORITY	BYTE	Nummer der Ablauebene Das Strukturelement PRIORITY liefert die zum aktuellen OB gehörige Prioritätsklasse.
NUM	BYTE	OB-Nummer NUM enthält die Nummer des aktuellen OBs bzw. des zuletzt gestarteten Anlauf-OBs.
TYP2_3	BYTE	Datenkennung 2_3: kennzeichnet die in ZI2_3 eingetragene Information
TYP1	BYTE	Datenkennung 1: kennzeichnet die in ZI1 eingetragene Information
ZI1	WORD	Zusatzinformation 1
ZI2_3	DWORD	Zusatzinformation 2_3

**Hinweis!**

Die Strukturelemente entsprechen inhaltlich genau den temporären Variablen eines OBs.

Die temporären Variablen können in den einzelnen OBs andere Namen und andere Datentypen haben. Beachten Sie, dass die Aufrufsstelle der OBs zusätzlich Datum und Uhrzeit enthält.

**RET_VAL
(Rückgabewert)**

Der SFC 6 liefert keine spezifischen, sondern nur allgemeine Fehlerinformationen zurück.

Beispiel

Der zuletzt aufgerufene, noch nicht vollständig abgearbeitete OB, ist der OB 80, der zuletzt gestartete Anlauf-OB ist der OB 100.

Die folgende Tabelle zeigt die Zuordnung zwischen den Strukturelementen des Parameters *TOP_SI* des SFC 6 und den lokalen Variablen des OB 80.

TOP_SI Strukturelement	Datentyp	Lokale Variable	Datentyp
EV_CLASS	BYTE	OB100_EV_CLASS	BYTE
EV_NUM	BYTE	OB80_FLT_ID	BYTE
PRIORITY	BYTE	OB80_PRIORITY	BYTE
NUM	BYTE	OB80_OB_NUMBR	BYTE
TYP2_3	BYTE	OB80_RESERVED_1	BYTE
TYP1	BYTE	OB80_RESERVED_2	BYTE
ZI1	WORD	OB80_ERROR_INFO	WORD
ZI2_3	DWORD	OB80_ERR_EV_CLASS OB80_ERR_EV_NUM OB80_OB_PRIORITY OB80_OB_NUM	BYTE BYTE BYTE BYTE

Die folgende Tabelle zeigt die Zuordnung zwischen den Strukturelementen des Parameters *START_UP_SI* des SFC 6 und den lokalen Variablen des OB 100.

START_UP_SI Strukturelement	Datentyp	Lokale Variable	Datentyp
EV_CLASS	BYTE	OB100_EV_CLASS	BYTE
EV_NUM	BYTE	OB100_STRTUP	BYTE
PRIORITY	BYTE	OB100_PRIORITY	BYTE
NUM	BYTE	OB100_OB_NUMBR	BYTE
TYP2_3	BYTE	OB100_RESERVED_1	BYTE
TYP1	BYTE	OB100_RESERVED_2	BYTE
ZI1	WORD	OB100_STOP	WORD
ZI2_3	DWORD	OB100_STRT_INFO	DWORD

SFC 12 - D_ACT_DP - DP-Slave aktivieren und deaktivieren

Beschreibung Mit dem SFC 12 D_ACT_DP können Sie projektierte DP-Slaves gezielt deaktivieren und wieder aktivieren. Darüber hinaus können Sie für jeden eingesetzten DP-Slave ermitteln, ob dieser momentan aktiviert oder deaktiviert ist.

Der SFC 12 ist nicht anwendbar auf PROFIBUS PA-Feldgeräte, die über DP/PA Link an ein DP-Mastersystem angeschlossen sind.



Hinweis!

So lange ein oder mehrere SFC 12 Aufträge aktiv sind, können Sie keine geänderte Konfiguration vom PG in die CPU laden. Während des Ladens einer geänderten Konfiguration vom PG in die CPU weist die CPU die Aktivierung eines SFC 12-Auftrages ab.

Anwendung Wenn Sie in einer CPU DP-Slaves konfigurieren, die real nicht vorhanden sind oder aktuell nicht benötigt werden, greift die CPU dennoch regelmäßig auf diese DP-Slaves zu. Nach deren Deaktivierung unterbleiben weitere CPU-Zugriffe. Dadurch kann der schnellstmögliche DP-Buszyklus erreicht werden, und die entsprechende Fehlerereignisse treten nicht mehr auf.

Beispiel Sämtliche mögliche Maschinenoptionen sind vom Hersteller als DP-Slaves projektiert, um ein gemeinsames Anwenderprogramm über alle möglichen Optionen erstellen und pflegen zu können. Mit dem SFC 12 können Sie im Anlauf der Maschine alle nicht vorhandenen DP-Slaves deaktivieren.

Arbeitsweise Der SFC 12 ist ein asynchron arbeitender SFC, d.h. die Bearbeitung erstreckt sich über mehrere SFC-Aufrufe. Sie starten den Auftrag, indem Sie den SFC 12 mit *REQ = 1* aufrufen.
Über die Ausgangsparameter *RET_VAL* und *BUSY* wird der Zustand des Auftrages angezeigt.

Identifikation eines Auftrages Falls Sie einen Deaktivierungs- bzw. Aktivierungsauftrag angestoßen haben und den SFC 12 erneut aufrufen, bevor dieser beendet wurde, hängt das weitere Verhalten des SFC entscheidend davon ab, ob es sich beim erneuten Aufruf um denselben Auftrag handelt. Stimmt der Eingangsparameter *LADDR* überein, so gilt der SFC-Aufruf als Folgeaufruf.

Deaktivieren von DP-Slaves

Wenn Sie einen DP-Slave mit dem SFC 12 deaktivieren, werden dadurch dessen Prozessausgänge auf die projektierten Ersatzwerte bzw. auf "0" gesetzt (sicherer Zustand).

Der zugehörige DP-Master spricht diesen DP-Slave im weiteren nicht mehr an. Deaktivierte DP-Slaves werden an den Fehler-LEDs des DP-Masters oder der CPU nicht als gestört oder fehlend gekennzeichnet.

Das Prozessabbild der Eingänge von deaktivierten DP-Slaves wird mit 0 aktualisiert, d.h. es wird wie bei ausgefallenen DP-Slaves behandelt.

**Hinweis!**

Sie können bei VIPA nicht alle DP-Slaves deaktivieren. Mindestens 1 Slave muss am Bus aktiviert bleiben.

Falls Sie in Ihrem Programm mittels Direktzugriff auf die Nutzdaten eines zuvor deaktivierten DP-Slaves zugreifen, wird der Peripheriezugriffsfehler-OB (OB 122) aufgerufen und das zugehörige Startereignis in den Diagnosepuffer eingetragen.

Falls Sie per SFC (z.B. SFC 59 RD_REC) auf einen deaktivierten DP-Slave zugreifen, erhalten Sie in *RET_VAL* dieselbe Fehlerinformation wie bei einem nicht verfügbaren DP-Slave.

Das Deaktivieren eines DP-Slaves verursacht keinen Start des Programmablauffehler-OB 85, auch wenn dessen Ein- bzw. Ausgänge zum systemseitig zu aktualisierenden Prozessabbild gehören. Es erfolgt auch kein Eintrag in den Diagnosepuffer.

Das Deaktivieren eines DP-Slaves hat keinen Start des Slave-Ausfall-OB 86 zur Folge, und das Betriebssystem veranlasst auch keinen Eintrag in den Diagnosepuffer.

Fällt eine DP-Station aus, nachdem Sie diese mit dem SFC 12 deaktiviert haben, wird der Ausfall vom Betriebssystem nicht erkannt. Es erfolgt daher weder ein OB 86-Start noch ein Diagnosepuffereintrag. Der Stationsausfall wird erst beim erneuten Aktivieren der Station festgestellt und Ihnen über den zugehörigen *RET_VAL* mitgeteilt.

Falls Sie DP-Slaves deaktivieren wollen, die als Sender am Querverkehr beteiligt sind, wird empfohlen, zuerst die Empfänger (Mithörer) zu deaktivieren, die mithören, welche Eingangsdaten der Sender seinem DP-Master schickt. Erst im Anschluss daran deaktivieren Sie den Sender.

Aktivieren von DP-Slaves

Wenn Sie einen DP-Slave mit dem SFC 12 wieder aktivieren, wird dieser vom zugehörigen DP-Master konfiguriert und parametrierung (wie bei der Wiederkehr einer ausgefallenen DP-Station). Die Aktivierung ist abgeschlossen, wenn der Slave Nutzdaten transferieren kann.

Das Aktivieren eines DP-Slaves verursacht keinen Start des Programmablauffehler-OB 85, auch wenn dessen Ein- bzw. Ausgänge zum systemseitig zu aktualisierenden Prozessabbild gehören. Es erfolgt auch kein Eintrag in den Diagnosepuffer.

Das Aktivieren eines DP-Slaves hat keinen Start des Slave-Ausfall-OB 86 zur Folge, und das Betriebssystem veranlasst auch keinen Eintrag in den Diagnosepuffer.

Wenn Sie versuchen, einen deaktivierten Slave, der physikalisch vom DP-Bus getrennt ist, mit dem SFC 12 zu aktivieren läuft eine Überwachungszeit von 10sec ab. Nach Ablauf dieser Überwachungszeit liefert der SFC den Fehlercode 80A2h, und der Slave bleibt deaktiviert. Falls er zu einem späteren Zeitpunkt wieder Verbindung zum DP-Bus hat, müssen Sie den Slave mit dem SFC 12 wieder aktivieren.

**Hinweis!**

Das Aktivieren eines DP-Slaves kann geraume Zeit dauern. Falls Sie einen laufenden Aktivierungsauftrag abbrechen wollen, starten Sie den SFC 12 mit dem gleichen Wert für *LADDR* und *MODE* = 2. Sie wiederholen den Aufruf des SFC 12 mit *MODE* = 2 so lange, bis der erfolgreiche Abbruch des Aktivierungsauftrags mit *RET_VAL* = 0 angezeigt wird.

Falls Sie DP-Slaves aktivieren wollen, die am Querverkehr beteiligt sind, wird empfohlen, zuerst die Sender und anschließend die Empfänger (Mithörer) zu aktivieren.

CPU-Anlauf

Bei einem Neustart werden deaktivierte Slaves automatisch wieder aktiviert. Nach dem CPU-Anlauf versucht die CPU zyklisch zu allen projektierten und nicht deaktivierten Slaves, die nicht vorhanden oder nicht ansprechbar sind, Kontakt aufzunehmen.

**Hinweis!**

Ein Aufruf des SFC 12 im Anlauf-OB 100 wird nicht unterstützt.

Parameter

Parameter	Deklaration	Datentyp	Speicherbereich	Beschreibung
REQ	INPUT	BOOL	E, A, M, D, L, Konstante	Pegelgetriggertes Steuerparameter <i>REQ</i> = 1: Aktivieren bzw. Deaktivieren durchführen
MODE	INPUT	BYTE	E, A, M, D, L, Konstante	Auftragskennung Mögliche Werte: 0: Auskunft einholen, ob der angesprochene DP-Slave aktiviert oder deaktiviert ist. 1: DP-Slave aktivieren 2: DP-Slave deaktivieren
LADDR	INPUT	WORD	E, A, M, D, L, Konstante	Beliebige logische Adresse des DP-Slaves
RET_VAL	OUTPUT	INT	E, A, M, D, L	Tritt während der Bearbeitung der Funktion ein Fehler auf, enthält der Rückgabewert einen Fehlercode.
BUSY	OUTPUT	BOOL	E, A, M, D, L	Aktivkennung: <i>BUSY</i> = 1: Der Auftrag ist noch aktiv. <i>BUSY</i> = 0: Der Auftrag wurde beendet.

RET_VAL**(Rückgabewert)**

Wert	Beschreibung
0000h	Der Auftrag wurde fehlerfrei durchgeführt.
0001h	DP-Slave aktiviert (Dieser Fehlercode ist nur möglich bei <i>MODE</i> = 0.)
0002h	DP-Slave deaktiviert (Dieser Fehlercode ist nur möglich bei <i>MODE</i> = 0.)
7000h	Erstaufruf mit <i>REQ</i> = 0. Der über <i>LADDR</i> festgelegte Auftrag ist nicht aktiv; <i>BUSY</i> hat den Wert 0.
7001h	Erstaufruf mit <i>REQ</i> = 1. Der über <i>LADDR</i> festgelegte Auftrag wurde angestoßen; <i>BUSY</i> hat den Wert 1.
7002h	Zwischenaufruf (<i>REQ</i> irrelevant). Der aktivierte Auftrag ist noch in Bearbeitung; <i>BUSY</i> hat den Wert 1.
8090h	Sie haben kein Modul mit der in <i>LADDR</i> angegebenen Adresse projiziert. Sie betreiben Ihre CPU als I-Slave und haben in <i>LADDR</i> eine Adresse dieses Slaves angegeben.

Fortsetzung ...

... Fortsetzung

Wert	Beschreibung
8092h	Der laufende Deaktivierungsvorgang eines DP-Slaves (<i>MODE</i> = 2) kann nicht durch seine Aktivierung (<i>MODE</i> = 1) abgebrochen werden. Aktivieren Sie den DP-Slave zu einem späteren Zeitpunkt.
8093h	Zu der in <i>LADDR</i> angegebenen Adresse gehört kein DP-Slave (es liegt keine Projektierung vor), oder der Parameter <i>MODE</i> ist nicht bekannt.
80A1h	Der angesprochene DP-Slave konnte nicht parametrierbar werden. (Dieser Fehlercode ist nur möglich bei <i>MODE</i> = 1.) Hinweis! Diese Fehlerinformation liefert der SFC nur dann, wenn während der Parametrierung des aktivierten Slaves dieser wieder ausfällt. Wenn nur die Parametrierung eines einzelnen Moduls nicht erfolgreich war, liefert der SFC die Fehlerinformation 0000h.
80A2h	Der angesprochene DP-Slave gibt keine Rückmeldung.
80A3h	Der betroffene DP-Master unterstützt diese Funktion nicht.
80A4h	Die CPU unterstützt diese Funktion bei externen DP-Mastern nicht.
80A6h	Steckplatzfehler im DP-Slave; es kann nicht auf alle Nutzdaten zugegriffen werden. (Dieser Fehlercode ist nur möglich bei <i>MODE</i> = 1.) Hinweis! Diese Fehlerinformation liefert der SFC nur dann, wenn nach der Parametrierung des aktivierten Slaves und vor dem Ende des SFC der Slave wieder ausfällt. Wenn nur ein einzelnes Modul nicht verfügbar ist, liefert der SFC die Fehlerinformation 0000h.
80C1h	Der SFC 12 wurde gestartet und wird mit einer anderen logischen Adresse fortgesetzt. (Dieser Fehlercode ist nur möglich bei <i>MODE</i> = 1.)
80C3h	<ul style="list-style-type: none"> • Temporärer Ressourcenfehler: die CPU bearbeitet momentan das mögliche Maximum an Aktivierungs-/Deaktivierungsaufträgen. (Dieser Fehlercode ist nur möglich bei <i>MODE</i> = 1 und <i>MODE</i> = 2.) • Die CPU erhält gerade eine geänderte Konfiguration. Das Aktivieren und Deaktivieren von DP-Slaves ist daher momentan nicht möglich.
F001h	Es dürfen nicht alle Slaves deaktiviert werden. Mindestens 1 Slave muss noch aktiv bleiben.
F002h	Unbekannte Slaveadresse

SFC 13 - DPNRM_DG - Slave-Diagnosedaten lesen

Beschreibung

Mit dem SFC 13 DPNRM_DG (read diagnosis data of a DP slave) werden die aktuellen Diagnosedaten eines DP-Slaves gelesen. Die Diagnosedaten eines jeden DP-Slave sind durch EN 50 170 Volume 2, PROFIBUS festgelegt.

Durch den Eingangsparameter *RECORD* wird der Zielbereich festgelegt, in den die gelesenen Daten nach fehlerfreier Datenübertragung eingetragen werden. Der Lesevorgang wird gestartet, wenn am Eingangsparameter *REQ* der Wert 1 anliegt.

Die nachfolgende Tabelle enthält Informationen zum prinzipiellen Aufbau der Slave-Diagnose.

Weitergehende Informationen entnehmen Sie bitte den Handbüchern zu den von Ihnen verwendeten DP-Slaves.

Byte	Bedeutung
0	Stationsstatus 1
1	Stationsstatus 2
2	Stationsstatus 3
3	Master-Stationsnummer
4	Herstellerkennung (High-Byte)
5	Herstellerkennung (Low-Byte)
6 ...	Weitere slavespezifische Diagnose

Arbeitsweise

Beim SFC 13 handelt es sich um einen asynchron arbeitenden SFC, d.h. dass sich die Bearbeitung über mehrere SFC-Aufrufe erstreckt. Die Ausgangsparameter *RET_VAL* und *BUSY* zeigen den Zustand des Auftrags an, siehe nachfolgende Tabelle.

Zusammenhang zwischen Aufruf, *REQ*, *RET_VAL* und *BUSY*:

Lfd. Nr. des Aufrufs	Aufrufart	REQ	RET_VAL	BUSY
1	erster Aufruf	1	7001h bzw. Fehlercode	1 0
2 ... (n-1)	Zwischenaufruf	irrelevant	7002h	1
n	letzter Aufruf	irrelevant	Falls keine Fehler aufgetreten sind, wird die Anzahl gelieferter Daten in Bytes als positive Zahl eingetragen bzw. Fehlercode, falls Fehler aufgetreten sind.	0

Parameter

Parameter	Deklaration	Datentyp	Speicherbereich	Beschreibung
REQ	INPUT	BOOL	E, A, M, D, L, Konstante	REQ = 1: Anforderung zum Lesen
LADDR	INPUT	WORD	E, A, M, D, L, Konstante	Projektierte Diagnoseadresse des DP-Slaves
RET_VAL	OUTPUT	INT	E, A, M, D, L	Rückgabewert
RECORD	OUTPUT	ANY	E, A, M, D, L	Zielbereich für die gelesenen Diagnosedaten. Es ist nur der Datentyp BYTE zulässig. Die Mindestlänge des zu lesenden Datensatzes bzw. des Zielbereichs beträgt 6. Die Maximallänge des zu lesenden Datensatzes beträgt 240. Bei Normslaves, bei denen die Anzahl der Normdiagnosedaten größer als 240 Bytes ist und maximal 244Bytes beträgt, werden die ersten 240Bytes in den Zielbereich übertragen und das entsprechende Overflow-Bit in den Daten gesetzt.
BUSY	OUTPUT	BOOL	E, A, M, D, L	BUSY = 1: Der Lesevorgang ist noch nicht beendet.

RECORD

Von der CPU wird die tatsächliche Länge der gelesenen Diagnosedaten ausgewertet:

Wenn die Längenangabe von *RECORD*

- kleiner als die Anzahl der gelieferten Daten ist, werden die Daten verworfen, und in *RET_VAL* wird die zugehörige Fehlerinformation eingetragen.
- größer oder gleich der Anzahl der gelieferten Daten ist, werden die Daten in den Zielbereich übernommen, und in *RET_VAL* wird die tatsächliche Länge als positiver Wert eingetragen.

**Hinweis!**

Sie müssen darauf achten, dass die Aktualparameter von *RECORD* bei allen Aufrufen, die zu einem Auftrag gehören, übereinstimmen.

Ein Auftrag ist eindeutig festgelegt durch den Eingangsparameter *LADDR* und *RECORD*.

Normslaves Bei Normslaves, bei denen die Anzahl der Normdiagnosedaten zwischen 241 und 244Bytes liegt, ist folgendes beachten:
Falls die Längenangabe von *RECORD*

- kleiner als 240Bytes ist, werden die Daten verworfen und in *RET_VAL* wird die zugehörige Fehlerinformation eingetragen.
- größer oder gleich 240Bytes ist, werden die ersten Bytes der Normdiagnosedaten in den Zielbereich übertragen und das entsprechende Overflow-Bit in den Daten gesetzt.

**RET_VAL
(Rückgabewert)** Trat während der Bearbeitung der Funktion ein Fehler auf, enthält der Rückgabewert einen Fehlercode.
Falls kein Fehler auftrat, steht in *RET_VAL* die Länge der tatsächlich übertragenen Daten.

**Hinweis!**

Die Anzahl der gelesenen Daten ist bei einem DP-Slave von seinem Diagnosezustand abhängig.

Fehlerinformation Nähere Informationen zu allgemeinen Fehlerinformationen finden Sie am Anfang dieses Kapitels.
Die für den SFC 13 spezifischen Fehlerinformationen sind eine Teilmenge der Fehlerinformationen für den SFC 59 RD_REC.
Nähere Informationen finden Sie im SFC 59.

SFC 14 - DPRD_DAT - Konsistente Nutzdaten lesen

Beschreibung Mit dem SFC 14 DPRD_DAT (read consistent data of a DP-normslave) werden die konsistenten Daten eines DP-Normslaves ausgelesen. Dabei muss die Länge der konsistenten Daten drei oder mehr als vier Bytes betragen, wobei die Maximallänge 64Byte beträgt. Der Eingangsparameter *RECORD* legt den Zielbereich fest, in den die gelesenen Daten nach einer fehlerfreien Datenübertragung eingetragen werden. Dabei muss der Zielbereich dieselbe Länge aufweisen wie von Ihnen für das selektierte Modul projektiert worden ist.

Handelt es sich um einen DP-Normslave mit modularem Aufbau bzw. mit mehreren DP-Kennungen, so kann mit einem SFC 14-Aufruf jeweils nur auf die Daten eines Moduls / DP-Kennung unter der projektierten Anfangsadresse zugegriffen werden.

Der SFC 14 wird verwendet, da mit den Ladebefehlen, die auf die Peripherie bzw. auf das Prozessabbild der Eingänge zugreifen, maximal vier Bytes zusammenhängend ausgelesen werden können.

Definition *Konsistente Daten*
 Als konsistente Daten werden Daten bezeichnet, die inhaltlich zusammengehören und nicht getrennt werden dürfen. Z.B. ist es wichtig die Werte von Analogmodulen immer konsistent zu behandeln, denn der Wert eines Analogmoduls darf durch das Auslesen zu zwei verschiedenen Zeitpunkten nicht verfälscht werden.

Parameter

Parameter	Deklaration	Datentyp	Speicherbereich	Beschreibung
LADDR	INPUT	WORD	E, A, M, D, L, Konstante	Projektierte Anfangsadresse aus dem E-Bereich des Moduls, aus dem gelesen werden soll
RET_VAL	OUTPUT	INT	E, A, M, D, L	Tritt während der Bearbeitung der Funktion ein Fehler auf, enthält der Rückgabewert einen Fehlercode.
RECORD	OUTPUT	ANY	E, A, M, D, L	Zielbereich für die gelesenen Nutzdaten. Er muss genauso lang sein, wie für das selektierte Modul projektiert wurde. Es ist nur der Datentyp BYTE zulässig.

RET_VAL
(Rückgabewert)

Wert	Beschreibung
0000h	Es ist kein Fehler aufgetreten.
8090h	Für die angegebene logische Basisadresse haben sie kein Modul projiziert, oder Sie haben die Einschränkung über die Länge der konsistenten Daten nicht beachtet.
8092h	In ANY-Referenz ist eine Typangabe ungleich BYTE angegeben.
8093h	Für die unter <i>LADDR</i> angegebene logische Adresse existiert kein DP-Modul, von der Sie konsistente Daten lesen können.
80A0h	Anfangsadresse des Adressbereichs im Übergabespeicher fehlerhaft.
80B0h	Slaveausfall an externer DP-Anschaltung
80B1h	Die Länge des angegebenen Zielbereichs ist ungleich der projizierten Nutzdatenlänge.
80B2h	Systemfehler bei externer DP-Anschaltung
80B3h	Systemfehler bei externer DP-Anschaltung
80C0h	Systemfehler bei externer DP-Anschaltung
80C2h	Systemfehler bei externer DP-Anschaltung
80F _x h	Systemfehler bei externer DP-Anschaltung
87 _{xy} h	Systemfehler bei externer DP-Anschaltung
808 _x h	Systemfehler bei externer DP-Anschaltung

SFC 15 - DPWR_DAT - Konsistente Nutzdaten schreiben

Beschreibung

Mit dem SFC 15 DPWR_DAT (write consistent data to a DP-normslave) werden die Daten, die im Eingangsparameter *RECORD* stehen, konsistent zum adressierten DP-Normslave übertragen. Dabei muss die Länge der konsistenten Daten drei oder mehr als vier Bytes betragen, wobei die Maximallänge 64Byte beträgt. Die Datenübertragung erfolgt synchron, d.h. nach Beendigung des SFC ist der Schreibvorgang abgeschlossen. Es ist darauf zu achten, dass der Quellbereich dieselbe Länge aufweist, wie von Ihnen für das selektierte Modul projektiert worden ist.

Handelt es sich um einen DP-Normslave mit modularem Aufbau, kann nur auf ein Modul des DP-Slaves zugegriffen werden.

Der SFC 15 wird verwendet, da mit den Transferbefehlen, die auf die Peripherie bzw. auf das Prozessabbild der Ausgänge zugreifen, maximal vier Bytes zusammenhängend geschrieben werden können.

Definition

Konsistente Daten

Als konsistente Daten werden Daten bezeichnet, die inhaltlich zusammengehören und nicht getrennt werden dürfen. Beispielsweise ist es wichtig die Werte von Analogmodulen immer konsistent zu behandeln, denn der Wert eines Analogmoduls darf durch das Auslesen zu zwei verschiedenen Zeitpunkten nicht verfälscht werden.

Parameter

Parameter	Deklaration	Datentyp	Speicherbereich	Beschreibung
LADDR	INPUT	WORD	E, A, M, D, L, Konstante	Projektierte Anfangsadresse aus dem A-Bereich des Moduls, auf die geschrieben werden soll
RECORD	INPUT	ANY	E, A, M, D, L	Quellbereich für die zu schreibenden Nutzdaten. Er muss genauso lang sein, wie für das selektierte Modul projektiert wurde. Es ist nur der Datentyp BYTE zulässig.
RET_VAL	OUTPUT	INT	E, A, M, D, L	Tritt während der Bearbeitung der Funktion ein Fehler auf, enthält der Rückgabewert einen Fehlercode.

RET_VAL
(Rückgabewert)

Wert	Beschreibung
0000h	Es ist kein Fehler aufgetreten.
8090h	Für die angegebene logische Basisadresse haben sie kein Modul projektiert, oder Sie haben die Einschränkung über die Länge der konsistenten Daten nicht beachtet.
8092h	In ANY-Referenz ist eine Typangabe ungleich BYTE angegeben.
8093h	Für die unter <i>LADDR</i> angegebene logische Adresse existiert kein DP-Modul, auf die Sie konsistente Daten schreiben können.
80A1h	Das selektierte Modul ist fehlerhaft.
80B0h	Slaveausfall an externer DP-Anschaltung
80B1h	Die Länge des angegebenen Quellbereichs ist ungleich der projektierten Nutzdatenlänge.
80B2h	Systemfehler bei externer DP-Anschaltung
80B3h	Systemfehler bei externer DP-Anschaltung
80C1h	Die Daten des auf dem Modul vorangegangenen Schreibauftrags sind von dem Modul noch nicht bearbeitet.
80C2h	Systemfehler bei externer DP-Anschaltung
80Fxh	Systemfehler bei externer DP-Anschaltung
85xyh	Systemfehler bei externer DP-Anschaltung
808xh	Systemfehler bei externer DP-Anschaltung

SFC 17 - ALARM_SQ und SFC 18 - ALARM_S

Beschreibung Der SFC 17 ALARM_SQ (Erzeugung quittierbarer bausteinbezogener Meldungen) und der SFC 18 ALARM_S (Erzeugung stets quittierbarer bausteinbezogener Meldungen) generieren bei jedem Aufruf eine Meldung, an die Sie einen Begleitwert anhängen können. Die Meldung wird an alle dafür angemeldeten Teilnehmer gesendet. Der SFC 17 und SFC 18 stellen Ihnen also einen einfachen Meldemechanismus zur Verfügung. Bitte beachten Sie hierbei, dass Sie den SFC 17 und SFC 18 nur dann aufrufen, wenn der Wert des meldeauslösenden Signals *SIG* gegenüber dem letzten Aufruf invertiert ist. Ist dies nicht der Fall, so wird Ihnen dies über *RET_VAL* mitgeteilt, und es wird keine Meldung gesendet. Beim ersten Aufruf des SFC 17 und SFC 18 müssen Sie dafür Sorge tragen, dass am Eingang *SIG* "1" anliegt. Sonst bekommen Sie über *RET-VAL* eine Fehlerinformation, und es wird keine Meldung gesendet.



Hinweis!

Rufen Sie den SFC 17 und den SFC 18 aus einem FB heraus auf, dem Sie zuvor die entsprechenden Systemattribute zugewiesen haben!

Systemressourcen Bei der Meldungserzeugung mit dem SFC 17 und dem SFC 18 belegt das Betriebssystem für die Dauer eines Signalzyklus eine Systemressource. Der Signalzyklus dauert beim SFC 18 vom SFC-Aufruf mit *SIG* = "1" bis zum erneuten Aufruf mit *SIG* = "0". Beim SFC 17 kommt zu dieser Zeitspanne ggf. noch die Zeit bis zur Quittierung des kommenden Signals durch eines der angemeldeten Anzeigeräte hinzu. Falls innerhalb des Signalzyklus ein Überladen oder Löschen des meldungserzeugenden Bausteins erfolgt, bleibt die zugehörige Systemressource bis zum nächsten Neustart belegt.

Meldungs-Quittierung Sie können die vom SFC 17 gesendeten Meldungen mit Signalzustand "1" an einem angemeldeten Anzeigerät quittieren. Den Quittierzustand der letzten "Gekommen-Meldung" und den Signalzustand beim letzten SFC 17-Aufruf können Sie mit Hilfe des SFC 19 ALARM_SC ermitteln. Meldungen, die Sie mit dem SFC 18 gesendet haben, sind immer implizit quittiert. Den Signalzustand beim letzten SFC 18-Aufruf können Sie mit Hilfe des SFC 19 ALARM_SC ermitteln.

**Zwischen-
speicherung**

Die SFCs 17 und 18 belegen Systemressourcen. Darin werden unter anderem die letzten beiden Signalzustände inklusiv Zeitstempel und Begleitwert zwischengespeichert. Erfolgt ein Aufruf des SFC 17 oder SFC 18 zu einem Zeitpunkt, an dem die Signalzustände der beiden letzten "gültigen" SFC-Aufrufe noch nicht gesendet sind (Signaloverflow), werden der aktuelle und der letzte Signalzustand verworfen und eine Overflow-Kennung im Zwischenspeicher gesetzt. Zum nächstmöglichen Zeitpunkt wird das vorletzte Signal samt Overflow-Kennung gesendet.

Instanzoverflow

Falls die Anzahl von SFC 17- und SFC 18-Aufrufen größer ist als die maximale Anzahl von Systemressourcen Ihrer CPU, kann es zu einem Ressourcenengpass (Instanzoverflow) kommen. Dies wird Ihnen sowohl durch eine Fehlerinformation in *RET_VAL* als auch an den angemeldeten Anzeigegeräten mitgeteilt.

Die maximale Anzahl von SFC 17- oder SFC 18-Aufrufen ist CPU-abhängig.

Parameter

Parameter	Deklaration	Datentyp	Speicherbereich	Beschreibung
SIG	INPUT	BOOL	E, A, M, D, L	Das meldungsauslösende Signal
ID	INPUT	WORD	E, A, M, D, L	Datenkanal für Meldungen: EEEh
EV_ID	INPUT	DWORD	Konst. (E, A, M, D, L)	Meldungsnummer (nicht erlaubt: 0)
SD	INPUT	ANY	E, A, M, D, T, Z	Begleitwert
RET_VAL	OUTPUT	INT	E, A, M, D, L	Rückgabewert

SD

Begleitwert

Maximale Länge: 12Byte

Zulässig sind nur die Datentypen:

BOOL (nicht erlaubt Bitfeld), BYTE, CHAR, WORD, INT, DWORD, DINT, REAL, DATE, TOD, TIME, S5TIME, DATE_AND_TIME

RET_VAL (Rückgabewert) tritt während der Bearbeitung der Funktion ein Fehler auf, enthält der Rückgabewert einen Fehlercode.

Wert	Beschreibung
0000h	kein Fehler
0001h	<ul style="list-style-type: none"> • Der Begleitwert ist länger als die maximal zulässige Länge oder • Der Zugriff auf den Anwenderspeicher ist nicht möglich (z.B. Zugriff auf gelöschten DB). Die Meldung wird gesendet. • Der Begleitwert zeigt auf einen Wert im Lokaldatenbereich. Die Meldung wird gesendet.
0002h	Warnung: Der letzte freie Meldequittierspeicher wurde belegt.
8081h	Der angegebene <i>EV_ID</i> liegt außerhalb des zulässigen Bereiches.
8082h	Meldungsverlust, da Ihre CPU keine Ressourcen für die Erzeugung bausteinbezogener Meldungen durch SFCs mehr hat.
8083h	Meldungsverlust, da derselbe Signalwechsel bereits vorliegt, aber noch nicht gesendet werden konnte (Signaloverflow).
8084h	Beim aktuellen und beim vorangegangenen SFC 17- / SFC 18-Aufruf hat das meldungsauslösende Signal <i>SIG</i> denselben Wert.
8085h	Für die angegebene <i>EV_ID</i> liegt keine Anmeldung vor.
8086h	Ein SFC-Aufruf für die angegebene <i>EV_ID</i> ist bereits in einer Prioritätsklasse niedriger Priorität in Bearbeitung.
8087h	Beim ersten Aufruf des SFC 17, SFC 18 hatte das meldungsauslösende Signal den Wert "0".
8088h	Die angegebene <i>EV_ID</i> wird bereits von einer anderen Systemressource (zu SFC 17, SFC 18) belegt.
8xyy	Allgemeine Fehlerinformation

SFC 19 - ALARM_SC - Quittierzustand der letzten Meldung

Beschreibung

Mit Hilfe des SFC 19 ALARM_SC können Sie:

- Den Quittierzustand der letzten ALARM_SQ-Gekommen-Meldung und den Zustand des meldungsauslösenden Signals beim letzten Aufruf des SFC 17 ermitteln
- Den Zustand des meldungsauslösenden Signals beim letzten Aufruf des SFC 18 ermitteln

Die Meldung bzw. das Signal ist über die von Ihnen vorgegebene Meldungsnummer eindeutig referenziert, falls Sie die Meldungsnummern mit Hilfe der Meldungsprojektierung vergeben haben.

Der SFC 19 greift auf die temporär belegten Speicher des SFC 17 und SFC 18 zu.

Parameter

Parameter	Deklaration	Datentyp	Speicherbereich	Beschreibung
EV_ID	INPUT	DWORD	E, A, M, D, L Konst.	Meldungsnummer, zu der Sie den Signalzustand beim letzten SFC-Aufruf bzw. den Quittierzustand der letzten Gekommen-Meldung (nur bei SFC 17) ermitteln möchten
RET_VAL	OUTPUT	INT	E, A, M, D, L	Rückgabewert
STATE	OUTPUT	BOOL	E, A, M, D, L	Zustand des meldungsauslösenden Signals beim letzten SFC-Aufruf
Q_STATE	OUTPUT	BOOL	E, A, M, D, L	Falls der angegebene Parameter <i>EV_ID</i> zu einem SFC 18-Aufruf gehört: "1" Falls der angegebene Parameter <i>EV_ID</i> zu einem SFC 17-Aufruf gehört: Quittierzustand der letzten Gekommen-Meldung: "0": nicht quittiert "1": quittiert

RET_VAL

(Rückgabewert)

Tritt während der Bearbeitung der Funktion ein Fehler auf, enthält der Rückgabewert einen Fehlercode.

Wert	Beschreibung
0000h	kein Fehler
8081h	Der angegebene <i>EV_ID</i> liegt außerhalb des zulässigen Bereiches.
8082h	Zu dieser <i>EV_ID</i> ist momentan kein Speicherplatz belegt (Mögliche Ursache: Das zugehörige Signal hatte noch nie den Zustand "1", oder es hat bereits wieder den Zustand "0" angenommen).
8xyy	Allgemeine Fehlerinformation

SFC 20 - BLKMOV - Variable kopieren

Beschreibung Mit dem SFC 20 BLKMOV (block move) wird der Inhalt eines Speicherbereiches (Quellfeld) in einen anderen Speicherbereich (Zielfeld) kopiert.

Es können alle Speicherbereiche kopiert werden, außer:

- folgende Bausteine: FC, SFC, FB, SFB, OB, SDB
- Zähler
- Zeiten
- Speicherbereiche des Peripheriebereiches.

Es besteht die Möglichkeit, dass der Quellparameter auch in einem nichtablaufrelevanten Datenbaustein (DB, der mit dem Schlüsselwort UNLINKED kompiliert wurde) im Ladespeicher liegen kann.

Unterbrechbarkeit Die Schachtelungstiefe ist nicht begrenzt, solange das Quellfeld nicht Teil eines Datenbausteins ist, der nur im Ladespeicher vorhanden ist. Wird jedoch eine SFC 20-Bearbeitung unterbrochen, bei der aus einem nicht ablaufrelevanten DB kopiert wird, kann eine solche SFC 20-Bearbeitung nicht mehr eingeschachtelt werden.

Parameter

Parameter	Deklaration	Datentyp	Speicherbereich	Beschreibung
SRCBLK	INPUT	ANY	E, A, M, D, L	Angabe des Speicherbereichs, der kopiert werden soll (Quellfeld). Arrays vom Datentyp STRING sind nicht erlaubt.
RET_VAL	OUTPUT	INT	E, A, M, D, L	Tritt während der Bearbeitung der Funktion ein Fehler auf, enthält der Rückgabewert einen Fehlercode.
DSTBLK	OUTPUT	ANY	E, A, M, D, L	Angabe des Speicherbereichs, in den kopiert werden soll (Zielfeld). Arrays vom Datentyp STRING sind nicht erlaubt.

**Hinweis!**

Quell- und Zielfeld dürfen sich nicht überlappen. Wenn das angegebene Zielfeld größer als das Quellfeld ist, dann werden auch nur so viele Daten in das Zielfeld kopiert, wie im Quellfeld stehen. Wenn das angegebene Zielfeld jedoch kleiner als das Quellfeld ist, dann werden nur so viele Daten kopiert, wie das Zielfeld aufnehmen kann.

Wenn der ANY-Pointer (Quelle oder Ziel) vom Typ BOOL ist, so muss die angegebene Länge durch 8 teilbar sein, da ansonsten der SFC nicht ausgeführt wird.

Wenn der ANY-Pointer vom Typ STRING ist, so muss die angegebene Länge 1 sein.

**RET_VAL
(Rückgabewert)**

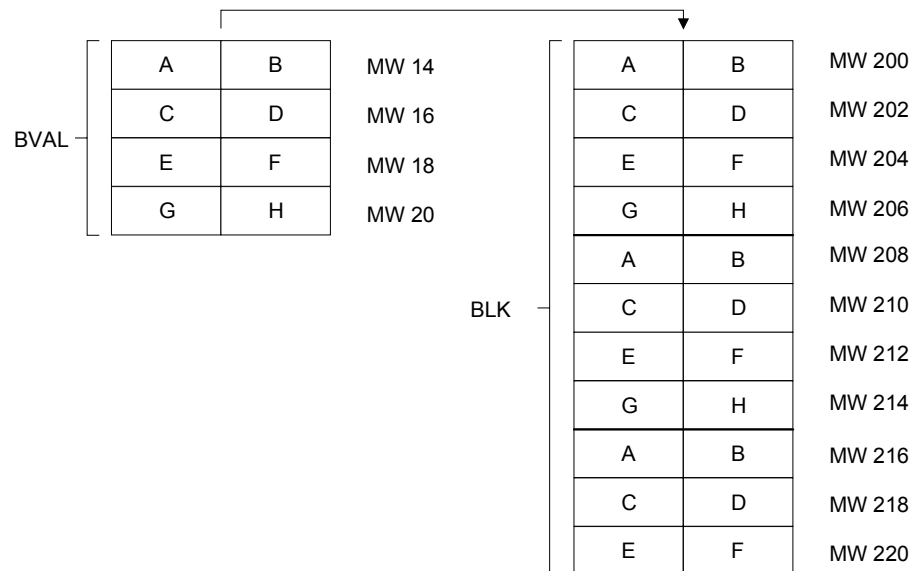
Tritt während der Bearbeitung der Funktion ein Fehler auf, enthält der Rückgabewert einen Fehlercode.

Wert	Beschreibung
0000h	kein Fehler
8091h	Die Schachtelungstiefe wurde überschritten.

SFC 21 - FILL - Feld vorbesetzen

Beschreibung

Mit dem SFC 21 FILL können Sie einen Speicherbereich (Zielfeld) mit dem Inhalt eines anderen Speicherbereiches (Quellfeld) vorbesetzen. In das angegebene Zielfeld kopiert der SFC 21 solange den Inhalt, bis der Speicherbereich komplett beschrieben ist.



Hinweis!

Quell- und Zielfeld dürfen sich nicht überlappen.

Wenn das vorzubelegende Zielfeld kein ganzzahliges Vielfaches der Länge des Eingangsparameters *BVAL* ist, wird das Zielfeld trotzdem bis zum letzten Byte beschrieben.

Wenn das vorzubelegende Zielfeld kleiner als das Quellfeld ist, dann werden nur so viele Daten kopiert, wie das Zielfeld aufnehmen kann.

Mit dem SFC 21 können Sie keine Werte schreiben in:

- folgende Bausteine: FB, SFB; FC, SFC, OB, SDB
- Zähler
- Zeiten
- Speicherbereiche des Peripheriebereiches

Parameter

Parameter	Deklaration	Datentyp	Speicherbereich	Beschreibung
BVAL	INPUT	ANY	E, A, M, D, L	Enthält den Wert bzw. die Beschreibung des Feldes, mit dessen Inhalt das Zielfeld vorbesetzt werden soll (Quellfeld). Arrays vom Datentyp STRING sind nicht erlaubt.
RET_VAL	OUTPUT	INT	E, A, M, D, L	Tritt während der Bearbeitung der Funktion ein Fehler auf, enthält der Rückgabewert einen Fehlercode.
BLK	OUTPUT	ANY	E, A, M, D, L	Enthält die Beschreibung des Feldes, das vorbesetzt werden soll (Zielfeld). Arrays vom Datentyp STRING sind nicht erlaubt.

Parameter ist eine Struktur

Wenn als Eingangsparameter eine Struktur übergeben wird, muss folgende Besonderheit berücksichtigt werden:

Die Länge einer Struktur wird immer auf eine gerade Anzahl von Bytes ausgerichtet. Wenn also eine Struktur mit einer ungeraden Anzahl Bytes deklariert wird, benötigt die Struktur ein Byte zusätzlichen Speicherplatz.

Beispiel:

Die Struktur wird folgendermaßen deklariert:

```
STRUKTUR_7_BYTE: STRUCT
BYTE_1_2 : WORD
BYTE_3_4 : WORD
BYTE_5_6 : WORD
BYTE_7: BYTE
END_STRUCT
```

Die deklarierte Struktur "STRUKTUR_7_BYTE" benötigt 8Bytes Speicherplatz.

**RET_VAL
(Rückgabewert)**

Tritt während der Bearbeitung der Funktion ein Fehler auf, enthält der Rückgabewert einen Fehlercode.

Der SFC 21 gibt keine spezifischen Fehlerinformationen aus.

SFC 22 - CREAT_DB - Datenbaustein erzeugen

Beschreibung Mit dem SFC 22 CREAT_DB (create data block) wird im Anwenderprogramm ein Datenbaustein erzeugt, der keine vorbesetzten Werte enthält. Es wird ein Datenbaustein mit einer Nummer aus einem angegebenen Bereich und mit vorgegebener Größe erzeugt, wobei aus dem angegebenen Bereich immer die kleinste mögliche Nummer an den DB vergeben wird. Um einen DB mit bestimmter Nummer zu erzeugen wird der Ober- und Untergrenze des vorzugebenden Bereichs die gleiche Nummer gegeben. Sind im Anwenderprogramm bereits DBs enthalten, können diese Nummern nicht mehr vergeben werden. Die Länge des DBs muss in einer geraden Zahl angegeben werden.

Unterbrechbarkeit Der SFC 22 kann durch höherpriorie OBs unterbrochen werden. Wird ein SFC 22 in einem höherpriorien OB aufgerufen, wird dieser Aufruf mit dem Fehlercode 8091h abgewiesen.

Parameter

Parameter	Deklaration	Datentyp	Speicherbereich	Beschreibung
LOW_LIMIT	INPUT	WORD	E, A, M, D, L, Konstante	Der untere Grenzwert ist die kleinste Nummer in dem Bereich der Nummern, die Sie Ihrem Datenbaustein zuordnen können.
UP_LIMIT	INPUT	WORD	E, A, M, D, L, Konstante	Der obere Grenzwert ist die größte Nummer in dem Bereich der Nummern, die Sie Ihrem Datenbaustein zuordnen können.
COUNT	INPUT	WORD	E, A, M, D, L, Konstante	Der Zählwert gibt die Anzahl der Datenbytes an, die Sie für ihren Datenbaustein reservieren möchten. Sie müssen hier eine gerade Zahl an Bytes (maximal 65534) angeben.
RET_VAL	OUTPUT	INT	E, A, M, D, L	Tritt während der Bearbeitung der Funktion ein Fehler auf, enthält der Rückgabewert einen Fehlercode.
DB_NUMBER	OUTPUT	WORD	E, A, M, D, L	Die Datenbausteinnummer ist die Nummer des erstellten Datenbausteins. Im Fehlerfall (Bit 15 von <i>RET_VAL</i> wurde gesetzt) wird in <i>DB_NUMBER</i> der Wert 0 eingetragen.

RET_VAL (Rückgabewert) tritt während der Bearbeitung der Funktion ein Fehler auf, enthält der Rückgabewert einen Fehlercode.

Wert	Beschreibung
0000h	kein Fehler
8091h	Sie haben den SFC 22 geschachtelt aufgerufen.
8092h	Die Funktion "Erzeugen eines DB" ist momentan nicht durchführbar, weil <ul style="list-style-type: none"> • die Funktion "Komprimieren des Anwenderspeichers" gerade aktiv ist.
80A1h	Fehler in der Nummer des DBs: <ul style="list-style-type: none"> • die Nummer ist 0 • die Nummer überschreitet die CPU-spezifische DB-Anzahl • Untergrenze > Obergrenze
80A2h	Fehler in der Länge des DBs: <ul style="list-style-type: none"> • die Länge ist 0 • die Länge wurde als ungerade Zahl angegeben • die Länge ist größer als die CPU zulässt
80B1h	Es ist keine DB-Nummer frei.
80B2h	Es steht nicht genügend freier Speicherplatz zur Verfügung.
80B3h	Es steht nicht genügend zusammenhängender Speicher zur Verfügung (Komprimieren durchführen!).

SFC 23 - DEL_DB - Datenbaustein löschen

Beschreibung Mit dem SFC 23 DEL_DB (delete data block) wird ein Datenbaustein im Arbeitsspeicher und gegebenenfalls im Ladespeicher der CPU gelöscht. Der angegebene DB darf weder in der aktuellen Ablaufebeine noch in einer Ablaufebeine mit niederer Priorität aufgeschlagen sein, also weder in einem der beiden DB-Register noch im B-Stack eingetragen sein, ansonsten wechselt die CPU beim Aufruf des SFC 23 in den Betriebszustand STOP. In der folgenden Tabelle ist aufgeführt, wann ein DB mit dem SFC 23 gelöscht werden kann.

Wenn der DB ...	dann ist er mit dem SFC 23 ...
durch Aufruf des SFC 22 "CREAT_DB" erzeugt wurde,	löschar.
nicht mit dem Schlüsselwort UNLINKED erzeugt wurde,	löschar.

Unterbrechbarkeit Der SFC 23 kann von höher prioren Ablaufebenen unterbrochen werden. Wird dort der SFC erneut aufgerufen, dann wird dieser zweite Aufruf abgebrochen, und in RET_VAL wird der Fehlercode 8091h eingetragen.

Parameter

Parameter	Deklaration	Datentyp	Speicherbereich	Beschreibung
DB_NUMBER	INPUT	WORD	E, A, M, D, L, Konstante	Nummer des zu löschenden DB.
RET_VAL	OUTPUT	INT	E, A, M, D, L	Tritt während der Bearbeitung der Funktion ein Fehler auf, enthält der Rückgabewert einen Fehlercode.

RET_VAL (Rückgabewert) Tritt während der Bearbeitung der Funktion ein Fehler auf, enthält der Rückgabewert einen Fehlercode.

Wert	Beschreibung
0000h	kein Fehler
8091h	Bei ineinandergeschachtelten SFC 23-Aufrufen wurde die maximale Schachtelungstiefe überschritten.
8092h	Die Funktion "Löschen eines DB" ist momentan nicht durchführbar, weil die Funktion: <ul style="list-style-type: none"> • "Komprimieren des Anwenderspeichers" oder • "Anwenderprogramm sichern" gerade aktiv ist.
80A1h	Fehler in der Nummer des DBs: <ul style="list-style-type: none"> • die Nummer ist 0 • die Nummer ist größer als die maximal mögliche DB-Anzahl
80B1h	Der DB mit der angegebenen Nummer ist auf der CPU nicht vorhanden.
80B2h	Der DB mit der angegebenen Nummer wurde mit dem Schlüsselwort UNLINKED erzeugt.
80B3h	Der DB befindet sich auf der Flashcard.

SFC 24 - TEST_DB - Datenbaustein testen

Beschreibung Mit dem SFC 24 TEST_DB (test data block) können Informationen über einen im Arbeitsspeicher der CPU liegenden Datenbaustein abgerufen werden. Der SFC ermittelt die Anzahl der Datenbytes und stellt fest, ob der ausgewählte DB schreibgeschützt ist.

Parameter

Parameter	Deklaration	Datentyp	Speicherbereich	Beschreibung
DB_NUMBER	INPUT	WORD	E, A, M, D, L, Konstante	Nummer des zu überprüfenden DBs
RET_VAL	OUTPUT	INT	E, A, M, D, L	Tritt während der Bearbeitung der Funktion ein Fehler auf, enthält der Rückgabewert einen Fehlercode.
DB_LENGTH	OUTPUT	WORD	E, A, M, D, L	Anzahl der Datenbytes, die der ausgewählte DB enthält.
WRITE_PROT	OUTPUT	BOOL	E, A, M, D, L	Information über die Schreibschutzkennung des ausgewählten DB (1 bedeutet schreibgeschützt).

RET_VAL (Rückgabewert) Tritt während der Bearbeitung der Funktion ein Fehler auf, enthält der Rückgabewert einen Fehlercode.

Wert	Beschreibung
0000h	kein Fehler
80A1h	Fehler beim Eingangsparameter <i>DB_NUMBER</i> : <ul style="list-style-type: none"> die Nummer ist 0 die Nummer ist größer als die maximal mögliche DB-Anzahl
80B1h	Der DB mit der angegebenen Nummer ist auf der CPU nicht vorhanden.
80B2h	Der DB mit der angegebenen Nummer wurde mit dem Schlüsselwort UNLINKED erzeugt.

SFC 28 ... SFC 31 - Uhrzeitalarm

Voraussetzungen Ein Uhrzeitalarm ist die Ursache für den uhrzeitgesteuerten Aufruf des Uhrzeitalarm-OB 10. Hierzu müssen folgende Voraussetzungen erfüllt sein:

- Der Uhrzeitalarm-OB ist parametrierbar über Hardware-Konfiguration oder mittels des SFC 28 (SET_TINT) im Anwenderprogramm.
- Der Uhrzeitalarm-OB ist aktiviert über Hardware-Konfiguration oder mittels des SFC 30 (ACT_TINT) im Anwenderprogramm.
- Der Uhrzeitalarm-OB darf nicht abgewählt worden sein.
- Der Uhrzeitalarm-OB muss in der CPU geladen sein.
- Falls das Stellen über den SFC 30 erfolgt, dürfen Startdatum und Startuhrzeit bei einmaliger Ausführung nicht abgelaufen sein, bei periodischer Ausführung wird der Uhrzeitalarm-OB zur nächsten abgelaufenen Periode aufgerufen (Startzeitpunkt + Vielfaches der Periodendauer).

SFCs 28 ... 31 Folgende SFCs kommen bei Uhrzeitalarmen zum Einsatz:

- Stellen: SFC 28
- Stornieren: SFC 29
- Aktivieren: SFC 30
- Abfragen: SFC 31

SFC 28 - SET_TINT

Mit dem SFC 28 SET_TINT (set time-of-day interrupt) können Startdatum und -uhrzeit der Uhrzeitalarm-Organisationsbausteine eingestellt werden. Bei der Startuhrzeit werden die Angaben für Sekunden- und Millisekunden ignoriert und auf 0 gesetzt.

Parameter

Parameter	Deklaration	Datentyp	Speicherbereich	Beschreibung
OB_NR	INPUT	INT	E, A, M, D, L, Konstante	Nummer des OBs, der zum Zeitpunkt <i>SDT</i> + Vielfaches von <i>PERIOD</i> gestartet wird (OB 10, OB 11).
SDT	INPUT	DT	D, L	Startdatum und Startuhrzeit
PERIOD	INPUT	WORD	E, A, M, D, L, Konstante	Periode vom Ausgangspunkt <i>SDT</i> an: 0000h = Einmal 0201h = Minütlich 0401h = Stündlich 1001h = Täglich 1201h = Wöchentlich 1401h = Monatlich 1801h = Jährlich 2001h = Monatsende
RET_VAL	OUTPUT	INT	E, A, M, D, L	Tritt während der Bearbeitung der Funktion ein Fehler auf, enthält der Rückgabewert einen Fehlercode.

RET_VAL (Rückgabewert) Tritt während der Bearbeitung der Funktion ein Fehler auf, enthält der Rückgabewert einen Fehlercode.

Wert	Beschreibung
0000h	kein Fehler
8090h	Fehlerhafter Parameter <i>OB_NR</i>
8091h	Fehlerhafter Parameter <i>SDT</i>
8092h	Fehlerhafter Parameter <i>PERIOD</i>
80A1h	Der eingestellte Startzeitpunkt liegt in der Vergangenheit.

SFC 29 - CAN_TINT - Uhrzeitalarm stornieren Mit dem SFC 29 CAN_TINT (cancel time-of-day interrupt) können Startdatum und -uhrzeit eines angegebenen Uhrzeitalarm - Organisationsbausteins gelöscht werden.

Parameter

Parameter	Deklaration	Datentyp	Speicherbereich	Beschreibung
OB_NR	INPUT	INT	E, A, M, D, L, Konstante	Nummer des OBs, dessen Startdatum und -uhrzeit gelöscht werden soll (OB 10, OB 11).
RET_VAL	OUTPUT	INT	E, A, M, D, L	Tritt während der Bearbeitung der Funktion ein Fehler auf, enthält der Rückgabewert einen Fehlercode.

RET_VAL (Rückgabewert)

Wert	Beschreibung
0000h	Es ist kein Fehler aufgetreten.
8090h	Fehlerhafter Parameter <i>OB_NR</i>
80A0h	Kein Startdatum/-uhrzeit festgelegt für den betreffenden Uhrzeitalarm-OB

**SFC 30 -
ACT_TINT -
Uhrzeitalarm
aktivieren**

Mit dem SFC 30 ACT_TINT (activate time-of-day interrupt) kann ein angegebener Uhrzeitalarm-Organisationsbaustein aktiviert werden.

Parameter

Parameter	Deklaration	Datentyp	Speicherbereich	Beschreibung
OB_NR	INPUT	INT	E, A, M, D, L, Konstante	Nummer des OBs, der aktiviert werden soll (OB 10, OB 11)
RET_VAL	OUTPUT	INT	E, A, M, D, L	Tritt während der Bearbeitung der Funktion ein Fehler auf, enthält der Rückgabewert einen Fehlercode.

**RET_VAL
(Rückgabewert)**

Wert	Beschreibung
0000h	Es ist kein Fehler aufgetreten.
8090h	Fehlerhafter Parameter <i>OB_NR</i>
80A0h	Kein Startdatum/-uhrzeit festgelegt für den betreffenden Uhrzeitalarm-OB
80A1h	Die aktivierte Zeit liegt in der Vergangenheit; Fehler tritt nur bei einmaliger Ausführung auf.

**SFC 31 -
QRY_TINT -
Uhrzeitalarm
abfragen**

Mit dem SFC 31 QRY_TINT (query time-of-day interrupt) kann der Status des angegebenen Uhrzeitalarm-Organisationsbausteins am Ausgangsparameter *STATUS* angezeigt werden.

Parameter

Parameter	Deklaration	Datentyp	Speicherbereich	Beschreibung
OB_NR	INPUT	INT	E, A, M, D, L, Konstante	Nummer des OBs, dessen Status abgefragt wird (OB 10, OB 11)
RET_VAL	OUTPUT	INT	E, A, M, D, L	Tritt während der Bearbeitung der Funktion ein Fehler auf, enthält der Rückgabewert einen Fehlercode.
STATUS	OUTPUT	WORD	E, A, M, D, L	Status des Uhrzeitalarms

**RET_VAL
(Rückgabewert)**

Wert	Beschreibung
0000h	Es ist kein Fehler aufgetreten.
8090h	Fehlerhafter Parameter <i>OB_NR</i>

STATUS

Bit	Wert	Beschreibung
0	0	Uhrzeitalarm ist vom Betriebssystem freigegeben.
1	0	Neue Uhrzeitalarme werden nicht verworfen.
2	0	Uhrzeitalarm ist nicht aktiviert oder abgelaufen.
3	-	reserviert
4	0	Uhrzeitalarm-OB ist nicht geladen.
5	0	Die Ausführung des Uhrzeitalarm-OBs ist durch eine laufende Testfunktion gesperrt.

SFC 32 - SRT_DINT - Verzögerungsalarm starten

Beschreibung Mit dem SFC 32 SRT_DINT (start time-delay interrupt) kann ein Verzögerungsalarm gestartet werden, der nach Ablauf der parametrisierten Verzögerungszeit (Parameter *DTIME*) einen Verzögerungsalarm-OB aufruft. Am Parameter *SIGN* kann ein anwenderspezifisches Kennzeichen angegeben werden, um den Start des Verzögerungsalarms zu kennzeichnen. Während der Ausführung erscheinen die Werte von *DTIME* und *SIGN* in der Startereignisinformation des angegebenen OB.

Voraussetzungen Damit ein Verzögerungsalarm aufgerufen werden kann, müssen folgende Voraussetzungen erfüllt werden:

- Der Verzögerungsalarm-OB muss gestartet sein (mit dem SFC 32).
- Der Verzögerungsalarm-OB darf nicht abgewählt worden sein.
- Der Verzögerungsalarm-OB muss in der CPU vorhanden sein.

Parameter

Parameter	Deklaration	Datentyp	Speicherbereich	Beschreibung
OB_NR	INPUT	INT	E, A, M, D, L, Konstante	Nummer des OB, der nach einer Verzögerungszeit gestartet wird (OB 20, OB 21)
DTIME	INPUT	TIME	E, A, M, D, L, Konstante	Zeitwert der Verzögerung (1 ... 60 000 ms)
SIGN	INPUT	WORD	E, A, M, D, L, Konstante	Kennzeichen, das beim Aufruf des Verzögerungsalarm-OBs in der Startereignisinformation des OBs erscheint
RET_VAL	OUTPUT	INT	E, A, M, D, L	Tritt während der Bearbeitung der Funktion ein Fehler auf, enthält der Rückgabewert einen Fehlercode.

Genauigkeit Die Zeit zwischen dem Aufruf des SFC 32 und dem Start des Verzögerungsalarm-OBs ist maximal eine Millisekunde kleiner als die parametrisierte Zeit, sofern keine Unterbrechungsereignisse den Aufruf verzögern.

RET_VAL (Rückgabewert)

Wert	Beschreibung
0000h	Es ist kein Fehler aufgetreten.
8090h	Fehlerhafter Parameter <i>OB_NR</i>
8091h	Fehlerhafter Parameter <i>DTIME</i>

SFC 33 - CAN_DINT - Verzögerungsalarm stornieren

Beschreibung Mit dem SFC 33 CAN_DINT (cancel time-delay interrupt) kann ein bereits gestarteter Verzögerungsalarm storniert werden. In diesem Fall wird der Verzögerungsalarm-OB nicht aufgerufen.

Voraussetzungen Damit ein Verzögerungsalarm aufgerufen werden kann, müssen folgende Voraussetzungen erfüllt werden:

- Der Verzögerungsalarm-OB muss gestartet sein (mit dem SFC 32).
- Der Verzögerungsalarm-OB darf nicht abgewählt worden sein.
- Der Verzögerungsalarm-OB muss in der CPU vorhanden sein.

Parameter

Parameter	Deklaration	Datentyp	Speicherbereich	Beschreibung
OB_NR	INPUT	INT	E, A, M, D, L, Konstante	Nummer des OBs, der storniert werden soll (OB 20, OB 21).
RET_VAL	OUTPUT	INT	E, A, M, D, L	Tritt während der Bearbeitung der Funktion ein Fehler auf, enthält der Rückgabewert einen Fehlercode.

RET_VAL (Rückgabewert)

Wert	Beschreibung
0000h	Es ist kein Fehler aufgetreten.
8090h	Fehlerhafter Parameter <i>OB_NR</i>
80A0h	Verzögerungsalarm ist nicht gestartet.

SFC 34 - QRY_DINT - Verzögerungsalarm Status abfragen

Beschreibung Mit dem SFC 34 QRY_DINT (query time-delay interrupt) kann der Zustand eines Verzögerungsalarms am Ausgangsparameter *STATUS* angezeigt werden.

Voraussetzungen Damit ein Verzögerungsalarm aufgerufen werden kann, müssen folgende Voraussetzungen erfüllt werden:

- Der Verzögerungsalarm-OB muss gestartet sein (mit dem SFC 32).
- Der Verzögerungsalarm-OB darf nicht abgewählt worden sein.
- Der Verzögerungsalarm-OB muss in der CPU vorhanden sein.

Parameter

Parameter	Deklaration	Datentyp	Speicherbereich	Beschreibung
OB_NR	INPUT	INT	E, A, M, D, L, Konstante	Nummer des OB, dessen Zustand abgefragt wird (OB 20, OB 21)
RET_VAL	OUTPUT	INT	E, A, M, D, L	Tritt während der Bearbeitung der Funktion ein Fehler auf, enthält der Rückgabewert einen Fehlercode.
STATUS	OUTPUT	WORD	E, A, M, D, L	Zustand des Verzögerungsalarms

RET_VAL (Rückgabewert)

Wert	Beschreibung
0000h	Es ist kein Fehler aufgetreten.
8090h	Fehlerhafter Parameter <i>OB_NR</i>

STATUS

Bit	Wert	Beschreibung
0	0	Verzögerungsalarm ist vom Betriebssystem freigegeben.
1	0	Neue Verzögerungsalarm werden nicht verworfen.
2	0	Verzögerungsalarm ist nicht aktiviert oder abgelaufen.
3	-	-
4	0	Verzögerungsalarm -OB ist nicht geladen.
5	0	Die Ausführung des Verzögerungsalarm-OBs ist durch eine laufende Testfunktion gesperrt.

SFC 36 - MSK_FLT - Synchronfehlerereignisse maskieren

Beschreibung Mit dem SFC 36 MSK_FLT (mask synchronous faults) können Sie die Reaktion der CPU auf Synchronfehlerereignisse steuern, indem die jeweiligen Synchronfehlerereignisse maskiert werden. Mit dem Aufruf des SFC 36 werden die Synchronfehlerereignisse in der aktuellen Prioritätsklasse maskiert. Wenn in den Eingangsparametern einzelne Bits der Synchronfehlermasken auf "1" gesetzt werden, dann behalten andere, vorher bereits gesetzte Bits ihre Wertigkeit "1". Dadurch erhält man neue Fehlermasken, die über die Ausgangsparameter ausgelesen werden können. Die maskierten Synchronfehlerereignisse werden in einem Ereignisstatusregister eingetragen und rufen keinen OB auf. Das Auslesen des Ereignisstatusregisters erfolgt mit dem SFC 38 READ_ERR.

Parameter

Parameter	Deklaration	Datentyp	Speicherbereich	Beschreibung
PRGFLT_SET_MASK	INPUT	DWORD	E, A, M, D, L, Konstante	Zu maskierende Programmierfehler
ACCFLT_SET_MASK	INPUT	DWORD	E, A, M, D, L, Konstante	Zu maskierende Zugriffsfehler
RET_VAL	OUTPUT	INT	E, A, M, D, L	Tritt während der Bearbeitung der Funktion ein Fehler auf, enthält der Rückgabewert einen Fehlercode.
PRGFLT_MASKED	OUTPUT	DWORD	E, A, M, D, L	Maskierte Programmierfehler
ACCFLT_MASKED	OUTPUT	DWORD	E, A, M, D, L	Maskierte Zugriffsfehler

RET_VAL (Rückgabewert)

Wert	Beschreibung
0000h	Keiner der Fehler war bereits maskiert.
0001h	Mindestens einer der Fehler war bereits maskiert, trotzdem werden die anderen Fehler maskiert.

SFC 37 - DMSK_FLT - Synchronfehlerereignisse demaskieren

Beschreibung Mit dem SFC 37 DMSK_FLT (unmask synchronous faults) werden bereits maskierte Synchronfehlerereignisse demaskiert. Mit dem Aufruf des SFC 37 werden die Synchronfehlerereignisse in der aktuellen Prioritätsklasse demaskiert. Hierzu werden in den Eingangsparametern die entsprechenden Bits der Fehlermasken auf "1" gesetzt. Dadurch erhält man neue Fehlermasken, die über die Ausgangsparameter ausgelesen werden können. Die abgefragten Einträge im Ereignisstatusregister werden gelöscht.

Parameter

Parameter	Deklaration	Datentyp	Speicherbereich	Beschreibung
PRGFLT_RESET_MASK	INPUT	DWORD	E, A, M, D, L, Konstante	Zu demaskierende Programmierfehler
ACCFLT_RESET_MASK	INPUT	DWORD	E, A, M, D, L, Konstante	Zu demaskierende Zugriffsfehler
RET_VAL	OUTPUT	INT	E, A, M, D, L	Tritt während der Bearbeitung der Funktion ein Fehler auf, enthält der Rückgabewert einen Fehlercode.
PRGFLT_MASKED	OUTPUT	DWORD	E, A, M, D, L	Noch maskierte Programmierfehler
ACCFLT_MASKED	OUTPUT	DWORD	E, A, M, D, L	Noch maskierte Zugriffsfehler

RET_VAL (Rückgabewert)

Wert	Beschreibung
0000h	Alle angegebenen Fehler wurden demaskiert.
0001h	Mindestens einer der Fehler war nicht maskiert, trotzdem werden die anderen Fehler demaskiert.

SFC 38 - READ_ERR - Ereignisstatusregister lesen

Beschreibung Mit dem SFC 38 READ_ERR (read error registers) wird das Ereignisstatusregister ausgelesen. Im Aufbau entspricht das Ereignisstatusregister den Programmier- und Zugriffsfehlermasken, die mit dem SFC 36 und 37 als Eingangsparameter programmiert wurden. Mit dem Aufruf des SFC 38 werden die gewünschten Einträge aus dem Ereignisstatusregister gelesen und gleichzeitig gelöscht. Über die Eingangsparameter wird festgelegt, welche Synchronfehlerereignisse aus dem Ereignisstatusregister abgefragt werden sollen. Sie erhalten die Information, welche der maskierten Synchronfehler in der aktuellen Prioritätsklasse mindestens einmal aufgetreten sind. Ein gesetztes Bit bedeutet, dass der zugehörige maskierte Synchronfehler mindestens einmal aufgetreten ist.

Parameter

Parameter	Deklaration	Datentyp	Speicherbereich	Beschreibung
PRGFLT_QUERY	INPUT	DWORD	E, A, M, D, L, Konstante	Abfrage der Programmierfehler
ACCFLT_QUERY	INPUT	DWORD	E, A, M, D, L, Konstante	Abfrage der Zugriffsfehler
RET_VAL	OUTPUT	INT	E, A, M, D, L	Tritt während der Bearbeitung der Funktion ein Fehler auf, enthält der Rückgabewert einen Fehlercode.
PRGFLT_ESR	OUTPUT	DWORD	E, A, M, D, L	Aufgetretene Programmierfehler
ACCFLT_ESR	OUTPUT	DWORD	E, A, M, D, L	Aufgetretene Zugriffsfehler

RET_VAL (Rückgabewert)

Wert	Beschreibung
0000h	Alle abgefragten Fehler sind maskiert.
0001h	Mindestens einer der abgefragten Fehler ist nicht maskiert.

SFC 39 - DIS_IRT - Alarmereignisse sperren

Beschreibung

Mit dem SFC 39 DIS_IRT (disable interrupt) sperren Sie die Bearbeitung neuer Alarm- und Asynchronfehlerereignisse. Sperren heißt, dass das Betriebssystem der CPU bei einem Unterbrechungsereignis

- weder einen Alarm-OB bzw. einen Asynchronfehler-OB aufruft,
- noch die festgelegte Reaktion bei nicht programmiertem Alarm-OB bzw. Asynchronfehler-OB auslöst.

Die Sperre der Bearbeitung von Alarm- und Asynchronfehlerereignissen bleibt über alle Prioritätsklassen gültig. Die Sperre kann nur mit dem SFC 40 aufgehoben werden bzw. auch bei Neustart.

Ob das Betriebssystem das aufgetretene Alarm- bzw. Asynchronfehlerereignis in den Diagnosepuffer einträgt, ist abhängig von Ihrer Wahl des Eingangsparameters *MODE*.



Hinweis!

Bitte beachten Sie, dass bei der Programmierung des SFC 39 alle eintreffenden Alarme verworfen werden.

Parameter

Parameter	Deklaration	Datentyp	Speicherbereich	Beschreibung
MODE	INPUT	BYTE	E, A, M, D, L, Konstante	Gibt an, welche Alarm- und Asynchronfehlerereignisse gesperrt werden.
OB_NR	INPUT	INT	E, A, M, D, L, Konstante	OB-Nummer
RET_VAL	OUTPUT	INT	E, A, M, D, L	Tritt während der Bearbeitung der Funktion ein Fehler auf, enthält der Rückgabewert einen Fehlercode.

MODE

MODE	Beschreibung
00	Alle neu aufgetretenen Alarm- und Asynchronfehlerereignisse werden gesperrt (nicht gesperrt werden Synchronfehlerereignisse).
01	Alle neu auftretenden Ereignisse einer angegebenen Alarmklasse werden gesperrt. Die Alarmklasse kennzeichnen Sie wie folgt: <ul style="list-style-type: none"> • Uhrzeitalarme: 10 • Verzögerungsalarml: 20 • Weckalarml: 30 • Prozessalarml: 40 • Alarml für DP-V1: 50 • Asynchrone Fehleralarml: 80 Die Einträge in den Diagnosepuffer erfolgen weiterhin.
02	Alle neu auftretenden Ereignisse eines angegebenen Alarms werden gesperrt. Den Alarm kennzeichnen Sie durch die OB-Nummer. Die Einträge in den Diagnosepuffer erfolgen weiterhin.
80	Alle neu auftretenden Alarm- und Asynchronfehlerereignisse werden gesperrt und sie werden auch nicht mehr in den Diagnosepuffer eingetragen. Das Betriebssystem trägt das Ereignis 5380h in den Diagnosepuffer ein.
81	Alle neu auftretenden Ereignisse einer angegebenen Alarmklasse werden gesperrt und sie werden auch nicht mehr in den Diagnosepuffer eingetragen. Das Betriebssystem trägt das Ereignis 5380h in den Diagnosepuffer ein.
82	Alle neu auftretenden Ereignisse einer angegebenen Alarms werden gesperrt und sie werden auch nicht mehr in den Diagnosepuffer eingetragen. Das Betriebssystem trägt das Ereignis 5380h in den Diagnosepuffer ein.

RET_VAL**(Rückgabewert)**

Wert	Beschreibung
0000h	Es ist kein Fehler aufgetreten.
8090h	Der Eingangsparameter <i>OB_NR</i> enthält einen unzulässigen Wert.
8091h	Der Eingangsparameter <i>MODE</i> enthält einen unzulässigen Wert.
8xyyh	Allgemeine Fehlerinformation, siehe Fehlerauswertung mit dem Ausgangsparameter <i>RET_VAL</i> .

SFC 40 - EN_IRT - Gesperrte Alarmereignisse freigeben

Beschreibung Mit dem SFC 40 EN_IRT (enable interrupt) geben Sie die mit dem SFC 39 gesperrte Bearbeitung neuer Alarm- und Asynchronfehlerereignisse wieder frei. Freigeben heißt, dass das Betriebssystem der CPU bei einem Unterbrechungsereignis:

- einen Alarm-OB bzw. einen Asynchronfehler-OB aufruft, oder
- die festgelegte Reaktion bei nicht programmiertem Alarm-OB bzw. Asynchronfehler-OB auslöst.

Parameter

Parameter	Deklaration	Datentyp	Speicherbereich	Beschreibung
MODE	INPUT	BYTE	E, A, M, D, L, Konstante	Gibt an, welche Alarm- und Asynchronfehlerereignisse freigegeben werden.
OB_NR	INPUT	INT	E, A, M, D, L, Konstante	OB-Nummer
RET_VAL	OUTPUT	INT	E, A, M, D, L	Tritt während der Bearbeitung der Funktion ein Fehler auf, enthält der Rückgabewert einen Fehlercode.

MODE

MODE	Beschreibung
00	Alle neu aufgetretenen Alarm- und Asynchronfehlerereignisse werden freigegeben.
01	Alle neu auftretenden Ereignisse einer angegebenen Alarmklasse werden freigegeben. Die Alarmklasse kennzeichnen Sie wie folgt: <ul style="list-style-type: none"> • Uhrzeitalarme: 10 • Verzögerungsalarml: 20 • Weckalarml: 30 • Prozessalarml: 40 • Alarml für DP-V1: 50 • Asynchrone Fehleralarml: 80
02	Alle neu auftretenden Ereignisse eines angegebenen Alarms werden freigegeben. Den Alarm kennzeichnen Sie durch die OB-Nummer.

RET_VAL (Rückgabewert)

Wert	Beschreibung
0000h	Es ist kein Fehler aufgetreten.
8090h	Der Eingangsparameter <i>OB_NR</i> enthält einen unzulässigen Wert.
8091h	Der Eingangsparameter <i>MODE</i> enthält einen unzulässigen Wert.
8xyyh	Allgemeine Fehlerinformation, Siehe Fehlerauswertung mit dem Ausgangsparameter <i>RET_VAL</i> .

SFC 41 - DIS_AIRT - Alarmereignisse verzögern

Beschreibung Mit dem SFC 41 DIS_AIRT (disable alarm interrupts) wird die Bearbeitung von Alarm-OBs und Asynchronfehler-OBs, deren Priorität höher ist als die des aktuellen OBs verzögert. Innerhalb eines OBs kann der SFC 41 mehrmals aufgerufen werden, wobei die Aufrufe vom Betriebssystem gezählt werden. Die Bearbeitungsverzögerung ist solange gültig, bis mit dem SFC 42 EN_AIRT jede mit einem SFC 41 verzögerte Bearbeitung von Alarm-OBs und Asynchronfehler-OBs aufgehoben wird oder der aktuelle OB abgearbeitet ist.

Die Bearbeitung der anstehenden Alarm- oder Asynchronfehlerereignisse findet statt, sobald die Bearbeitungsverzögerung mit dem SFC 42 EN_AIRT aufgehoben wurde oder die aktuelle OB-Bearbeitung beendet ist.

Parameter

Parameter	Deklaration	Datentyp	Speicherbereich	Beschreibung
RET_VAL	OUTPUT	INT	E, A, M, D, L	Anzahl der Verzögerungen (= Anzahl der Aufrufe des SFC 41)

RET_VAL (Rückgabewert) Der Rückgabewert *RET_VAL* zeigt nach Ablauf des SFC die Anzahl der Bearbeitungsverzögerungen, also der Aufrufe des SFC 41 an (Die Alarmbearbeitung ist erst wieder freigegeben, wenn *RET_VAL* = 0 ist).

SFC 42 - EN_AIRT - Verzögerte Alarmereignissen freigeben

Beschreibung Mit dem SFC 42 EN_AIRT (enable alarm interrupts) wird die mit dem SFC 41 verzögerte Bearbeitung von höherprioren Alarm- bzw. Asynchronfehlerereignissen wieder freigegeben. Dabei muss jede einzelne Bearbeitungsverzögerung mit dem SFC 42 beendet werden. Wenn also z.B. mit 5 SFC 41-Aufrufen 5 verschiedene Alarme verzögert wurden, so muss mit 5 SFC 42-Aufrufen jede dieser Alarmverzögerungen auch wieder aufgehoben werden.

Parameter

Parameter	Deklaration	Datentyp	Speicherbereich	Beschreibung
RET_VAL	OUTPUT	INT	E, A, M, D, L	Anzahl der noch programmierten Verzögerungen nach Ablauf des SFC 42 bzw. wenn während der Bearbeitung der Funktion ein Fehler auftritt, enthält der Rückgabewert einen Fehlercode.

RET_VAL (Rückgabewert) Der Rückgabewert *RET_VAL* zeigt nach Ablauf des SFC die Anzahl der Bearbeitungsverzögerungen, also der Aufrufe des SFC 41 an (Die Alarmbearbeitung ist erst wieder freigegeben, wenn *RET_VAL* = 0 ist).

Wert	Beschreibung
8080h	Obwohl die Alarmbearbeitung bereits freigegeben war, wurde die Funktion aufgerufen.

SFC 43 - RE_TRIGR - Zykluszeitüberwachung neu starten

Beschreibung Mit dem SFC 43 RE_TRIGR (retrigger watchdog) wird die Zykluszeitüberwachung der CPU neu gestartet.

Parameter und Rückmeldungen Der SFC 43 besitzt weder Parameter noch werden Fehlerinformationen zurückgeliefert.

SFC 44 - REPL_VAL - Ersatzwert in AKKU1 übertragen

Beschreibung Mit dem SFC 44 REPL_VAL (replace value) wird ein Wert in den AKKU1 der fehlerverursachenden Programmebene übertragen. Der SFC 44 darf nur in Synchronfehler-OBs (OB 121, OB 122) aufgerufen werden.

Anwendungsbeispiel des SFC 44:

Wenn an einem Eingabemodul eine Störung vorhanden ist, die dazu führt, dass von dem Modul keine Werte mehr gelesen werden können, so wird nach jedem Zugriffsversuch auf dieses Modul der OB 122 gestartet. Mit Hilfe des SFC 44 REPL_VAL kann im OB 122 ein geeigneter Ersatzwert in den AKKU1 der unterbrochenen Programmebene übertragen werden, wobei die Programmbearbeitung dann mit diesem Ersatzwert fortgesetzt wird. Die Informationen, die für die Auswahl des Ersatzwertes notwendig sind (z.B. Baustein, in dem der Fehler auftrat, betroffene Adresse), können den lokalen Variablen des OB 122 entnommen werden.

Parameter

Parameter	Deklaration	Datentyp	Speicherbereich	Beschreibung
VAL	INPUT	DWORD	E, A, M, D, L, Konstante	Ersatzwert
RET_VAL	OUTPUT	INT	E, A, M, D, L	Tritt während der Bearbeitung der Funktion ein Fehler auf, enthält der Rückgabewert einen Fehlercode.

RET_VAL

(Rückgabewert)

Wert	Beschreibung
0000h	Es ist kein Fehler aufgetreten. Ein Ersatzwert wurde eingegeben.
8080h	Der SFC 44 wurde nicht von einem Synchronfehler-OB (OB 121, OB 122) aus aufgerufen.

SFC 46 - STP - CPU in STOP überführen

Beschreibung Mit dem SFC 46 STP wird die CPU in den Betriebszustand STOP überführt.

Parameter und Rückmeldungen Der SFC 46 besitzt weder Parameter noch werden Fehlerinformationen zurückgeliefert.

SFC 47 - WAIT - Verzögern des Anwenderprogramms

Beschreibung Mit dem SFC 47 WAIT werden Zeitverzögerungen bzw. Wartezeiten von 1 bis zu 32767µs in Ihrem Anwenderprogramm programmiert.

Unterbrechbarkeit Der SFC 47 kann durch höherpriorie OBs unterbrochen werden.



Hinweis!

Bei der mit dem SFC 47 programmierten Verzögerungszeit handelt es sich um eine Mindestzeit, die um die Ausführungszeit der eingeschachtelten Prioritätsklassen sowie um Systemlasten verlängert werden kann!

Parameter

Parameter	Deklaration	Datentyp	Speicherbereich	Beschreibung
WT	INPUT	INT	E, A, M, D, L, Konstante	Der Parameter <i>WT</i> enthält die Verzögerungszeit in µs.

Fehlerinformationen Der SFC 47 gibt keine spezifischen Fehlerinformationen aus.

SFC 49 - LGC_GADR - Steckplatz ermitteln

Beschreibung Mit dem SFC 49 LGC_GADR (convert logical address to geographical address) wird der zu einer logischen Adresse gehörende Modulsteckplatz sowie der Offset im Nutzdatenadressraum des Moduls ermittelt.

Parameter

Parameter	Deklaration	Datentyp	Speicherbereich	Beschreibung
IOID	INPUT	BYTE	E, A, M, D, L, Konstante	Kennung des Adressbereichs: 54h = Peripherie-Eingang (PE) 55h = Peripherie-Ausgang (PA) Handelt es sich um ein Mischmodul, ist die Bereichskennung der niedrigeren Adresse anzugeben. Bei gleichen Adressen ist 54h anzugeben.
LADDR	INPUT	WORD	E, A, M, D, L, Konstante	Logische Adresse. Bei einem Mischmodul ist die kleinere der beiden Adressen anzugeben.
RET_VAL	OUTPUT	INT	E, A, M, D, L	Tritt während der Bearbeitung der Funktion ein Fehler auf, enthält der Rückgabewert einen Fehlercode.
AREA	OUTPUT	BYTE	E, A, M, D, L	Bereichskennung: Sie gibt an, wie die restlichen Ausgangsparameter zu interpretieren sind.
RACK	OUTPUT	WORD	E, A, M, D, L	Siehe nächste Seite.
SLOT	OUTPUT	WORD	E, A, M, D, L	
SUBADDR	OUTPUT	WORD	E, A, M, D, L	

AREA *AREA* gibt an, wie die Ausgangsparameter *RACK*, *SLOT* und *SUBADDR* zu interpretieren sind. Diese Abhängigkeit ist nachfolgend aufgeführt.

Wert von AREA	System	Bedeutung von RACK, SLOT und SUBADDR
0	-	reserviert
1	Siemens S7-300	<i>RACK</i> : Nummer Baugruppenträger <i>SLOT</i> : Nummer Steckplatz <i>SUBADDR</i> : Adressoffset zur Anfangsadresse
2	Dezentrale Peripherie	<i>RACK</i> (Low-Byte): Stationsnummer <i>RACK</i> (High-Byte): DP-Mastersystem-ID <i>SLOT</i> : Nummer Steckplatz in Station <i>SUBADDR</i> : Adressoffset zur Anfangsadresse
3 ... 6	-	reserviert

RET_VAL (Rückgabewert) Tritt während der Bearbeitung der Funktion ein Fehler auf, enthält der Rückgabewert einen Fehlercode.

Wert	Beschreibung
0000h	Es ist kein Fehler aufgetreten.
8090h	Angegebene logische Adresse ungültig oder unzulässiger Wert beim Parameter <i>IOID</i>

SFC 50 - RD_LGADR - Alle Adressen eines Moduls lesen

Beschreibung Mit dem SFC 50 RD_LGADR (read module logical addresses) werden alle vereinbarten logischen Adressen eines Moduls ermittelt, ausgehend von einer logischen Adresse dieses Moduls.
Die Zuordnung von logischen Adressen zum Modul muss vorher von Ihnen projiziert worden sein. Die ermittelten logischen Adressen werden in das Feld *PEADDR* bzw. in das Feld *PAADDR* in aufsteigender Reihenfolge eingetragen.

Parameter

Parameter	Deklaration	Datentyp	Speicherbereich	Beschreibung
IOID	INPUT	BYTE	E, A, M, D, L, Konstante	Bereichskennung: 54h = Peripherie-Eingang (PE) 55h = Peripherie-Ausgang (PA)
LADDR	INPUT	WORD	E, A, M, D, L, Konstante	Eine logische Adresse
RET_VAL	OUTPUT	INT	E, A, M, D, L	Tritt während der Bearbeitung der Funktion ein Fehler auf, enthält der Rückgabewert einen Fehlercode.
PEADDR	OUTPUT	ANY	E, A, M, D, L	Feld für die PE-Adressen, Feldelemente müssen vom Datentyp WORD sein.
PECOUNT	OUTPUT	INT	E, A, M, D, L	Anzahl der zurückgelieferten PE-Adressen
PAADDR	OUTPUT	ANY	E, A, M, D, L	Feld für die PA-Adressen, Feldelemente müssen vom Datentyp WORD sein.
PACOUNT	OUTPUT	INT	E, A, M, D, L	Anzahl der zurückgelieferten PA-Adressen

RET_VAL (Rückgabewert) Tritt während der Bearbeitung der Funktion ein Fehler auf, enthält der Rückgabewert einen Fehlercode.

Wert	Beschreibung
0000h	Es ist kein Fehler aufgetreten.
8090h	Angegebene logische Adresse ungültig oder unzulässiger Wert beim Parameter <i>IOID</i> .
80A0h	Fehler beim Ausgangsparameter <i>PEADDR</i> : Der Datentyp der Feldelemente ist nicht WORD.
80A1h	Fehler beim Ausgangsparameter <i>PAADDR</i> : Der Datentyp der Feldelemente ist nicht WORD.
80A2h	Fehler beim Ausgangsparameter <i>PEADDR</i> : Das angegebene Feld konnte nicht alle logischen Adressen aufnehmen.
80A3h	Fehler beim Ausgangsparameter <i>PAADDR</i> : Das angegebene Feld konnte nicht alle logischen Adressen aufnehmen.

SFC 51 - RDSYSST - Auslesen der Informationen der SZL

Beschreibung Mit dem SFC 51 RDSYSST (read system status) können Sie eine Teilliste bzw. einen Teillistenauszug der SZL (**S**ystemzustandsliste) anfordern. Hierbei bestimmen Sie mit den Parametern *SZL_ID* und *INDEX* was Sie auslesen möchten. Der *INDEX* ist nicht immer erforderlich. Er dient der Bestimmung eines Objekts innerhalb einer Teilliste. Durch Setzen von *REQ* starten Sie die Abfrage. Sobald *BUSY* = 0 zurückgemeldet wird, liegen die Daten im Zielbereich *DR* ab. Informationen zur SZL finden Sie im Teil "Systemzustandsliste SZL".

Parameter

Parameter	Deklaration	Datentyp	Speicherbereich	Beschreibung
REQ	INPUT	BOOL	E, A, M, D, L, Konstante	REQ = 1: Anstoß der Bearbeitung
SZL_ID	INPUT	WORD	E, A, M, D, L, Konstante	SZL_ID der Teilliste oder des Teillistenauszugs
INDEX	INPUT	WORD	E, A, M, D, L, Konstante	Typ oder Nummer eines Objekts in einer Teilliste
RET_VAL	OUTPUT	INT	E, A, M, D, L	Tritt während der Bearbeitung der Funktion ein Fehler auf, enthält der Rückgabewert einen Fehlercode
BUSY	OUTPUT	BOOL	E, A, M, D, L	BUSY = 1: Lesevorgang noch nicht abgeschlossen
SZL_HEADER	OUTPUT	STRUCT	D, L	Struktur mit 2 WORD-Typen: LENGTHDR: Länge Datensatz WORD N_DR: Anzahl vorhandener zugehöriger Datensätze (bei Zugriff auf Teillistenkopfinfo) oder Anzahl der in DR übertragenen Datensätze.
DR	OUTPUT	ANY	E, A, M, D, L	Zielbereich für die gelesene SZL-Teilliste bzw. den gelesenen SZL-Teillistenauszug: Falls Sie nur die SZL-Teillistenkopfinfo einer SZL-Teilliste ausgelesen haben, dürfen Sie DR nicht auswerten, sondern nur SZL_HEADER. Andernfalls gibt das Produkt aus LENGTHDR und N_DR an, wie viele Bytes in DR eingetragen wurden.

RET_VAL (Rückgabewert) tritt während der Bearbeitung der Funktion ein Fehler auf, enthält der Rückgabewert einen Fehlercode.

Wert	Beschreibung
0000h	kein Fehler
0081h	Länge des Ergebnisfeldes zu klein. Es werden trotzdem so viele Datensätze wie möglich geliefert. Der SZL-Header zeigt diese Anzahl an.
7000h	Erstaufruf mit <i>REQ</i> = 0: keine Datenübertragung aktiv; <i>BUSY</i> = 0.
7001h	Erstaufruf mit <i>REQ</i> = 1: Datenübertragung angestoßen; <i>BUSY</i> = 1.
7002h	Zwischenaufruf (<i>REQ</i> irrelevant): Datenübertragung bereits aktiv; <i>BUSY</i> = 1.
8081h	Länge des Ergebnisfeldes zu klein. Platz reicht nicht für einen Datensatz.
8082h	<i>SZL_ID</i> ist falsch oder in der CPU bzw. in dem SFC unbekannt.
8083h	<i>INDEX</i> ist falsch oder nicht erlaubt.
8085h	Die Information ist systembedingt momentan nicht verfügbar, z.B. wegen Ressourcenmangels.
8086h	Datensatz ist nicht lesbar wegen eines Systemfehlers.
8087h	Datensatz ist nicht lesbar, weil das Modul nicht vorhanden ist oder nicht quittiert.
8088h	Datensatz ist nicht lesbar, weil die tatsächliche Typkennung von der Solltypkennung abweicht.
8089h	Datensatz ist nicht lesbar, weil das Modul nicht diagnosefähig ist.
80A2h	DP-Protokollfehler - Layer-2-Fehler (temporärer Fehler).
80A3h	DP-Protokollfehler bei User-Interface/User (temporärer Fehler)
80A4h	Kommunikation am Bus ist gestört. Fehler tritt auf zwischen CPU und externer DP-Anschaltung (temporärer Fehler).
80C5h	Dezentrale Peripherie nicht verfügbar (temporärer Fehler).

SFC 52 - WR_USMSG - Eintrag in Diagnosepuffer schreiben

Beschreibung	Mit dem SFC 52 WR_USMSG (write user element in diagnosis buffer) wird ein anwenderdefiniertes Diagnoseereignis in den Diagnosepuffer geschrieben.
Diagnosemeldung senden	Um zu prüfen, ob das Senden von anwenderdefinierten Diagnosemeldungen momentan möglich ist, rufen Sie den SFC 51 "RDSYSST" mit den Parametern <i>SZL_ID</i> = 0132h und <i>INDEX</i> = 0005h auf. Enthält das vierte Wort des dabei gelieferten Datensatzes den Wert 1 ist das Senden möglich, enthält es den Wert 0 ist das Senden nicht möglich.
Sendepuffer voll	Die Diagnosemeldung kann nur dann in den Sendepuffer eingetragen werden, wenn der Sendepuffer nicht voll ist. Maximal können 50 Einträge im Sendepuffer abgelegt werden. Ist der Sendepuffer voll, dann: <ul style="list-style-type: none">• erfolgt trotzdem der Eintrag des Diagnoseereignisses in den Diagnosepuffer.• wird die entsprechende Fehlermeldung (8092h) im Parameter <i>RET_VAL</i> angezeigt.
Teilnehmer nicht angemeldet	Wenn eine anwenderdefinierte Diagnosemeldung gesendet werden soll und kein Teilnehmer angemeldet ist, dann <ul style="list-style-type: none">• erfolgt trotzdem der Eintrag des anwenderdefinierten Diagnoseereignisses in den Diagnosepuffer,• wird die entsprechende Fehlermeldung (0091h oder 8091h) im Parameter <i>RET_VAL</i> angezeigt.

Aufbau eines Eintrags

Ein Eintrag in den Diagnosepuffer hat intern den folgenden Aufbau:

Byte	Inhalt
1, 2	Ereignis-ID
3	Prioritätsklasse
4	OB-Nummer
5, 6	reserviert
7, 8	Zusatzinformation 1
9, 10, 11, 12	Zusatzinformation 2
13 ... 20	Zeitstempel: Der Zeitstempel ist vom Datentyp Date_and_Time.

Ereignis-ID

Jedem Ereignis ist eine Ereignis-ID zugeordnet.

**Zusatz-
informationen**

Die Zusatzinformationen beinhalten zusätzliche Informationen zum Ereignis. Diese Zusatzinformationen können für jedes Ereignis unterschiedlich sein. Wenn ein Diagnoseereignis erzeugt wird, dann kann der Inhalt dieser Einträge selbst bestimmt werden.

Wenn eine anwenderdefinierte Diagnosemeldung versendet wird, können die Zusatzinformationen als Begleitwerte in den (Ereignis-ID-spezifischen) Meldetext integriert werden.

Parameter

Parameter	Deklaration	Datentyp	Speicherbereich	Beschreibung
SEND	INPUT	BOOL	E, A, M, D, L, Konstante	Senden der anwenderdefinierten Diagnosemeldung an alle angemeldeten Teilnehmer freigeben
EVENTN	INPUT	WORD	E, A, M, D, L, Konstante	Ereignis-ID. Sie vergeben die Ereignis-ID. Die Vorgabe erfolgt nicht durch den Meldeserver.
INFO1	INPUT	ANY	E, A, M, D, L	Zusatzinformation 1 Wort lang
INFO2	INPUT	ANY	E, A, M, D, L	Zusatzinformation 2 Worte lang
RET_VAL	OUTPUT	INT	E, A, M, D, L	Tritt während der Bearbeitung der Funktion ein Fehler auf, enthält der Rückgabewert einen Fehlercode.

SEND

Wenn *SEND* den Wert 1 hat, wird die anwenderdefinierte Diagnosemeldung an alle dafür angemeldeten Teilnehmer gesendet. Das Senden erfolgt nur unter der Voraussetzung, dass mindestens ein Teilnehmer angemeldet und der Sendepuffer nicht voll ist. Das Senden erfolgt asynchron zum Anwenderprogramm.

EVENTN

In *EVENTN* wird die Ereignis-ID des Anwenderereignisses eingetragen. Die Ereignis-IDs dürfen nur in der Form 8xyzh, 9xyzh, Axyzh und Bxyzh eingetragen werden, wobei die IDs der Form 8xyzh und 9xyzh zu den vordefinierten Ereignissen und die IDs der Form Axyzh und Bxyzh zu den frei definierten Ereignissen gehören.

Ein kommendes Ereignis wird durch $x = 1$, ein gehendes Ereignis durch $x = 0$ gekennzeichnet.

Bei den Ereignissen der Klasse A und B ist yz die in der Meldungsprojektierung für die zugehörige Meldung vergebene Meldungsnummer in hexadezimaler Darstellung.

INFO1

INFO1 enthält eine Information, die ein Wort lang ist. Folgende Datentypen sind zulässig:

- WORD
- INT
- ARRAY [0...1] OF CHAR

INFO1 kann als Begleitwert in den Meldetext integriert werden und damit können der Meldung aktuelle Informationen hinzugefügt werden.

INFO2 *INFO2* enthält eine Information, die zwei Worte lang ist. Folgende Datentypen sind zulässig:

- DWORD
- DINT
- REAL
- TIME
- ARRAY [0...3] OF CHAR

INFO2 kann als Begleitwert in den Meldetext integriert werden und damit können der Meldung aktuelle Informationen hinzugefügt werden.

RET_VAL (Rückgabewert) Tritt während der Bearbeitung der Funktion ein Fehler auf, enthält der Rückgabewert einen Fehlercode.

Wert	Beschreibung
0000h	kein Fehler
0091h	kein Teilnehmer angemeldet (Eintrag des Diagnoseereignisses in den Diagnosepuffer erfolgt)
8083h	Datentyp <i>INFO1</i> nicht zulässig
8084h	Datentyp <i>INFO2</i> nicht zulässig
8085h	<i>EVENTN</i> nicht zulässig
8086h	Länge von <i>INFO1</i> nicht zulässig
8087h	Länge von <i>INFO2</i> nicht zulässig
8091h	Fehlermeldung identisch mit Fehlercode 0091h
8092h	Senden derzeit nicht möglich, Sendepuffer ist voll (Eintrag des Diagnoseereignisses in den Diagnosepuffer erfolgt)

SFC 54 - RD_DPARM - Vordefinierte Parameter lesen

Beschreibung Mit dem SFC 54 RD_DPARM (read defined parameter) wird der Datensatz mit der Nummer *RECNUM* des adressierten Moduls aus dem zugehörigen SDB1xy gelesen.

Durch den Parameter *RECORD* wird der Zielbereich festgelegt, in den der gelesene Datensatz eingetragen wird.

Parameter

Parameter	Deklaration	Datentyp	Speicherbereich	Beschreibung
IOID	INPUT	BYTE	E, A, M, D, L, Konstante	Kennung des Adressbereichs: 54h = Peripherie Eingang (PE) 55h = Peripherie Ausgang (PA) Handelt es sich um ein Mischmodul, ist die Bereichskennung der niedrigeren Adresse anzugeben. Bei gleichen Adressen ist 54h anzugeben.
LADDR	INPUT	WORD	E, A, M, D, L, Konstante	Logische Basisadresse des Moduls. Bei einem Mischmodul ist die kleinere der beiden Adressen anzugeben.
RECNUM	INPUT	BYTE	E, A, M, D, L, Konstante	Datensatznummer (zulässige Werte: 0 ... 240)
RET_VAL	OUTPUT	INT	E, A, M, D, L	Tritt während der Bearbeitung der Funktion ein Fehler auf, enthält der Rückgabewert einen Fehlercode. Zusätzlich: Länge des gelesenen Datensatzes in Bytes, falls der gelesene Datensatz in den Zielbereich passt und bei der Übertragung kein Fehler auftrat.
RECORD	OUTPUT	ANY	E, A, M, D, L	Zielbereich für den gelesenen Datensatz. Es ist nur der Datentyp BYTE zulässig.

**RET_VAL
(Rückgabewert)**

Bei *RET_VAL* = 8xxxh sind zwei Fälle zu unterscheiden:

- Temporäre Fehler (Fehlercodes 80A2h ... 80A4h, 80C3h):

Bei dieser Fehlerart besteht die Möglichkeit, dass sich der Fehler ohne Ihr Zutun behebt, es wäre also sinnvoll, den SFC erneut (ggf. mehrfach) aufzurufen.

Beispiel für temporäre Fehler: Benötigte Betriebsmittel sind momentan belegt (80C3h).

- Permanente Fehler (Fehlercodes 809xh, 80A1h, 80Bxh, 80Dxh):

Bei dieser Fehlerart kann der Fehler nicht ohne Ihr Zutun behoben werden. Ein erneuter Aufruf des SFC ist erst wieder sinnvoll, wenn der Fehler beseitigt wurde.

Beispiel für permanente Fehler: Falsche Länge des zu übertragenden Datensatzes (80B1h).

Wert	Beschreibung
7000h	Erstauf Ruf mit <i>REQ</i> = 0: keine Datenübertragung aktiv; <i>BUSY</i> hat den Wert 0.
7001h	Erstauf Ruf mit <i>REQ</i> = 1: Datenübertragung angestoßen; <i>BUSY</i> hat den Wert 1.
7002h	Zwischenauf Ruf (<i>REQ</i> irrelevant): Datenübertragung bereits aktiv; <i>BUSY</i> hat den Wert 1.
8090h	Angegebene logische Basisadresse ungültig: Es ist keine Zuordnung im SDB1/SDB2x vorhanden, oder es ist keine Basisadresse.
8092h	In ANY-Referenz ist eine Typangabe ungleich BYTE angegeben.
8093h	Für das über <i>LADDR</i> und <i>IOID</i> ausgewählte Modul ist dieser SFC nicht zulässig.
80B1h	Die Länge des durch <i>RECORD</i> festgelegten Zielbereichs ist zu klein.
80D0h	Im zugehörigen SDB ist kein Eintrag für das Modul vorhanden.
80D1h	Die Datensatznummer ist im zugehörigen SDB für das Modul nicht projektiert.
80D2h	Das Modul ist laut Typkennung nicht parametrierbar.
80D3h	Auf den SDB kann nicht zugegriffen werden, da er nicht vorhanden ist.
80D4h	SDB-Strukturfehler: SDB-interner Zeiger zeigt außerhalb SDB.

SFC 55 - WR_PARM - Dynamische Parameter schreiben

Beschreibung Mit dem SFC 55 WR_PARM (write parameter) wird der Datensatz *RECORD* zum adressierten Modul übertragen. Die Parameter dieses Moduls im zugehörigen SDB werden, falls diese dort vorhanden sind, nicht durch die Parameter, die zum Modul übertragen werden, überschrieben. Dieser SFC kann für Digital-, Analogmodule, FMs, CPs und über PROFIBUS DP-V1 eingesetzt werden.

Voraussetzungen Es ist darauf zu achten das der zu übertragende Datensatz nicht statisch ist:

- Datensatz 0 darf nicht verwendet werden, da Datensatz 0 systemweit statisch ist.
- Ist der Datensatz in den SDBs 100 ... 129 vermerkt, dann darf das Statisch-Bit nicht gesetzt sein.

Parameter

Parameter	Deklaration	Datentyp	Speicherbereich	Beschreibung
REQ	INPUT	BOOL	E, A, M, D, L, Konstante	REQ = 1: Anforderung zum Schreiben
IOID	INPUT	BYTE	E, A, M, D, L, Konstante	Kennung des Adressbereichs: 54h = Peripherie Eingang (PE) 55h = Peripherie Ausgang (PA) Handelt es sich um ein Mischmodul, ist die Bereichskennung der niedrigeren Adresse anzugeben. Bei gleichen Adressen ist 54h anzugeben.
LADDR	INPUT	WORD	E, A, M, D, L, Konstante	Logische Basisadresse des Moduls. Bei einem Mischmodul ist die kleinere der beiden Adressen anzugeben.
RECNUM	INPUT	BYTE	E, A, M, D, L, Konstante	Datensatznummer (zulässige Werte: 0 ... 240)
RECORD	INPUT	ANY	E, A, M, D, L	Datensatz
RET_VAL	OUTPUT	INT	E, A, M, D, L	Tritt während der Bearbeitung der Funktion ein Fehler auf, enthält der Rückgabewert einen Fehlercode.
BUSY	OUTPUT	BOOL	E, A, M, D, L	BUSY = 1: Der Schreibvorgang ist noch nicht beendet.

RECORD Beim ersten Aufruf des SFC werden die zu übertragenden Daten aus dem Parameter *RECORD* gelesen. Sollte die Übertragung des Datensatzes jedoch länger als einen Aufruf dauern, dann ist bei den Folgeaufrufen des SFC (zum gleichen Auftrag) der Inhalt des Parameters *RECORD* nicht mehr relevant.

RET_VAL (Rückgabewert) Bei *RET_VAL* = 8xxxh sind zwei Fälle zu unterscheiden:

- Temporäre Fehler (Fehlercodes 80A2h ... 80A4h, 80C3h):
Bei dieser Fehlerart besteht die Möglichkeit, dass sich der Fehler ohne Ihr Zutun behebt, es wäre also sinnvoll, den SFC erneut (ggf. mehrfach) aufzurufen.
Beispiel für temporäre Fehler: Benötigte Betriebsmittel sind momentan belegt (80C3h).
- Permanente Fehler (Fehlercodes 809xh, 80A1h, 80Bxh, 80Dxh):
Bei dieser Fehlerart kann der Fehler nicht ohne Ihr Zutun behoben werden. Ein erneuter Aufruf des SFC ist erst wieder sinnvoll, wenn der Fehler beseitigt wurde.
Beispiel für permanente Fehler: Falsche Länge des zu übertragenden Datensatzes (80B1h).

Wert	Beschreibung
7000h	Erstaufwurf mit <i>REQ</i> = 0: keine Datenübertragung aktiv; <i>BUSY</i> hat den Wert 0.
7001h	Erstaufwurf mit <i>REQ</i> = 1: Datenübertragung angestoßen; <i>BUSY</i> hat den Wert 1.
7002h	Zwischenaufwurf (<i>REQ</i> irrelevant): Datenübertragung bereits aktiv; <i>BUSY</i> hat den Wert 1.
8090h	Angegebene logische Basisadresse ungültig: Es ist keine Zuordnung im SDB1/SDB2x vorhanden, oder es ist keine Basisadresse.
8092h	In ANY-Referenz ist eine Typangabe ungleich BYTE angegeben.
8093h	Für die über <i>LADDR</i> und <i>IOID</i> ausgewählte Modul ist dieser SFC nicht zulässig.
80A1h	Negative Quittung beim Senden des Datensatzes zum Modul (Modul während des Sendens gezogen oder Modul defekt)
80A2h	DP-Protokollfehler bei Layer 2, evtl. Hardware-/ Schnittstellenfehler im DP-Slave
80A3h	DP-Protokollfehler bei User Interface/User
80A4h	Kommunikation gestört (Fehler tritt auf zwischen CPU und externer DP-Anschaltung)
80B0h	SFC für Modultyp nicht möglich oder Modul kennt den Datensatz nicht

Fortsetzung ...

... Fortsetzung

Fehlercode	Beschreibung
80B1h	Die Länge des durch <i>RECORD</i> festgelegten Zielbereichs ist zu klein.
80B2h	Der projektierte Steckplatz ist nicht belegt.
80B3h	Ist-Modultyp ungleich Soll-Modultyp im SDB1.
80C1h	Die Daten des auf dem Modul vorangegangenen Schreibauftrags für denselben Datensatz sind von dem Modul noch nicht verarbeitet.
80C2h	Das Modul bearbeitet momentan das mögliche Maximum an Aufträgen für eine CPU.
80C3h	Benötigte Betriebsmittel (Speicher etc.) sind momentan belegt.
80C4h	Kommunikationsfehler
80C5h	Dezentrale Peripherie nicht verfügbar.
80C6h	Datensatzübertragung wurde abgebrochen wegen Prioritätsklassenabbruchs
80D0h	Im zugehörigen SDB ist kein Eintrag für das Modul vorhanden.
80D1h	Die Datensatznummer ist im zugehörigen SDB für das Modul nicht projektiert.
80D2h	Das Modul ist laut Typkennung nicht parametrierbar.
80D3h	Auf den SDB kann nicht zugegriffen werden, da er nicht vorhanden ist.
80D4h	SDB-Strukturfehler: SDB-interner Zeiger zeigt außerhalb SDB.
80D5h	Der Datensatz ist statisch.

SFC 56 - WR_DPARM - Vordefinierte Parameter schreiben

Beschreibung Mit dem SFC 56 WR_DPARM (write default parameter) wird der Datensatz mit der Nummer *RECNUM* aus dem zuständigen SDB zum adressierten Modul übertragen. Ob es sich dabei um einen statischen oder dynamischen Datensatz handelt ist ohne Bedeutung.

Dieser SFC kann für Digital-, Analogmodule, FMs, CPs und über PROFIBUS DP-V1 eingesetzt werden.

Parameter

Parameter	Deklaration	Datentyp	Speicherbereich	Beschreibung
REQ	INPUT	BOOL	E, A, M, D, L, Konstante	REQ = 1: Anforderung zum Schreiben
IOID	INPUT	BYTE	E, A, M, D, L, Konstante	Kennung des Adressbereichs: 54h = Peripherie Eingang (PE) 55h = Peripherie Ausgang (PA) Handelt es sich um ein Mischmodul, ist die Bereichskennung der niedrigeren Adresse anzugeben. Bei gleichen Adressen ist 54h anzugeben.
LADDR	INPUT	WORD	E, A, M, D, L, Konstante	Logische Basisadresse des Moduls. Bei einem Mischmodul ist die kleinere der beiden Adressen anzugeben.
RECNUM	INPUT	BYTE	E, A, M, D, L, Konstante	Datensatznummer (zulässige Werte: 0 ... 240)
RET_VAL	OUTPUT	INT	E, A, M, D, L	Tritt während der Bearbeitung der Funktion ein Fehler auf, enthält der Rückgabewert einen Fehlercode.
BUSY	OUTPUT	BOOL	E, A, M, D, L	BUSY = 1: Der Schreibvorgang ist noch nicht beendet.

**RET_VAL
(Rückgabewert)**

Bei *RET_VAL* = 8xxxh sind zwei Fälle zu unterscheiden:

- Temporäre Fehler (Fehlercodes 80A2h ... 80A4h, 80C3h):
Bei dieser Fehlerart besteht die Möglichkeit, dass sich der Fehler ohne Ihr Zutun behebt, es wäre also sinnvoll den SFC erneut (ggf. mehrfach) aufzurufen.
Beispiel für temporäre Fehler: Benötigte Betriebsmittel sind momentan belegt (80C3h).
- Permanente Fehler (Fehlercodes 8093h, 80A1h, 80B3h, 80D3h):
Bei dieser Fehlerart kann der Fehler nicht ohne Ihr Zutun behoben werden. Ein erneuter Aufruf des SFC ist erst wieder sinnvoll, wenn der Fehler beseitigt wurde.
Beispiel für permanente Fehler: Falsche Länge des zu übertragenden Datensatzes (80B1h).

Wert	Beschreibung
7000h	Erstaufruf mit <i>REQ</i> = 0: keine Datenübertragung aktiv; <i>BUSY</i> hat den Wert 0.
7001h	Erstaufruf mit <i>REQ</i> = 1: Datenübertragung angestoßen; <i>BUSY</i> hat den Wert 1.
7002h	Zwischenaufruf (<i>REQ</i> irrelevant): Datenübertragung bereits aktiv; <i>BUSY</i> hat den Wert 1.
8090h	Angegebene logische Basisadresse ungültig: Es ist keine Zuordnung im SDB1/SDB2x vorhanden, oder es ist keine Basisadresse.
8093h	Für das über <i>LADDR</i> und <i>IOID</i> ausgewählte Modul ist dieser SFC nicht zulässig.
80A1h	Negative Quittung beim Senden des Datensatzes zum Modul (Modul während des Sendens gezogen oder Modul defekt).
80A2h	DP-Protokollfehler bei Layer 2, evtl. Hardware-/ Schnittstellenfehler im DP-Slave
80A3h	DP-Protokollfehler bei User Interface/User
80A4h	Kommunikation gestört (Fehler tritt auf zwischen CPU und externer DP-Anschaltung).
80B0h	SFC für Modultyp nicht möglich, oder das Modul kennt den Datensatz nicht.
80B1h	Die Länge des durch <i>RECORD</i> festgelegten Zielbereichs ist zu klein.
80B2h	Der projektierte Steckplatz ist nicht belegt.
80B3h	Ist-Modultyp ungleich Soll-Modultyp im SDB1.
80C1h	Die Daten des auf dem Modul vorangegangenen Schreibauftrags für denselben Datensatz sind von dem Modul noch nicht verarbeitet.
80C2h	Das Modul bearbeitet momentan das mögliche Maximum an Aufträgen für eine CPU.
80C3h	Benötigte Betriebsmittel (Speicher etc.) sind momentan belegt.
80C4h	Kommunikationsfehler
80C5h	Dezentrale Peripherie nicht verfügbar.
80C6h	Datensatzübertragung wurde abgebrochen wegen Prioritätsklassenabbruchs.
80D0h	Im zugehörigen SDB ist kein Eintrag für das Modul vorhanden.
80D1h	Die Datensatznummer ist im zugehörigen SDB für das Modul nicht projektiert.
80D2h	Das Modul ist laut Typkennung nicht parametrierbar.
80D3h	Auf den SDB kann nicht zugegriffen werden, da er nicht vorhanden ist.
80D4h	SDB-Strukturfehler: SDB-interner Zeiger zeigt außerhalb SDB.

SFC 57 - PARM_MOD - Modul parametrieren

Beschreibung Mit dem SFC 57 PARM_MOD (parameterize module) werden alle Datensätze eines Moduls, die im zugehörigen SDB projiziert wurden, zum Modul übertragen. Ob es sich dabei um statische oder dynamische Datensätze handelt ist ohne Bedeutung.

Dieser SFC kann für Digital-, Analogmodule, FMs, CPs und über PROFIBUS DP-V1 eingesetzt werden.

Parameter

Parameter	Deklaration	Datentyp	Speicherbereich	Beschreibung
REQ	INPUT	BOOL	E, A, M, D, L, Konstante	REQ = 1: Anforderung zum Schreiben
IOID	INPUT	BYTE	E, A, M, D, L, Konstante	Kennung des Adressbereichs: 54h = Peripherie Eingang (PE) 55h = Peripherie Ausgang (PA) Handelt es sich um eine Mischmodul, ist die Bereichskennung der niedrigeren Adresse anzugeben. Bei gleichen Adressen ist 54h anzugeben.
LADDR	INPUT	WORD	E, A, M, D, L, Konstante	Logische Basisadresse des Moduls. Bei einem Mischmodul ist die kleinere der beiden Adressen anzugeben.
RET_VAL	OUTPUT	INT	E, A, M, D, L	Tritt während der Bearbeitung der Funktion ein Fehler auf, enthält der Rückgabewert einen Fehlercode.
BUSY	OUTPUT	BOOL	E, A, M, D, L	BUSY = 1: Der Schreibvorgang ist noch nicht beendet.

**RET_VAL
(Rückgabewert)**

Bei *RET_VAL* = 8xxxh sind zwei Fälle zu unterscheiden:

- Temporäre Fehler (Fehlercodes 80A2h ... 80A4h, 80C3h):
Bei dieser Fehlerart besteht die Möglichkeit, dass sich der Fehler ohne Ihr Zutun behebt, es wäre also sinnvoll den SFC erneut (ggf. mehrfach) aufzurufen.
Beispiel für temporäre Fehler: Benötigte Betriebsmittel sind momentan belegt (80C3h).
- Permanente Fehler (Fehlercodes 8093h, 80A1h, 80B3h, 80D3h):
Bei dieser Fehlerart kann der Fehler nicht ohne Ihr Zutun behoben werden. Ein erneuter Aufruf des SFC ist erst wieder sinnvoll, wenn der Fehler beseitigt wurde.
Beispiel für permanente Fehler: Falsche Länge des zu übertragenden Datensatzes (80B1h).

Wert	Beschreibung
7000h	Erstaufruf mit <i>REQ</i> = 0: keine Datenübertragung aktiv; <i>BUSY</i> hat Wert 0.
7001h	Erstaufruf mit <i>REQ</i> = 1: Datenübertragung angestoßen; <i>BUSY</i> hat den Wert 1.
7002h	Zwischenaufruf (<i>REQ</i> irrelevant): Datenübertragung bereits aktiv; <i>BUSY</i> hat den Wert 1.
8090h	Angegebene logische Basisadresse ungültig: Es ist keine Zuordnung im SDB1/SDB2x vorhanden, oder es ist keine Basisadresse.
8093h	Für das über <i>LADDR</i> und <i>IOID</i> ausgewählte Modul ist dieser SFC nicht zulässig.
80A1h	Negative Quittung beim Senden des Datensatzes zum Modul (Modul während des Sendens gezogen oder Modul defekt)
80A2h	DP-Protokollfehler bei Layer 2, evtl. Hardware-/ Schnittstellenfehler im DP-Slave
80A3h	DP-Protokollfehler bei User Interface/User
80A4h	Kommunikation gestört (Fehler tritt auf zwischen CPU und externer DP-Anschaltung)
80B0h	SFC für Modultyp nicht möglich, oder das Modul kennt Datensatz nicht.
80B1h	Die Länge des durch <i>RECORD</i> festgelegten Zielbereichs ist zu klein.
80B2h	Der projektierte Steckplatz ist nicht belegt.
80B3h	Ist-Modultyp ungleich Soll-Modultyp im SDB1
80C1h	Die Daten des auf dem Modul vorangegangenen Schreibauftrags für denselben Datensatz sind von dem Modul noch nicht verarbeitet.
80C2h	Das Modul bearbeitet momentan das mögliche Maximum an Aufträgen für eine CPU.
80C3h	Benötigte Betriebsmittel (Speicher etc.) sind momentan belegt.
80C4h	Kommunikationsfehler
80C5h	Dezentrale Peripherie nicht verfügbar
80C6h	Datensatzübertragung wurde abgebrochen wegen Prioritätsklassenabbruchs.
80D0h	Im zugehörigen SDB ist kein Eintrag für das Modul vorhanden.
80D1h	Die Datensatznummer ist im zugehörigen SDB für das Modul nicht projektiert.
80D2h	Das Modul ist laut Typkennung nicht parametrierbar.
80D3h	Auf den SDB kann nicht zugegriffen werden, da er nicht vorhanden ist.
80D4h	SDB-Strukturfehler: SDB-interner Zeiger zeigt außerhalb SDB.

SFC 58 - WR_REC - Datensatz schreiben

Beschreibung Mit dem SFC 58 WR_REC (write record) wird der Datensatz *RECORD* zum adressierten Modul übertragen.

Der Schreibvorgang wird gestartet, wenn beim Aufruf des SFC 58 der Eingangsparameter *REQ* mit 1 belegt wird. Der Ausgangsparameter *BUSY* liefert den Wert 0, wenn der Schreibvorgang sofort ausgeführt werden konnte. Wenn der Schreibvorgang noch nicht abgeschlossen ist, liefert *BUSY* den Wert 1.

Dieser SFC kann für Digital-, Analogmodule, FMs, CPs und über PROFIBUS DP-V1 eingesetzt werden.

Systembedingt ist dieser Baustein nicht unterbrechbar!

Parameter

Parameter	Deklaration	Datentyp	Speicherbereich	Beschreibung
REQ	INPUT	BOOL	E, A, M, D, L, Konstante	REQ = 1: Anforderung zum Schreiben
IOID	INPUT	BYTE	E, A, M, D, L, Konstante	Kennung des Adressbereichs: 54h = Peripherie Eingang (PE) 55h = Peripherie Ausgang (PA) Handelt es sich um eine Mischmodul, ist die Bereichskennung der niedrigeren Adresse anzugeben. Bei gleichen Adressen ist 54h anzugeben.
LADDR	INPUT	WORD	E, A, M, D, L, Konstante	Logische Basisadresse des Moduls. Bei einem Mischmodul ist die kleinere der beiden Adressen anzugeben.
RECNUM	INPUT	BYTE	E, A, M, D, L, Konstante	Datensatznummer (zulässige Werte: 2 ... 240)
RECORD	INPUT	ANY	E, A, M, D, L	Datensatz. Es ist nur der Datentyp BYTE zulässig.
RET_VAL	OUTPUT	INT	E, A, M, D, L	Tritt während der Bearbeitung der Funktion ein Fehler auf, enthält der Rückgabewert einen Fehlercode.
BUSY	OUTPUT	BOOL	E, A, M, D, L	BUSY = 1: Der Schreibvorgang ist noch nicht beendet.

RECORD

Beim ersten Aufruf des SFC werden die zu übertragenden Daten aus dem Parameter *RECORD* gelesen. Sollte die Übertragung des Datensatzes jedoch länger als einen Aufruf dauern, dann ist bei den Folgeaufrufen des SFC (zum gleichen Auftrag) der Inhalt des Parameters *RECORD* nicht mehr relevant.

**RET_VAL
(Rückgabewert)**

Bei *RET_VAL* = 8xxxh sind zwei Fälle zu unterscheiden:

- Temporäre Fehler (Fehlercodes 80A2h ... 80A4h, 80C3h):
Bei dieser Fehlerart besteht die Möglichkeit, dass sich der Fehler ohne Ihr Zutun behebt, es wäre also sinnvoll den SFC erneut (ggf. mehrfach) aufzurufen.
Beispiel für temporäre Fehler: Benötigte Betriebsmittel sind momentan belegt (80C3h).
- Permanente Fehler (Fehlercodes 809xh, 80A0h, 80A1h, 80Bxh):
Bei dieser Fehlerart kann der Fehler nicht ohne Ihr Zutun behoben werden. Ein erneuter Aufruf des SFC ist erst wieder sinnvoll, wenn der Fehler beseitigt wurde.
Beispiel für permanente Fehler: Falsche Länge des zu übertragenden Datensatzes (80B1h).

Wert	Beschreibung
7000h	Erstaufwurf mit <i>REQ</i> = 0: keine Datenübertragung aktiv; <i>BUSY</i> hat den Wert 0.
7001h	Erstaufwurf mit <i>REQ</i> = 1: Datenübertragung angestoßen; <i>BUSY</i> hat den Wert 1.
7002h	Zwischenaufwurf (<i>REQ</i> irrelevant): Datenübertragung bereits aktiv; <i>BUSY</i> hat den Wert 1.
8090h	Angegebene logische Basisadresse ungültig: Es ist keine Zuordnung im SDB1/SDB2x vorhanden, oder es ist keine Basisadresse.
8092h	In ANY-Referenz ist eine Typangabe ungleich BYTE angegeben.
8093h	Für das über <i>LADDR</i> und <i>IOID</i> ausgewählte Modul ist dieser SFC nicht zulässig.
80A1h	Negative Quittung beim Schreiben zum Modul (Modul während des Schreibvorgangs gezogen oder Modul defekt).
80A2h	DP-Protokollfehler bei Layer 2, evtl. Hardware-/ Schnittstellenfehler im DP-Slave
80A3h	DP-Protokollfehler bei User Interface/User
80A4h	Kommunikation gestört (Fehler tritt auf zwischen CPU und externer DP-Anschaltung)

Fortsetzung ...

... Fortsetzung

Wert	Beschreibung
80B0h	<ul style="list-style-type: none"> • SFC für Modultyp nicht möglich. • Modul kennt den Datensatz nicht. • Datensatznummer ≥ 241 ist unzulässig. • Die Datensätze 0 und 1 sind nicht erlaubt.
80B1h	Die Längenangabe im Parameter <i>RECORD</i> ist falsch.
80B2h	Der projektierte Steckplatz ist nicht belegt.
80B3h	Ist-Modultyp ungleich Soll-Modultyp im SDB1
80C1h	Die Daten des auf dem Modul vorangegangenen Schreibauftrags für denselben Datensatz sind von dem Modul noch nicht verarbeitet.
80C2h	Das Modul bearbeitet momentan das mögliche Maximum an Aufträgen für eine CPU.
80C3h	Benötigte Betriebsmittel (Speicher etc.) sind momentan belegt.
80C4h	Kommunikationsfehler
80C5h	Dezentrale Peripherie nicht verfügbar.
80C6h	Datensatzübertragung wurde abgebrochen wegen Prioritätsklassenabbruchs.

**Hinweis!**

Sollte der allgemeine Fehler 8544h auftreten, wird dadurch nur angezeigt, dass der Zugriff auf mindestens ein Byte des den Datensatz enthaltenden E/A-Speicherbereichs gesperrt war. Die Datenübertragung wurde jedoch fortgesetzt.

SFC 59 - RD_REC - Datensatz lesen

Beschreibung

Mit dem SFC 59 RD_REC (read record) wird der Datensatz mit der Nummer *RECNUM* von dem adressierten Modul gelesen.

Dieser SFC kann für Digital-, Analogmodule, FMs, CPs und über PROFIBUS DP-V1 eingesetzt werden.

Der Lesevorgang wird gestartet, wenn beim Aufruf des SFC 59 der Eingangsparameter *REQ* mit 1 belegt wird. Der Ausgangsparameter *BUSY* liefert den Wert 0, wenn der Lesevorgang sofort ausgeführt werden konnte. Wenn der Lesevorgang noch nicht abgeschlossen ist, liefert *BUSY* den Wert 1. Durch den Parameter *RECORD* wird der Zielbereich festgelegt, in den der gelesene Datensatz nach fehlerfreier Datenübertragung eingetragen wird.

Systembedingt ist dieser Baustein nicht unterbrechbar!

Parameter

Parameter	Deklaration	Datentyp	Speicherbereich	Beschreibung
REQ	INPUT	BOOL	E, A, M, D, L, Konstante	<i>REQ</i> = 1: Anforderung zum Lesen
IOID	INPUT	BYTE	E, A, M, D, L, Konstante	Kennung des Adressbereichs: 54h = Peripherie Eingang (PE) 55h = Peripherie Ausgang (PA) Handelt es sich um ein Mischmodul, ist die Bereichskennung der niedrigeren Adresse anzugeben. Bei gleichen Adressen ist 54h anzugeben.
LADDR	INPUT	WORD	E, A, M, D, L, Konstante	Logische Basisadresse des Moduls. Bei einem Mischmodul ist die kleinere der beiden Adressen anzugeben.
RECNUM	INPUT	BYTE	E, A, M, D, L, Konstante	Datensatznummer (zulässige Werte: 0 ... 240)
RET_VAL	OUTPUT	INT	E, A, M, D, L	Tritt während der Bearbeitung der Funktion ein Fehler auf, enthält der Rückgabewert einen Fehlercode. Zusätzlich: Länge des tatsächlich übertragenen Datensatzes in Bytes (mögliche Werte: +1 ... +240), falls der Zielbereich größer ist als der übertragene Datensatz und bei der Übertragung kein Fehler auftrat.
BUSY	OUTPUT	BOOL	E, A, M, D, L	<i>BUSY</i> = 1: Der Schreibvorgang ist noch nicht beendet.

Fortsetzung ...

... Fortsetzung

Parameter	Deklaration	Datentyp	Speicherbereich	Beschreibung
RECORD	OUTPUT	ANY	E, A, M, D, L	Zielbereich für den gelesenen Datensatz. Sie müssen bei asynchroner Bearbeitung des SFC 59 darauf achten, dass die Aktualparameter von <i>RECORD</i> bei allen Aufrufen dieselbe Längenangabe haben. Es ist nur der Datentyp BYTE zulässig.

Geeignete Wahl von RECORD

Wenn Sie sicherstellen wollen, dass immer der gesamte Datensatz gelesen wird, wählen Sie einen Zielbereich mit der Länge 241 Bytes. Bei einer fehlerfreien Datenübertragung steht dann in *RET_VAL* die tatsächliche Datensatzlänge.

RET_VAL (Rückgabewert)

RET_VAL enthält einen Fehlercode, wenn während der Bearbeitung der Funktion ein Fehler auftrat.

Bei einer fehlerfreien Übertragung enthält *RET_VAL*:

- den Wert 0 wenn der gesamte Zielbereich mit Daten aus dem selektierten Datensatz gefüllt wurde (Der Datensatz kann aber unvollständig sein).
- die Länge des tatsächlich übertragenen Datensatzes in Bytes (mögliche Werte: 1 ... 240), wenn der Zielbereich größer ist als der übertragene Datensatz.

Fehlerinformationen

Bei *RET_VAL* = 8xxxh sind zwei Fälle zu unterscheiden:

- Temporäre Fehler (Fehlercodes 80A2h ... 80A4h, 80Cxh):
Bei dieser Fehlerart besteht die Möglichkeit, dass sich der Fehler ohne Ihr Zutun behebt. Es wäre also sinnvoll, den SFC erneut (ggf. mehrfach) aufzurufen.
Beispiel für temporäre Fehler: Benötigte Betriebsmittel sind momentan belegt (80C3h).
- Permanente Fehler (Fehlercodes 809xh, 80A0h, 80A1h, 80Bxh):
Bei dieser Fehlerart kann der Fehler nicht ohne Ihr Zutun behoben werden. Ein erneuter Aufruf des SFC ist erst wieder sinnvoll, wenn der Fehler beseitigt wurde.
Beispiel für permanente Fehler: Falsche Länge des zu übertragenden Datensatzes (80B1h).

Fehlerinformationen

Wert	Beschreibung
7000h	Erstaufruf mit <i>REQ</i> = 0: keine Datenübertragung aktiv; <i>BUSY</i> hat den Wert 0.
7001h	Erstaufruf mit <i>REQ</i> = 1: Datenübertragung angestoßen; <i>BUSY</i> hat den Wert 1.
7002h	Zwischenaufruf (<i>REQ</i> irrelevant): Datenübertragung bereits aktiv; <i>BUSY</i> hat den Wert 1.
8090h	Angegebene logische Basisadresse ungültig: Es ist keine Zuordnung im SDB1/SDB2x vorhanden, oder es ist keine Basisadresse.
8092h	In ANY-Referenz ist eine Typangabe ungleich BYTE angegeben.
8093h	Für das über <i>LADDR</i> und <i>IOID</i> ausgewählte Modul ist dieser SFC nicht zulässig.
80A0h	Negative Quittung beim Lesen vom Modul (Modul während des Lesevorgangs gezogen oder Modul defekt)
80A2h	DP-Protokollfehler bei Layer 2, evtl. Hardware-/ Schnittstellenfehler im DP-Slave
80A3h	DP-Protokollfehler bei User Interface/User
80A4h	Kommunikation gestört (Fehler tritt auf zwischen CPU und externer DP-Anschaltung)
80B0h	<ul style="list-style-type: none"> • SFC für Modultyp nicht möglich • Modul kennt den Datensatz nicht. • Datensatznummer ≥ 241 ist unzulässig.
80B1h	Die Längenangabe im Parameter <i>RECORD</i> ist falsch.
80B2h	Der projektierte Steckplatz ist nicht belegt.
80B3h	Ist-Modultyp ungleich Soll-Modultyp im SDB1
80C0h	Das Modul führt den Datensatz, aber es sind noch keine Lesedaten da.
80C1h	Die Daten des auf dem Modul vorangegangenen Schreibauftrags für denselben Datensatz sind von dem Modul noch nicht verarbeitet.
80C2h	Das Modul bearbeitet momentan das mögliche Maximum an Aufträgen für eine CPU.
80C3h	Benötigte Betriebsmittel (Speicher etc.) sind momentan belegt.
80C4h	Kommunikationsfehler
80C5h	Dezentrale Peripherie nicht verfügbar.
80C6h	Datensatzübertragung wurde abgebrochen wegen Prioritätsklassenabbruchs.

**Hinweis!**

Sollte der allgemeine Fehler 8745h auftreten, wird dadurch nur angezeigt, dass beim Schreibvorgang auf das Prozessabbild der Zugriff auf mindestens ein Byte nicht möglich war. Der Datensatz wurde jedoch ordnungsgemäß von dem Modul gelesen und in den E/A-Speicherbereich geschrieben.

SFC 64 - TIME_TCK - Systemzeit lesen

Beschreibung Mit dem SFC 64 TIME_TCK (time tick) wird die Systemzeit der CPU ausgelesen. Das kann zum Beispiel dazu genutzt werden, die Dauer von Vorgängen zu messen, indem eine Differenzbildung der Ergebnisse von zwei SFC 64-Aufrufen vorgenommen wird. Bei der Systemzeit handelt es sich um einen "Zeitzähler", der von 0 bis max. 2147483647ms zählt und bei einem Überlauf der Systemzeit wieder von 0 beginnt. Das Zeitraster und die Genauigkeit der Systemzeit hängen von der jeweiligen CPU ab. Die Systemzeit kann nur von den Betriebszuständen der CPU beeinflusst werden.

Systemzeit und Betriebszustände

Betriebszustand	Systemzeit ...
Anlauf RUN	... wird ständig aktualisiert.
STOP	... wird angehalten und behält den aktuellen Wert.
Neustart	... wird gelöscht und beginnt wieder von "0" an zu laufen.

Parameter

Parameter	Deklaration	Datentyp	Speicherbereich	Beschreibung
RET_VAL	OUTPUT	TIME	E, A, M, D, L	Der Parameter <i>RET_VAL</i> enthält die gelesene Systemzeit im Bereich von 0 ... $2^{31} - 1$ ms.

RET_VAL (Rückgabewert) Der SFC 64 gibt keine spezifischen Fehlerinformationen aus.

SFC 65 - X_SEND - Daten senden

Beschreibung Mit dem SFC 65 X_SEND können Daten an einen außerhalb der eigenen Station liegenden Kommunikationspartner gesendet werden. Der Kommunikationspartner empfängt die Daten mit Hilfe des SFC 66 X_RCV. Die Kennzeichnung der Sendedaten erfolgt über den Eingangsparameter *REQ_ID*. Diese Auftragskennung wird mit übertragen und kann beim Kommunikationspartner ausgewertet werden, um die Herkunft der Daten zu ermitteln. Der Auftrag wird gestartet, wenn am Eingangsparameter *REQ* der Wert 1 anliegt. Der über den Parameter *SD* definierte Sendebereich (auf der sendenden CPU) muss kleiner oder gleich dem über den Parameter *RD* definierten Empfangsbereich (beim Kommunikationspartner) sein, des Weiteren müssen die Datentypen von Sendebereich und Empfangsbereich übereinstimmen.

Parameter

Parameter	Deklaration	Datentyp	Speicherbereich	Beschreibung
REQ	INPUT	BOOL	E, A, M, D, L, Konstante	Steuerparameter "request to activate", dient dazu, den Auftrag anzustoßen
CONT	INPUT	BOOL	E, A, M, D, L, Konstante	Steuerparameter "continue", legt fest, ob die Verbindung zum Kommunikationspartner nach Abschluss des Auftrags bestehen bleibt
DEST_ID	INPUT	WORD	E, A, M, D, L, Konstante	Adressierungsparameter "destination ID". Er enthält die MPI-Adresse des Kommunikationspartners.
REQ_ID	INPUT	DWORD	E, A, M, D, L, Konstante	Auftragskennung. Sie dient zur Identifizierung der Daten beim Kommunikationspartner.
SD	INPUT	ANY	E, A, M, D	Referenz auf den Sendebereich. Folgende Datentypen sind erlaubt: BOOL, BYTE, CHAR, WORD, INT, DWORD, DINT, REAL, DATE, TOD, TIME, S5_TIME, DATE_AND_TIME sowie Arrays der genannten Datentypen mit Ausnahme von BOOL.
RET_VAL	OUTPUT	INT	E, A, M, D, L	Tritt während der Bearbeitung der Funktion ein Fehler auf, enthält der Rückgabewert einen Fehlercode.
BUSY	OUTPUT	BOOL	E, A, M, D, L	<i>BUSY</i> = 1: Der Sendevorgang ist noch nicht abgeschlossen. <i>BUSY</i> = 0: Der Sendevorgang ist abgeschlossen, bzw. es ist kein Sendevorgang aktiv.

REQ_ID Der Eingangsparameter *REQ_ID* dient zur Kennzeichnung der Sendedaten.

Auf Empfängerseite wird der Parameter *REQ_ID* benötigt, wenn

- auf einer Sende-CPU mehrere SFC 65 mit unterschiedlichen Parametern *REQ_ID* aufgerufen und die Daten an einen Kommunikationspartner übertragen werden.
- von mehreren Sende-CPU's Daten mit Hilfe des SFC 65 an einen Kommunikationspartner übertragen werden.

Die Empfangsdaten können mit Hilfe der Auswertung des Parameters *REQ_ID* in unterschiedlichen Speicherbereichen abgelegt werden.

Datenkonsistenz

Da die Sendedaten beim ersten Aufruf des SFC in einen internen Puffer des Betriebssystems umkopiert werden, ist darauf zu achten, dass der Sendebereich nicht beschrieben wird, bevor der Erstaufruf abgeschlossen ist, da ansonsten inkonsistente Daten übertragen werden können.

Bei einem schreibenden Zugriff auf die Sendedaten nach dem Erstaufruf wird die Datenkonsistenz nicht beeinträchtigt.

RET_VAL (Rückgabewert) Tritt während der Bearbeitung der Funktion ein Fehler auf, enthält der Rückgabewert einen Fehlercode.

Die in der Tabelle "Spezifische Fehlerinformationen" u.a. angegebenen "echten" Fehlerinformationen können wie folgt klassifiziert werden:

Wert	Beschreibung
809xh	Fehler auf der CPU, auf der der SFC abläuft
80Axh	Permanenter Kommunikationsfehler
80Bxh	Fehler beim Kommunikationspartner
80Cxh	Temporärer Fehler

Spezifische Fehlerinformationen:

Wert	Beschreibung
0000h	Die Bearbeitung wurde fehlerfrei abgeschlossen.
7000h	Erstaufruf mit <i>REQ</i> = 0: keine Datenübertragung aktiv; <i>BUSY</i> hat Wert 0.
7001h	Erstaufruf mit <i>REQ</i> = 1: Datenübertragung angestoßen; <i>BUSY</i> hat den Wert 1.
7002h	Zwischenaufruf (<i>REQ</i> irrelevant): Datenübertragung bereits aktiv; <i>BUSY</i> hat den Wert 1.
8090h	Angegebene Zieladresse des Kommunikationspartners ist ungültig, z.B. <ul style="list-style-type: none"> • falsche <i>IOID</i> • falsche Basisadresse vorhanden • falsche MPI-Adresse (> 126)
8092h	Fehler bei <i>SD</i> oder <i>RD</i> , z.B.: <ul style="list-style-type: none"> • unzulässige Länge von <i>SD</i> • <i>SD</i> = NIL ist unzulässig
8095h	Baustein wird bereits in einer Prioritätsklasse mit niedrigerer Priorität bearbeitet.
80A0h	Fehler in empfangener Quittung
80A1h	Kommunikationsprobleme: SFC-Aufruf nach Abbruch einer bestehenden Verbindung.
80B1h	Fehler im ANY-Pointer. Die Länge des zu übertragenden Datenbereichs ist falsch.
80B4h	Datentypfehler bei ANY-Pointer, oder ARRAY des angegebenen Datentyps nicht erlaubt.
80B5h	Bearbeitung wegen unzulässigem Betriebszustand abgelehnt.
80B6h	In der empfangenen Quittung steht ein unbekannter Fehlercode.
80B8h	Der SFC 66 "X_RCV" des Kommunikationspartners hat die Datenübernahme abgelehnt (<i>RD</i> = NIL).
80B9h	Der Datenblock wurde vom Kommunikationspartner identifiziert (Aufruf des SFC 66 "X_RCV" mit <i>EN_D T</i> = 0), jedoch wegen Betriebszustand STOP noch nicht ins Anwenderprogramm übernommen.
80BAh	Antwort des Kommunikationspartners passt nicht ins Kommunikationstelegramm.
80C0h	Die angegebene Verbindung ist durch einen anderen Auftrag bereits belegt.
80C1h	Ressourcenengpass bei der CPU, auf der der SFC abläuft, z.B.: <ul style="list-style-type: none"> • Die maximale Anzahl verschiedener Sendeaufträge des Moduls wird bereits bearbeitet. • Die Verbindungsressource ist z.B. durch Empfang belegt.
80C2h	Temporärer Ressourcenmangel des Kommunikationspartners, z.B.: <ul style="list-style-type: none"> • Der Kommunikationspartner bearbeitet momentan das Maximum an Aufträgen. • Die benötigten Betriebsmittel (Speicher etc.) sind belegt. • Zu wenig Arbeitsspeicher vorhanden. (Stoßen Sie Komprimiervorgang an.)
80C3h	Fehler beim Verbindungsaufbau, z.B.: <ul style="list-style-type: none"> • Die eigene Station befindet sich nicht am MPI-Subnetz. • Sie haben die eigene Station am MPI-Subnetz adressiert. • Der Kommunikationspartner ist nicht mehr erreichbar. • Temporärer Ressourcenmangel des Kommunikationspartners.

SFC 66 - X_RCV - Daten empfangen

Beschreibung Mit dem SFC 66 X_RCV können Daten empfangen werden, die ein oder mehrere außerhalb der eigenen Station liegende Kommunikationspartner mit Hilfe des SFC 65 X_SEND gesendet haben.

Mit dem SFC 66 kann festgestellt werden, ob zum aktuellen Zeitpunkt gesendete Daten bereitstehen. Diese Daten wurden vom Betriebssystem gegebenenfalls in eine interne Warteschlange gestellt. Sind in dieser Warteschlange Daten vorhanden, so kann der älteste Datenblock in der Warteschlange in einen vorgegebenen Empfangsbereich kopiert werden.

Parameter

Parameter	Deklaration	Datentyp	Speicherbereich	Beschreibung
EN_DT	INPUT	BOOL	E, A, M, D, L, Konstante	Steuerparameter "enable data transfer". Mit dem Wert 0 überprüfen Sie, ob mindestens ein Datenblock bereitsteht. Der Wert 1 bewirkt das Umkopieren des ältesten in der Warteschlange vorhandenen Datenblocks in den Bereich des Arbeitsspeichers, den Sie durch <i>RD</i> vorgegeben haben.
RET_VAL	OUTPUT	INT	E, A, M, D, L	Tritt während der Bearbeitung der Funktion ein Fehler auf, enthält der Rückgabewert einen Fehlercode.
REQ_ID	OUTPUT	DWORD	E, A, M, D, L	Auftragskennung desjenigen SFC 65 "X_SEND", dessen gesendete Daten in der Warteschlange an erster Stelle stehen, d.h. die ältesten Daten in der Warteschlange. Falls kein Datenblock in der Warteschlange steht, enthält <i>REQ_ID</i> den Wert 0.
NDA	OUTPUT	BOOL	E, A, M, D, L	Zustandsparameter "new data arrived". <i>NDA</i> = 0: <ul style="list-style-type: none"> In der Warteschlange ist kein Datenblock vorhanden. <i>NDA</i> = 1: <ul style="list-style-type: none"> In der Warteschlange ist mindestens ein Datenblock vorhanden. (Aufruf des SFC 66 mit <i>EN_DT</i> = 0). Der älteste Datenblock in der Warteschlange wurde ins Anwenderprogramm kopiert. (Aufruf des SFC 66 mit <i>EN_DT</i> = 1).

Fortsetzung ...

... Fortsetzung

Parameter	Deklaration	Datentyp	Speicherbereich	Beschreibung
RD	OUTPUT	ANY	E, A, M, D	Referenz auf den Empfangsbereich (receive data area). Folgende Datentypen sind erlaubt: BOOL, BYTE, CHAR, WORD, INT, DWORD, DINT, REAL, DATE, TOD, TIME, S5_TIME, DATE_AND_TIME sowie Arrays der genannten Datentypen mit Ausnahme von BOOL. Falls Sie den ältesten in der Warteschlange vorhandenen Datenblock verwerfen wollen, belegen Sie <i>RD</i> mit dem Wert NIL.

Datenempfang anzeigen

mit *EN_DT = 0*

Die empfangenen Daten eines Kommunikationspartners werden vom Betriebssystem in der Reihenfolge ihres Empfangs in die Warteschlange eingereiht.

Zur Überprüfung, ob mindestens ein Datenblock bereitsteht, wird der SFC 66 mit *EN_DT = 0* aufgerufen und der Ausgangsparameter *NDA* ausgewertet.

- *NDA = 0* bedeutet, in der Warteschlange ist kein Datenblock enthalten. *REQ_ID* ist irrelevant, *RET_VAL* enthält den Wert 7000h.
- *NDA = 1* bedeutet, in der Warteschlange ist mindestens ein Datenblock enthalten.

Ist ein Datenblock in der Warteschlange enthalten, sollten zusätzlich die Ausgangsparameter *RET_VAL* und *REQ_ID* ausgewertet werden. *RET_VAL* enthält die Länge des Datenblocks in Byte, *REQ_ID* die Auftragskennung des Sendebausteins. Wenn sich in der Warteschlange mehrere Datenblöcke befinden, so gehören *REQ_ID* und *RET_VAL* zum ältesten in der Warteschlange liegenden Datenblock.

Daten in Empfangsbereich übernehmen	mit $EN_DT = 1$ Wenn der Eingangsparameter $EN_DT = 1$ ist, wird der älteste in der Warteschlange vorhandene Datenblock in den durch RD vorgegebenen Bereich kopiert. Es ist darauf zu achten, das RD größer oder gleich dem über den Parameter SD definierten Sendebereich des zugehörigen SFC 65 X_SEND ist und das die Datentypen übereinstimmen. Wenn die empfangenen Daten in unterschiedlichen Bereichen abgelegt werden sollen, kann im Erstaufwurf REQ_ID ermittelt (SFC-Aufruf mit $EN_DT = 0$) und im Folgeaufwurf (mit $EN_DT = 1$) RD geeignet gewählt werden. Wurde der Auftrag fehlerfrei bearbeitet, enthält RET_VAL die Länge des kopierten Datenblocks in Byte und der Sender erhält eine positive Quittung.
Daten verwerfen	Wenn die empfangenen Daten nicht übernommen werden sollen, belegen Sie RD mit dem Wert NIL. Der zugehörige Kommunikationspartner erhält dann eine negative Quittung (RET_VAL des zugehörigen SFC 65 X_SEND hat den Wert 80B8h) und der Parameter RET_VAL hat den Wert 0.
Datenkonsistenz	Es ist darauf zu achten, das der Empfangsbereich nicht ausgelesen wird, bevor der Auftrag nicht abgeschlossen ist, da ansonsten inkonsistente Daten ausgelesen werden können.
Betriebszustandsübergang nach STOP	Wenn die CPU in den Betriebszustand STOP übergeht, <ul style="list-style-type: none">• werden alle neu ankommenden Aufträge negativ quittiert.• gilt für bereits angekommene Aufträge: Alle in der Empfangswarteschlange eingetragenen Aufträge werden negativ quittiert.• werden bei einem anschließenden Neustart alle Datenblöcke verworfen.
Verbindungsabbruch	Sollte es zu einem Abbruch der Verbindung kommen, wird ein in der Empfangswarteschlange bereits eingetragener Auftrag, der zu dieser Verbindung gehört, verworfen. Ausnahme: Falls es sich bei diesem Auftrag um den ältesten Auftrag in der Warteschlange handelt und er durch einen SFC-Aufruf mit $EN_DT = 0$ bereits erkannt wurde, kann er mit $EN_DT = 1$ in den Empfangsbereich übernommen werden.

RET_VAL
(Rückgabewert)

Tritt kein Fehler auf, enthält *RET_VAL*:

- bei *EN_DT* = 0/1 und *NDA* = 0: 7000h. In diesem Fall steht kein Datenblock in der Warteschlange.
- bei *EN_DT* = 0 und *NDA* = 1 die Länge des ältesten in der Warteschlange eingetragenen Datenblocks in Byte als positive Zahl.
- bei *EN_DT* = 1 und *NDA* = 1 die Länge des in den Empfangsbereich *RD* kopierten Datenblocks in Byte als positive Zahl.

Fehlerinformationen

Die in der Tabelle "Spezifische Fehlerinformationen" u.a. angegebenen "echten" Fehlerinformationen können wie folgt klassifiziert werden:

Wert	Beschreibung
809xh	Fehler auf der CPU, auf der der SFC abläuft
80Axh	Permanenter Kommunikationsfehler
80Bxh	Fehler beim Kommunikationspartner
80Cxh	Temporärer Fehler

Spezifische Fehlerinformationen:

Wert	Beschreibung
0000h	Die Bearbeitung wurde fehlerfrei abgeschlossen.
00xyh	Bei $NDA = 1$ und $RD \neq NIL$: RET_VAL enthält die Länge des empfangenen (bei $EN_DT = 0$) bzw. des in RD kopierten Datenblocks (bei $EN_DT = 1$).
7000h	$EN_DT = 0/1$ und $NDA = 0$
7001h	Erstaufruf mit $REQ = 1$: Datenübertragung angestoßen; $BUSY$ hat den Wert 1.
7002h	Zwischenaufruf (REQ irrelevant): Datenübertragung bereits aktiv; $BUSY =$ Wert 1.
8090h	Angegebene Zieladresse des Kommunikationspartners ist ungültig, z.B. <ul style="list-style-type: none"> • falsche $IOID$ • falsche Basisadresse vorhanden • falsche MPI-Adresse (> 126)
8092h	Fehler bei SD oder RD , z.B.: <ul style="list-style-type: none"> • Es wurden mehr Daten empfangen als in den durch RD vorgegebenen Bereich hineinpassen. • RD ist vom Datentyp $BOOL$, die Länge der empfangenen Daten ist jedoch größer als ein Byte.
8095h	Der Baustein wird bereits in einer Prioritätsklasse mit niedrigerer Priorität bearbeitet.
80A0h	Fehler in empfangener Quittung.
80A1h	Kommunikationsprobleme: SFC-Aufruf nach Abbruch einer bestehenden Verbindung.
80B1h	Fehler im ANY-Pointer. Die Länge des zu übertragenden Datenbereichs falsch.
80B4h	Datentypfehler bei ANY-Pointer, oder ARRAY des angegebenen Datentyps nicht erlaubt.
80B6h	In der empfangenen Quittung steht ein unbekannter Fehlercode.
80BAh	Die Antwort des Kommunikationspartners passt nicht in das Kommunikationstelegramm.
80C0h	Die angegebene Verbindung ist durch einen anderen Auftrag bereits belegt.
80C1h	Ressourcenengpass bei der CPU, auf der der SFC abläuft, z.B.: <ul style="list-style-type: none"> • Die maximale Anzahl verschiedener Sendeaufträge des Moduls wird bereits bearbeitet. • Die Verbindungsressource ist z.B. durch Empfang belegt.
80C2h	Temporärer Ressourcenmangel des Kommunikationspartners, z.B.: <ul style="list-style-type: none"> • Der Kommunikationspartner bearbeitet momentan das Maximum an Aufträgen. • Die benötigten Betriebsmittel (Speicher etc.) sind belegt. • Zu wenig Arbeitsspeicher vorhanden. (Stoßen Sie Komprimiervorgang an.)
80C3h	Fehler beim Verbindungsaufbau, z.B.: <ul style="list-style-type: none"> • Die eigene Station befindet sich nicht am MPI-Subnetz. • Sie haben die eigene Station am MPI-Subnetz adressiert. • Der Kommunikationspartner ist nicht mehr erreichbar. • Temporärer Ressourcenmangel des Kommunikationspartners.

SFC 67 - X_GET - Daten lesen

Beschreibung

Mit dem SFC 67 X_GET können Daten aus einem Kommunikationspartner, der außerhalb der eigenen Station liegt, gelesen werden, wobei auf dem Kommunikationspartner kein zugehöriger SFC existiert. Der Auftrag wird gestartet, wenn am Eingangsparameter *REQ* der Wert 1 anliegt. Danach wird der SFC 67 so lange aufgerufen, bis am Ausgangsparameter *BUSY* der Wert 0 anliegt. Der Ausgangsparameter *RET_VAL* enthält die Länge des empfangenen Datenblocks in Byte.

Der über den Parameter *RD* definierte Empfangsbereich (auf der empfangenden CPU) muss mindestens so lang wie der über den Parameter *VAR_ADDR* definierte Lesebereich (beim Kommunikationspartner) sein, des Weiteren müssen die Datentypen von *RD* und *VAR_ADDR* übereinstimmen.

Parameter

Parameter	Deklaration	Datentyp	Speicherbereich	Beschreibung
REQ	INPUT	BOOL	E, A, M, D, L, Konstante	Steuerparameter "request to activate", dient dazu, den Auftrag anzustoßen.
CONT	INPUT	BOOL	E, A, M, D, L, Konstante	Steuerparameter "continue", legt fest, ob die Verbindung zum Kommunikationspartner nach Abschluss des Auftrags bestehen bleibt.
DEST_ID	INPUT	WORD	E, A, M, D, L, Konstante	Adressierungsparameter "destination ID". Er enthält die MPI-Adresse des Kommunikationspartners.
VAR_ADDR	INPUT	ANY	E, A, M, D	Referenz auf den Bereich in der Partner-CPU, der gelesen werden soll. Sie müssen einen Datentyp wählen, der vom Kommunikationspartner unterstützt wird.
RET_VAL	OUTPUT	INT	E, A, M, D, L	Tritt während der Bearbeitung der Funktion ein Fehler auf, enthält der Rückgabewert einen Fehlercode. Tritt kein Fehler auf, enthält <i>RET_VAL</i> die Länge des in den Empfangsbereich <i>RD</i> kopierten Datenblocks in Byte als positive Zahl.
BUSY	OUTPUT	BOOL	E, A, M, D, L	<i>BUSY</i> = 1: Der Empfangsvorgang ist noch nicht abgeschlossen. <i>BUSY</i> = 0: Der Empfangsvorgang ist abgeschlossen, bzw. es ist kein Empfangsvorgang aktiv.
RD	OUTPUT	ANY	E, A, M, D	Referenz auf den Empfangsbereich (receive data area). Folgende Datentypen sind erlaubt: BOOL, BYTE, CHAR, WORD, INT, DWORD, DINT, REAL, DATE, TOD, TIME, S5_TIME, DATE_AND_TIME sowie Arrays der genannten Datentypen mit Ausnahme von BOOL

- Datenkonsistenz** Um sicherzustellen, dass die Datenkonsistenz nicht beeinträchtigt wird sollten folgende Konsistenzregeln beachtet werden:
- **Aktive CPU (Datenempfänger):**
Der Empfangsbereich sollte in dem OB ausgelesen werden, in dem der entsprechende SFC aufgerufen wird. Sollte dies nicht möglich sein, so sollte der Empfangsbereich erst dann ausgelesen werden, wenn die Bearbeitung des zugehörigen SFC abgeschlossen ist.
 - **Passive CPU (Datensender):**
Es dürfen höchstens so viele Daten in den Sendebereich geschrieben werden, wie die Blockgröße der passiven CPU (Datensender) angibt.
 - **Passive CPU (Datensender):**
Die zu sendenden Daten sollten unter Interrupt-Sperre in den Sendebereich geschrieben werden.
- Betriebszustandsübergang nach STOP** Wenn die CPU in den Betriebszustand STOP übergeht, wird die vom SFC 67 aufgebaute Verbindung abgebaut. Von der Art des anschließenden Anlaufs hängt ab, ob die bereits empfangenen Daten, die in einem Zwischenspeicher des Betriebssystems stehen, verloren gehen. Bei einem anschließenden Neustart werden die Daten verworfen.
- Betriebszustandsübergang des Partners nach STOP** Da die Daten auch im Betriebszustand STOP gelesen werden können, hat ein Übergang der CPU des Kommunikationspartners in den Betriebszustand STOP keine Auswirkung auf die Datenübertragung.

RET_VAL (Rückgabewert) Die in der Tabelle "Spezifische Fehlerinformationen" u.a. angegebenen "echten Fehlerinformationen" können wie folgt klassifiziert werden:

Wert	Beschreibung
809xh	Fehler auf der CPU, auf der der SFC abläuft
80Axh	Permanenter Kommunikationsfehler
80Bxh	Fehler beim Kommunikationspartner
80Cxh	Temporärer Fehler

Spezifische Fehlerinformationen:

Wert	Beschreibung
0000h	Die Bearbeitung wurde fehlerfrei abgeschlossen.
00xyh	<i>RET_VAL</i> enthält die Länge des empfangenen Datenblocks.
7000h	Aufruf mit <i>REQ</i> = 0 (Aufruf ohne Bearbeitung), <i>BUSY</i> hat den Wert 0, es ist keine Datenübertragung aktiv.
7001h	Erstaufruf mit <i>REQ</i> = 1: Datenübertragung angestoßen; <i>BUSY</i> hat den Wert 1.
7002h	Zwischenaufruf (<i>REQ</i> irrelevant): Datenübertragung bereits aktiv; <i>BUSY</i> hat den Wert 1.
8090h	Angegebene Zieladresse des Kommunikationspartners ist ungültig, z.B. <ul style="list-style-type: none"> • falsche <i>IOID</i> • falsche Basisadresse vorhanden • falsche MPI-Adresse (> 126)
8092h	Fehler bei <i>SD</i> oder <i>RD</i> , z.B.: <ul style="list-style-type: none"> • unzulässige Länge bei <i>RD</i> • Die Länge oder der Datentyp von <i>RD</i> stimmt mit den empfangenen Daten nicht überein. • <i>RD</i> = NIL ist unzulässig
8095h	Der Baustein wird bereits in einer Prioritätsklasse mit niedrigerer Priorität bearbeitet.
80A0h	Fehler in empfangener Quittung
80A1h	Kommunikationsprobleme: SFC-Aufruf nach Abbruch einer bestehenden Verbindung.
80B0h	Objekt ist nicht erreichbar, z.B. DB nicht geladen.
80B1h	Fehler im ANY-Pointer. Die Länge des zu übertragenden Datenbereichs ist falsch.

Fortsetzung ...

... Fortsetzung

Wert	Beschreibung
80B2h	HW-Fehler: Modul nicht vorhanden <ul style="list-style-type: none"> • Der projektierte Steckplatz ist nicht belegt. • Ist-Modultyp ungleich Soll-Modultyp • Dezentrale Peripherie ist nicht verfügbar. • Im zugehörigen SDB ist kein Eintrag für das Modul vorhanden.
80B3h	Daten dürfen entweder nur gelesen oder nur geschrieben werden, z.B. schreibgeschützter DB
80B4h	Der in <i>VAR_ADDR</i> angegebene Datentyp wird vom Kommunikationspartner nicht unterstützt.
80B6h	In der empfangenen Quittung steht ein unbekannter Fehlercode.
80BAh	Die Antwort des Kommunikationspartners passt nicht in das Kommunikationstelegramm.
80C0h	Die angegebene Verbindung ist durch einen anderen Auftrag bereits belegt.
80C1h	Ressourcenengpass bei der CPU, auf der der SFC abläuft, z.B.: <ul style="list-style-type: none"> • Die maximale Anzahl verschiedener Sendeaufträge des Moduls wird bereits bearbeitet. • Die Verbindungsressource ist z.B. durch Empfang belegt.
80C2h	Temporärer Ressourcenmangel des Kommunikationspartners, z.B.: <ul style="list-style-type: none"> • Der Kommunikationspartner bearbeitet momentan das Maximum an Aufträgen. • Die benötigten Betriebsmittel (Speicher etc.) sind belegt. • Zu wenig Arbeitsspeicher vorhanden. (Stoßen Sie einen Komprimierungsvorgang an.)
80C3h	Fehler beim Verbindungsaufbau, z.B.: <ul style="list-style-type: none"> • Die eigene Station befindet sich nicht am MPI-Subnetz. • Sie haben die eigene Station am MPI-Subnetz adressiert. • Der Kommunikationspartner ist nicht mehr erreichbar. • Temporärer Ressourcenmangel des Kommunikationspartners.

SFC 68 - X_PUT - Daten schreiben

Beschreibung

Mit dem SFC 68 X_PUT können Daten in einen Kommunikationspartner, der außerhalb der eigenen Station liegt, geschrieben werden, wobei auf dem Kommunikationspartner kein zugehöriger SFC existiert. Der Auftrag wird gestartet, wenn am Eingangsparameter *REQ* der Wert 1 anliegt. Danach wird der SFC 68 so lange aufgerufen, bis am Ausgangsparameter *BUSY* der Wert 0 anliegt. Der über den Parameter *SD* definierte Sendebereich (auf der sendenden CPU) muss mindestens so lang wie der über den Parameter *VAR_ADDR* definierte Empfangsbereich (beim Kommunikationspartner) sein, des Weiteren müssen die Datentypen von *SD* und *VAR_ADDR* übereinstimmen.

Parameter

Parameter	Deklaration	Datentyp	Speicherbereich	Beschreibung
REQ	INPUT	BOOL	E, A, M, D, L, Konstante	Steuerparameter "request to activate", dient dazu, den Auftrag anzustoßen.
CONT	INPUT	BOOL	E, A, M, D, L, Konstante	Steuerparameter "continue", legt fest, ob die Verbindung zum Kommunikationspartner nach Abschluss des Auftrags bestehen bleibt
DEST_ID	INPUT	WORD	E, A, M, D, L, Konstante	Adressierungsparameter "destination ID". Er enthält die MPI-Adresse des Kommunikationspartners.
VAR_ADDR	INPUT	ANY	E, A, M, D	Referenz auf den Bereich in der Partner-CPU, in den geschrieben werden soll. Sie müssen einen Datentyp wählen, der vom Kommunikationspartner unterstützt wird.
SD	INPUT	ANY	E, A, M, D	Referenz auf den Bereich in der eigenen CPU, der die zu versendenden Daten enthält. Folgende Datentypen sind erlaubt: BOOL, BYTE, CHAR, WORD, INT, DWORD, DINT, REAL, DATE, TOD, TIME, S5_TIME, DATE_AND_TIME sowie Arrays der genannten Datentypen mit Ausnahme von BOOL.
RET_VAL	OUTPUT	INT	E, A, M, D, L	Tritt während der Bearbeitung der Funktion ein Fehler auf, enthält der Rückgabewert einen Fehlercode.
BUSY	OUTPUT	BOOL	E, A, M, D, L	<i>BUSY</i> = 1: Der Sendevorgang ist noch nicht abgeschlossen. <i>BUSY</i> = 0: Der Sendevorgang ist abgeschlossen, bzw. es ist kein Sendevorgang aktiv.

- Datenkonsistenz** Um sicherzustellen, dass die Datenkonsistenz nicht beeinträchtigt wird, sollten folgende Konsistenzregeln beachtet werden:
- **Aktive CPU (Datensender):**
Der Sendebereich sollte von dem OB aus beschrieben werden, in dem der entsprechende SFC aufgerufen wird. Sollte dies nicht möglich sein, so sollte der Sendebereich erst dann beschrieben werden, wenn der erste Aufruf des zugehörigen SFC abgeschlossen ist.
 - **Aktive CPU (Datensender):**
Es dürfen höchstens so viele Daten in den Sendebereich geschrieben werden, wie die Blockgröße der passiven CPU (Datenempfänger) angibt.
 - **Passive CPU (Datenempfänger):**
Die zu empfangenden Daten sollten unter Interrupt-Sperre aus dem Empfangsbereich ausgelesen werden.
- Betriebszustandsübergang nach STOP** Wenn die CPU in den Betriebszustand STOP übergeht, wird die vom SFC 68 aufgebaute Verbindung abgebaut und die Daten können nicht mehr gesendet werden. Wenn die Sendedaten zum Zeitpunkt des Betriebszustandsübergangs nach STOP bereits in den internen Puffer kopiert waren, wird der Pufferinhalt verworfen.
- Betriebszustandsübergang des Partners nach STOP** Da die Daten auch im Betriebszustand STOP geschrieben werden können, hat ein Übergang der CPU des Kommunikationspartners in den Betriebszustand STOP keine Auswirkung auf die Datenübertragung.
- RET_VAL (Rückgabewert)** Die in der Tabelle "Spezifische Fehlerinformationen" u.a. angegebenen "echten Fehlerinformationen" können wie folgt klassifiziert werden:

Wert	Beschreibung
809xh	Fehler auf der CPU, auf der der SFC abläuft
80Axh	Permanenter Kommunikationsfehler
80Bxh	Fehler beim Kommunikationspartner
80Cxh	Temporärer Fehler

Spezifische Fehlerinformationen:

Wert	Beschreibung
0000h	Die Bearbeitung wurde fehlerfrei abgeschlossen.
7000h	Aufruf mit <i>REQ</i> = 0 (Aufruf ohne Bearbeitung), <i>BUSY</i> hat den Wert 0, es ist keine Datenübertragung aktiv.
7001h	Erstaufwurf mit <i>REQ</i> = 1: Datenübertragung angestoßen; <i>BUSY</i> hat den Wert 1.
7002h	Zwischenaufwurf (<i>REQ</i> irrelevant): Datenübertragung bereits aktiv; <i>BUSY</i> hat den Wert 1.
8090h	Angegebene Zieladresse des Kommunikationspartners ist ungültig, z.B. <ul style="list-style-type: none"> • falsche <i>IOID</i> • falsche Basisadresse vorhanden • falsche MPI-Adresse (> 126)
8092h	Fehler bei <i>SD</i> oder <i>RD</i> , z.B.: <ul style="list-style-type: none"> • unzulässige Länge bei <i>SD</i> • <i>SD</i> = NIL ist unzulässig
8095h	Der Baustein wird bereits in einer Prioritätsklasse mit niedrigerer Priorität bearbeitet.
80A0h	Der in <i>SD</i> der sendenden CPU angegebene Datentyp wird vom Kommunikationspartner nicht unterstützt.
80A1h	Kommunikationsprobleme: SFC-Aufruf nach Abbruch einer bestehenden Verbindung.
80B0h	Objekt ist nicht erreichbar, z.B. DB nicht geladen.
80B1h	Fehler im ANY-Pointer. Die Länge des zu übertragenden Datenbereichs ist falsch.
80B2h	HW-Fehler: Modul nicht vorhanden <ul style="list-style-type: none"> • Der projektierte Steckplatz ist nicht belegt. • Ist-Modultyp ungleich Soll-Modultyp • Dezentrale Peripherie ist nicht verfügbar. • Im zugehörigen SDB ist kein Eintrag für das Modul vorhanden.
80B3h	Daten dürfen entweder nur gelesen oder nur geschrieben werden, z.B. schreibgeschützter DB
80B4h	Der in <i>VAR_ADDR</i> angegebene Datentyp wird vom Kommunikationspartner nicht unterstützt.
80B6h	In der empfangenen Quittung steht ein unbekannter Fehlercode.
80B7h	Datentyp und/oder Länge der übertragenen Daten passen nicht zum Bereich in der Partner-CPU, in den geschrieben werden soll.
80BAh	Die Antwort des Kommunikationspartners passt nicht in das Kommunikationstelegramm.
80C0h	Die angegebene Verbindung ist durch einen anderen Auftrag bereits belegt.
80C1h	Ressourcenengpass bei der CPU, auf der der SFC abläuft, z.B.: <ul style="list-style-type: none"> • Die maximale Anzahl verschiedener Sendeaufträge des Moduls wird bereits bearbeitet. • Die Verbindungsressource ist z.B. durch Empfang belegt.
80C2h	Temporärer Ressourcenmangel des Kommunikationspartners, z.B.: <ul style="list-style-type: none"> • Der Kommunikationspartner bearbeitet momentan das Maximum an Aufträgen. • Die benötigten Betriebsmittel (Speicher etc.) sind belegt. • Zu wenig Arbeitsspeicher vorhanden (Stoßen Sie einen Komprimierungsvorgang an).
80C3h	Fehler beim Verbindungsaufbau, z.B.: <ul style="list-style-type: none"> • Die eigene Station befindet sich nicht am MPI-Subnetz. • Sie haben die eigene Station am MPI-Subnetz adressiert. • Der Kommunikationspartner ist nicht mehr erreichbar. • Temporärer Ressourcenmangel des Kommunikationspartners.

SFC 69 - X_ABORT - Verbindung abbrechen

Beschreibung

Mit dem SFC 69 X_ABORT kann die Verbindung zu einem Kommunikationspartner, der außerhalb der eigenen Station liegt, abgebrochen werden, wenn die Verbindung mit einem der SFCs 65, 67 oder 68 aufgebaut wurde. Der Auftrag wird gestartet, wenn am Eingangsparameter *REQ* der Wert 1 anliegt.

Wenn der zu den SFCs 65, 67 oder 68 gehörende Auftrag bereits abgeschlossen wurde (*BUSY* = 0), dann sind nach dem Aufruf des SFC 69 die dafür belegten Verbindungsressourcen auf beiden Seiten wieder freigegeben. Ist der zugehörige Auftrag jedoch noch nicht abgeschlossen (*BUSY* = 1), muss nach Abschluss des Verbindungsabbruchs der zugehörige SFC 65, 67 oder 68 erneut mit *REQ* = 0 und *CONT* = 0 aufgerufen und *BUSY* = 0 abgewartet werden, denn erst dann sind alle belegten Verbindungsressourcen wieder freigegeben. Der SFC 69 kann nur auf derjenigen Seite aufgerufen werden, auf der der SFC 65, 67 oder 68 abläuft.

Parameter

Parameter	Deklaration	Datentyp	Speicherbereich	Beschreibung
REQ	INPUT	BOOL	E, A, M, D, L, Konstante	Steuerparameter "request to activate", dient dazu, den Auftrag anzustoßen
DEST_ID	INPUT	WORD	E, A, M, D, L, Konstante	Adressierungsparameter "destination ID". Er enthält die MPI-Adresse des Kommunikationspartners.
RET_VAL	OUTPUT	INT	E, A, M, D, L	Tritt während der Bearbeitung der Funktion ein Fehler auf, enthält der Rückgabewert einen Fehlercode.
BUSY	OUTPUT	BOOL	E, A, M, D, L	<i>BUSY</i> = 1: Der Verbindungsabbruch ist noch nicht abgeschlossen. <i>BUSY</i> = 0: Der Verbindungsabbruch ist abgeschlossen.

Betriebszustandsübergang nach STOP Wenn die CPU in den Betriebszustand STOP übergeht, wird der vom SFC 69 angestoßene Verbindungsabbruch dennoch zu Ende bearbeitet.

Betriebszustandsübergang des Partners nach STOP Ein Übergang der CPU des Kommunikationspartners in den Betriebszustand STOP hat keine Auswirkung auf den Verbindungsabbruch, die Verbindung wird ungeachtet des Betriebszustandsübergangs abgebrochen.

RET_VAL (Rückgabewert) Die in der Tabelle "Spezifische Fehlerinformationen" u.a. angegebenen "echten Fehlerinformationen" können wie folgt klassifiziert werden:

Wert	Beschreibung
809xh	Fehler auf der CPU, auf der der SFC abläuft
80Axh	Permanenter Kommunikationsfehler
80Bxh	Fehler beim Kommunikationspartner
80Cxh	Temporärer Fehler

Spezifische Fehlerinformationen:

Wert	Beschreibung
0000h	<i>REQ</i> = 1, und die angegebene Verbindung ist nicht aufgebaut.
7000h	Aufruf mit <i>REQ</i> = 0 (Aufruf ohne Bearbeitung), <i>BUSY</i> hat den Wert 0, es ist keine Datenübertragung aktiv.
7001h	Erstaufwurf mit <i>REQ</i> = 1: Datenübertragung angestoßen; <i>BUSY</i> hat den Wert 1.
7002h	Zwischenaufwurf mit <i>REQ</i> = 1.
8090h	Angegebene Zieladresse des Kommunikationspartners ist ungültig, z.B. <ul style="list-style-type: none"> • falsche <i>IOID</i> • falsche Basisadresse vorhanden • falsche MPI-Adresse (> 126)
8095h	Der Baustein wird bereits in einer Prioritätsklasse mit niedrigerer Priorität bearbeitet.
80A0h	Fehler in empfangener Quittung.
80A1h	Kommunikationsprobleme: SFC-Aufruf nach Abbruch einer bestehenden Verbindung.
80B1h	Fehler im ANY-Pointer. Die Länge des zu übertragenden Datenbereichs ist falsch.
80B4h	Datentypfehler bei ANY-Pointer, oder ARRAY des angegebenen Datentyps nicht erlaubt.
80B6h	In der empfangenen Quittung steht ein unbekannter Fehlercode.
80BAh	Die Antwort des Kommunikationspartners passt nicht in das Kommunikations-telegramm.
80C0h	Die angegebene Verbindung ist durch einen anderen Auftrag bereits belegt.
80C1h	Ressourcenengpass bei der CPU, auf der der SFC abläuft, z.B.: <ul style="list-style-type: none"> • Die maximale Anzahl verschiedener Sendeaufträge des Moduls wird bereits bearbeitet. • Die Verbindungsressource ist z.B. durch Empfang belegt.
80C2h	Temporärer Ressourcenmangel des Kommunikationspartners, z.B.: <ul style="list-style-type: none"> • Der Kommunikationspartner bearbeitet momentan das Maximum an Aufträgen. • Die benötigten Betriebsmittel (Speicher etc.) sind belegt. • Zu wenig Arbeitsspeicher vorhanden (Stoßen Sie einen Komprimiervorgang an).
80C3h	Fehler beim Verbindungsaufbau, z.B.: <ul style="list-style-type: none"> • Die eigene Station befindet sich nicht am MPI-Subnetz. • Sie haben die eigene Station am MPI-Subnetz adressiert. • Der Kommunikationspartner ist nicht mehr erreichbar. • Temporärer Ressourcenmangel des Kommunikationspartners.

SFC 81 - UBLKMOV - Variable ununterbrechbar kopieren

Beschreibung Mit dem SFC 81 UBLKMOV (uninterruptable move) kopieren Sie den Inhalt eines Speicherbereiches (= Quellbereich) konsistent in einen anderen Speicherbereich (=Zielbereich). Der Kopiervorgang kann nicht durch andere Tätigkeiten des Betriebssystems unterbrochen werden.

Mit dem SFC 81 UBLKMOV können Sie alle Speicherbereiche kopieren außer:

- folgende Bausteine: FB, SFB, FC, SFC, OB, SDB
- Zähler
- Zeiten
- Speicherbereiche des Peripheriebereiches
- nicht ablaufrelevante Datenbausteine.

Die maximale Datenmenge, die Sie kopieren können, beträgt 512Byte.

Unterbrechbarkeit Der Kopiervorgang ist nicht unterbrechbar. Sie müssen daher beachten, dass sich die Alarmreaktionszeit Ihrer CPU bei Einsatz des SFC 81 erhöhen kann.

Parameter

Parameter	Deklaration	Datentyp	Speicherbereich	Beschreibung
SRCBLK	INPUT	ANY	E, A, M, D, L	Angabe des Speicherbereiches, der kopiert werden soll (Quellfeld). Arrays vom Datentyp STRING sind nicht erlaubt.
RET_VAL	OUTPUT	INT	E, A, M, D, L	Tritt während der Bearbeitung der Funktion ein Fehler auf, enthält der Rückgabewert einen Fehlercode.
DSTBLK	OUTPUT	ANY	E, A, M, D, L	Angabe des Speicherbereiches, in den kopiert werden soll (Zielfeld). Arrays vom Datentyp STRING sind nicht erlaubt.

**Hinweis!**

Quell- und Zielbereich dürfen sich nicht überlappen.

Ist der angegebene Zielbereich größer als der Quellbereich, dann werden auch nur so viele Daten in den Zielbereich kopiert, wie im Quellbereich stehen.

Ist der angegebenen Zielbereich kleiner als der Quellbereich, dann werden auch nur so viele Daten kopiert, wie der Zielbereich aufnehmen kann.

Ist der ANY-Pointer (Quelle oder Ziel) vom Typ BOOL, so muss die angegebene Länge durch 8 teilbar sein, da sonst der SFC nicht ausgeführt wird.

Ist der ANY-Pointer vom Typ STRING, so muss die angegebene Länge 1 sein.

RET_VAL**(Rückgabewert)**

Wert	Beschreibung
0000h	kein Fehler
8091h	Der Quellbereich liegt in einem nicht ablaufrelevanten Datenbaustein.

Teil 5 VIPA-spezifische Bausteine

Überblick

Hier finden Sie die Beschreibung der VIPA spezifischen Bausteine, die ausschließlich in den Standard-CPU's der Systeme 100V, 200V, 300V und 500V der VIPA Verwendung finden. Bitte beachten Sie, dass manche der hier aufgeführten Bausteine nicht in allen System-CPU's integriert sind. Die Zuordnung der entsprechenden Bausteine zu den Systemen finden Sie im Kapitel "Übersicht" in einer Tabelle.

Inhalt

Thema	Seite
Teil 5 VIPA-spezifische Bausteine.....	5-1
Übersicht.....	5-2
VIPA-Bibliothek einbinden.....	5-4
FB 55 - IP_CONFIG - Programmierte Kommunikationsverbindung.....	5-5
FC 0 - SEND - Senden an CP 240.....	5-10
FC 1 - RECEIVE - Empfangen von CP 240.....	5-11
FC 5 - AG_SEND / FC 6 - AG_RECV - CP 243 Kommunikation.....	5-12
FC 8 - STEUERBIT - Modemfunktionalität CP 240.....	5-17
FC 9 - SYNCHRON_RESET - Synchronisation CPU und CP 240.....	5-18
FC 11 - ASCII_FRAGMENT - Fragment Datenempfang CP 240.....	5-19
Serielle Kommunikation - SFC 207 und SFC 216...218.....	5-20
SFC 207 - SER_CTRL.....	5-21
SFC 216 - SER_CFG.....	5-22
SFC 217 - SER_SND.....	5-26
SFC 218 - SER_RCV.....	5-29
SFC 219 - CAN_TLGR - CAN-Telegramm senden.....	5-30
MMC-Zugriff - SFC 220...222.....	5-33
SFC 220 - MMC_CR_F.....	5-34
SFC 221 - MMC_RD_F.....	5-36
SFC 222 - MMC_WR_F.....	5-37
SFC 223 - PWM - Pulsweitenmodulation.....	5-38
SFC 224 - HSC - High-speed-Counter.....	5-40
SFC 225 - HF_PWM - HF Pulsweitenmodulation.....	5-42
SFC 227 - TD_PRM - TD200-Kommunikation.....	5-44
SFC 228 - RW_KACHEL - Kacheldirektzugriff.....	5-46
Kachelkommunikation - SFC 230 ... 238.....	5-48
Kachelkommunikation - Parameterübergabe.....	5-51
Kachelkommunikation - Quell- bzw. Zielangaben.....	5-52
Kachelkommunikation - Anzeigenwort ANZW.....	5-55
Kachelkommunikation - Parametrierfehler PAFE.....	5-62
SFC 230 - SEND.....	5-63
SFC 231 - RECEIVE.....	5-64
SFC 232 - FETCH.....	5-65
SFC 233 - CONTROL.....	5-66
SFC 234 - RESET.....	5-67
SFC 235 - SYNCHRON.....	5-68
SFC 236 - SEND_ALL.....	5-69
SFC 237 - RECEIVE_ALL.....	5-70
SFC 238 - CTRL1.....	5-71

Übersicht

Allgemeines Die integrierten SFCs sind in Maschinensprache programmiert und laufen deshalb mit hoher Geschwindigkeit ab. Sie belegen keinen Platz im internen Programmspeicher. Die integrierten Bausteine werden im Anwenderprogramm aufgerufen.

Zuordnungstabelle Baustein ↔ CPU Da nicht jeder Baustein in jede CPU-Familie integriert ist, finden Sie nachfolgend eine Zuordnungstabelle zwischen Baustein und CPU. Über die jeweilige 1. Ziffer ist die CPU-Familie spezifiziert beispielsweise bedeutet 31x: CPU 31x aus dem System 300V. Zur Übersicht finden Sie nachfolgend bei jedem Baustein den entsprechenden Auszug aus dieser Zuordnungstabelle.

Baustein	Bezeichnung	Beschreibung	11x	21x	31x	51x
FB 55	IP_CONF	Programmierte Kommunikationsverbindungen		•*		
FC 0	SEND	Daten an CP 240 senden		•		
FC 1	RECEIVE	Daten von CP 240 empfangen		•		
FC 5	AG_SEND	Daten an Ethernet-CP 243 senden		•*		
FC 6	AG_RECEIVE	Daten von Ethernet-CP 243 empfangen		•*		
FC 8	STUERBIT	Modemfunktionalität des CP 240		•		
FC 9	SYNCHRON_RESET	Synchronisation zwischen CPU und CP 240		•		
FC 11	ASCII_FRAGMENT	Fragmentierter ASCII-Datenempfang mit CP 240		•		
SFC 204	IP_CONF	wird intern für FB 55 IP_CONF verwendet		•*		
SCF 205	AG_SEND	wird intern für FC 5 AG_SEND verwendet		•*		
SCF 206	AG_RECV	wird intern für FC 6 AG_RECEIVE verwendet		•*		
SFC 207	SER_CTRL	RS232 Modemfunktionalität	•	•		
SFC 216	SER_CFG	RS232 Parametrieren	•	•		
SFC 217	SER_SND	RS232 Senden	•	•		
SFC 218	SER_RCV	RS232 Empfangen	•	•		
SFC 219	CAN_TLGR	CAN Telegramm senden	•	•	•	•
SFC 220	MMC_CR_F	File auf MMC erzeugen	•	•	•	•
SFC 221	MMC_RD_F	von File auf MMC lesen	•	•	•	•
SFC 222	MMC_WR_F	in File auf MMC schreiben	•	•	•	•
SFC 223	PWM	Pulsweitenmodulation parametrieren	•			
SFC 224	HSC	High-speed-Counter parametrieren	•			
SFC 225	HF_PWM	HF Pulsweitenmodulation (bis 50kHz) parametrieren	•			
SFC 227	TD_PRM	Parametrierung für TD200 Kommunikation	•	•	•	•

*) nur in CPU 21x-2BT1x vorhanden

Fortsetzung ...

... Fortsetzung

Baustein	Bezeichnung	Beschreibung	11x	21x	31x	51x
SFC 228	RW_KACHEL	Kachel lesen/schreiben		•	•	•
SFC 230	SEND	Send über Kachel		•	•	•
SFC 231	RECEIVE	Receive über Kachel		•	•	•
SFC 232	FETCH	Fetch über Kachel		•	•	•
SFC 233	CONTROL	Control für Kachelkommunikation		•	•	•
SFC 234	RESET	Reset für Kachelkommunikation		•	•	•
SFC 235	SYNCHRON	Synchron für Kachelkommunikation		•	•	•
SFC 236	SEND_ALL	Send_All über Kachel		•	•	•
SFC 237	RECV_ALL	Receive All über Kachel		•	•	•
SFC 238	CONTROL1	Control für Kachelkommunikation mit Typ ANZW: Zeiger und Parameter IND.		•	•	•

VIPA-Bibliothek einbinden

Übersicht

Die VIPA-spezifischen Bausteine finden Sie im Service-Bereich auf www.vipa.de unter *Downloads > VIPA LIB* als Bibliothek zum Download.

Die Bibliothek liegen als gepackte zip-Datei vor.

Sobald Sie VIPA-spezifische Bausteine verwenden möchten, sind diese in Ihr Projekt zu importieren.

Folgende Schritte sind hierzu erforderlich:

- Zum Entpacken der Datei FX000011_Vxxx.zip entpacken
- Bibliothek "dearchivieren"
- Bibliothek öffnen und Bausteine in Projekt übertragen

FX000011_Vxxx.zip entpacken

Starten Sie mit einem Doppelklick auf die Datei FX000011_Vxxx.zip Ihr Unzip-Programm und kopieren Sie die Datei VIPA.ZIP in Ihr Arbeitsverzeichnis. Es ist nicht erforderlich diese Datei weiter zu entpacken.

Bibliothek dearchivieren

Zur Dearchivierung Ihrer Bibliothek für die SPEED7-CPU's starten Sie den SIMATIC Manager von Siemens. Über **Datei > Dearchivieren** öffnen Sie ein Dialogfenster zur Auswahl des Archivs. Navigieren Sie in Ihr Arbeitsverzeichnis.

Wählen Sie VIPA.ZIP an und klicken Sie auf [Öffnen].

Geben Sie ein Zielverzeichnis an, in dem die Bausteine abzulegen sind. Mit [OK] startet der Entpackvorgang.

Bibliothek öffnen und Bausteine in Projekt übertragen

Öffnen Sie die Bibliothek nach dem Entpackvorgang.

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.

11x	21x	31x	51x
	✓*		

FB 55 - IP_CONFIG - Programmierte Kommunikationsverbindung

Übersicht

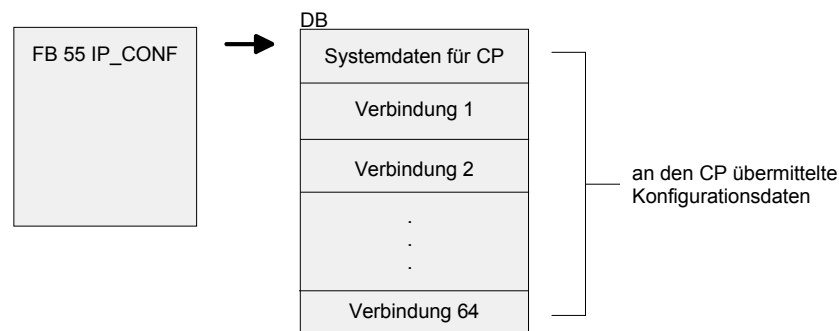
Es gibt Anwendungsbereiche, bei denen es vorteilhaft ist, die Kommunikationsverbindungen nicht über Siemens NetPro, sondern über eine spezifische Applikation programmgesteuert einzurichten.

Für diese Anwendungen steht Ihnen ein VIPA-Baustein (FB 55) zur Verfügung, der die flexible Übergabe von Datenbausteinen mit Projektiertdaten an einen CP ermöglicht.

Innerhalb des FB 55 wird der SFC 204 aufgerufen, der als Sonderfunktionsbaustein in der CPU abliegt.

Prinzip

Über den FB 55, der im Anwenderprogramm aufgerufen wird, können Konfigurationsdaten für Kommunikationsverbindungen an den CP übertragen werden.



Der Konfigurations-DB kann jederzeit in den CP geladen werden.



Achtung!

Sobald das Anwenderprogramm über den FB 55 IP_CONFIG die Verbindungsdaten übergibt, schaltet die CPU den CP kurzzeitig in STOP. Der CP übernimmt die Systemdaten (inklusive IP-Adresse) und die neuen Verbindungsdaten und arbeitet diese im Anlauf ab (RUN).

*) nur in CPU 21x-2BT1x vorhanden

**FB 55 -
IP_CONFIG**

Je nach Größe des Konfigurations-DB erfolgt die Übertragung zum CP in mehreren Segmenten. Sie müssen daher den FB solange erneut aufrufen, bis der FB mit dem *DONE*-Bit = 1 die vollständige Übertragung signalisiert. Der Auftrag wird ausgeführt, sobald *ACT* = 1 übergeben wird.

Parameter

Parameter	Deklaration	Datentyp	Speicherbereich	Beschreibung
ACT	INPUT	BOOL	E, A, M, D, L	Beim FB-Aufruf mit <i>ACT</i> = 1 wird der DBxx an den CP gesendet. Beim FB-Aufruf mit <i>ACT</i> = 0 werden nur die Statusanzeigen <i>DONE</i> , <i>ERROR</i> und <i>STATUS</i> aktualisiert.
LADDR	INPUT	WORD	E, A, M, D, Konstante	Baugruppen-Anfangsadresse Bei der Konfiguration des CP mit dem Hardware-Konfigurator von Siemens wird die Baugruppen-Anfangsadresse in der Konfigurationstabelle ausgegeben. Geben Sie diese Adresse hier an.
CONF_DB	INPUT	ANY	E, A, M, D	Der Parameter zeigt die Anfangsadresse des Konfigurations-Datenbereichs in einem DB.
LEN	INPUT	INT	E, A, M, D, Konstante	Längenangabe in Byte für den Konfigurations-Datenbereich.
DONE	OUTPUT	BOOL	E, A, M, D, L	Der Parameter zeigt an, ob der Konfigurations-Datenbereich vollständig übertragen wurde. Beachten Sie, dass der FB je nach Größe des Konfigurations-Datenbereichs (in mehreren Zyklen) mehrfach angestoßen werden muss, bis die Anzeige <i>DONE</i> = 1 den Abschluss signalisiert.
ERROR	OUTPUT	BOOL	E, A, M, D, L	Fehleranzeige
STATUS	OUTPUT	WORD	E, A, M, D	Statusanzeige
EXT_STATUS	OUTPUT	WORD	E, A, M, D	Der Parameter zeigt bei einer fehlerhaften Auftragsausführung an, welcher Parameter im Konfigurations-DB als Fehlerursache erkannt wurde. High Byte: Index des Parameter-Blocks Low Byte: Index des Subblocks innerhalb des Parameterblocks

**Fehler-
informationen**

ERROR	STATUS	Beschreibung
0	0000h	Auftrag fertig ohne Fehler
0	8181h	Auftrag läuft
1	80B1h	Anzahl der zu sendenden Daten überschreitet die für diesen Dienst zulässige Obergrenze.
1	80C4h	Kommunikationsfehler Fehler kann temporär auftreten, deshalb ist eine Wiederholung im Anwenderprogramm sinnvoll.
1	80D2h	Projektierungsfehler, die eingesetzte Baugruppe unterstützt diesen Dienst nicht.
1	8183h	CP lehnt die angeforderte Datensatznummer ab.
1	8184h	Systemfehler bzw. unzulässiger Parametertyp.
1	8185h	Wert des Parameters <i>LEN</i> ist größer als der <i>CONF_DB</i> abzüglich des reservierten Headers (4Byte) oder die Längenangabe ist falsch.
1	8186h	Unzulässigen Parameter erkannt. ANY-Pointer <i>CONF_DB</i> verweist nicht auf einen DB.
1	8187h	Ungültiger Zustand des FBs. Daten im Header des <i>CONF_DB</i> überschrieben oder Instanz-DB nicht vorhanden.
1	8A01h	Statusanzeige im gelesenen Datensatz ist ungültig (Wert >=3)
1	8A02h	Es läuft kein Auftrag auf dem CP; der FB hat jedoch eine Quittung für gelaufenen Auftrag erwartet.
1	8A03h	Es läuft kein Auftrag auf dem CP und der CP ist nicht bereit; der FB hat einen ersten Auftrag für Datensatz-Lesen angestoßen.
1	8A04h	Es läuft kein Auftrag auf dem CP und der CP ist nicht bereit; der FB hat jedoch eine Quittung für den gelaufenen Auftrag erwartet.
1	8A05h	Auftrag läuft, eine Quittung ist jedoch noch nicht erfolgt; der FB hat einen ersten Auftrag für Datensatz-Lesen angestoßen.
1	8A06h	Auftrag ist fertig; der FB hat jedoch einen ersten Auftrag für Datensatz-Lesen angestoßen.
1	8B01h	Kommunikationsfehler, DB konnte nicht übertragen werden.
1	8B02h	Parameterfehler, Doppelter Parameterblock
1	8B03h	Parameterfehler, Subblock im Parameterblock ist nicht erlaubt.
1	8B04h	Parameterfehler, Länge die im FB angegeben wurde, stimmt nicht mit der Länge der Parameterblöcke/Subblöcke überein.
1	8B05h	Parameterfehler, Länge des Parameterblocks ist ungültig.
1	8B06h	Parameterfehler, Länge des Subblocks ist ungültig.
1	8B07h	Parameterfehler, ID des Parameterblocks ist ungültig.
1	8B08h	Parameterfehler, ID des Subblocks ist ungültig.
1	8B09h	Systemfehler, Verbindung existiert nicht.
1	8B0Ah	Datenfehler, Inhalt des Subblocks ist nicht korrekt.
1	8B0Bh	Strukturfehler, Subblock ist doppelt aufgeführt.
1	8B0Ch	Datenfehler, im Parameterblock sind nicht alle erforderlichen Parameter enthalten.
1	8B0Dh	Datenfehler, der <i>CONF_DB</i> enthält keinen Parameterblock für Systemdaten.

... Fortsetzung

... Fortsetzung

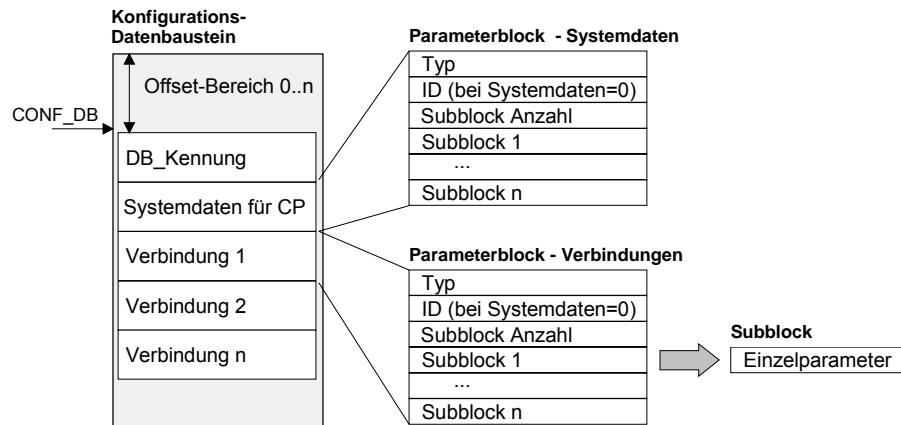
ERROR	STATUS	Beschreibung
1	8B0Eh	Datenfehler/Strukturfehler, Typ des <i>CONF_DB</i> ist ungültig.
1	8B0Fh	Systemfehler, CP hat zu wenig Ressourcen, um den <i>CONF_DB</i> vollständig bearbeiten zu können.
1	8B10	Datenfehler, es ist nicht eingestellt, das die Konfiguration über das Anwenderprogramm erfolgt.
1	8B11	Datenfehler, der angegebene Typ des Parameterblocks ist ungültig.
1	8B12	Datenfehler, es wurden zu viele Verbindungen angegeben.
1	8B13	CP interner Fehler
1	8F22h	Bereichslängenfehler beim Lesen eines Parameters.
1	8F23h	Bereichslängenfehler beim Schreiben eines Parameters.
1	8F24h	Bereichsfehler beim Lesen eines Parameters.
1	8F25h	Bereichsfehler beim Schreiben eines Parameters.
1	8F28h	Ausrichtungsfehler beim Lesen eines Parameters.
1	8F29h	Ausrichtungsfehler beim Schreiben eines Parameters.
1	8F30h	Parameter liegt im schreibgeschützten 1. aktuellen DB
1	8F31h	Parameter liegt im schreibgeschützten 2. aktuellen DB
1	8F32h	Parameter enthält eine zu große DB-Nummer.
1	8F33h	DB-Nummernfehler
1	8F3Ah	Zielbereich wurde nicht geladen (DB).
1	8F42h	Quittungsverzug beim Lesen eines Parameters aus dem Peripheriebereich.
1	8F43h	Quittungsverzug beim Schreiben eines Parameters in den Peripheriebereich.
1	8F44h	Zugriff auf einen in der Bausteinbearbeitung zu lesenden Parameter ist gesperrt.
1	8F45h	Zugriff auf einen in der Bausteinbearbeitung zu schreibenden Parameter ist gesperrt.
1	8F7Fh	Interner Fehler

Konfigurations-Datenbaustein

Der Konfigurations-Datenbaustein (CONF_DB) enthält sämtliche Verbindungsdaten und Konfigurationsdaten (IP-Adresse, Subnetz-Maske, Default-Router, NTP-Uhrzeit-Server und weitere) für einen Ethernet-CP. Der Konfigurations-DB wird mit dem FB 55 an den CP übergeben.

Aufbau

Der CONF_DB kann über einen Offset-Bereich an beliebiger Stelle innerhalb eines Datenbausteines beginnen. Jede Verbindung sowie die Systemdaten werden durch einen identisch aufgebauten Parameterblock beschrieben. Einzelne Parameter werden durch Subblöcke typisiert.



Parameterblock für Systemdaten für CP

Nachfolgend finden Sie die für die Vernetzung des CPs relevanten Subblöcke. Diese sind im Parameterblock für Systemdaten für CP anzugeben.

Je nach Anwendungsfall sind nicht alle Subblock-Typen erforderlich.

Aufbau

Typ = 0
ID = 0
Subblock Anzahl = n
Subblock 1
Subblock 2
Subblock n

Subblock				Parameter	
ID	Typ	Länge (Byte)	Bedeutung	Besonderheiten	Anwendung
1	SUB_IP_V4	4 + 4	IP-Adresse lokale Station gemäß IPv4		zwingend
2	SUB_NETMASK	4 + 4	Subnetzmaske der lokalen Station		zwingend
4	SUB_DNS_SERV_ADDR	4 + 4	DNS Server Adresse	Kann bis zu 4 mal vorkommen. Erster Eintrag ist der primäre DNS Server.	optional
8	SUB_DEF_ROUTER	4 + 4	IP Adresse des Default Routers		optional
14	SUB_DHCP_ENABLE	4 + 1	IP-Adresse von einem DHCP-Server beziehen	0: kein DHCP 1: DHCP	optional
15	SUB_CLIENT_ID	Länge Client-ID + 4	-	-	optional

11x	21x	31x	51x
	✓		

FC 0 - SEND - Senden an CP 240

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

Parameter

Name	Deklaration	Typ	Beschreibung
ADR	IN	INT	Peripherieadresse
_DB	IN	BLOCK_DB	DB-Nummer mit den Sendedaten
ABD	IN	WORD	Nummer des 1. Datenworts
ANZ	IN	WORD	Anzahl der Bytes
FRG	IN_OUT	BOOL	Auftrag anstoßen
GESE	IN_OUT	WORD	wird intern verwendet
ANZ_INT	IN_OUT	WORD	wird intern verwendet
ENDE_KOMM	IN_OUT	BOOL	wird intern verwendet
LETZTER_BLOCK	IN_OUT	BOOL	wird intern verwendet
SENDEN_LAEUFT	IN_OUT	BOOL	Status der Funktion
FEHLER_KOM	IN_OUT	BOOL	wird intern verwendet
PAFE	OUT	BYTE	Parametrierfehler Code (0 = OK)

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

_DB Nummer des Datenbausteins, der die zu sendenden Daten beinhaltet.

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

ANZ Anzahl der Bytes, die zu übertragen sind.

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

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

1 = Datenbaustein nicht vorhanden

2 = Datenbaustein zu kurz

3 = Datenbausteinnummer nicht im gültigen Bereich

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

11x	21x	31x	51x
	✓		

FC 1 - RECEIVE - Empfangen von CP 240

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

Parameter

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

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

_DB Nummer des Datenbausteins, der die empfangenen Daten beinhaltet.

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

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

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

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

1 = Datenbaustein nicht vorhanden

2 = Datenbaustein zu kurz

3 = Datenbausteinnummer nicht im gültigen Bereich

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

11x	21x	31x	51x
	✓*		

FC 5 - AG_SEND / FC 6 - AG_RECV - CP 243 Kommunikation

Übersicht

Die beiden Bausteine dienen der Verarbeitung von Verbindungsaufträgen auf SPS-Seite eines Ethernet-CP 243. Durch Einbindung dieser Bausteine in den Zyklus-Baustein OB 1 können Sie zyklisch Daten senden und empfangen.

Innerhalb dieser Bausteine werden die SFCs 205 und 206 aufgerufen, die als Sonderfunktionsbausteine in der CPU abliegen.



Hinweis!

Bitte beachten Sie, dass Sie in Ihrem Anwenderprogramm für die Kommunikation mit VIPA-CPs ausschließlich die SEND/RECV-FCs von VIPA einsetzen dürfen. Bei Wechsel zu VIPA-CPs in einem schon bestehenden Projekt können die bestehenden AG_SEND/AG_LSEND bzw. AG_RECV/AG_LRECV durch AG_SEND bzw. AG_RECV von VIPA ohne Anpassung ersetzt werden. Da sich der CP automatisch an die Länge der zu übertragenden Daten anpasst ist die L-Variante von SEND bzw. RECV bei VIPA nicht erforderlich.

Kommunikationsbausteine

Für die Kommunikation zwischen CPU und Ethernet-CP 243 stehen Ihnen folgende FCs zur Verfügung:

AG_SEND (FC 5)

Dieser Baustein übergibt die Nutzdaten aus dem über *SEND* angegebenen Datenbereich an den über *ID* und *LADDR* spezifizierten CP. Als Datenbereich können Sie einen PA-, Merker- oder Datenbaustein-Bereich angeben. Wurde der Datenbereich fehlerfrei übertragen, so wird "Auftrag fertig ohne Fehler" zurückgemeldet.

AG_RECV (FC 6)

Der Baustein übernimmt vom CP die Nutzdaten und legt sie in dem über *RECV* definierten Datenbereich ab. Als Datenbereich können Sie einen PE-, Merker- oder Datenbaustein-Bereich angeben. Wurde der Datenbereich fehlerfrei übernommen, so wird "Auftrag fertig ohne Fehler" zurückgemeldet.

Statusanzeigen

Der CP bearbeitet Sende- und Empfangsaufträge unabhängig vom CPU-Zyklus und benötigt hierzu eine Übertragungszeit. Die Schnittstelle mit den FC-Bausteinen zum Anwenderprogramm wird hierbei über Quittungen synchronisiert.

Für die Statusauswertung liefern die Kommunikationsbausteine Parameter zurück, die Sie in Ihrem Anwenderprogramm direkt auswerten können.

Diese Statusanzeigen werden bei jedem Baustein-Aufruf aktualisiert.

Einsatz unter hoher Kommunikationslast

Verwenden Sie keine zyklischen Aufrufe der Kommunikationsbausteine im OB 1. Dies führt zu einer ständigen Kommunikation zwischen CPU und CP. Programmieren Sie stattdessen Ihre Kommunikationsbausteine in einem Zeit-OB, deren Zykluszeit größer ist als die des OB1 bzw. ereignisgesteuert.

*) nur in CPU 21x-2BT1x vorhanden

Aufruf FC schneller als CP-Übertragungszeit

Wird ein Baustein im Anwenderprogramm erneut aufgerufen, bevor die Daten vollständig gesendet oder empfangen wurden, wird an der Schnittstelle der FC-Bausteine wie folgt verfahren:

AG_SEND

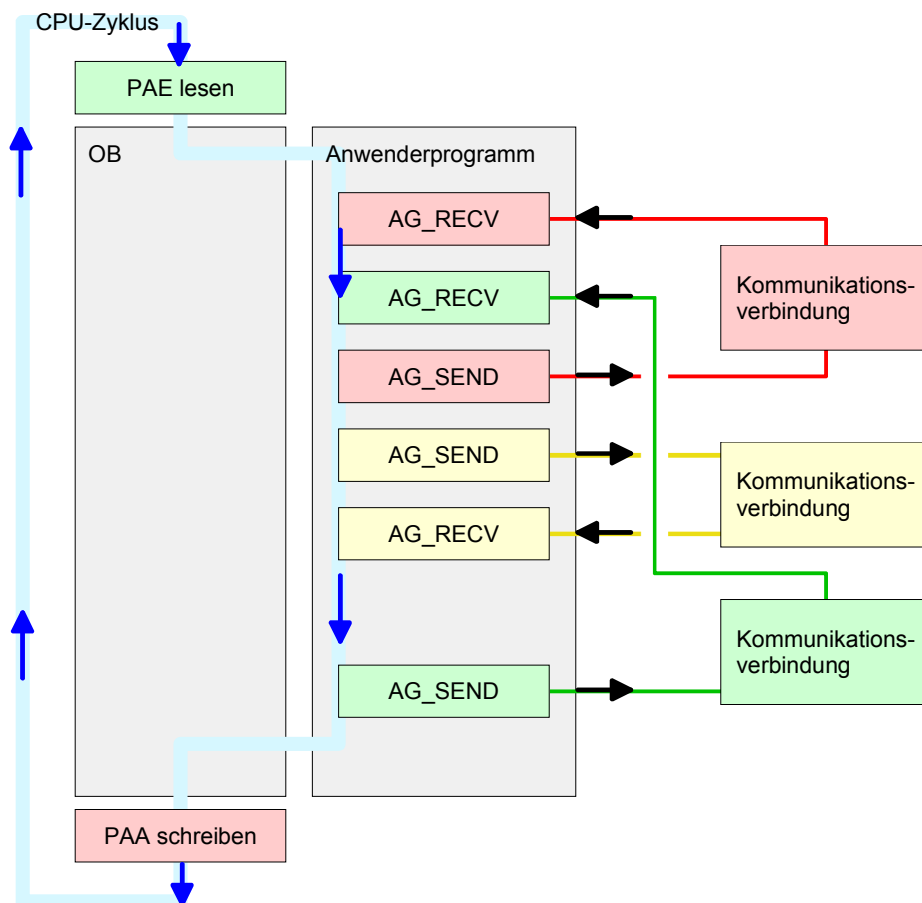
Es wird kein Auftrag entgegen genommen, bis die Datenübertragung über die Verbindung vom Partner quittiert wurde. Solange erhalten Sie die Meldung "Auftrag läuft", bis der CP den nächsten Auftrag für die gleiche Verbindung übernehmen kann.

AG_RECV

Der Auftrag wird mit der Meldung "Es liegen noch keine Daten vor" quittiert, solange der CP die Empfangsdaten noch nicht vollständig empfangen hat.

AG_SEND, AG_RECV im Anwenderprogramm

Eine mögliche Ablaufsequenz für die FC-Bausteine zusammen mit den Organisations- und Programmbausteinen im CPU-Zyklus ist nachfolgend dargestellt:



Die FC-Bausteine mit zugehöriger Kommunikationsverbindung sind farblich zusammengefasst. Hier können Sie auch erkennen, dass Ihr Anwenderprogramm aus beliebig vielen Bausteinen bestehen kann. Somit können Sie ereignis- bzw. programmgesteuert an beliebiger Stelle im CPU-Zyklus mit AG_SEND Daten senden bzw. mit AG_RECV Daten empfangen.

Sie können die Bausteine für **eine** Kommunikationsverbindung auch mehrmals in einem Zyklus aufrufen.

AG_SEND (FC 5) Mit AG_SEND werden die zu sendenden Daten an den Ethernet-CP 243 übertragen.

Parameter

Name	Deklaration	Typ	Beschreibung
ACT	IN	BOOL	Aktivierung des Senders 0: Aktualisiert die <i>DONE</i> , <i>ERROR</i> und <i>STATUS</i> 1: Der unter SEND mit der Länge <i>LEN</i> abgelegte Datenbereich wird gesendet
ID	IN	INT	Verbindungsnummer 1 ... 16 (identisch mit <i>ID</i> aus NetPro)
LADDR	IN	WORD	Logische Basisadresse des CPs (identisch mit <i>LADDR</i> aus NetPro)
SEND	IN	ANY	Datenbereich
LEN	IN	INT	Anzahl der Bytes, die aus dem Datenbereich zu übertragen sind
DONE	OUT	BOOL	Zustandsparameter für den Auftrag 0: Auftrag läuft 1: Auftrag fertig ohne Fehler
ERROR	OUT	BOOL	Fehleranzeige 0: Auftrag läuft (bei <i>DONE</i> = 0) 0: Auftrag fertig ohne Fehler (bei <i>DONE</i> = 1) 1: Auftrag fertig mit Fehler
STATUS	OUT	WORD	Statusanzeige, die in Verbindung mit <i>DONE</i> und <i>ERROR</i> zurückgeliefert wird. Näheres hierzu finden Sie in der nachfolgenden Tabelle.

AG_RECV (FC 6) Mit AG_RECV werden die Daten, die der Ethernet-CP 243 empfangen hat, in die CPU übertragen.

Parameter

Name	Deklaration	Typ	Beschreibung
ID	IN	INT	Verbindungsnummer 1 ... 16 (identisch mit <i>ID</i> aus NetPro)
LADDR	IN	WORD	Logische Basisadresse des CPs (identisch mit <i>LADDR</i> aus NetPro)
RECV	IN	ANY	Datenbereich für die empfangenen Daten
NDR	OUT	BOOL	Zustandsparameter für den Auftrag 0: Auftrag läuft 1: Auftrag fertig Daten wurden ohne Fehler übernommen
ERROR	OUT	BOOL	Fehleranzeige 0: Auftrag läuft (bei <i>NDR</i> = 0) 0: Auftrag fertig ohne Fehler (<i>NDR</i> = 1) 1: Auftrag fertig mit Fehler
STATUS	OUT	WORD	Statusanzeige, die in Verbindung mit <i>NDR</i> und <i>ERROR</i> zurückgeliefert wird. Näheres hierzu finden Sie in der nachfolgenden Tabelle.
LEN	OUT	INT	Anzahl der Bytes, die empfangen wurden

**DONE, ERROR,
STATUS**

In der nachfolgenden Tabelle sind alle Meldungen aufgeführt, die der Ethernet-CP 243 nach einem SEND-Auftrag bzw. RECV-Auftrag zurückliefern kann.

Ein "-" bedeutet, dass diese Meldung für den entsprechenden SEND- bzw. RECV-Auftrag nicht existiert.

DONE (SEND)	NDR (RECV)	ERROR	STATUS	Bedeutung
1	-	0	0000h	Auftrag fertig ohne Fehler
-	1	0	0000h	Neue Daten wurden ohne Fehler übernommen
0	-	0	0000h	Kein Auftrag in Bearbeitung
-	0	0	8180h	Es liegen noch keine Daten vor
0	0	0	8181h	Auftrag läuft
0	0	1	8183h	Für diesen Auftrag gibt es keine CP-Projektierung
0	-	1	8184h	Es ist ein Systemfehler aufgetreten
-	0	1	8184h	Es ist ein Systemfehler aufgetreten (Quelldatenbereich fehlerhaft)
0	-	1	8185h	Parameter <i>LEN</i> größer als Quell-Bereich <i>SEND</i>
	0	1	8185h	Ziel-Puffer (RECV) ist zu klein
0	0	1	8186h	Parameter <i>ID</i> ungültig (nicht im Bereich 1 ... 16)
0	-	1	8302h	Keine Empfangsressourcen bei Ziel-Station, Empfänger-Station kann empfangene Daten nicht schnell genug verarbeiten bzw. hat keine Empfangsressourcen bereitgestellt.
0	-	1	8304h	Die Verbindung ist nicht aufgebaut. Der Sendeauftrag sollte erst nach einer Wartezeit >100 ms erneut abgesetzt werden.
-	0	1	8304h	Die Verbindung ist nicht aufgebaut. Der Empfangsauftrag sollte erst nach einer Wartezeit > 100ms erneut abgesetzt werden.
0	-	1	8311h	Zielstation ist unter der angegebenen Ethernet-Adresse nicht erreichbar.
0	-	1	8312h	Ethernet-Fehler im CP
0		1	8F22h	Quell-Bereich ungültig, wenn beispielsweise Bereich im DB nicht vorhanden, Parameter <i>LEN</i> < 0
-	0	1	8F23h	Quell-Bereich ungültig, wenn beispielsweise Bereich im DB nicht vorhanden, Parameter <i>LEN</i> < 0
0	-	1	8F24h	Bereichsfehler beim Lesen eines Parameters.
-	0	1	8F25h	Bereichsfehler beim Schreiben eines Parameters.
0	-	1	8F28h	Ausrichtungsfehler beim Lesen eines Parameters.
-	0	1	8F29h	Ausrichtungsfehler beim Schreiben eines Parameters.
-	0	1	8F30h	Parameter liegt im schreibgeschützten 1. akt. Datenbaustein
-	0	1	8F31h	Parameter liegt im schreibgeschützten 2. akt. Datenbaustein
0	0	1	8F32h	Parameter enthält zu große DB-Nummer.
0	0	1	8F33h	DB-Nummer Fehler
0	0	1	8F3Ah	Bereich nicht geladen (DB)

Fortsetzung ...

... Fortsetzung *DONE*, *ERROR*, *STATUS*

DONE (SEND)	NDR (RECV)	ERROR	STATUS	Bedeutung
0	-	1	8F42h	Quittungsverzug beim Lesen eines Parameters aus dem Peripheriebereich.
-	0	1	8F43h	Quittungsverzug beim Schreiben eines Parameters in den Peripheriebereich.
0	-	1	8F44h	Adresse des zu lesenden Parameters in der Zugriffsspur gesperrt.
-	0	1	8F45h	Adresse des zu schreibenden Parameters in der Zugriffsspur gesperrt.
0	0	1	8F7Fh	Interner Fehler z.B. unzulässige ANY-Referenz z.B. Parameter <i>LEN</i> = 0.
0	0	1	8090h	Baugruppe mit dieser Baugruppen-Anfangsadresse nicht vorhanden oder CPU in STOP.
0	0	1	8091h	Baugruppen-Anfangsadresse nicht auf Doppel-Wort-Raster.
0	0	1	8092h	In ANY-Referenz ist eine Typangabe ungleich BYTE angegeben.
-	0	1	80A0h	Negative Quittung beim Lesen von Baugruppe.
0	0	1	80A4h	reserviert
0	0	1	80B0h	Baugruppe kennt den Datensatz nicht.
0	0	1	80B1h	Die Längenangabe (im Parameter <i>LEN</i>) ist falsch.
0	0	1	80B2h	reserviert
0	0	1	80C0h	Datensatz kann nicht gelesen werden.
0	0	1	80C1h	Der angegebene Datensatz ist gerade in Bearbeitung.
0	0	1	80C2h	Es liegt ein Auftragsstau vor.
0	0	1	80C3h	Die Betriebsmittel (Speicher) der CPU sind temporär belegt.
0	0	1	80C4h	Kommunikationsfehler (tritt temporär auf; daher ist eine Wiederholung im Anwenderprogramm sinnvoll.)
0	0	1	80D2h	Baugruppen-Anfangsadresse ist falsch.

Status-Parameter bei Neuanlauf

Bei einem Neuanlauf des CP werden die Ausgabe-Parameter wie folgt zurückgesetzt:

- *DONE* = 0
- *NDR* = 0
- *ERROR* = 0
- *STATUS* = 8180h (bei AG_RECV)
- *STATUS* = 8181h (bei AG_SEND)

11x	21x	31x	51x
	✓		

FC 8 - STEUERBIT - Modemfunktionalität CP 240

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

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

Schreiben: DTR, RTS

Parameter

Name	Deklaration	Typ	Beschreibung
ADR	IN	INT	Peripherieadresse
RTS	IN	BOOL	Neuer Zustand RTS
DTR	IN	BOOL	Neuer Zustand DTR
MASKE_RTS	IN	BOOL	0: nichts ändern 1: Zustand an RTS übergeben
MASKE_DTR	IN	BOOL	0: nichts ändern 1: Zustand an DTR übergeben
STATUS	OUT	BYTE	Status
DELTA_STATUS	OUT	BYTE	Statusänderungen seit dem letzten Zugriff
START	IN_OUT	BOOL	Auftrag anstoßen
AUFTRAG_LAEU	IN_OUT	BOOL	Status der Funktion
RET_VAL	OUT	WORD	Rückgabewert (0 = OK)



Hinweis!

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

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

RTS, DTR Mit diesem Parameter geben Sie den Status für *RTS* bzw. *DTR* vor, den Sie über *MASK_RTS* bzw. *MASK_DTR* aktivieren können.

MASK_RTS, MASK_DTR Hier wird mit 1 der Status des entsprechenden Parameters übernommen, sobald Sie *START* auf 1 setzen.

STATUS, DELTA_STATUS *STATUS* liefert den aktuellen Status der Modem-Leitungen zurück. *DELTA_STATUS* liefert den Status der Modem-Leitungen zurück, die sich seit dem letzten Zugriff geändert haben.

Die Bytes haben folgenden Aufbau:

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

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

AUFTRAG_LAEU Solange die Funktion abgearbeitet wird, bleibt dieses Bit gesetzt.

RET_VAL Dieser Parameter liefert zur Zeit immer 00h zurück und dient zukünftigen Fehlermeldungen.

11x	21x	31x	51x
	✓		

FC 9 - SYNCHRON_RESET - Synchronisation CPU und CP 240

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



Hinweis!

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

Parameter

Name	Deklaration	Typ	Beschreibung
ADR	IN	INT	Peripherieadresse
TIMER_NR	IN	WORD	Timer-Nummer
ANL	IN_OUT	BOOL	CPU-Neustart erfolgt
NULL	IN_OUT	BOOL	wird intern verwendet
RESET	IN_OUT	BOOL	Reset an CP
STUERB_S	IN_OUT	BYTE	wird intern verwendet
STUERB_R	IN_OUT	BYTE	wird intern verwendet

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

TIMER_NR Nummer des Timers für die Wartezeit.

ANL Mit *ANL* = 1 wird dem Hantierungsbaustein mitgeteilt, dass an der CPU STOP/START bzw. NETZ-AUS/NETZ-EIN erfolgt ist und nun eine Synchronisation erfolgen muss. Nach der Synchronisation wird *ANL* automatisch zurückgesetzt.

NULL Parameter wird intern verwendet.

RESET Mit *RESET* = 1 können Sie den CP aus Ihrem Anwenderprogramm zurücksetzen.

STUERB_S Hier ist das Merkerbyte anzugeben, in dem die Steuerbits ENDE_KOM, LETZTER_BLOCK, SENDEN_LAEUFT und FEHLER_KOM für den SEND-FC abgelegt sind.

STUERB_R Hier ist das Merkerbyte anzugeben, in dem die Steuerbits LETZTER_BLOCK, EMPF_LAEUFT und FEHLER_EMPF für den RECEIVE-FC abgelegt sind.

11x	21x	31x	51x
	✓		

FC 11 - ASCII_FRAGMENT - Fragment Datenempfang CP 240

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

Parameter

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

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

_DB Nummer des Datenbausteins, der die empfangenen Daten beinhaltet.

ABD Wortvariable, die die Nummer des Datenworts enthält, ab dem die empfangenen Zeichen abgelegt sind.

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

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

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

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

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

11x	21x	31x	51x
✓	✓		

Serielle Kommunikation - SFC 207 und SFC 216...218

Allgemein Zur seriellen Prozessankopplung zu verschiedenen Ziel- oder Quellsystemen besitzen manche CPUs eine serielle Schnittstelle, die Sie mittels dieser hier aufgeführten Bausteine zur Laufzeit ansteuern können.

Protokolle Unterstützt werden die Protokolle ASCII, STX/ETX, 3964R, USS und Modbus.

Parametrierung Die Parametrierung erfolgt zur Laufzeit unter Einsatz des SFC 216 (SER_CFG). Hierbei sind für alle Protokolle mit Ausnahme von ASCII die Parameter in einem DB abzulegen.

Kommunikation Mit SFCs steuern Sie die Kommunikation. Das Senden erfolgt unter Einsatz des SFC 217 (SER_SND) und das Empfangen über SFC 218 (SER_RCV).
 Durch erneuten Aufruf des SFC 217 SER_SND bekommen Sie bei 3964R, USS und Modbus über RetVal einen Rückgabewert geliefert, der unter anderem auch aktuelle Informationen über die Quittierung der Gegenseite beinhaltet.
 Bei den Protokollen USS und Modbus können Sie durch Aufruf des SFC 218 SER_RCV nach einem SER_SND das Quittungstelegramm auslesen.

Übersicht der SFCs für die serielle Kommunikation Folgende SFCs kommen für die serielle Kommunikation zum Einsatz:

SFC		Beschreibung
SFC 207	SER_CTRL	Modemfunktionalität
SFC 216	SER_CFG	RS232/RS485 Parametrieren
SFC 217	SER_SND	RS232/RS485 Senden
SFC 218	SER_RCV	RS232/RS485 Empfangen

11x	21x	31x	51x
✓	✓		

SFC 207 - SER_CTRL

Modem-funktionalität

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

FLOWCONTROL=0: Lesen: DTR, RTS, DSR, RI, CTS, CD
Schreiben: DTR, RTS

FLOWCONTROL>0: Lesen: DTR, RTS, DSR, RI, CTS, CD
Schreiben: nicht möglich

Parameter

Name	Deklaration	Typ	Beschreibung
WRITE	IN	BYTE	Bit 0: Neuer Zustand DTR Bit 1: Neuer Zustand RTS
MASKWRITE	IN	BYTE	Bit 0: Zustand an DTR übergeben Bit 1: Zustand an RTS übergeben
READ	OUT	BYTE	Status (CTS, DSR, RI, CD, DTR, RTS)
READDELTA	OUT	BYTE	Status Änderung seit letztem Zugriff
RETVAL	OUT	WORD	Rückgabewert (0 = OK)

WRITE

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

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

MASKWRITE

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

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

READ

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

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

RETVAL (Rückgabewert)

Wert	Beschreibung
0000h	kein Fehler
8x24h	Fehler in SFC-Parameter x, mit x: 1: Fehler in <i>WRITE</i> 2: Fehler in <i>MASKWRITE</i> 3: Fehler in <i>READ</i> 4: Fehler in <i>READDELTA</i>
809Ah	Schnittstelle ist nicht vorhanden
809Bh	Schnittstelle ist nicht konfiguriert (SFC 216)

11x	21x	31x	51x
✓	✓		

SFC 216 - SER_CFG

Beschreibung Die Parametrierung erfolgt zur Laufzeit unter Einsatz des SFC 216 SER_CFG. Hierbei sind die Parameter für STX/ETX, 3964R, USS und Modbus in einem DB abzulegen.

Bitte beachten Sie, dass nicht für alle Protokolle der gesamte Wertebereich der Parameter unterstützt wird. Näheres hierzu finden Sie direkt bei der Beschreibung des entsprechenden Parameters.



Hinweis!

Bitte beachten Sie, dass der SFC 216 während einer Kommunikation nicht mehr aufgerufen wird, da hierdurch alle Puffer gelöscht werden.

Sollen keine Kommunikations-Parameter mehr geändert werden, sollten Sie den Aufruf des SFC 216 in den Anlauf-OB 100 legen.

Parameter

Name	Deklaration	Typ	Beschreibung
PROTOCOL	IN	BYTE	Nummer für das Protokoll
PARAMETER	IN	ANY	Zeiger auf die Protokoll-Parameter
BAUDRATE	IN	BYTE	Nummer für die Baudrate
CHARLEN	IN	BYTE	Anzahl der Datenbits
PARITY	IN	BYTE	Parität
STOPBITS	IN	BYTE	Anzahl der Stopbits
FLOWCONTROL	IN	BYTE	Flusskontrolle (1 fix)
RETVL	OUT	WORD	Rückgabewert (0 = OK)

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

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

**PARAMETER
(als DB)**

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

Datenbaustein bei STX/ETX

DBB0:	STX1	BYTE	(1. Start-Zeichen in hexadezimaler Form)
DBB1:	STX2	BYTE	(2. Start-Zeichen in hexadezimaler Form)
DBB2:	ETX1	BYTE	(1. Ende-Zeichen in hexadezimaler Form)
DBB3:	ETX2	BYTE	(2. Ende-Zeichen in hexadezimaler Form)
DBW4:	TIMEOUT	WORD	(max. zeitlicher Abstand zwischen 2 Telegrammen im Zeitraster von 10ms)

**Hinweis!**

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

Tragen Sie immer für nicht benutzte Zeichen FFh ein!

Datenbaustein bei 3964R

DBB0:	Prio	BYTE	(Die Priorität beider Partner muss unterschiedlich sein. Prio 0 und 1 ist möglich)
DBB1:	ConnAttmptNr	BYTE	(Anzahl der Verbindungsaufbauversuche)
DBB2:	SendAttmptNr	BYTE	(Anzahl der Telegrammwiederholungen)
DBW4:	CharTimeout	WORD	(Zeichenverzugszeit in 10 ms Zeitraster)
DBW6:	ConfTimeout	WORD	(Quittungsverzugszeit in 10ms Zeitraster)

Datenbaustein bei USS

DBW0:	Timeout	WORD	(Verzugszeit in 10ms Zeitraster)
-------	---------	------	----------------------------------

Datenbaustein bei Modbus-Master

DBW0:	Timeout	WORD	(Antwort-Verzugszeit in 10ms Zeitraster)
-------	---------	------	--

Datenbaustein bei Modbus-Slave

DBB0:	Adresse	BYTE	(Adresse 1 ... 247 im Modbus-Netz)
DBW2:	Timeout	WORD	(Antwort-Verzugszeit in 10ms Zeitraster)

BAUDRATE Geschwindigkeit der Datenübertragung in Bit/s (Baud).
 01h: 150 Baud 05h: 1800 Baud 09h: 9600 Baud 0Dh: 57600 Baud
 02h: 300 Baud 06h: 2400 Baud 0Ah: 14400 Baud 0Eh: 115200 Baud
 03h: 600 Baud 07h: 4800 Baud 0Bh: 19200 Baud
 04h: 1200 Baud 08h: 7200 Baud 0Ch: 38400 Baud

CHARLEN Anzahl der Datenbits, auf die ein Zeichen abgebildet wird.
 0: 5Bit 1: 6Bit 2: 7Bit 3: 8Bit

Unterstützte Werte:

Bit	ASCII	STX/ETX	3964R	USS	Modbus RTU	Modbus ASCII
5	x		x			
6	x	x	x			
7	x	x	x			x
8	x	x	x	x	x	x

PARITY Die Parität ist je nach Wert gerade oder ungerade. Zur Paritätskontrolle werden die Informationsbits um das Paritätsbit erweitert, das durch seinen Wert ("0" oder "1") den Wert aller Bits auf einen vereinbarten Zustand ergänzt. Ist keine Parität vereinbart, wird das Paritätsbit auf "1" gesetzt, aber nicht ausgewertet.
 0: NONE 1: ODD 2: EVEN

STOPBITS Die Stopbits werden jedem zu übertragenden Zeichen nachgesetzt und kennzeichnen das Ende eines Zeichens.
 1: 1Bit 2: 1,5Bit 3: 2Bit
 Die 1,5Bit können ausschließlich bei einer *CHARLEN* von 5 verwendet werden, bei dieser Datenlänge sind 2Bit nicht möglich.

FLOWCONTROL Mit diesem Bit beeinflussen Sie das Verhalten der Request to send-Leitung.

- 0: RTS wird nicht beeinflusst
- 1: RTS ist "0" beim Senden (AutoRTS)
 RTS ist "1" beim Empfangen (AutoRTS)
- 2: HW Flow (nur bei ASCII Protokollen)



Hinweis!

Hinweis: Bei RS485 wird *FLOWCONTROL* nicht ausgewertet.
FLOWCONTROL = "1" (AutoRTS).

**RETVAL
(Rückgabewert)**

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

11x	21x	31x	51x
✓	✓		

SFC 217 - SER_SND

Beschreibung Mit diesem Baustein werden Daten über die serielle Schnittstelle gesendet.

Parameter

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

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

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

DATALEN Wort, in dem die Anzahl der gesendeten Bytes abgelegt wird.
 Bei **STX/ETX** und **3964R** wird immer die unter **DATAPTR** angegebene Länge oder 0 eingetragen.
 Wird unter **ASCII** sehr schnell gesendet, so dass nicht mehr alle Daten im 256Byte großen Sendepuffer eingetragen werden können, kann der Wert von der Sendelänge abweichen.

RetVal (Rückgabewert)

Wert	Beschreibung
0000h	Daten gesendet - fertig.
1000h	Nichts gesendet (Datenlänge 0).
20xxh	Protokoll wurde fehlerfrei ausgeführt mit xx-Bitmuster für Diagnose.
7001h	Daten liegen im internen Puffer - aktiv (busy).
7002h	Transfer - aktiv.
80xxh	Protokoll wurde fehlerhaft ausgeführt mit xx-Bitmuster für Diagnose (keine Quittung der Gegenseite).
90xxh	Protokoll wurde nicht ausgeführt mit xx-Bitmuster für Diagnose (keine Quittung der Gegenseite).
8x24h	Fehler in SFC-Parameter x, mit x: 1: Fehler in "DataPtr" 2: Fehler in "DataLen".
8122h	Fehler in Parameter "DataPtr" (z.B. DB zu kurz).
807Fh	Interner Fehler
809Ah	Schnittstelle nicht vorhanden.
809Bh	Schnittstelle nicht konfiguriert .

Protokollspezifische
RETVAL-Werte

ASCII

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

STX/ETX

Wert	Beschreibung
9000h	Pufferüberlauf (keine Daten gesendet)
9001h	Daten sind zu lang (>256Byte)
9002h	Daten sind zu kurz (0Byte)
9004h	Unzulässiges Zeichen

3964R

Wert	Beschreibung
2000h	Senden fertig ohne Fehler
80FFh	NAK empfangen - Fehler in der Kommunikation
80FEh	Datenübertragung ohne Quittierung der Gegenseite oder mit fehlerhafter Quittierung
9000h	Pufferüberlauf (keine Daten gesendet)
9001h	Daten sind zu lang (>256Byte)
9002h	Daten sind zu kurz (0Byte)

USS

Wert	Beschreibung
2000h	Senden fertig ohne Fehler
8080h	Empfangspuffer voll (kein Platz für Quittung)
8090h	Quittungsverzugszeit überschritten
80F0h	Falsche Checksumme in Rückantwort
80FEh	Falsches Startzeichen in der Rückantwort
80FFh	Falsche Slave-Adresse in der Rückantwort
9000h	Pufferüberlauf (keine Daten gesendet)
9001h	Daten sind zu lang (>256Byte)
9002h	Daten sind zu kurz (<2Byte)

Modbus RTU/ASCII Master

Wert	Beschreibung
2000h	Senden fertig ohne Fehler
2001h	Senden fertig mit Fehler
8080h	Empfangspuffer voll (kein Platz für Quittung)
8090h	Quittungsverzugszeit überschritten
80F0h	Falsche Checksumme in Rückantwort
80FDh	Länge der Rückantwort ist zu lang
80FEh	Falscher Funktionscode in der Rückantwort
80FFh	Falsche Slave-Adresse in der Rückantwort
9000h	Pufferüberlauf (keine Daten gesendet)
9001h	Daten sind zu lang (>256Byte)
9002h	Daten sind zu kurz (<2Byte)

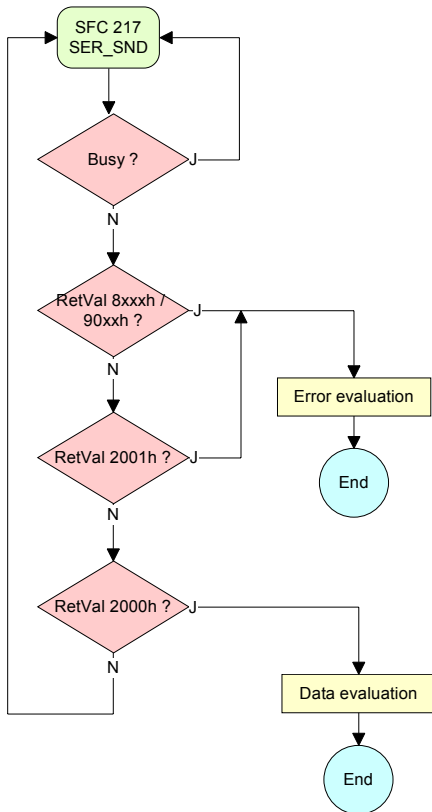
Modbus RTU/ASCII Slave

Wert	Beschreibung
0000h	Senden fertig ohne Fehler
9001h	Daten sind zu lang (>256Byte)

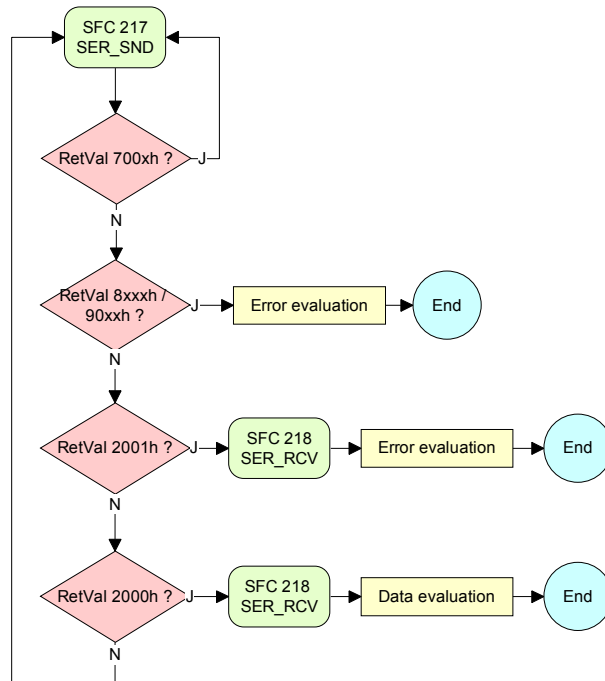
Prinzip der Programmierung

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

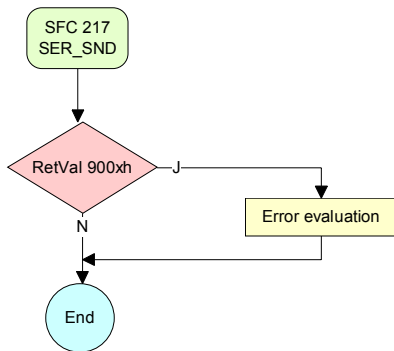
3964R



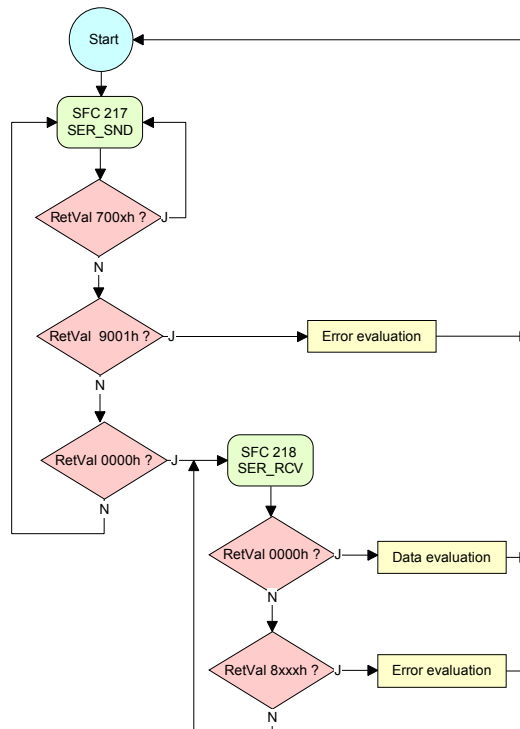
USS / Modbus-Master



ASCII / STX/ETX



Modbus Slave



11x	21x	31x	51x
✓	✓		

SFC 218 - SER_RCV

Beschreibung Mit diesem Baustein werden Daten über die serielle Schnittstelle empfangen.

Parameter

Name	Deklaration	Typ	Beschreibung
DATAPTR	IN	ANY	Zeiger auf Datenpuffer der Empfangsdaten
DATALEN	OUT	WORD	Länge der empfangenen Daten
ERROR	OUT	WORD	Fehlernummer
RETVAL	OUT	WORD	Rückgabewert (0 = OK)

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

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

DATALEN Wort, in dem die Anzahl der empfangenen Bytes abgelegt wird.
 Bei **STX/ETX** und **3964R** wird immer die Länge der empfangenen Nutzdaten oder 0 eingetragen.

Unter **ASCII** wird hier die Anzahl der gelesenen Zeichen eingetragen. Dieser Wert kann von der gelesenen Telegrammlänge abweichen.

ERROR In diesem Wort erfolgt ein Eintrag im Fehlerfall unter ASCII. Folgende Fehlermeldungen können generiert werden:

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

RETVAL
(Rückgabewert)

Wert	Beschreibung
0000h	kein Fehler
1000h	Empfangspuffer ist zu klein (Datenverlust).
8x24h	Fehler in SFC-Parameter x, mit x: 1: Fehler in <i>DATAPTR</i> 2: Fehler in <i>DATALEN</i> 3: Fehler in <i>ERROR</i>
8122h	Fehler in Parameter <i>DATA</i> von SFC 216 (z.B. kein DB).
809Ah	Schnittstelle nicht vorhanden.
809Bh	Schnittstelle ist nicht konfiguriert.

11x	21x	31x	51x
✓	✓	✓	✓

SFC 219 - CAN_TLGR - CAN-Telegramm senden

SFC 219 CAN_TLGR SDO-Anforderung an CAN-Master

Mittels dieses Bausteins 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

Name	Deklaration	Typ	Beschreibung
REQUEST	IN	BOOL	Auftrag anstoßen
SLOT_MASTER	IN	BYTE	Steckplatz des Masters
NODEID	IN	BYTE	CAN-Adresse
TRANSFERTYP	IN	BYTE	Transfertype
INDEX	IN	DWORD	CANopen Index
SUBINDEX	IN	DWORD	CANopen Subindex
CANOPENERROR	OUT	DWORD	Fehlernummer
RETVAl	OUT	WORD	Rückgabewert
BUSY	OUT	BOOL	Auftragsstatus
DATABUFFER	IN_OUT	ANY	Datenbereich

REQUEST Steuerparameter: 1: Anstoß des Auftrags

SLOT_MASTER System 100V: Steckplatz 0: 21x-2CM02
Steckplatz 1 ... 4: 208-1CA00

System 200V: Steckplatz 0: 21x-2CM02
Steckplatz 1 ... 32: 208-1CA00

NODEID 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

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
0x00000000	Service erfolgreich
0x05030000	Toggle-Bit nicht geändert
0x05040000	SDO Protokoll Time-out
0x05040001	Client/Server Befehlsspezifizierung nicht gültig oder unbekannt
0x05040002	Ungültige Blockgröße (nur Block-Modus)
0x05040003	Ungültige Sequenznummer (nur Block-Modus)
0x05040004	CRC Fehler (nur Block-Modus)
0x05040005	Unzureichender Speicher
0x06010000	Nicht unterstützter Zugriff auf ein Objekt
0x06010001	Lesezugriff auf ein Nur-Schreiben-Objekt
0x06010002	Schreibzugriff auf ein Nur-Lesen-Objekt
0x06020000	Objekt nicht im Objektverzeichnis vorhanden
0x06040041	Objekt kann nicht ins PDO gemappt werden
0x06040042	Anzahl und Länge der zu mappenden Objekte überschreitet PDO-Länge
0x06040043	Generelle Parameterinkompatibilität
0x06040047	Generelle interne Inkompatibilität im Gerät
0x06060000	Zugriffsfehler wegen Hardwareausfall
0x06070010	Datentyp nicht korrekt, Länge der Serviceparameter nicht korrekt
0x06070012	Datentyp nicht korrekt, Serviceparameter zu lang
0x06070013	Datentyp nicht korrekt, Serviceparameter zu kurz
0x06090011	Subindex existiert nicht
0x06090030	Wertebereich der Parameter überschritten (nur für Schreibzugriff)
0x06090031	Zu schreibender Parameterwert ist zu hoch
0x06090032	Zu schreibender Parameterwert ist zu niedrig
0x06090036	Maximumwert ist kleiner als Minimumwert
0x08000000	Genereller Fehler
0x08000020	Die Daten können entweder nicht transferiert oder nicht in der SPS gespeichert werden.
0x08000021	Die Daten können wegen lokaler Kontrollen entweder nicht transferiert oder nicht in der SPS gespeichert werden.
0x08000022	Die Daten können wegen aktuellem Modulstatus entweder nicht transferiert oder nicht in der SPS gespeichert werden.
0x08000023	Dynamische Objektverzeichnisgenerierung fehlgeschlagen oder kein Objektverzeichnis gefunden (z.B. Objektverzeichnis wird aus Datei generiert und ein Dateifehler ist aufgetreten).

RETVAL (Rückgabewert) 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 Fehlercodes.

Wert	Beschreibung
0xF021	Ungültige Slave-Adresse (Aufrufparameter gleich 0 oder größer 127).
0xF022	Ungültiger Transfertyp (Wert ungleich 60h, 61h).
0xF023	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).
0xF024	Der SFC wird nicht unterstützt.
0xF025	Schreibpuffer im CANopen-Master ist voll, Service kann zur Zeit nicht bearbeitet werden.
0xF026	Lesebuffer im CANopen-Master ist voll, Service kann zur Zeit nicht bearbeitet werden.
0xF027	Der SDO-Lese- oder Schreibzugriff wurde fehlerhaft beantwortet, siehe CANopen Error Codes.
0xF028	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 SFC kommuniziert.
SDO-Lesezugriff: Zielbereich für die gelesenen Nutzdaten.
SDO-Schreibzugriff: Quellbereich für die zu schreibenden Nutzdaten.



Hinweis!

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

11x	21x	31x	51x
✓	✓	✓	✓

MMC-Zugriff - SFC 220...222

Übersicht

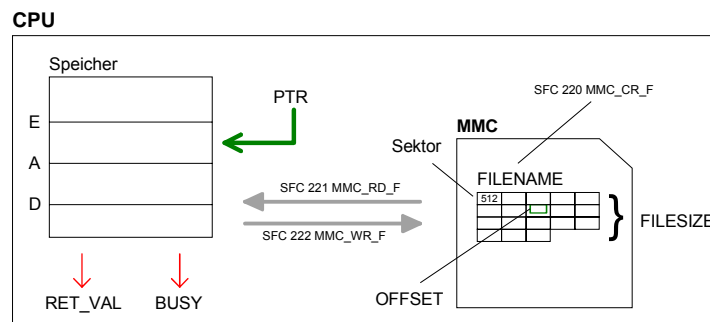
Mit den hier aufgeführten SFCs haben Sie die Möglichkeit den Zugriff auf eine MMC in Ihr Anwenderprogramm einzubinden. Hierbei können Sie bei einer gesteckten MMC eine neue Datei anlegen bzw. eine bestehende Datei für den Zugriff öffnen. Solange Sie keine neue Datei öffnen haben Sie über Lese-/Schreib-Befehle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



Hinweis!

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

11x	21x	31x	51x
✓	✓	✓	✓

SFC 220 - MMC_CR_F

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.



Hinweis!

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



Hinweis!

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.

Folgende Diagnose-/Fehlermeldungen können ausgegeben werden:

Wert	Beschreibung
<i>Diagnosemeldungen</i>	
0000h	keine Fehler (tritt nur beim Erzeugen einer neuen Datei auf).
0001h	Datei existiert schon, ist unfragmentiert und die Längenangabe <i>FILESIZE</i> ist identisch oder kleiner als die reale Dateigröße.
8001h	Es ist keine oder eine vom Typ unbekannte MMC gesteckt.
<i>Fehlermeldungen</i>	
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 <i>FILESIZE</i> 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.

11x	21x	31x	51x
✓	✓	✓	✓

SFC 221 - MMC_RD_F

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 finden Sie in der Beschreibung des SFC 220.

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.

Folgende Diagnose-/Fehlermeldungen können ausgegeben werden:

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.

11x	21x	31x	51x
✓	✓	✓	✓

SFC 222 - MMC_WR_F

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 finden Sie in der Beschreibung des SFC 220.

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 Wort, in das eine Diagnose-/Fehlermeldung zurückgeliefert wird.

(Rückgabewert) 0 bedeutet, dass alles OK ist.

Folgende Diagnose-/Fehlermeldungen können ausgegeben werden:

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.

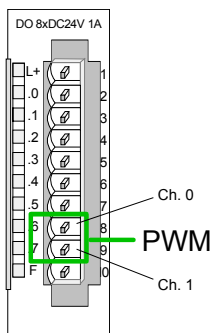
11x	21x	31x	51x
✓			

SFC 223 - PWM - Pulsweitenmodulation

Beschreibung Dieser Baustein dient zur Parametrierung der Pulsweitenmodulation für die letzten beiden Ausgabe-Kanäle von X5.

Parameter

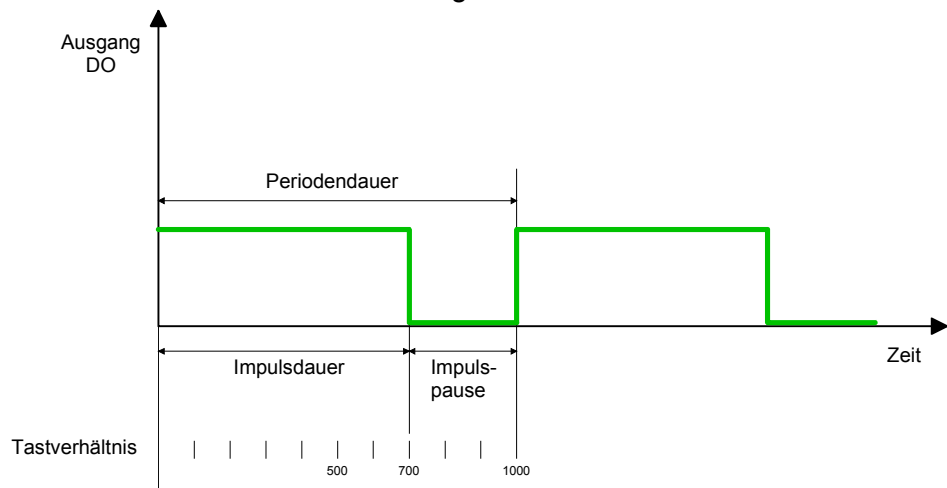
Name	Deklaration	Typ	Beschreibung
CHANNEL	IN	INT	Nummer des Ausgabekanals für PWM
ENABLE	IN	BOOL	Auftrag anstoßen
TIMEBASE	IN	INT	Zeitbasis
PERIOD	IN	DINT	Periode der PWM
DUTY	IN	DINT	Tastverhältnis in Promille
MINLEN	IN	DINT	Minimale Impulsdauer
RET_VAL	OUT	WORD	Rückgabewert (0 = OK)



Sie geben Zeitbasis, Periode, Tastverhältnis und minimale Impulsdauer vor. Hieraus ermittelt die CPU eine Impulsfolge mit entsprechendem Impuls/Pausenverhältnis und gibt dies über den entsprechenden Ausgabe-Kanal aus.

Der SFC liefert einen Fehlercode zurück. Die entsprechenden Fehlermeldung finden Sie in der Tabelle auf der nächsten Seite.

Die PWM-Parameter stehen in folgendem Verhältnis:



$$\begin{aligned} \text{Periodendauer} &= \text{Zeitbasis} \times \text{Periode} \\ \text{Impulsdauer} &= (\text{Periodendauer} / 1000) \times \text{Tastverhältnis} \\ \text{Impulspause} &= \text{Periodendauer} - \text{Impulsdauer} \end{aligned}$$

Die Parameter haben folgende Bedeutung:

Channel Geben Sie hier den Ausgabe-Kanal an, den Sie ansteuern möchten.
Wertebereich: 0 ... 1

- ENABLE** Über diesen Parameter können Sie die PWM-Funktion aktivieren (true) bzw. deaktivieren (false).
Wertebereich: true, false
- TIMEBASE** *TIMEBASE* bedeutet "Zeitbasis" über die Sie die Auflösung und den Wertebereich der Impuls-, Perioden- und Mindestimpulsdauer je Kanal bestimmen.
Eingestellt können die Werte 0 für 0,1ms und 1 für 1ms.
Wertebereich: 0 ... 1
- PERIOD** Durch Multiplikation des unter *PERIOD* vorgegebenen Werts mit der *TIMEBASE* erhalten Sie die Periodendauer.
Wertebereich: 0 ... 60000
- DUTY** Mit diesem Parameter geben Sie das Tastverhältnis in Promille an. Hiermit bestimmen Sie, bezogen auf eine Periode, das Verhältnis zwischen Impulsdauer und Impulspause.
1 Promille = 1 *TIMEBASE*
Ist die errechnete Impulsdauer kein Vielfaches von *TIMEBASE*, wird auf die nächst kleinere *TIMEBASE*-Grenze abgerundet.
Wertebereich: 0 ... 1000
- MINLEN** Über *MINLEN* bestimmen Sie die minimale Impulsdauer. Schalthandlungen werden nur dann durchgeführt, wenn der Impuls die hier eingestellte minimale Zeitdauer überschreitet.
Wertebereich: 0 ... 60000
- RET_VAL (Rückgabewert)** Über den Parameter *RET_VAL* bekommen Sie eine Fehler-Nr. zurückgeliefert. 0 bedeutet, dass alles OK ist. Die entsprechende Fehlermeldung entnehmen Sie bitte der nachfolgenden Tabelle:

Wert	Bedeutung
0000h	Kein Fehler
8005h	Parameter <i>MINLEN</i> außerhalb der zulässigen Grenzen.
8006h	Parameter <i>DUTY</i> außerhalb der zulässigen Grenzen.
8007h	Parameter <i>PERIOD</i> außerhalb der zulässigen Grenzen.
8008h	Parameter <i>TIMEBASE</i> außerhalb der zulässigen Grenzen.
8009h	Parameter <i>CHANNEL</i> außerhalb der zulässigen Grenzen.
9001h	Interner Fehler: Für einen Parameter konnte keine gültige Adresse zugeordnet werden.
9002h	Interner Hardwarefehler: Setzen Sie sich bitte mit dem VIPA-Service in Verbindung.
9003h	Ausgang ist nicht als PWM-Ausgang parametrierbar bzw. Hardware-Konfiguration ist fehlerhaft.
9004h	HF-PWM wurde parametrierbar aber SFC 223 wurde aufgerufen (bitte SFC 225 HF_PWM verwenden!).

11x	21x	31x	51x
✓			

SFC 224 - HSC - High-speed-Counter

Beschreibung Dieser SFC dient zur Parametrierung der Zählfunktionen (high speed counter) für die ersten 4 Eingänge.

Parameter

Name	Deklaration	Typ	Beschreibung
CHANNEL	IN	INT	Nummer des Eingabekanals für HSC
ENABLE	IN	BOOL	Auftrag anstoßen
DIRECTION	IN	INT	Zählrichtung
PRESETVALUE	IN	DINT	Ladewert
LIMIT	IN	DINT	Zählgrenze
RET_VAL	OUT	WORD	Rückgabewert (0 = OK)
SETCOUNTER	IN_OUT	BOOL	Ladewert laden

CHANNEL Geben Sie hier den Eingabe-Kanal an, den Sie als Zähler aktivieren möchten.

Wertebereich: 0 ... 3

ENABLE Über diesen Parameter können Sie die Zählerfunktion aktivieren (true) bzw. deaktivieren (false).

Wertebereich: true, false

DIRECTION Bestimmen sie mit *DIRECTION* die Zählrichtung.

Hierbei bedeuten: 0: Zähler ist deaktiviert, entspricht *ENABLE* = false
1: hochzählen
2: runterzählen

PRESETVALUE Hiermit können sie einen Zählerinhalt vorgeben, der über *SETCOUNTER* = true in den entsprechenden Zähler transferiert wird.

Wertebereich: 0 ... FFFFFFFFh

LIMIT Über Limit geben Sie eine obere bzw. untere Grenze an, für die Zählrichtung Auf- bzw. Abwärts. Bei Erreichen der Grenze wird der entsprechende Zähler auf 0 gestellt und neu gestartet; ggf. erfolgt eine Alarmausgabe.

Wertebereich: 0 ... FFFFFFFFh

**RET_VAL
(Rückgabewert)**

Über den Parameter *RET_VAL* bekommen Sie eine Fehler-Nr. zurückgeliefert. Die entsprechende Fehlermeldung entnehmen Sie bitte der nachfolgenden Tabelle:

Wert	Bedeutung
0000h	Kein Fehler
8002h	Der ausgewählte Kanal ist nicht als Zähler konfiguriert (Fehler in Hardware-Konfiguration).
8008h	Parameter <i>DIRECTION</i> außerhalb der zulässigen Grenzen.
8009h	Parameter <i>CHANNEL</i> außerhalb der zulässigen Grenzen.
9001h	Interner Fehler Für einen Parameter konnte keine gültige Adresse zugeordnet werden.
9002h	Interner Hardwarefehler Setzen Sie sich bitte mit dem VIPA-Service in Verbindung.

SETCOUNTER

Durch *SETCOUNTER* = true wird der mit *PRESETVALUE* übergebene Wert in den entsprechenden Zähler übertragen.

Das Bit wird vom SFC wieder zurückgesetzt.

Wertebereich: true, false

11x	21x	31x	51x
✓			

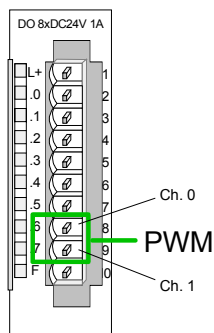
SFC 225 - HF_PWM - HF Pulsweitenmodulation

Beschreibung

Dieser Baustein dient zur Parametrierung der Pulsweitenmodulation für die letzten beiden Ausgabe-Kanäle. Dieser Baustein hat die gleiche Funktion wie der SFC 223. Anstelle von *TIMEBASE* und *PERIODE* geben Sie hier eine Frequenz (bis zu 50kHz) vor.

Parameter

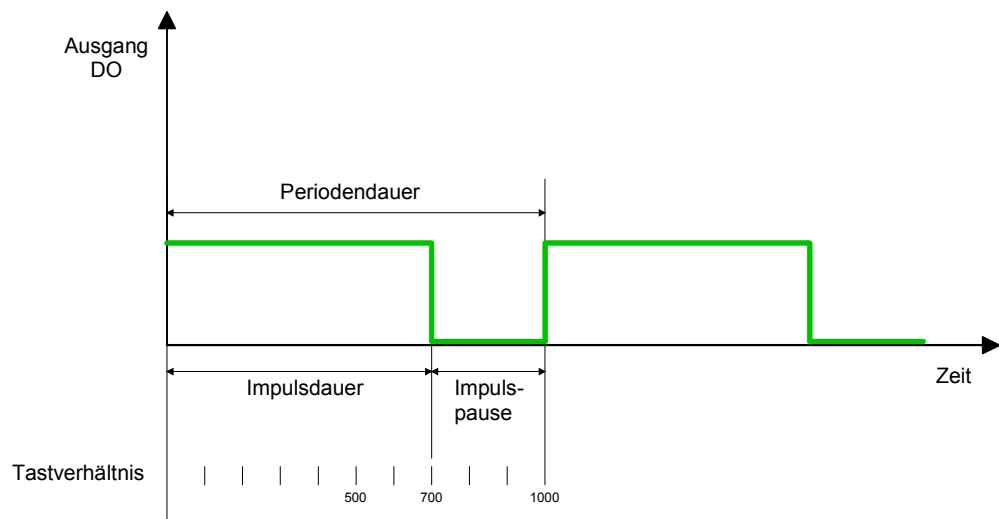
Name	Deklaration	Typ	Beschreibung
CHANNEL	IN	INT	Nummer des Ausgabekanals für HF-PWM
ENABLE	IN	BOOL	Auftrag anstoßen
FREQUENCE	IN	WORD	Frequenz der HF-PWM
DUTY	IN	DINT	Tastverhältnis in Promille
MINLEN	IN	DINT	Minimale Impulsdauer
RET_VAL	OUT	WORD	Rückgabewert (0 = OK)



Sie geben Frequenz, Tastverhältnis und minimale Impulsdauer vor. Hieraus ermittelt die CPU eine Impulsfolge mit entsprechendem Impuls/Pausenverhältnis und gibt dies über den entsprechenden Ausgabe-Kanal aus.

Der SFC liefert einen Fehlercode zurück. Die entsprechenden Fehlermeldungen finden Sie in der Tabelle auf der nächsten Seite.

Die PWM-Parameter stehen in folgendem Verhältnis:



$$\text{Periodendauer} = 1 / \text{Frequenz}$$

$$\text{Impulsdauer} = (\text{Periodendauer} / 1000) \times \text{Tastverhältnis}$$

$$\text{Impulspause} = \text{Periodendauer} - \text{Impulsdauer}$$

- CHANNEL** Geben Sie hier den Ausgabe-Kanal an, den Sie ansteuern möchten.
Wertebereich: 0 ... 1
- ENABLE** Über diesen Parameter können Sie die PWM-Funktion aktivieren (true) bzw. deaktivieren (false).
Wertebereich: true, false
- FREQUENCY** Geben Sie hier die Frequenz in Hz als hexadezimalen Wert an.
Wertebereich: 09C4h ... C350h (2,5kHz ... 50kHz)
- DUTY** Mit diesem Parameter geben Sie das Tastverhältnis in Promille an. Hiermit bestimmen Sie, bezogen auf eine Periode, das Verhältnis zwischen Impulsdauer und Impulspause.
1 Promille = 1 Timebase
Ist die errechnete Impulsdauer kein Vielfaches von Timebase, wird auf die nächst kleinere Timebase-Grenze abgerundet.
Wertebereich: 0 ... 1000
- MINLEN** Über *MINLEN* bestimmen Sie die minimale Impulsdauer in μ s. Schaltungen werden nur dann durchgeführt, wenn der Impuls die hier eingestellte minimale Zeitdauer überschreitet.
Wertebereich: 0 ... 60000
- RET_VAL (Rückgabewert)** Über den Parameter *RET_VAL* bekommen Sie eine Fehler-Nr. zurückgeliefert. Die entsprechende Fehlermeldung entnehmen Sie bitte der nachfolgenden Tabelle:

Wert	Beschreibung
0000h	Kein Fehler
8005h	Parameter <i>MINLEN</i> außerhalb der zulässigen Grenzen
8006h	Parameter <i>DUTY</i> außerhalb der zulässigen Grenzen
8007h	Parameter <i>FREQUENCY</i> außerhalb der zulässigen Grenzen
8009h	Parameter <i>CHANNEL</i> außerhalb der zulässigen Grenzen
9001h	Interner Fehler: Für einen Parameter konnte keine gültige Adresse zugeordnet werden.
9002h	Interner Hardwarefehler: Setzen Sie sich bitte mit dem VIPA-Service in Verbindung.
9003h	Ausgang ist nicht als PWM-Ausgang parametrierbar bzw. Hardware-Konfiguration ist fehlerhaft.
9004h	PWM wurde parametrierbar aber SFC 225 für HF-PWM wurde aufgerufen (bitte SFC 223 PWM verwenden!).

11x	21x	31x	51x
✓	✓	✓	✓

SFC 227 - TD_PRM - TD200-Kommunikation

Beschreibung Der SFC 227 dient zur Ankopplung des Terminals TD200 von Siemens an die VIPA CPUs.

Bitte beachten Sie, dass zum Einsatz des Terminals vor Aufruf des SFC 227 ein Datenbaustein mit den TD200-Konfigurationsdaten vorliegen muss. Diesen Datenbaustein können Sie mit dem TDWizard von VIPA erzeugen. Im Datenbaustein werden neben den allgemeinen Einstellungen wie Sprache und Display-Modus auch die Meldetexte abgelegt, die Sie komfortabel mit dem TDWizard von VIPA eingeben können. TDWizard ist Bestandteil von WinPLC7 und ist bei VIPA erhältlich.

Parameter Durch Aufruf des SFC 227 spezifizieren Sie das Terminal, mit dem kommuniziert werden soll. Beim Aufruf des SFC sind folgende Parameter zu übergeben:

Name	Deklaration	Typ	Beschreibung
MPI_ADDR	IN	BYTE	MPI Adresse des TD200
TD_STRUCT_PTR	IN	ANY	Zeiger zum Anfang der TD-Daten
FUNC_KEY_PTR	IN	ANY	Zeiger auf Statusbyte der Funktionstasten
OFFSET_INPUT	IN	BYTE	Offset des Eingabebereichs
OFFSET_OUTPUT	IN	BYTE	Offset des Ausgabebereichs
RET_VAL	OUT	BYTE	Rückgabewert (0 = OK)

MPI_ADR MPI-Adresse
Geben Sie hier die MPI-Adresse des angeschlossenen TD200 Terminals an.
Parameterart: Byte

TD_STRUCT_PTR Terminal Strukture Pointer
Zeigt auf den Anfang des Datenbaustein, der die Parametrierung und die Textblöcke des Terminals beinhaltet.
Den Datenbaustein können Sie mit dem TDWizard erstellen. Dieses Tool liegt der Software WinPLC7 bei.
Parameterart: Zeiger
Sinnvoller Bereich: DB

FUNC_KEY_PTR Function Key Pointer
Zeigt auf den Bereich, in dem das Status-Byte für gedrückte Tasten abliegt.
Parameterart: Zeiger
Sinnvoller Bereich: DB, M

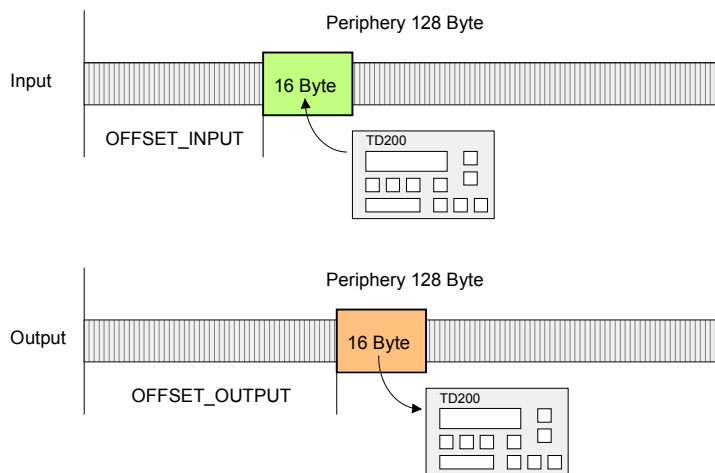
OFFSET_INPUT
OFFSET_OUTPUT

Offset Input, Offset Output

Mit dem Terminal haben Sie die Möglichkeit Ein- und Ausgabe-Byte zu setzen. Das TD200 von Siemens unterstützt für Ein- und Ausgabe lediglich einen Bereich von 16Byte.

Bei Einsatz der CPUs von VIPA haben Sie Zugriff auf das gesamte Prozessabbild (je 128Byte für Ein- und Ausgänge).

Durch Angabe eines Offset können Sie für die Ein- und Ausgabe ein 16Byte großes "Fenster" über den 128Byte großen Peripheriebereich legen. Folgende Abbildung soll dies verdeutlichen:



RET_VAL
(Rückgabewert)

Geben Sie hier das Merker-Byte an, in dem die aus dem Aufruf resultierenden Meldung abgelegt wird.

Eine Auflistung der (Fehler-)Meldungen finden Sie weiter unten.

Meldungen

Wert	Bedeutung
00h	kein Fehler aufgetreten
10h	Fehler in Angabe von <i>MPI_ADDR</i>
11h	<i>MPI_ADDR</i> beinhaltet MPI-Adresse der CPU
12h	Angabe in <i>MPI_ADDR</i> ist größer als max. MPI-Adresse
20h	Fehler in <i>OFFSET_INPUT</i>
21h	Angabe in <i>OFFSET_INPUT</i> ist zu groß
30h	Fehler in <i>OFFSET_OUTPUT</i>
31h	Angabe in <i>OFFSET_OUTPUT</i> ist zu groß
40h	Fehler in <i>TD_STRUCT_PTR</i>
41h	<i>TD_STRUCT_PTR</i> zeigt auf keinen DB
50h	Fehler in <i>FUNC_KEY_PTR</i>
51h	Fehler in <i>FUNC_KEY_PTR</i>
52h	<i>FUNC_KEY_PTR</i> zeigt nicht auf einen I-, A- oder M-Bereich
60h	Ein interner Fehler ist aufgetreten

11x	21x	31x	51x
	✓	✓	✓

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.



Hinweis!

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.

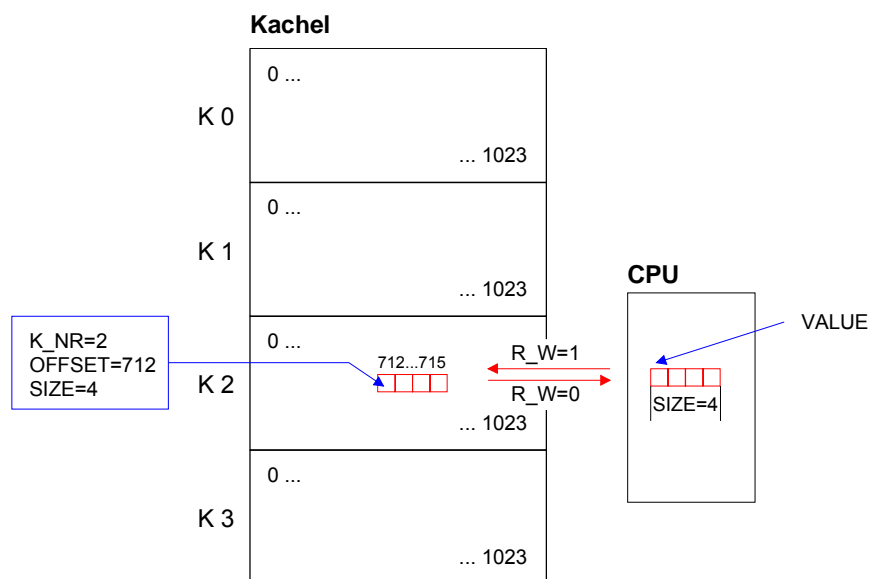
Bei einem Schreibzugriff werden aus diesem Bereich die bis zu 4Byte breiten Daten in den Kachelbereich übertragen.

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 \neq 1, 2 oder 4 beim Lesezugriff
08h	der Parameter SIZE \neq 1, 2 oder 4 beim Schreibzugriff
09h	der Parameter R_W ist \neq 0 oder 1

11x	21x	31x	51x
	✓	✓	✓

Kachelkommunikation - SFC 230 ... 238

Übersicht

Durch die mitgelieferten 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

Parameter- beschreibung

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

SSNR:	Schnittstellenummer
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

Eine Beschreibung dieser Parameter finden Sie auf den Folgeseiten.

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 <i>QANF</i> zeigt. 0: <i>QANF</i> zeigt direkt auf den Datenanfang der Quell- bzw. Zieldaten. 1: der Zeiger <i>QANF/ZANF</i> zeigt auf eine Speicherzelle, ab der die Quell- bzw. Zieldaten definiert sind (indirekt). 2: der Zeiger <i>QANF/ZANF</i> zeigt auf einen Speicherbereich in dem sich die Quell- und Zielangaben befinden (indirekt). 5: der Zeiger <i>QANF/ZANF</i> zeigt auf eine Speicherzelle, ab der die Quell- bzw. Zielparameter und Parameter zum Anzeigenwort definiert sind (indirekt). 6: der Zeiger <i>QANF/ZANF</i> 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

**Hinweis!**

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 Blockgröße = 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:

<i>Wert</i>	<i>Blockgröße</i>	<i>Wert</i>	<i>Blockgröße</i>
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

Kachelkommunikation - Parameterübergabe

direkte/indirekte Parametrierung

Ein Hantierungsbaustein kann direkt oder indirekt parametrierung 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 parametrierung werden.

Beispiel direkte und indirekte Parameterübergabe

Direkte Parameterübergabe

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

Indirekte Parameterübergabe

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

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

Die direkte bzw. indirekte Übergabe der Quell und Zielparameter finden Sie auf den Folgeseiten.

Kachelkommunikation - 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 Beschreibung	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 C P#DB10.DBX 0.0 BYTE 8	P#M b.0 BYTE c P#M 5.0 BYTE 10	P#A b.0 BYTE c P#A 0.0 BYTE 2	P#E b.0 BYTE c P#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 ge- schrieben 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	Länge des Quell-/ Ziel- Datenblocks in Worten.	Länge des Quell-/ Ziel-Datenblocks in Bytes.	Länge des Quell-/ Ziel-Datenblocks in Bytes.	Länge des Quell-/ Ziel-Datenblocks in Bytes.
erlaubter Bereich für "c"	1 ... 2048	1 ... 255	1 ... 128	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 Beschreibung	IND = 1	IND = 2																										
Bedeutung	<p>Indirekte Adressierung für Quell- oder Zielparameter. Die Quell- oder Zielparameter werden in einem DB hinterlegt.</p> <p>QANF/ZANF</p> <table border="1"> <tr> <td>DW +0</td> <td>Datentyp Quelle</td> </tr> <tr> <td>+2</td> <td>DB-Nr. bei Typ "DB" ansonsten irrelevant</td> </tr> <tr> <td>+4</td> <td>Anfangsadresse</td> </tr> <tr> <td>+6</td> <td>Länge in Byte</td> </tr> </table>	DW +0	Datentyp Quelle	+2	DB-Nr. bei Typ "DB" ansonsten irrelevant	+4	Anfangsadresse	+6	Länge in Byte	<p>Indirekte Adressierung für Quell- und Zielparameter. Die Quell- und Zielparameter werden hintereinander in einem DB hinterlegt.</p> <p>QANF/ZANF</p> <table border="1"> <tr> <td>DW +0</td> <td>Datentyp Quelle</td> <td rowspan="4">Beschreibung Datenquelle</td> </tr> <tr> <td>+2</td> <td>DB-Nr. bei Typ "DB" ansonsten irrelevant</td> </tr> <tr> <td>+4</td> <td>Anfangsadresse</td> </tr> <tr> <td>+6</td> <td>Länge in Byte</td> </tr> <tr> <td>+8</td> <td>Datentyp Ziel</td> <td rowspan="4">Beschreibung Datenziel</td> </tr> <tr> <td>+10</td> <td>DB-Nr. bei Typ "DB" ansonsten irrelevant</td> </tr> <tr> <td>+12</td> <td>Anfangsadresse</td> </tr> <tr> <td>+14</td> <td>Länge in Byte</td> </tr> </table>	DW +0	Datentyp Quelle	Beschreibung Datenquelle	+2	DB-Nr. bei Typ "DB" ansonsten irrelevant	+4	Anfangsadresse	+6	Länge in Byte	+8	Datentyp Ziel	Beschreibung Datenziel	+10	DB-Nr. bei Typ "DB" ansonsten irrelevant	+12	Anfangsadresse	+14	Länge in Byte
DW +0	Datentyp Quelle																											
+2	DB-Nr. bei Typ "DB" ansonsten irrelevant																											
+4	Anfangsadresse																											
+6	Länge in Byte																											
DW +0	Datentyp Quelle	Beschreibung Datenquelle																										
+2	DB-Nr. bei Typ "DB" ansonsten irrelevant																											
+4	Anfangsadresse																											
+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																										
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 Anzeigewort einen Bereich angeben (*IND* = 5) oder für Datenquelle, Datenziel und Anzeigewort jeweils getrennte Bereiche bestimmen (*IND* = 6).

In der nachfolgenden Tabelle finden Sie mögliche *QANF* / *ZANF*-Parameter bei der indirekten Parametrierung:

QTYP/ZTYP Beschreibung	IND = 5	IND = 6																																													
Bedeutung	<p>Indirekte Adressierung für Quell- oder Zielparameter und Anzeigewort. Die Quell- oder Ziel und ANZW-Parameter werden hintereinander in einem DB hinterlegt.</p> <p>QANF/ZANF</p> <table border="1"> <tr> <td>DW +0</td> <td>Datentyp Quelle</td> <td rowspan="4">Beschreibung Datenquelle/-ziel</td> </tr> <tr> <td>+2</td> <td>DB-Nr. bei Typ "DB" ansonsten irrelevant</td> </tr> <tr> <td>+4</td> <td>Anfangsadresse</td> </tr> <tr> <td>+6</td> <td>Länge in Byte</td> </tr> <tr> <td>+8</td> <td>Datentyp Quelle</td> <td rowspan="4">Beschreibung Anzeigewort</td> </tr> <tr> <td>+10</td> <td>DB-Nr. bei Typ "DB" ansonsten irrelevant</td> </tr> <tr> <td>+12</td> <td>Anfangsadresse</td> </tr> <tr> <td></td> <td></td> </tr> </table>	DW +0	Datentyp Quelle	Beschreibung Datenquelle/-ziel	+2	DB-Nr. bei Typ "DB" ansonsten irrelevant	+4	Anfangsadresse	+6	Länge in Byte	+8	Datentyp Quelle	Beschreibung Anzeigewort	+10	DB-Nr. bei Typ "DB" ansonsten irrelevant	+12	Anfangsadresse			<p>Indirekte Adressierung für Quell- und Zielparameter und Anzeigewort. Die Quell-, Ziel- und ANZW-Parameter werden hintereinander in einem DB hinterlegt.</p> <p>QANF/ZANF</p> <table border="1"> <tr> <td>DW +0</td> <td>Datentyp Quelle</td> <td rowspan="4">Beschreibung Datenquelle</td> </tr> <tr> <td>+2</td> <td>DB-Nr. bei Typ "DB" ansonsten irrelevant</td> </tr> <tr> <td>+4</td> <td>Anfangsadresse</td> </tr> <tr> <td>+6</td> <td>Länge in Byte</td> </tr> <tr> <td>+8</td> <td>Datentyp Ziel</td> <td rowspan="4">Beschreibung Datenziel</td> </tr> <tr> <td>+10</td> <td>DB-Nr. bei Typ "DB" ansonsten irrelevant</td> </tr> <tr> <td>+12</td> <td>Anfangsadresse</td> </tr> <tr> <td>+14</td> <td>Länge in Byte</td> </tr> <tr> <td>+16</td> <td>Datentyp Quelle</td> <td rowspan="4">Beschreibung Anzeigewort</td> </tr> <tr> <td>+18</td> <td>DB-Nr. bei Typ "DB" ansonsten irrelevant</td> </tr> <tr> <td>+20</td> <td>Anfangsadresse</td> </tr> <tr> <td></td> <td></td> </tr> </table>	DW +0	Datentyp Quelle	Beschreibung Datenquelle	+2	DB-Nr. bei Typ "DB" ansonsten irrelevant	+4	Anfangsadresse	+6	Länge in Byte	+8	Datentyp Ziel	Beschreibung Datenziel	+10	DB-Nr. bei Typ "DB" ansonsten irrelevant	+12	Anfangsadresse	+14	Länge in Byte	+16	Datentyp Quelle	Beschreibung Anzeigewort	+18	DB-Nr. bei Typ "DB" ansonsten irrelevant	+20	Anfangsadresse		
DW +0	Datentyp Quelle	Beschreibung Datenquelle/-ziel																																													
+2	DB-Nr. bei Typ "DB" ansonsten irrelevant																																														
+4	Anfangsadresse																																														
+6	Länge in Byte																																														
+8	Datentyp Quelle	Beschreibung Anzeigewort																																													
+10	DB-Nr. bei Typ "DB" ansonsten irrelevant																																														
+12	Anfangsadresse																																														
DW +0	Datentyp Quelle	Beschreibung Datenquelle																																													
+2	DB-Nr. bei Typ "DB" ansonsten irrelevant																																														
+4	Anfangsadresse																																														
+6	Länge in Byte																																														
+8	Datentyp Ziel	Beschreibung Datenziel																																													
+10	DB-Nr. bei Typ "DB" ansonsten irrelevant																																														
+12	Anfangsadresse																																														
+14	Länge in Byte																																														
+16	Datentyp Quelle	Beschreibung Anzeigewort																																													
+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																																													
Länge Bedeutung	Länge des DBs in Byte	Länge des DBs in Byte																																													
erlaubter Bereich	14 fix	22 fix																																													

Kachelkommunikation - 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	<i>Statusverwaltung CPU</i> 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 <i>Datenverwaltung Hantierungsbaustein</i> Bit 4: Datenübernahme/-übergabe läuft Bit 5: Datenübergabe erfolgt Bit 6: Datenübernahme erfolgt Bit 7: Disable/Enable Datenblock 0: freigegeben 1: gesperrt
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 parametrier.

2 Bereich im AG nicht vorhanden

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

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 parametrierete 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 READ/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 Buskabels 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 Umladefunktion

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

F Auftrag nicht vorhanden

Der angesprochene Auftrag ist nicht auf dem CP parametrierbar. 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 PRIO 1 oder RECEIVE-Anstoß möglich bei PRIO 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 PRIO 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 (DMA-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 PRIO 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 Triggerung) 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 Triggerung) 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 Datenverkehr 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
Wichtige Status- und Fehleranzeigen der CPU

Im Folgenden sind wichtige Status- und Fehlermeldungen 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.

Mögliche Anzeigenworte

Anzeigenwort: 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).

Anzeigenwort: X A X A

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

Anzeigenwort: X 0 X 8

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

Anzeigenwort: X 0 X 9

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

Anzeigenwort: X 0 2 4

Der SEND ist ohne Fehler abgearbeitet worden, die Daten wurden übertragen.

Anzeigenwort: X 0 4 5

Der RECEIVE ist ohne Fehler abgearbeitet worden, die Daten sind auf dem AG angekommen.

Anzeigenwort: 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 Anzeigenwort-zustände

In den folgenden Tabelle sind die wichtigsten Anzeigenwortzustände aufgeführt:

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	

Kachelkommunikation - Parametrierfehler PAFE

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 bis 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

11x	21x	31x	51x
	✓	✓	✓

SFC 230 - SEND

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	Schnittstellenummer
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.

11x	21x	31x	51x
	✓	✓	✓

SFC 231 - RECEIVE

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 Anzeigewort 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	Anzeigewort

Im "Leerlauf" des Bausteins wird nur das Anzeigewort 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 Empfangstelegramms benutzt und die Daten über die "Hintergrundkommunikation" der CPU übergibt.

11x	21x	31x	51x
	✓	✓	✓

SFC 232 - FETCH

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 in *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	Schnittstellenummer
ANR	IN	INT	Auftragsnummer
IND	IN	INT	Adressierungsmodus
ZANF	IN	ANY	Zeiger auf Datenziel
PAFE	OUT	BYTE	Parametrierungsfehler
ANZW	IN_OUT	DWORD	Anzeigenwort



Hinweis!

Nähere Angaben zur indirekten Parametrierung finden Sie in diesem Kapitel unter "**Kachelkommunikation - Quell- bzw. Zielangaben**".

11x	21x	31x	51x
	✓	✓	✓

SFC 233 - CONTROL

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 parametrisierten Anzeigenwort. Der Baustein ist nicht VKE abhängig und sollte im zyklischen Teil des Programms aufgerufen werden.

Parameter

Name	Deklaration	Typ	Beschreibung
SSNR	IN	INT	Schnittstellenummer
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.

11x	21x	31x	51x
	✓	✓	✓

SFC 234 - RESET

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 rückgesetzt.

Der Baustein arbeitet VKE-abhängig und kann von zyklischen, zeitgesteuerten oder alarmgesteuerten Programmteilen aus aufgerufen werden.

Parameter

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

Betriebsarten

Der Baustein kennt folgende beiden Betriebsarten:

- RESET ALL
- RESET DIREKT

11x	21x	31x	51x
	✓	✓	✓

SFC 235 - SYNCHRON

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 Blockgröße = 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

11x	21x	31x	51x
	✓	✓	✓

SFC 236 - SEND_ALL

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	Schnittstellenummer
PAFE	OUT	BYTE	Parametrierfehler
ANZW	IN_OUT	DWORD	Anzeigenwort

ANZW

Im Baustein-Anzeigenwort, dem Anzeigenwort, das im SEND_ALL-Baustein parametrierbar 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.



Hinweis!

In folgenden Fällen ist mindestens einmal SEND_ALL im Zyklus-Baustein OB1 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.

11x	21x	31x	51x
	✓	✓	✓

SFC 237 - RECEIVE_ALL

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



Hinweis!

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.

11x	21x	31x	51x
	✓	✓	✓

SFC 238 - CTRL1

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 parametrisierten 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 $ANR \neq 0$ wird das Anzeigenwort in der gleichen Weise aufgebaut und bearbeitet wie bei allen anderen "Hantierungsbausteinen".

Wird der Parameter *ANR* mit 0 versorgt, überträgt der 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.

