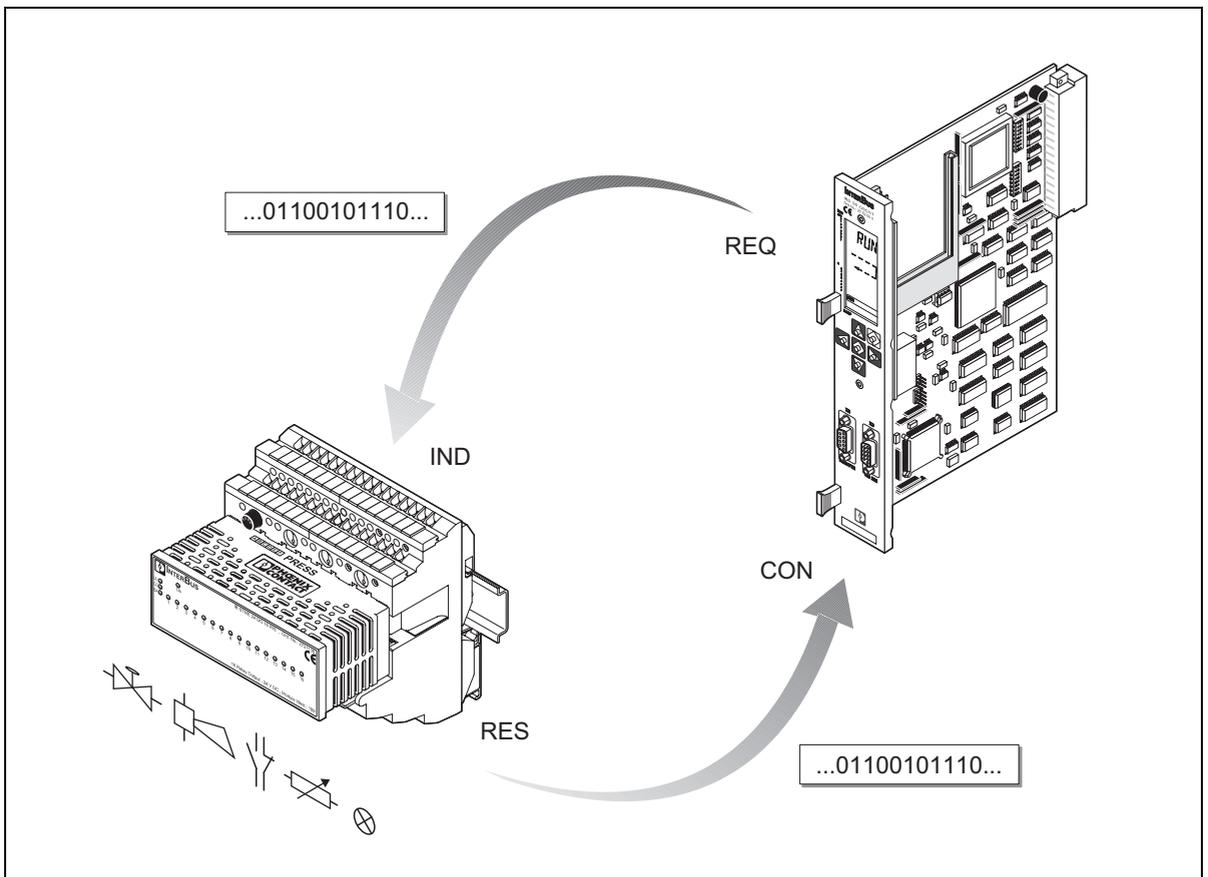


Anwenderhandbuch
Dienste und Fehlermeldungen
der Firmware

Bezeichnung: IBS SYS FW G4 UM
Art.-Nr.: 27 45 13 0



Dienste und Fehlermeldungen der Firmware

Bezeichnung: IBS SYS FW G4 UM

Revision: D

Art.-Nr.: 27 45 13 0

Dieses Handbuch ist gültig für:
Anschaltbaugruppen der Generation 4 ab Firmware 4.60

© Phoenix Contact 12/2002

5150D



Bitte beachten Sie folgende Hinweise:

Damit Sie Ihr Gerät sicher einsetzen können, sollten Sie dieses Handbuch aufmerksam lesen und beachten. Die folgenden Hinweise geben Ihnen eine erste Orientierung zum Gebrauch des Handbuchs.

Einschränkung der Anwendergruppe

Der in diesem Handbuch beschriebene Produktgebrauch richtet sich ausschließlich an Elektrofachkräfte oder von Elektrofachkräften unterwiesene Personen, die mit den geltenden nationalen Normen vertraut sind. Für Fehlhandlungen und Schäden, die an Produkten von Phoenix Contact und Fremdprodukten durch Missachtung der Informationen dieses Handbuchs entstehen, übernimmt Phoenix Contact keine Haftung.

Erklärungen zu den verwendeten Symbolen



Das Symbol *Achtung* bezieht sich auf Handlungen, die einen Schaden der Hard- oder Software oder Personenschaden (im indirekten Zusammenhang mit gefährlicher Prozessperipherie) zur Folge haben können.



Das Symbol *Hinweis* vermittelt Bedingungen, die für einen fehlerfreien Betrieb unbedingt beachtet werden müssen. Es weist außerdem auf Tipps und Ratschläge für den effizienten Geräteinsatz und die Software-Optimierung hin, um Ihnen Mehrarbeit zu ersparen.



Das Symbol *Text* verweist Sie auf weiterführende Informationsquellen (Handbücher, Datenblätter, Literatur etc.) zu dem angesprochenen Thema, Produkt o.ä. Ebenso gibt dieser Text hilfreiche Hinweise zur Orientierung im Handbuch.

Ihre Meinung interessiert uns

Wir sind ständig bemüht, die Qualität unserer Handbücher zu verbessern.

Sollten Sie Anregungen und Verbesserungsvorschläge zu Inhalt und Gestaltung unseres Handbuchs haben, würden wir uns freuen, wenn Sie uns Ihre Vorschläge zusenden. Verwenden Sie dazu bitte den universellen Telefax-Vordruck am Ende dieses Handbuchs.

Erklärungen zu den rechtlichen Grundlagen

Dieses Handbuch ist einschließlich aller darin enthaltenen Abbildungen urheberrechtlich geschützt. Jede Drittverwendung dieses Handbuchs, die von den urheberrechtlichen Bestimmungen abweicht, ist verboten. Die Re-

produktion, Übersetzung sowie die elektronische und fotografische Archivierung und Veränderung bedarf der schriftlichen Genehmigung der Firma Phoenix Contact. Zuwiderhandlungen verpflichten zu Schadenersatz.

Phoenix Contact behält sich jegliche Änderungen, die dem technischen Fortschritt dienen, vor.

Alle Rechte für den Fall der Patenterteilung oder Gebrauchsmustereintragung sind Phoenix Contact vorbehalten. Fremdprodukte werden stets ohne Vermerk auf Patentrechte genannt. Die Existenz solcher Rechte ist daher nicht auszuschließen.

Internet

Aktuelle Informationen zu Produkten von Phoenix Contact finden Sie im Internet unter www.phoenixcontact.com.

Inhaltsverzeichnis

1	Einleitung und Übersicht.....	1-3
1.1	Einführung.....	1-3
1.2	Unterschiede zwischen Firmware Generation 3.x und 4.x.....	1-5
1.3	Leistungsmerkmale ab Firmware 4.6x	1-10
1.4	Allgemeine Informationen	1-12
1.5	Funktionen zur Konfigurationsrahmen-Verwaltung.....	1-45
1.6	Funktionen zur Prozessdatenverwaltung.....	1-48
1.7	Funktionen zur Fehler- und Diagnose-Verwaltung.....	1-55
1.8	Funktionen zur Bussteuerung	1-57
1.9	Allgemeine Funktionen der Firmware	1-60
2	Dienste der Firmware	2-5
2.1	Übersicht	2-5
2.2	Hinweise zu den Dienstbeschreibungen	2-8
2.3	Dienste zum Parametrieren der Anschaltbaugruppe	2-11
2.4	Dienste zum Definieren von Prozessdatenbeschreibungen	2-68
2.5	Dienste zum Zuordnen von Prozessdaten	2-80
2.6	Dienste für den direkten Zugriff auf den INTERBUS.....	2-98
2.7	Dienste zur Diagnose.....	2-111
2.8	Dienste zum Anlegen von Funktionen	2-133
2.9	Dienste für den Parametrierungsspeicher.....	2-174
2.10	Selbstständige Fehleranzeigen der Anschaltbaugruppe.....	2-204
3	Fehler-Codes	3-3
3.1	Fehler-Codes zu Anwender-Fehlern (USER FAIL)	3-3
3.2	Fehler-Codes zu allgemeinen Busfehlern (BUS FAIL).....	3-55
3.3	Fehler-Codes zu Fernbus- und Lokalbusfehlern.....	3-67
3.4	Fehler-Codes zu Systemfehlern.....	3-87

3.5	Fehler-Codes zu Anschaltbaugruppen mit Slaveteil	3-91
3.6	Fehler-Codes zu Anschaltbaugruppen mit COP-Karte	3-93

Dieses Kapitel informiert Sie über

- Leistungsmerkmale der Firmware 4.6x
- Unterschiede zwischen den Firmware-Generationen G3 und G4
- Grundlagen der Kommunikation zwischen PCP-Teilnehmern und Anschaltbaugruppen
- Funktionen zur Konfigurationsrahmen-, Prozessdaten- und Fehler-/Diagnose-Verwaltung

Einleitung und Übersicht.....	1-3
1.1 Einführung.....	1-3
1.2 Unterschiede zwischen Firmware Generation 3.x und 4.x.....	1-5
1.2.1 Leistungsmerkmale.....	1-5
1.2.2 Anlaufverhalten.....	1-7
1.2.3 Gegenüberstellung von Diensten der Firmware- Generationen 3.x und 4.x.....	1-8
1.3 Leistungsmerkmale ab Firmware 4.6x.....	1-10
1.3.1 Rückwirkungsfreies Schalten im Falle eines Busfehlers.....	1-10
1.3.2 Dienste zum Parametrierungsspeicher-Handling.....	1-11
1.4 Allgemeine Informationen.....	1-12
1.4.1 Kommunikation mit der Anschaltbaugruppe.....	1-12
1.4.2 Standardregister.....	1-15
1.4.3 Zustandsmaschine.....	1-22
1.4.4 Befehlsgruppen der Firmware.....	1-25
1.4.5 Konfigurationsrahmen.....	1-27
1.4.6 Prozessdaten.....	1-39
1.4.7 Zustandsbeschreibung während der Konfiguration.....	1-43
1.5 Funktionen zur Konfigurationsrahmen-Verwaltung.....	1-45
1.5.1 Automatisches Laden der Konfiguration.....	1-45
1.5.2 Manuelles Erzeugen einer optimierten Konfiguration ..	1-46
1.6 Funktionen zur Prozessdatenverwaltung.....	1-48
1.6.1 Prozessdatenobjekte eines Teilnehmers definieren	1-48
1.6.2 Definition der Prozessdatenbeschreibungslisten.....	1-50
1.6.3 Projektieren der Prozessdatenzuordnung.....	1-52

1.7	Funktionen zur Fehler- und Diagnoseverwaltung	1-55
1.8	Funktionen zur Bussteuerung	1-57
1.8.1	Dienste zur Systemzustandssteuerung	1-57
1.8.2	Dienste zum Schalten von Gruppen und Alternativen .	1-59
1.9	Allgemeine Funktionen der Firmware	1-60
1.9.1	Umgang mit dem Parametrierungsspeicher	1-60
1.9.2	Verhalten der Firmware bei bestimmten Anschaltbaugruppen-Kommandos.....	1-61

1 Einleitung und Übersicht

1.1 Einführung

Dieses Handbuch beschreibt den direkten Zugriff auf INTERBUS-Anschaltbaugruppen der Generation 4 über die Firmware-Kommandos.



In dieser Revision des Firmware-Handbuches wird auf die Leistungsmerkmale der aktuellen **Firmware-Revision 4.6x** eingegangen. Die Informationen finden Sie im Kapitel „**Leistungsmerkmale ab Firmware 4.6x**“ auf **Seite 1-10**.



Die Firmware der Generation 4 – im Folgenden kurz mit G4 gekennzeichnet – wurde entwickelt, um den Anforderungen an einen Systembus für die nächsten Jahre zu entsprechen. Dazu wurden u. a. die Dienste neu geordnet und die Zustandsmaschine verändert. Auf Firmware 3.x aufsetzende Programme können Generation 4-Baugruppen nicht mehr betreiben.



Nähere Informationen zu den Unterschieden zwischen den Firmware-Diensten der Generation 3 (G3) und der Generation 4 (G4) entnehmen Sie dem Kapitel 1.2.

Die Firmware-Änderung betrifft die Programmierung des Masters. Alle INTERBUS-Teilnehmer können weiterhin am Bussystem betrieben werden. Ausnahme hiervon sind 8-Leiter-Fernbus-Teilnehmer.

Zu den Möglichkeiten, die die Generation 4-Firmware bietet, gehören

- die logische Adressierung der Prozessdaten einzelner Teilnehmer über vordefinierte Prozessdatenbeschreibungen (PDD),
- die Schachtelung der Bustopologie auf bis zu 16 Ebenen,
- die Unterstützung von Loop-Teilnehmern,
- verschiedene Synchronmodi zwischen Host und Anschaltbaugruppe inklusive einer definierten Buszykluszeit,
- die Projektierung von Durchgangsdaten,
- die Projektierung von Prozessdatenvorverarbeitung,
- eine ereignisgesteuerte Funktionsaktivierung,
- die Verwaltung und Parametrierung alternativer oder wechselnder Busteilnehmer und
- eine verbesserte Diagnose für jeden einzelnen Teilnehmer.

Die Firmware der Generation 4 bietet damit eine flexible, für alle Anschaltbaugruppen fast identische Schnittstelle.

Wie in einem verteilten Schieberegister durchlaufen die zu übertragenden Daten beim INTERBUS-Protokoll jeden einzelnen Teilnehmer. An der für ihn reservierten Position kann jeder Teilnehmer seine Eingangsdaten (z. B. Eingangsparameter, Sollwerte, Stellwerte, ...) auslesen und seine Ausgangsdaten (Ausgangsparameter, Istwerte, ...) direkt hineinkopieren. Es werden zwei Verfahren unterschieden:

- Der Prozessdatenkanal für den zyklischen E/A-Datenaustausch.
- Der Parameterkanal für die Übertragung längerer Datensätze.

Die Übertragung über den Parameterkanal übernimmt das Peripherals Communication Protocol (PCP). Die Dienste dieses Protokolls sind dafür ausgelegt, lange Parametersätze und Programme zu übertragen sowie Programme über Befehle zu kontrollieren. Hierfür werden von der Firmware die Dienste in kleine Pakete zerlegt und bei jedem Buszyklus Stück für Stück in das Schieberegister kopiert. Der Zielteilnehmer setzt diese Informationen wieder zu einem Dienst zusammen.



Weitergehende Informationen entnehmen Sie bitte dem Anwenderhandbuch Peripherals Communication Protocol (PCP) *IBS SYS PCP G4 UM* (Artikel-Nr. 27 45 11 4).

Prozessdaten- und Parameterkanal greifen auf getrennte Bereiche im Summenrahmen des INTERBUS zu. Durch den hybriden Aufbau des Summenrahmens werden auch bei starker Beanspruchung des Parameterkanals die Buszykluszeiten nicht beeinflusst.

Alle Dienste zur Steuerung der gesicherten Datenübertragung über den Prozessdatenkanal werden in diesem Handbuch beschrieben.

Kapitel 1 gibt eine kurze Einführung in die Mechanismen der Firmware. Der Anwender soll in die Lage versetzt werden, die einzelnen Dienste den speziellen Aufgabenbereichen zuzuordnen. Es enthält:

- Einen Überblick über die Schnittstellen zur Firmware,
- Eine Beschreibung der Zustandsmaschine,
- Eine Erklärung der Teilnehmernummern,
- Eine Erklärung der Adressierung,
- Die wichtigsten Dienstreihenfolgen für die Konfiguration,
- Die Bussteuerung (exemplarisch),
- Die Fehlerbehandlung (exemplarisch).

- Kapitel 2 dient als Nachschlagewerk für die Aufrufparameter aller Dienste, d. h. der Requests (Dienstanforderungen) und der zugehörigen Confirmations (Dienstquittungen).
- Kapitel 3 beinhaltet die sortierte Aufstellung aller Firmware-Fehlermeldungen.

IBS CMD G4

Das Programm IBS CMD G4 erleichtert die Projektierung Ihrer Anlage.

IBS CMD G4 ist eine Software, die eine interaktive und steuerungsunabhängige Projektierung, Bedienung und Diagnose aller angeschlossenen Geräte in einem INTERBUS-Netzwerk ermöglicht.

IBS CMD G4 läuft auf Standard PCs unter MS Windows und kann für eine Vielzahl von INTERBUS-Anschaltbaugruppen eingesetzt werden. Die Verbindung zu einer Anschaltbaugruppe kann auf drei Kommunikationswegen erfolgen:

- Über eine serielle Schnittstelle (RS-232-Pegel) mit der Diagnose-Schnittstelle der Anschaltbaugruppe,
- Über den jeweiligen Steckplatz bei PC-Anschaltbaugruppen, z. B. ISA-Bus oder
- Über Ethernet bei einigen Field Controllern.

1.2 Unterschiede zwischen Firmware Generation 3.x und 4.x

1.2.1 Leistungsmerkmale

Firmware 3.x

- Unstrukturierte Kommando- und Meldungs-Codes
- Auf ein Kommando können eine, zwei oder keine Meldung folgen.
- Am gesetzten Bit 15 des Meldungs-Codes ist erkennbar, dass Parameter folgen.

Firmware 4.x

- Strukturierte Dienst-Codes (Request- und Confirmation-Codes)

- Auf jeden Request (Dienstanforderung) erfolgt eine Confirmation (Dienstquittung), Ausnahmen sind: Reset und unquittierte PCP-Dienste.
- In der binären Darstellungsform des Codes eines Requests enthält das Bit 15 immer den Wert „0“. Der Code, der auf diesen Request folgenden Confirmation, ist bis auf das Bit 15 identisch. Das Bit 15 der Confirmation hat immer den Wert „1“. Hieraus folgt, dass in hexadezimaler Darstellung der Code einer Confirmation immer um 8000_{hex} größer ist als der Code des Requests, auf den die Confirmation erfolgt.

Beispiel:

Request: Start_Data_Transfer_Request (0701_{hex})

Confirmation: Start_Data_Transfer_Confirmation (8701_{hex})
 $8701_{\text{hex}} = 0701_{\text{hex}} + 8000_{\text{hex}}$

- Alle Dienste enthalten einen Parameterzähler (*Parameter_Count*), der angibt, wie viele Parameter folgen.
- Positive und negative Confirmations haben denselben Meldungs-Code; der Parameter *Result* gibt Auskunft, ob der Dienst erfolgreich bearbeitet werden konnte.

1.2.2 Anlaufverhalten

Firmware 3.x

Eine Anschaltbaugruppe der Generation 3.x liest nach dem Power-On-Selbsttest (POST) generell die angeschlossene Anlagenkonfiguration (Busaufbau) ein und legt sie als sogenannte „Urkonfiguration“ ab. Diese ist mit dem Dienst „Send_Physical_Configuration“ aus der G3-Anschaltbaugruppe auslesbar. Der Anwender kann jetzt die (physikalisch) vorhandene Konfiguration mit seiner geplanten Konfiguration vergleichen.

Ein Großteil der Anwendungen übergibt mit dem Dienst „Check_Physical_Configuration“ die geplante Konfiguration der G3-Anschaltbaugruppe. Die Quittung dieses Dienstes meldet Abweichungen von der tatsächlich vorhandenen Konfiguration.

Firmware 4.x

Nach erfolgreichem Power-On-Selbsttest (POST) befindet sich die G4-Anschaltbaugruppe im Zustand „Ready“ (RDY). Ein automatisches Einlesen der vorhandenen Konfiguration gibt es nicht.

Die bevorzugte Vorgehensweise bei der Projektierung einer Anlage ist, zuerst eine Anlagenkonfiguration zu erstellen. Diese Konfiguration wird mit dem Dienst „Complete_Load_Configuration“ (vgl. Seite 2-54) auf die Anschaltbaugruppe geladen. Beim Aktivieren der geladenen Konfiguration mit dem Dienst „Activate_Configuration“ (vgl. Seite 2-64) wird die Übereinstimmung mit der tatsächlich angeschlossenen Konfiguration durch die Anschaltbaugruppe geprüft. Die Confirmation meldet Abweichungen von der tatsächlich vorhandenen Konfiguration.

Daneben ist es möglich, die angeschlossene Anlagenkonfiguration mit dem Dienst „Create_Configuration“ (vgl. Seite 2-62) in die G4-Anschaltbaugruppe einzulesen. Der Anwender kann jetzt diese Anlagenkonfiguration mit dem Dienst „Read_Configuration“ (vgl. Seite 2-38) aus der G4-Anschaltbaugruppe auslesen und mit der geplanten Konfiguration vergleichen.

1.2.3 Gegenüberstellung von Diensten der Firmware-Generationen 3.x und 4.x



Für Anschaltbaugruppen der Generation 4 benutzen Sie zum Stoppen der INTERBUS-Datenzyklen den Dienst „Alarm_Stop“ (G4). Mit diesem Dienst werden gleichzeitig die Ausgänge zurückgesetzt. Der Dienst „Stop_Data_Transfer“ (G4) stoppt nur die Datenzyklen.

Für Anschaltbaugruppen der Generation 3 haben Sie dafür den Dienst „Stop_INTERBUS“ (G3) benutzt. Die Ausgänge wurden nicht zurückgesetzt.



Die einander gegenüberstehenden G3- und G4-Dienste unterscheiden sich teilweise in ihrer Funktionalität. Schlagen Sie die Funktionalität der G4-Dienste im Kapitel 2 „Dienste der Firmware“ nach.

Tabelle 1-1 Gegenüberstellung von G3-Diensten und G4-Diensten

Firmware-Generation 3		Firmware-Generation 4	
Dienst (G3)		Dienst (G4)	
Request	Confirmation (positiv / negativ)	Request	Confirmation
Start_INTERBUS		Start_Data_Transfer	
0001 _{hex}	0088 _{hex} / 00E3 _{hex}	0701 _{hex}	8701 _{hex}
Alarm_Stop		Alarm_Stop	
004A _{hex}	00D8 _{hex} / —	1303 _{hex}	9303 _{hex}
Stop_INTERBUS		Stop_Data_Transfer	
0002 _{hex}	00C6 _{hex} / —	0702 _{hex}	8702 _{hex}
Configure_INTERBUS		Create_Configuration	
0023 _{hex}	00CA _{hex} / —	0710 _{hex}	8710 _{hex}
Warm_Start		Reset_Controller_Board	
004C _{hex}	— / —	0956 _{hex}	—
Clear_Display		Confirm_Diagnostics	
004E _{hex}	00E2 _{hex} / —	0760 _{hex}	8760 _{hex}
Receive_CRL		Load_CRL_Attributes_Loc	
0053 _{hex}	00E8 _{hex} / 80F1 _{hex}	0264 _{hex}	8264 _{hex}
Init_Communication		nicht erforderlich !	
0054 _{hex}	00E9 _{hex} / 80F0 _{hex}		

Tabelle 1-1 Gegenüberstellung von G3-Diensten und G4-Diensten

Firmware-Generation 3		Firmware-Generation 4	
Dienst (G3)		Dienst (G4)	
Request	Confirmation (positiv / negativ)	Request	Confirmation
Check_Physical_Configuration		Initiate_Load_Configuration	
0058 _{hex}	00AB _{hex} / 0068 _{hex}	0306 _{hex}	8306 _{hex}
		Complete_Load_Configuration	
		030A _{hex}	830A _{hex}
		Terminate_Load_Configuration	
		0308 _{hex}	8308 _{hex}
		Activate_Configuration	
		0711 _{hex}	8711 _{hex}
Send_Localbus_Module_Error		Read_Device_State	
005B _{hex}	80EE _{hex} / 00ED _{hex}	0315 _{hex}	8315 _{hex}
Send_Physical_Configuration		Complete_Read_Configuration	
005E _{hex}	80F4 _{hex} / 004A _{hex}	030B _{hex}	830B _{hex}
Quit_Module_Error_All		Control_Device_Function	
0065 _{hex}	00FE _{hex} / 80FF _{hex}	0714 _{hex}	8714 _{hex}
Read		Read	
0081 _{hex}	8181 _{hex}	0081 _{hex}	8081 _{hex}
Write		Write	
0082 _{hex}	8182 _{hex}	0082 _{hex}	8082 _{hex}
Start		Start	
0083 _{hex}	8183 _{hex}	0083 _{hex}	8083 _{hex}
Stop		Stop	
0084 _{hex}	8184 _{hex}	0084 _{hex}	8084 _{hex}
Get_OD		Get_OD	
0088 _{hex}	8188 _{hex}	0088 _{hex}	8088 _{hex}
Initiate		Initiate	
008B _{hex}	818B _{hex} / 818C _{hex}	008B _{hex}	808B _{hex}
Abort		Abort	
008D _{hex}	—	088D _{hex}	—

1.3 Leistungsmerkmale ab Firmware 4.6x

Die folgenden Firmware-Funktionalitäten stehen ab Version 4.6x zur Verfügung.



Diese Funktionalitäten sind abhängig von der eingesetzten Anschaltbaugruppe.

- Das rückwirkungsfreie Schalten von Teilnehmern/Bussegmenten im Falle eines Busfehlers ist implementiert (siehe „Rückwirkungsfreies Schalten im Falle eines Busfehlers“ auf Seite 1-10).
- Die Firmware unterstützt bis zu 8192 E/A-Punkte (bei gleicher Anzahl Teilnehmer wie bisher), abhängig von der eingesetzten Anschaltbaugruppe.
- Bis zu 126 PCP-Teilnehmer (127 KR-Einträge) können angesprochen werden.
- Die Funktionalitäten der Produktfamilien Rugged Line und Inline, insbesondere Einzelkanal-Fehlermeldungen und optische Diagnose, stehen mit diesem Systemschritt auch für die Übertragungsraten 2 Mbaud zur Verfügung.
- Optimiertes Zu- und Abschalten von Schnittstellen
Die INTERBUS-Controller der Generation 4 besitzen die Funktionalität, die abzweigende Lokalbus- und die weiterführende Fernbus-Schnittstelle und, soweit vorhanden, die Fernbus-Stichschnittstelle abzuschalten.
- Parametrierungsspeicher-Handling
Die Dienste zum Umgang mit dem Parametrierungsspeicher wurden erweitert. Lesen Sie näheres im Kapitel „Dienste zum Parametrierungsspeicher-Handling“ auf Seite 1-11.

1.3.1 Rückwirkungsfreies Schalten im Falle eines Busfehlers

Zusätzlich zur bisherigen Möglichkeit Teilnehmer/Segmente durch die Applikation und/oder im Fehlerfall durch das Steuerungsprogramm (durch ein vom Anwender definiertes Kommando) zu schalten, bietet die Firmware 4.6 die Funktionalität an, abzweigende INTERBUS-Fernbus-Stich- und -Lokalbus-Schnittstellen rückwirkungsfrei abzuschalten. Die Datenübertragung im übrigen INTERBUS wird dabei nicht unterbrochen.

Das rückwirkungsfreie Abschalten wird durch einen aufgetretenen Busfehler ausgelöst. Das rückwirkungsfreie Zuschalten als Reaktion auf das rückwirkungsfrei Abschalten ist abhängig von der Applikation. Das Steuerungsprogramm schaltet das Bussegment nach Behebung und Quittierung des Fehlers wieder zu.

Voraussetzungen für das rückwirkungsfreie Schalten:

- Das eingesetzte System ist ein reines SUPI-3-System.
- Die Busklemme, an der rückwirkungsfrei geschaltet werden soll, enthält einen SUPI-3-OPC.
- Nur abzweigende Fernbus-Stich- und Lokalbus-Schnittstellen sind rückwirkungsfrei abschaltbar.



Eine Fehlerlokalisierung ist innerhalb des rückwirkungsfrei abgeschalteten Bussegmentes nicht möglich.

Parametrieren Sie das rückwirkungsfreie Schalten über den Parameter *Fehlercharakteristik* im Konfigurationsrahmen.



Weitere Hinweise zum rückwirkungsfreien Schalten finden Sie im Datenblatt DB D IBS SYS SCHALTEN.

1.3.2 Dienste zum Parametrierungsspeicher-Handling

Aufgrund steigender Projektgrößen müssen zunehmend größere Parametrierungsspeicherkarten eingesetzt werden. Das Handling dieser großen Parametrierungsspeicher (16/32/64 MB) unterscheidet sich im Wesentlichen nicht vom Handling der kleineren Speicher (1/2/4 MB). Der Unterschied liegt in der erhöhten Zeitdauer bestimmter Vorgänge, wie z. B. dem Formatieren der Karte und dem Löschen von Daten auf dem Parametrierungsspeicher. Zur Optimierung des Handlings wurden Firmware-Dienste ergänzt und neu programmiert.

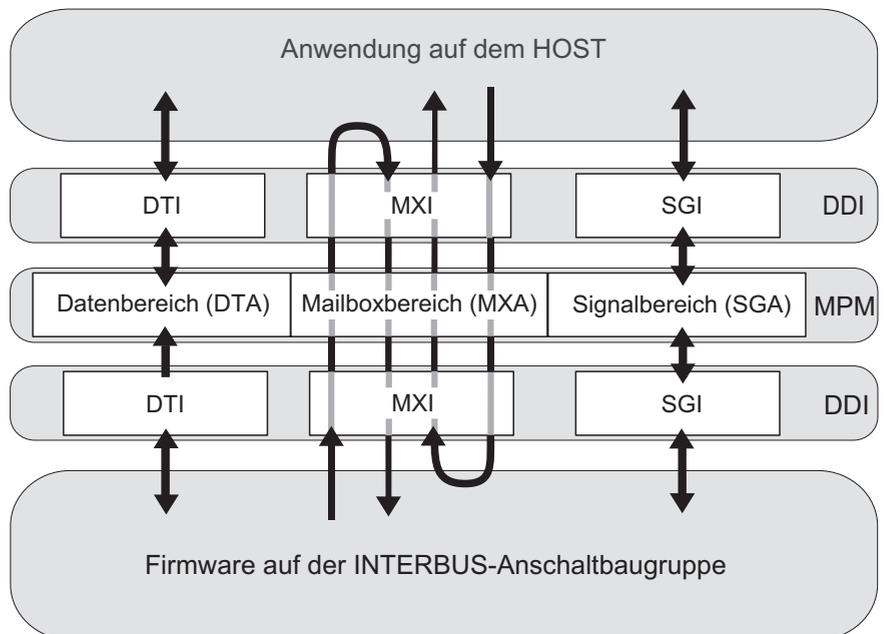


Weitere Informationen finden Sie in den Kapiteln „Umgang mit dem Parametrierungsspeicher“ auf Seite 1-60 und „Dienste für den Parametrierungsspeicher“ auf Seite 2-174.

1.4 Allgemeine Informationen

1.4.1 Kommunikation mit der Anschaltbaugruppe

Alle Informationen zwischen einem Host und der Anschaltbaugruppe werden über einen gemeinsam genutzten Speicherbereich geführt. Dieser Speicherbereich kann als Dual-Port-Memory (DPM) oder Multi-Port-Memory (MPM) für die Kopplung von mehr als zwei Teilnehmern ausgeführt sein. Zur Vereinfachung gehen wir in der weiteren Beschreibung immer von einem Multi-Port-Memory aus.



5150C002

Bild 1-1 Kommunikationsmechanismen im MPM

Beschreibung MPM

Über das MPM können zwei oder mehr Teilnehmer (Nodes) Informationen miteinander austauschen. Das MPM gliedert sich dabei allgemein in den Datenbereich (Data Transmission Area; DTA), in den Signalbereich (Signal Area; SGA) und den Mailbox-Bereich (Mailbox Area; MXA).

Die Größe des MPM und die physikalische Lage der einzelnen Speicherbereiche ist host-abhängig, so dass genaue Adressangaben nicht gemacht werden können. Diese host-spezifischen Informationen sind dem entsprechenden Treiber-Referenzhandbuch oder der Anschaltbaugruppenbeschreibung zu entnehmen.



Im MPM können in der erweiterten Diagnose nicht alle Informationen aller Teilnehmer im MPM gespeichert werden. Anstatt der möglichen 253 Einträge werden nur 251 Einträge gespeichert.

Device Driver Interface

Die Funktionen im Device Driver Interface (DDI) erleichtern den Zugriff auf das MPM, so dass hardware- und host-spezifische Informationen nicht von der Firmware und dem Treiber benötigt werden. Das DDI ist mit seinen Basisfunktionen für alle Anschaltbaugruppen gleich.

- Die Teilnehmer müssen ihren MPM-Bereich für die Kommunikation öffnen oder schließen.
- Es werden zwei Funktionen zum Lesen und Schreiben von Prozessdaten im DTA, zwei Funktionen zum Senden und Empfangen von Diensten über das MXA und zwei Funktionen zur Bedienung des Signal-Interface (SGI) zur Verfügung gestellt.

Prozessdaten-Interface

Das Prozessdaten-Interface (DTI) bietet in einem Speicherauszug alle Ein- und Ausgangsdaten der angeschlossenen Peripherie.

Über das DTI können alle MPM-Teilnehmer lesend oder schreibend auf den Datenbereich (DTA) zugreifen. Aus Sicht des Hosts werden die Eingangsdaten im MPM in jedem INTERBUS-Zyklus mit den Informationen der Teilnehmer aktualisiert und Ausgangsdaten an die INTERBUS-Teilnehmer ausgegeben.

Der Zugriff von Host und Anschaltbaugruppe auf das MPM erfolgt standardmäßig in asynchroner Betriebsart und kann optional in synchroner Betriebsart erfolgen. Dabei werden entweder die Buszyklen mit einem Signal des Hosts gesteuert oder die Anschaltbaugruppe generiert ein Signal zur Steuerung des Hosts.

Fast alle Anschaltbaugruppen verfügen über einen Standardregistersatz, der zur Diagnose und zur einfachen Steuerung des Bussystems eingerichtet ist. Die Register werden im Ein- und Ausgangsbereich der Steuerung abgebildet und können dort frei positioniert werden (siehe Kapitel 1.4.2, Seite 1-15).

Mailbox-Interface

Das Mailbox-Interface (MXI) bietet eine Schnittstelle zur Übergabe von Diensten zwischen zwei MPM-Teilnehmern. Hierfür ist im MPM ein Protokoll implementiert, das Übergabe und Auslesen der Nachrichten gesichert

durchführt. Fast alle Dienste bestehen aus der Anforderung (Request) und der zugehörigen Antwort (Response). Bei jeder Antwort werden der Status der Dienstausführung und eventuell zusätzliche Parameter übergeben.



Eine Nachricht besteht immer aus einem Dienst-Code und der Parameteranzahl, auch wenn keine Parameter folgen (Parameterlänge = 0). Eine nähere Beschreibung des Dienstaufbaus finden Sie im einführenden Kapitel zu den einzelnen Diensten.

Signal-Interface

Das Signal-Interface wird in das Standard-Signal-Interface (SSGI) und das Erweiterte-Signal-Interface (XSIGI) unterteilt. Das Standard-Signal-Interface (SSGI) bietet eine Schnittstelle zur bitgesteuerten Konfiguration über vordefinierte Adressbereiche im Signalbereich (SGA). Das Erweiterte-Signal-Interface (XSIGI) bietet eine Schnittstelle zum bitgesteuerten Aktivieren von anwenderparametrierbaren Diensten.

1.4.2 Standardregister

Der Standardregistersatz besteht aus drei Diagnose-Registern:

- Diagnose-Statusregister (Eingangswort)
- Diagnose-Parameterregister (Eingangswort)
- Erweitertes Diagnose-Parameterregister (Eingangswort)

und drei Standardfunktionsregistern:

- Standardfunktions-Startregister (Ausgangswort)
- Standardfunktions-Parameterregister (Ausgangswort)
- Standardfunktions-Statusregister (Eingangswort)

Über den Firmware-Dienst „Set_Value“ (vgl. Seite 2-18) kann die Position der Register im MPM frei verschoben werden.

Diagnose-Register

Die Diagnose-Register der Anschaltbaugruppe (Diagnose-Statusregister, Diagnose-Parameterregister und erweitertes Diagnose-Parameterregister) bilden den aktuellen Zustand des INTERBUS-Systems für den Anwender ab. Dem Anwendungsprogramm können damit der Zustand des Bus-systems, Störungsgründe und weitere Informationen übermittelt werden.

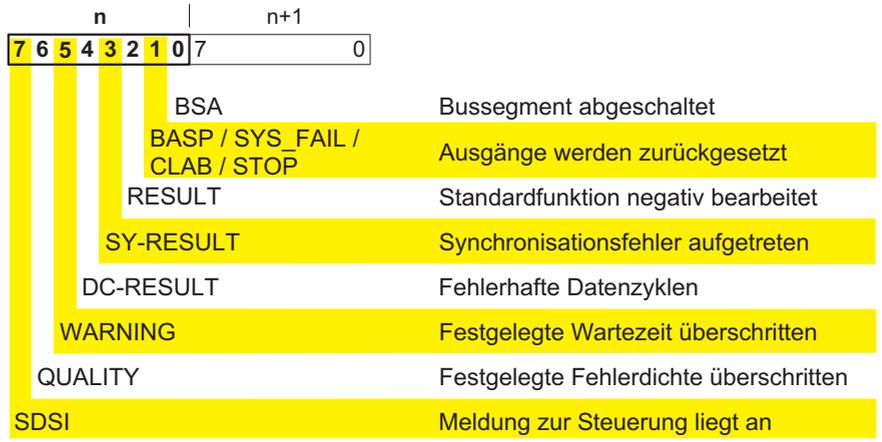
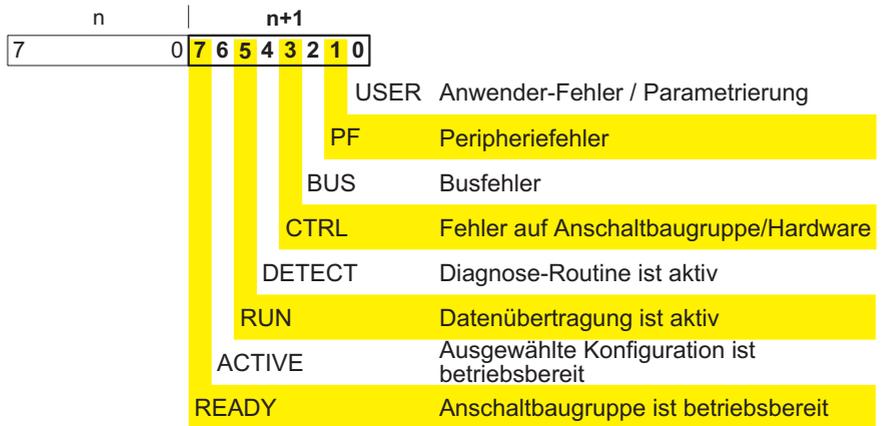
Jedem Bit im Diagnose-Statusregister ist ein Zustand der Anschaltbaugruppe zugeordnet. Die Zustände in den Fehlerbits (USER, PF, BUS, CTRL) werden über das Diagnose-Parameterregister und das erweiterte Diagnose-Parameterregister näher beschrieben. Diese Register werden immer dann neu beschrieben, wenn eines der o. g. Fehlerbits gesetzt wird. Andernfalls enthalten diese Register den Wert 0000_{hex}.



Das erweiterte Diagnose-Parameterregister liegt default-mäßig im MPM auf Adresse 37E6_{hex} und kann über den Dienst „Set_Value“ (Variable_ID 010C_{hex}) auch in den I/O-Bereich gelegt werden (siehe „Dienst „Set_Value““ auf Seite 2-18).

Beachten Sie, dass in dem erweiterten Diagnose-Parameterregister die Daten nur nach dem Absetzen des Dienstes „Confirm_Diagnostics“ (0760_{hex}) aktualisiert werden.

Diagnose-Statusregister



5150C003

Bild 1-2 Diagnose-Statusregister

Betriebsanzeigen

Die Betriebsanzeigen „READY“, „ACTIVE“ und „RUN“ zeigen den aktuellen Zustand des INTERBUS-Systems an. Das Diagnose-Parameterregister wird nicht genutzt.

Nach dem Selbsttest ist die Anschaltbaugruppe betriebsbereit. Das Anzeigebit „READY“ ist gesetzt (READY = 1).

Ist die Anschaltbaugruppe konfiguriert und ein Konfigurationsrahmen fehlerfrei aktiviert worden, meldet sich das System als aktiv. Die Anzeigebits „READY“ und „ACTIVE“ sind gesetzt (READY = 1, ACTIVE = 1).

Mit dem Start des Datenaustauschs wird zusätzlich das Anzeigebit „RUN“ gesetzt (READY = 1, ACTIVE = 1 und RUN = 1).

Fehleranzeigen

Das Fehlerbit „DETECT“ zeigt an, dass ein Fehler den weiteren Busbetrieb nicht zulässt (DETECT = 1). Die Ausgänge fallen auf den Zustand NULL ab. Die Diagnose-Routine sucht die Fehlerursache.

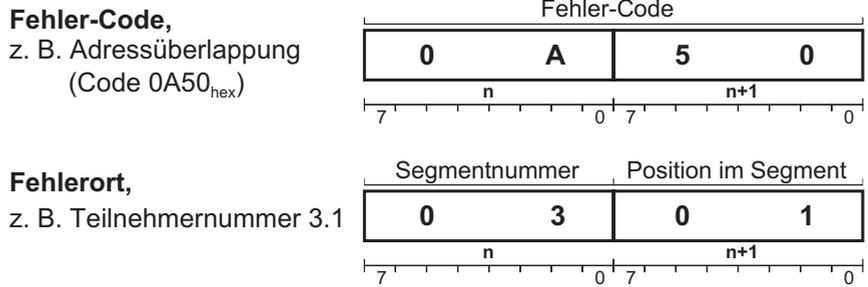
Nachdem die Ursache gefunden wurde, wird das Fehlerbit „DETECT“ zurückgenommen (DETECT = 0) und der Fehler in den Bits „USER“, „PF“, „BUS“ und „CTRL“ angezeigt. Das Diagnose-Parameterregister beschreibt die Fehlerursache näher.

Tabelle 1-2 Fehler mit Busabschaltung

Fehlerbit / -ort	Inhalt des Diagnose-Parameterregisters
CTRL = 1 Fehler wird auf der Anschaltbaugruppe / Hardware vermutet.	Fehler-Code, siehe Kapitel 3
BUS = 1 Der Fehler betrifft ein Fernbus- oder Lokaltbus-Segment.	Fehlerort, siehe Kapitel 3.3

Tabelle 1-3 Fehler ohne Busabschaltung

Fehlerbit / -ort	Inhalt des Diagnose-Parameterregisters
PF = 1 Fehler auf der Peripherieseite eines INTERBUS-Teilnehmers: – Kurzschluss am Ausgang – Sensor-Aktor Versorgung fehlt	Fehlerort, siehe Kapitel 3.3
USER = 1 Anwenderfehler, z. B. durch falsche Parameter	Fehler-Code, siehe Kapitel 3



5150C004

Bild 1-3 Inhalt des Diagnose-Parameterregisters (Beispiel)

Fehlerort

Bei lokalisierten Fern- oder Lokalbus-Fehlern beinhaltet das Diagnose-Parameterregister den Fehlerort:

- Fehler im Lokalbus: Teilnehmernummer eines Teilnehmers, z. B. „1.3“ für Bussegment 1; Teilnehmer 3
- Fehler im Fernbus: Teilnehmernummer der Busklemme, z. B. „1.0“ für Bussegment 1; Teilnehmer 0

Der genaue Fehlerort wird nur angegeben, wenn kein Schnittstellenfehler vorliegt (Bit 7 gleich 0). Liegt ein Schnittstellenfehler vor (Bit 7 gleich 1), wenn z. B. der dort angeschlossene Bus nicht betreibbar ist, wird nur das fehlerhafte Bussegment angegeben. Am Bit 0 erkennen Sie, ob der Fehlerort an der weiterführenden Fernbus-Schnittstelle (Bit 0 gleich 0) oder ob der Fehlerort an der abzweigenden Fernbus-Schnittstelle (Bit 0 gleich 1) liegt.



Beachten Sie, dass die INTERBUS-Diagnose in Einzelfällen nur eine eingeschränkte Fehlerlokalisierung zulässt:

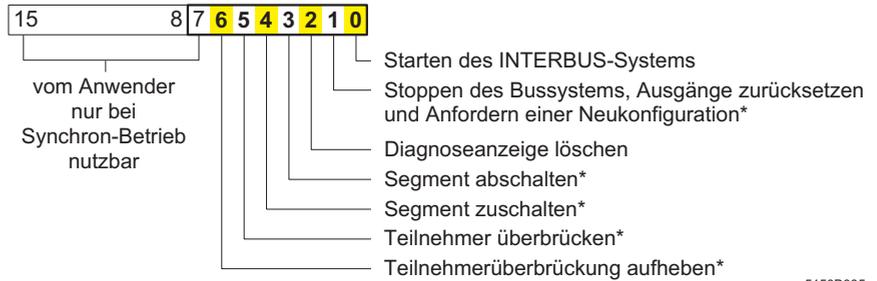
- bei einer direkt an die Anschaltbaugruppe angeschlossenen Inline-Station arbeitet die Fehlerlokalisierung nur mit Einschränkungen,
- Fehlerorte im Loop-Abzweig werden unter bestimmten Voraussetzungen direkt auf die erste folgende Inline-Scheibe abgebildet.

Standardfunktionsregister

Über die Standardfunktionsregister können Sie vordefinierte und häufig verwendete Funktionen durch Setzen eines Ausgangsbits ausführen lassen sowie die Ausführung und das Ergebnis überwachen.

Über das Standardfunktions-Startregister führen Sie die gewünschte Funktion aus. Einige Funktionen benötigen für den Funktionsaufruf Zusatzinformationen im zugehörigen Standardfunktions-Parameterregister.

Den Bits im Standardfunktions-Startregister sind folgende Funktionen zugeordnet:



5150B005

Bild 1-4 Zuordnung von häufig genutzten Funktionen im Standardfunktions-Startregister

Die mit „*“ gekennzeichneten Funktionen erfordern zur Ausführung noch weitere Parameter (z. B. eine Teilnehmernummer), die Sie im separaten Standardfunktions-Parameterregister übergeben müssen.



Da zu einem Zeitpunkt immer nur eine Standardfunktion aktiviert werden darf, lässt sich eine neue Funktion nur dann ausführen, wenn eine zuvor aktivierte Funktion beendet wurde.

Den Bearbeitungsablauf kontrollieren Sie über das Standardfunktions-Statusregister. Jedem Bit im Standardfunktions-Startregister ist ein gleichnamiges Bit im Standardfunktions-Statusregister zugeordnet. Während der Bearbeitung einer Funktion ist das zugehörige Statusbit aktiv, und zwar unabhängig davon, ob die Funktion erfolgreich ausgeführt werden konnte oder nicht. Das Statusbit wird nach der Bearbeitung der Funktion auf „0“ gesetzt.

Das Standardfunktions-Ergebnisbit („RESULT“) im Diagnose-Statusregister zeigt an, ob die Funktion erfolgreich beendet wurde oder nicht.

RESULT = 1 Funktion nicht erfolgreich beendet

RESULT = 0 Funktion erfolgreich beendet

Funktionsausführung

Die zwei folgenden Bilder verdeutlichen den Ablauf einer Funktionsausführung mit und ohne Parameterübergabe.

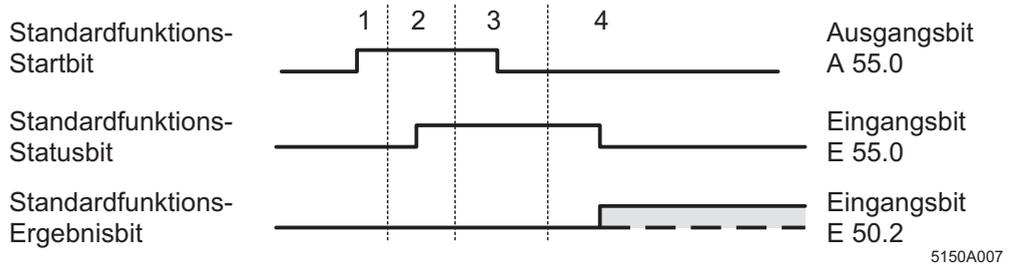


Bild 1-5 Ablauf einer Funktionsausführung ohne Parameterübergabe

1. Die Funktion wird vom Steuerungsprogramm über das entsprechende Funktionsbit im Standardfunktions-Startregister aktiviert.
2. Die Funktionsausführung wird über das zugehörige Statusbit im Standardfunktions-Statusregister angezeigt.
3. Mit dem Zurücksetzen des Funktionsbits durch den Host wird eine erneute Funktionsaktivierung verhindert.
4. Das Ergebnis der Funktionsausführung zeigt das Bit „RESULT“ im Diagnose-Statusregister an.

RESULT = 0 Funktion erfolgreich beendet

RESULT = 1 Funktion nicht erfolgreich beendet

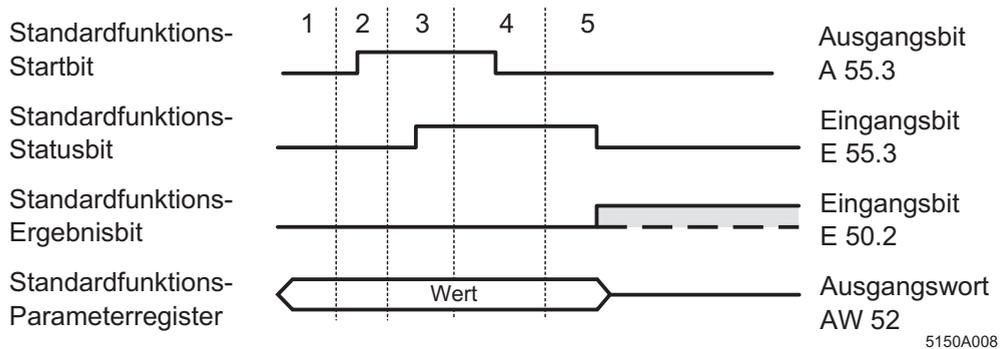


Bild 1-6 Ablauf einer Funktionsausführung mit Parameterübergabe

1. Tragen Sie den benötigten Parameter in das Standardfunktions-Parameterregister ein.
2. Die Funktion wird vom Steuerungsprogramm über das entsprechende Funktionsbit im Standardfunktions-Startregister aktiviert.
3. Die Funktionsausführung wird über das zugehörige Statusbit im Standardfunktions-Statusregister angezeigt.
4. Mit dem Zurücksetzen des Funktionsbits durch den Host wird eine erneute Funktionsaktivierung verhindert.
5. Das Ergebnis der Funktionsausführung zeigt das Bit „RESULT“ im Diagnose-Statusregister an.

RESULT = 0 Funktion erfolgreich beendet

RESULT = 1 Funktion nicht erfolgreich beendet

1.4.3 Zustandsmaschine

Die Zustandsmaschine beschreibt den fest definierten Ablauf von Zuständen beim Hochlaufen und im Betrieb des INTERBUS. Mit Hilfe der Zustandsmaschine wird jeder einzelne Dienst auf Zulässigkeit geprüft, um vorzeitig auf Fehler zu reagieren.

Folgendes Bild zeigt die Zustandsmaschine der Firmware-Generation 4:

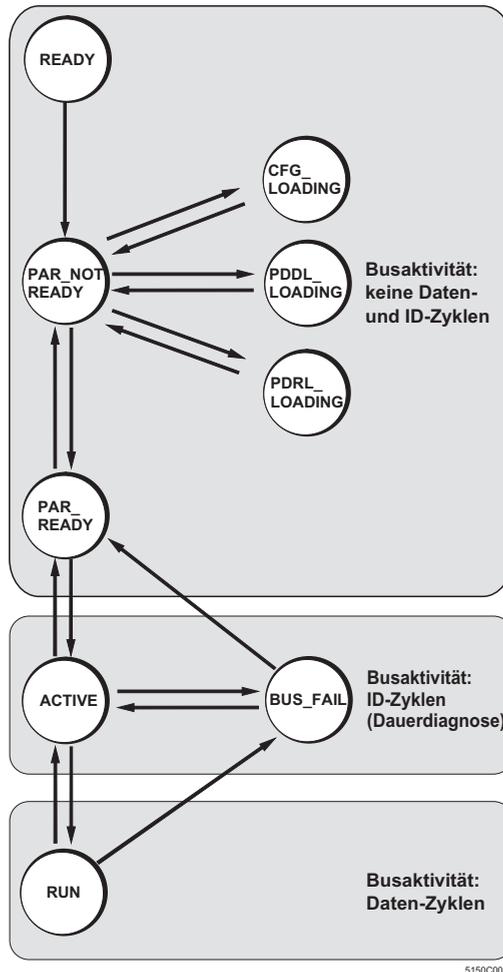


Bild 1-7 Die Zustandsmaschine

Bedeutung der Zustände

READY	„READY“ ist der Ausgangszustand der Firmware nach jedem Hochlaufen (Kalt- oder Warmstart). In diesem Zustand liegt keine gültige Konfiguration vor. Der Bus ist nicht aktivierbar.
PAR_NOT_READY	Nach Einleiten der Parametrierungsphase befindet sich das System-Management im Zustand „PAR_NOT_READY“. Aus diesem Zustand heraus können die drei Parametrierungssequenzen zum Laden der <ul style="list-style-type: none"> – Konfigurationsrahmen – Prozessdatenbeschreibungslisten (PDDL) und – Prozessdatenreferenzlisten (PDRL) angestoßen werden. Während der Parametrierungsphase ist die Gültigkeit der auslesbaren Konfigurationsparameter nicht garantiert.
CFG_LOADING	Um einen Konfigurationsrahmen zu laden, muss sich das System im Zustand „CFG_LOADING“ befinden.
PDDL_LOADING	Um Prozessdaten zu definieren, muss sich das System im Zustand „PDDL_LOADING“ befinden.
PDRL_LOADING	Um Prozessdaten zu verknüpfen, muss sich das System im Zustand „PDRL_LOADING“ befinden.
PAR_READY	„PAR_READY“ ist der Zustand der Firmware, nachdem die Parametrierungsphase erfolgreich abgeschlossen wurde. Es wurde zumindest ein Konfigurationsrahmen definiert. Dieser wurde während der Parametrierung erfolgreich auf Vollständigkeit und Widerspruchsfreiheit getestet.
	In den Zuständen „PAR_NOT_READY“, „PAR_READY“, „CFG_LOADING“, „PDDL_LOADING“ und „PDRL_LOADING“ ist der Bus inaktiv geschaltet. Es werden keine Daten- und ID-Zyklen gefahren.
ACTIVE	Der Bus ist aktiv. Es werden ausschließlich ID-Zyklen gefahren. Es findet kein Datentransport über den Bus statt. Der Bus wird mit der Konfiguration betrieben, die im Konfigurationsrahmen als aktive Konfiguration gekennzeichnet ist.
RUN	In diesem Zustand werden Daten-Zyklen gefahren. Der eigentliche Daten-Transfer findet nur in diesem Zustand statt.

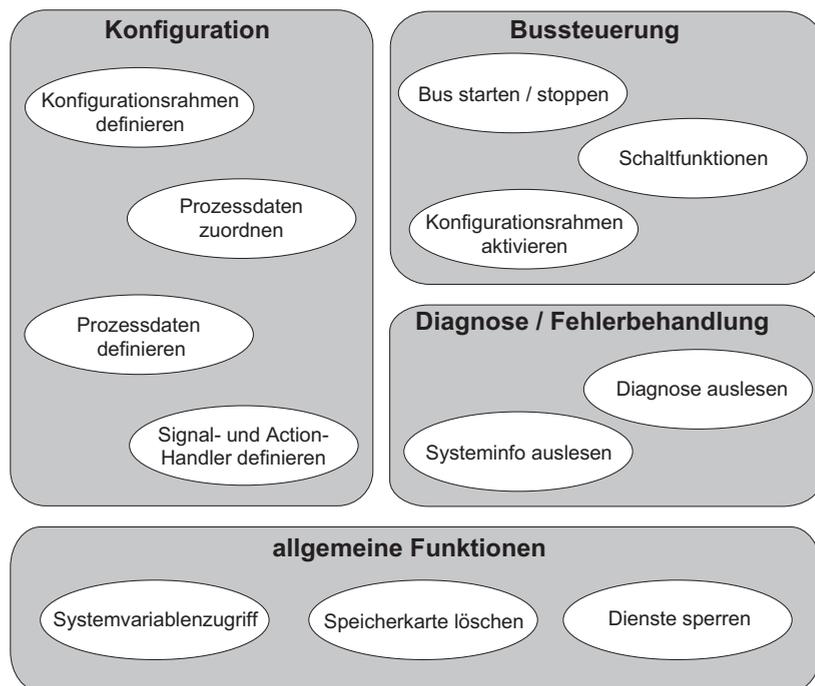
BUS_FAIL Zustand des System-Managements, wenn in den Zuständen ACTIVE oder RUN Busfehler auftreten. Es werden weiterhin ID-Zyklen mit der größtmöglich zu betreibenden Konfiguration gefahren.



Die Zustände „ACTIVE“, „RUN“ und „BUS_FAIL“ können Sie auch direkt aus dem Diagnose-Statusregister ablesen.

1.4.4 Befehlsgruppen der Firmware

Die Dienste der Firmware im System-Management sind sehr eng miteinander verzahnt. Man kann zur Übersicht den Großteil der Dienste nach ihrer Funktionalität in die Gruppen Konfiguration, Bussteuerung, Diagnose und allgemeine Funktionen einteilen.



5150C010

Bild 1-8 Dienstgruppenübersicht

Konfiguration

Die größte Zahl der Dienste ist für die Konfiguration der Anschaltbaugruppe zuständig. Diese gliedern sich in die Bereiche Konfigurationsrahmen-Verwaltung, Prozessdatenverwaltung und Action- und Signal-Object-Verwaltung.

Konfigurationsrahmen-Verwaltung:

Mit Hilfe der Konfigurationsrahmen-Verwaltung werden die für jeden Teilnehmer notwendigen Informationen im Konfigurationsrahmen bearbeitet. Der Konfigurationsrahmen beschreibt jeden enthaltenen INTERBUS-Teilnehmer in einem eigenen nummerierten Eintrag. Sie können mehrere

Konfigurationsrahmen einrichten, wieder löschen oder vergleichen. Die Reihenfolge der Teilnehmer im Konfigurationsrahmen entspricht dem physikalischen Busaufbau inklusive aller Alternativen.

Prozessdatenverwaltung:

Die Prozessdatenverwaltung ermöglicht für jeden Teilnehmer Prozessdatenbeschreibungen über die Prozessdatenbeschreibungsliste einzurichten und Adressen im MPM über die Prozessdatenreferenzliste zuzuordnen. Diese Prozessdatenbeschreibungen bieten die Möglichkeit, die Informationen im Datenwort eines Teilnehmers bis auf ein einzelnes Bit direkt zu adressieren. Über die Zuordnung der Prozessdaten können Sie damit ein logisches Prozessdatenabbild frei definieren. Änderungen im Busaufbau benötigen keine Adressverschiebung im Steuerungsprogramm, sondern nur eine Anpassung der internen Zuordnung.

Action- und Signal-Object-Verwaltung:

Die Action- und Signal-Object-Verwaltung ist ein flexibler Mechanismus, um über Steuerbytes vorprojektierte Funktionen zu aktivieren. Diese vorprojektierten Funktionen sind auf der Anschaltbaugruppe abgelegt. Hiermit können anwendungsspezifische Reaktionen mit einem einzigen Befehl angestoßen werden. Die notwendigen Übergabeparameter müssen in zuvor definierten Adressen im Datenbereich des MPM abgelegt werden.

Bussteuerung

Die Bussteuerung beschreibt die direkte Beeinflussung des Bussystems mit Firmware-Diensten. Hierbei können zwei Gruppen unterschieden werden. Erstens die Dienste für die Änderung des Systemstatus auf dem Bus, zweitens die direkten Schaltfunktionen einzelner Teilnehmer, Gruppen oder Alternativen während des Busbetriebs ohne eine Beeinflussung der Zustandsmaschine.

Diagnose-/ Fehlerbehandlung

Die Diagnose-Funktionen bieten umfangreiche Informationen über den Zustand des Bussystems. Hierzu zählen u. a. Statistikinformationen über Fehler im Bussystem für jeden einzelnen Teilnehmer und eine Information über Hard- und Software-Revision.

Allgemeine Funktionen

Die allgemeinen Funktionen können nicht einer der obigen Gruppen zugeordnet werden. Hierunter fallen u. a. Dienste für den Zugriff auf eine eventuell vorhandene Speicherkarte, Dienste für Zugriff auf Systemvariablen und vieles mehr.

1.4.5 Konfigurationsrahmen

Der Konfigurationsrahmen spiegelt tabellarisch den Aufbau des Bussystems wider. Über die Einträge „Teilnehmer-Identifikationscode“ (ID-Code), „Längen-Code“ und „Busebene“ sowie die Position des Eintrages in der Tabelle ist das Bussystem mit seiner Topologie und dem Summenrahmen komplett beschrieben.

Teilnehmer-ID: Die Teilnehmer-ID bezeichnet den Typ eines INTERBUS-Teilnehmers und damit seine Funktion.



Informationen zum ID-Code von INTERBUS-Teilnehmern können Sie der INTERBUS-Teilnehmerliste entnehmen oder wenden Sie sich an den INTERBUS-Club.

Teilnehmer-Längen-Code: Der Längen-Code beschreibt den Adressraum eines INTERBUS-Teilnehmers und damit in codierter Form die Länge der Ein- und Ausgangs-Prozessdaten.



Lesen Sie in Abschnitt „Zusammensetzung des Teilnehmer-Längen-Codes“ auf Seite 1-36 dieses Kapitels nach, wie der Längencode eines INTERBUS-Teilnehmers zusammengesetzt ist.

Busebene: Die Busebene beschreibt, auf welcher Ebene der INTERBUS-Baumstruktur sich jeder Teilnehmer befindet.

Der Stamm des Baumes besitzt ausgehend von der Anschaltbaugruppe die Ebene Null. Busklemmen eröffnen mit ihrem Lokalbus- und Installationsfernbus-Anschluss einen neuen Zweig (Busebene). Diese Struktur ist begrenzt auf 16 Ebenen.

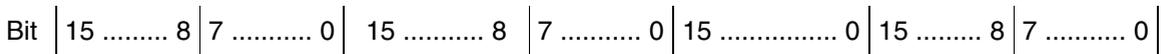


Die Informationen „ID-Code“, „Längen-Code“ und „Busebene“ sind Pflichteinträge für jeden Teilnehmer. Die Reihenfolge der Teilnehmer im Rahmen stimmt mit der Position innerhalb des Bussystems überein. Ausnahmen hiervon sind alternative Teilnehmer, die nicht gleichzeitig am Bus betrieben werden.

Die zwei Felder „logische Teilnehmernummer“ und „logische Gruppennummer“ bieten die Basis für Mechanismen zum Schalten von verteilten Teilnehmergruppen, Bussegmenten und alternativen Gruppen.

Tabelle 1-4 Prinzipieller Aufbau eines Konfigurationsrahmens

Nr.	logische Teilnehmernummer		Teilnehmer-Kurzbezeichnung		Busebene	logische Gruppennummer	
	Segment	Position	Längen-Code	ID-Code		Gruppe	Alternative
1							
2							
...							
n							



Der definierte Konfigurationsrahmen darf zu keiner Zeit durch die angeschlossene Buskonfiguration überschritten werden. Ebenso muss die Position der Teilnehmer immer mit den Angaben im Konfigurationsrahmen übereinstimmen. Die einzig mögliche Abweichung ist, wenn die aktive Konfiguration nach Abschalten von Bussegmenten oder Gruppen eine Untermenge des Konfigurationsrahmens darstellt.

Um die Funktionalität der Konfigurationsrahmen-Verwaltung besser darstellen zu können, werden zuerst die Begriffe *angeschlossene* und *aktive Konfiguration* und ihre Beziehungen zueinander genauer erläutert.

angeschlossene Konfiguration

Die angeschlossene Konfiguration umfasst alle tatsächlich am INTERBUS angeschlossenen Teilnehmer, die physikalisch erreicht werden können, wenn alle Bussegmente zugeschaltet sind. Abgezogen werden die im aktuellen Zustand nicht angeschlossenen Alternativen. Die angeschlossene Konfiguration ist damit eine Teilmenge des kompletten Konfigurationsrahmens.

aktive Konfiguration

Die aktive Konfiguration umfasst alle tatsächlich am INTERBUS angeschlossenen Teilnehmer, die mit ihren Daten im Summenrahmen liegen. Das heißt, die aktive Konfiguration ist gleich der angeschlossenen

Konfiguration abzüglich der zur Zeit abgeschalteten Bussegmente. Die aktive Konfiguration ist immer eine Teilmenge der angeschlossenen Konfiguration.

Folgendes Beispiel verdeutlicht die Unterschiede zwischen angeschlossener und aktiver Konfiguration:

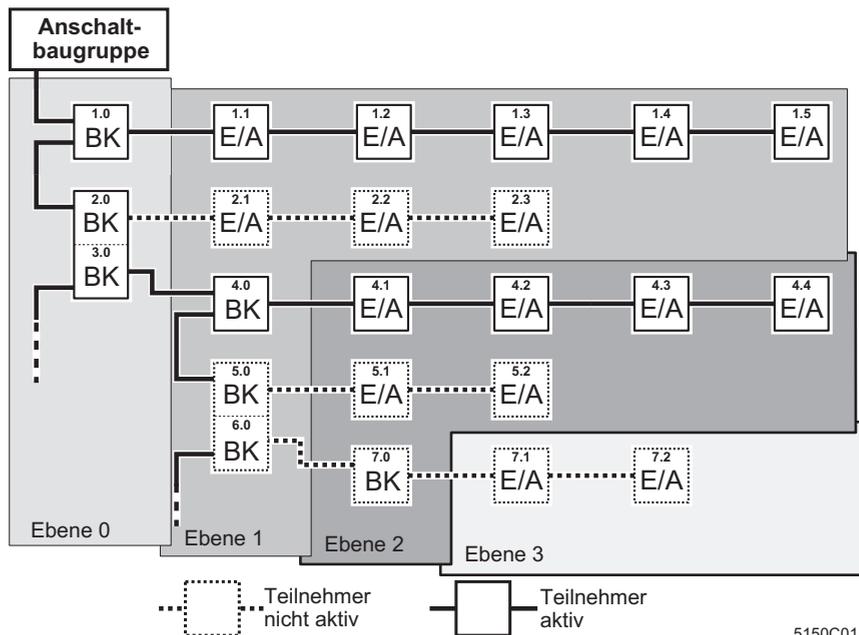


Bild 1-9 Busaufbau (angeschlossene und aktive Konfiguration)

Tabelle 1-5 Konfigurationsrahmen

log. Teilnehmer-Nr.	Teilnehmer-Kurzbez.	Busebene	logische Gruppen-Nr.	aktiv
1.0	...	0	–	ja
1.1	...	1	–	ja
1.2	...	1	–	ja
1.3	...	1	–	ja
1.4	...	1	–	ja
1.5	...	1	–	ja

Tabelle 1-5 Konfigurationsrahmen

log. Teilnehmer-Nr.	Teilnehmer-Kurzbez.	Busebene	logische Gruppen-Nr.	aktiv
2.0	...	0	1	nein
2.1	...	1	1	nein
2.2	...	1	1	nein
2.3	...	1	1	nein
3.0	...	1	2	ja
4.0	...	1	2	ja
4.1	...	2	2	ja
4.2	...	2	2	ja
4.3	...	2	2	ja
4.4	...	2	2	ja
5.0	...	1	2	nein
5.1	...	2	2	nein
5.2	...	2	2	nein
6.0	...	1	2	nein
7.0	...	2	2	nein
7.1	...	3	2	nein
7.2	...	3	2	nein

Jeder Konfigurationsrahmen enthält neben den eigentlichen Konfigurationsdaten (ID-Code, Längen-Code, Teilnehmernummer, Busebene, logische Gruppennummer, etc.) des Busaufbaus noch weitere Listen:

- Die Prozessdatenbeschreibungsliste (PDDL)
- Die Prozessdatenreferenzliste (PDRL)
- Die Kommunikationsbeziehungsliste (CRL)
- Spezifische Betriebsparameter (z. B. Update-Time, Timeout)
- Statistische Diagnose-Daten usw.

Logische Teilnehmernummern

Die logische Teilnehmernummer besteht im höherwertigen Byte aus dem Bussegment und im niederwertigen Byte aus der Position in diesem Bussegment.

Bussegment

Ein Bussegment besteht aus einem Fernbus-Teilnehmer und dem ankommenden Fernbus-Kabel. Falls dieser Fernbus-Teilnehmer eine Busklemme ist, gehören auch die Teilnehmer des ggf. abzweigenden Lokalbusses zu diesem Bussegment.

Bussegment 1 ist das Bussegment, dessen Busklemme über den Fernbus mit der Anschaltbaugruppe verbunden ist. Bussegment 2 ist über den Fernbus an die Busklemme des Bussegments 1 gekoppelt.

Busklemmen für den Installationsfernbus und Installationsfernbus-Teilnehmer besitzen eine eigene Bussegmentnummer und koppeln eine untergeordnete Teilnehmerebene an.

Die Positionsinformation in der logischen Teilnehmernummer entspricht der Kennung dieses Teilnehmers in diesem Bussegment. Über die Bussegment- und Positionsnummer ist damit jeder Teilnehmer eindeutig identifiziert.

Physikalische Teilnehmer-nummerierung

Die einfachste Variante für die Nummerierung eines Teilnehmers im Konfigurationsrahmen ist die physikalische Nummerierung, d. h. alle Teilnehmer werden in der Reihenfolge nummeriert, in der sie sich physikalisch im Netz befinden.

Alle Fernbus-Teilnehmer erhalten bei Eröffnung eines neuen Bussegmentes als Positionskennzeichnung eine 0 (x.0). Die einzelnen Segmente werden in aufsteigender Reihenfolge durchnummeriert. Wird von einem Fernbus-Teilnehmer sowohl eine neue Ebene eröffnet (Stich) als auch ein weiteres Fernbus-Segment derselben Ebene, so wird die Nummerierung in der neu eröffneten Ebene fortgesetzt, anschließend in der nächst höheren Ebene.

Innerhalb eines Bussegments wird die Segmentnummer beibehalten und die weiteren Teilnehmer werden durch die in aufsteigender Reihenfolge nummerierte Position gekennzeichnet.

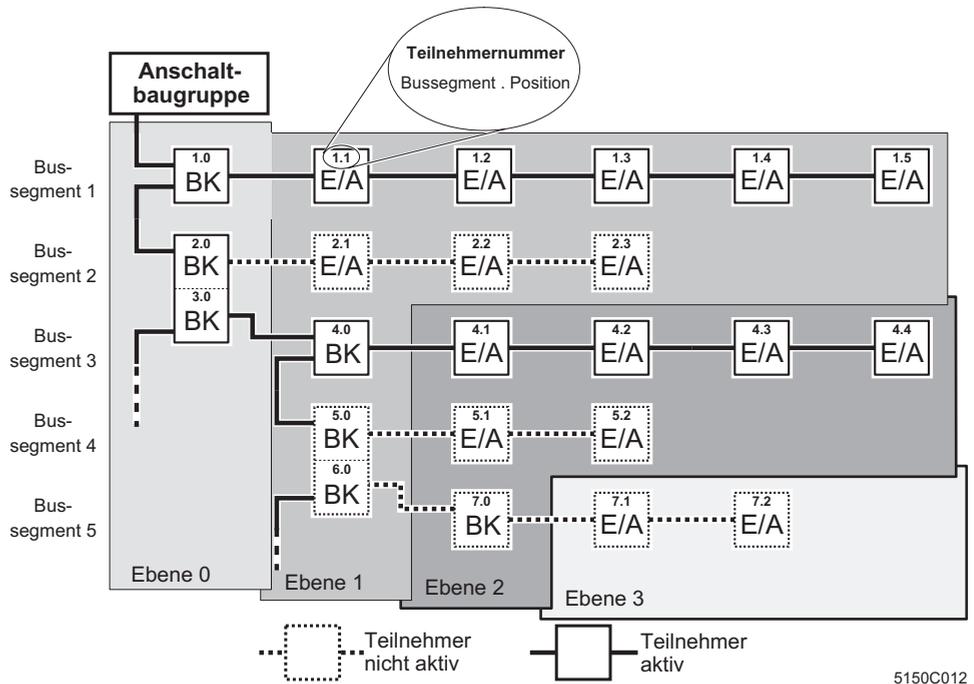


Bild 1-10 Aufbau der Teilnehmernummern

Die physikalische Teilnehmernummerierung wird nur im Test-Mode und nach dem Dienst „Create_Configuration“ (vgl. Seite 2-62) benutzt. Für das Verfahren gilt, dass das Einfügen neuer Teilnehmer die Teilnehmernummern aller unterlagerten Teilnehmer verschiebt.

Logische Teilnehmernummerierung

Mit der logischen Teilnehmernummerierung können die Bussegmentnummer und die Positionsnummer eines Teilnehmers frei gesetzt werden. Das Einfügen eines neuen Zweiges oder einzelnen Teilnehmers verändert dann nicht mehr die Nummerierung der schon bestehenden Teilnehmer. Hierfür gelten zwei Regeln:

- Busklemmen bekommen immer die Positionsnummer „0“.
- Teilnehmer eines Lokalbusses erhalten immer die Bussegmentnummer der überlagerten Busklemme.

Die logische Teilnehmernummerierung erfolgt durch das Herunterladen eines vordefinierten Konfigurationsrahmens mittels der Dienste

- „Load_Configuration“ (vgl. Seite 2-31) oder
- „Complete_Load_Configuration“ (vgl. Seite 2-54).

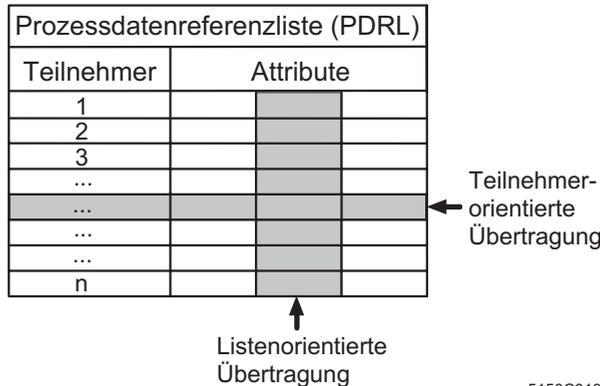
Um das Laden einzelner Teile des Konfigurationsrahmens zu ermöglichen, bietet die Firmware zwei verschiedene Modi (die teilnehmer- und die listenorientierte Übertragung) an, so dass über Parameter auf jede Teilmenge des Konfigurationsrahmens zugegriffen werden kann.

teilnehmerorientierte Übertragung

Bei der teilnehmerorientierten Übertragung werden unter Angabe der physikalischen Teilnehmernummer (Zeilennummer des Konfigurationsrahmens) vollständige Teilnehmerbeschreibungen in den Konfigurationsrahmen geladen. Mit Hilfe des Parameters *Used_Attributes* können zusätzlich einzelne Spalten für die Übertragung selektiert werden.

listenorientierte Übertragung

Bei der listenorientierten Übertragung werden die Informationen des Konfigurationsrahmens in Listenform (z. B.: ID-Liste, Liste der Gruppennummern, etc.) geladen. Geordnet werden sie dabei nach der physikalischen Busposition. Es können auch Teillisten durch Setzen der Parameter *Start_Entry_No* und *Entry_Count* übertragen werden.



5150C013

Bild 1-11 Teilnehmerorientierte und listenorientierte Übertragung

Die Art des Ladens wird durch die Parameter *Used_Attributes*, *Device_Number* und *Entry_Count* bestimmt. Der Parameter *Used_Attributes* wählt aus der Konfigurationsmatrix die Spalten aus, wobei eine beliebige Kombination möglich ist. Die Parameter *Device_Number* und *Entry_Count* entscheiden, ab welcher Listenposition wie viele Einträge der Liste geladen werden. Hierbei besteht die Einschränkung, dass immer nur ein zusammenhängendes Stück einer Liste geladen werden kann.

Mit diesen Parametern lassen sich alle Formen des Ladens realisieren. So könnte z. B. die gesamte Matrix geladen werden, wenn alle Bits des Parameters *Used_Attributes* gesetzt sind, die *Device_Number* gleich Null und *Entry_Count* gleich der Anzahl der Geräte des Konfigurationsrahmens ist.

Logische Gruppennummern

Die logische Gruppennummer teilt sich auf in Gruppennummer und Alternativen-Information. Standardmäßig wird keine logische Gruppennummer vergeben (Wert FFFF_{hex}).



Eine Gruppennummer braucht nur dann vergeben zu werden, wenn physikalisch entfernt liegende Bussegmente zusammen mit einem Befehl geschaltet werden sollen oder alternative Gruppen im Busaufbau vorgesehen sind. Diese Gruppennummer ist nur für die angesprochenen Teilnehmer notwendig. Alle anderen Teilnehmer können weiterhin mit der Standardkennung versehen bleiben.

Gruppenzugehörigkeit

Das höherwertige Byte enthält die Gruppennummer. Mit dieser können auch im System verteilte Teilnehmer einer Gruppe zugewiesen werden. Im niederwertigen Byte der Gruppennummer ist die Information enthalten, ob es sich bei der Gruppe um eine alternativ schaltbare Gruppe handelt.

Alternativen

Alternativ schaltbare Gruppen ermöglichen dem Anwender das Schalten von unterschiedlich strukturierten Bussegmenten an ein und demselben Fernbus-Ausgang einer Busklemme. Alternative Teilnehmer werden im gleichen Konfigurationsrahmen mit der gleichen Gruppennummer, aber einer unterschiedlichen Alternativnummer geführt. Die Teilnehmer in unterschiedlichen Alternativen müssen immer unterschiedliche logische Teilnehmernummern besitzen, da zugehörige Prozessdatenbeschreibungen (PDD) nur auf die Teilnehmernummer verweisen. Es ist unzulässig, eine logische Teilnehmernummer doppelt zu vergeben.

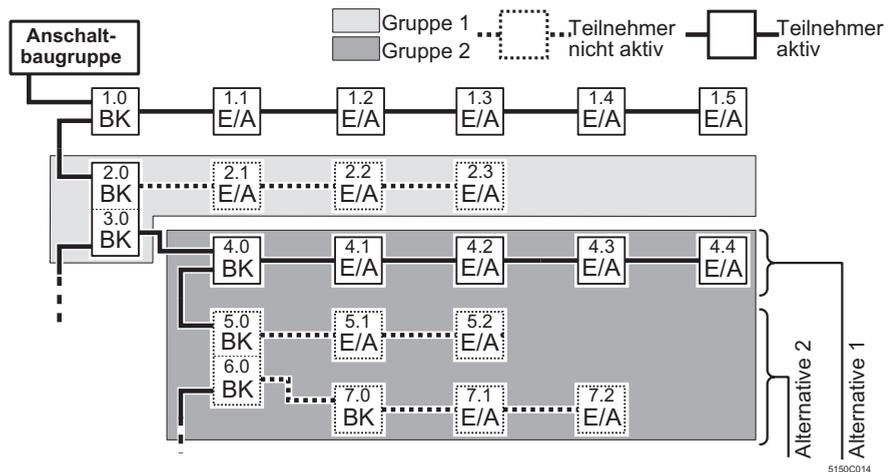


Bild 1-12 Busaufbau (Gruppen und Alternativen)

5150C014

Tabelle 1-6 Konfigurationsrahmen

log. Teilnehmer-Nr.	Teilnehmer-Kurzbez.	Busebene	logische Gruppen-Nr.	aktiv
1.0	...	0	–	ja
1.1	...	1	–	ja
1.2	...	1	–	ja
1.3	...	1	–	ja
1.4	...	1	–	ja
1.5	...	1	–	ja
2.0	...	0	1.1	nein
2.1	...	1	1.1	nein
2.2	...	1	1.1	nein
2.3	...	1	1.1	nein
3.0	...	0	2.1	nein
4.0	...	1	2.1	nein
4.1	...	2	2.1	ja
4.2	...	2	2.1	ja
4.3	...	2	2.1	ja
4.4	...	2	2.1	ja
5.0	...	1	2.2	nein
5.1	...	2	2.2	nein
5.2	...	2	2.2	nein
6.0	...	2	2.2	nein
7.0	...	2	2.2	nein
7.1	...	3	2.2	nein
7.2	...	3	2.2	nein



Alternativen werden nicht automatisch beim Anlauf im System mit eingebunden, sondern müssen explizit mit dem Dienst „Control_Active_Configuration“ (vgl. Seite 2-104) hinzugeschaltet werden.

Zusammensetzung des Teilnehmer-Längen-Codes

In diesem Abschnitt lesen Sie, wie der Teilnehmer-Längen-Code (im Folgenden: Längen-Code) gebildet wird und wie Sie daraus die Breite des Prozessdatenkanals ablesen können.

Beim Längen-Code unterscheidet man zwischen zwei Darstellungen.

- 1 Längen-Code, der in der INTERBUS-Norm DIN EN 50254 : 1999-07 definiert ist.

Dieser Code wird ausschließlich in den Schichten 1 und 2 des OSI-Referenzmodells verwendet. Jeder Code definiert eine bestimmte Länge, die aus einer Tabelle ermittelt werden kann.

- 2 Längen-Code, der durch die Firmware dem Anwender nutzbar gemacht wird.

In der Firmware findet eine Umsetzung des Längen-Codes entsprechend der Norm auf eine Darstellung statt, die für den Anwender auch ohne Referenztafel eindeutig ist. Diese Darstellung wird in der Produkt-Dokumentation (z. B. Packungsbeilagen, Datenblätter) als Längen-Code dokumentiert.

Der durch die Firmware generierte Längen-Code wird, wie im folgenden Bild dargestellt, in einem Byte abgebildet.

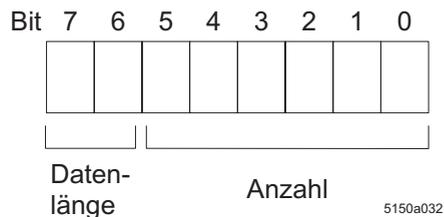


Bild 1-13 Zusammensetzung des Längen-Codes

Bit 7 und 6 geben die Datenlänge an. Dabei sind:

Bit 7	Bit 6	Datenlänge
0	0	Wort (16 Bit)
1	0	Byte (8 Bit)
0	1	Nibble (4 Bit)
1	1	Bit (1 Bit)



Bit 5 bis 0 geben die Anzahl der Dateneinheiten mit dieser Länge an.

Als Datenlänge wird immer die maximal mögliche Länge angegeben (z. B. nicht 2 Byte, sondern 1 Wort).

Beispiele:

Längen-Code für einen Prozessdatenkanal mit einer Länge von 2 Worten:	Bit	7	6	5	4	3	2	1	0
		0	0	0	0	0	0	1	0
	Wort	2						Längen-Code	
		0			2			02_{hex}	

Längen-Code für einen Prozessdatenkanal mit einer Länge von 1 Byte:	Bit	7	6	5	4	3	2	1	0
		1	0	0	0	0	0	0	1
	Byte	1						Längen-Code	
		8			1			81_{hex}	

Längen-Code für einen Prozessdatenkanal mit einer Länge von 1 Nibble:	Bit	7	6	5	4	3	2	1	0
		0	1	0	0	0	0	0	1
	Nibble	1						Längen-Code	
		4			1			41_{hex}	

Längen-Code für einen Prozessdatenkanal mit einer Länge von 2 Bit:	Bit	7	6	5	4	3	2	1	0
		1	1	0	0	0	0	1	0
	Bit	2						Längen-Code	
		C			2			C2_{hex}	

Übersicht über die Längen-Codes

Zuordnung der Längen-Codes entsprechend DIN EN 50254 zu den Längen-Codes für den Anwender:

DIN EN 50254		Anwender	
Bit 12 bis Bit 8	Daten- breite	Längen- Code [hex]	Daten- breite
0 0 0 0 0	0	00	0
0 1 1 0 0	1 Bit	C1	1 Bit
0 1 1 0 1	2 Bit	C2	2 Bit
0 1 0 0 0	4 Bit	41	1 Nibble
0 1 0 0 1	1 Byte	81	1 Byte
0 0 0 0 1	2 Byte	01	1 Wort
0 1 0 1 1	3 Byte	83	3 Byte
0 0 0 1 0	4 Byte	02	2 Worte
0 0 0 1 1	6 Byte	03	3 Worte
0 0 1 0 0	8 Byte	04	4 Worte
0 0 1 0 1	10 Byte	05	5 Worte
0 1 1 1 0	12 Byte	06	6 Worte
0 1 1 1 1	14 Byte	07	7 Worte
0 0 1 1 0	16 Byte	08	8 Worte
0 0 1 1 1	18 Byte	09	9 Worte
1 0 1 0 1	20 Byte	0A	10 Worte
1 0 1 1 0	24 Byte	0C	12 Worte
1 0 1 1 1	28 Byte	0E	14 Worte
1 0 0 1 0	32 Byte	10	16 Worte
1 0 0 1 1	48 Byte	18	24 Worte
1 0 0 0 1	52 Byte	1A	26 Worte
1 0 1 0 0	64 Byte	20	32 Worte
1 0 0 0 0	Reserviert		
1 1 x x x			

1.4.6 Prozessdaten

Prozessdaten- verwaltung

In der Firmware der Generation 4 wird der Prozessdatenkanal durch die Prozessdatenverwaltung erweitert, die flexibel auf alle Anwendungen angepasst werden kann. Mit den Diensten der Prozessdatenverwaltung können folgende Funktionen ausgeführt werden:

- Aufteilung des Prozessdatenkanals eines Teilnehmers in mehrere kleine Teile, sogenannte Prozessdatenbeschreibungen (PDD = process data description), bis auf Bit-Größe
- Völlig unabhängige Adressierung dieser PDDs im Host-Adressraum
- Broadcast-Adressierung der PDDs
- Definition von Durchgangsdaten
- Überlagerung von PDDs mit niedriger Priorität durch PDDs mit höherer Priorität.

Die Prozessdatenverwaltung wird in Dienste zur Definition von PDDs und zur Zuordnung der PDDs unterteilt.

Prozessdaten- beschreibungen

Über die PDDs in der Prozessdatenbeschreibungsliste (PDDL = process data description list) wird eine optimale, auf die Anwendung zugeschnittene Verwaltung der Modul-Eingänge und -Ausgänge erreicht. Sie können damit das Prozessabbild bis auf Bitebene aufteilen und getrennt adressieren.

Prozessdaten- zuordnung

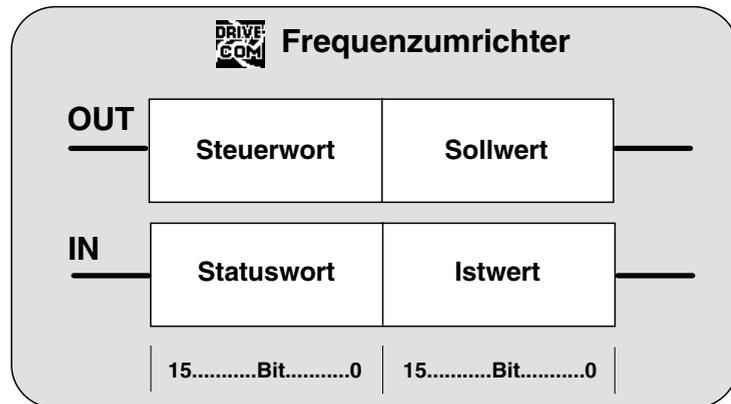
Die Prozessdatenreferenzliste (PDRL = process data reference list) legt die Zuordnung der Ein- und Ausgangsdaten zu der zugehörigen Speicherposition im MPM fest. Damit können Sie sämtliche Ein- und Ausgänge frei im Adressraum des Host verteilen und dort darauf zugreifen. Dies ist ein umfangreicher Mechanismus, um die Daten aller Teilnehmer funktional oder speicheroptimiert zu gruppieren.

Diese Prozessdatenverknüpfung bildet die Basis für eine Prozessdatenvorverarbeitung auf der Anschaltbaugruppe.

Vorteile der anwenderdefinierten Prozessdaten

Die Definition und Zuordnung von Prozessdatenbeschreibungen (PDD) hat wesentliche Vorteile.

1. Das Hinzufügen oder Entfernen eines Teilnehmers hat keine Verschiebung der bestehenden Adressen im MPM und damit im Anwenderprogramm zur Folge, sondern lediglich eine Änderung der Prozessdatenreferenzliste (PDRL).
2. Einige Geräte verfügen über einen E/A-Kanal, der mehrere logische Funktionen enthält. Zum Beispiel ist das 32-Bit-Ausgangswort eines INTERBUS-kompatiblen Frequenzumrichters in Steuerwort und Sollwert unterteilt. Daher bietet es sich an, dieses Doppelwort in die entsprechenden funktionellen Einheiten zu untergliedern.



5150A015

Bild 1-14 Prozessdatenkanal eines Frequenzumrichters

3. Die Prozessdaten können anschließend frei im Adressraum des Steuerungssystems adressiert werden und so beispielsweise in funktionelle Gruppen zusammengefasst werden. Darüber hinaus besteht die Möglichkeit, über die Definition der Prozessdatenreferenzliste eine Broadcast-Funktion zu realisieren. Eine Sollwertvorgabe belegt dann nur noch eine einzige Adresse im Host-System. Die INTERBUS-Anschaltbaugruppe schreibt den Sollwert vom Host-System auf die anderen im System vorhandenen Umrichter. Neben Prozessdatenbeschreibungen über ein Doppelwort, Wort oder Byte,

sind Prozessdatenbeschreibungen über ein Bit möglich. Dies bietet die Möglichkeit, Statusbits verschiedener Geräte in einem einzigen Wort im Host-System zusammenzufassen und damit durch einen einfachen Wortzugriff schnell und übersichtlich viele Gerätestatus darzustellen.

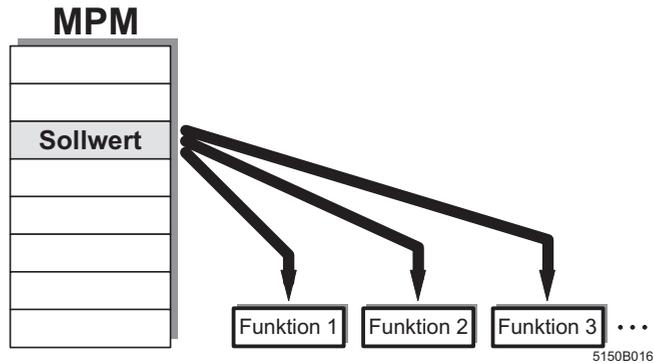


Bild 1-15 Broadcast

4. Darüber hinaus werden Prozessdatenbeschreibungen über ein Bit für die Durchgangsverbindungen benötigt. Eine Durchgangsverbindung bedeutet, dass Ausgangsbits direkt von der Anschaltbaugruppe gesetzt werden. Dies geschieht in Abhängigkeit von zugeordneten Eingangsbits, also unabhängig vom Steuerungsprogramm. Dabei wird die Programmverarbeitungszeit des Anwendungsprogramms ausgeklammert. Die Reaktionsgeschwindigkeit wird auf die INTERBUS-Antwortzeit reduziert.

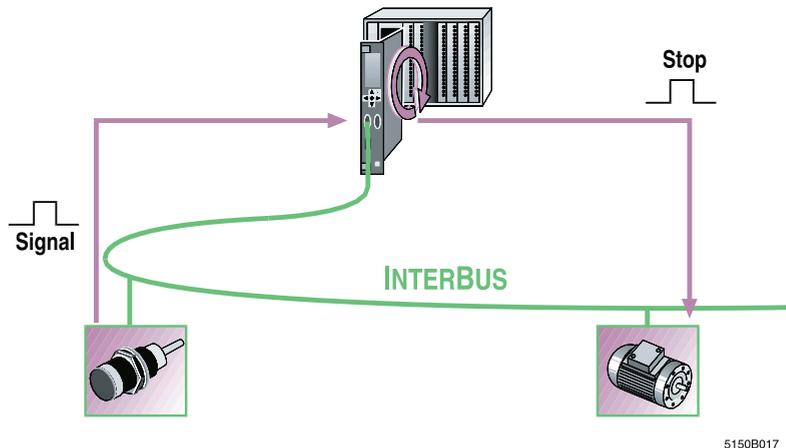


Bild 1-16 Durchgangsverbindung

Automatische Adresszuweisung von Prozessdaten

Es gibt drei Situationen, in denen automatisch Prozessdatenbeschreibungen (PDD) eingerichtet und Prozessdaten im MPM zugeordnet werden.

1. Im Test-Mode der Firmware.
2. Wenn Sie den Dienst „Create_Configuration“ ausführen.
3. Wenn Sie das Laden eines Konfigurationsrahmens mit dem Dienst „Terminate_Load_Configuration“ abschließen. Der Parameter *Default_Parameter* muss auf einen der Werte 0001_{hex} oder 0003_{hex} gesetzt sein.

Default-PDD

Der Prozessdatenkanal jedes Teilnehmers wird durch die Firmware standardmäßig durch sogenannte Default-Prozessdatenbeschreibungen (PDD) beschrieben. Die E/A-Daten der Teilnehmer verkörpern dann ein (für IN oder OUT) oder zwei (für IN und OUT) Prozessdatenbeschreibungen. Diese PDDs enthalten Größe, Datenrichtung und Namen der Teilnehmer sowie den Default-Index für IN- oder OUT-Datenrichtung.

Default-PDRL

Die Prozessdaten werden von der Firmware ab der Basisadresse für den IN- und OUT-Bereich im MPM möglichst lückenlos und entsprechend der physikalischen Reihenfolge der Teilnehmer im Ring vergeben. Für die Belegung der Prozessdatenadressen im MPM gilt Folgendes:

- Teilnehmer mit einem oder mehreren Datenworten können nur auf geraden MPM-Adressen liegen.
- Byte-Module können auf geraden oder ungeraden Byte-Adressen beginnen.
- Nibble-Module starten auf dem oberen oder unteren Nibble eines jeden Bytes.

1.4.7 Zustandsbeschreibung während der Konfiguration

Die Konfiguration besteht aus drei Phasen:

- Phase 1: Konfigurationsrahmen laden
- Phase 2: Prozessdatenbeschreibungen (PDD) definieren und
- Phase 3: Prozessdaten zuordnen

Die Zustände PDDL_LOADING (Laden der Prozessdatenbeschreibungsliste) und PDRL_LOADING (Laden der Prozessdatenreferenzliste) müssen nicht durchlaufen werden. Diese Listen werden bei Abschluss der Phase CFG_LOADING mit dem Dienst „Terminate_Load_Configuration“ automatisch aus den Informationen des geladenen Konfigurationsrahmens erzeugt, wenn die entsprechenden Parameter gesetzt wurden.

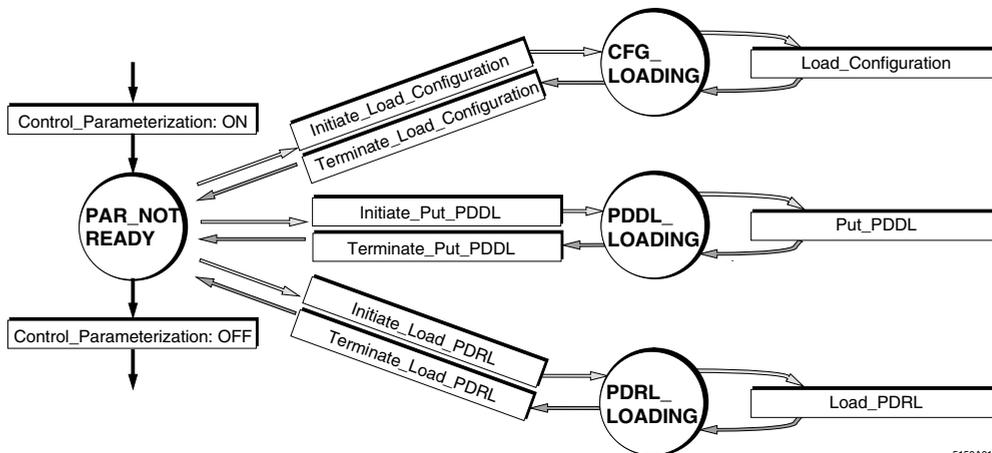


Bild 1-17 Dienste zur Konfiguration der Anschaltbaugruppe

Phase 1: Laden des Konfigurationsrahmens

Im Konfigurationsrahmen sind Eigenschaften der Busteilnehmer hinterlegt. Dazu gehören z. B.:

- Teilnehmerart: digital/analog, Anzahl der Kanäle
- ID-Code
- Position des Teilnehmers im INTERBUS

Der Anwender legt mit dem Laden des Konfigurationsrahmens die Auswahl der Module und die Bustopologie sowie die Definition funktionell zusammengehörender Busteile/-segmente als Gruppe fest. Alle Ladevorgänge (auch die der folgenden Listen) werden durch einen Initialisierungsdienst eingeleitet und durch einen beendenden Dienst abgeschlossen.

Phase 2: Laden der Prozessdatenbeschreibungsliste

Eine Prozessdatenbeschreibungsliste wird angelegt, um eine optimal auf die Anwendung zugeschnittene Verwaltung der Modul-Ein-/Ausgänge und eine speicheroptimierte Adressierung durchzuführen. Außerdem können Beschreibungen für besonders zeitkritische Bitoperationen definiert werden, die von der Anschaltbaugruppe selbstständig und ohne Steuerung durch das Anwendungsprogramm ausgeführt werden, z. B. für Positionieraufgaben.

Prozessdatenbeschreibungen (PDD) müssen nicht zwingend angelegt werden, da die Anschaltbaugruppe automatisch Default-PDDs definiert. Für ein digitales Eingangsmodul mit 16 Eingängen (1 Wort) wird z. B. automatisch eine PDD über ein Wort angelegt. Mit Hilfe einer selbstdefinierten Prozessdatenbeschreibungsliste kann der Anwender aus diesem Wort beispielsweise zwei PDDs über jeweils ein Byte erstellen, die frei im Adressraum des Host-Systems adressiert werden können.

Phase 3: Erstellen einer Prozessdatenreferenzliste

Die Prozessdatenreferenzliste legt die Beziehungen der Prozessdaten untereinander fest. Damit kann der Anwender sämtliche Ein- und Ausgänge in beliebiger Kombination frei im Adressraum des Host-Systems adressieren und auf Änderungen und Erweiterungen flexibel reagieren. Eine Erweiterung hat keine Änderung bestehender Adressen im Anwendungsprogramm zur Folge, sondern eine Änderung der Prozessdatenreferenzliste. Diese sogenannte logische Adressierung ist nicht zwingend erforderlich, aber aufgrund der gewonnenen Flexibilität unbedingt zu empfehlen.

1.5 Funktionen zur Konfigurationsrahmen-Verwaltung

Die Konfigurationsrahmen-Verwaltung stellt Dienste zur Verfügung mit denen Sie die Konfiguration des Bussystems verwalten und verändern können. Hierzu zählen:

- das Laden, Auslesen und Löschen von Konfigurationsrahmen,
- das Erzeugen eines Konfigurationsrahmens anhand der aktuellen Konfiguration und
- ein Umschalten zwischen verschiedenen Konfigurationsrahmen.

Es gibt zwei unterschiedliche Verfahren, um einen Konfigurationsrahmen auf der Anschaltbaugruppe zu hinterlegen:

- Automatisches Erzeugen eines Konfigurationsrahmens durch die Anschaltbaugruppe anhand des physikalischen Busaufbaus,
- Manuelles Erstellen eines optimierten Konfigurationsrahmens durch den Anwender.

Sowohl ein automatisch erzeugter, als auch ein vom Anwender geladener Konfigurationsrahmen kann in der Parametrierungsphase der Anschaltbaugruppe durch Überschreiben der jeweiligen Einträge verändert werden. Ein Ändern bestimmter Einträge (z. B. ID-Code) des Konfigurationsrahmens kann Rückwirkungen auf den Prozessdaten- und Parameterkanal haben. Die Anschaltbaugruppe überprüft, ob diese Rückwirkungen so gewichtig sind, dass eine zuvor projektierte Prozessdatenbeschreibungsliste (PDDL), die Kommunikationsbeziehungsliste (CRL) oder die Prozessdatenreferenzliste (PDRL) für ungültig erklärt werden müssen.

1.5.1 Automatisches Laden der Konfiguration



Da beim automatischen Laden der Konfiguration nicht die gleichen Diagnose-Möglichkeiten bestehen wie beim gezielten Aufschalten einer vorgegeben Konfiguration, sollte es nur zu Testzwecken verwendet werden.

Nach Aufruf des Dienstes „Create_Configuration“ (0710_{hex}) erzeugt die Anschaltbaugruppe automatisch einen Konfigurationsrahmen anhand des im Moment angeschlossenen Busaufbaus. Dieser Konfigurationsrahmen wird im Konfigurationsverzeichnis gespeichert, und zwar unter der beim

Aufruf des Dienstes angegebenen Frame-Referenz. Falls unter dieser Frame-Referenz bereits ein Konfigurationsrahmen existiert, wird dieser überschrieben.

Bei diesem Verfahren werden alle dem Konfigurationsrahmen zugeordneten Beschreibungen (PDRL, CRL usw.) und Betriebsparameter automatisch erzeugt oder eingestellt. Damit kann die erzeugte Konfiguration mit dem Dienst „Read_Configuration“ (0309_{hex}) ausgelesen und mit der Planung verglichen werden.

1.5.2 Manuelles Erzeugen einer optimierten Konfiguration

Die zweite Möglichkeit ist das Laden eines neuen Konfigurationsrahmens durch den Anwender. Sie können dem Konfigurationsrahmen eine beliebige Frame-Referenz zuordnen. Falls unter dieser Frame-Referenz bereits ein Konfigurationsrahmen existiert, wird dieser überschrieben.

Führen Sie das Laden einer Konfiguration schrittweise durch, da die Anzahl der Parameter eines Dienstes begrenzt ist:

- Leiten Sie das Laden mit dem Dienst „Initiate_Load_Configuration“ (0306_{hex}) ein.
- Übertragen Sie die Konfigurationsdaten durch mehrmaliges Aufrufen des Dienstes „Load_Configuration“ (0307_{hex}) zur Anschaltbaugruppe.
- Beenden Sie das Laden mit dem Dienst „Terminate_Load_Configuration“ (0308_{hex}). Dieser Dienst überprüft außerdem die geladenen Daten in ihrer Gesamtheit. Wird kein Fehler festgestellt, dann werden die Daten übernommen. Sind die Daten fehlerhaft, dann wird der Dienst mit der entsprechenden Fehlermeldung negativ quittiert. Zusätzlich können Sie beim Beenden des Ladevorgangs mit dem Parameter *Default_Parameter* angeben, ob Sie den Prozessdaten- und/oder den Parameterkanal anhand des geladenen Konfigurationsrahmens automatisch generieren lassen wollen.

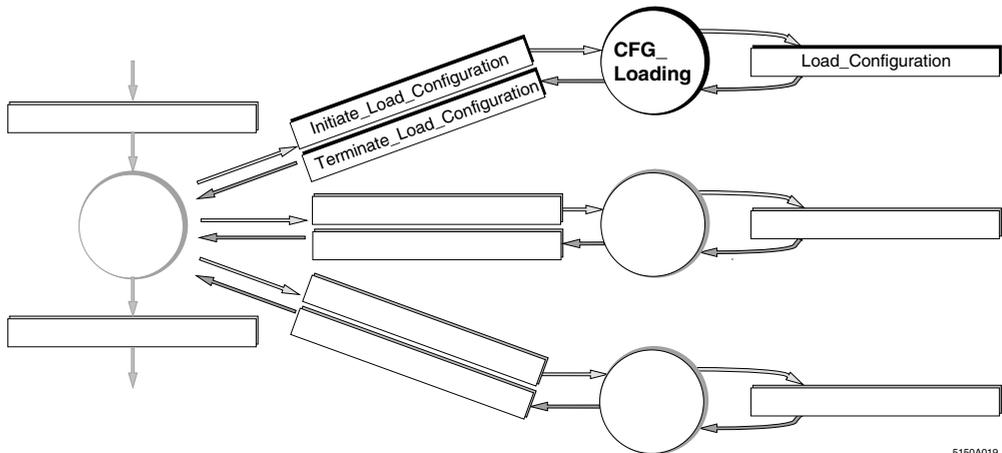


Bild 1-18 Laden eines Konfigurationsrahmens

Sie können die Konfigurationsdaten teilnehmerorientiert, listenorientiert oder in einer Mischform aus beiden zur Anschaltbaugruppe übertragen.

Um SPS-Funktionsbausteine in ihrer Wirkungsweise zu unterstützen und die Anzahl der Parameter in dem Lade-Dienst so gering wie möglich zu halten, bietet die Firmware zusätzlich den Dienst „Complete_Load_Configuration“ (030A_{hex}) an, bei welchem immer eine mit dem Parameter *Used_Attributes* selektierbare (aber immer vollständige) Spalte des Konfigurationsrahmens geladen wird. Äquivalent dazu gibt es auch den Dienst „Complete_Read_Configuration“ (030B_{hex}).

- PDDs dürfen sich innerhalb des internen Adressraums eines PD-Teilnehmers überschneiden.
- IN-und OUT-PDDs dürfen nicht den internen Adressraum des jeweiligen PD-Teilnehmers überschreiten.
- Die Länge des PDDs muss dem Datentyp entsprechen, außer es handelt sich um einen Datentyp variabler Länge.
- Bit-Strings dürfen durch ihre Bitadresse und Länge Byte-Grenzen nicht überschreiten: (Bitadresse + Länge \geq 8).

1.6.2 Definition der Prozessdatenbeschreibungslisten

Prozessdaten sind teilnehmerorientiert. Für jeden INTERBUS-Teilnehmer können mehrere Prozessdaten definiert werden, die in die Prozessdatenbeschreibungsliste des Teilnehmers eingetragen werden. Dazu wird der Dienst „Put_Process_Data_Description_List“ (0321_{hex}) benutzt. Dieser Dienst definiert alle Prozessdatenbeschreibungen eines Teilnehmers. Der Dienstaufbau ist in der folgenden Abbildung dargestellt.

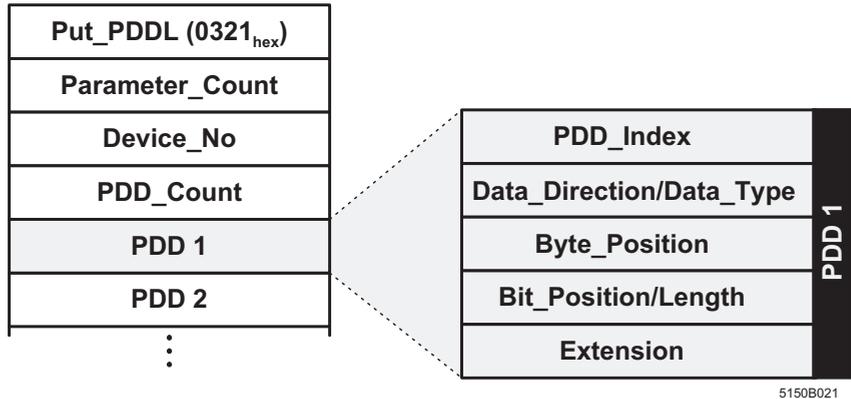


Bild 1-20 Dienstaufbau Prozessdatenbeschreibungsliste

Die einzelnen Parameter haben folgende Bedeutung:

- PDD-Index: Kennzeichnet die Prozessdatenbeschreibung (PDD). Zulässig sind Werte im Bereich von 1000_{hex} bis BFFF_{hex}, mit Ausnahme der Werte 6010_{hex} (Default-PDD-Index für Eingänge) und 6011_{hex} (Default-PDD-Index für Ausgänge).
- Data_Direction: Datenrichtung; Vergeben Sie 0C_{hex} für Eingänge und 0D_{hex} für Ausgänge.
- Data_Type: Daten-Typ; Vergeben Sie 0A_{hex} für Byte-String-Prozessdaten und 0F_{hex} für Bit-String-Prozessdaten.
- Byte_Position: Byte-Offset
- Bit_Position: Bit-Offset
- Length: String-Länge (Ein Bit-String darf keine Byte-Grenze überschreiten!)
- Extension: Erweiterung, 0000_{hex}

Der PDD-Index kennzeichnet, ob es sich um ein anwenderdefiniertes Prozessdatum oder ein Default-Prozessdatum handelt. Eine Default-PDD für Eingangsdaten besitzt immer den Index 6010_{hex}, eine Default-PDD für Ausgangsdaten den Index 6011_{hex}. Innerhalb der Typendefinition wird der Prozessdatentyp festgelegt. Prozessdaten können vom Typ Byte-String oder Bit-String sein. Ein Wort-Prozessdatum wird z. B. aus zwei Byte-Prozessdaten gebildet, wobei die Parameterlänge 02_{hex} ist. Die Parameter *Byte_Position* und *Bit_Position* geben jeweils den Byte- und Bit-Offset an. Innerhalb der Erweiterung kann ein Kommentar als ASCII-Text vergeben werden.

Die folgende Abbildung zeigt ein Beispiel von Byte-IN- und OUT-Prozessdaten sowie ein Bit-Prozessausgangsdatum von zwei Bit Länge.



Bild 1-21 Byte-IN- und OUT-Prozessdaten

1.6.3 Projektieren der Prozessdatenzuordnung

Gehen Sie folgendermaßen vor, wenn Sie die Prozessdatenzuordnung projektieren:

- Leiten Sie das Laden mit dem Dienst „Initiate_Load_Process_Data_Reference_List“ (0324_{hex}, vgl. Seite 2-80) ein.
- Übertragen Sie die Liste durch mehrmaliges Aufrufen des Dienstes „Load_Process_Data_Reference_List“ (0325_{hex}, vgl. Seite 2-82) zur Anschaltbaugruppe.
- Beenden Sie das Laden mit dem Dienst „Terminate_Load_Process_Data_Reference_List“ (0326_{hex}, vgl. Seite 2-86).

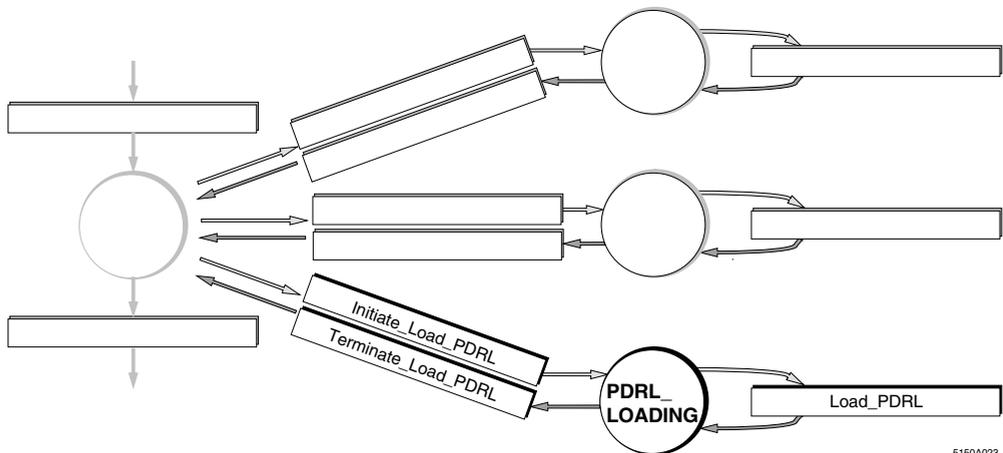
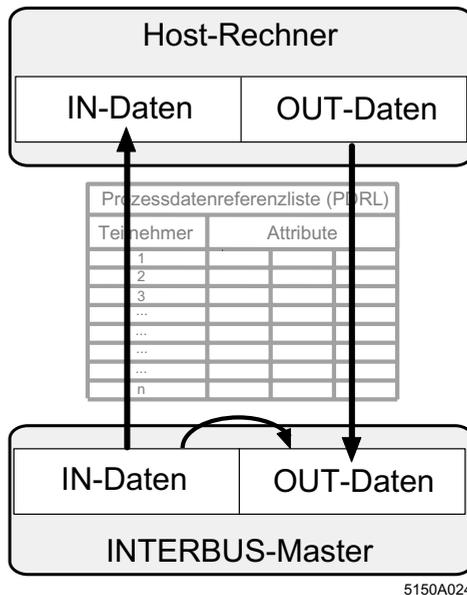


Bild 1-22 Laden der Prozessdatenreferenzliste

Es gibt drei Kombinationsmöglichkeiten zur Verknüpfung der Prozessdaten:

1. Abbildung eines Eingangs-Prozessdatums auf eine Eingangsadresse des Host-Systems.
2. Abbildung einer Ausgangsadresse des Host-Systems auf ein Ausgangs-Prozessdatum.
3. Direkte Abbildung eines Eingangs-Prozessdatums auf ein Ausgangs-Prozessdatum (Durchgangsverbindung).



5150A024

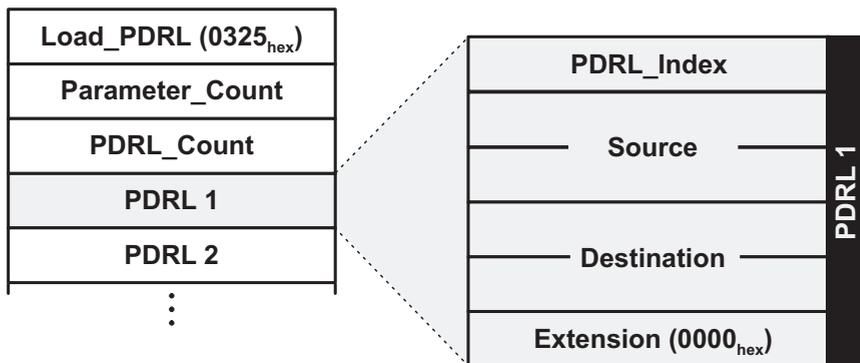
Bild 1-23 Die Prozessdatenzuordnung

PDRL-Einträge definieren

Werden PDRL-Einträge definiert, gelten folgende Regeln:

- Prinzipiell werden nur Einträge akzeptiert, die einer von den oben beschriebenen Kombinationen entsprechen.
- Eingangs(-IN)-Prozessdatenbeschreibungen oder -adressen dürfen mehrfach als Quelle in die PDRL eingetragen werden.
- Gleiche oder sich überschneidende Ausgangs(-OUT)-Prozessdatenbeschreibungen oder -adressen dürfen nicht mehrfach als Ziel in der PDRL auftreten. Eine Ausnahme besteht, wenn eine der beiden Prozessdatenbeschreibungen ein Prozessdaten-Bitstring ist. Dieser überschreibt dann das andere Objekt. Damit ist es möglich einzelne Bits oder Bitstrings in unterlagerte Objekte einzublenden.

Jeder Eintrag belegt insgesamt sechs Worte innerhalb des Dienstes „Load_Process_Data_Reference_List“ (0325_{hex}). Die Struktur eines solchen Eintrags innerhalb des Dienstes ist nachfolgend dargestellt:



5150C025

Bild 1-24 Struktur des Dienstes „Load_Process_Data_Reference_List“

Tabelle 1-7 Parameter der Prozessdatenreferenzliste

Parameter	Eingangsdaten	Ausgangsdaten	Durchgangsdaten	—
1 PDRL-Index	0001 ... 3FFF	4000 ... 7FFF	8000 ... BFFF	C000 ... FFFF
2 Source	Teilnehmer-Nr.	Adresse	Teilnehmer-Nr.	—
3 (Quelle)	Index	Bit-Position; Konsistenz	Index	
4 Destination	Adresse	Teilnehmer-Nr.	Teilnehmer-Nr.	—
5 (Ziel)	Bit-Position; Konsistenz	Index	Index	
6 Extension (Erweiterung)	reserviert (0000 _{hex})			

1.7 Funktionen zur Fehler- und Diagnose-Verwaltung

Fehlerverwaltung	<p>Die Fehlerverwaltung ist für die Fehlererkennung, -meldung und -bearbeitung verantwortlich. Hierzu zählen:</p> <ul style="list-style-type: none"> – Das Melden von während des Betriebs aufgetretenen Konfigurationsfehlern – Das Melden von Systemfehlern (Hardware und Firmware) – Die Möglichkeit des Meldens einzelner Busfehler – Das Bereitstellen von Diensten zum <ul style="list-style-type: none"> – nachträglichen Auslesen von aufgetretenen Fehlern, – Quittieren von Gerätestatusmeldungen und – Auslesen der Geräte-Status. <p>Die Fehler lassen sich in verschiedene Klassen einteilen:</p>
Anwenderfehler	<p>Anwenderfehler sind Fehler, die der Anwender beim Absetzen eines Dienstes gemacht hat (z. B. illegale Parameter, Zustandskonflikt, etc.). Diese Fehler führen zu einer negativen Quittierung des Dienstes, aber nicht zu einer Änderung des Buszustandes (z. B. zu einem Stopp der Datenzyklen).</p>
Systemfehler	<p>Systemfehler sind Hardware- oder Firmware-Fehler, die zu einer Änderung des Buszustandes führen können. Zu den Systemfehlern zählen auch alle Fehler, die das Betriebssystem meldet. Handelt es sich um einen schwerwiegenden Systemfehler, so führt dies neben einem Bus-Reset zu einer Fehlermeldung und in einen Zustand der Anschaltbaugruppe, in welchem keine Dienste außer dem Dienst „Reset_Controller_Board“ (0956_{hex}, vgl. Seite 2-110) mehr bearbeitet werden.</p>
Gerätestatus-meldungen	<p>Gerätestatusmeldungen sind Ereignisse, die an den INTERBUS-Teilnehmern auftreten und durch die Nachricht „Device_Fail_Indication“ (5340_{hex}, vgl. Seite 2-206) angezeigt werden. Diese Statusmeldungen führen nicht zu einer Änderung des Buszustandes und können vom Anwender mit dem Dienst „Control_Device_Function“ (0714_{hex}, vgl. Seite 2-107) bestätigt werden.</p> <p>Zu Diagnose-Zwecken besteht die Möglichkeit, auswählbare Statusinformationen aller aktiven Teilnehmer mit dem Dienst „Read_Device_State“ (0315_{hex}, vgl. Seite 2-116) auszulesen.</p>

Buseinzelfehler

Buseinzelfehler sind Fehler, die die Datensicherungsschicht erkennt und in einer Busfehlerstatistik aufbereitet. Diese Buseinzelfehler bewirken eine Fehleranalyse (ID-Zyklus), führen aber nicht zu einer Änderung des Buszustandes, solange es Einzelfehler bleiben.

Busfehler

Busfehler sind Fehler, die z. B. durch eine Konfigurationsänderung, Kabelbruch oder zu starke Störungen ausgelöst werden. Diese Fehler führen zu einem Stopp der Datenzyklen (sofern vorher welche gefahren wurden) und einer Fehlermeldung mittels der Nachricht „Bus_Error_Indication“ (6342_{hex}). Die genaue Fehlerursache eines Busfehlers wird nicht automatisch gemeldet, sondern muss durch den Anwender mit dem Dienst „Get_Error_Info“ (0316_{hex}) angefordert werden. Dieser Dienst liest die Fehlerursache aus einem Fehlerpuffer aus. Ein Fehlerpuffer kann sich in einem der drei folgenden Zustände befinden:

leer	Kein Fehler ist aufgetreten.
noch nicht analysiert	Die Fehleranalyse ist noch nicht beendet.
analysiert	Bis zu zehn Fehler-Codes inklusive Fehlerort können ausgelesen werden.

Durch Fehlerbehebung oder geeignete Schalt-Dienste (z. B. Abschalten des defekten Segments) kann im Zustand „analysiert“ die Fehlerursache beseitigt werden. Danach können mit dem Dienst „Start_Data_Transfer“ (0701_{hex}) neue Datenzyklen angestoßen werden.

Diagnose

Die Diagnose ist eine Komponente der Fehlerverwaltung, die die Dienste der Diagnose- und Statistik-Funktionalitäten bereitstellt.

Die statistische Diagnose zur Überwachung der Busqualität läuft automatisch im Hintergrund ab. Die Diagnose-Daten beziehen sich immer auf den aktuell eingeschalteten Konfigurationsrahmen. Der Anwender kann diese Daten mit dem Dienst „Get_Diag_Info“ (032B_{hex}) auslesen.

Zusätzlich kann der Anwender mit dem Dienst „Control_Statistics“ (030F_{hex}) die statistische Diagnose zurücksetzen.

1.8 Funktionen zur Bussteuerung

Die Bussteuerung beinhaltet Dienste, die während des Betriebes direkt den Systemstatus beeinflussen. Hierzu zählen die Aktivierung oder Deaktivierung verschiedener Konfigurationsrahmen, das Starten oder Stoppen des Datentransports sowie der Reset/Alarm-Stopp der Baugruppe. Ausgenommen von dieser Definition sind die Konfigurationsdienste mit Zustandsänderung.

Ein weiterer Bereich, der zur Bussteuerung gezählt wird, beinhaltet die Schaltfunktionen für Gruppen und Alternativen.

1.8.1 Dienste zur Systemzustandssteuerung

Während des Busbetriebs kann mittels unterschiedlicher Dienste zwischen den Zuständen „PAR_READY“, „ACTIVE“ und „RUN“ umgeschaltet werden.

Im Zustand „PAR_READY“ liegt mindestens ein kompletter Konfigurationsrahmen vor. Mit dem Dienst „Activate_Configuration“ (0711_{hex}) wird ein Konfigurationsrahmen aktiviert. Dieser Dienst kontrolliert die angeschlossene Konfiguration mit der im Konfigurationsrahmen vorgegebenen Konfiguration. Wurden keine Abweichungen festgestellt, werden vom INTERBUS ID-Zyklen ausgeführt. Mit „Deactivate_Configuration“ (0712_{hex}) kann dieser Konfigurationsrahmen wieder ausgeschaltet werden.

Um den Datentransfer auf dem Bussystem zu starten, muss aus dem Zustand „ACTIVE“ der Dienst „Start_Data_Transfer“ (0701_{hex}) ausgeführt werden. Mit dem Dienst „Stop_Data_Transfer“ (0702_{hex}) wird der zyklische Datenverkehr wieder ausgeschaltet.

Treten beim Aktivieren und/oder Starten des Bussystems Fehler auf, wechselt die Zustandsmaschine in den Zustand „BUS_FAIL“. Nach Beseitigung der Fehlerursache können von hier aus, die gewünschten Dienste erneut aufgerufen werden.

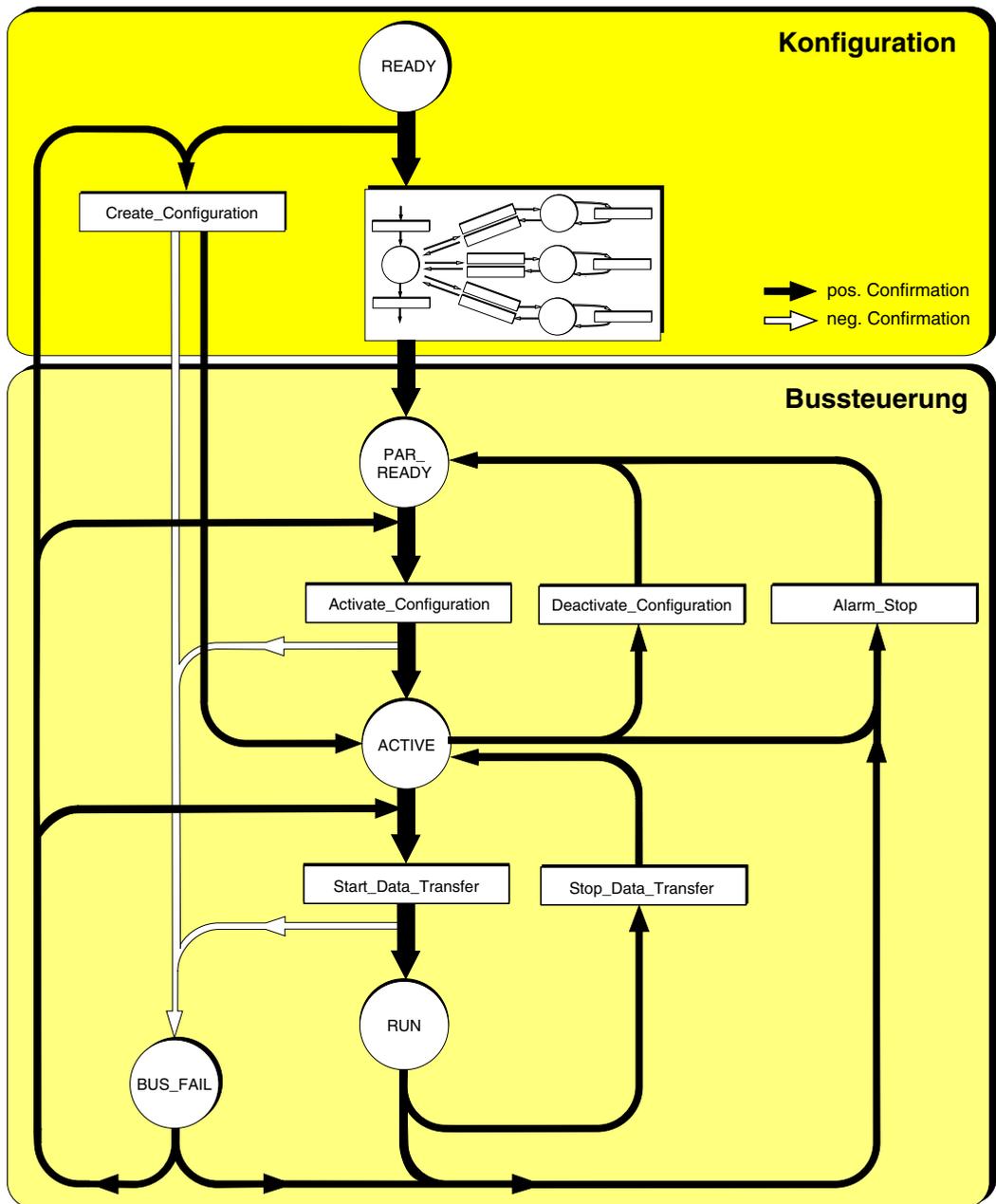


Bild 1-25 Dienste zur Systemzustandssteuerung

1.8.2 Dienste zum Schalten von Gruppen und Alternativen

Während des Busbetriebes können einzelne Teilnehmer, Bussegmente, logische Gruppen und/oder alternative Teilnehmergruppen über den Dienst „Control_Active_Configuration“ (0713_{hex}) hinzu- oder abgeschaltet werden. Als Parameter wird eine Liste der betreffenden Teilnehmer übergeben. Dieser Schaltvorgang betrifft die in der Liste aufgeführten und alle davon abhängigen Teilnehmer, d. h. alle INTERBUS-Teilnehmer,

- die zum gleichen Bussegment gehören,
- die zur gleichen logischen Gruppe gehören oder
- die im physikalischen Ring hinter einem in der Liste aufgeführten INTERBUS-Teilnehmer liegen.

1.9 Allgemeine Funktionen der Firmware

Zu den allgemeinen Funktionen der Firmware zählen u. a. Dienste zum Umgang mit dem Parametrierungsspeicher der Anschaltbaugruppe.

1.9.1 Umgang mit dem Parametrierungsspeicher

Über diese Dienste können Sie den Parametrierungsspeicher formatieren, Dienstfolgen und Parametersätze, die im Arbeitsspeicher der Anschaltbaugruppe gespeichert sind auf den Parametrierungsspeicher übertragen, einzelne Dateien auf dem Parametrierungsspeicher öffnen oder schließen oder löschen. Sie können eine bestimmte Datei suchen und Daten in diese Datei schreiben oder aus dieser Datei herauslesen.



Beachten Sie bei Anschaltbaugruppen mit steckbarem Parametrierungsspeicher das folgende Verhalten der Firmware:

Neuer Parametrierungsspeicher

Wenn beim Booten einer Anschaltbaugruppe ein neuer Parametrierungsspeicher eingesetzt ist, wird der Parametrierungsspeicher formatiert. Das Display der Anschaltbaugruppe zeigt, soweit vorhanden, die Meldung 3030_{hex} . Beachten Sie, dass dieser Formatierungsvorgang je nach Größe des Parametrierungsspeichers bis zu 12 Minuten dauern kann.

Große Parametrierungsspeicher

Der Bootvorgang eines Controllers kann bis zu 10 Minuten dauern, wenn während dieses Vorganges ein voller großer Parametrierungsspeicher (z. B. 64 MB) eingesetzt ist. Während des Bootvorganges wird auf dem Display, soweit vorhanden, ein Wertebereich von 3100 bis 3200 durchgezählt. Dieser Zählvorgang des Counters wiederholt sich solange, bis der Bootvorgang des Controllers abgeschlossen ist.

Ablegen der IP-Adresse

- auf 1 MB-Speicherkarten:

1 MB-Parametrierungsspeicher unterstützen die Speicherung einer IP-Adresse nicht. Der Controller speichert die IP-Adresse bei Verwendung dieser Karten intern ab.

- auf Speicherkarten > 1 MB:

Parametrierungsspeicher > 1 MB unterstützen die Speicherung einer IP-Adresse auf diesen Karten.



Weitere Informationen finden Sie in den Handbüchern zu den Baugruppen.

1.9.2 Verhalten der Firmware bei bestimmten Anschaltbaugruppen-Kommandos

1.9.2.1 Debug oder Auto Debug auf der Anschaltbaugruppe

Wenn ein Fernbus-Teilnehmer einen SUP1 1-Chip besitzt, kann die Debug- und die Auto Debug-Funktion der Anschaltbaugruppe nicht genutzt werden.

Dieses Kapitel informiert Sie über

- die Aufgaben und den Aufruf von Diensten der Firmware
- die Parameter dieser Dienste

Dienste der Firmware	2-5
2.1 Übersicht	2-5
2.2 Hinweise zu den Dienstbeschreibungen	2-8
2.3 Dienste zum Parametrieren der Anschaltbaugruppe	2-11
2.3.1 Dienst „Control_Parameterization“	2-11
2.3.2 Dienst „Change_Exclusive_Rights“	2-13
2.3.3 Dienst „Set_Indication“	2-15
2.3.4 Dienst „Set_Value“	2-18
2.3.5 Dienst „Read_Value“	2-25
2.3.6 Dienst „Initiate_Load_Configuration“	2-28
2.3.7 Dienst „Load_Configuration“	2-31
2.3.8 Dienst „Terminate_Load_Configuration“	2-36
2.3.9 Dienst „Read_Configuration“	2-38
2.3.10 Dienst „Compare_Configuration“	2-51
2.3.11 Dienst „Complete_Load_Configuration“	2-54
2.3.12 Dienst „Complete_Read_Configuration“	2-57
2.3.13 Dienst „Delete_Configuration“	2-60
2.3.14 Dienst „Create_Configuration“	2-62
2.3.15 Dienst „Activate_Configuration“	2-64
2.3.16 Dienst „Deactivate_Configuration“	2-66
2.4 Dienste zum Definieren von Prozessdatenbeschreibungen	2-68
2.4.1 Dienst „Initiate_Put_Process_Data_Description_List“	2-68
2.4.2 Dienst „Put_Process_Data_Description_List“	2-70
2.4.3 Dienst „Terminate_Put_Process_Data_	
Description_List“	2-74
2.4.4 Dienst „Get_Process_Data_Description_List“	2-76
2.5 Dienste zum Zuordnen von Prozessdaten	2-80
2.5.1 Dienst „Initiate_Load_Process_Data_Reference_List“	2-80
2.5.2 Dienst „Load_Process_Data_Reference_List“	2-82

2.5.3	Dienst „Terminate_Load_Process_Data_Reference_List“	2-86
2.5.4	Dienst „Read_Process_Data_Reference_List“	2-88
2.5.5	Dienst „Compact_Load_Process_Data_Reference_List“	2-91
2.5.6	Dienst „Compact_Read_Process_Data_Reference_List“	2-95
2.6	Dienste für den direkten Zugriff auf den INTERBUS.....	2-98
2.6.1	Dienst „Start_Data_Transfer“	2-98
2.6.2	Dienst „Alarm_Stop“	2-100
2.6.3	Dienst „Stop_Data_Transfer“	2-102
2.6.4	Dienst „Control_Active_Configuration“	2-104
2.6.5	Dienst „Control_Device_Function“	2-107
2.6.6	Dienst „Reset_Controller_Board“	2-110
2.7	Dienste zur Diagnose.....	2-111
2.7.1	Dienst „Confirm_Diagnostics“	2-111
2.7.2	Dienst „Get_Error_Info“	2-113
2.7.3	Dienst „Read_Device_State“	2-116
2.7.4	Dienst „Get_Version_Info“	2-120
2.7.5	Dienst „Get_Diag_Info“	2-124
2.7.6	Dienst „Control_Statistics“	2-131
2.8	Dienste zum Anlegen von Funktionen	2-133
2.8.1	Dienst „Initiate_Load_Action_Object“	2-133
2.8.2	Dienst „Load_Action_Object“	2-135
2.8.3	Dienst „Terminate_Load_Action_Object“	2-137
2.8.4	Dienst „Read_Action_Object“	2-139
2.8.5	Dienst „Delete_Action_Object“	2-142
2.8.6	Dienst „Initiate_Load_Signal_Object“	2-144
2.8.7	Dienst „Load_Signal_Object“	2-146
2.8.8	Dienst „Terminate_Load_Signal_Object“	2-152
2.8.9	Dienst „Read_Signal_Object“	2-154
2.8.10	Dienst „Delete_Signal_Object“	2-158
2.8.11	Dienst „Initiate_Load_Event_Object“	2-160
2.8.12	Dienst „Load_Event_Object“	2-162
2.8.13	Dienst „Terminate_Load_Event_Object“	2-167

2.8.14	Dienst „Read_Event_Object“	2-169
2.8.15	Dienst „Delete_Event_Object“	2-172
2.9	Dienste für den Parametrierungsspeicher.....	2-174
2.9.1	Dienst „Program_Resident_Actions“.....	2-174
2.9.2	Dienst „Clear_Parameterization_Memory“.....	2-176
2.9.3	Dienst „File_Open“.....	2-179
2.9.4	Dienst „File_Close“	2-182
2.9.5	Dienst „File_Remove“	2-186
2.9.6	Dienst „File_Write“	2-189
2.9.7	Dienst „File_Seek“	2-192
2.9.8	Dienst „File_Read“	2-195
2.9.9	Dienst File_Remove_II.....	2-198
2.9.10	Dienst Get_Card_Information	2-200
2.10	Selbstständige Fehleranzeigen der Anschaltbaugruppe.....	2-204
2.10.1	Fehleranzeige „Fault“	2-204
2.10.2	Fehleranzeige „Lower_API_Fault“	2-205
2.10.3	Fehleranzeige „Device_Fail“	2-206
2.10.4	Fehleranzeige „Bus_Error“	2-207

2 Dienste der Firmware

2.1 Übersicht

Tabelle 2-1 Übersicht der Dienste (sortiert nach Kommando-Codes)

Code	Dienste	Seite
0140 _{hex}	Initiate_Load_Action_Object	2-133
0141 _{hex}	Load_Action_Object	2-135
0142 _{hex}	Terminate_Load_Action_Object	2-137
0143 _{hex}	Read_Action_Object	2-139
0144 _{hex}	Delete_Action_Object	2-142
0145 _{hex}	Initiate_Load_Signal_Object	2-144
0146 _{hex}	Load_Signal_Object	2-146
0147 _{hex}	Terminate_Load_Signal_Object	2-152
0148 _{hex}	Read_Signal_Object	2-154
0149 _{hex}	Delete_Signal_Object	2-158
014A _{hex}	Initiate_Load_Event_Object	2-160
014B _{hex}	Load_Event_Object	2-162
014C _{hex}	Terminate_Load_Event_Object	2-167
014D _{hex}	Read_Event_Object	2-169
014E _{hex}	Delete_Event_Object	2-172
014F _{hex}	Change_Exclusive_Rights	2-13
0152 _{hex}	Set_Indication	2-15
0158 _{hex}	Program_Resident_Actions	2-174
0159 _{hex}	Clear_Parameterization_Memory	2-176
015B _{hex}	File_Open	2-179
015C _{hex}	File_Close	2-182

Tabelle 2-1 Übersicht der Dienste (sortiert nach Kommando-Codes)

Code	Dienste	Seite
015D _{hex}	File_Remove	2-186
015E _{hex}	File_Write	2-189
015F _{hex}	File_Seek	2-192
0160 _{hex}	File_Read	2-195
0165 _{hex}	File_Remove_II	2-198
0166 _{hex}	Get_Card_Information	2-200
0306 _{hex}	Initiate_Load_Configuration	2-28
0307 _{hex}	Load_Configuration	2-31
0308 _{hex}	Terminate_Load_Configuration	2-36
0309 _{hex}	Read_Configuration	2-38
030A _{hex}	Complete_Load_Configuration	2-54
030B _{hex}	Complete_Read_Configuration	2-57
030C _{hex}	Delete_Configuration	2-60
030E _{hex}	Control_Parameterization	2-11
030F _{hex}	Control_Statistics	2-131
0314 _{hex}	Control_Device_Function_Not_Exclusive	2-107
0315 _{hex}	Read_Device_State	2-116
0316 _{hex}	Get_Error_Info	2-113
0317 _{hex}	Compare_Configuration	2-51
0320 _{hex}	Initiate_Put_Process_Data_Description_List	2-68
0321 _{hex}	Put_Process_Data_Description_List	2-70
0322 _{hex}	Terminate_Put_Process_Data_Description_List	2-74
0323 _{hex}	Get_Process_Data_Description_List	2-76
0324 _{hex}	Initiate_Load_Process_Data_Reference_List	2-80
0325 _{hex}	Load_Process_Data_Reference_List	2-82

Tabelle 2-1 Übersicht der Dienste (sortiert nach Kommando-Codes)

Code	Dienste	Seite
0326 _{hex}	Terminate_Load_Process_Data_Reference_List	2-86
0327 _{hex}	Read_Process_Data_Reference_List	2-88
0328 _{hex}	Compact_Load_Process_Data_Reference_List	2-91
0329 _{hex}	Compact_Read_Process_Data_Reference_List	2-95
032A _{hex}	Get_Version_Info	2-120
032B _{hex}	Get_Diag_Info	2-124
0351 _{hex}	Read_Value	2-25
0701 _{hex}	Start_Data_Transfer	2-98
0702 _{hex}	Stop_Data_Transfer	2-102
0710 _{hex}	Create_Configuration	2-62
0711 _{hex}	Activate_Configuration	2-64
0712 _{hex}	Deactivate_Configuration	2-66
0713 _{hex}	Control_Active_Configuration	2-104
0714 _{hex}	Control_Device_Function	2-107
0750 _{hex}	Set_Value	2-18
0760 _{hex}	Confirm_Diagnostics	2-111
0956 _{hex}	Reset_Controller_Board	2-110
1303 _{hex}	Alarm_Stop	2-100

Tabelle 2-2 Selbständige Fehleranzeigen (Indications)

Code	Dienste	Seite
4341 _{hex}	Fault	2-204
5340 _{hex}	Device_State	2-206
6342 _{hex}	Bus_Error	2-207
4B58 _{hex}	Lower_API_Fault	2-205

2.2 Hinweise zu den Dienstbeschreibungen

Benutzung der Dienste

Das Benutzen eines Dienstes umfasst das Absenden eines Requests (Dienst Anforderung) und die Auswertung der Confirmation (Dienst quittung).

Die Codes eines Requests und der darauf folgenden Confirmation unterscheiden sich in der binären Darstellungsform nur im Bit 15. Das Bit 15 einer Confirmation ist immer gesetzt.

D. h. in hexadezimaler Darstellung ist der Code einer Confirmation immer um den Betrag 8000_{hex} größer als der Code des vorausgegangenen Requests.

Beispiel

„Start_Data_Transfer“

Request:

„Start_Data_Transfer_Request“ 0701_{hex}

Confirmation:

„Start_Data_Transfer_Confirmation“ $8701_{\text{hex}} = 0701_{\text{hex}} + 8000_{\text{hex}}$

– Parameter *Result* = 0000_{hex} ⇒ Dienst erfolgreich ausgeführt

– Parameter *Result* $\neq 0000_{\text{hex}}$ ⇒ Fehler bei Dienstauführung

Die Confirmation bestätigt in einer positiven Meldung die erfolgreiche Durchführung eines Dienstes und stellt evtl. angeforderte Daten zur Verfügung. In einer negativen Meldung zeigt die Confirmation den Fehler an, der bei der Durchführung des Dienstes aufgetreten ist.

Dem Parameter *Result* in der Confirmation entnehmen Sie, ob der Dienst erfolgreich ausgeführt wurde (Parameter *Result* = 0000_{hex}), oder ob ein Fehler aufgetreten ist (Parameter *Result* $\neq 0000_{\text{hex}}$ beschreibt die Fehlerursache).

Aufbau der Dienstbeschreibung

Requests und Confirmations bestehen aus Blöcken von Datenworten. Die in diesen Blöcken enthaltenen Parameter werden hexadezimal ($_{\text{hex}}$) oder binär ($_{\text{bin}}$) dargestellt.

Alle Dienstbeschreibungen sind nach dem folgenden Schema aufgebaut:

2.x.x Dienst „Name_des_Dienstes“

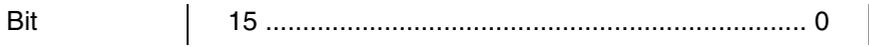
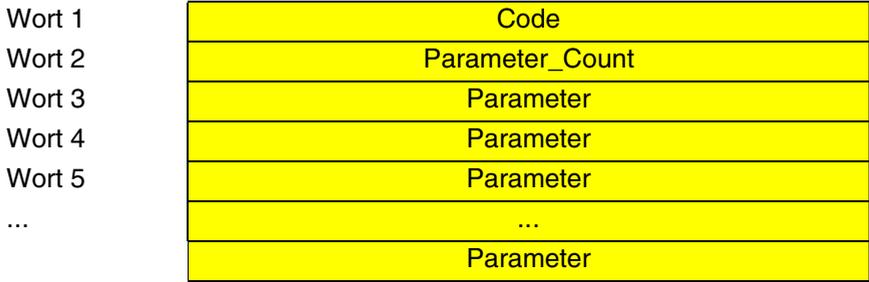
Aufgabe:

Beschreibt die Funktionalität des Dienstes.

Voraussetzung:

Alle Bedingungen, die für eine erfolgreiche Bearbeitung eines Dienstes vor dessen Aufruf erfüllt sein müssen.

Syntax: **Name_des_Dienstes_Request** **Code_{hex}**



Legende:

Code: 0xxx_{hex} Kommando-Code des Requests (hexadezimale Darstellung)

Parameter_Count: Anzahl der nachfolgenden Worte
 0000_{hex} falls der Request keine Parameter enthält.
 xxxx_{hex} sonst Länge des Parameter-Datensatzes (Anzahl der Parameterworte).

Parameter: Die Parameter werden einzeln beschrieben. Byteweise organisierte Parameter sind durch eine Spaltenlinie getrennt. Erstreckt sich ein Parameter über mehrere Datenworte, dann ist das durch eine Reihe mit 3 Punkten dargestellt.

Parameterblöcke: Parameterblöcke sind fett umrandet. Die einzelnen darin enthaltenen Parameter werden nachfolgend beschrieben.

2.3 Dienste zum Parametrieren der Anschaltbaugruppe

2.3.1 Dienst „Control_Parameterization“

Aufgabe:

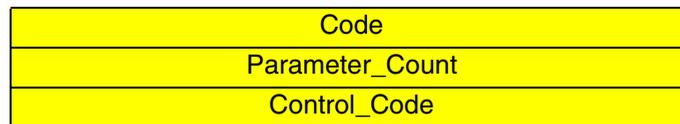
Dieser Dienst leitet die Parametrierungsphase ein oder beendet sie. Dies ist erforderlich, um ein definiertes Anlaufverhalten des INTERBUS zu garantieren. Während der Parametrierungsphase ist z. B. die Gültigkeit von ausgelesenen Objekten nicht garantiert. Wenn die Parametrierungsphase beendet ist, wird im MPM das Bit *MPM_Node_Parameterization_Ready* gesetzt. Hierdurch kann eine SPS im Anlauf erkennen, wann die auf dem Parametrierungsspeicher gespeicherte Parametrierungssequenz erfolgreich abgearbeitet worden ist.

Syntax:

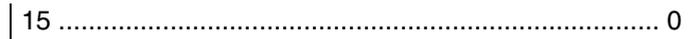
Control_Parameterization_Request

030E_{hex}

Wort 1
Wort 2
Wort 3



Bit



Legende:

- Code: 030E_{hex} Kommando-Code des Requests
- Parameter_Count: Anzahl der nachfolgenden Worte
0001_{hex} 1 Parameterwort
- Control_Code: Aufgabe des Dienstes
0001_{hex} leitet die Parametrierungsphase ein.
0000_{hex} beendet die Parametrierungsphase.

Syntax:

Control_Parameterization_Confirmation

830E_{hex}

Positive Meldung

Wort 1	Code
Wort 2	Parameter_Count
Wort 3	Result

Negative Meldung

Wort 1	Code
Wort 2	Parameter_Count
Wort 3	Result
Wort 4	Add_Error_Info

Bit | 15 0 |

Legende:

- Code: 830E_{hex} Meldungs-Code der Confirmation
- Parameter_Count: Anzahl der nachfolgenden Worte
 bei positiver Meldung:
 0001_{hex} 1 Parameterwort
 bei negativer Meldung:
 0002_{hex} 2 Parameterworte
- Result: Ergebnis der Dienstbearbeitung
 0000_{hex} zeigt eine positive Meldung an.
 Die Anschaltbaugruppe hat den Dienst erfolgreich bearbeitet.
 xxxx_{hex} zeigt eine negative Meldung an.
 Die Anschaltbaugruppe konnte den aufgerufenen Dienst nicht bearbeiten.
 Der Parameter *Result* enthält die Ursache, warum der Dienst nicht bearbeitet werden konnte.
- Add_Error_Info: Zusätzliche Informationen zur Fehlerursache

2.3.2 Dienst „Change_Exclusive_Rights“

Aufgabe: Mit diesem Dienst kann eine Anwendung das Recht, exklusive Dienste ausführen zu lassen, anfordern oder abgeben.

Voraussetzung: Die anfordernde Anwendung kann das Exklusiv-Recht nur bekommen, wenn dies zur Zeit keine andere Anwendung besitzt, da immer nur eine Anwendung Master sein darf.



Exklusive Dienste können nur von einer einzigen, dazu berechtigten Anwendung aufgerufen werden. In der binären Darstellung des Dienst-Codes (Kommando- und Meldungs-Code) eines exklusiven Dienstes ist das Bit 10 gesetzt.

Syntax: **Change_Exclusive_Rights_Request** **014F_{hex}**

Wort 1	Code
Wort 2	Parameter_Count
Wort 3	Function

Bit | 15 0 |

- Legende:
- Code: 014F_{hex} Kommando-Code des Requests
 - Parameter_Count: Anzahl der nachfolgenden Worte
0001_{hex} 1 Parameterwort
 - Function: Funktion des Dienstes
0000_{hex} gibt das Exklusiv-Recht ab.
0001_{hex} fordert das Exklusiv-Recht an.

Syntax: **Change_Exclusive_Rights_Confirmation** **814F_{hex}**

Positive Meldung

Wort 1	Code
Wort 2	Parameter_Count
Wort 3	Result

Negative Meldung

Wort 1	Code
Wort 2	Parameter_Count
Wort 3	Result
Wort 4	Add_Error_Info

Bit | 15 0 |

Legende:

Code:	814F _{hex} Meldungs-Code der Confirmation
Parameter_Count:	Anzahl der nachfolgenden Worte bei positiver Meldung: 0001 _{hex} 1 Parameterwort bei negativer Meldung: 0002 _{hex} 2 Parameterworte
Result:	Ergebnis der Dienstbearbeitung 0000 _{hex} zeigt eine positive Meldung an. Die Anschaltbaugruppe hat den Dienst erfolgreich bearbeitet. xxxx _{hex} zeigt eine negative Meldung an. Die Anschaltbaugruppe konnte den aufgerufenen Dienst nicht bearbeiten. Der Parameter <i>Result</i> enthält die Ursache, warum der Dienst nicht bearbeitet werden konnte.
Add_Error_Info:	Zusätzliche Informationen zur Fehlerursache

2.3.3 Dienst „Set_Indication“

Aufgabe: Mit diesem Dienst können Sie ausgewählte Fehleranzeigen (Indications) für bestimmte Schnittstellen freischalten oder sperren.

Syntax: **Set_Indication_Request** **0152_{hex}**

Wort 1	Code
Wort 2	Parameter_Count
Wort 3	Indication_Ability
Wort 4	Distribution
Wort 5	Indication_Code

Bit | 15 0 |

Legende:

Code: 0152_{hex} Kommando-Code des Requests

Parameter_Count: Anzahl der nachfolgenden Worte
0003_{hex} 3 Parameterworte

Indication_Ability: Gibt die Fehleranzeige frei oder sperrt sie für die angegebenen Schnittstellen (Parameter *Distribution*):
0001_{hex} schaltet Schnittstellen frei (enable).
0000_{hex} sperrt Schnittstellen (disable).

Distribution: Schnittstellen, für die die Einstellung geändert werden sollen
Der Parameter beschreibt ein 16 Bit breites Feld, wobei jedes Bit einer Schnittstelle entspricht. Setzen Sie bei den Schnittstellen, deren Einstellung Sie ändern wollen, das entsprechende Bit auf 1. Belegung des Parameters *Distribution*:

Bit 0 Diagnose-Display auf der Frontblende (falls vorhanden). Das Diagnose-Display gibt im Auslieferungszustand sämtliche Fehleranzeigen wieder.

Bit 1 Standard-Signal-Interface (SSGI)

Bit 2 Mailbox-Interface (MXI) zum Anwendungsprogramm

Bit 3 Diagnose-Schnittstelle auf der Frontblende der Anschaltbaugruppe

Bit 4 Mailbox-Interface (MXI) zum MPM-Node 0

- Bit 5 Mailbox-Interface (MXI) zum MPM-Node 2
- Bit 6 Mailbox-Interface (MXI) zum MPM-Node 3
- Bit 7...15 reserviert (immer 0_{bin}!)



Beachten Sie, dass das Bit 2 nur dann gesetzt werden muss, wenn die Bits 4, 5 und 6 nicht gesetzt wurden.

Indication_Code: Code der Fehleranzeige, die freigeschaltet oder gesperrt werden soll,
z. B. 4341_{hex} für „Fault_Indication“.

Syntax: **Set_Indication_Confirmation** **8152_{hex}**

Positive Meldung

Wort 1	Code
Wort 2	Parameter_Count
Wort 3	Result

Negative Meldung

Wort 1	Code
Wort 2	Parameter_Count
Wort 3	Result
Wort 4	Add_Error_Info

Bit | 15 0 |

Legende:

Code: 8152_{hex} Meldungs-Code der Confirmation

Parameter_Count: Anzahl der nachfolgenden Worte
bei positiver Meldung:
0001_{hex} 1 Parameterwort
bei negativer Meldung:
0002_{hex} 2 Parameterworte

Result: Ergebnis der Dienstbearbeitung
0000_{hex} zeigt eine positive Meldung an.
Die Anschaltbaugruppe hat den Dienst erfolgreich bearbeitet.
xxxx_{hex} zeigt eine negative Meldung an.
Die Anschaltbaugruppe konnte den aufgerufenen Dienst nicht bearbeiten.
Der Parameter *Result* enthält die Ursache, warum der Dienst nicht bearbeitet werden konnte.

Add_Error_Info: Zusätzliche Informationen zur Fehlerursache

2.3.4 Dienst „Set_Value“

Aufgabe:

Dieser Dienst weist den Systemparametern (Variablen) des INTERBUS neue Werte zu. Ein neuer Wert wird nur übernommen, wenn bei der Überprüfung des Wertebereiches kein Fehler erkannt wird.

Es sind folgende Systemparameter definiert:

Tabelle 2-3 Systemparameter

Variable ID	Systemparameter	Wert / Anmerkung
0101 _{hex}	Zustand des System-Managements	nur lesbar
0102 _{hex}	Adresse des Diagnose-Statusregisters	Der zulässige Adressbereich ist abhängig vom jeweiligen Host-System.
0103 _{hex}	Adresse des Diagnose-Parameterregisters	
0104 _{hex}	Inhalt / Wert des Diagnose-Statusregisters	nur lesbar
0105 _{hex}	Inhalt / Wert des Diagnose-Parameterregisters	nur lesbar
0108 _{hex}	Steuerung der Vorverarbeitungs-Task	0 = ausschalten 1 = freigeben 2 = sofortiges Rücksetzen der Ausgänge der Vorverarbeitung 3 = Rücksetzen der Ausgänge der Vorverarbeitung beim SYSFAIL der Steuerung
0109 _{hex}	Adresse des Kommunikations-Registers (nur für Siemens S5)	nur lesbar
010C _{hex}	Adresse des erweiterten Diagnose-Parameterregisters	Der zulässige Adressbereich ist abhängig vom jeweiligen Host-System.
010D _{hex}	Inhalt / Wert des erweiterten Diagnose-Parameterregisters	nur lesbar
0110 _{hex}	Slave_ID_Code (nur für Slave-Karten und Systemkoppler)	Geräte-Code und Prozessdaten-Länge des Slaves

Tabelle 2-3 Systemparameter

Variable ID	Systemparameter	Wert / Anmerkung
0117 _{hex}	Adresse des Slave-Diagnose-Statusregisters	Der zulässige Adressbereich ist abhängig vom jeweiligen Host-System.
0118 _{hex}	Adresse des Slave-Diagnose-Parameterregisters	
0119 _{hex}	Inhalt / Wert des Slave-Diagnose-Statusregisters	nur lesbar
011A _{hex}	Inhalt / Wert des Slave-Diagnose-Parameterregisters	nur lesbar
011D _{hex}	Slave-Konfigurationsdaten (nur für Slave-Karten und Systemkoppler)	Slave-Daten (Baudrate, ...)
020A _{hex}	Freischalten der optischen Diagnose nach dem Schalten von Schnittstellen	nur lesbar Value 1: Teilnehmer am Anfang der optischen Strecke Value 2: Teilnehmer am Ende der optischen Strecke
0306 _{hex}	Parametrierung des Host-Interface (speziell bei S5/S7)	Hier parametrieren Sie die Interrupt-Leitung zur SPS.
0407 _{hex}	Basis-Adressen für Default-Adressierung 1. Wort digital-in Basis-Adresse 2. Wort digital-out Basis-Adresse 3. Wort analog-in Basis-Adresse 4. Wort analog-out Basis-Adresse	Wenn die digital-in (oder -out) Basis-Adresse gleich der analog-in (oder -out) Basis-Adresse ist, dann werden digitale und analoge Geräte im Adressraum gemischt.
1101 _{hex}	Adresse des Standardfunktions-Startregisters	Der zulässige Adressbereich ist abhängig vom jeweiligen Host-System.
1102 _{hex}	Adresse des Standardfunktions-Statusregisters	
1103 _{hex} 1200 _{hex}	Adresse des Standardfunktions-Parameterregisters (Langwort) 1103 _{hex} nur eine Wortadresse angeben. Die Anschaltbaugruppe vergibt für das zweite Wort automatisch die nächstfolgende Wortadresse. 1200 _{hex} für jedes der beiden Worte die gewünschte Adresse angeben.	

Tabelle 2-3 Systemparameter

Variable ID	Systemparameter	Wert / Anmerkung
1210 _{hex}	Rückmeldeverzögerung in ms (32-Bit-Wert) Nach Absetzen des Dienstes „Set_Value“ wird eine Rückmeldung vom Steuerungssystem um die mit diesem Parameter eingestellte Zeitspanne verzögert an die Anschaltbaugruppe zurückgegeben.	Im Anlauf der Steuerung kann über diesen Parameter auch eine Anlaufverzögerung realisiert werden. Wertangabe in ms.
1711 _{hex}	Systemparameter zum Auslesen des File-Systems des Parametrierungsspeichers.	
1712 _{hex}		
1713 _{hex}		
2200 _{hex}	Betriebsart	0000 0000 _{hex} : asynchron 0000 0001 _{hex} : bussynchron 0000 0002 _{hex} : programmsyn. 0600 0000 _{hex} : Asynchron mit Synchronisationsimpuls
2204 _{hex}	Das Verhalten der Eingangsdaten im Busfehlerfall kann mit diesem Parameter gesteuert werden.	0 = Eingangsdaten im MPM werden eingefroren (Default-Einstellung) 1 = Eingangsdaten im MPM werden auf Null gesetzt
2210 _{hex}	PD-Zykluszeit-Vorgabe in μ s (32-Bit-Wert) Die PD-Zykluszeit ist die tatsächliche Zeit zwischen den Starts von zwei aufeinanderfolgenden Prozessdatenübertragungszyklen (vgl. 2216 _{hex}).	Zulässiger Wertebereich: 0000 0000 _{hex} ... 0001 FB00 _{hex} (entspricht bis zu ca. 130 ms)

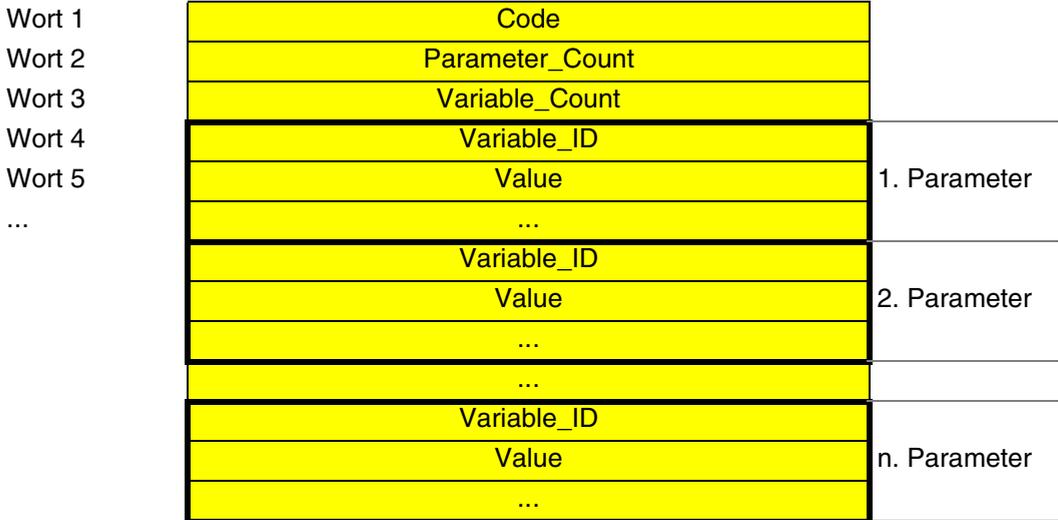
Tabelle 2-3 Systemparameter

Variable ID	Systemparameter	Wert / Anmerkung
2211 _{hex}	Busabschaltzeit in μs (32-Bit-Wert) Default-Einstellung: 200 ms Wenn innerhalb dieser Zeit kein fehlerfreier Datenzyklus möglich ist, beendet die Anschaltbaugruppe die Prozessdatenübertragung und führt einen Bus-Reset aus.	Zulässiger Wertebereich: 0000 0000 _{hex} : FFFF FFFF _{hex} Die maximale Busabschaltzeit / Buswarnzeit beträgt ca. 71 min.
2212 _{hex}	Buswarnzeit in μs (32-Bit-Wert) Default-Einstellung: 0 (nicht aktiviert) Wenn innerhalb dieser Zeit keine fehlerfreie Prozessdatenübertragung möglich ist, setzt die Anschaltbaugruppe das <i>Warningbit</i> im Diagnosestatus-Register.	
2215 _{hex}	Aktiviert oder deaktiviert den Fehler-Code 0BD2 _{hex} , der anzeigt, wenn innerhalb der mit der Variablen 2212 _{hex} vorgegebenen Buswarnzeit kein Datenzyklus übertragen werden konnte.	0000 0001 _{hex} : Fehler-Code aktiviert 0000 0000 _{hex} : Fehler-Code deaktiviert
2216 _{hex}	Tatsächliche PD-Zykluszeit in μs (32-Bit-Wert) Wenn die vorgegebene PD-Zykluszeit (2210 _{hex}) nicht eingehalten werden kann, wird an dieser Stelle die tatsächliche Zeit zwischen den Starts von zwei aufeinanderfolgenden Prozessdatenübertragungszyklen angezeigt.	
221B _{hex}	Das Verhalten der INTERBUS-Ausgänge bei einem Stopp der Steuerung (SYSFAIL) kann mit diesem Parameter gesteuert werden.	0 = INTERBUS-Datenzyklen werden mit auf Null gesetzten Ausgängen gefahren (Default-Einstellung). Ausgänge von Analogmodulen werden auch auf Null gesetzt. 1 = die Anschaltbaugruppe generiert einen Alarm-Stopp, der INTERBUS wird zurückgesetzt.

Tabelle 2-3 Systemparameter

Variable ID	Systemparameter	Wert / Anmerkung
2252 _{hex}	Zeigt an, welche SUPI-Typen in der aktuellen Konfiguration enthalten sind (32-Bit-Wert).	0000 0000 _{hex} : nur Teilnehmer mit SUPI älter als SUPI3 0000 0100 _{hex} : nur SUPI3-Teilnehmer 0000 0200 _{hex} : SUPI3-Teilnehmer und ältere Teilnehmer
2254 _{hex}	Mit diesem Wert wird die Anzahl der Busfehler vorgegeben, die innerhalb von 1.000.000. INTERBUS-Zyklen aufgetreten sein müssen, um im Diagnose-Statusregister das Quality-Bit zu setzen.	Default-Wert: n = 20. Beachten Sie: Das Quality-Bit wird nach 100.000 Zyklen neu berechnet.
A255 _{hex}	Einzelkanal-Diagnose freischalten	0000 0404 _{hex} : Alle Fehlermeldungen sperren 0000 047C _{hex} : Alle Fehlermeldungen freischalten 0000 046C _{hex} : Nur Fehlermeldung zur Spannungsversorgung U _{S2} sperren

Syntax: **Set_Value_Request** **0750_{hex}**



- Legende:
- Code: 0750_{hex} Kommando-Code des Requests
 - Parameter_Count: Anzahl der nachfolgenden Worte
Der Wert ist abhängig von der Anzahl und Länge der Systemparameter. Die Länge des Systemparameters (max. 16 Worte) wird in der dritten Hexadezimalstelle (Bit 8 ... 11) des Parameters *Variable_ID* angegeben.
 - Variable_Count: Anzahl der Systemparameter, denen neue Werte zugewiesen werden
 - Variable_ID: ID des Systemparameters, dem neue Werte zugewiesen werden (siehe Tabelle 2-3)
 - Value: Neuer Wert des Systemparameters

Syntax: **Set_Value_Confirmation** **8750_{hex}**

Positive Meldung

Wort 1	Code
Wort 2	Parameter_Count
Wort 3	Result

Negative Meldung

Wort 1	Code
Wort 2	Parameter_Count
Wort 3	Result
Wort 4	Add_Error_Info

Bit | 15 0 |

Legende:

Code:	8750 _{hex}	Meldungs-Code der Confirmation
Parameter_Count:		Anzahl der nachfolgenden Worte bei positiver Meldung: 0001 _{hex} 1 Parameterwort bei negativer Meldung: 0002 _{hex} 2 Parameterworte
Result:		Ergebnis der Dienstbearbeitung 0000 _{hex} zeigt eine positive Meldung an. Die Anschaltbaugruppe hat den Dienst erfolgreich bearbeitet. xxxx _{hex} zeigt eine negative Meldung an. Die Anschaltbaugruppe konnte den aufgerufenen Dienst nicht bearbeiten. Der Parameter <i>Result</i> enthält die Ursache, warum der Dienst nicht bearbeitet werden konnte.
Add_Error_Info:		Zusätzliche Informationen zur Fehlerursache

2.3.5 Dienst „Read_Value“

Aufgabe: Mit diesem Dienst können Sie die Systemparameter (Variablen) des INTERBUS auslesen.



Eine Aufstellung der definierten Systemparameter (Variablen) finden Sie in der Beschreibung des Dienstes „Set_Value“ (Tabelle 2-3 auf Seite 2-18).

Syntax: **Read_Value_Request** **0351_{hex}**

Wort 1	Code	
Wort 2	Parameter_Count	
Wort 3	Variable_Count	
Wort 4	Variable_ID	1. Parameter
Wort 5	Variable_ID	2. Parameter
...	...	
	Variable_ID	n. Parameter

Bit | 15 0 |

- Legende:
- Code: 0351_{hex} Kommando-Code des Requests
 - Parameter_Count: Anzahl der nachfolgenden Worte
 $xxxx_{hex} = 1 + Variable_Count$
 - Variable_Count: Anzahl der auszulesenden Systemparameter
 - Variable_ID: ID des auszulesenden Systemparameters

Syntax:

Read_Value_Confirmation

8351_{hex}

Positive Meldung

Wort 1	Code	
Wort 2	Parameter_Count	
Wort 3	Result	
Wort 4	Variable_Count	
Wort 5	Variable_ID	1. System-parameter
...	Value	
	...	
	Variable_ID	2. System-parameter
	Value	
	...	
	...	
	Variable_ID	n. System-parameter
	Value	
	...	

Negative Meldung

Wort 1	Code
Wort 2	Parameter_Count
Wort 3	Result
Wort 4	Add_Error_Info

Bit | 15 0 |

Legende:

Code: 8351_{hex} Meldungs-Code der Confirmation

Parameter_Count: Anzahl der nachfolgenden Worte
 bei positiver Meldung:
 xxxx_{hex} Der Wert ist abhängig von der Anzahl und Länge der Systemparameter. Die Länge des Systemparameters (max. 16 Worte) wird in der dritten Hexadezimalstelle (Bit 8 ... 11) des Parameters *Variable_ID* angegeben.

bei negativer Meldung:
 0002_{hex} 2 Parameterworte

Result: Ergebnis der Dienstbearbeitung

0000_{hex} zeigt eine positive Meldung an.
Die Anschaltbaugruppe hat den Dienst erfolgreich bearbeitet.

xxxx_{hex} zeigt eine negative Meldung an.
Die Anschaltbaugruppe konnte den aufgerufenen Dienst nicht bearbeiten.
Der Parameter *Result* enthält die Ursache, warum der Dienst nicht bearbeitet werden konnte.

Variable_Count: Anzahl der ausgelesenen Systemparameter
Variable_ID: ID des ausgelesenen Systemparameters
Value: Wert des Systemparameters
Add_Error_Info: Zusätzliche Informationen zur Fehlerursache

2.3.6 Dienst „Initiate_Load_Configuration“

Aufgabe:

Der Dienst „Initiate_Load_Configuration“ bereitet die Anschaltbaugruppe vor, eine Konfiguration mit den Diensten

- „Load_Configuration“ (0307_{hex}) oder
- „Complete_Load_Configuration“ (030A_{hex})

auf den INTERBUS-Master zu übertragen.

Zum Übertragen eines neuen Konfigurationsrahmens (Parameter *New_Config* = 0001_{hex}) müssen Sie die Parameter *Frame_Reference* und *Device_Count* (Gesamtanzahl der Teilnehmer) angeben.

Voraussetzung:

Die Parametrierungsphase muss zuvor mit dem Dienst „Control_Parameterization“ (030E_{hex}) eingeleitet worden sein.

Syntax:

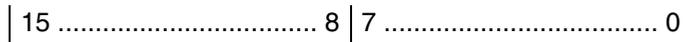
Initiate_Load_Configuration_Request

0306_{hex}

Wort 1
Wort 2
Wort 3
Wort 4
Wort 5
Wort 6
...

Code	
Parameter_Count	
New_Config	
Frame_Reference	
Device_Count	
Extension_Length	Extension
...	Extension

Bit



Legende:

- Code: 0306_{hex} Kommando-Code des Requests
- Parameter_Count: Anzahl der nachfolgenden Worte
 $xxxx_{hex} = 3 + (Extension_Length + 1) / 2$
- New_Config: 0001_{hex} erstellt den Konfigurationsrahmen neu. Ein existierender Konfigurationsrahmen wird überschrieben.
 0000_{hex} aktualisiert einen existierenden Konfigurationsrahmen.

Frame_Reference:	Logische Nummer, unter der der Konfigurationsrahmen abgelegt werden soll Wenn Sie einen bereits existierenden Konfigurationsrahmen überschreiben wollen (Parameter <i>New_Config</i> = 0000 _{hex}), werden nur einzelne Einträge der bestehenden Konfiguration überschrieben.
Device_Count:	Anzahl der INTERBUS-Teilnehmer im neu zu ladenden oder existierenden Konfigurationsrahmen
Extension_Length:	Länge (in Bytes) des Parameters <i>Extension</i> Wertebereich: 00 _{hex} ... 7F _{hex} (0 ... 127 Zeichen)
Extension:	Beliebige Informationen in ASCII-Code Mit diesem Parameter können Sie beispielsweise einen Namen für den Konfigurationsrahmen vergeben.

Syntax: **Initiate_Load_Configuration_Confirmation** **8306_{hex}**

Positive Meldung

Wort 1	Code
Wort 2	Parameter_Count
Wort 3	Result

Negative Meldung

Wort 1	Code
Wort 2	Parameter_Count
Wort 3	Result
Wort 4	Add_Error_Info

Bit | 15 0 |

Legende:

Code:	8306 _{hex}	Meldungs-Code der Confirmation
Parameter_Count:		Anzahl der nachfolgenden Worte bei positiver Meldung: 0001 _{hex} 1 Parameterwort bei negativer Meldung: 0002 _{hex} 2 Parameterworte
Result:		Ergebnis der Dienstbearbeitung 0000 _{hex} zeigt eine positive Meldung an. Die Anschaltbaugruppe hat den Dienst erfolgreich bearbeitet. xxxx _{hex} zeigt eine negative Meldung an. Die Anschaltbaugruppe konnte den aufgerufenen Dienst nicht bearbeiten. Der Parameter <i>Result</i> enthält die Ursache, warum der Dienst nicht bearbeitet werden konnte.
Add_Error_Info:		Zusätzliche Informationen zur Fehlerursache

2.3.7 Dienst „Load_Configuration“

Aufgabe:

Der Konfigurationsrahmen beschreibt jeden enthaltenen INTERBUS-Teilnehmer in einem eigenen nummerierten Eintrag. Die Reihenfolge und Nummerierung der Einträge entspricht dem physikalischen Busaufbau inklusive aller Alternativen.

Dieser Dienst überträgt Konfigurationsdaten in Form einer Liste zur Anschaltbaugruppe. Mit dem Parameter *Used_Attributes* bestimmen Sie, welche Attribute die Einträge enthalten sollen.



Der Dienst „Load_Configuration“ überprüft nicht die Konsistenz der Attribute untereinander, sondern nur ihre prinzipielle Zulässigkeit, z. B. auf Wertebereichsüberschreitung.

Voraussetzung:

Die Anschaltbaugruppe muss zuvor mit den Diensten
– „Control_Parameterization“ (030E_{hex}) und
– „Initiate_Load_Configuration“ (0306_{hex})
auf die Übertragung vorbereitet worden sein.

Syntax:

Load_Configuration_Request **0307_{hex}**

Wort 1	Code	
Wort 2	Parameter_Count	
Wort 3	Used_Attributes	
Wort 4	Start_Entry_No	
Wort 5	Entry_Count	
Wort 6	Configuration_Entry	1. Teilnehmer
...	...	
	Configuration_Entry	n. Teilnehmer

Bit | 15 0 |

Legende:

Code: 0307_{hex} Kommando-Code des Requests
Parameter_Count: Anzahl der nachfolgenden Parameterworte
xxxx_{hex} Der Wert ist abhängig von Parameter *Entry_Count* und Parameter *Used_Attributes*.

Used_Attributes:	<p>Auswahl der freizuschaltenden Attribute Der Parameter beschreibt ein 16 Bit breites Feld, wobei jedes Bit einem Attribut entspricht. Setzen Sie bei den Attributen, die Sie übergeben wollen, das entsprechende Bit auf 1 (vgl. Syntax „Configuration_Entry“ auf Seite 2-33). Belegung des Parameters <i>Used_Attributes</i>:</p> <table><tr><td>Bit 0</td><td>Teilnehmernummer</td></tr><tr><td>Bit 1</td><td>Teilnehmer-Code</td></tr><tr><td>Bit 2</td><td>reserviert (immer 0_{bin}!)</td></tr><tr><td>Bit 3</td><td>Teilnehmerebene</td></tr><tr><td>Bit 4</td><td>logische Gruppennummer</td></tr><tr><td>Bit 5</td><td>Fehlercharakteristik</td></tr><tr><td>Bit 6...13</td><td>reserviert (immer 0_{bin}!)</td></tr><tr><td>Bit 14</td><td>Erweiterte Teilnehmerinformationen</td></tr><tr><td>Bit 15</td><td>reserviert (immer 0_{bin}!)</td></tr></table> <p>Beispiel: Sollen die Einträge lediglich aus dem Teilnehmer-Code und der Teilnehmerebene bestehen, so geben Sie für den Parameter <i>Used_Attributes</i> den Wert 000A_{hex} an (Bit 1 und Bit 3 sind gesetzt).</p>	Bit 0	Teilnehmernummer	Bit 1	Teilnehmer-Code	Bit 2	reserviert (immer 0 _{bin} !)	Bit 3	Teilnehmerebene	Bit 4	logische Gruppennummer	Bit 5	Fehlercharakteristik	Bit 6...13	reserviert (immer 0 _{bin} !)	Bit 14	Erweiterte Teilnehmerinformationen	Bit 15	reserviert (immer 0 _{bin} !)
Bit 0	Teilnehmernummer																		
Bit 1	Teilnehmer-Code																		
Bit 2	reserviert (immer 0 _{bin} !)																		
Bit 3	Teilnehmerebene																		
Bit 4	logische Gruppennummer																		
Bit 5	Fehlercharakteristik																		
Bit 6...13	reserviert (immer 0 _{bin} !)																		
Bit 14	Erweiterte Teilnehmerinformationen																		
Bit 15	reserviert (immer 0 _{bin} !)																		
Start_Entry_No:	Nummer des ersten Teilnehmers, dessen Attribute Sie übertragen wollen																		
Entry_Count:	Anzahl der Teilnehmer, deren Attribute Sie übertragen wollen																		
Configuration_Entry:	Zu übertragende Attributwerte der einzelnen Teilnehmer in der Reihenfolge des physikalischen Busaufbaus (vgl. Syntax auf Seite 2-33).																		



Tragen Sie in dem Parameterblock „Configuration_Entry“ nach der folgenden Syntax nur die Attribute ein, die Sie durch den Parameter *Used_Attributes* freigeschaltet haben (nicht freigeschaltete Attribute entfallen).



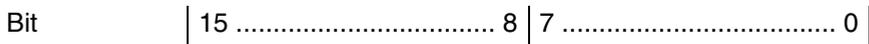
Bei gleichzeitigem Laden mehrerer Einträge mit jeweils mehreren Attributen werden zuerst alle Attribute eines Eintrags geladen, bevor der nächste Eintrag geladen wird.

Syntax

„Configuration_Entry“

Attribute

Wort x	Bus_Segment_No	Position	Teilnehmer- nummer
Wort x+1	Length_Code	ID_Code	Teilnehmer- Code
Wort x+2	Device_Level		Teilnehmer- ebene
Wort x+3	Group	Alternative	log. Gruppen- nummer
Wort x+4	reserviert		Fehler- charakteristik
Wort x+5...(259)			



Attribute:

- Bus_Segment_No: Nummer des Bussegments, in dem sich der Teilnehmer befindet
Wertebereich: 01_{hex} .. FF_{hex} (1 .. 255_{dez})
- Position: Physikalische Platznummer im Bussegment
Wertebereich: 00_{hex} .. 3F_{hex} (63_{dez}) für Lokalbus
Die Parameter *Bus_Segment_No* und *Position* bilden zusammen die Teilnehmernummer.
- Length_Code: Längen-Code
Der Längen-Code beschreibt den Adressraumbedarf des Teilnehmers im Host.
- ID_Code: ID-Code
Der ID-Code beschreibt die Art des Teilnehmers. Er ist als *Module Ident* den Modulen in dezimaler Form aufgedruckt (bei RT-Modulen hexadezimal).
Die Parameter *Length_Code* und *ID_Code* bilden zusammen die Teilnehmernummer.
- Device_Level: Teilnehmerebene
Wertebereich: 00_{hex} ... 0F_{hex}, (0 ... 15_{dez})
- Group: Gruppe, zu der der Teilnehmer gehört.

Alternative: Alternatives Bussegment, zu dem der Teilnehmer gehört.
Die Parameter *Group* und *Alternative* bilden zusammen die logische Gruppennummer.

Fehlercharakteristik:
Bit 0 = 1: Rückwirkungsfreies Schalten aktiv
= 0: Rückwirkungsfreies Schalten nicht aktiv reserviert

Bits 1 bis 15
Erweiterte Teilnehmerinformationen: Beschreibung siehe „Attribut: Erweiterte Teilnehmerinformationen“ auf Seite 2-49



Entsprechend dem Beispiel auf Seite 2-49 gilt für die Configuration_Entry nach Aufruf des Dienstes Load_Configuration als Startwert (Wort x+5].

Syntax:

Load_Configuration_Confirmation

8307_{hex}

Positive Meldung

Wort 1	Code
Wort 2	Parameter_Count
Wort 3	Result

Negative Meldung

Wort 1	Code
Wort 2	Parameter_Count
Wort 3	Result
Wort 4	Add_Error_Info

Bit | 15 0 |

Legende:

Code: 8307_{hex} Meldungs-Code der Confirmation

Parameter_Count: Anzahl der nachfolgenden Worte
bei positiver Meldung:
0001_{hex} immer 1 Parameterwort
bei negativer Meldung:
0002_{hex} immer 2 Parameterworte

Result: Ergebnis der Dienstbearbeitung
0000_{hex} zeigt eine positive Meldung an.
Die Anschaltbaugruppe hat den Dienst erfolgreich bearbeitet.
xxxx_{hex} zeigt eine negative Meldung an.

Die Anschaltbaugruppe konnte den aufgerufenen Dienst nicht bearbeiten. Der Parameter *Result* enthält die Ursache, warum der Dienst nicht bearbeitet werden konnte.

Add_Error_Info:

Zusätzliche Informationen zur Fehlerursache

2.3.8 Dienst „Terminate_Load_Configuration“

Aufgabe: Dieser Dienst beendet das segmentierte Laden von Konfigurationsdaten. Er überprüft außerdem die geladenen Konfigurationsdaten auf Zulässigkeit und Widersprüche. Falls kein Fehler gefunden wird, legt die Anschaltbaugruppe die Daten unter der im Dienst „Initiate_Load_Configuration“ (0306_{hex}) angegebenen *Frame_Reference* im Konfigurationsverzeichnis ab. Wenn die Überprüfung der Konfigurationsdaten einen Fehler ergibt, wird der Dienst mit einer entsprechenden Fehlermeldung negativ quittiert.

Bemerkung: Zusätzlich können Sie mit dem Parameter *Default_Parameter* angeben, ob Sie den Prozessdatenkanal (PD-Kanal) und/oder den PCP-Kanal anhand des geladenen Konfigurationsrahmens parametrieren lassen möchten. Die Firmware erstellt in diesem Fall automatisch die Prozessdatenreferenzliste („physikalische Adressierung“) und/oder eine Kommunikationsbeziehungsliste (CRL).



Der Dienst „Terminate_Load_Configuration“ aktiviert die neugeladene Konfiguration noch nicht. Dies geschieht erst durch den Dienst „Activate_Configuration“ (0711_{hex}). Dadurch können auch Konfigurationen geladen und projiziert werden, die nicht am Bus angeschlossen sind.

Syntax: **Terminate_Load_Configuration_Request** **0308_{hex}**

Wort 1	Code
Wort 2	Parameter_Count
Wort 3	Default_Parameter

Bit | 15 0 |

Legende:

Code: 0308_{hex} Kommando-Code des Requests

Parameter_Count: Anzahl der nachfolgenden Worte
0001_{hex} 1 Parameterwort

Default_Parameter: Der Parameter gibt an, ob für die geladene Konfiguration eine Default-Parametrierung des PCP- und/oder PD-Kanals vorgenommen werden soll:
0000_{hex} Keine automatische Parametrierung
0001_{hex} Automatische Parametrierung des Prozessdatenkanals durch die Erstellung der Prozessdatenreferenzliste

- 0002_{hex} Automatische Parametrierung des PCP-Kanals durch die Erstellung der Kommunikationsbeziehungsliste
- 0003_{hex} Automatische Parametrierung des Prozessdaten- und des PCP-Kanals

Syntax: **Terminate_Load_Configuration_Confirmation** **8308_{hex}**

Positive Meldung

Wort 1	Code
Wort 2	Parameter_Count
Wort 3	Result

Negative Meldung

Wort 1	Code
Wort 2	Parameter_Count
Wort 3	Result
Wort 4	Add_Error_Info

Bit | 15 0 |

- Legende:
- Code: 8308_{hex} Meldungs-Code der Confirmation
 - Parameter_Count: Anzahl der nachfolgenden Worte
 bei positiver Meldung:
 0001_{hex} 1 Parameterwort
 bei negativer Meldung:
 0002_{hex} 2 Parameterworte
 - Result: Ergebnis der Dienstbearbeitung
 0000_{hex} zeigt eine positive Meldung an.
 Die Anschaltbaugruppe hat den Dienst erfolgreich bearbeitet.
 xxxx_{hex} zeigt eine negative Meldung an.
 Die Anschaltbaugruppe konnte den aufgerufenen Dienst nicht bearbeiten.
 Der Parameter *Result* enthält die Ursache, warum der Dienst nicht bearbeitet werden konnte.
 - Add_Error_Info: Zusätzliche Informationen zur Fehlerursache

2.3.9 Dienst „Read_Configuration“

Aufgabe:

Dieser Dienst liest in Abhängigkeit der Parameter *Frame_Reference* und *Start_Entry_No* verschiedene Einträge des Konfigurationsverzeichnisses aus:

Frame_Reference	Start_Entry_No	Einträge, die der Dienst ausliest
0000 _{hex}	nicht relevant (0000 _{hex})	Header-Informationen des Konfigurationsverzeichnisses (CFG_OD_Header), d. h. Nummern aller projizierten Konfigurationsrahmen in aufsteigender Reihenfolge
> 0000 _{hex}	0000 _{hex}	Header-Informationen des mit dem Parameter <i>Frame_Reference</i> ausgewählten Konfigurationsrahmens (CFG_Header)
> 0000 _{hex}	> 0000 _{hex}	Einträge des mit dem Parameter <i>Frame_Reference</i> ausgewählten Konfigurationsrahmens (CFG_Entry). Dabei können Sie wahlweise den gesamten Konfigurationsrahmen oder nur einen Teil, z. B. eine einzelne INTERBUS-Teilnehmerbeschreibung, auslesen.

Syntax:

Read_Configuration_Request

0309_{hex}

- Wort 1
- Wort 2
- Wort 3
- Wort 4
- Wort 5
- Wort 6

Code
Parameter_Count
Frame_Reference
Used_Attributes
Start_Entry_No
Entry_Count

Bit | 15 0 |

Legende:

- Code: 0309_{hex} Kommando-Code des Requests
- Parameter_Count: Anzahl der nachfolgenden Worte
0004_{hex} 4 Parameterworte
- Frame_Reference: Nummer des Konfigurationsrahmens
0000_{hex} liest das Konfigurationsrahmenverzeichnis aus.
xxxx_{hex} liest den Konfigurationsrahmen mit dieser Nummer aus.

Nur relevant, wenn
Frame_Reference
> 0000_{hex}

Used_Attributes:

Attribute, die ausgelesen werden sollen.

Der Parameter beschreibt ein 16 Bit breites Feld, wobei jedes Bit einem Attribut entspricht. Setzen Sie bei den Attributen, die Sie auslesen wollen, das entsprechende Bit auf 1.

Belegung des Parameters *Used_Attributes*:

Bit 0	Teilnehmernummer
Bit 1	Teilnehmer-Code
Bit 2	reserviert (immer 0 _{bin} !)
Bit 3	Teilnehmerebene
Bit 4	logische Gruppennummer
Bit 5	Fehlercharakteristik
Bit 6...7	reserviert (immer 0 _{bin} !)
Bit 8	global_bus_error
Bit 9	separate_bus_error
Bit 10	Teilnehmerstatus
Bit 11	Erweiterte Modulstatusinformationen (u. a. Einzelkanal-Diagnose)
Bit 12	Optische Diagnose-Informationen
Bit 13	reserviert (immer 0 _{bin} !)
Bit 14	Erweiterte Teilnehmerinformationen
Bit 15	reserviert (immer 0 _{bin} !)

Start_Entry_No:

Position des ersten Eintrags

0000_{hex} liest nur die Header-Informationen des Konfigurationsrahmens.

xxxx_{hex}: liest die Einträge aus dem Konfigurationsverzeichnis ab dieser Nummer.

Entry_Count:

Anzahl der auszulesenden Einträge

Bit 15	Dieses Bit bestimmt die Einträge im Parameter <i>Start_Entry_No</i> : = 0: physikalische Teilnehmernummer = 1: logische Teilnehmernummer
--------	--

Die positive Meldung überträgt die gewünschten Einträge des Konfigurationsverzeichnisses. Sie hat in Abhängigkeit der Parameter *Frame_Reference* und *Start_Entry_No* in dem Request eine der drei folgenden Zusammensetzungen.

Syntax

Read_Configuration_Confirmation

8309_{hex}

1. Zusammensetzung

Positive Meldung bei Request mit:

- Frame_Reference = 0000_{hex},
- Start_Entry_No nicht relevant (= 0000_{hex})

Wort 1	Code	
Wort 2	Parameter_Count	
Wort 3	Result	
Wort 4	More_Follows	
Wort 5	Frame_Reference	= 0000 _{hex}
Wort 6	Current_Configuration	
Wort 7	Configuration_Count	
Wort 8	Frame_Reference 1	
...	...	
	Frame_Reference n	

2. Zusammensetzung

Positive Meldung bei Request mit:

- Frame_Reference > 0000_{hex},
- Start_Entry_No = 0000_{hex}

Wort 1	Code	
Wort 2	Parameter_Count	
Wort 3	Result	
Wort 4	More_Follows	
Wort 5	Frame_Reference	> 0000 _{hex}
Wort 6	Used_Attributes	nicht relevant!
Wort 7	Start_Entry_No	= 0000 _{hex}
Wort 8	Frame_Device_Count	
Wort 9	Active_Device_Count	
Wort 10	Frame_IO_Bit_Count	
Wort 11	Active_IO_Bit_Count	
Wort 12	Frame_PCP_Device_Count	
Wort 13	Active_PCP_Device_Count	

Wort 14	Frame_PCP_Word_Count	
Wort 15	Active_PCP_Word_Count	
Wort 16	Extension	Extension
Bit	15 0	

3. Zusammensetzung

Positive Meldung bei Request mit:

- Frame_Reference > 0000_{hex},
- Start_Entry_No > 0000_{hex}

Wort 1	Code	
Wort 2	Parameter_Count	
Wort 3	Result	
Wort 4	More_Follows	
Wort 5	Frame_Reference	
Wort 6	Used_Attributes	
Wort 7	Start_Entry_No	
Wort 8	Entry_Count	
Wort 9	Configuration_Entry	1. Teilnehmer
...	...	
	Configuration_Entry	n. Teilnehmer

Negative Meldung

Wort 1	Code
Wort 2	Parameter_Count
Wort 3	Result
Wort 4	Add_Error_Info
Bit	15 0

Legende:

- Code: 8309_{hex} Meldungs-Code der Confirmation
- Parameter_Count: Anzahl der nachfolgenden Worte bei positiver Meldung und wenn *Frame_Reference* = 0000_{hex}:
 $xxxx_{hex} = 5 + Configuration_Count$

	bei positiver Meldung und wenn <i>Frame_Reference</i> > 0000 _{hex} und <i>Start_Entry_No</i> = 0000 _{hex} : 000D _{hex} 13 Parameterworte
	bei positiver Meldung und wenn <i>Frame_Reference</i> > 0000 _{hex} und <i>Start_Entry_No</i> > 0000 _{hex} : xxxx _{hex} Der Wert ist abhängig von der Anzahl der Teilnehmer im Konfigurationsrahmen und der Anzahl der freigeschalteten Attribute.
	bei negativer Meldung: 0002 _{hex} 2 Parameterworte
Result:	Ergebnis der Dienstbearbeitung 0000 _{hex} zeigt eine positive Meldung an. Der Request wurde erfolgreich ausgeführt. Die Daten stehen in den nachfolgenden Parametern bereit. xxxx _{hex} zeigt eine negative Meldung an. D. h. die Anschaltbaugruppe konnte den aufgerufenen Dienst nicht bearbeiten. Der Parameter <i>Result</i> enthält die Ursache, warum der Dienst nicht bearbeitet werden konnte (siehe auch <i>Add_Error_Info</i>).
Add_Error_Info:	Zusätzliche Informationen zur Fehlerursache
More_Follows:	0000 _{hex} zeigt an, dass alle angeforderten Einträge in der Confirmation enthalten sind. 0001 _{hex} zeigt an, dass noch nicht alle angeforderten Einträge in der Confirmation enthalten sind. Die Datenmenge ist größer als die für Dienste zur Verfügung stehende Mailbox (MXI). Rufen Sie den Dienst erneut auf, um den Rest auszulesen.
Frame_Reference:	Nummer des ausgelesenen Konfigurationsrahmens Der Parameter enthält den Wert, der mit dem Request übergeben wurde.
Current_Configuration:	Nummer des aktuell eingeschalteten Konfigurationsrahmens

Configuration_Count:	Anzahl der projektierten Konfigurationsrahmen
Frame_Reference x:	Nummern aller projektierten Konfigurationsrahmen in aufsteigender Reihenfolge
Frame_Device_Count:	Anzahl der im ausgewählten Konfigurationsrahmen projektierten INTERBUS-Teilnehmer
Active_Device_Count:	Anzahl der im ausgewählten Konfigurationsrahmen aktiven INTERBUS-Teilnehmer
Frame_IO_Bit_Count:	Anzahl der im ausgewählten Konfigurationsrahmen projektierten IO-Bits
Active_IO_Bit_Count:	Anzahl der im ausgewählten Konfigurationsrahmen aktiven IO-Bits
Frame_PCP_Device_Count:	Anzahl der im ausgewählten Konfigurationsrahmen projektierten PCP-Teilnehmer
Active_PCP_Device_Count:	Anzahl der im ausgewählten Konfigurationsrahmen aktiven PCP-Teilnehmer
Frame_PCP_Word_Count:	Anzahl der im ausgewählten Konfigurationsrahmen projektierten PCP-Worte
Active_PCP_Word_Count:	Anzahl der im ausgewählten Konfigurationsrahmen aktiven PCP-Worte
Extension:	Mit dem Dienst „Initiate_Load_Configuration“ übertragene Informationen, z. B. ein Name für den Konfigurationsrahmen.
Used_Attributes:	Ausgelesene Attribute Der Parameter enthält den Wert, der mit dem Request übergeben wurde.
Start_Entry_No:	Position des ersten Eintrags oder 0000 _{hex} , wenn nur die Header-Informationen ausgelesen wurden
Entry_Count:	Anzahl der Einträge, die mit dieser Confirmation übertragen werden. Der Parameter <i>More_Follows</i> zeigt an, ob weitere Einträge vorhanden sind.

Configuration_Entry: Ausgewählte Einträge in der Reihenfolge des physikalischen Busaufbaus
 Die in jedem Eintrag enthaltenen Attribute werden in dem Request mit dem Parameter *Used_Attributes* freigeschaltet (vgl. Syntax „Configuration_Entry“ auf Seite 2-44).



Ein Konfigurationseintrag eines Teilnehmers muss nicht alle Attribute enthalten. Ist ein Attribut über den Parameter *Used_Attributes* in dem Request nicht freigeschaltet, dann ist der Konfigurationseintrag um die zugehörigen Datenworte verkürzt.

Der folgende Block zeigt den Aufbau eines Konfigurationseintrages, wenn **alle** Attribute freigeschaltet sind.

Syntax

„Configuration_Entry“

Attribut:

Wort x	Bus_Segment_No	Position	Teilnehmernummer
Wort x+1	Length_Code	ID_Code	Teilnehmer-Code
Wort x+2	Device_Level		Teilnehmerebene
Wort x+3	Group	Alternative	log. Gruppennummer
Wort x+4	reserviert		Bit 0 Fehlercharakteristik
Wort x+5	SUPI_Type	Add_Info	„global_bus_error“
Wort x+6	Transmission_Error		
Wort x+7	Device_Error		
Wort x+8	Transmission_Fail_DO		
Wort x+9	Transmission_Fail_DI		
Wort x+10	MAU_FAIL_DO		
Wort x+11	MAU_FAIL_DI		
Wort x+12	MAU_Warning_DO	MAU_Warning_DI	
Wort x+13	Out1_FAIL		
Wort x+14	Out2_FAIL		
Wort x+15	Reconfigure_Request		„separate_bus_error“
Wort x+16	Peripheral_State		
Wort x+17	Microprocessor_Watchdog		
Wort x+18	Device_State		

Wort x+19	Channel 16 ... 31		Erweiterte Modul-Status-Informationen
Wort x+20	Channel 0 ... 15		
Wort x+21	Initiator	Power	Optische Diagnose-Informationen
Wort x+22	Length	Optical Info Forward Line	
Wort x+23	Optical Info Return Line	- -	Erweiterte Teilnehmer-Information
Wort x+24...(278)			
Bit	15 8 7 0		

Legende:

Attribut: Teilnehmernummer

Bus_Segment_No: Nummer des Bussegments, in dem sich der INTERBUS-Teilnehmer befindet.
 Wertebereich: 00_{hex} ... FF_{hex} (0_{dez} ... 255_{dez})

Position: Physikalische Platznummer im Bussegment
 Wertebereich:
 00_{hex} ... 3F_{hex} (0_{dez} ... 63_{dez}) für Installationslokalbus)

Attribut: Teilnehmer-Code

Length_Code: Längen-Code
 Der Längen-Code beschreibt den Adressraumbedarf des INTERBUS-Teilnehmers im Host.

ID_Code: ID-Code
 Der ID-Code beschreibt die Funktion des INTERBUS-Teilnehmers. Er ist als *Module Ident* den Modulen in dezimaler Form aufgedruckt (bei RT-Modulen hexadezimal).

Attribut: Teilnehmerebene

Device_Level: Teilnehmerebene
Die Teilnehmerebene beschreibt die Verzweigungstiefe, in der sich der Teilnehmer befindet.
Wertebereich: 00_{hex} bis 0F_{hex} (0 bis 15_{dez})

Attribut: Logische Gruppennummer

Group: Gruppe, zu der der Teilnehmer gehört
Alternative: Alternatives Bussegment, zu dem der Teilnehmer gehört

Attribut: Fehlercharakteristik

Bit 0 = 1: Rückwirkungsfreies Schalten aktiv
= 0: Rückwirkungsfreies Schalten nicht aktiv
Bits 1 bis 15 reserviert

Attribut: „global_bus_error“

SUPI_Type: SUPI-Typ
xx00 ältere SUPIs und SUPI2
xx01 SUPI1
xx03 SUPI3 und neuer

Add_Info: Chip-Kennung des SUPIs
00_{hex} SUPI älter als SUPI3
A1_{hex} IB8052
A2_{hex} LPC2
A3_{hex} LPC1
A4_{hex} SUPI3-DPC
A5_{hex} SUPI3
D0_{hex} SUPI3-OPC
FF_{hex} reserviert

Transmission_Error: Übertragungsfehler
Device_Error: Peripheriefehler

Attribut: „separate_bus_error“

Transmission_Fail_DO:	Zähler für Übertragungsfehler im Datenhinweg
Transmission_Fail_DI:	Zähler für Übertragungsfehler im Datenrückweg
MAU_FAIL_DO:	Zähler für Leitungsunterbrechung im Datenhinweg
MAU_FAIL_DI:	Zähler für Leitungsunterbrechung im Datenrückweg
MAU_Warning_DO:	Zähler für Verschlechterung der Übertragungsqualität im Hinweg (bei LWL-Übertragung); bei Chip LPC2: überlastete interne Stromquelle
MAU_Warning_DI:	Zähler für Verschlechterung der Übertragungsqualität im Rückweg (bei LWL-Übertragung); bei Chip LPC2: zu hohe Temperatur des IBS-Protokoll-Chips
Out1_FAIL:	Zähler für Fehler des RBST-Signals (Brücke im Stecker) an der weiterführenden Schnittstelle oder Fehler an nicht belegter OUT1-Schnittstelle.
Out2_FAIL:	Zähler für Fehler des LBST-Signals (Brücke im Stecker) an der weiterführenden Schnittstelle oder Fehler an nicht belegter OUT2-Schnittstelle.
Reconfigure_Request:	Zähler für Rekonfigurations-Anforderungen; bei Chip LPC2: zu geringe Spannung für die Initiatoren
Peripheral_Fault:	Zähler für Peripheriefehler
Microprocessor_Watchdog:	Zähler für Resets des angeschlossenen Mikroprozessors; bei Chip LPC2: das Überschreiten des erlaubten Ausgangstroms vom Leistungstreiber

Attribut: Teilnehmerstatus

Device_State	Zustand des Teilnehmers
Bit 0, 1	nicht relevant
Bit 2	1 _{bin} Teilnehmer aktiv 0 _{bin} Teilnehmer nicht aktiv
Bit 3, 4	nicht relevant
Bit 5	1 _{bin} Teilnehmer überbrückt 0 _{bin} Teilnehmer nicht überbrückt
Bit 6...11	reserviert
Bit 12	1 _{bin} Teilnehmer meldet PF (Peripheral Fault) 0 _{bin} Teilnehmer meldet kein PF
Bit 13...15	reserviert

Attribut: Erweiterte Modulstatusinformationen

Channel 16...31	Bit 0...15 Ein gesetztes Bit gibt eine Peripherie- störung an einem der Modulkanaäle 16...31 an.
Channel 0...15	Bit 0...15 Ein gesetztes Bit gibt eine Peripherie- störung an einem der Modulkanaäle 0...15 an.
Initiator	Bit 8...15 Ein gesetztes Bit gibt den Ausfall der Initiatorversorgung an einer der Modulgruppen 1 bis 8 an.
Power	Bit 0...3 Ein gesetztes Bit gibt den Ausfall einer der Versorgungsspannungen U1 bis U4 an.

Attribut: Optische Diagnose-Informationen

Length	Bit 8...13	Länge der ankommenden Strecke
	Bit 14	Optisches Medium
	Bit 15	Freigabestatus
Optical Info Forward Line	Bit 0...3	Leistungsstufe
	Bit 4	reserviert
	Bit 5	Verzerrung
	Bit 6, 7	reserviert
Optical Info Return Line	Bit 8...11	Leistungsstufe
	Bit 12	reserviert
	Bit 13	Verzerrung
	Bit 14, 15	reserviert

Attribut: Erweiterte Teilnehmerinformationen

Die erweiterte Teilnehmerinformation wird als Byte-String mit einer maximalen Länge von 255 Byte übertragen.



Bedingt durch die direkt aufeinander folgende Übertragung der Daten dieses Byte-Strings können auf die erweiterte Teilnehmerinformation folgende Konfigurationseinträge (Configuration_Entries) auf ungeraden Adressen beginnen, wenn die erweiterte Teilnehmerinformation eine ungerade Länge besitzt.

Aufbau der erweiterten Teilnehmerinformation:

x_0	x_{11}	x_{12}	x_{13}	x_{21}	x_{22}	x_{23}	...
-------	----------	----------	----------	----------	----------	----------	-----

Legende:

		Beispiel	Beschreibung
$x_0 =$	Länge der erweiterten Teilnehmerinformation in Byte	$0F_{hex}$	
$x_{11} =$	Kennung der Extension 1	01_{hex}	Service-Info (IBS CMD G4), ASCII-codiert, maximal 12 Zeichen 01_{hex} = Stationsname
$x_{12} =$	Länge der Extension 1 in Byte	07_{hex}	
$x_{13} =$	Extension 1	$4D\ 6F\ 74\ 6F\ 72\ 30\ 31_{hex}$	„Motor01“
$x_{21} =$	Kennung der Extension 2	02_{hex}	
$x_{22} =$	Länge der Extension 2 in Byte	04_{hex}	
$x_{23} =$	Extension 2	$19\ 04\ 20\ 02_{hex}$	Information frei definierbar
... =	weitere Extensions bis zu einer Gesamtlänge des Byte-Strings von maximal 255 Byte.		

Im folgenden Beispiel wird die Service-Info „Teilnehmername“ als erweiterte TeilnehmerinformationExtension übertragen. Die Anfangsadresse beginnt bei „Wort x + 24“, basierend auf der Configuration_Entry auf Seite 2-44. Dort beginnt die erweiterte Teilnehmerinformation mit der gleichen Adresse.

Beispiel:

Übertragung der Service-Info Teilnehmername (IBS CMD)

Wort x+24	Länge der gesamten Extension in Byte [0E _{hex}]	Kennung der Extension <i>Service-Info</i>	Extension:
Wort x+25	Länge der Extension <i>Service-Info</i> [0C _{hex}]	Byte 2	<i>Service-Info:</i> Teilnehmername
Wort x+26	Byte 3	Byte 4	
Wort x+27	Byte 5	Byte 6	
Wort x+28	Byte 7	Byte 8	
Wort x+29	Byte 9	Byte 10	
Wort x+30	Byte 11	Byte 12	
Wort x+31	Byte 13	- -	
Bit	15 8	7 0	

2.3.10 Dienst „Compare_Configuration“

Aufgabe:

Dieser Dienst überträgt Konfigurationsdaten in Form einer Liste von Einträgen zur Anschaltbaugruppe. Mit dem Parameter *Used_Attributes* bestimmen Sie, welche Attribute die Einträge enthalten sollen.

Die Anschaltbaugruppe vergleicht diese Liste mit dem Konfigurationsrahmen, der durch den Parameter *Frame_Reference* angegeben wird. Falls die Liste und der Konfigurationsrahmen nicht übereinstimmen, erfolgt die negative Confirmation und die Anschaltbaugruppe überträgt Informationen zum Fehler in Form von Fehler-Codes.

Voraussetzung:

Der mit dem Parameter *Frame_Reference* anzugebende Konfigurationsrahmen muss auf der Anschaltbaugruppe vorhanden sein!



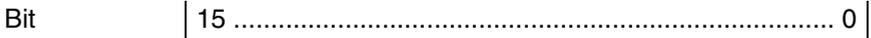
Der Konfigurationsrahmen beschreibt **jeden** enthaltenen INTERBUS-Teilnehmer in einem eigenen nummerierten Eintrag. Die Reihenfolge und Nummerierung der Einträge entspricht dem physikalischen Busaufbau inklusive **aller** Alternativen.

Syntax:

Compare_Configuration_Request

0317_{hex}

Wort 1	Code	
Wort 2	Parameter_Count	
Wort 3	Frame_Reference	
Wort 4	Used_Attributes	
Wort 5	Start_Entry_No	
Wort 6	Entry_Count	
Wort 7	Configuration_Entry	1. Teilnehmer
...	...	
	Configuration_Entry	n. Teilnehmer



Legende:

Code: 0317_{hex} Kommando-Code des Requests

Parameter_Count:	Anzahl der nachfolgenden Worte $xxxx_{hex}$ Der Wert ist abhängig von der Anzahl der Einträge und die Anzahl und Art der Attribute, die Sie vergleichen wollen.
Frame_Reference:	Nummer des Konfigurationsrahmens, mit dem die Anschaltbaugruppe die übertragene Liste vergleichen soll.
Used_Attributes:	Attribute des auszulesenden Konfigurationsrahmens Der Parameter beschreibt ein 16 Bit breites Feld, wobei jedes Bit einem Attribut entspricht. Setzen Sie bei den Attributen, die Sie auslesen wollen, die entsprechenden Bits auf 1. Belegung des Parameters <i>Used_Attributes</i> : Bit 0 Teilnehmernummer Bit 1 Teilnehmer-Code Bit 2 reserviert (immer 0 _{bin} !) Bit 3 Teilnehmerebene Bit 4 logische Gruppennummer Bit 5 Fehlercharakteristik Bit 6...15 reserviert (immer 0 _{bin} !)
Start_Entry_No:	Eintragsnummer des ersten Teilnehmers im Konfigurationsrahmen, dessen Attribute mit der übertragenen Liste verglichen werden sollen.
Entry_Count:	Anzahl der Einträge, die Sie auf die Anschaltbaugruppe übertragen möchten.
Configuration_Entry:	Attributwerte der einzelnen Teilnehmer Übergeben Sie nur die Attributwerte, die Sie mit dem Parameter <i>Used_Attributes</i> ausgewählt haben, und beachten Sie dabei die Reihenfolge des physikalischen Busaufbaus. Die Parameter von <i>Configuration_Entry</i> werden beim Dienst „Read_Configuration“ auf Seite 2-38 beschrieben.

Syntax: **Compare_Configuration_Confirmation** **8317_{hex}**

Positive Meldung

Wort 1	Code
Wort 2	Parameter_Count
Wort 3	Result

Negative Meldung

Wort 1	Code
Wort 2	Parameter_Count
Wort 3	Result
Wort 4	Add_Error_Info

Bit | 15 0 |

Legende:

Code: 8317_{hex} Meldungs-Code der Confirmation

Parameter_Count: Anzahl der nachfolgenden Worte
bei positiver Meldung:
0001_{hex} 1 Parameterwort
bei negativer Meldung:
0002_{hex} 2 Parameterworte

Result: Ergebnis der Dienstbearbeitung
0000_{hex} zeigt eine positive Meldung an.
Die Anschaltbaugruppe hat den Dienst erfolgreich bearbeitet.
xxxx_{hex} zeigt eine negative Meldung an.
Die Anschaltbaugruppe konnte den aufgerufenen Dienst nicht bearbeiten.
Der Parameter *Result* enthält die Ursache, warum der Dienst nicht bearbeitet werden konnte.

Add_Error_Info: Zusätzliche Informationen zur Fehlerursache

2.3.11 Dienst „Complete_Load_Configuration“

Aufgabe: Dieser Dienst überträgt Konfigurationsdaten in Form von mit *Used_Attributes* ausgewählten Attributen zum INTERBUS-Master. Er ist speziell auf die Anforderungen der SPS-Programmierung abgestimmt.

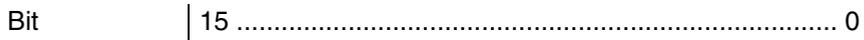
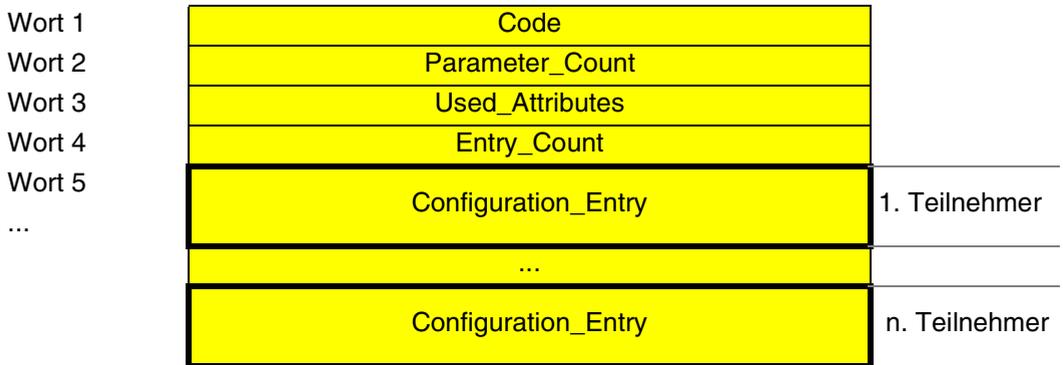
Voraussetzung: Die Anschaltbaugruppe muss zuvor mit den Diensten

- „Control_Parameterization“ (030E_{hex}, vgl. Seite 2-11) und
- „Initiate_Load_Configuration“ (0306_{hex} vgl. Seite 2-28)

auf die Übertragung vorbereitet worden sein.

Bemerkung: Der Dienst kann als Meta-Dienst für den Dienst „Load_Configuration“ (0307_{hex}, vgl. Seite 2-31) verstanden werden. Dieser Dienst überträgt alle Einträge des Konfigurationsrahmens (*Start_Entry_No* = 1).

Syntax: **Complete_Load_Configuration_Request** **030A_{hex}**



Legende:

Code:	030A _{hex}	Kommando-Code des Requests
Parameter_Count:		Anzahl der nachfolgenden Worte
	xxxx _{hex}	Der Wert ist abhängig von der Anzahl der Einträge und von der Anzahl und Art der Attribute, die Sie übertragen wollen.

Used_Attributes: Zu übergebende Attribute
Der Parameter beschreibt ein 16 Bit breites Feld, wobei jedes Bit einem Attribut entspricht. Setzen Sie bei den Attributen, die Sie übergeben wollen, die entsprechenden Bits auf 1.
Belegung des Parameters *Used_Attributes*:
Bit 0 Teilnehmernummer
Bit 1 Teilnehmer-Code
Bit 2 reserviert (immer 0_{bin}!)
Bit 3 Teilnehmerebene
Bit 4 logische Gruppennummer
Bit 5 Fehlercharakteristik
Bit 6...13 reserviert (immer 0_{bin}!)
Bit 14 Erweiterte Teilnehmerinformationen
Bit 15 reserviert (immer 0_{bin}!)

Entry_Count: Anzahl der zu übertragenden Einträge

Configuration_Entry: Attributwerte der einzelnen Teilnehmer
Übergeben Sie nur die Attributwerte, die Sie mit dem Parameter *Used_Attributes* ausgewählt haben, und beachten Sie dabei die Reihenfolge des physikalischen Busaufbaus. Für die Beschreibung der Parameter von *Configuration_Entry* siehe Dienst „Read_Configuration“ (0309_{hex}) auf Seite 2-38.

Syntax: **Complete_Load_Configuration_Confirmation** **830A_{hex}**

Positive Meldung

Wort 1	Code
Wort 2	Parameter_Count
Wort 3	Result

Negative Meldung

Wort 1	Code
Wort 2	Parameter_Count
Wort 3	Result
Wort 4	Add_Error_Info

Bit | 15 0 |

Legende:

Code:	830A _{hex}	Meldungs-Code der Confirmation
Parameter_Count:		Anzahl der nachfolgenden Worte bei positiver Meldung: 0001 _{hex} 1 Parameterwort bei negativer Meldung: 0002 _{hex} 2 Parameterworte
Result:		Ergebnis der Dienstbearbeitung 0000 _{hex} zeigt eine positive Meldung an. Die Anschaltbaugruppe hat den Dienst erfolgreich bearbeitet. xxxx _{hex} zeigt eine negative Meldung an. Die Anschaltbaugruppe konnte den aufgerufenen Dienst nicht bearbeiten. Der Parameter <i>Result</i> enthält die Ursache, warum der Dienst nicht bearbeitet werden konnte.
Add_Error_Info:		Zusätzliche Informationen zur Fehlerursache

2.3.12 Dienst „Complete_Read_Configuration“

Aufgabe: Dieser Dienst liest Einträge des Konfigurationsverzeichnisses in Form einer oder mehrerer mit dem Parameter *Used_Attributes* selektierter Spalten aus. Er ist speziell auf die Anforderungen der SPS-Programmierung abgestimmt.

Bemerkung: Der Dienst kann als Meta-Dienst für den Dienst „Read_Configuration“ (0309_{hex}) verstanden werden. Der Parameter *Start_Entry_No* muss nicht angegeben werden, da dieser Dienst alle Einträge des Konfigurationsrahmens ausliest (*Start_Entry_No* = 1).

Syntax: **Complete_Read_Configuration_Request** **030B_{hex}**

Wort 1	Code
Wort 2	Parameter_Count
Wort 3	Used_Attributes

Bit | 15 0 |

- Legende:**
- Code: 030B_{hex} Kommando-Code des Requests
 - Parameter_Count: Anzahl der nachfolgenden Worte
0001_{hex} immer 1 Parameterwort
 - Used_Attributes: Der Parameter beschreibt ein 16 Bit breites Feld, wobei jedes Bit einem Attribut entspricht. Setzen Sie bei den Attributen, die Sie lesen wollen, die entsprechenden Bits auf 1.
Belegung des Parameters *Used_Attributes*:
 - Bit 0 Teilnehmernummer
 - Bit 1 Teilnehmer-Code
 - Bit 2 reserviert (immer 0_{bin}!)
 - Bit 3 Teilnehmerebene
 - Bit 4 logische Gruppennummer
 - Bit 5 Fehlercharakteristik
 - Bit 6...7 reserviert (immer 0_{bin}!)
 - Bit 8 global_bus_error
 - Bit 9 separate_bus_error
 - Bit 10 Teilnehmerstatus
 - Bit 11 Erweiterte Modul-Statusinformationen
 - Bit 12 Optische Diagnose-Informationen
 - Bit 13 reserviert (immer 0_{bin}!)
 - Bit 14 Erweiterte Teilnehmerinformationen
 - Bit 15 reserviert (immer 0_{bin}!)

Syntax:

Complete_Read_Configuration_Confirmation

830B_{hex}

Positive Meldung

Wort 1	Code	
Wort 2	Parameter_Count	
Wort 3	Result	
Wort 4	More_Follows	
Wort 5	Frame_Reference	
Wort 6	Used_Attributes	
Wort 7	Start_Entry_No	0001 _{hex}
Wort 8	Entry_Count	
Wort 9	Configuration_Entry	1. Teilnehmer
...	...	
	Configuration_Entry	n. Teilnehmer

Negative Meldung

Wort 1	Code
Wort 2	Parameter_Count
Wort 3	Result
Wort 4	Add_Error_Info

Bit | 15 0 |

Legende:

Code: 830B_{hex} Meldungs-Code der Confirmation

Parameter_Count: Anzahl der nachfolgenden Worte
 bei positiver Meldung:
 xxxx_{hex} Der Wert ist abhängig von der Anzahl der Einträge und von der Anzahl und Art der Attribute, die gelesen wurden.
 bei negativer Meldung:
 0002_{hex} 2 Parameterworte

Result:	Ergebnis der Dienstbearbeitung 0000 _{hex} zeigt eine positive Meldung an. Die Anschaltbaugruppe hat den Dienst erfolgreich bearbeitet. xxxx _{hex} zeigt eine negative Meldung an. Die Anschaltbaugruppe konnte den aufgerufenen Dienst nicht bearbeiten. Der Parameter <i>Result</i> enthält die Ursache, warum der Dienst nicht bearbeitet werden konnte.
Add_Error_Info:	Zusätzliche Informationen zur Fehlerursache
More_Follows:	0000 _{hex} zeigt an, dass alle angeforderten Einträge in der Confirmation enthalten sind. 0001 _{hex} zeigt an, dass noch nicht alle angeforderten Einträge in der Confirmation enthalten sind. Die Datenmenge ist größer als die für Dienste zur Verfügung stehende Mailbox (MXI). Rufen Sie den Dienst „Read_Configuration“ (0309 _{hex}) auf, um den Rest auszu-lesen.
Frame_Reference:	Nummer des aktiven Konfigurationsrahmens
Used_Attributes:	Ausgelesene Attribute Der Parameter enthält den Wert, der mit dem Request übergeben wurde.
Start_Entry_No:	Nummer des ersten Eintrags 0001 _{hex} Bei diesem Dienst werden vom ersten Eintrag an alle Einträge ausgelesen.
Entry_Count:	Anzahl der Einträge, die von dieser Confirmation übertragen werden.
Configuration_Entry:	Einträge in Reihenfolge des physikalischen Busaufbaus Die in jedem Eintrag enthaltenen Attribute werden in dem Request mit dem Parameter <i>Used_Attributes</i> freigeschaltet. Für die Beschreibung der Parameter von <i>Configuration_Entry</i> siehe Dienst „Read_Configuration“ (0309 _{hex}) auf Seite 2-38.

2.3.13 Dienst „Delete_Configuration“

Aufgabe: Dieser Dienst löscht einen inaktiven Konfigurationsrahmen aus dem Konfigurationsverzeichnis.

Syntax: **Delete_Configuration_Request** **030C_{hex}**

Wort 1	Code
Wort 2	Parameter_Count
Wort 3	Frame_Reference

Bit | 15 0 |

Legende:

Code:	030C _{hex} Kommando-Code des Requests
Parameter_Count:	Anzahl der nachfolgenden Worte
	0001 _{hex} 1 Parameterwort
Frame_Reference	Nummer des zu löschenden Konfigurationsrahmens

Syntax: **Delete_Configuration_Confirmation** **830C_{hex}**

Positive Meldung

Wort 1	Code
Wort 2	Parameter_Count
Wort 3	Result

Negative Meldung

Wort 1	Code
Wort 2	Parameter_Count
Wort 3	Result
Wort 4	Add_Error_Info

Bit | 15 0 |

Legende:

Code: 830C_{hex} Meldungs-Code der Confirmation

Parameter_Count: Anzahl der nachfolgenden Worte
bei positiver Meldung:
0001_{hex} 1 Parameterwort
bei negativer Meldung:
0002_{hex} 2 Parameterworte

Result: Ergebnis der Dienstbearbeitung
0000_{hex} zeigt eine positive Meldung an.
Die Anschaltbaugruppe hat den Dienst erfolgreich bearbeitet.
xxxx_{hex} zeigt eine negative Meldung an.
Die Anschaltbaugruppe konnte den aufgerufenen Dienst nicht bearbeiten.
Der Parameter *Result* enthält die Ursache, warum der Dienst nicht bearbeitet werden konnte.

Add_Error_Info: Zusätzliche Informationen zur Fehlerursache

2.3.14 Dienst „Create_Configuration“

Aufgabe:

Dieser Dienst veranlasst die Anschaltbaugruppe, aus der aktuell angeschlossenen Konfiguration automatisch einen Konfigurationsrahmen zu erzeugen und diesen aufzuschalten, um den Bus in Betrieb zu nehmen. Nach Ausführung des Dienstes befindet sich die Anschaltbaugruppe im Zustand *Active* (Anzeige: *ACTV*).

Der neue Konfigurationsrahmen und die aktive Konfiguration werden unter der im Parameter *Frame_Reference* angegebenen Nummer im Konfigurationsverzeichnis abgelegt. Falls unter dieser Nummer bereits ein Konfigurationsrahmen existiert, wird dieser überschrieben. Außerdem erzeugt die Anschaltbaugruppe anhand der aktuell angeschlossenen Buskonfiguration Default-Prozessdatenbeschreibungslisten, eine Default-Prozessdatenreferenzliste und eine Default-Kommunikationsbeziehungsliste (CRL). In den Gerätebeschreibungen werden die Attribute wie folgt initialisiert:

- Device_Number:* Entsprechend der aktuellen Konfiguration
- Length_Code:* Entsprechend der aktuellen Konfiguration
- ID_Code:* Entsprechend der aktuellen Konfiguration
- Device_Level:* Entsprechend der aktuellen Konfiguration
- Group_Number:* Für alle INTERBUS-Teilnehmer FFFF_{hex} (d. h. keine Gruppennummer)
- Device_State:* Alle INTERBUS-Teilnehmer sind aktiv

Syntax:

Create_Configuration_Request

0710_{hex}

Wort 1	Code
Wort 2	Parameter_Count
Wort 3	Frame_Reference

Bit | 15 0 |

Legende:

- Code: 0710_{hex} Kommando-Code des Requests
- Parameter_Count: Anzahl der nachfolgenden Worte
0001_{hex} 1 Parameterwort
- Frame_Reference: Nummer für den automatisch zu erzeugenden Konfigurationsrahmen

Syntax: **Create_Configuration_Confirmation** **8710_{hex}**

Positive Meldung

Wort 1	Code
Wort 2	Parameter_Count
Wort 3	Result

Negative Meldung

Wort 1	Code
Wort 2	Parameter_Count
Wort 3	Result
Wort 4	Add_Error_Info

Bit | 15 0 |

Legende:

Code: 8710_{hex} Meldungs-Code der Confirmation

Parameter_Count: Anzahl der nachfolgenden Worte
bei positiver Meldung:
0001_{hex} 1 Parameterwort
bei negativer Meldung:
0002_{hex} 2 Parameterworte

Result: Ergebnis der Dienstbearbeitung
0000_{hex} zeigt eine positive Meldung an.
Die Anschaltbaugruppe hat den Dienst erfolgreich bearbeitet.
xxxx_{hex} zeigt eine negative Meldung an.
Die Anschaltbaugruppe konnte den aufgerufenen Dienst nicht bearbeiten.
Der Parameter *Result* enthält die Ursache, warum der Dienst nicht bearbeitet werden konnte.

Add_Error_Info: Zusätzliche Informationen zur Fehlerursache

2.3.15 Dienst „Activate_Configuration“

Aufgabe: Dieser Dienst veranlasst die Anschaltbaugruppe, die Daten eines Konfigurationsrahmens auf

- Übereinstimmung mit der aktuell angeschlossenen Konfiguration und
- Adressüberlagerungen

zu überprüfen.

Wird kein Fehler gefunden, schaltet die Anschaltbaugruppe diesen Konfigurationsrahmen aktiv (Anzeige: *ACTV*) und fährt in regelmäßigen Abständen ID-Zyklen. Die Nummer des Konfigurationsrahmens teilen Sie der Anschaltbaugruppe in dem Parameter *Frame_Reference* mit.

Voraussetzung: Wenn Sie einen Konfigurationsrahmen aktivieren möchten, darf kein anderer Konfigurationsrahmen aktiv sein. Einen Konfigurationsrahmen deaktivieren Sie mit dem Dienst „Deactivate_Configuration“ (0712_{hex}).

Syntax: **Activate_Configuration_Request** **0711_{hex}**

Wort 1	Code
Wort 2	Parameter_Count
Wort 3	Frame_Reference

Bit | 15 0 |

Legende:

Code:	0711 _{hex}	Kommando-Code des Requests
Parameter_Count:	0001 _{hex}	1 Parameterwort
Frame_Reference:		Nummer des zu aktivierenden Konfigurationsrahmens

Syntax: **Activate_Configuration_Confirmation** **8711_{hex}**

Positive Meldung

Wort 1	Code
Wort 2	Parameter_Count
Wort 3	Result

Negative Meldung

Wort 1	Code
Wort 2	Parameter_Count
Wort 3	Result
Wort 4	Add_Error_Info

Bit | 15 0 |

Legende:

Code: 8711_{hex} Meldungs-Code der Confirmation

Parameter_Count: Anzahl der nachfolgenden Worte
bei positiver Meldung:
0001_{hex} 1 Parameterwort
bei negativer Meldung:
0002_{hex} 2 Parameterworte

Result: Ergebnis der Dienstbearbeitung
0000_{hex} zeigt eine positive Meldung an.
Die Anschaltbaugruppe hat den Dienst erfolgreich bearbeitet.
xxxx_{hex} zeigt eine negative Meldung an.
Die Anschaltbaugruppe konnte den aufgerufenen Dienst nicht bearbeiten.
Der Parameter *Result* enthält die Ursache, warum der Dienst nicht bearbeitet werden konnte.

Add_Error_Info: Zusätzliche Informationen zur Fehlerursache

2.3.16 Dienst „Deactivate_Configuration“

Aufgabe:

Dieser Dienst schaltet den angegebenen Konfigurationsrahmen inaktiv. Es werden keine ID-Zyklen mehr gefahren. Nach Ausführung des Dienstes befindet sich die Anschaltbaugruppe im Zustand *Ready* (Anzeige: *RDY*).



Falls Sie zuvor den Dienst „Stop_Data_Transfer“ (0702_{hex}) ausgeführt haben, hat dieser eingefrorene Ausgangszustände (gesetzte Ausgänge) zurückgelassen. Diese Ausgänge werden beim Deaktivieren des Konfigurationsrahmens zurückgesetzt.

Voraussetzung:

Der angegebene Konfigurationsrahmen muss existieren und vor dem Aufruf des Dienstes aktiv geschaltet sein.

Syntax:

Deactivate_Configuration_Request

0712_{hex}

Wort 1
Wort 2
Wort 3

Code
Parameter_Count
Frame_Reference

Bit

| 15 0 |

Legende:

- Code: 0712_{hex} Kommando-Code des Requests
- Parameter_Count: Anzahl der nachfolgenden Worte
0001_{hex} 1 Parameterwort
- Frame_Reference: Nummer des zu deaktivierenden Konfigurationsrahmens

Syntax: **Deactivate_Configuration_Confirmation** **8712_{hex}**

Positive Meldung

Wort 1	Code
Wort 2	Parameter_Count
Wort 3	Result

Negative Meldung

Wort 1	Code
Wort 2	Parameter_Count
Wort 3	Result
Wort 4	Add_Error_Info

Bit | 15 0 |

Legende:

Code:	8712 _{hex}	Meldungs-Code der Confirmation
Parameter_Count:		Anzahl der nachfolgenden Worte bei positiver Meldung: 0001 _{hex} 1 Parameterwort bei negativer Meldung: 0002 _{hex} 2 Parameterworte
Result:		Ergebnis der Dienstbearbeitung 0000 _{hex} zeigt eine positive Meldung an. Die Anschaltbaugruppe hat den Dienst erfolgreich bearbeitet. xxxx _{hex} zeigt eine negative Meldung an. Die Anschaltbaugruppe konnte den aufgerufenen Dienst nicht bearbeiten. Der Parameter <i>Result</i> enthält die Ursache, warum der Dienst nicht bearbeitet werden konnte.
Add_Error_Info:		Zusätzliche Informationen zur Fehlerursache

2.4 Dienste zum Definieren von Prozessdatenbeschreibungen

2.4.1 Dienst „Initiate_Put_Process_Data_Description_List“

Aufgabe: Dieser Dienst leitet das Definieren von Prozessdatenbeschreibungen ein. Er bereitet die Anschaltbaugruppe auf den Dienst „Put_Process_Data_Description_List“ (0321_{hex}, vgl. Seite 2-70) vor.

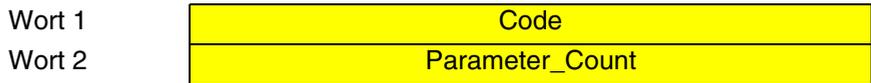
Voraussetzung: Die Anschaltbaugruppe muss zuvor mit den Diensten

- „Control_Parameterization“ (030E_{hex}),
- „Initiate_Load_Configuration“ (0306_{hex}) und
- „Complete_Load_Configuration“ (030A_{hex})

auf die Parametrierung vorbereitet worden sein.

Der Konfigurationsrahmen muss geladen werden, weil er die Teilnehmernummern enthält, denen die Prozessdatenbeschreibungen zugeordnet werden sollen.

Syntax: **Initiate_Put_Process_Data_Description_List_Request** **0320_{hex}**



Legende:

Code:	0320 _{hex}	Kommando-Code des Requests
Parameter_Count:		Anzahl der nachfolgenden Worte
	0000 _{hex}	kein Parameterwort

Syntax: **Initiate_Put_Process_Data_Description_List_Confirmation** **8320_{hex}**

Positive Meldung

Wort 1	Code
Wort 2	Parameter_Count
Wort 3	Result

Negative Meldung

Wort 1	Code
Wort 2	Parameter_Count
Wort 3	Result
Wort 4	Add_Error_Info

Bit | 15 0 |

Legende:

Code: 8320_{hex} Meldungs-Code der Confirmation

Parameter_Count: Anzahl der nachfolgenden Worte
 bei positiver Meldung:
 0001_{hex} 1 Parameterwort
 bei negativer Meldung:
 0002_{hex} 2 Parameterworte

Result: Ergebnis der Dienstbearbeitung
 0000_{hex} zeigt eine positive Meldung an.
 Die Anschaltbaugruppe hat den Dienst erfolgreich bearbeitet.
 xxxx_{hex} zeigt eine negative Meldung an.
 Die Anschaltbaugruppe konnte den aufgerufenen Dienst nicht bearbeiten.
 Der Parameter *Result* enthält die Ursache, warum der Dienst nicht bearbeitet werden konnte.

Add_Error_Info: Zusätzliche Informationen zur Fehlerursache

2.4.2 Dienst „Put_Process_Data_Description_List“

Für jeden INTERBUS-Teilnehmer definiert die Firmware automatisch eine Prozessdatenbeschreibung (PDD), die seine gesamten Prozessdaten umfasst.

So definiert die Firmware z. B. für einen INTERBUS-Teilnehmer mit 16 Bit Eingangs- und 16 Bit Ausgangsdaten automatisch eine

- Prozessdatenbeschreibung über 16 Eingangs-Bits und
- Prozessdatenbeschreibung über 16 Ausgangs-Bits.

Aufgabe:

Mit dem Dienst „Put_Process_Data_Description_List“ können Sie für einen INTERBUS-Teilnehmer mehrere Prozessdatenbeschreibungen definieren und somit die Prozessdaten des Teilnehmers in kleine Elemente aufteilen. Der Dienst schreibt diese Definitionen in die Prozessdatenbeschreibungsliste.

Voraussetzung:

Die Anschaltbaugruppe muss zuvor mit den Diensten

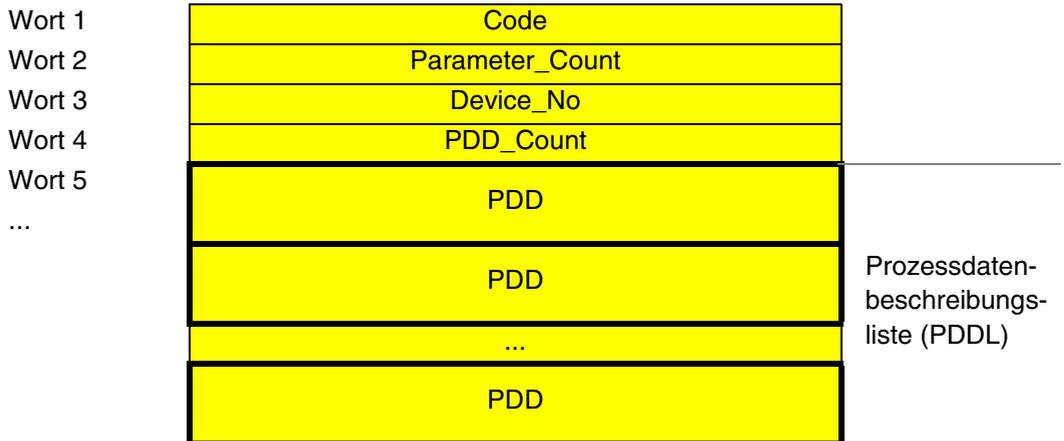
- „Control_Parameterization“ (030E_{hex}) und
- „Initiate_Put_Process_Data_Description_List“ (0320_{hex})

vorbereitet worden sein.

Syntax:

Put_Process_Data_Description_List_Request

0321_{hex}



Legende:

Code: 0321_{hex} Kommando-Code des Requests

- Parameter_Count: Anzahl der nachfolgenden Worte
 xxx_{hex} Die Anzahl der Parameterworte ist abhängig von der Anzahl und Größe der Prozessdatenbeschreibungen.
- Device_No: Teilnehmernummer des Teilnehmers, in dessen Prozessdatenbeschreibungsliste (PDDL) Prozessdatenbeschreibungen (PDD) geschrieben werden sollen.
- PDD_Count: Anzahl der Prozessdatenbeschreibungen
- PDD: Prozessdatenbeschreibung (PDD)
 (vgl. nachfolgende Syntax)

Syntax

Prozessdatenbeschreibung (PDD):

PDD_Index	
Data_Direction	Data_Type
Byte_Position	
Bit_Position	Length
Extension_Length	Extension
...	Extension



Legende:

- PDD_Index: Index der Prozessdatenbeschreibung
 Vergeben Sie für jede Prozessdatenbeschreibung einen anderen *PDD_Index*, um diese eindeutig zu kennzeichnen. Jeder *PDD_Index* darf pro Teilnehmer nur einmal vergeben werden!
 Zulässiger Wertebereich für *PDD_Index*:
 0000_{hex} ... 7FFF_{hex}
 bereits vergeben:
 6010_{hex} Default Eingangs-Prozessdatenbeschreibungen
 6011_{hex} Default Ausgangs-Prozessdatenbeschreibungen
- Data_Direction: Datenrichtung der Prozessdatenbeschreibung
 0C_{hex} für Eingangs-Prozessdaten
 0D_{hex} für Ausgangs-Prozessdaten

Data_Type:	Art der Prozessdatenbeschreibung 0A _{hex} Byte-String-Prozessdatum 0F _{hex} Bit-String-Prozessdatum
Byte_Position:	Byte-Offset (Byte im Prozessdatenbereich des Teilnehmers) 0000 _{hex} 1. Byte (Bit 0 ... 7) d. h. niederwertigstes Byte 0001 _{hex} 2. Byte (Bit 8 ... 15), d. h. nächst höherwertiges Byte 0002 _{hex} 3. Byte (Bit 16 ... 23) ... n. Byte je nach Größe des Teilnehmers
Bit_Position:	Bit-Offset. Beginn des Bit-Strings im mit Parameter <i>Byte_Position</i> ausgewählten Byte. Zulässiger Wertebereich: 00 _{hex} ... 07 _{hex} für Bit 0 bis Bit 7
Length:	Länge des Prozessdatums Wertebereich: 1 ... 64 Byte für Byte-String-Prozessdatum 1 ... 8 Bit für Bit-String-Prozessdatum Ein Bit-String darf keine Byte-Grenze überschreiten!
Extension_Length:	Länge des Parameters <i>Extension</i> Wertebereich: 00 _{hex} ... 7F _{hex} , d. h. 0 ... 127 Byte
Extension:	Informationen in ASCII-Code zu der Prozessdatenbeschreibung, z. B. einen Namen

Syntax: **Put_Process_Data_Description_List_Confirmation** **8321_{hex}**

Positive Meldung

Wort 1	Code
Wort 2	Parameter_Count
Wort 3	Result

Negative Meldung

Wort 1	Code
Wort 2	Parameter_Count
Wort 3	Result
Wort 4	Add_Error_Info

Bit | 15 0 |

Legende:

Code:	8321 _{hex}	Meldungs-Code der Confirmation
Parameter_Count:		Anzahl der nachfolgenden Worte bei positiver Meldung: 0001 _{hex} 1 Parameterwort bei negativer Meldung: 0002 _{hex} 2 Parameterworte
Result:		Ergebnis der Dienstbearbeitung 0000 _{hex} zeigt eine positive Meldung an. Die Anschaltbaugruppe hat den Dienst erfolgreich bearbeitet. xxxx _{hex} zeigt eine negative Meldung an. Die Anschaltbaugruppe konnte den aufgerufenen Dienst nicht bearbeiten. Der Parameter <i>Result</i> enthält die Ursache, warum der Dienst nicht bearbeitet werden konnte.
Add_Error_Info:		Zusätzliche Informationen zur Fehlerursache

Syntax: **Terminate_Put_Process_Data_Description_List_Confirmation** **8322_{hex}**

Positive Meldung

Wort 1	Code
Wort 2	Parameter_Count
Wort 3	Result

Negative Meldung

Wort 1	Code
Wort 2	Parameter_Count
Wort 3	Result
Wort 4	Add_Error_Info

Bit | 15 0 |

- Legende:
- Code: 8322_{hex} Meldungs-Code der Confirmation
 - Parameter_Count: Anzahl der nachfolgenden Worte
 bei positiver Meldung:
 0001_{hex} 1 Parameterwort
 bei negativer Meldung:
 0002_{hex} 2 Parameterworte
 - Result: Ergebnis der Dienstbearbeitung
 0000_{hex} zeigt eine positive Meldung an.
 Die Anschaltbaugruppe hat den Dienst erfolgreich bearbeitet.
 xxxx_{hex} zeigt eine negative Meldung an.
 Die Anschaltbaugruppe konnte den aufgerufenen Dienst nicht bearbeiten.
 Der Parameter *Result* enthält die Ursache, warum der Dienst nicht bearbeitet werden konnte.
 - Add_Error_Info: Zusätzliche Informationen zur Fehlerursache

2.4.4 Dienst „Get_Process_Data_Description_List“

Aufgabe:

Dieser Dienst liest eine oder mehrere Prozessdatenbeschreibungen (PDD) aus der Prozessdatenbeschreibungsliste eines INTERBUS-Teilnehmers aus.

Über den Parameter *Start_PDD_Index* stellen Sie ein, ob Sie

- nur den Kopf der Prozessdatenbeschreibungsliste oder
- eine oder mehrere Prozessdatenbeschreibungen

auslesen wollen.

Syntax:

Get_Process_Data_Description_List_Request

0323_{hex}

Wort 1	Code
Wort 2	Parameter_Count
Wort 3	Device_No
Wort 4	Start_PDD_Index
Wort 5	PDD_Count

Bit | 15 0 |

Legende:

- Code: 0323_{hex} Kommando-Code des Requests
- Parameter_Count: Anzahl der nachfolgenden Worte
0003_{hex} immer 3 Parameterworte
- Device_No: Teilnehmernummer des INTERBUS-Teilnehmers, aus dessen Prozessdatenbeschreibungsliste (PDDL) Sie Prozessdatenbeschreibungen (PDD) auslesen möchten.
- Start_PDD_Index: 0000_{hex} nur Kopf der Prozessdatenbeschreibungsliste auslesen
xxxx_{hex} Index der ersten auszulesenden Prozessdatenbeschreibung (PDD).
- PDD_Count: Anzahl der Prozessdatenbeschreibungen (nur relevant, wenn *Start_PDD_Index* > 0000_{hex})

Syntax: **Get_Process_Data_Description_List_Confirmation** **8323_{hex}**

Positive Meldung und *Start_PDD_Index* > 0000_{hex}

Wort 1	Code	
Wort 2	Parameter_Count	
Wort 3	Result	
Wort 4	More_Follows	
Wort 5	Device_No	
Wort 6	Start_PDD_Index	
Wort 7	PDD_Count	
...	PDD	Prozessdaten- beschreibungs- liste (PDDL)
	...	
	PDD	

Positive Meldung und *Start_PDD_Index* = 0000_{hex}

Wort 1	Code
Wort 2	Parameter_Count
Wort 3	Result
Wort 4	More_Follows
Wort 5	Device_No
Wort 6	Start_PDD_Index
Wort 7	PDD_Count

Negative Meldung

Wort 1	Code
Wort 2	Parameter_Count
Wort 3	Result
Wort 4	Add_Error_Info

Bit | 15 0 |

Legende: Code: 8323_{hex} Meldungs-Code der Confirmation
Parameter_Count: Anzahl der nachfolgenden Worte
bei positiver Meldung und
Start_PDD_Index > 0000_{hex}:

	xxxx _{hex}	Die Anzahl der Parameterworte ist abhängig von der Anzahl und Größe der Prozessdatenbeschreibungen.
		bei positiver Meldung und Parameter <i>Start_PDD_Index</i> = 0000 _{hex} :
	0005 _{hex}	5 Parameterworte
		bei negativer Meldung:
	0002 _{hex}	2 Parameterworte
Result:		Ergebnis der Dienstbearbeitung
	0000 _{hex}	zeigt eine positive Meldung an. Die Anschaltbaugruppe konnte den Dienst erfolgreich bearbeiten.
	xxxx _{hex}	zeigt eine negative Meldung an. Die Anschaltbaugruppe konnte den aufgerufenen Dienst nicht bearbeiten. Der Parameter <i>Result</i> enthält die Ursache, warum der Dienst nicht bearbeitet werden konnte.
Add_Error_Info:		Zusätzliche Informationen zur Fehlerursache
More_Follows:	0000 _{hex}	zeigt an, dass alle angeforderten Prozessdatenbeschreibungen in der Confirmation enthalten sind.
	0001 _{hex}	zeigt an, dass noch nicht alle angeforderten Einträge in der Confirmation enthalten sind. Die Datenmenge ist größer als die für Dienste zur Verfügung stehende Mailbox (MXI). Rufen Sie den Dienst erneut auf, um den Rest auszulesen. Geben Sie dabei für den Parameter <i>Start_PDD_Index</i> den Index der letzten übertragenen Prozessdatenbeschreibung, erhöht um 0001 _{hex} , an.
Device_No:		INTERBUS-Teilnehmernummer des Teilnehmers, aus dessen Prozessdatenbeschreibungsliste Prozessdatenbeschreibungen ausgelesen wurden.
Start_PDD_Index:		Index der ersten ausgelesenen Prozessdatenbeschreibung

PDD_Count:	Anzahl der Prozessdatenbeschreibungen, die ausgelesen wurden oder in der Prozessdatenbeschreibungsliste vorhanden sind.
PDD:	Prozessdatenbeschreibungen (Syntax vgl. Dienst „Put_Process_Data_Description_List“ ab Seite 2-70)

2.5 Dienste zum Zuordnen von Prozessdaten

2.5.1 Dienst „Initiate_Load_Process_Data_Reference_List“

Aufgabe:

Dieser Dienst leitet das Festlegen von Prozessdatenreferenzen (PDR) ein. Er bereitet die Anschaltbaugruppe auf den Dienst „Load_Process_Data_Reference_List“ (0325_{hex}) vor.



Geben Sie mit dem Parameter *New_PDRL* an, ob der Dienst eine bereits existierende Prozessdatenreferenzenliste (PDRL) überschreiben soll. Dies ist sinnvoll, wenn Sie sich zuvor mit dem Dienst „Create_Configuration“ (0710_{hex}) eine Prozessdatenreferenzenliste erstellen lassen haben („physikalische Adressierung“), nun aber eine eigene Projektierung durchführen möchten.

Voraussetzung:

Die Anschaltbaugruppe muss zuvor auf die Parametrierung vorbereitet worden sein („Control_Parameterization“). Der Konfigurationsrahmen muss geladen sein („Initiate_Load_Configuration“, dann „Complete_Load_Configuration“). Alle Prozessdatenbeschreibungen, die in der PDRL benutzt werden, müssen geladen sein („Initiate_Put_Process_Data_Description_List“, dann „Put_Process_Data_Description_List“).

Syntax:

Initiate_Load_Process_Data_Reference_List_Request **0324_{hex}**

Wort 1	Code
Wort 2	Parameter_Count
Wort 3	New_PDRL

Bit | 15 0 |

Legende:

- Code: 0324_{hex} Kommando-Code des Requests
- Parameter_Count: Anzahl der nachfolgenden Worte
 - 0001_{hex} 1 Parameterwort
- New_PDRL: Funktionsweise
 - 0000_{hex} überschreibt eine existierende PDRL nicht.
 - 0001_{hex} überschreibt eine existierende PDRL.

Syntax: **Initiate_Load_Process_Data_Reference_List_Confirmation** **8324_{hex}**

Positive Meldung

Wort 1	Code
Wort 2	Parameter_Count
Wort 3	Result

Negative Meldung

Wort 1	Code
Wort 2	Parameter_Count
Wort 3	Result
Wort 4	Add_Error_Info

Bit | 15 0 |

Legende:

Code: 8324_{hex} Meldungs-Code der Confirmation

Parameter_Count: Anzahl der nachfolgenden Worte
 bei positiver Meldung:
 0001_{hex} 1 Parameterwort
 bei negativer Meldung:
 0002_{hex} 2 Parameterworte

Result: Ergebnis der Dienstbearbeitung
 0000_{hex} zeigt eine positive Meldung an.
 Die Anschaltbaugruppe hat den Dienst erfolgreich bearbeitet.
 xxxx_{hex} zeigt eine negative Meldung an.
 Die Anschaltbaugruppe konnte den aufgerufenen Dienst nicht bearbeiten.
 Der Parameter *Result* enthält die Ursache, warum der Dienst nicht bearbeitet werden konnte.

Add_Error_Info: Zusätzliche Informationen zur Fehlerursache

2.5.2 Dienst „Load_Process_Data_Reference_List“

Aufgabe:

Mit diesem Dienst legen Sie Prozessdatenreferenzen (PDR) in einer Prozessdatenreferenzliste (PDRL) fest. Sie können Prozessdatenreferenzen für Eingangsdaten, Ausgangsdaten und Durchgangsdaten festlegen.



Die Reihenfolge, in der Sie die Prozessdatenreferenzen (PDR) für Eingangs-, Ausgangs- und Durchgangsdaten übergeben, ist nicht von Bedeutung. Die Anschaltbaugruppe speichert die Daten in der Reihenfolge der aufsteigenden *PDR-Indices* (vgl. Syntax „Prozessdatenreferenz“).

Voraussetzung:

Die Anschaltbaugruppe muss zuvor mit den Diensten

- „Control_Parameterization (030E_{hex}) und
- „Initiate_Load_Process_Data_Reference_List“ (0324_{hex})

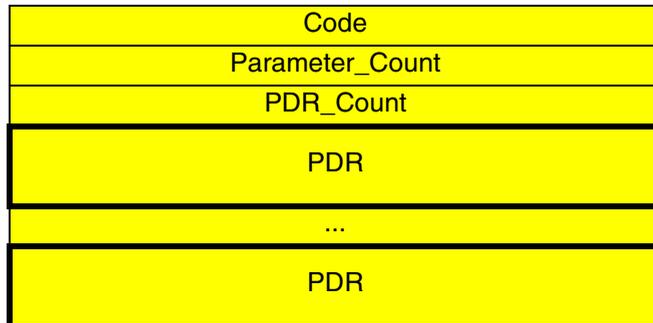
vorbereitet worden sein.

Syntax:

Load_Process_Data_Reference_List_Request

0325_{hex}

Wort 1
Wort 2
Wort 3
Wort 4
...



Prozessdatenreferenzliste (PDRL)

Bit



Legende:

- Code: 0325_{hex} Kommando-Code des Requests
- Parameter_Count: Anzahl der nachfolgenden Worte
xxxx_{hex} Der Wert ist abhängig von der Anzahl und Größe der Prozessdatenreferenzen.
- PDR_Count: Anzahl der Prozessdatenreferenzen, die übertragen werden
- PDR: Prozessdatenreferenzen

Syntax

Prozessdatenreferenz für **Eingangsdaten**:

Wort x+1	PDR_Index		Source
Wort x+2	Device_No		
Wort x+3	PDD_Index		
Wort x+4	Destination_Address		Destination
Wort x+5	Bit_Position	Data_Cons	
Wort x+6	Extension_Length	Extension	
Wort x+7	...	Extension	

Prozessdatenreferenz für **Ausgangsdaten**:

Wort x+1	PDR_Index		Source
Wort x+2	Source_Address		
Wort x+3	Bit_Position	Data_Cons	
Wort x+4	Device_No		Destination
Wort x+5	PDD_Index		
Wort x+6	Extension_Length	Extension	
Wort x+7	...	Extension	

Prozessdatenreferenz für **Durchgangsdaten**:

Wort x+1	PDR_Index		Source
Wort x+2	Source_Device_No		
Wort x+3	Source_PDD_Index		
Wort x+4	Dest_Device_No		Destination
Wort x+5	Dest_PDD_Index		
Wort x+6	Extension_Length	Extension	
Wort x+7	...	Extension	

Bit | 15 8 | 7 0 |

Legende:

PDR_Index: Index der Prozessdatenreferenz (PDR)
 Zulässige Wertebereiche:
 0001_{hex} ... 3FFF_{hex} für Eingangsdaten
 4000_{hex} ... 7FFF_{hex} für Ausgangsdaten
 8000_{hex} ... BFFF_{hex} für Durchgangsdaten



Vergeben Sie für jede Prozessdatenreferenz einen anderen *PDR_Index*, um diese eindeutig zu kennzeichnen. Jeder *PDR_Index* darf nur einmal vergeben werden!

Device_No:	INTERBUS-Teilnehmer (Segment . Position)
PDD_Index:	Index des Prozessdatums
Source_Address:	Quelladresse im MPM (Ausgangsdaten), empfohlener Wertebereich: 0000 _{hex} ... 0400 _{hex}
Destination_Address:	Zieladresse im MPM (Eingangsdaten), empfohlener Wertebereich: 1000 _{hex} ... 1400 _{hex}
Bit_Position:	Beginn eines Bit-Strings auf der Ziel-/Quell- adresse im MPM Zulässiger Wertebereich: 00 _{hex} ... 07 _{hex} für Bit 0 ... Bit 7
Data_Cons:	Datenkonsistenz für den Prozessdatenzugriff Zulässige Werte: 00 _{hex} 16 Bit (Standard) 01 _{hex} 32 Bit (z.B. Encoder, Bedienterminals) 02 _{hex} 8 Bit für Module oder Prozessdaten, die kleiner/gleich 8 Bit sind 03 _{hex} 64 Bit (z.B. Encoder, Bedienterminals)



Die Datenkonsistenz garantiert, dass die angegebene Datenbreite aus einem INTERBUS-Zyklus stammt.

Der Standardwert für E/A-Module ist 16 Bit. Für INTERBUS-Teilnehmer, welche die zusammenhängende Übertragung größerer Datenbreiten verlangen, müssen Sie die Datenkonsistenz erhöhen. Dies sind z. B. Encoder, Bedienterminals oder Analogmodule mit mehr als 16 Bit Auflösung. Die Datenkonsistenz muss innerhalb eines Wortes identisch sein.

Sie können ein Wort

- aufteilen in zwei, jeweils in sich konsistente Byte (je 8 Bit),
- als in sich konsistent definieren (16 Bit) oder
- einem größeren Konsistenzbereich (32 oder 64 Bit) zuweisen.

Es ist nicht zulässig, einem Byte eines Wortes eine Datenkonsistenz von 8 Bit und dem anderen Byte einen anderen, größeren Konsistenzbereich zuzuweisen.

Extension_Length:	Länge des Parameters <i>Extension</i> Wertebereich: 00 _{hex} ... 7F _{hex} , d. h. 0 ... 127 Zeichen
Extension:	Informationen in ASCII-Code zu der Prozess- datenreferenz, z. B. einen Kommentar
Source_Device_No:	INTERBUS-Teilnehmer, dessen Eingangsdaten Sie zuordnen möchten.
Source_PDD_Index:	Index des Prozesseingangsdatums

Dest_Device_No: INTERBUS-Teilnehmer, dem Sie die Daten zuordnen möchten
 Dest_PDD_Index: Index des Prozessausgangsdatums

Syntax: **Load_Process_Data_Reference_List_Confirmation** **8325_{hex}**

Positive Meldung

Wort 1	Code
Wort 2	Parameter_Count
Wort 3	Result

Negative Meldung

Wort 1	Code
Wort 2	Parameter_Count
Wort 3	Result
Wort 4	Add_Error_Info

Bit | 15 0 |

Legende:

Code: 8325_{hex} Meldungs-Code der Confirmation

Parameter_Count: Anzahl der nachfolgenden Worte
 bei positiver Meldung:
 0001_{hex} 1 Parameterwort
 bei negativer Meldung:
 0002_{hex} 2 Parameterworte

Result: Ergebnis der Dienstbearbeitung
 0000_{hex} zeigt eine positive Meldung an.
 Die Anschaltbaugruppe hat den Dienst erfolgreich bearbeitet.
 xxxx_{hex} zeigt eine negative Meldung an.
 Die Anschaltbaugruppe konnte den aufgerufenen Dienst nicht bearbeiten.
 Der Parameter *Result* enthält die Ursache, warum der Dienst nicht bearbeitet werden konnte.

Add_Error_Info: Zusätzliche Informationen zur Fehlerursache

Syntax: **Terminate_Load_Process_Data_Reference_List_Confirmation** **8326_{hex}**

Positive Meldung

Wort 1	Code
Wort 2	Parameter_Count
Wort 3	Result

Negative Meldung

Wort 1	Code
Wort 2	Parameter_Count
Wort 3	Result
Wort 4	Add_Error_Info

Bit | 15 0 |

- Legende:
- Code: 8326_{hex} Meldungs-Code der Confirmation
 - Parameter_Count: Anzahl der nachfolgenden Worte
 bei positiver Meldung:
 0001_{hex} 1 Parameterwort
 bei negativer Meldung:
 0002_{hex} 2 Parameterworte
 - Result: Ergebnis der Dienstbearbeitung
 0000_{hex} zeigt eine positive Meldung an.
 Die Anschaltbaugruppe hat den Dienst erfolgreich bearbeitet.
 xxxx_{hex} zeigt eine negative Meldung an.
 Die Anschaltbaugruppe konnte den aufgerufenen Dienst nicht bearbeiten.
 Der Parameter *Result* enthält die Ursache, warum der Dienst nicht bearbeitet werden konnte.
 - Add_Error_Info: Zusätzliche Informationen zur Fehlerursache

2.5.4 Dienst „Read_Process_Data_Reference_List“

Aufgabe: Mit diesem Dienst können Sie die aktuelle Prozessdatenreferenzliste (PDRL) auslesen.

Über den Parameter *Start_PDR_Index* stellen Sie ein, ob Sie

- nur den Kopf der Prozessdatenreferenzliste oder
- eine oder mehrere Prozessdatenreferenzen

auslesen wollen.

Syntax: **Read_Process_Data_Reference_List_Request** 0327_{hex}

Wort 1	Code
Wort 2	Parameter_Count
Wort 3	Start_PDR_Index
Wort 4	Entry_Count

Bit | 15 0 |

Legende:

Code:	0327 _{hex}	Kommando-Code des Requests
Parameter_Count:	0002 _{hex}	Anzahl der nachfolgenden Worte 2 Parameterworte
Start_PDR_Index:	0000 _{hex}	nur Kopf der Prozessdatenreferenz- liste (PDRL) auslesen,
	xxxx _{hex}	Index der ersten auszulesenden Prozessdatenreferenz (PDR).
Entry_Count:		Anzahl der Prozessdatenreferenzen (nur relevant, wenn <i>Start_PDR_Index</i> > 0000 _{hex})

Syntax: **Read_Process_Data_Reference_List_Confirmation** **8327_{hex}**

Positive Meldung und *Start_PDR_Index* > 0000_{hex}

Wort 1	Code	
Wort 2	Parameter_Count	
Wort 3	Result	
Wort 4	More_Follows	
Wort 5	Start_PDR_Index	
Wort 6	PDR_Count	
Wort 7	PDR	Prozessdatenreferenzliste
...	...	
	PDR	

Positive Meldung und *Start_PDR_Index* = 0000_{hex}

Wort 1	Code
Wort 2	Parameter_Count
Wort 3	Result
Wort 4	More_Follows
Wort 5	Start_PDR_Index
Wort 6	PDR_Count

Negative Meldung

Wort 1	Code
Wort 2	Parameter_Count
Wort 3	Result
Wort 4	Add_Error_Info

Bit | 15 0 |

Legende: Code: 8327_{hex} Meldungs-Code der Confirmation
 Parameter_Count: Anzahl der nachfolgenden Worte bei positiver Meldung und *Start_PDR_Index* > 0000_{hex}:
 xxxx_{hex} Der Wert ist abhängig von Anzahl und Größe der Prozessdatenreferenzen.

	bei positiver Meldung und <i>Start_PDR_Index</i> = 0000 _{hex} : 0004 _{hex} 4 Parameterworte
	bei einer negativen Meldung: 0002 _{hex} 2 Parameterworte
Result:	Ergebnis der Dienstbearbeitung 0000 _{hex} zeigt eine positive Meldung an. Die Anschaltbaugruppe konnte den Dienst erfolgreich bearbeiten. xxxx _{hex} zeigt eine negative Meldung an. Die Anschaltbaugruppe konnte den aufgerufenen Dienst nicht bearbeiten. Der Parameter <i>Result</i> enthält die Ursache, warum der Dienst nicht bearbeitet werden konnte.
Add_Error_Info:	Zusätzliche Informationen zur Fehlerursache
More_Follows:	0000 _{hex} zeigt an, dass alle angeforderten Prozessdatenreferenzen in der Confirmation enthalten sind. 0001 _{hex} zeigt an, dass noch nicht alle angeforderten Prozessdatenreferenzen in der Confirmation enthalten sind. Die Datenmenge ist größer als die für Dienste zur Verfügung stehende Mailbox (MXI). Rufen Sie den Dienst erneut auf, um den Rest auszulesen. Geben Sie dabei für den Parameter <i>Start_PDR_Index</i> den Index der letzten übertragenden Prozessdatenreferenz erhöht um 0001 _{hex} an.
Start_PDR_Index:	Index der ersten Prozessdatenreferenz (PDR), die ausgelesen wurde. Zulässige Wertebereiche für <i>PDR_Indices</i> : für Eingangsdaten 0000 _{hex} ... 3FFF _{hex} für Ausgangsdaten 4000 _{hex} ... 7FFF _{hex} für Durchgangsdaten 8000 _{hex} ... BFFF _{hex}
PDR_Count:	Anzahl der Prozessdatenreferenzen (PDR), die ausgelesen wurden, oder in der Prozessdatenreferenzliste enthalten sind.
PDR:	Prozessdatenreferenzen in der Reihenfolge aufsteigender Indizes (siehe Seite 2-83)

2.5.5 Dienst „Compact_Load_Process_Data_Reference_List“

Aufgabe:

Mit diesem Dienst können SPS-Anwender die Anschaltbaugruppe ohne Unterstützung durch Software wie IBS CMD SWT projektieren. Erstellen Sie hierzu je eine Host-Adressliste für die Eingangs- und Ausgangs-Prozessdaten. Aus diesen Listen erstellt die Firmware automatisch entsprechende Einträge für die Prozessdatenreferenzliste. Der Dienst „Compact_Load_Process_Data_Reference_List“ ersetzt die Dienstfolge aus den Diensten:

- „Initiate_Load_Process_Data_Reference_List“ (0324_{hex}),
- „Load_Process_Data_Reference_List“ (0325_{hex}) und
- „Terminate_Load_Process_Data_Reference_List“ (0326_{hex}).

Er ermöglicht eine Adressvergabe, wie sie bei den Anschaltbaugruppen mit Firmware-Version 3.x üblich war.

Übergeben Sie die Adressen im hostspezifischen Adressformat. Die Firmware wandelt das hostspezifische Adressformat in die physikalischen 32-Bit-Adressen des MPMs.

Syntax:

Compact_Load_Process_Data_Reference_List_Request 0328_{hex}

Wort 1	Code		
Wort 2	Parameter_Count		
Wort 3	Address_Direction		
Wort 4	Entry_Count		
Wort 5	Data_Cons	PD_Address	1. Teilnehmer
Wort 6	Data_Cons	PD_Address	2. Teilnehmer
	
	Data_Cons	PD_Address	n. Teilnehmer



Legende:

- Code: 0328_{hex} Kommando-Code des Requests
- Parameter_Count: Anzahl der nachfolgenden Worte
 xxx_{hex} = 2 + Entry_Count
- Address_Direction: Datenrichtung der Adressen:
 1000_{hex} Eingangsadressen
 2000_{hex} Ausgangsadressen

Entry_Count:	Anzahl der Einträge in dieser Host-Adressliste.
Data_Cons:	Datenkonsistenz für den Prozessdatenzugriff
	Zulässige Werte:
00 _{bin}	16 Bit (Standard)
01 _{bin}	32 Bit (z. B. Encoder, Bedienterminals)
10 _{bin}	8 Bit für Module oder Prozessdaten, die kleiner/gleich 8 Bit sind
11 _{bin}	64 Bit (z. B. Encoder, Bedienterminals)



Die Datenkonsistenz garantiert, dass die angegebene Datenbreite aus einem INTERBUS-Zyklus stammt.

Der Standardwert für E/A-Module ist 16 Bit. Für INTERBUS-Teilnehmer, welche die zusammenhängende Übertragung größerer Datenbreiten verlangen, müssen Sie die Datenkonsistenz erhöhen. Dies sind z. B. Encoder, Bedienterminals oder Analogmodule mit mehr als 16 Bit Auflösung. Die Datenkonsistenz muss innerhalb eines Wortes identisch sein.

Sie können ein Wort

- aufteilen in zwei, jeweils in sich konsistente Byte (je 8 Bit),
- als in sich konsistent definieren (16 Bit) oder
- einem größeren Konsistenzbereich (32 oder 64 Bit) zuweisen.

Es ist nicht zulässig, einem Byte eines Wortes eine Datenkonsistenz von 8 Bit und dem anderen Byte einen anderen, größeren Konsistenzbereich zuzuweisen.

PD_Adresse:	Eingangsadressliste (<i>Address_Direction</i> = 1000 _{hex}): Tragen Sie in die Eingangsadressliste für alle Teilnehmer mit Eingangs-Prozessdaten die Host-Adressen in der Reihenfolge der physikalischen Anordnung ein. Für Teilnehmer mit mehr als einem Wort Eingangs-Prozessdaten (z. B. mit 32 Bit) werden automatisch die nächsthöheren Adressen mitbelegt. Tragen Sie für alle Module ohne Eingangs-Prozessdaten (z. B. reine Busklemmen oder reine Ausgangsmodule) in das Wort <i>Data_Cons</i> / <i>PD_Address</i> den Wert FFFF _{hex} ein.
-------------	--

Ausgangsadressliste

(*Address_Direction* = 2000_{hex}):

Tragen Sie hier für alle Teilnehmer mit Ausgangs-Prozessdaten die Host-Adressen in der Reihenfolge der physikalischen Anordnung ein. Für Teilnehmer mit mehr als einem Wort Ausgangs-Prozessdaten (z. B. mit 32 Bit) werden automatisch die nächsthöheren Adressen mitbelegt.

Tragen Sie für alle Module ohne Ausgangs-Prozessdaten (z. B. reine Busklemmen oder reine Eingangsmodule) in das Wort *Data_Cons / PD_Address* den Eintrag FFFF_{hex} ein.



Legen Sie die Startadressen von INTERBUS-Teilnehmern mit 16 Bit (oder mehr) Adressraum nur auf gerade Adressen.

Syntax:

Compact_Load_Process_Data_Reference_List_Confirmation

8328_{hex}

Positive Meldung

Wort 1	Code
Wort 2	Parameter_Count
Wort 3	Result

Negative Meldung

Wort 1	Code
Wort 2	Parameter_Count
Wort 3	Result
Wort 4	Add_Error_Info

Bit | 15 0 |

Legende:

- Code: 8328_{hex} Meldungs-Code der Confirmation
- Parameter_Count: Anzahl der nachfolgenden Worte
 bei positiver Meldung:
 0001_{hex} 1 Parameterwort
 bei negativer Meldung:
 0002_{hex} 2 Parameterworte
- Result: Ergebnis der Dienstbearbeitung
 0000_{hex} zeigt eine positive Meldung an.
 Die Anschaltbaugruppe hat den Dienst erfolgreich bearbeitet.
 xxxx_{hex} zeigt eine negative Meldung an.
 Die Anschaltbaugruppe konnte den aufgerufenen Dienst nicht bearbeiten.
 Der Parameter *Result* enthält die Ursache, warum der Dienst nicht bearbeitet werden konnte.
- Add_Error_Info: Zusätzliche Informationen zur Fehlerursache

2.5.6 Dienst „Compact_Read_Process_Data_Reference_List“

Aufgabe: Mit diesem Dienst können Sie Adresslisten Ihres Steuerungs- oder Rechnersystems auslesen, die mit dem Dienst „Compact_Load_Process_Data_Reference_List“ (0328_{hex}) projiziert wurden. Die Firmware wandelt die physikalischen 32-Bit-Adressen des MPM in das spezifische Adressformat Ihres Steuerungs- oder Rechnersystems um.

Voraussetzung: Dieser Dienst kann nur erfolgreich ausgeführt werden, wenn die Prozessdatenreferenzlisten mit dem Dienst „Compact_Load_Process_Data_Reference_List“ (0328_{hex}) projiziert wurden.

Syntax: **Compact_Read_Process_Data_Reference_List_Request** 0329_{hex}

Wort 1	Code
Wort 2	Parameter_Count
Wort 3	Address_Direction

Bit | 15 0 |

Legende:

Code:	0329 _{hex}	Kommando-Code des Requests
Parameter_Count:	Anzahl der nachfolgenden Worte	
	0001 _{hex}	1 Parameterwort
Address_Direction:	Datenrichtung der Adressen:	
	1000 _{hex}	Eingangsadressliste
	2000 _{hex}	Ausgangsadressliste

Syntax:

Compact_Read_Process_Data_Reference_List_Confirmation

8329_{hex}

Positive Meldung

Wort 1	Code		
Wort 2	Parameter_Count		
Wort 3	Result		
Wort 4	Address_Direction		
Wort 5	Entry_Count		
Wort 6	Data_Cons	PD_Address	1. Teilnehmer
Wort 7	Data_Cons	PD_Address	2. Teilnehmer
...
	Data_Cons	PD_Address	n. Teilnehmer

Negative Meldung

Wort 1	Code	
Wort 2	Parameter_Count	
Wort 3	Result	
Wort 4	Add_Error_Info	

Bit | 15 | 14 | 13 0 |

Legende:

Code: 8329_{hex} Meldungs-Code der Confirmation

Parameter_Count: Anzahl der nachfolgenden Worte
 bei positiver Meldung:
 $xxxx_{hex} = 3 + Entry_Count$
 bei negativer Meldung:
 0002_{hex} 2 Parameterworte

Result: Ergebnis der Dienstbearbeitung
 $xxxx_{hex}$ zeigt eine positive Meldung an.
 Die Anschaltbaugruppe hat den Dienst erfolgreich bearbeitet.
 0002_{hex} zeigt eine negative Meldung an.
 Die Anschaltbaugruppe konnte den aufgerufenen Dienst nicht bearbeiten.
 Der Parameter *Result* enthält die Ursache, warum der Dienst nicht bearbeitet werden konnte.

Add_Error_Info:	Zusätzliche Informationen zur Fehlerursache
Address_Direction:	Datenrichtung der Adressen: 1000 _{hex} Eingangsadressliste 2000 _{hex} Ausgangsadressliste
Entry_Count:	Anzahl der Einträge in der Adressliste
PD_Address:	Eingangsadressliste (<i>Address_Direction</i> = 1000 _{hex}): Die Eingangsadressliste enthält für alle Teilnehmer mit Eingangs-Prozessdaten die Host-Adressen in der Reihenfolge der physikalischen Anordnung. Für Teilnehmer mit mehr als einem Wort Eingangs-Prozessdaten (z. B. mit 32 Bit) werden automatisch die nächsthöheren Adressen mitbelegt. Ausgangsadressliste (<i>Address_Direction</i> = 2000 _{hex}): Tragen Sie hier für alle Teilnehmer mit Ausgangs-Prozessdaten die Host-Adressen in der Reihenfolge der physikalischen Anordnung ein. Für Teilnehmer mit mehr als einem Wort Ausgangs-Prozessdaten (z. B. mit 32 Bit) werden automatisch die nächsthöheren Adressen mitbelegt. Für INTERBUS-Teilnehmer ohne Prozessdaten des jeweiligen Listentyps gibt die Firmware als Adresse den Wert FFFF _{hex} an.

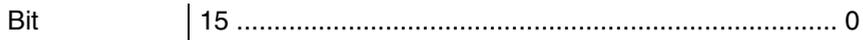
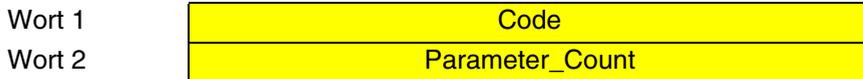
2.6 Dienste für den direkten Zugriff auf den INTERBUS

2.6.1 Dienst „Start_Data_Transfer“

Aufgabe: Dieser Dienst aktiviert den zyklischen Datenverkehr auf dem Bus. Nach Ausführung des Dienstes befindet sich die Anschaltbaugruppe im Zustand *Run* (Anzeige: *RUN*).

Voraussetzung: Die Anschaltbaugruppe muss sich vor Aufruf des Dienstes im Zustand *Active* befinden, d. h. es ist ein Konfigurationsrahmen aktiviert und es werden bereits in regelmäßigen Abständen ID-Zyklen gefahren.

Syntax: **Start_Data_Transfer_Request** **0701_{hex}**



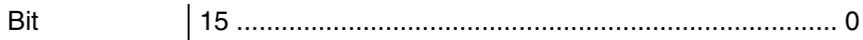
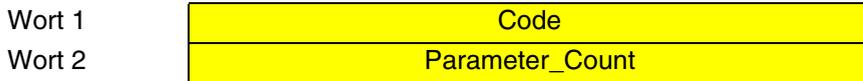
Legende:

Code:	0701 _{hex}	Kommando-Code des Requests
Parameter_Count:		Anzahl der nachfolgenden Worte
	0000 _{hex}	kein Parameterwort

2.6.2 Dienst „Alarm_Stop“

Aufgabe: Dieser Dienst löst auf dem Bus einen langen Reset aus. Der Datenverkehr wird gestoppt. Module mit Prozessdaten setzen ihre Ausgänge auf den Wert NULL. Die Ausführung des Kommandos erfolgt direkt nach der Beendigung der laufenden Datenzyklen. Nach Ausführung des Dienstes befindet sich die Anschaltbaugruppe im Zustand *Ready* (Anzeige: *RDY*).

Syntax: **Alarm_Stop_Request** **1303_{hex}**



Legende:

Code:	1303 _{hex}	Kommando-Code des Requests
Parameter_Count:		Anzahl der nachfolgenden Worte
	0000 _{hex}	kein Parameterwort

Syntax: **Alarm_Stop_Confirmation** 9303_{hex}

Positive Meldung

Wort 1	Code
Wort 2	Parameter_Count
Wort 3	Result

Negative Meldung

Wort 1	Code
Wort 2	Parameter_Count
Wort 3	Result
Wort 4	Add_Error_Info

Bit | 15 0 |

Legende:

Code:	9303 _{hex}	Meldungs-Code der Confirmation
Parameter_Count:		Anzahl der nachfolgenden Worte bei positiver Meldung: 0001 _{hex} 1 Parameterwort bei negativer Meldung: 0002 _{hex} 2 Parameterworte
Result:		Ergebnis der Dienstbearbeitung 0000 _{hex} zeigt eine positive Meldung an. Die Anschaltbaugruppe hat den Dienst erfolgreich bearbeitet. xxxx _{hex} zeigt eine negative Meldung an. Die Anschaltbaugruppe konnte den aufgerufenen Dienst nicht bearbeiten. Der Parameter <i>Result</i> enthält die Ursache, warum der Dienst nicht bearbeitet werden konnte.
Add_Error_Info:		Zusätzliche Informationen zur Fehlerursache

2.6.3 Dienst „Stop_Data_Transfer“

Aufgabe:

Dieser Dienst stoppt den zyklischen Datenverkehr auf dem Bus. Nach Ausführung des Dienstes befindet sich die Anschaltbaugruppe im Zustand *Active* (Anzeige: *ACTV*) und es werden in regelmäßigen Abständen ID-Zyklen gefahren.



Die Anschaltbaugruppe schaltet die Ausgangsdaten nicht in den sicheren Zustand! Gesetzte Ausgänge werden **nicht zurückgesetzt**, sondern bleiben statisch erhalten.

Voraussetzung:

Die Anschaltbaugruppe muss sich vor Aufruf des Dienstes im Zustand *Run* befinden.

Syntax:

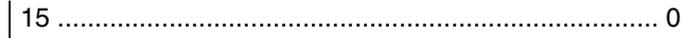
Stop_Data_Transfer_Request

0702_{hex}

Wort 1
Wort 2
Wort 3

Code
Parameter_Count
Stop_Type

Bit



Legende:

- Code: 0702_{hex} Kommando-Code des Requests
- Parameter_Count: Anzahl der nachfolgenden Worte
0001_{hex} 1 Parameterwort
- Stop_Type: Art des Stopps
0000_{hex} Data-Transfer-Stop

Syntax: **Stop_Data_Transfer_Confirmation** **8702_{hex}**

Positive Meldung

Wort 1	Code
Wort 2	Parameter_Count
Wort 3	Result

Negative Meldung

Wort 1	Code
Wort 2	Parameter_Count
Wort 3	Result
Wort 4	Add_Error_Info

Bit | 15 0 |

Legende:

Code: 8702_{hex} Meldungs-Code der Confirmation

Parameter_Count: Anzahl der nachfolgenden Worte
bei positiver Meldung:
0001_{hex} 1 Parameterwort
bei negativer Meldung:
0002_{hex} 2 Parameterworte

Result: Ergebnis der Dienstbearbeitung
0000_{hex} zeigt eine positive Meldung an.
Die Anschaltbaugruppe hat den Dienst erfolgreich bearbeitet.
xxxx_{hex} zeigt eine negative Meldung an.
Die Anschaltbaugruppe konnte den aufgerufenen Dienst nicht bearbeiten.
Der Parameter *Result* enthält die Ursache, warum der Dienst nicht bearbeitet werden konnte.

Add_Error_Info: Zusätzliche Informationen zur Fehlerursache

2.6.4 Dienst „Control_Active_Configuration“

Aufgabe:

Mit diesem Dienst können Sie INTERBUS-Teilnehmer gezielt ein- oder abschalten. Abhängig davon, ob dieser Dienst im aktiven oder inaktiven Zustand des Bussystems benutzt wird, werden die Veränderungen im Konfigurationsrahmen sofort oder erst beim Aufruf des Dienstes „Activate_Configuration“ (0711_{hex}) wirksam.

Syntax:

Control_Active_Configuration_Request

0713_{hex}

Wort 1	Code	Liste der INTERBUS-Teilnehmernummern
Wort 2	Parameter_Count	
Wort 3	Switch_Code	
Wort 4	Entry_Count	
Wort 5	Device_No	
Wort 6	Device_No	
...	...	
	Device_No	

Bit | 15 0 |

Legende:

- Code: 0713_{hex} Kommando-Code des Requests
- Parameter_Count: Anzahl der nachfolgenden Worte
 $xxxx_{hex} = 2 + Entry_Count$
- Switch_Code: Kennung, wie die aufgeführten INTERBUS-Teilnehmer geschaltet werden sollen
 Mögliche Werte:
 0000_{hex} Segment_Off
 Der angegebene und alle von ihm abhängigen Teilnehmer werden abgeschaltet. Dies sind:
 - alle Teilnehmer, die zum gleichen Bussegment gehören,
 - alle Teilnehmer, die zur gleichen logischen Gruppe gehören,
 - Teilnehmer, die im Datenring physikalisch hinter dem Teilnehmer *Device_No* x angeordnet sind.

0001 _{hex}	Segment_On Der angegebene und alle von ihm abhängigen Teilnehmer werden aktiv geschaltet. Beachten Sie die Sonderbehandlung bei alternativ schaltbaren Gruppen.
0002 _{hex}	Device_Off Nur der angegebene Teilnehmer wird im Konfigurationsrahmen inaktiv geschaltet. Dazu müssen alle von diesem Teilnehmer abhängigen Teilnehmer einzeln und manuell abgeschaltet werden.
0003 _{hex}	Device_On Nur der angegebene Teilnehmer wird im Konfigurationsrahmen aktiv geschaltet. Dazu müssen alle von diesem Teilnehmer abhängigen Teilnehmer einzeln und manuell aktiv geschaltet werden.
0004 _{hex}	Device_Disable Nur der angegebene Teilnehmer wird im Konfigurationsrahmen inaktiv geschaltet. Er darf physikalisch nicht im Datenring verbleiben und muss manuell überbrückt werden.
0005 _{hex}	Device_Enable Nur der angegebene Teilnehmer wird im Konfigurationsrahmen aktiv geschaltet. Fügen Sie ihn manuell wieder in den Datenring ein.
Entry_Count:	Anzahl der Teilnehmer in der Liste der INTERBUS-Teilnehmernummern.
Device_No:	INTERBUS-Teilnehmernummer der Teilnehmer, die geschaltet werden sollen.

Syntax: **Control_Active_Configuration_Confirmation** **8713_{hex}**

Positive Meldung

Wort 1	Code
Wort 2	Parameter_Count
Wort 3	Result

Negative Meldung

Wort 1	Code
Wort 2	Parameter_Count
Wort 3	Result
Wort 4	Add_Error_Info

Bit | 15 0 |

Legende:

Code:	8713 _{hex}	Meldungs-Code der Confirmation
Parameter_Count:		Anzahl der nachfolgenden Worte bei positiver Meldung: 0001 _{hex} 1 Parameterwort bei negativer Meldung: 0002 _{hex} 2 Parameterworte
Result:		Ergebnis der Dienstbearbeitung 0000 _{hex} zeigt eine positive Meldung an. Die Anschaltbaugruppe hat den Dienst erfolgreich bearbeitet. xxxx _{hex} zeigt eine negative Meldung an. Die Anschaltbaugruppe konnte den aufgerufenen Dienst nicht bearbeiten. Der Parameter <i>Result</i> enthält die Ursache, warum der Dienst nicht bearbeitet werden konnte.
Add_Error_Info:		Zusätzliche Informationen zur Fehlerursache

2.6.5 Dienst „Control_Device_Function“

Aufgabe: Mit diesem Dienst können Sie an einen oder mehrere INTERBUS-Teilnehmer Steuerbefehle senden, um z. B. Geräte-Statusfehler zu quittieren oder einen Alarm-Ausgang zu setzen.

Mit dem Code 0314_{hex} ist ein identischer Dienst „Control_Device_Function_Not_Exclusive“ als nicht exklusiv aufrufbar.

Syntax: **Control_Device_Function_Request** **0714_{hex}**

Wort 1	Code	Liste der INTERBUS-Teilnehmer
Wort 2	Parameter_Count	
Wort 3	Device_Function	
Wort 4	Entry_Count	
Wort 5	Device_No	
Wort 6	Device_No	
...	...	
	Device_No	
Bit	15 0	

Legende:

Code: 0714_{hex} Kommando-Code des Requests

Parameter_Count: Anzahl der nachfolgenden Worte
 $xxxx_{hex} = 2 + Entry_Count$

Device_Function: Steuerbefehl, der zu den Teilnehmern gesendet wird.

Folgende Steuerbefehle stehen zur Verfügung:

- 0001_{hex} Set_Alarm
Alarm-Ausgänge der Busklemmen setzen.
- 0002_{hex} Reset_Alarm
Alarm-Ausgänge der Busklemmen zurücksetzen.
- 0003_{hex} Conf_Dev_Err
Peripheriefehler (PF) eines Gerätes quittieren.
- 0004_{hex} Conf_Dev_Err_All
Peripheriefehler (PF) aller Geräte quittieren.

Da die Angabe von Teilnehmern nicht notwendig ist, entfällt bei diesem Steuerbefehl die Liste der Teilnehmernummern.

Setzen Sie den Parameter

Entry_Count = 0000_{hex}.

0005_{hex} reserviert

0006_{hex} reserviert

0007_{hex} Refreshen der optischen Diagnose

Entry_Count:

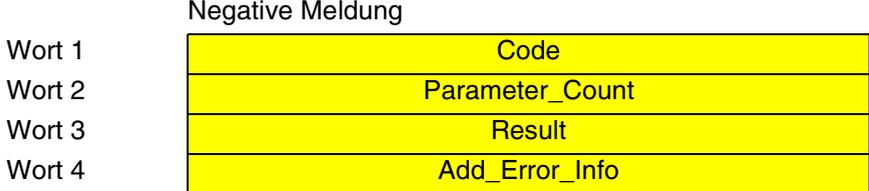
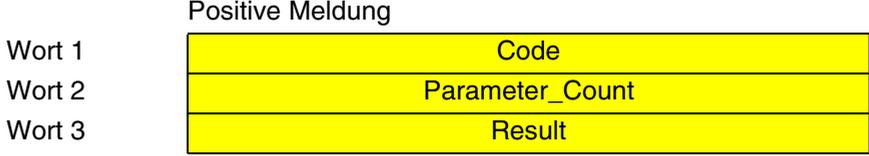
Anzahl der Teilnehmer, an die der Steuerbefehl gesendet wird .

Setzen Sie diesen Parameter auf 0000_{hex}, wenn Parameter *Device_Function* = 0004_{hex} ist.

Device_No:

Teilnehmernummern der Teilnehmer, zu denen der Steuerbefehl gesendet wird (entfällt, wenn *Device_Function* = 0004_{hex}).

Syntax: **Control_Device_Function_Confirmation** **8714_{hex}**



Legende:

Code: 8714_{hex} Meldungs-Code der Confirmation

Parameter_Count: Anzahl der nachfolgenden Worte
bei positiver Meldung:
0001_{hex} 1 Parameterwort
bei negativer Meldung:
0002_{hex} 2 Parameterworte

Result: Ergebnis der Dienstbearbeitung
0000_{hex} zeigt eine positive Meldung an.
Die Anschaltbaugruppe hat den Dienst erfolgreich bearbeitet.
xxxx_{hex} zeigt eine negative Meldung an.
Die Anschaltbaugruppe konnte den aufgerufenen Dienst nicht bearbeiten.
Der Parameter *Result* enthält die Ursache, warum der Dienst nicht bearbeitet werden konnte.

Add_Error_Info: Zusätzliche Informationen zur Fehlerursache

2.6.6 Dienst „Reset_Controller_Board“

Aufgabe: Mit diesem Dienst „Reset_Controller_Board“ lösen Sie einen Reset der Anschaltbaugruppe aus.

Voraussetzung: Stellen Sie vor dem Aufruf des Dienstes sicher, dass der Zustand Ihrer Anlage den Reset der Anschaltbaugruppe erlaubt.

Syntax: **Reset_Controller_Board_Request** **0956_{hex}**

Wort 1	Code
Wort 2	Parameter_Count
Wort 3	Reset_Type

Bit | 15 0 |

Legende:

Code:	0956 _{hex}	Kommando-Code des Requests
Parameter_Count:		Anzahl der nachfolgenden Worte
	0001 _{hex}	1 Parameterwort
Reset_Type:		Art des Resets
	0000 _{hex}	Kaltstart
	0001 _{hex}	Warmstart



Dieser Dienst wird nicht quittiert. Stellen Sie sicher, dass ein neuer Zugriff auf die Anschaltbaugruppe erst nach einer Wartezeit von **7** Sekunden erfolgt. Verriegeln Sie während dieser Wartezeit den Zugriff auf die Anschaltbaugruppe. Während dieser Zeit dürfen Sie keine Mailbox-Funktionen aufrufen.

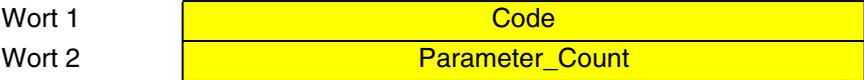
Bei Treiberaufrufen während dieser Zeit kann es zu Fehlsteuerungen des INTERBUS-Systems kommen, die den Anlauf der Anschaltbaugruppe verhindern.

2.7 Dienste zur Diagnose

2.7.1 Dienst „Confirm_Diagnostics“

Aufgabe: Dieser Dienst aktualisiert den Inhalt der Diagnose-Register, die Fehlerbereiche im MPM und die Diagnose-Anzeigen auf der Frontblende der Anschaltbaugruppe.

Syntax: **Confirm_Diagnostics_Request** **0760_{hex}**



Legende:

Code:	0760 _{hex}	Kommando-Code des Requests
Parameter_Count:		Anzahl der nachfolgenden Worte
	0000 _{hex}	kein Parameterwort

Syntax: **Confirm_Diagnostics_Confirmation** **8760_{hex}**

Positive Meldung

Wort 1	Code
Wort 2	Parameter_Count
Wort 3	Result

Negative Meldung

Wort 1	Code
Wort 2	Parameter_Count
Wort 3	Result
Wort 4	Add_Error_Info

Bit | 15 0 |

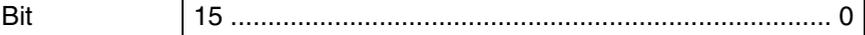
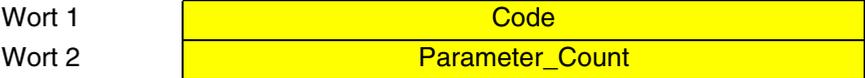
Legende:

Code:	8760 _{hex}	Meldungs-Code der Confirmation
Parameter_Count:		Anzahl der nachfolgenden Worte bei positiver Meldung: 0001 _{hex} 1 Parameterwort bei negativer Meldung: 0002 _{hex} 2 Parameterworte
Result:		Ergebnis der Dienstbearbeitung 0000 _{hex} zeigt eine positive Meldung an. Die Anschaltbaugruppe hat den Dienst erfolgreich bearbeitet. xxxx _{hex} zeigt eine negative Meldung an. Die Anschaltbaugruppe konnte den aufgerufenen Dienst nicht bearbeiten. Der Parameter <i>Result</i> enthält die Ursache, warum der Dienst nicht bearbeitet werden konnte.
Add_Error_Info:		Zusätzliche Informationen zur Fehlerursache

2.7.2 Dienst „Get_Error_Info“

Aufgabe: Mit diesem Dienst können Sie nach der Meldung eines Busfehlers die genaue Fehlerursache und den genauen Fehlerort auslesen. Maximal werden zehn Fehler analysiert.

Syntax: **Get_Error_Info_Request** **0316_{hex}**



Legende:
Code: 0316_{hex} Kommando-Code des Requests
Parameter_Count: Anzahl der nachfolgenden Worte
0000_{hex} kein Parameterwort

Syntax:

Get_Error_Info_Confirmation

8316_{hex}

Positive Meldung, solange Fehlerlokalisierung nicht abgeschlossen ist

Wort 1	Code	
Wort 2	Parameter_Count	
Wort 3	Result	
Wort 4	Entry_Count	= 0001 _{hex}
Wort 5	Error_Code	= 0BDF _{hex}
Wort 6	Add_Error_Info	= FFFF _{hex}

Positive Meldung, wenn Fehlerlokalisierung abgeschlossen ist

Wort 1	Code	
Wort 2	Parameter_Count	
Wort 3	Result	
Wort 4	Entry_Count	
Wort 5	Error_Code	1. Fehler
Wort 6	Add_Error_Info	
Wort 7	Error_Code	2. Fehler
Wort 8	Add_Error_Info	
...
	Error_Code	n. Fehler
	Add_Error_Info	

Negative Meldung

Wort 1	Code
Wort 2	Parameter_Count
Wort 3	Result
Wort 4	Add_Error_Info

Bit | 15 0 |

Legende:	Code:	8316 _{hex} Meldungs-Code der Confirmation
	Parameter_Count:	Anzahl der nachfolgenden Worte bei positiver Meldung (während der Fehlerlokalisierung): 0004 _{hex} 4 Parameterworte bei positiver Meldung (nach der Fehlerlokalisierung): 00xx _{hex} = 2 + 2 × <i>Entry_Count</i> (max. 20 Worte) bei negativer Meldung: 0002 _{hex} immer 2 Parameterworte
	Result:	Ergebnis der Dienstbearbeitung 0000 _{hex} zeigt eine positive Meldung an. Die Anschaltbaugruppe hat den Dienst erfolgreich bearbeitet. xxxx _{hex} zeigt eine negative Meldung an. Die Anschaltbaugruppe konnte den aufgerufenen Dienst nicht bearbeiten. Der Parameter <i>Result</i> enthält die Ursache, warum der Dienst nicht bearbeitet werden konnte.
	Entry_Count:	Anzahl der angezeigten Fehler (maximal 10) Jede Fehlerbeschreibung besteht aus den zwei Worten <i>Error_Code</i> und <i>Add_Error_Info</i> .
	Error_Code:	Informationen zur Fehlerart
	Add_Error_Info:	Bei positiver Meldung: Fehlerort (<i>Bussegment . Position</i>), falls dieser lokalisiert werden konnte. Bei negativer Meldung: Zusätzliche Informationen zur Fehlerursache

2.7.3 Dienst „Read_Device_State“

Aufgabe: Dieser Dienst liest zu Test- und Diagnose-Zwecken auswählbare Statusinformationen aller INTERBUS-Teilnehmer der aktiven Konfiguration aus.

Syntax: **Read_Device_State_Request** **0315_{hex}**

Wort 1	Code
Wort 2	Parameter_Count
Wort 3	Device_State_Mask

Bit | 15 0 |

Legende:

Code: 0315_{hex} Kommando-Code des Requests

Parameter_Count: Anzahl der nachfolgenden Worte
0001_{hex} 1 Parameterwort

Device_State_Mask: Zu lesende Statusinformationen

Der Parameter beschreibt ein 16 Bit breites Feld, wobei jedes Bit einer Statusinformation entspricht. Setzen Sie bei den Statusinformationen, die Sie lesen wollen, die entsprechenden Bits auf 1.

Zuordnungen:

- Bit 0: Alarm Output
Der Alarmausgang des angegebenen INTERBUS-Teilnehmers ist gesetzt.
- Bit 1: Error Output
Der angegebene INTERBUS-Teilnehmer meldet einen Fehler.
- Bit 2: IVC-Detection
Die Spannung für die Initiatoren ist zu gering.
- Bit 3: TEMP-Detection
Die Umgebungstemperatur ist zu hoch.
- Bit 4: CSD-Detection
Die interne Stromquelle ist überlastet.

- Bit 5: PSD_Detection
Der erlaubte Ausgangsstrom des Leistungstreibers wurde überschritten.
- Bit 6-8: reserviert (immer 0_{bin}!)
- Bit 9: MAU-Detection der ankommenden Fernbus-Schnittstelle (Datenring-Hinweg)
MAU = **M**edium **A**ttachment **U**nit
Das gesetzte Bit zeigt an, das die Übertragungstrecke an der angegebenen Schnittstelle gerade noch funktionsfähig ist und eine zu hohe Dämpfung aufweist, z. B. bei LWL-Strecken.
- Bit 10: MAU-Detection der ankommenden Fernbus-Schnittstelle (Datenring-Rückweg)
- Bit 11: Peripheral Fault
Der INTERBUS-Teilnehmer meldet je nach Wert des Bits 12 (Peripheral Fault Mode) einen Peripheriefehler oder einen Mikroprozessor-Reset.
- Bit 12: Peripheral Fault-Mode
0_{bin} Bit 11 (Peripheral Fault) meldet einen Peripheriefehler.
1_{bin} Bit 11 (Peripheral Fault) meldet einen Mikroprozessor-Reset.
- Bit 13: Peripheral Fault-Extension
Wenn Bit 13 und Bit 11 (Peripheral Fault) gesetzt sind, liefert die Einzelkanal-Diagnose zusätzliche Details über den Peripheriefehler. Diese Daten können Sie mit dem Dienst „Read_Configuration_Request“ (0309_{hex}) auslesen (siehe Seite 2-38).
- Bit 14,15: reserviert (immer 0_{bin}!)

Syntax:

Read_Device_State_Confirmation

8315_{hex}

Positive Meldung

Wort 1	Code	
Wort 2	Parameter_Count	
Wort 3	Result	
Wort 4	More_Follows	
Wort 5	States_Count	
Wort 6	Device_No	1. Meldung
Wort 7	Device_States	
...	...	
	Device_No	n. Meldung
	Device_States	

Negative Meldung

Wort 1	Code
Wort 2	Parameter_Count
Wort 3	Result
Wort 4	Add_Error_Info

Bit | 15 0 |

Legende:

Code: 8315_{hex} Meldungs-Code der Confirmation

Parameter_Count: Anzahl der nachfolgenden Worte
 bei positiver Meldung:
 $xxxx_{hex} = 3 + 2 \times States_Count$
 bei negativer Meldung:
 0002_{hex} 2 Parameterworte

Result: Ergebnis der Dienstbearbeitung
 0000_{hex} zeigt eine positive Meldung an.
 Die Anschaltbaugruppe hat den Dienst erfolgreich bearbeitet.
 xxxx_{hex} zeigt eine negative Meldung an.
 Die Anschaltbaugruppe konnte den aufgerufenen Dienst nicht bearbeiten.
 Der Parameter *Result* enthält die Ursache, warum der Dienst nicht bearbeitet werden konnte.

Add_Error_Info:	Zusätzliche Informationen zur Fehlerursache
More_Follows:	0000 _{hex} zeigt an, dass alle angeforderten Statusmeldungen in der Confirmation enthalten sind. 0001 _{hex} zeigt an, dass noch nicht alle angeforderten Statusmeldungen in der Confirmation enthalten sind. Die Datenmenge ist größer als die für Dienste zur Verfügung stehende Mailbox (MXI). Quittieren Sie die übertragenen Statusmeldungen mit dem Dienst „Control_Device_Function“ (0714 _{hex}) und rufen Sie danach den Dienst „Read_Device_State“ (0315 _{hex}) erneut auf.
States_Count:	Anzahl der Statusmeldungen Jede Statusmeldung besteht aus den zwei Parametern <i>Device_No</i> und <i>Device_States</i> .
Device_No:	Nummer des INTERBUS-Teilnehmers, der die Statusinformationen geliefert hat.
Device_States:	Statusinformationen des Teilnehmers Die Bedeutung der Bits entspricht der des Parameters <i>Device_State_Mask</i> in dem Request.

2.7.4 Dienst „Get_Version_Info“

Aufgabe: Mit diesem Dienst können Sie Versionsinformationen wie Typ, Versionsstand, Herstellungsdatum usw. von Hardware und Firmware Ihrer Anschaltbaugruppe auslesen.

Syntax: **Get_Version_Info_Request** **032A_{hex}**

Wort 1	Code
Wort 2	Parameter_Count

Bit | 15 0 |

Legende: Code: 032A_{hex} Kommando-Code des Requests
 Parameter_Count: Anzahl der nachfolgenden Worte
 0000_{hex} kein Parameterwort

Syntax: **Get_Version_Info_Confirmation** **832A_{hex}**

Positive Meldung

Wort 1	Code			
Wort 2	Parameter_Count			
Wort 3	Result			
Wort 4 + 5	FW_Version	(Byte 1)	FW_Version	(Byte 2)
	FW_Version	(Byte 3)	FW_Version	(Byte 4)
Wort 6 ... 8	FW_State	(Byte 1)	...	
	...		FW_State	(Byte 6)
Wort 9 ... 11	FW_Date	(Byte 1)	...	
	...		FW_Date	(Byte 6)
Wort 12 ... 14	FW_Time	(Byte 1)	...	
	...		FW_Time	(Byte 6)
Wort 15 ... 24	Host_Type	(Byte 1)	...	
	...		Host_Type	(Byte 20)
Wort 25 + 26	Host_Version	(Byte 1)	Host_Version	(Byte 2)
	Host_Version	(Byte 3)	Host_Version	(Byte 4)

Wort 27 ... 29	Host_State (Byte 1)
	...	Host_State (Byte 6)
Wort 30 ... 32	Host_Date (Byte 1)
	...	Host_Date (Byte 6)
Wort 33 ... 35	Host_Time (Byte 1)
	...	Host_Time (Byte 6)
Wort 36 + 37	Start_FW_Version (Byte 1) Start_FW_Version (Byte 2)	...
	Start_FW_Version (Byte 3) Start_FW_Version (Byte 4)	...
Wort 38 ... 40	Start_FW_State (Byte 1)
	...	Start_FW_State (Byte 6)
Wort 41 ... 43	Start_FW_Date (Byte 1)
	...	Start_FW_Date (Byte 6)
Wort 44 ... 46	Start_FW_Time (Byte 1)
	...	Start_FW_Time (Byte 6)
Wort 47 ... 50	HW_Art_No (Byte 1)
	...	HW_Art_No (Byte 8)
Wort 51 ... 65	HW_Art_Name (Byte 1)
	...	HW_Art_Name (Byte 30)
Wort 66 + 67	HW_Motherboard_ID (Byte 1) HW_Motherboard_ID (Byte 2)	...
	HW_Motherboard_ID (Byte 2) HW_Motherboard_ID (Byte 4)	...
Wort 68	HW_Version (Byte 1) HW_Version (Byte 2)	...
	HW_Vendor_Name (Byte 1)
Wort 69 ... 78	...	HW_Vendor_Name (Byte 20)
	HW_Serial_No (Byte 1)
Wort 79 ... 84	...	HW_Serial_No (Byte 12)
	HW_Date (Byte 1)
Wort 85 ... 87	...	HW_Date (Byte 6)

Bit | 15 0 |

Negative Meldung

Wort 1	Code
Wort 2	Parameter_Count
Wort 3	Result
Wort 4	Add_Error_Info

Bit | 15 0 |

- Legende:
- Code: 832A_{hex} Meldungs-Code der Confirmation
 - Parameter_Count: Anzahl der nachfolgenden Worte
 bei positiver Meldung:
 0055_{hex} 55 Parameterworte
 bei negativer Meldung:
 0002_{hex} 2 Parameterworte
 - Result: Ergebnis der Dienstbearbeitung
 0000_{hex} zeigt eine positive Meldung an.
 Die Anschaltbaugruppe hat den Dienst erfolgreich bearbeitet.
 xxxx_{hex} zeigt eine negative Meldung an.
 Die Anschaltbaugruppe konnte den aufgerufenen Dienst nicht bearbeiten.
 Der Parameter *Result* enthält die Ursache, warum der Dienst nicht bearbeitet werden konnte.
 - Add_Error_Info: Zusätzliche Informationen zur Fehlerursache

Versionsinformationen zu Hardware und Firmware. Jedes Byte gibt den ASCII-Code für ein Zeichen an:

- FW_Version: Version des Firmware-Kerns (4 Byte)
 (z. B. 33 2E 39 37_{hex} für „Version 3.97“)
- FW_State: Status der Firmware (6 Byte)
 (z. B. 62 65 64 61 00 00_{hex} für „beta“ bei Vorabversionen)
- FW_Date: Erstellungsdatum der Firmware (6 Byte)
 (z. B. 33 31 31 30 39 35_{hex} für den 31.10.95)

FW_Time:	Erstellungsuhrzeit der Firmware (z. B. 31 34 31 30 32 30 _{hex} für 14:10:20 Uhr)	(6 Byte)
Host_Type:	Typ des hostspezifischen Firmware- Interfaces (z. B. IBS MA S5 DSC)	(20 Byte)
Host_Version:	Version des hostspezifischen Firmware-Interfaces	(4 Byte)
Host_State:	Status des hostspezifischen Firmware-Interfaces	(6 Byte)
Host_Date:	Erstellungsdatum des host- spezifischen Firmware-Interfaces	(6 Byte)
Host_Time:	Erstellungsuhrzeit des host- spezifischen Firmware-Interfaces	(6 Byte)
Start_FW_Version:	Version der Start-Firmware	(4 Byte)
Start_FW_State:	Status der Start-Firmware	(6 Byte)
Start_FW_Date:	Erstellungsdatum der Start-Firmware	(6 Byte)
Start_FW_Time:	Erstellungsuhrzeit der Start-Firmware	(6 Byte)
HW_Art_No:	Artikelnummer der Anschaltbaugruppe	(8 Byte)
HW_Art_Name:	Artikelbezeichnung der Anschaltbaugruppe	(30 Byte)
HW_Motherboard_ID:	Identifikation der Hauptplatine (z. B. 32 43 _{hex} für „2C“ bei IBS S5 DSC)	(4 Byte)
HW_Version:	Version der Hardware	(2 Byte)
HW_Vendor_Name:	Hersteller der Anschaltbaugruppe	(20 Byte)
HW_Serial_No:	Seriennummer der Anschaltbaugruppe	(12 Byte)
HW_Date:	Herstellungsdatum der Anschaltbaugruppe	(6 Byte)

2.7.5 Dienst „Get_Diag_Info“

Aufgabe:

Mit diesem Dienst können Sie statistische, globale Diagnose-Informationen eines Konfigurationsrahmens auslesen. Die positive Meldung der Confirmation enthält die Diagnose-Informationen in mehreren, logisch zusammengehörenden Blöcken. Jeder Block enthält einen Zähler (*Changed_Info_Count*, kurz *CIC*), der bei einer Änderung der Diagnose-Information dieses Blockes inkrementiert wird.

Syntax:

Get_Diag_Info_Request

032B_{hex}

Wort 1	Code
Wort 2	Parameter_Count
Wort 3	Frame_Reference
Wort 4	Diag_Info_Attr
Bit	15 0

Legende:

- Code: 032B_{hex} Kommando-Code des Requests
- Parameter_Count: Anzahl der nachfolgenden Worte
0002_{hex} 2 Parameterworte
- Frame_Reference: Nummer des Konfigurationsrahmens, dessen Diagnose-Informationen Sie auslesen wollen
- Diag_Info_Attr: Auswahl der Diagnose-Informationen, die ausgelesen werden sollen.
Der Parameter beschreibt ein 16 Bit breites Feld, wobei jedes Bit einem Diagnose-Block entspricht. Setzen Sie bei den Diagnose-Blöcken, die Sie auslesen wollen, die entsprechenden Bits auf 1.
Zuordnung (die wichtigsten sind fett hervorgehoben):
 - Bit 0 Changed_Info_Count (CIC)
 - Bit 1 Statistics_State
 - Bit 2 Global_Count
 - Bit 3 **Top_Ten_CRC**
 - Bit 4 **Last_Ten_PF**
 - Bit 5 IPMS_Error
 - Bit 6 **Bus_Error**
 - Bit 7 PF_Error (Peripheral Fault)
 - Bit 8 Time_Out_Error
 - Bit 9...15 reserviert (immer 0_{bin}!)



Die folgende Dienstbeschreibung zeigt die positive Meldung der Confirmation für den Fall, das Sie mit dem Parameter *Diag_Info_Attr* alle zulässigen Attribute freigeschaltet haben.

Schalten Sie in der Praxis immer nur ein Attribut frei, um die Diagnose-Informationen gezielt und übersichtlich auszulesen. Dabei sollten Sie zuerst den Diagnose-Block **Changed_Info_Count** auslesen, um zu erfahren, welche Informationen sich geändert haben.

Syntax:

Get_Diag_Info_Confirmation

832B_{hex}

Positive Meldung

Wort 1	Code		
Wort 2	Parameter_Count		
Wort 3	Result		
Wort 4	Frame_Reference		
Wort 5	Diag_Info_Attr		
Wort 6	Diag_Block	Diagnose-Blöcke wie im Attribut <i>Diag_Info_Attr</i> festgelegt	
...			

Negative Meldung

Wort 1	Code
Wort 2	Parameter_Count
Wort 3	Result
Wort 4	Add_Error_Info

Bit | 15 0 |

Legende:

- Code: 832A_{hex} Meldungs-Code der Confirmation
- Parameter_Count: Anzahl der nachfolgenden Worte bei positiver Meldung:
 - xxxx_{hex} Der Wert ist abhängig von der Anzahl und der Größe der ausgewählten Diagnose-Blöcke.
 - bei negativer Meldung:
 - 0002_{hex} 2 Parameterworte

Result: Ergebnis der Dienstbearbeitung
 0000_{hex} zeigt eine positive Meldung an. Die Anschaltbaugruppe hat den Dienst erfolgreich bearbeitet.
 xxxx_{hex} zeigt eine negative Meldung an. Die Anschaltbaugruppe konnte den aufgerufenen Dienst nicht bearbeiten. Der Parameter *Result* enthält die Ursache, warum der Dienst nicht bearbeitet werden konnte.

Add_Error_Info: Zusätzliche Informationen zur Fehlerursache

Frame_Reference: Konfigurationsrahmen, aus dem die Diagnose-Informationen gelesen wurden
 Der Parameter enthält den Wert, der mit dem Request übergeben wurde.

Diag_Info_Attr: Auswahl der ausgelesenen Diagnose-Informationen
 Der Parameter enthält den Wert, der mit dem Request übergeben wurde.

Diag_Block: Alle angeforderten Diagnose-Informationen in Form von Diagnose-Blöcken, die im folgenden beschrieben sind. Die einzelnen Diagnose-Blöcke sind in der Reihenfolge der Bitnummern des Parameters *Diag_Info_Attributes* aufgeführt.

Syntax

Diagnose-Block *Changed_Info_Count*

Wort 1 + 2	Top_Ten_CRC_CIC	(Bit 31 ... 16)	Changed_- Info_Count
	Top_Ten_CRC_CIC	(Bit 15 ... 0)	
Wort 3 + 4	Last_Ten_PF_CIC	(Bit 31 ... 16)	
	Last_Ten_PF_CIC	(Bit 15 ... 0)	
Wort 5 + 6	IPMS_Error_CIC	(Bit 31 ... 16)	
	IPMS_Error_CIC	(Bit 15 ... 0)	
Wort 7 + 8	Bus_Error_CIC	(Bit 31 ... 16)	
	Bus_Error_CIC	(Bit 15 ... 0)	
Wort 9 + 10	PF_CIC	(Bit 31 ... 16)	
	PF_CIC	(Bit 15 ... 0)	
Wort 11 +12	Time_Out_Error_CIC	(Bit 31 ... 16)	
	Time_Out_Error_CIC	(Bit 15 ... 0)	

Der Diagnose-Block enthält die Zählerstände der 32-Bit-Zähler für die Diagnose-Blöcke. Die Zähler werden jedesmal um den Wert „+1“ erhöht, wenn sich die Informationen des zugehörigen Diagnose-Blocks ändern.

- Top_Ten_CRC_CIC: Zähler für Diagnose-Block *Top_Ten_CRC*
- Last_Ten_PF_CIC: Zähler für Diagnose-Block *Last_Ten_PF*
- IPMS_Error_CIC: Zähler für Diagnose-Block *IPMS_Error*
- Bus_Error_CIC: Zähler für Diagnose-Block *Bus_Error*
- PF_CIC: Zähler für Diagnose-Block *PF*
- Time_Out_Error_CIC: Zähler für Diagnose-Block *Time_Out_Error*

Syntax

Diagnose-Block *Statistics_State*

Wort 1

Statistics_State

- Statistics_State: Status der statistischen Diagnose:
 - 0000_{hex} statistische Diagnose ist aktiviert.
 - 0001_{hex} statistische Diagnose ist deaktiviert.

Syntax

Diagnose-Block *Global_Count*

Wort 1 + 2

Cycle_Count	(Bit 31 ... 16)
Cycle_Count	(Bit 15 ... 0)

Wort 3 + 4

Cycle_Error_Count	(Bit 31 ... 16)
Cycle_Error_Count	(Bit 15 ... 0)

Wort 5 + 6

ID_Cycle_Count	(Bit 31 ... 16)
ID_Cycle_Count	(Bit 15 ... 0)

Wort 7 + 8

ID_Cycle_Error_Count	(Bit 31 ... 16)
ID_Cycle_Error_Count	(Bit 15 ... 0)

Wort 9 + 10

Data_Cycle_Count	(Bit 31 ... 16)
Data_Cycle_Count	(Bit 15 ... 0)

Wort 11 + 12

Data_Cycle_Error_Count	(Bit 31 ... 16)
Data_Cycle_Error_Count	(Bit 15 ... 0)

Global_Count

Der Diagnose-Block *Global_Count* enthält die Anzahl der INTERBUS-Zyklen (Daten- und ID-Zyklen) als 32 Bit-Zählerstände.

- Cycle_Count: Zähler für INTERBUS-Zyklen
- Cycle_Error_Count: Zähler für fehlerhafte INTERBUS-Zyklen

ID_Cycle_Count: Zähler für ID-Zyklen
 ID_Cycle_Error_Count: Zähler für fehlerhafte ID-Zyklen
 Data_Cycle_Count: Zähler für Daten-Zyklen
 Data_Cycle_Error_Count: Zähler für fehlerhafte Daten-Zyklen

Syntax

Diagnose-Block *Top_Ten_CRC*

Wort 1 + 2	Top_Ten_CRC_CIC	(Bit 31 ... 16)	Top_Ten_- CRC
	Top_Ten_CRC_CIC	(Bit 15 ... 0)	
Wort 3	1. Device_No_Bus_Fault		
	Error_Count	(Bit 31 ... 16)	
Wort 4 + 5	Error_Count	(Bit 15 ... 0)	
Wort 6	Error_Code		
...	...		
Wort 39	10. Device_No_Bus_Fault		
	Error_Count	(Bit 31 ... 16)	
Wort 40 + 41	Error_Count	(Bit 15 ... 0)	
Wort 42	Error_Code		

Der Diagnose-Block *Top_Ten_CRC* enthält die Liste der INTERBUS-Teilnehmer mit den häufigsten Busfehlern. Die zehn Teilnehmer sind nach der Anzahl der Busfehler (*Error_Count*) absteigend sortiert.

Top_Ten_CRC_CIC: Dieser 32-Bit-Zähler wird jedesmal um den Wert „+1“ erhöht, wenn sich die Informationen des Diagnose-Blocks *Top_Ten_CRC* ändern. Der Diagnose-Block *Top_Ten_CRC* besteht aus 10 Diagnose-Informationen von jeweils 4 Worten Länge:

Device_No_Bus_Fault: INTERBUS-Teilnehmernummer (*Device_No*)
Error_Count: Fehleranzahl dieses Fehlers (32-Bit-Zähler)
Error_Code: Informationen zur Fehlerart

Syntax

Diagnose-Block *Last_Ten_PF*

Wort 1 + 2	Last_Ten_PF_CIC (Bit 31 ... 16)	Last_Ten_PF
	Last_Ten_PF_CIC (Bit 15 ... 0)	
Wort 3	1. Device_No_PF	
Wort 4	Error_Code	
...	...	
Wort 21	10. Device_No_PF	
Wort 22	Error_Code	

Der Diagnose-Block *Last_Ten_PF* enthält die Liste der letzten zehn INTERBUS-Teilnehmer, auf denen ein Peripheriefehler (PF = Peripheral Fault) vorlag. Der Teilnehmer, auf dem zuletzt ein Peripheriefehler vorlag, wird als erster angeführt.

Last_Ten_PF_CIC: Dieser 32-Bit-Zähler wird jedesmal um den Wert „+1“ erhöht, wenn sich die Informationen des Diagnose-Blocks *Last_Ten_PF* ändern. Der Diagnose-Block *Last_Ten_PF* besteht aus 10 Diagnose-Informationen von jeweils 2 Worten Länge:

Device_No_PF: INTERBUS-Teilnehmernummer (*Device_No*)
Error_Code: Informationen zur Fehlerart

Syntax

Diagnose-Block *IPMS_Error*

Wort 1 + 2	IPMS_Error_CIC (Bit 31 ... 16)	IPMS_Error
	IPMS_Error_CIC (Bit 15 ... 0)	
Wort 3	Type	

IPMS_Error_CIC: Dieser 32-Bit-Zähler wird jedesmal um den Wert „+1“ erhöht, wenn sich Informationen des Diagnose-Blocks *IPMS_Error* ändern.

Type: Inhalt des IPMS-Error-Registers (siehe Handbuch zum Master-Protokollchip IBS IPMS3 UM)

Syntax

Diagnose-Block *Bus_Error*

Wort 1 + 2	Bus_Error_CIC	(Bit 31 ... 16)	Bus_Error
	Bus_Error_CIC	(Bit 15 ... 0)	
Wort 3	Type		
Wort 4	Device_No		

Bus_Error_CIC: Dieser 32-Bit-Zähler wird jedesmal um den Wert „+1“ erhöht, wenn sich Informationen des Diagnose-Blocks *Bus_Error* ändern.

Type: Informationen zur Fehlerart

Device_No: Nummer des INTERBUS-Teilnehmers

Syntax

Diagnose-Block *PF*

Wort 1 + 2	PF_CIC	(Bit 31 ... 16)	PF
	PF_CIC	(Bit 15 ... 0)	
Wort 3	Type		
Wort 4	Device_No		

PF_CIC: Dieser 32-Bit-Zähler wird jedesmal um den Wert „+1“ erhöht, wenn sich Informationen des Diagnose-Blocks *PF* ändern.

Type: Informationen zur Fehlerart

Device_No: Nummer des INTERBUS-Teilnehmers

Syntax

Diagnose-Block *Time_Out_Error*

Wort 1 + 2	Time_Out_Error_CIC	(Bit 31 ... 16)	Time_Out_Error
	Time_Out_Error_CIC	(Bit 15 ... 0)	
Wort 3	Type		
Wort 4	Device_No		

Time_Out_Error_CIC: Dieser 32-Bit-Zähler wird jedesmal um den Wert „+1“ erhöht, wenn sich Informationen des Diagnose-Blocks *Time_Out_Error* ändern.

Type: Informationen zur Fehlerart

Device_No: Nummer des INTERBUS-Teilnehmers

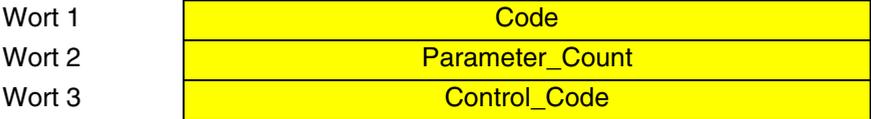


Die Worte 3 und 4 des Diagnose-Blocks *Time_Out_Error* werden insgesamt zehnmal, d. h. für die letzten 10 Teilnehmer übertragen

2.7.6 Dienst „Control_Statistics“

Aufgabe: Mit diesem Dienst können Sie die statistische Diagnose steuern.

Syntax: **Control_Statistics_Request** **030F_{hex}**



- Legende:**
- Code: 030F_{hex} Kommando-Code des Requests
 - Parameter_Count: Anzahl der nachfolgenden Worte
 - 0001_{hex} 1 Parameterwort
 - Control_Code: Funktionalität des Dienstes
 - 0000_{hex} Reset
 - Rücksetzen der statistischen Diagnose-Informationen (vgl. Dienst „Get_Diag_Info“ (032B_{hex}) ab Seite 2-124).

Syntax:

Control_Statistics_Confirmation

830F_{hex}

Positive Meldung

Wort 1	Code
Wort 2	Parameter_Count
Wort 3	Result

Negative Meldung

Wort 1	Code
Wort 2	Parameter_Count
Wort 3	Result
Wort 4	Add_Error_Info

Bit | 15 0 |

Legende:

- Code: 830F_{hex} Meldungs-Code der Confirmation
- Parameter_Count: Anzahl der nachfolgenden Worte
 bei positiver Meldung:
 0001_{hex} 1 Parameterwort
 bei negativer Meldung:
 0002_{hex} 2 Parameterworte
- Result: Ergebnis der Dienstbearbeitung
 0000_{hex} zeigt eine positive Meldung an.
 Die Anschaltbaugruppe hat den Dienst erfolgreich bearbeitet.
 xxxx_{hex} zeigt eine negative Meldung an.
 Die Anschaltbaugruppe konnte den aufgerufenen Dienst nicht bearbeiten.
 Der Parameter *Result* enthält die Ursache, warum der Dienst nicht bearbeitet werden konnte.
- Add_Error_Info: Zusätzliche Informationen zur Fehlerursache

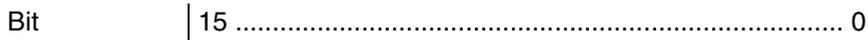
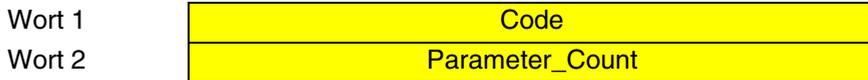
2.8 Dienste zum Anlegen von Funktionen

Sie können für die Anschaltbaugruppe Funktionen definieren. Eine Funktion besteht aus einer frei zusammengestellten Dienstfolge (*Action_Object*) und den zugehörigen Parametersätzen (*Signal_Object*). Diese Parametersätze können fest vorgegeben sein oder erst zur Laufzeit der Funktion hinzugeladen werden. So können Sie die Funktion z. B. in Abhängigkeit Ihres Anlagenzustandes variieren.

2.8.1 Dienst „Initiate_Load_Action_Object“

Aufgabe: Der Dienst leitet das Zusammenstellen von Dienstfolgen ein. Er bereitet die Anschaltbaugruppe auf den Dienst „Load_Action_Object“ (0141_{hex}) vor.

Syntax: **Initiate_Load_Action_Object_Request** **0140_{hex}**



Legende:

Code:	0140 _{hex}	Kommando-Code des Requests
Parameter_Count:		Anzahl der nachfolgenden Worte
	0000 _{hex}	kein Parameterwort

Syntax: **Initiate_Load_Action_Object_Confirmation** **8140_{hex}**

Positive Meldung

Wort 1	Code
Wort 2	Parameter_Count
Wort 3	Result

Negative Meldung

Wort 1	Code
Wort 2	Parameter_Count
Wort 3	Result
Wort 4	Add_Error_Info

Bit | 15 0 |

Legende:

Code:	8140 _{hex}	Meldungs-Code der Confirmation
Parameter_Count:		Anzahl der nachfolgenden Worte bei positiver Meldung: 0001 _{hex} 1 Parameterwort bei negativer Meldung: 0002 _{hex} 2 Parameterworte
Result:		Ergebnis der Dienstbearbeitung 0000 _{hex} zeigt eine positive Meldung an. Die Anschaltbaugruppe hat den Dienst erfolgreich bearbeitet. xxxx _{hex} zeigt eine negative Meldung an. Die Anschaltbaugruppe konnte den aufgerufenen Dienst nicht bearbeiten. Der Parameter <i>Result</i> enthält die Ursache, warum der Dienst nicht bearbeitet werden konnte.
Add_Error_Info:		Zusätzliche Informationen zur Fehlerursache

2.8.2 Dienst „Load_Action_Object“

Aufgabe:

Mit diesem Dienst stellen Sie eine Dienstfolge (*Action_Object*) zusammen. Eine Dienstfolge wird unter einem Index (*Action_Index*) abgespeichert. Eine Dienstfolge besteht aus einzelnen Diensten (*Action*), die durch ihren Kommando-Code angegeben sind. Anstelle eines Kommando-Codes können Sie auch einen Verweis (*Action_Index*) auf eine andere Dienstfolge angeben.



Der Index der Dienstfolge (*Action_Index*) muss eindeutig sein, d. h. er darf nur einmal vergeben werden.

Voraussetzung:

Die Anschaltbaugruppe muss zuvor mit dem Dienst „Initiate_Load_Action_Object“ (0140_{hex}) auf das Zusammenstellen von Dienstfolgen vorbereitet worden sein.

Syntax:

Load_Action_Object_Request

0141_{hex}

Wort 1	Code	
Wort 2	Parameter_Count	
Wort 3	Action_Index	
Wort 4	Action_Count	
...	1. Action	Dienstfolge (<i>Action_Object</i>)
	2. Action	
	...	

Bit | 15 0 |

Legende:

- Code: 0141_{hex} Kommando-Code des Requests
- Parameter_Count: Anzahl der nachfolgenden Worte
 $xxxx_{hex} = 2 + Action_Count$
- Action_Index: Eindeutiger Index der Dienstfolge
 Jeder *Action_Index* darf nur einmal vergeben werden! Die Action-Indices 0000_{hex} ... 0063_{hex} sind für firmware-interne Dienstfolgen reserviert!
- Action_Count: Anzahl der Dienste (*Action*) dieser Dienstfolge

Action: Kommando-Code oder Verweis (*Action_Index*) auf eine Dienstfolge an.
 Setzen Sie das Bit 15 auf 1, wenn Sie einen Verweis auf eine Dienstfolge angeben.
 Beispielsweise müssen Sie für den *Action_Index* 0567_{hex} den Wert 8567_{hex} eintragen.

Syntax: **Load_Action_Object_Confirmation** **8141_{hex}**

Positive Meldung

Wort 1	Code
Wort 2	Parameter_Count
Wort 3	Result

Negative Meldung

Wort 1	Code
Wort 2	Parameter_Count
Wort 3	Result
Wort 4	Add_Error_Info

Bit | 15 0 |

Legende:

Code: 8141_{hex} Meldungs-Code der Confirmation

Parameter_Count: Anzahl der nachfolgenden Worte
 bei positiver Meldung:
 0001_{hex} 1 Parameterwort
 bei negativer Meldung:
 0002_{hex} 2 Parameterworte

Result: Ergebnis der Dienstbearbeitung
 0000_{hex} zeigt eine positive Meldung an.
 Die Anschaltbaugruppe hat den Dienst erfolgreich bearbeitet.
 xxxx_{hex} zeigt eine negative Meldung an.
 Die Anschaltbaugruppe konnte den aufgerufenen Dienst nicht bearbeiten.
 Der Parameter *Result* enthält die Ursache, warum der Dienst nicht bearbeitet werden konnte.

Add_Error_Info: Zusätzliche Informationen zur Fehlerursache

2.8.3 Dienst „Terminate_Load_Action_Object“

Aufgabe: Dieser Dienst schließt das Zusammenstellen von Dienstfolgen ab.

Syntax: **Terminate_Load_Action_Object_Request** **0142_{hex}**

Wort 1	Code
Wort 2	Parameter_Count

Bit | 15 0 |

Legende:
Code: 0142_{hex} Kommando-Code des Requests
Parameter_Count: Anzahl der nachfolgenden Worte
0000_{hex} kein Parameterwort

Syntax: **Terminate_Load_Action_Object_Confirmation** **8142_{hex}**

Positive Meldung

Wort 1	Code
Wort 2	Parameter_Count
Wort 3	Result

Negative Meldung

Wort 1	Code
Wort 2	Parameter_Count
Wort 3	Result
Wort 4	Add_Error_Info

Bit | 15 0 |

Legende:

Code:	8142 _{hex}	Meldungs-Code der Confirmation
Parameter_Count:		Anzahl der nachfolgenden Worte bei positiver Meldung: 0001 _{hex} 1 Parameterwort bei negativer Meldung: 0002 _{hex} 2 Parameterworte
Result:		Ergebnis der Dienstbearbeitung 0000 _{hex} zeigt eine positive Meldung an. Die Anschaltbaugruppe hat den Dienst erfolgreich bearbeitet. xxxx _{hex} zeigt eine negative Meldung an. Die Anschaltbaugruppe konnte den aufgerufenen Dienst nicht bearbeiten. Der Parameter <i>Result</i> enthält die Ursache, warum der Dienst nicht bearbeitet werden konnte.
Add_Error_Info:		Zusätzliche Informationen zur Fehlerursache

2.8.4 Dienst „Read_Action_Object“

Aufgabe: Dieser Dienst liest die Kommando-Codes einer Dienstfolge (Action Object) aus.

Syntax: **Read_Action_Object_Request** **0143_{hex}**

Wort 1	Code
Wort 2	Parameter_Count
Wort 3	Action_Index

Bit | 15 0 |

Legende:

Code:	0143 _{hex}	Kommando-Code des Requests
Parameter_Count:		Anzahl der nachfolgenden Worte
	0001 _{hex}	1 Parameterwort
Action_Index:		Index der Dienstfolge, deren Kommando-Codes Sie auslesen möchten.

Syntax: **Read_Action_Object_Confirmation** **8143_{hex}**

Positive Meldung

Wort 1	Code	Dienstfolge
Wort 2	Parameter_Count	
Wort 3	Result	
Wort 4	Action_Index	
Wort 5	Action_Count	
Wort 6	1. Action	
...	2. Action	

Negative Meldung

Wort 1	Code
Wort 2	Parameter_Count
Wort 3	Result
Wort 4	Add_Error_Info

Bit | 15 0 |

Legende:

Code:	8143 _{hex} Meldungs-Code der Confirmation
Parameter_Count:	Anzahl der nachfolgenden Worte bei positiver Meldung: $xxxx_{hex} = 3 + Action_Count$ bei negativer Meldung: 0002 _{hex} 2 Parameterworte
Result:	Ergebnis der Dienstbearbeitung 0000 _{hex} zeigt eine positive Meldung an. Die Anschaltbaugruppe hat den Dienst erfolgreich bearbeitet. Die Meldung enthält die angeforderten Kommando-Codes xxxx _{hex} zeigt eine negative Meldung an. Die Anschaltbaugruppe konnte den aufgerufenen Dienst nicht bearbeiten. Der Parameter <i>Result</i> enthält die Ursache, warum der Dienst nicht bearbeitet werden konnte.

Add_Error_Info:	Zusätzliche Informationen zur Fehlerursache
Action_Index:	Index der Dienstfolge (<i>Action_Object</i>), deren Kommando-Codes ausgelesen wurden.
Action_Count:	Anzahl der ausgelesenen Kommando-Codes oder Verweise, d. h. die Länge der Dienstfolge
Action:	Kommando-Code oder Verweis (<i>Action_Index</i>) auf eine Dienstfolge Ist das Bit 15 auf 1 gesetzt, handelt es sich um einen Verweis auf eine Dienstfolge. Beispielsweise bedeutet der Wert 8567_{hex} , dass es sich um einen Verweis auf der Dienstfolge mit dem <i>Action_Index</i> 0567_{hex} handelt.

2.8.5 Dienst „Delete_Action_Object“

Aufgabe: Dieser Dienst löscht eine Dienstfolge (*Action_Object*).

Syntax: **Delete_Action_Object_Request** **0144_{hex}**

Wort 1	Code
Wort 2	Parameter_Count
Wort 3	Action_Index

Bit | 15 0 |

Legende:

Code:	0144 _{hex} Kommando-Code des Requests
Parameter_Count:	Anzahl der nachfolgenden Worte
	0001 _{hex} 1 Parameterwort
Action_Index:	Index der zu löschenden Dienstfolge
	FFFF _{hex} löscht alle projektierten Dienstfolgen.
	xxxx _{hex} löscht die Dienstfolge mit diesem Index.

Syntax: **Delete_Action_Object_Confirmation** **8144_{hex}**

Positive Meldung

Wort 1	Code
Wort 2	Parameter_Count
Wort 3	Result

Negative Meldung

Wort 1	Code
Wort 2	Parameter_Count
Wort 3	Result
Wort 4	Add_Error_Info

Bit | 15 0 |

Legende:

Code: 8144_{hex} Meldungs-Code der Confirmation

Parameter_Count: Anzahl der nachfolgenden Worte
bei positiver Meldung:
0001_{hex} 1 Parameterwort
bei negativer Meldung:
0002_{hex} 2 Parameterworte

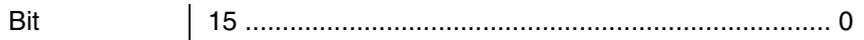
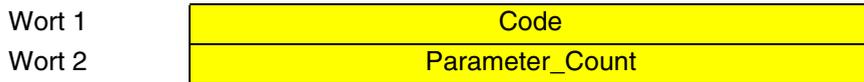
Result: Ergebnis der Dienstbearbeitung
0000_{hex} zeigt eine positive Meldung an.
Die Anschaltbaugruppe hat den Dienst erfolgreich bearbeitet.
xxxx_{hex} zeigt eine negative Meldung an.
Die Anschaltbaugruppe konnte den aufgerufenen Dienst nicht bearbeiten.
Der Parameter *Result* enthält die Ursache, warum der Dienst nicht bearbeitet werden konnte.

Add_Error_Info: Zusätzliche Informationen zur Fehlerursache

2.8.6 Dienst „Initiate_Load_Signal_Object“

Aufgabe: Dieser Dienst leitet die Projektierung von Parametersätzen (*Signal Objects*) ein. Er bereitet die Anschaltbaugruppe auf den Dienst „Load_Signal_Object“ (0146_{hex}) vor.

Syntax: **Initiate_Load_Signal_Object_Request** 0145_{hex}



Legende:

Code:	0145 _{hex}	Kommando-Code des Requests
Parameter_Count:		Anzahl der nachfolgenden Worte
	0000 _{hex}	kein Parameterwort

Syntax: **Initiate_Load_Signal_Object_Confirmation** 8145_{hex}

Positive Meldung

Wort 1	Code
Wort 2	Parameter_Count
Wort 3	Result

Negative Meldung

Wort 1	Code
Wort 2	Parameter_Count
Wort 3	Result
Wort 4	Add_Error_Info

Bit | 15 0 |

Legende:

Code: 8145_{hex} Meldungs-Code der Confirmation

Parameter_Count: Anzahl der nachfolgenden Worte
bei positiver Meldung:
0001_{hex} 1 Parameterwort
bei negativer Meldung:
0002_{hex} 2 Parameterworte

Result: Ergebnis der Dienstbearbeitung
0000_{hex} zeigt eine positive Meldung an.
Die Anschaltbaugruppe hat den Dienst erfolgreich bearbeitet.
xxxx_{hex} zeigt eine negative Meldung an.
Die Anschaltbaugruppe konnte den aufgerufenen Dienst nicht bearbeiten.
Der Parameter *Result* enthält die Ursache, warum der Dienst nicht bearbeitet werden konnte.

Add_Error_Info: Zusätzliche Informationen zur Fehlerursache

2.8.7 Dienst „Load_Signal_Object“

Aufgabe: Mit diesem Dienst können Sie Parametersätze (*Signal Objects*) für Dienstfolgen projektieren.

Voraussetzung: Die Anschaltbaugruppe muss zuvor mit dem Dienst „Initiate_Load_Signal_Object“ (0145_{hex}) auf die Projektierung vorbereitet worden sein.

Syntax: **Load_Signal_Object_Request** **0146hex**

Wort 1	Code			
Wort 2	Parameter_Count			
Wort 3	Signal_Index			
Wort 4	Action_Index			
Wort 5	Signal_Type			
Wort 6	Bit_Number			
Wort 7	Start_Register_Address			
Wort 8	State_Register_Address			
Wort 9	Result_Register_Address			
Wort 10	Write_PB_Count (n _{Write})			Anzahl
Wort 11	W_Mode	res.	Write_Parameter_Count	Write-Parameterblock (Action 1)
Wort 12	Write_Parameter			
...	...			
...
	W_Mode	res.	Write_Parameter_Count	Write-Parameterblock (Action x)
	Write_Parameter			
	...			
	Read_PB_Count (n _{Read})			Anzahl
	R_Mode	res.	Read_Parameter_Count	Read-Parameterblock (Action 1)
	Read_Parameter			
	...			
	...			
	R_Mode	res.	x. Read_Parameter_Count	Read-Parameterblock (Action x)
	Read_Parameter			
	...			
Bit	15 / 14	13 / 12	11..... 0	

Legende:	Code:	0146 _{hex} Kommando-Code des Requests
	Parameter_Count:	Anzahl der nachfolgenden Worte
		xxxx _{hex} Der Wert ist abhängig von der Anzahl und Größe der Write- und Read-Parameterblöcke.
	Signal_Index:	Index des Parametersatzes Der Index muss eindeutig sein, er darf nur einmal vergeben werden! Wertebereiche: 0000 _{hex} ... 0063 _{hex} reserviert für firmware-interne Parametersätze! 0064 _{hex} ... 1000 _{hex} Parametersätze mit diesen Indices werden beim Hochlaufen der Anschaltbaugruppe aktiviert! 1001 _{hex} ... FFFF _{hex} Parametersätze mit diesen Indices werden durch das Anwendungsprogramm aktiviert!
	Action_Index:	Index der Dienstfolge, der die Parameter dieses Parametersatzes zur Verfügung stehen sollen.
	Signal_Type:	Kennung, für die Verhaltensweise von Dienstfolgen und Parametersätzen Der Parameter beschreibt ein 16 Bit breites Feld, wobei jedes Bit einer Verhaltensweise entspricht. Setzen Sie bei den Verhaltensweisen, die Sie einstellen wollen, die entsprechenden Bits auf 1. Zuordnungen: Bit 0 - 6 reserviert (immer 0 _{bin} !) Bit 7 Resident-Flag Legt fest, ob Dienstfolge und Parametersatz bei Aufruf des Dienstes „Program_Resident_Actions“ (0158 _{hex}) auf dem Parametrierungsspeicher gespeichert werden sollen. 1 _{bin} Dienstfolge und Parametersatz werden gespeichert. 0 _{bin} Dienstfolge und Parametersatz werden nicht gespeichert. Bit 8 - 15 reserviert (immer 0 _{bin} !)

Bit_Number:	Lage des Signalbits innerhalb der Byte-Adresse Wertebereich: Bit 0 ... 7						
Start_Register_Address:	Lage des Start-Bits (Byte-Adresse im MPM)						
State_Register_Address:	Lage des Status-Bits (Byte-Adresse im MPM)						
Result_Register_Address:	Lage des Ergebnis-Bits (Byte-Adresse im MPM)						
Write_PB_Count:	Anzahl der folgenden Write-Parameterblöcke. Diese Anzahl muss mit der Anzahl der Dienste der referenzierten Dienstfolge übereinstimmen.						
W_Mode:	Struktur des folgenden Write-Parameterblocks und die Einheit (Worte oder Bytes) des Parameters <i>Write_Parameter_Count</i> . <table><tr><td>00_{bin}</td><td>Compact-Data-Parameter-Mode <i>Write_Parameter_Count</i> bestimmt die Anzahl der folgenden festen Parameter in Worten. Der <i>k</i>-te <i>Write_Parameter_Count</i> stimmt mit dem <i>Parameter_Count</i> des <i>k</i>-ten Dienstes der referenzierten Dienstfolge überein.</td></tr><tr><td>10_{bin}</td><td>Compact-Address-Parameter-Mode <i>Write_Parameter_Count</i> bestimmt die Anzahl der folgenden Parameter in Byte, damit die Firmware sie von der im nächsten Wort stehenden MPM-Adresse an lesen kann.</td></tr><tr><td>11_{bin}</td><td>Split-Parameter-Mode Im Split-Parameter-Mode sind MPM-Adressen und feste Write-Parameter innerhalb des Write-Parameterblocks zugelassen. <i>Write_Parameter_Count</i> gibt die Summe aus der Anzahl der folgenden MPM-Adressen und der Write-Parameter (Byte) an. Feste Write-Parameter werden durch den Wert FFFF_{hex} gekennzeichnet und an das Ende des MPM-Address-Blocks angehängt.</td></tr></table>	00 _{bin}	Compact-Data-Parameter-Mode <i>Write_Parameter_Count</i> bestimmt die Anzahl der folgenden festen Parameter in Worten. Der <i>k</i> -te <i>Write_Parameter_Count</i> stimmt mit dem <i>Parameter_Count</i> des <i>k</i> -ten Dienstes der referenzierten Dienstfolge überein.	10 _{bin}	Compact-Address-Parameter-Mode <i>Write_Parameter_Count</i> bestimmt die Anzahl der folgenden Parameter in Byte, damit die Firmware sie von der im nächsten Wort stehenden MPM-Adresse an lesen kann.	11 _{bin}	Split-Parameter-Mode Im Split-Parameter-Mode sind MPM-Adressen und feste Write-Parameter innerhalb des Write-Parameterblocks zugelassen. <i>Write_Parameter_Count</i> gibt die Summe aus der Anzahl der folgenden MPM-Adressen und der Write-Parameter (Byte) an. Feste Write-Parameter werden durch den Wert FFFF _{hex} gekennzeichnet und an das Ende des MPM-Address-Blocks angehängt.
00 _{bin}	Compact-Data-Parameter-Mode <i>Write_Parameter_Count</i> bestimmt die Anzahl der folgenden festen Parameter in Worten. Der <i>k</i> -te <i>Write_Parameter_Count</i> stimmt mit dem <i>Parameter_Count</i> des <i>k</i> -ten Dienstes der referenzierten Dienstfolge überein.						
10 _{bin}	Compact-Address-Parameter-Mode <i>Write_Parameter_Count</i> bestimmt die Anzahl der folgenden Parameter in Byte, damit die Firmware sie von der im nächsten Wort stehenden MPM-Adresse an lesen kann.						
11 _{bin}	Split-Parameter-Mode Im Split-Parameter-Mode sind MPM-Adressen und feste Write-Parameter innerhalb des Write-Parameterblocks zugelassen. <i>Write_Parameter_Count</i> gibt die Summe aus der Anzahl der folgenden MPM-Adressen und der Write-Parameter (Byte) an. Feste Write-Parameter werden durch den Wert FFFF _{hex} gekennzeichnet und an das Ende des MPM-Address-Blocks angehängt.						
res.	reserviert (immer 00 _{bin} !)						
Write_Parameter_Count:	Anzahl der folgenden Write-Parameter, je nach <i>W_Mode</i> in Wort oder Byte angegeben						

- Write_Parameter: Zu schreibende Daten
(siehe nachfolgende Beispiele)
- Read_PB_Count: Anzahl der folgenden Read-Parameterblöcke
Diese Anzahl muss mit der Anzahl der Dienste des referenzierten Dienstfolge übereinstimmen.
- R_Mode: Struktur des Read-Parameterblocks und die Einheit (Worte/Bytes) des Parameters
Read_Parameter_Count; vgl. *W_Mode*:
 - 10_{bin} Compact-Address-Parameter-Mode
 - 11_{bin} Split-Parameter-Mode
- Read_Parameter_Count: Anzahl der folgenden Read-Parameter, je nach *R_Mode* in Wort oder Byte angegeben
- Read_Parameter: Zu lesende Daten

Beispiel

Write-Parameterblock im Compact-Data-Parameter-Mode:

Mode	res.	Write_Parameter_Count
1. Fixed Parameter		
2. Fixed Parameter		
3. Fixed Parameter		



- Legende:
- Mode: 00_{bin} Compact-Data-Parameter-Mode
 - res.: 00_{bin} reserviert für spätere Erweiterungen
 - Write_Parameter_Count: 003_{hex} Anzahl der nachfolgenden **Worte** (Wertebereich: 001_{hex} ... 1F4_{hex}).
 - Write_Parameter: Parametersatz

Beispiel

Write-Parameterblocks im Compact-Address-Parameter-Mode:

Mode	res.	Write_Parameter_Count
MPM-Address		



- Legende:
- Mode: 10_{bin} Compact-Address-Parameter-Mode
 - res.: 00_{bin} reserviert

Write_Parameter_Count: Anzahl der **Byte** im durch den folgenden Parameter bezeichneten Buffer; z. B. 006_{hex} für einen Buffer von 6 Byte-Länge (Wertebereich: 001_{hex} ... 3E8_{hex}).

MPM-Address: MPM-Startadresse des Byte-Buffers

Beispiel

Write-Parameterblocks im Split-Parameter-Mode:

Mode	res.	Write_Parameter_Count	
		1. MPM-Address	
		2. MPM-Address	
		1. Dummy_Parameter	1. Fixed Byte
		3. MPM-Address	
		2. Dummy_Parameter	2. Fixed Byte
		3. Dummy_Parameter	3. Fixed Byte
	00 _{hex}		1. Fixed Byte
	00 _{hex}		2. Fixed Byte
	00 _{hex}		3. Fixed Byte



Legende: Mode: 11_{bin} Split-Parameter-Mode
 res.: 00_{bin} reserviert für spätere Erweiterungen
 Write_Parameter_Count: Summe aus Anzahl der festen **Byte** und Anzahl der variablen **Byte** im durch den folgenden Parameter bezeichneten Buffer (Wertebereich: 001_{hex} ... 3E8_{hex})
 MPM-Address: MPM-Adressen der variablen Byte
 Dummy_Parameter: FFFF_{hex} Platzhalter für die festen Byte, die am Ende des MPM-Adressen-Blocks folgen.

Syntax: **Load_Signal_Object_Confirmation** **8146_{hex}**

Positive Meldung

Wort 1	Code
Wort 2	Parameter_Count
Wort 3	Result

Negative Meldung

Wort 1	Code
Wort 2	Parameter_Count
Wort 3	Result
Wort 4	Add_Error_Info

Bit | 15 0 |

Legende:

Code: 8146_{hex} Meldungs-Code der Confirmation

Parameter_Count: Anzahl der nachfolgenden Worte
bei positiver Meldung:
0001_{hex} 1 Parameterwort
bei negativer Meldung:
0002_{hex} 2 Parameterworte

Result: Ergebnis der Dienstbearbeitung
0000_{hex} zeigt eine positive Meldung an.
Die Anschaltbaugruppe hat den Dienst erfolgreich bearbeitet.
xxxx_{hex} zeigt eine negative Meldung an.
Die Anschaltbaugruppe konnte den aufgerufenen Dienst nicht bearbeiten.
Der Parameter *Result* enthält die Ursache, warum der Dienst nicht bearbeitet werden konnte.

Add_Error_Info: Zusätzliche Informationen zur Fehlerursache

2.8.8 Dienst „Terminate_Load_Signal_Object“

Aufgabe: Dieser Dienst schließt die Projektierung von Parametersätzen (*Signal Objects*) ab.

Syntax: **Terminate_Load_Signal_Object_Request** **0147_{hex}**



Legende:

Code:	0147 _{hex}	Kommando-Code des Requests
Parameter_Count:		Anzahl der nachfolgenden Worte
	0000 _{hex}	kein Parameterwort

Syntax: **Terminate_Load_Signal_Object_Confirmation** **8147_{hex}**

Positive Meldung

Wort 1	Code
Wort 2	Parameter_Count
Wort 3	Result

Negative Meldung

Wort 1	Code
Wort 2	Parameter_Count
Wort 3	Result
Wort 4	Add_Error_Info

Bit | 15 0 |

Legende:

Code: 8147_{hex} Meldungs-Code der Confirmation

Parameter_Count: Anzahl der nachfolgenden Worte
bei positiver Meldung:
0001_{hex} 1 Parameterwort
bei negativer Meldung:
0002_{hex} 2 Parameterworte

Result: Ergebnis der Dienstbearbeitung
0000_{hex} zeigt eine positive Meldung an.
Die Anschaltbaugruppe hat den Dienst erfolgreich bearbeitet.
xxxx_{hex} zeigt eine negative Meldung an.
Die Anschaltbaugruppe konnte den aufgerufenen Dienst nicht bearbeiten.
Der Parameter *Result* enthält die Ursache, warum der Dienst nicht bearbeitet werden konnte.

Add_Error_Info: Zusätzliche Informationen zur Fehlerursache

2.8.9 Dienst „Read_Signal_Object“

Aufgabe: Der Dienst „Read_Signal_Object“ liest den Parametersatz (*Signal Object*) einer Dienstfolge aus.

Syntax: **Read_Signal_Object_Request** **0148_{hex}**

Wort 1	Code
Wort 2	Parameter_Count
Wort 3	Signal_Index

Bit | 15 0 |

Legende:

Code: 0148_{hex} Kommando-Code des Requests

Parameter_Count: Anzahl der nachfolgenden Worte
0001_{hex} 1 Parameterwort

Signal_Index: Index des Parametersatzes (*Signal Object*), den Sie auslesen möchten.

Syntax: **Read_Signal_Object_Confirmation** **8148_{hex}**

Positive Meldung

Wort 1	Code		
Wort 2	Parameter_Count		
Wort 3	Result		
Wort 4	Signal_Index		
Wort 5	Action_Index		
Wort 6	Signal_Type		
Wort 7	Bit_Number		
Wort 8	Start_Register_Address		
Wort 9	State_Register_Address		
Wort 10	Result_Register_Address		
Wort 11	Write_PB_Count (n _{Write})		Anzahl
...	W_Mode	res.	Write_Parameter_Count
	Write_Parameter		Write-Parameterblock (Action 1)
	...		

W_Mode	res.	Write_Parameter_Count	Write-Parameterblock (Action x)
Write_Parameter			
...			
Read_PB_Count (n _{Read})			Anzahl
R_Mode	res.	Read_Parameter_Count	Read-Parameterblock (Action 1)
Read_Parameter			
...			
...			
R_Mode	res.	x. Read_Parameter_Count	Read-Parameterblock (Action x)
Read_Parameter			
...			

Negative Meldung

Wort 1
Wort 2
Wort 3
Wort 4

Code
Parameter_Count
Result
Add_Error_Info

Bit | 15 / 14 | 13 / 12 | 11..... 0 |

Legende:

Code: 8148_{hex} Meldungs-Code der Confirmation

Parameter_Count: Anzahl der nachfolgenden Worte bei positiver Meldung
 xxxx_{hex} Der Wert ist abhängig von der Anzahl und Größe der Write- und Read-Parameterblöcke.
 bei negativer Meldung:
 0002_{hex} 2 Parameterworte

Result: Ergebnis der Dienstbearbeitung
 0000_{hex} zeigt eine positive Meldung an. Die Anschaltbaugruppe hat den Dienst erfolgreich bearbeitet.
 xxxx_{hex} zeigt eine negative Meldung an. Die Anschaltbaugruppe konnte den aufgerufenen Dienst nicht bearbeiten. Der Parameter *Result* enthält die Ursache, warum der Dienst nicht bearbeitet werden konnte.

Add_Error_Info:	Zusätzliche Informationen zur Fehlerursache
Signal_Index:	Index des ausgelesenen Parametersatzes
Action_Index:	Index der Dienstfolge, der die Parameter dieses Parametersatzes zur Verfügung stehen
Signal_Type:	Interrupt-Verhalten Der Parameter beschreibt ein 16 Bit breites Feld, wobei jedes Bit einem Interrupt-Verhalten entspricht. Setzen Sie bei den Interrupt-Verhalten, die Sie freigeben wollen, die entsprechenden Bits auf 1. Zuordnungen: Bit 0 Interrupt bei positiver Flanke des Status-Bits Bit 1 Interrupt bei negativer Flanke des Status-Bits Bit 2 Interrupt bei negativer Flanke des Status-Bits und gesetztem Ergebnis-Bit (= 1 _{bin}) Bit 3 Interrupt bei negativer Flanke des Status-Bits und ungesetztem Ergebnis-Bit (= 0 _{bin}) Bit 4 Interrupt bei positiver Flanke des Start-Bits Bit 5 Interrupt bei negativer Flanke des Start-Bits Bit 6 - 15 reserviert
Bit_Number:	Lage des Signalbits innerhalb der Byte-Adresse Wertebereich: 0 _{bin} ... 7 _{bin} .
Start_Register_Address:	Byte-Adresse im MPM (Offset) für die Lage des Start-Bits
State_Register_Address:	Byte-Adresse im MPM (Offset) für die Lage des Status-Bits
Result_Register_Address:	Byte-Adresse im MPM (Offset) für die Lage des Ergebnis-Bits
Write_PB_Count:	Anzahl der folgenden Write-Parameterblöcke Diese Anzahl muss mit der Anzahl der Dienste der referenzierten Dienstfolge übereinstimmen.
W_Mode:	Struktur des folgenden Write-Parameterblocks Der Parameter <i>W_Mode</i> bestimmt die Einheit (Worte oder Bytes) des Parameters <i>Write_Parameter_Count</i> .

00 _{bin}	<p>Compact-Data-Parameter-Mode</p> <p><i>Write_Parameter_Count</i> bestimmt die Anzahl der folgenden festen Parameter in Worten. Der <i>k</i>-te <i>Write_Parameter_Count</i> stimmt mit dem <i>Parameter_Count</i> des <i>k</i>-ten Dienstes der referenzierten Dienstfolge überein.</p>
10 _{bin}	<p>Compact-Address-Parameter-Mode</p> <p><i>Write_Parameter_Count</i> bestimmt die Anzahl der folgenden Parameter in Byte, damit die Firmware sie von der in nächsten Wort stehenden MPM-Adresse an lesen kann.</p>
11 _{bin}	<p>Split-Parameter-Mode</p> <p>Im Split-Parameter-Mode sind MPM-Adressen und feste Write-Parameter innerhalb des Write-Parameterblocks zugelassen. <i>Write_Parameter_Count</i> gibt die Summe aus der Anzahl der folgenden MPM-Adressen und der Write-Parameter (Byte) an. Feste Write-Parameter werden durch den Wert FFFF_{hex} gekennzeichnet und an das Ende des MPM-Address-Blocks angehängt.</p>
res.:	reserviert (immer 00 _{bin} !)
Write_Parameter_Count:	Anzahl der folgenden Write-Parameter, je nach <i>Mode</i> in Wort oder Byte angegeben
Write_Parameter:	Zu schreibende Daten
Read_PB_Count:	Anzahl der folgenden Read-Parameterblöcke Diese Anzahl muss mit der Anzahl der Dienste der referenzierten Dienstfolge übereinstimmen.
R_Mode:	Struktur des Read-Parameterblocks und Einheit (Worte/Bytes) des Parameters <i>Read_Parameter_Count</i> ; vgl. <i>W_Mode</i> :
	10 _{bin} Compact-Address-Parameter-Mode
	11 _{bin} Split-Parameter-Mode
Read_Parameter_Count:	Anzahl der folgenden Read-Parameter, je nach <i>Mode</i> in Wort oder Byte angegeben
Read_Parameter:	Zu lesende Daten

2.8.10 Dienst „Delete_Signal_Object“

Aufgabe: Dieser Dienst löscht einen Parametersatz (*Signal Object*).

Syntax: **Delete_Signal_Object_Request** **0149_{hex}**

Wort 1	Code
Wort 2	Parameter_Count
Wort 3	Signal_Index

Bit | 15.....0 |

- Legende:**
- Code: 0149_{hex} Kommando-Code des Requests
 - Parameter_Count: Anzahl der nachfolgenden Worte
 - 0001_{hex} 1 Parameterwort
 - Signal_Index: Index des zu löschenden Parametersatzes
 - FFFF_{hex} löscht alle projektierten Parametersätze.
 - xxxx_{hex} löscht den Parametersatz mit diesem Index.

Syntax: **Delete_Signal_Object_Confirmation** 8149_{hex}

Positive Meldung

Wort 1	Code
Wort 2	Parameter_Count
Wort 3	Result

Negative Meldung

Wort 1	Code
Wort 2	Parameter_Count
Wort 3	Result
Wort 4	Add_Error_Info

Bit | 15 0 |

Legende:

Code: 8149_{hex} Meldungs-Code der Confirmation

Parameter_Count: Anzahl der nachfolgenden Worte
bei positiver Meldung:
0001_{hex} 1 Parameterwort
bei negativer Meldung:
0002_{hex} 2 Parameterworte

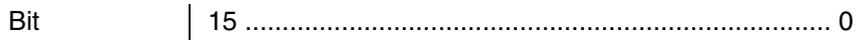
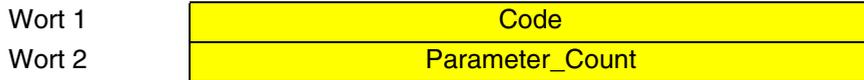
Result: Ergebnis der Dienstbearbeitung
0000_{hex} zeigt eine positive Meldung an.
Die Anschaltbaugruppe hat den Dienst erfolgreich bearbeitet.
xxxx_{hex} zeigt eine negative Meldung an.
Die Anschaltbaugruppe konnte den aufgerufenen Dienst nicht bearbeiten.
Der Parameter *Result* enthält die Ursache, warum der Dienst nicht bearbeitet werden konnte.

Add_Error_Info: Zusätzliche Informationen zur Fehlerursache

2.8.11 Dienst „Initiate_Load_Event_Object“

Aufgabe: Dieser Dienst leitet die Projektierung von Ereignisbeschreibungen (*Event Object*) ein. Er bereitet die Anschaltbaugruppe auf den Dienst „Load_Event_Object“ (014B_{hex}) vor.

Syntax: **Initiate_Load_Event_Object_Request** **014A_{hex}**



Legende:

Code:	014A _{hex}	Kommando-Code des Requests
Parameter_Count:		Anzahl der nachfolgenden Worte
	0000 _{hex}	kein Parameterwort

Syntax: **Initiate_Load_Event_Object_Confirmation** **814A_{hex}**

Positive Meldung

Wort 1	Code
Wort 2	Parameter_Count
Wort 3	Result

Negative Meldung

Wort 1	Code
Wort 2	Parameter_Count
Wort 3	Result
Wort 4	Add_Error_Info

Bit | 15 0 |

Legende:

Code: 814A_{hex} Meldungs-Code der Confirmation

Parameter_Count: Anzahl der nachfolgenden Worte
bei positiver Meldung:
0001_{hex} 1 Parameterwort
bei negativer Meldung:
0002_{hex} 2 Parameterworte

Result: Ergebnis der Dienstbearbeitung
0000_{hex} zeigt eine positive Meldung an.
Die Anschaltbaugruppe hat den Dienst erfolgreich bearbeitet.
xxxx_{hex} zeigt eine negative Meldung an.
Die Anschaltbaugruppe konnte den aufgerufenen Dienst nicht bearbeiten.
Der Parameter *Result* enthält die Ursache, warum der Dienst nicht bearbeitet werden konnte.

Add_Error_Info: Zusätzliche Informationen zur Fehlerursache

2.8.12 Dienst „Load_Event_Object“

Aufgabe: Mit diesem Dienst können Sie Ereignisbeschreibungen (*Event Objects*) projektieren. Ereignisbeschreibungen stellen die Parametersätze für Dienstfolgen zur Verfügung.

Syntax: **Load_Event_Object_Request** **014B_{hex}**

Wort 1	Code
Wort 2	Parameter_Count
Wort 3	Event_Index
Wort 4	Message_Code
Wort 5	Notification_Byte_Address
Wort 6	Acknowledge_Byte_Address
Wort 7	Event_Signal_Bit
Wort 8	Event_Signal_Type
Wort 9	Event_Interface_Selector
Wort 10	Interface_Parameter
...	

Bit | 15 0 |

- Legende:**
- Code: 014B_{hex} Kommando-Code des Requests
 - Parameter_Count: Anzahl der nachfolgenden Worte
 xxxx_{hex} Die Anzahl der Parameterworte ist abhängig von der gewählten Schnittstelle (Parameter *Event_Interface_Selector*) und der damit verbundenen Anzahl an Parametern (Parameterblock *Interface_Parameter*).
 - Event_Index: Index des projizierten Ereignisses
 - Message_Code: Message-Code des eingetretenen Ereignisses (Zugriff über *Event_Code*)
 - Notification_Byte_Address: Byte-Adresse im MPM (Offset) für die Lage des Notification-Bits

Acknowledge_Byte_Address:	Byte-Adresse im MPM (Offset) für die Lage des Acknowledge-Bits
Event_Signal_Bit:	Lage des Signal-Bits innerhalb der Byte-Adresse Wertebereich: 0 ... 7.
Event_Signal_Type:	Verhaltensweise der Ereignisbeschreibungen Der Parameter beschreibt ein 8 Bit breites Feld, wobei jedes Bit einer Verhaltensweise der Ereignisbeschreibungen (<i>Event Objects</i>) entspricht. Setzen Sie bei der Verhaltensweise, die Sie einstellen wollen, die entsprechenden Bits auf 1. Zuordnungen: Bit 0 ... 6 reserviert (immer 0 _{bin} !) Bit 7 Resident-Flag; Dieses Bit gibt an, ob die Ereignisbeschreibung bei Aufruf des Dienstes „Program_Resident_Actions“ (0158 _{hex}) auf dem Parametrierungsspeicher gespeichert wird: 1 _{bin} Ereignisbeschreibung wird gespeichert 0 _{bin} Ereignisbeschreibung wird nicht gespeichert
Event_Interface_Selector:	Schnittstelle, an die die Parameter gesendet werden: <ul style="list-style-type: none"> – das Ereignis wird über die Mailbox gemeldet. – das Ereignis wird über das SGI gemeldet. – das Ereignis soll eine Aktion auslösen. – das Ereignis soll eine Aktion auslösen und gleichzeitig über das XSGI gemeldet werden. – das Ereignis (die Configuration) wird über das SSGI gemeldet.
Interface_Parameter:	Die Struktur des Parameterblocks <i>Interface_Parameter</i> wird vom Parameter <i>Event_Interface_Selector</i> festgelegt:

1. *Event_Interface_Selector* = E_MXI_INTERFACE
Melden des Ereignisses (Event) über das Mailbox-Interface. Es folgen keine weiteren Parameter in dem Parameterblock *Interface_Parameter*.
2. *Event_Interface_Selector* = E_XSGI_INTERFACE
Melden des Ereignisses über das Extended-Signal-Interface. Es folgt hierbei in dem Parameterblock *Interface_Parameter* die Definition einer Liste von *Read_Parameter*-Adressen. Struktur und Bedeutung siehe Dienst „Load_Signal_Object“ (0146_{hex}).

Syntax:

Wort 10	Anzahl <i>Read_Copy_Address</i> -Blöcke
Wort 11	Anzahl <i>Read_Copy_Address</i> (n)
Wort 12	<i>Read_Copy_Address</i> 1
...	<i>Read_Copy_Address</i> 2
	...
	<i>Read_Copy_Address</i> n

Bit | 15 0 |

3. *Event_Interface_Selector* = E_ACTION_INTERFACE
Melden des Ereignisses durch Auslösen einer Aktion im Action-Handler. Die durch den *Action_Index* referenzierte Dienstfolge (*Action_Object*) muss vom Action-Handler zur Verfügung gestellt werden. Es folgt hierbei im Parameterblock *Interface_Parameter* die Definition eines *Action_Index*, mehrerer *Write_Parameter*-Blöcke und mehrerer *Read_Parameter*-Blöcke. Struktur und Bedeutung siehe „Load_Signal_Object“ (0146_{hex}) auf Seite 2-135.

Syntax:

Wort 10	<i>Action_Index</i>
Wort 11	Anzahl <i>Write_Parameter</i> -Blöcke (m)
Wort 12	Anzahl <i>Write_Parameter</i> der 1. Action (n)
Wort 13	<i>Write_Parameter</i> 1
...	<i>Write_Parameter</i> 2
	...
	<i>Write_Parameter</i> n
	...
	...

Anzahl Write_Parameter der m. Action (n)
Write_Parameter 1
Write_Parameter 2
...
Write_Parameter n
Anzahl Read_Parameter-Blöcke (m Action)
Anzahl Read_Parameter der 1. Action (n)
Read_Parameter 1
Read_Parameter 2
...
Read_Parameter n
...
...
Anzahl Read_Parameter der m. Action (n)
Read_Parameter 1
Read_Parameter 2
...
Read_Parameter n

Bit | 15 0 |

- 4. *Event_Interface_Selector* =
 E_ACTION_INTERFACE_AND_E_XSGI_INTERFACE
 Melden des Ereignisses durch Auslösen einer Action im Action-Handler. Zusätzlich werden im Parameterblock *Interface_Parameter* die entsprechenden Read_Parameter des Ereignisses über das XSGI bekanntgegeben (kopiert).

Syntax:

Wort 10
 Wort 11
 Wort 12
 Wort 13
 ...

Action_Index
Anzahl Write_Parameter-Blöcke (m Action)
Anzahl Write_Parameter der 1. Action (n)
Write_Parameter 1
Write_Parameter 2
...
Write_Parameter n

...
...
Anzahl Write_Parameter der m. Action (n)
Write_Parameter 1
Write_Parameter 2
...
Write_Parameter n
Anzahl Read_Parameter-Blöcke (m Action)
Anzahl Read_Parameter der 1. Action (n)
Read_Parameter 1
Read_Parameter 2
...
Read_Parameter n
...
...
Anzahl Read_Parameter-Blöcke der m. Action (n)
Read_Parameter 1
Read_Parameter 2
...
Read_Parameter n
Anzahl Read_Copy-Blöcke
Anzahl Read_Copy_Address (n)
Read_Copy_Address 1
Read_Copy_Address 2
...
Read_Copy_Address n

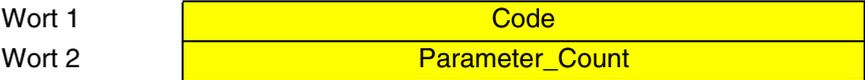
Bit | 15 0 |

- 5. *Event_Interface_Selector* = E_SSGI_INTERFACE (8)
Die Meldung des Ereignisses über das Standard-Signal-Interface wird zur Zeit nicht unterstützt.

2.8.13 Dienst „Terminate_Load_Event_Object“

Aufgabe: Dieser Dienst schließt die Projektierung von Ereignisbeschreibungen (*Event Object*) ab.

Syntax: **Terminate_Load_Event_Object_Request** **014C_{hex}**



Legende:
Code: 014C_{hex} Kommando-Code des Requests
Parameter_Count: Anzahl der nachfolgenden Worte
0000_{hex} kein Parameterwort

Syntax: **Terminate_Load_Event_Object_Confirmation** **814C_{hex}**

Positive Meldung

Wort 1	Code
Wort 2	Parameter_Count
Wort 3	Result

Negative Meldung

Wort 1	Code
Wort 2	Parameter_Count
Wort 3	Result
Wort 4	Add_Error_Info

Bit | 15 0 |

Legende:

Code:	814C _{hex}	Meldungs-Code der Confirmation
Parameter_Count:		Anzahl der nachfolgenden Worte bei positiver Meldung: 0001 _{hex} 1 Parameterwort bei negativer Meldung: 0002 _{hex} 2 Parameterworte
Result:		Ergebnis der Dienstbearbeitung 0000 _{hex} zeigt eine positive Meldung an. Die Anschaltbaugruppe hat den Dienst erfolgreich bearbeitet. xxxx _{hex} zeigt eine negative Meldung an. Die Anschaltbaugruppe konnte den aufgerufenen Dienst nicht bearbeiten. Der Parameter <i>Result</i> enthält die Ursache, warum der Dienst nicht bearbeitet werden konnte.
Add_Error_Info:		Zusätzliche Informationen zur Fehlerursache

2.8.14 Dienst „Read_Event_Object“

Aufgabe: Dieser Dienst liest den Parametersatz einer Ereignisbeschreibung (*Event Object*) aus.

Syntax: **Read_Event_Object_Request** **014D_{hex}**

Wort 1	Code
Wort 2	Parameter_Count
Wort 3	Event_Index

Bit | 15 0 |

Legende:

Code: 014D_{hex} Kommando-Code des Requests

Parameter_Count: Anzahl der nachfolgenden Worte
0001_{hex} 1 Parameterwort

Event_Index: Index des Ereignisses, das gelesen werden soll

Syntax: **Read_Event_Object_Confirmation** **814D_{hex}**

Positive Meldung

Wort 1	Code
Wort 2	Parameter_Count
Wort 3	Event_Index
Wort 4	Message (Event oder Indication)-Code
Wort 5	Notification_Byte_Address
Wort 6	Acknowledge_Byte_Address
Wort 7	Event_Signal_Bit
Wort 8	Event_Signal_Type
Wort 9	Event_Interface_Selector
Wort 10	Interface_Parameter
...	

Negative Meldung

Wort 1	Result
Wort 2	Add_Error_Info

Bit | 15 0 |

Legende:	Code:	814D _{hex}	Meldungs-Code der Confirmation
	Parameter_Count:		Anzahl der nachfolgenden Worte bei positiver Meldung: xxxx _{hex} Die Anzahl der Parameterworte ist abhängig von der gewählten Schnittstelle (Parameter <i>Event_Interface_Selector</i>) und der damit verbundenen Anzahl an Parametern (Parameterblock <i>Interface_Parameter</i>). bei negativer Meldung: 0002 _{hex} 2 Parameterworte
	Result:	0000 _{hex}	Ergebnis der Dienstbearbeitung zeigt eine positive Meldung an. D. h. die Anschaltbaugruppe hat den Dienst erfolgreich bearbeitet.
		xxxx _{hex}	zeigt eine negative Meldung an. D. h. die Anschaltbaugruppe konnte den aufgerufenen Dienst nicht bearbeiten. Der Parameter <i>Result</i> enthält die Ursache, warum der Dienst nicht bearbeitet werden konnte.
	Add_Error_Info:		Zusätzliche Informationen zur Fehlerursache
	Event_Index:		Index des projektierten Ereignisses
	Message_Code:		Message-Code des eingetretenen Ereignisses
	Notification_Byte_Address:		Byte-Adresse im MPM (Offset) für die Lage des Notification-Bits
	Acknowledge_Byte_Address:		Byte-Adresse im MPM (Offset) für die Lage des Acknowledge-Bits
	Event_Signal_Bit:		Lage des Signal-Bits innerhalb der Byte-Adresse
	Event_Signal_Type:		Verhaltensweise der Ereignisbeschreibungen Der Parameter beschreibt ein 8 Bit breites Feld, wobei jedes Bit einer Verhaltensweise der Ereignisbeschreibungen (<i>Event Objects</i>) entspricht. Setzen Sie bei der Verhaltensweise, die Sie einstellen wollen, die entsprechenden Bits auf 1.

Zuordnungen:

Bit 0 ... 6 reserviert (immer 0_{bin}!)

Bit 7 Resident-Flag;

Dieses Bit gibt an, dass die Ereignisbeschreibung bei Aufruf des Dienstes „Program_Resident_Actions“ (0158_{hex}) auf dem Parametrierspeicher gespeichert wird:

1_{bin} Ereignisbeschreibung wird gespeichert

0_{bin} Ereignisbeschreibung wird nicht gespeichert

Event_Interface_Selector:

Schnittstelle, an die die Parameter gesendet werden

Interface_Parameter:

Die Struktur des Parameterblocks *Interface_Parameter* wird vom Parameter *Event_Interface_Selector* festgelegt. (Beschreibung siehe Dienst „Load_Event_Object“ ab Seite 2-162)

2.8.15 Dienst „Delete_Event_Object“

Aufgabe: Dieser Dienst löscht die Parameter einer Ereignisbeschreibung (*Event Object*).

Syntax: **Delete_Event_Object_Request** **014E_{hex}**

Wort 1	Code
Wort 2	Parameter_Count
Wort 3	Event_Index

Bit | 15 0 |

Legende:

Code:	014E _{hex} Kommando-Code des Requests
Parameter_Count:	Anzahl der nachfolgenden Worte
	0001 _{hex} 1 Parameterwort
Event_Index:	Index der zu löschenden Ereignisbeschreibung
	FFFF _{hex} löscht alle projizierten Ereignis- beschreibungen.
	xxxx _{hex} löscht die Ereignisbeschreibung mit diesem Index.

Syntax: **Delete_Event_Object_Confirmation** **814E_{hex}**

Positive Meldung

Wort 1	Code
Wort 2	Parameter_Count
Wort 3	Result

Negative Meldung

Wort 1	Code
Wort 2	Parameter_Count
Wort 3	Result
Wort 4	Add_Error_Info

Bit | 15 0 |

Legende:

Code: 814E_{hex} Meldungs-Code der Confirmation

Parameter_Count: Anzahl der nachfolgenden Worte
bei positiver Meldung:
0001_{hex} 1 Parameterwort
bei negativer Meldung:
0002_{hex} 2 Parameterworte

Result: Ergebnis der Dienstbearbeitung
0000_{hex} zeigt eine positive Meldung an.
Die Anschaltbaugruppe hat den Dienst erfolgreich bearbeitet.
xxxx_{hex} zeigt eine negative Meldung an.
Die Anschaltbaugruppe konnte den aufgerufenen Dienst nicht bearbeiten.
Der Parameter *Result* enthält die Ursache, warum der Dienst nicht bearbeitet werden konnte.

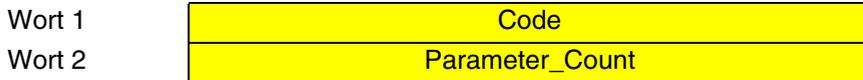
Add_Error_Info: Zusätzliche Informationen zur Fehlerursache

2.9 Dienste für den Parametrierungsspeicher

2.9.1 Dienst „Program_Resident_Actions“

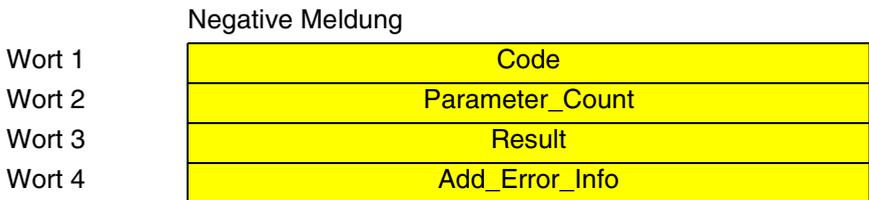
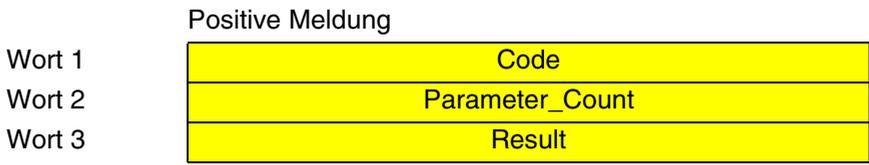
Aufgabe: Dieser Dienst überträgt die im Arbeitsspeicher der Anschaltbaugruppe gespeicherten Dienstfolgen (Action Object) und Parametersätze (*Signal Object*) in den Parametrierungsspeicher. Dienstfolgen und Parametersätze, die beim Aufruf des Dienstes „Load_Signal_Object“ (0146_{hex}) mit dem Bit 7 des Parameters *Signal_Type* als resident gekennzeichnet wurden, werden dort resident gespeichert.

Syntax: **Program_Resident_Actions_Request** 0158_{hex}



Legende: Code: 0158_{hex} Kommando-Code des Requests
 Parameter_Count: Anzahl der nachfolgenden Worte
 0000_{hex} kein Parameterwort

Syntax: **Program_Resident_Actions_Confirmation** **8158_{hex}**



Legende:

Code: 8158_{hex} Meldungs-Code der Confirmation

Parameter_Count: Anzahl der nachfolgenden Worte
bei positiver Meldung:
0001_{hex} 1 Parameterwort
bei negativer Meldung:
0002_{hex} 2 Parameterworte

Result: Ergebnis der Dienstbearbeitung
0000_{hex} zeigt eine positive Meldung an.
Die Anschaltbaugruppe hat den Dienst erfolgreich bearbeitet.
xxxx_{hex} zeigt eine negative Meldung an.
Die Anschaltbaugruppe konnte den aufgerufenen Dienst nicht bearbeiten.
Der Parameter *Result* enthält die Ursache, warum der Dienst nicht bearbeitet werden konnte.

Add_Error_Info: Zusätzliche Informationen zur Fehlerursache

2.9.2 Dienst „Clear_Parameterization_Memory“

Aufgabe: Dieser Dienst formatiert den Parametrierungsspeicher.

Syntax: **Clear_Parameterization_Memory_Request** **0159_{hex}**

Wort 1	Code
Wort 2	Parameter_Count
Wort 3	Action_Flag
Wort 4	Area_Flag

Bit | 150 |

Legende:

Code:	0159 _{hex}	Kommando-Code des Requests
Parameter_Count:	Anzahl der nachfolgenden Worte	
	0002 _{hex}	2 Parameterworte
Action_Flag:	Auszuführende Aktion	
	0002 _{hex}	Parametrierungsspeicher formatieren
	000B _{hex}	Der Dienst wurde gestartet.
Area_Flag:	Bereichseinschränkungen	
	000A _{hex}	Keine Einschränkungen

Syntax: **Clear_Parameterization_Memory_Confirmation** **8159_{hex}**

Positive Meldung

Wort 1	Code
Wort 2	Parameter_Count
Wort 3	Result

Negative Meldung

Wort 1	Code
Wort 2	Parameter_Count
Wort 3	Result
Wort 4	Add_Error_Info

Bit | 15 0 |

Legende:

Code: 8159_{hex} Meldungs-Code der Confirmation

Parameter_Count: Anzahl der nachfolgenden Worte
 bei positiver Meldung:
 0001_{hex} 1 Parameterwort
 bei negativer Meldung:
 0002_{hex} 2 Parameterworte

Result: Ergebnis der Dienstbearbeitung
 0000_{hex} zeigt eine positive Meldung an.
 Die Anschaltbaugruppe hat den Dienst erfolgreich bearbeitet.
 xxxx_{hex} zeigt eine negative Meldung an.
 Die Anschaltbaugruppe konnte den aufgerufenen Dienst nicht bearbeiten.
 Der Parameter *Result* enthält die Ursache, warum der Dienst nicht bearbeitet werden konnte.

Add_Error_Info: Zusätzliche Informationen zur Fehlerursache



Der Zustand des Dienstes „Clear_Parameterization_Memory“ kann über den Dienst „Read_Value“ 0351_{hex} (siehe Seite 2-25) mit den folgenden Einstellungen abgefragt werden.

Parameter_Count: 0002_{hex}
 Variable_Count: 0001_{hex}
 Variable_ID: 1711_{hex}

Beachten Sie, dass der Dienst zyklisch abgesetzt werden muss. Da der Clear-Vorgang bei jedem Read-Vorgang unterbrochen wird, verlängert sich die Zeitdauer bis zum Abschluss des Clear-Vorganges.

Syntax: **Read_Value_Confirmation** **8351_{hex}**

Positive Meldung

Wort 1	Code		
Wort 2	Parameter_Count		000A _{hex}
Wort 3	Result		0000 _{hex}
Wort 4	Variable_Count		0001 _{hex}
Wort 5	Variable_ID		1711 _{hex}
Wort 6 + 7	File_Format_Result		
	File_Format_Add_Err		
Wort 8 + 9	Max_Duration	(Bit 31 ... 16)	
	Max_Duration	(Bit 15 ... 0)	
Wort 10 + 11	Current_Duration	(Bit 31 ... 16)	
	Current_Duration	(Bit 15 ... 0)	
Wort 12	Ready Flag	reserviert	
Bit	158 70		

- Legende:
- Code: 8351_{hex} Meldungs-Code der Confirmation
 - Parameter_Count: Anzahl der nachfolgenden Worte
 - Result: Ergebnis der Dienstbearbeitung
 - Variable_Count: Anzahl der ausgelesenen Systemparameter
 - Variable_ID: ID des ausgelesenen Systemparameters
 - File_Format_Result: Fehler-Code, siehe Kapitel „Fehler-Codes“ auf Seite 3-3
 - File_Format_Add_Err: Zusätzliche Fehlerangaben
 - Max_Duration: Gibt die Anzahl der Schritte (Ticks) an, die für diesen Dienst benötigt werden.
 - Current_Duration: Gibt die Anzahl der Schritte (Ticks) an, die seit dem Start des Vorganges vergangen sind.
 - Ready Flag: = 0: Dienstbearbeitung noch aktiv
<> 0: Dienstbearbeitung abgeschlossen

2.9.3 Dienst „File_Open“

Aufgabe: Dieser Dienst öffnet eine Datei. Das erlaubte Zugriffsrecht wird mit den beiden Parametern *Access* und *Mode* bestimmt.

Syntax: **File_Open_Request** **015B_{hex}**

Wort 1	Code		
Wort 2	Parameter_Count		
Wort 3 + 4	Access	(Bit 31 ... 16)	
	Access	(Bit 15 ... 0)	
Wort 5 + 6	Mode	(Bit 31 ... 16)	0200 _{hex}
	Mode	(Bit 15 ... 0)	0400 _{hex}
Wort 7 ... (13)	Name_Length	Name	Dateiname
	
	...	Name	

Bit | 158 | 70 |

Legende:

Code: 015B_{hex} Kommando-Code des Requests

Parameter_Count: Anzahl der nachfolgenden Worte
 $000x_{hex} = 4 + (Name_Length + 1) / 2$
 Wertebereich: 0005_{hex} ... 000B_{hex}

Access: 32-Bit-Flag, über das die Zugriffsrechte gesetzt werden:
 0000 0001_{hex} Datei öffnen (nur Lesen)
 0000 0002_{hex} Datei öffnen (nur Schreiben)
 0000 0008_{hex} Datei öffnen (Daten hinzufügen)
 0000 0010_{hex} Datei erzeugen (bei Bedarf)
 0000 0020_{hex} Datei exklusiv öffnen
 0000 0040_{hex} Datei öffnen (Daten löschen)
 0000 0100_{hex} Binär-Datei

Mode: 32-Bit-Flag (0200 0400_{hex})

Name_Length: Länge des Dateinamens (in Bytes)

Name: Name der Datei, die geöffnet werden soll (max. 12 Zeichen, MS DOS-Format)

Syntax:

File_Open_Confirmation

815B_{hex}

Positive Meldung

Wort 1	Code		
Wort 2	Parameter_Count		
Wort 3	Result		
Wort 4 + 5	File_Handle	(Bit 31 ... 16)	Zugriffs-ID
	File_Handle	(Bit 15 ... 0)	
Wort 6 ... (12)	Name_Length	Name	Dateiname
	
	...	Name	

Negative Meldung

Wort 1	Code		
Wort 2	Parameter_Count		
Wort 3	Result		
Wort 4 + 5	File_Error	(Bit 31 ... 16)	zus. Fehler-Information
	File_Error	(Bit 15 ... 0)	
Wort 6 ... (12)	Name_Length	Name	Dateiname
	
	...	Name	

Bit | 158 | 70 |

Legende:

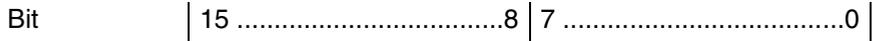
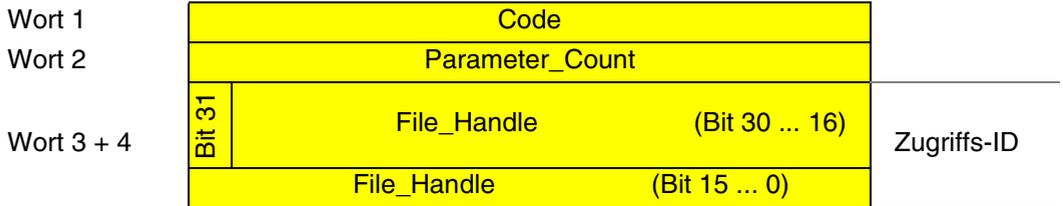
Code: 815B_{hex} Meldungs-Code der Confirmation
 Parameter_Count: Anzahl der nachfolgenden Worte
 bei positiver und negativer Meldung:
 $000x_{hex} = 3 + (Name_Length + 1) / 2$
 Wertebereich: 0004_{hex} ... 000A_{hex}

Result:	Ergebnis der Dienstbearbeitung 0000 _{hex} zeigt eine positive Meldung an. Die Anschaltbaugruppe hat den Dienst erfolgreich bearbeitet und die Datei auf der Speicherkarte geöffnet. 09DE _{hex} zeigt eine negative Meldung an. Die Anschaltbaugruppe konnte den aufgerufenen Dienst nicht bearbeiten. Die Datei auf der Speicherkarte konnte nicht geöffnet werden (siehe Fehler-Codes, Seite 3-20).
File_Handle:	Zugriffs-ID Diese 32-Bit-Zugriffs-ID ist mit dem Dateinamen assoziiert und muss bei den anschließenden Dateioperationen benutzt werden.
Name_Length:	Länge des Dateinamens (in Bytes)
Name:	Dateiname der Datei, die geöffnet werden soll (max. 12 Zeichen, MS DOS-Format)
File_Error:	Zusätzliche Informationen zur Fehlerursache (32-Bit-Wert)

2.9.4 Dienst „File_Close“

Aufgabe: Dieser Dienst schließt eine geöffnete Datei.

Syntax: **File_Close_Request** **015C_{hex}**



- Legende:**
- Code: 015C_{hex} Kommando-Code des Requests
 - Parameter_Count: Anzahl der nachfolgenden Worte
0002_{hex} 2 Parameterworte
 - File_Handle: Zugriffs-ID
Die Zugriffs-ID (32-Bit-Wert) ist mit dem Dateinamen assoziiert. Sie erhalten sie beim erfolgreichen Öffnen der Datei von der Confirmation „File_Open_Confirmation“ (815B_{hex}).
 - Bit 31 (No-Wait-Bit)
 - = 1: Der Controller generiert sofort eine Confirmation, der Dienst „File_Close“ wird im Hintergrund abgearbeitet.
 - = 0: Die Confirmation wird erst dann vom Controller generiert, wenn der Dienst „File_Close“ vollständig abgearbeitet wurde.

Syntax: **File_Close_Confirmation** **815C_{hex}**

Positive Meldung

Wort 1	Code		
Wort 2	Parameter_Count		
Wort 3	Result		0000 _{hex}
Wort 4 + 5	Bit 31	File_Handle (Bit 30 ... 16)	Zugriffs-ID
		File_Handle (Bit 15 ... 0)	

Negative Meldung

Wort 1	Code		
Wort 2	Parameter_Count		
Wort 3	Result		09DD _{hex}
Wort 4 + 5	Bit 31	File_Handle (Bit 30 ... 16)	Zugriffs-ID
		File_Handle (Bit 15 ... 0)	
Wort 6 + 7		File_Error (Bit 31 ... 16)	zus. Fehler-Information
		File_Error (Bit 15 ... 0)	

Bit | 158 | 70 |

Legende:

Code: 815C_{hex} Meldungs-Code der Confirmation

Parameter_Count: Anzahl der nachfolgenden Worte
 bei positiver Meldung:
 0003_{hex} 3 Parameterworte
 bei negativer Meldung:
 0005_{hex} 5 Parameterworte

Result: Ergebnis der Dienstbearbeitung
 0000_{hex} zeigt eine positive Meldung an.
 Die Datei auf der Speicherkarte wurde geschlossen.
 09DD_{hex} zeigt eine negative Meldung an.
 Die Datei auf der Speicherkarte konnte nicht geschlossen werden.
 (Siehe Fehler-Codes, Seite 3-20).

File_Handle: Zugriffs-ID
 Der Parameter enthält den Wert, der mit dem Request übergeben wurde.
 File_Error: Zusätzliche Informationen zur Fehlerursache (32-Bit-Wert)



Der Zustand des Dienstes „File_Close“ kann über den Dienst „Read_Value“ 0351_{hex} (siehe Seite 2-25) mit den folgenden Einstellungen abgefragt werden.

Parameter_Count: 0002_{hex}
 Variable_Count: 0001_{hex}
 Variable_ID: 1712_{hex}

Beachten Sie, dass der Dienst zyklisch abgesetzt werden muss. Da der Close-Vorgang bei jedem Read-Vorgang unterbrochen wird, verlängert sich die Zeitdauer bis zum Abschluss des Close-Vorganges.

Syntax: Read_Value_Confirmation 8351_{hex}

Positive Meldung

Wort 1	Code		
Wort 2	Parameter_Count		000A _{hex}
Wort 3	Result		0000 _{hex}
Wort 4	Variable_Count		0001 _{hex}
Wort 5	Variable_ID		1712 _{hex}
Wort 6 + 7	File_Close_Result		
	File_Close_Add_Err		
Wort 8 + 9	Max_Duration	(Bit 31 ... 16)	
	Max_Duration	(Bit 15 ... 0)	
Wort 10 + 11	Current_Duration	(Bit 31 ... 16)	
	Current_Duration	(Bit 15 ... 0)	
Wort 12	Ready Flag	reserviert	
Bit	158 70		

Legende: Code: 8351_{hex} Meldungs-Code der Confirmation
 Parameter_Count: Anzahl der nachfolgenden Worte
 Result: Ergebnis der Dienstbearbeitung
 Variable_Count: Anzahl der ausgelesenen Systemparameter
 Variable_ID: ID des ausgelesenen Systemparameters

File_Close_Result:	Fehler-Code, siehe Kapitel „Fehler-Codes“ auf Seite 3-3
File_Close_Add_Err:	Zusätzliche Fehlerangaben
Max_Duration:	Gibt die Anzahl der Schritte (Ticks) an, die für diesen Dienst benötigt werden.
Current_Duration:	Gibt die Anzahl der Schritte (Ticks) an, die seit dem Start des Vorganges vergangen sind.
Ready Flag:	= 0: Dienstbearbeitung noch aktiv <> 0: Dienstbearbeitung abgeschlossen

2.9.5 Dienst „File_Remove“

Aufgabe: Dieser Dienst löscht die angegebene Datei.

Syntax: **File_Remove_Request** **015D_{hex}**

Wort 1	Code		Dateiname
Wort 2	Parameter_Count		
Wort 3 ... (9)	Name_Length	Name	
	
	...	Name	

Bit | 158 | 70 |

- Legende:**
- Code: 015D_{hex} Kommando-Code des Requests
 - Parameter_Count: Anzahl der nachfolgenden Worte
 $000x_{hex} = (Name_Length + 1) / 2$
 Wertebereich: 0001_{hex} ... 0007_{hex}
 - Name_Length: Länge des Dateinamens (in Bytes)
 - Name: Dateiname der Datei, die gelöscht werden soll
 (max. 12 Zeichen, MS DOS-Format)

Syntax: **File_Remove_Confirmation** **815D_{hex}**

Positive Meldung

Wort 1	Code		
Wort 2	Parameter_Count		
Wort 3	Result		0000 _{hex}
Wort 4 ... (10)	Name_Length	Name	Dateiname
	
	...	Name	

Negative Meldung

Wort 1	Code		
Wort 2	Parameter_Count		
Wort 3	Result		09DF _{hex}
Wort 4 ... (10)	Name_Length	Name	Dateiname
	
	...	Name	
Wort (11)+(12)	File_Error	(Bit 31 ... 16)	zus. Fehler-Information
	File_Error	(Bit 15 ... 0)	

Bit | 158 | 70 |

Legende:

Code: 815D_{hex} Meldungs-Code der Confirmation

Parameter_Count: Anzahl der nachfolgenden Worte
 bei positiver Meldung:
 $000x_{hex} = 1 + (Name_Length + 1) / 2$
 Wertebereich: 0001_{hex} ... 0008_{hex}

bei negativer Meldung:
 $000x_{hex} = 3 + (Name_Length + 1) / 2$
 Wertebereich: 0003_{hex} ... 000A_{hex}

Result: Ergebnis der Dienstbearbeitung

0000_{hex} zeigt eine positive Meldung an.
 Die Datei wurde von der Speicherkarte gelöscht.

09DF_{hex} zeigt eine negative Meldung an.
 Die Datei konnte nicht von der Speicherkarte gelöscht werden
 (vgl. Fehler-Codes, Seite 3-21).

Name_Length:	Länge des Dateinamens (in Bytes)
Name:	Name der Datei, die gelöscht wurde (max. 12 Zeichen, MS DOS-Format)
File_Error:	Zusätzliche Informationen zur Fehlerursache (32-Bit-Wert)

2.9.6 Dienst „File_Write“

Aufgabe: Dieser Dienst schreibt Daten in die Datei mit der Zugriffs-ID *File_Handle*, die vorher mit dem Dienst „File_Open“ zum Schreiben geöffnet wurde.

Syntax: **File_Write_Request** **015E_{hex}**

Wort 1	Code		
Wort 2	Parameter_Count		
Wort 3 + 4	File_Handle	(Bit 31 ... 16)	Zugriffs-ID
	File_Handle	(Bit 15 ... 0)	
Wort 5	Data_Length		zu schrei- bende Daten
Wort 6 ...	Data	Data	
	

Bit | 158 | 70 |

- Legende:**
- Code: 015E_{hex} Kommando-Code des Requests
 - Parameter_Count: Anzahl der nachfolgenden Worte
 $xxxx_{hex} = 3 + Data_Length / 2$
 - File_Handle: Zugriffs-ID
 Die Zugriffs-ID (32-Bit-Wert) ist mit dem Dateinamen assoziiert. Sie erhalten sie beim erfolgreichen Öffnen der Datei von der Confirmation „File_Open_Confirmation“ (815B_{hex}).
 - Data_Length: Länge des Parameters *Data* (in Bytes), Anzahl der Bytes, die geschrieben werden sollen
 - Data: Daten, die geschrieben werden sollen

Syntax:

File_Write_Confirmation

815E_{hex}

Positive Meldung

Wort 1	Code		
Wort 2	Parameter_Count		
Wort 3	Result		
Wort 4 + 5	File_Handle	(Bit 31 ... 16)	Zugriffs-ID
	File_Handle	(Bit 15 ... 0)	
Wort 6	Data_Length		

Negative Meldung

Wort 1	Code		
Wort 2	Parameter_Count		
Wort 3	Result		
Wort 4 + 5	File_Handle	(Bit 31 ... 16)	Zugriffs-ID
	File_Handle	(Bit 15 ... 0)	
Wort 6	Data_Length		
Wort 7 + 8	File_Error	(Bit 31 ... 16)	zus. Fehler-Information
	File_Error	(Bit 15 ... 0)	

Bit | 150 |

Legende:

Code: 815E_{hex} Meldungs-Code der Confirmation

Parameter_Count: Anzahl der nachfolgenden Worte
 bei positiver Meldung:
 0004_{hex} 4 Parameterworte
 bei negativer Meldung:
 0006_{hex} 6 Parameterworte

Result: Ergebnis der Dienstbearbeitung
 0000_{hex} zeigt eine positive Meldung an.
 Die Anschaltbaugruppe hat den Dienst erfolgreich bearbeitet und die Daten in die Datei geschrieben.
 09DB_{hex} zeigt eine negative Meldung an.
 Die Anschaltbaugruppe konnte den aufgerufenen Dienst nicht bearbeiten.
 In die Datei konnte nicht geschrieben werden (Siehe Fehler-Codes, Seite 3-20)

File_Handle:	Zugriffs-ID Der Parameter enthält den Wert, der mit dem Request übergeben wurde.
Data_Length:	Länge der Daten Der Parameter enthält den Wert, der mit dem Request übergeben wurde.
File_Error:	Zusätzliche Informationen zur Fehlerursache (32-Bit-Wert)

2.9.7 Dienst „File_Seek“

Aufgabe: Dieser Dienst positioniert den Datei-Zeiger auf die mit Parametern *Whence* und *Offset* angegebene Adresse.

Syntax: **File_Seek_Request** **015F_{hex}**

Wort 1	Code		
Wort 2	Parameter_Count		
Wort 3 + 4	File_Handle	(Bit 31 ... 16)	Zugriffs-ID
	File_Handle	(Bit 15 ... 0)	
Wort 5 + 6	Offset	(Bit 31 ... 16)	Abstand von Bezugsadr.
	Offset	(Bit 15 ... 0)	
Wort 7 + 8	Whence	(Bit 31 ... 16)	Bezugs- adresse
	Whence	(Bit 15 ... 0)	

Bit | 150 |

Legende:

Code: 015F_{hex} Kommando-Code des Requests

Parameter_Count: Anzahl der nachfolgenden Worte
0006_{hex} 6 Parameterworte

File_Handle: Zugriffs-ID
Die Zugriffs-ID (32-Bit-Wert) ist mit dem Dateinamen assoziiert. Sie erhalten sie beim erfolgreichen Öffnen der Datei von der Confirmation „File_Open_Confirmation“ (815B_{hex}).

Offset: Offset-Wert
Der Offset-Wert (32-Bit-Wert) ist die Entfernung der neuen Adresse des Datei-Zeigers von der durch den Parameter *Whence* angegebenen Ausgangsadresse.

Whence: Offset-Bezug
Der Offset-Bezug ist die Ausgangsadresse (32-Bit-Wert) von der aus die neue Adresse (Parameter *Offset*) des Datei-Zeigers gerechnet wird.

SEEK_END	Datei-Ende	(2 _{dez})
SEEK_CUR	aktuelle Adresse	(1 _{dez})
SEEK_SET	Datei-Anfang	(0 _{dez})

Syntax:

File_Seek_Confirmation

815F_{hex}

Positive Meldung

Wort 1	Code		
Wort 2	Parameter_Count		
Wort 3	Result		
Wort 4 + 5	File_Handle	(Bit 31 ... 16)	Zugriffs-ID
	File_Handle	(Bit 15 ... 0)	
Wort 6	Count_Of_Possible_Seek	(Bit 31 ... 16)	wie <i>Offset</i>
Wort 7	Count_Of_Possible_Seek	(Bit 15 ... 0)	

Negative Meldung

Wort 1	Code		
Wort 2	Parameter_Count		
Wort 3	Result		
Wort 4 + 5	File_Handle	(Bit 31 ... 16)	Zugriffs-ID
	File_Handle	(Bit 15 ... 0)	
Wort 6 + 7	Count_Of_Possible_Seek	(Bit 31 ... 16)	Anzahl der Suchschritte
	Count_Of_Possible_Seek	(Bit 15 ... 0)	
Wort 8 + 9	File_Error	(Bit 31 ... 16)	Zus. Fehlerinformation
	File_Error	(Bit 15 ... 0)	

Bit | 150 |

Legende:

Code: 815F_{hex} Meldungs-Code der Confirmation
 Parameter_Count: Anzahl der nachfolgenden Worte
 bei positiver Meldung:
 0005_{hex} 5 Parameterworte
 bei negativer Meldung:
 0007_{hex} 7 Parameterworte

Result:	Ergebnis der Dienstbearbeitung 0000 _{hex} zeigt eine positive Meldung an. Die Anschaltbaugruppe hat den Dienst erfolgreich bearbeitet und den Datei-Zeiger neu positioniert. 09DC _{hex} zeigt eine negative Meldung an. Die Anschaltbaugruppe konnte den aufgerufenen Dienst nicht bearbeiten. Der Datei-Zeiger konnte nicht neu positioniert werden (Siehe Fehler-Codes, Seite 3-20).
File_Handle:	Zugriffs-ID Der Parameter enthält den Wert, der mit dem Request übergeben wurde.
Count_Of_Possible_Seek:	Anzahl der durchgeführten Offset-Schritte Der Wert des Parameters <i>Count_Of_Possible_Seek</i> entspricht dem Wert des Parameters <i>Offset</i> in dem Request.
File_Error:	Zusätzliche Informationen zur Fehlerursache (32-Bit-Wert)

2.9.8 Dienst „File_Read“

Aufgabe: Dieser Dienst liest aus einer Datei Daten aus. Die Datei wird über die Zugriffs-ID *File_Handle* angesprochen.

Syntax: **File_Read_Request** 0160_{hex}

Wort 1	Code		
Wort 2	Parameter_Count		
Wort 3 + 4	File_Handle	(Bit 31 ... 16)	Zugriffs-ID
	File_Handle	(Bit 15 ... 0)	
Wort 5	Data_Length		

Bit | 150 |

- Legende:**
- Code: 0160_{hex} Kommando-Code des Requests
 - Parameter_Count: Anzahl der nachfolgenden Worte
0003_{hex} 3 Parameterworte
 - File_Handle: Zugriffs-ID
Die Zugriffs-ID (32-Bit-Wert) ist mit dem Dateinamen assoziiert. Sie erhalten sie beim erfolgreichen Öffnen der Datei von der Confirmation „File_Open_Confirmation“ (815B_{hex}).
 - Data_Length: Anzahl der Daten-Bytes, die aus der Datei gelesen werden sollen

Syntax: **File_Read_Confirmation** **8160_{hex}**

Positive Meldung

Wort 1	Code		
Wort 2	Parameter_Count		
Wort 3	Result		
Wort 4 + 5	File_Handle	(Bit 31 ... 16)	Zugriffs-ID
	File_Handle	(Bit 15 ... 0)	
Wort 6	Data_Length		ausgelesene Daten
Wort 7 ...	Data	Data	
	

Negative Meldung

Wort 1	Code		
Wort 2	Parameter_Count		
Wort 3	Result		
Wort 4 + 5	File_Handle	(Bit 31 ... 16)	Zugriffs-ID
	File_Handle	(Bit 15 ... 0)	
Wort 6 + 7	File_Error	(Bit 31 ... 16)	zus. Fehler-Information
	File_Error	(Bit 15 ... 0)	

Bit | 150 |

Legende:

Code: 8160_{hex} Meldungs-Code der Confirmation

Parameter_Count: Anzahl der nachfolgenden Worte
 bei positiver Meldung:
 $xxxx_{hex} = 4 + Data_Length / 2$
 bei negativer Meldung:
 0005_{hex} 5 Parameterworte

Result: Ergebnis der Dienstbearbeitung
 0000_{hex} zeigt eine positive Meldung an.
 Die Anschaltbaugruppe hat den Dienst erfolgreich bearbeitet und die angeforderten Daten ausgelesen.
 09DA_{hex} zeigt eine negative Meldung an.
 Die Anschaltbaugruppe konnte den aufgerufenen Dienst nicht bearbeiten.
 Es wurden keine Daten gelesen (Siehe Fehler-Codes, Seite 3-20).

File_Handle:	Zugriffs-ID Der Parameter enthält den Wert, der mit dem Request übergeben wurde.
Data_Length:	Anzahl der Bytes, die gelesen wurden
Data:	Ausgelesene Daten (byteweise)
File_Error:	Zusätzliche Informationen zur Fehlerursache (32-Bit-Wert)

2.9.9 Dienst File_Remove_II

Aufgabe:



Dieser Dienst löscht die angegebene Datei.

Der Unterschied dieses Dienstes zum Dienst „File_Remove“ 015D_{hex} besteht darin, dass bei diesem Dienst die Confirmation sofort, und nicht erst nach Abarbeitung dieses Dienstes zurück kommt.

Syntax:

File_Remove_II_Request

0165_{hex}

Wort 1	Code		Dateiname
Wort 2	Parameter_Count		
Wort 3 ... (9)	Name_Length	Name	
	
	...	Name	

Bit | 158 | 70 |

Legende:

- Code: 0165_{hex} Kommando-Code des Requests
- Parameter_Count: Anzahl der nachfolgenden Worte
 $000x_{hex} = (Name_Length + 1) / 2$
 Wertebereich: 0001_{hex} ... 0007_{hex}
- Name_Length: Länge des Dateinamens (in Bytes)
- Name: Dateiname der Datei, die gelöscht werden soll (max. 12 Zeichen, MS DOS-Format)



Der Zustand des Dienstes „File_Remove_II“ kann über den Dienst „Read_Value“ 0351_{hex} (siehe Seite 2-25) mit den folgenden Einstellungen abgefragt werden.

- Parameter_Count: 0002_{hex}
- Variable_Count: 0001_{hex}
- Variable_ID: 1713_{hex}

Beachten Sie, dass der Dienst zyklisch abgesetzt werden muss. Da der Remove-Vorgang bei jedem Read-Vorgang unterbrochen wird, verlängert sich die Zeitdauer bis zum Abschluss des Remove-Vorganges.

Syntax: **Read_Value_Confirmation** **8351_{hex}**

Positive Meldung

Wort 1	Code		
Wort 2	Parameter_Count		000A _{hex}
Wort 3	Result		0000 _{hex}
Wort 4	Variable_Count		0001 _{hex}
Wort 5	Variable_ID		1713 _{hex}
Wort 6 + 7	File_Remove_Result		
	File_Remove_Add_Err		
Wort 8 + 9	Max_Duration	(Bit 31 ... 16)	
	Max_Duration	(Bit 15 ... 0)	
Wort 10 + 11	Current_Duration	(Bit 31 ... 16)	
	Current_Duration	(Bit 15 ... 0)	
Wort 12	Ready Flag	reserviert	
Bit	158 70		

- Legende:
- Code: 8351_{hex} Meldungs-Code der Confirmation
 - Parameter_Count: Anzahl der nachfolgenden Worte
 - Result: Ergebnis der Dienstbearbeitung
 - Variable_Count: Anzahl der ausgelesenen Systemparameter
 - Variable_ID: ID des ausgelesenen Systemparameters
 - File_Remove_Result: Fehler-Code, siehe Kapitel „Fehler-Codes“ auf Seite 3-3
 - File_Remove_Add_Err: Zusätzliche Fehlerangaben
 - Max_Duration: Gibt die Anzahl der Schritte (Ticks) an, die für diesen Dienst benötigt werden.
 - Current_Duration: Gibt die Anzahl der Schritte (Ticks) an, die seit dem Start des Vorganges vergangen sind.
 - Ready Flag: = 0: Dienstbearbeitung noch aktiv
<> 0: Dienstbearbeitung abgeschlossen

2.9.10 Dienst Get_Card_Information

Aufgabe: Dieser Dienst liest Informationen des Parametrierungsspeichers aus. Über den Inhalt des *Directory_Flags* können Sie einstellen, ob die Directory oder die technischen Daten des Parametrierungsspeichers ausgelesen werden sollen.

Der Aufbau der Confirmation ist abhängig von der Einstellung des Parameters *Directory_Flag*.

2.9.10.1 Auslesen der Directory

Syntax: **Get_Card_Information_Request** **0166_{hex}**

Wort 1	Code
Wort 2	Parameter_Count
Wort 3	Directory_Flag
Wort 4	Start_Flag

Bit | 150 |

- Legende:**
- Code: 0166_{hex} Kommando-Code des Requests
 - Parameter_Count: Anzahl der nachfolgenden Worte
0002_{hex} 2 Parameterworte
 - Directory_Flag: 0001_{hex} Directory des Parametrierungsspeichers auslesen
 - Start_Flag: = 0: Auslesen der Directory ab dem ersten Eintrag
= 1: Auslesen der Directory ab dem zuletzt ausgelesenen Eintrag plus 1



Wenn der Rest der Directory mit dem nächsten Request gelesen werden soll, dann muss das Start_Flag auf 1 gesetzt werden. Bedingung: In der Confirmation muss das More_Follows_Flag gesetzt worden sein.

Syntax: **Get_Card_Information_Confirmation** **8166_{hex}**

Positive Meldung

Wort 1	Code
Wort 2	Parameter_Count
Wort 3	Result
Wort 4	Directory_Flag
Wort 5	More_Follows
Wort 6	File_Counts
Wort 7	Directory-Eintrag 1
Wort 8	Directory-Eintrag 2
Wort 9	...
Wort 10	Directory-Eintrag n - 1 (n = File_Counts)
Wort 11	Directory-Eintrag n (n = File_Counts)

Bit | 150 |

- Legende:
- Code: 8166_{hex} Meldungs-Code der Confirmation
 - Parameter_Count: Anzahl der nachfolgenden Worte
 - Result: Ergebnis der Dienstbearbeitung
 - Directory_Flag: 0001_{hex} Directory des Parametrierungs-speichers
 - More_Follows: 0001_{hex} Die Directory ist noch nicht vollständig ausgelesen worden.
 - File_Counts: xxxx_{hex} Anzahl der Directory-Einträge
 - Directory-Eintrag x: Einträge in der Directory entsprechend dem folgenden Aufbau:

Einträge in der Directory

Filename	Data [00]	Data [01]
Filename	Data [02]	Data [03]
Filename	Data [04]	Data [05]
Filename	Data [06]	Data [07]
Filename	Data [08]	Data [09]
Filename	Data [10]	Data [11]

Filename	Data [12]	Data [13]
Filesize	File_Size	(Bit 31 ... 16)
Filesize	File_Size	(Bit 15 ... 0)

Bit | 150 |

Legende: Filename: Dateiname
 Filesize: Dateigröße
 Data [x]: ASCII-codierte Daten

2.9.10.2 Auslesen der Parametrierungsspeicher-Beschreibung

Hier werden die technischen Daten des Parametrierungsspeichers ausgelesen, die nach der Formatierung des Parametrierungsspeichers zur Verfügung stehen. Sie können beispielsweise den maximal von der Karte nach der Formatierung zur Verfügung gestellten Speicher auslesen.

Syntax: **Get_Card_Information_Request** **0166_{hex}**

Wort 1	Code
Wort 2	Parameter_Count
Wort 3	Directory_Flag
Wort 4	reserviert

Bit | 150 |

Legende: Code: 0166_{hex} Kommando-Code des Requests
 Parameter_Count: Anzahl der nachfolgenden Worte
 0002_{hex} 2 Parameterworte
 Directory_Flag: 0002_{hex} Auslesen der Maximal-Daten des
 Parametrierungsspeichers
 Start_Flag: = 0: nicht relevant (immer = 0)

Syntax:

Get_Card_Information_Confirmation

8166_{hex}

Positive Meldung

Wort 1	Code	
Wort 2	Parameter_Count	
Wort 3	Result	
Wort 4	Directory_Flag	
Wort 5	Max_Anzahl_Files	
Wort 6	Max_File_Size	(Bit 31 ... 16)
Wort 7	Max_File_Size	(Bit 15 ... 0)
Wort 8	Volume_Size	(Bit 31 ... 16)
Wort 9	Volume_Size	(Bit 15 ... 0)

Bit | 150 |

Legende:

- Code: 8166_{hex} Meldungs-Code der Confirmation
- Parameter_Count: Anzahl der nachfolgenden Worte
- Result: Ergebnis der Dienstbearbeitung
- Directory_Flag: 0002_{hex} Die ausgelesenen Daten geben die Maximal-Daten des eingesetzten Parametrierungsspeichers an.
- Max_Anzahl_Files: Maximal Anzahl abspeicherbarer Dateien
- Max_File-Size: Maximale Dateigröße beim eingesetzten Parametrierungsspeicher-Typ
- Volume_Size: Maximal zur Verfügung stehender Speicher



Bedingt durch das Filesystem, kann der Parametrierungsspeicher nach der Formatierung eine bestimmte Anzahl von Dateien speichern. Nach Erreichen dieser Anzahl, muss der Parametrierungsspeicher erneut formatiert werden, um weitere Dateien zu speichern.

Beispiel:

Mit dem Parameter Max_Anzahl_Files lesen Sie die maximal auf dem Parametrierungsspeicher speicherbare Anzahl von Dateien aus, z. B. 60. Nach 45 abgelegten Dateien können 15 weitere abgelegt werden, bevor ein erneuter Formatierungsvorgang notwendig wird, um weitere Dateien abzulegen. Auch wenn z. B. durch einen „File_Remove“-Dienst 8 Dateien gelöscht werden, können nur 15 Dateien abgelegt werden. Erst nach einer erneuten Formatierung können wieder 60 Dateien gespeichert werden.

2.10 Selbstständige Fehleranzeigen der Anschaltbaugruppe

Mit den folgenden Fehleranzeigen (Indications) meldet die Anschaltbaugruppe von sich aus Informationen an das Anwenderprogramm.



Zuvor müssen Sie die Übertragung dieser Fehler-Codes über das Mailbox-Interface der Anschaltbaugruppe an das Anwenderprogramm mit dem Dienst „Set_Indication“ (0152_{hex}, siehe Seite 2-15) freigeben.

2.10.1 Fehleranzeige „Fault“

Bedeutung:

Diese Fehleranzeige (Indication) überträgt Fehler-Codes der Anschaltbaugruppe an das Anwenderprogramm.

Syntax:

Fault_Indication

4341_{hex}

Wort 1	Code
Wort 2	Parameter_Count
Wort 3	Entry_Count
Wort 4	Error_Code
Wort 5	Add_Error_Info

Bit | 15 0 |

Legende:

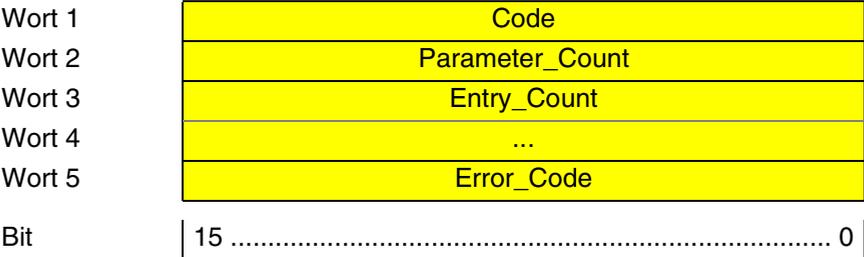
- Code: 4341_{hex} Code der Fehleranzeige
- Parameter_Count: Anzahl der nachfolgenden Worte
0003_{hex} 3 Parameterworte
- Entry_Count: Anzahl der Einträge bestehend aus 2 Worten:
0001_{hex} Die Fehleranzeige „Fault“ überträgt immer nur einen Fehler-Code.
- Error_Code: Fehler-Code, der den aufgetretenen Fehler näher erläutert, z. B.:
0B8D_{hex} In der Betriebsart *Asynchron* ist ein Synchroninterrupt aufgetreten.
0B8F_{hex} Überlauf d. Eingangs-FIFO des IPMS
0BD2_{hex} Die Buswarnzeit ist abgelaufen.
- Add_Error_Info: Zusätzliche Informationen zur Fehlerursache

2.10.2 Fehleranzeige „Lower_API_Fault“

Bedeutung: Diese Fehleranzeige (Indication) überträgt Fehler-Codes der Anschaltbaugruppe an das Anwenderprogramm.

Die Fehleranzeige tritt im Zusammenhang mit der Fehlermeldung 0914_{hex} auf.

Syntax: **Lower_API_Fault_Indication** 4B58_{hex}



Legende:

Code: 4B58_{hex} Code der Fehleranzeige

Parameter_Count: Anzahl der nachfolgenden Worte
0003_{hex} 3 Parameterworte

Entry_Count: Anzahl der Einträge bestehend aus 2 Worten:
0001_{hex} Die Fehleranzeige „Lower_API_Fault“ überträgt immer nur einen Fehler-Code.

Error_Code: Fehler-Code, der den aufgetretenen Fehler näher erläutert

2.10.3 Fehleranzeige „Device_Fail“

Bedeutung: Diese Fehleranzeige (Indication) überträgt Fehlermeldungen von maximal zehn INTERBUS-Teilnehmern von der Anschaltbaugruppe an das Anwenderprogramm.

Syntax: **Device_Fail_Indication** **5340_{hex}**

Wort 1	Code	
Wort 2	Parameter_Count	
Wort 3	Entry_Count	
Wort 4	Error_Code	1. Fehler
	Device_No	
...	...	
	Error_Code	(10). Fehler
Wort (21)	Device_No	

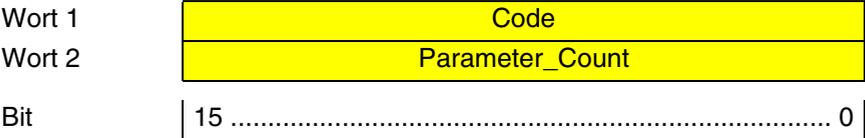
Bit | 15 0 |

- Legende:**
- Code: 5340_{hex} Code der Fehleranzeige
 - Parameter_Count: Anzahl der nachfolgenden Worte
 $00xx_{hex} = 1 + 2 \times Entry_Count$
 Wertebereich: 0001_{hex} ... 0017_{hex}
 - Entry_Count: Anzahl der übertragenen Fehlermeldungen bestehend aus je 2 Worten.
 Es werden max. 10 Fehler-Codes und die zugehörigen Teilnehmernummern übertragen.
 - Error_Code: Fehler-Code
 Die Bedeutung der Fehler-Codes ist in Kapitel 3 beschrieben.
 - Device_No: INTERBUS-Teilnehmer, der den Fehler gemeldet hat. Der Teilnehmer wird anhand seiner Teilnehmernummer identifiziert.

2.10.4 Fehleranzeige „Bus_Error“

- Bedeutung:** Der Datenverkehr auf dem Bus wurde gestoppt.
- Ursache:** Fehler oder Störungen bei der Datenübertragung.
- Abhilfe:** Lesen Sie die Fehlerursachen mit dem Dienst „Get_Error_Info“ (0316_{hex}, siehe Seite 2-113) aus.

Syntax: **Bus_Error_Indication** **6342_{hex}**



- Legende:
- Code: 6342_{hex} Code der Fehleranzeige
 - Parameter_Count: Anzahl der nachfolgenden Worte
 - 0000_{hex} kein Parameterwort

Diese Kapitel informiert Sie über

- die Bedeutung der Fehler-Codes
- die Ursachen der Fehler
- Hinweise zur Abhilfe und Behebung der Fehler

Fehler-Codes	3-3
3.1 Fehler-Codes zu Anwender-Fehlern (USER FAIL)	3-3
3.2 Fehler-Codes zu allgemeinen Busfehlern (BUS FAIL)	3-55
3.3 Fehler-Codes zu Fernbus- und Lokalbusfehlern	3-67
3.4 Fehler-Codes zu Systemfehlern	3-87
3.5 Fehler-Codes zu Anschaltbaugruppen mit Slaveteil	3-91
3.6 Fehler-Codes zu Anschaltbaugruppen mit COP-Karte	3-93

3 Fehler-Codes

3.1 Fehler-Codes zu Anwender-Fehlern (USER FAIL)

0902_{hex} (CTRL FAIL)

- Bedeutung:** Die Anschaltbaugruppe konnte den zuletzt aufgerufenen Dienst nicht bearbeiten.
- Ursache:** Auf der Anschaltbaugruppe ist ein Hardware- oder Firmware-Fehler aufgetreten.
- Abhilfe:** Tauschen Sie die Anschaltbaugruppe aus.

0903_{hex} (CTRL FAIL)

- Bedeutung:** Auf der Anschaltbaugruppe ist zu wenig Arbeitsspeicher frei.
- Ursache:** Beispielsweise kann der Arbeitsspeicher zu stark fragmentiert sein.
- Abhilfe:** Führen Sie mit dem Dienst „Reset_Controller_Board“ (0956_{hex}) einen Warmstart der Anschaltbaugruppe durch und versuchen Sie es erneut.

0904_{hex} (USER FAIL)

- Bedeutung:** Sie haben Parameter angegeben, die nicht aufeinander abgestimmt sind.
- Abhilfe:** Prüfen Sie die Parameter des zuletzt aufgerufenen Dienstes auf Widersprüche.

0905_{hex} (USER FAIL)

- Bedeutung:** Sie haben unzulässige Parameter angegeben.
- Abhilfe:** Prüfen Sie die Parameter des zuletzt aufgerufenen Dienstes.

0906_{hex} (USER FAIL)

Bedeutung: Der Zugriff wird für dieses Objekt nicht unterstützt.
Abhilfe: Prüfen Sie den letzten Zugriff.

0907_{hex} (USER FAIL)

Bedeutung: Sie haben versucht auf ein Objekt zuzugreifen, das nicht existiert.
Abhilfe: Prüfen Sie den letzten Zugriff.

0908_{hex} (USER FAIL)

Bedeutung: Die Firmware kann zwei Dienste nicht gleichzeitig bearbeiten.
Ursache: Sie haben einen Dienst aufgerufen, der mit einem anderen zuvor aufgerufenen Dienst in Bearbeitungskonflikt gerät.
Abhilfe: Warten Sie die Bearbeitung des zuvor aufgerufenen Dienstes ab, und senden Sie den Dienst anschließend erneut.
Add_Error_Info: Code des zuvor aufgerufenen Dienstes

0909_{hex} (USER FAIL)

Bedeutung: Die auf einen Dienst folgende Confirmation überschreitet die maximal zulässige Mailbox-Größe.
Abhilfe: Prüfen Sie das Kommando. Wenn das nicht hilft, setzen Sie sich mit Phoenix Contact in Verbindung.
Add_Error_Info: Größe dieser Confirmation in Byte

090A_{hex} (USER FAIL)

Bedeutung: Der Parameter *Parameter_Count* stimmt nicht mit der Anzahl der nachfolgenden Worte überein.
Add_Error_Info: Falscher Wert des Parameters *Parameter_Count*

090B_{hex} (USER FAIL)

Bedeutung:	Die Anschaltbaugruppe kann den angegebenen Dienst nicht freischalten.
Ursache:	Sie haben beim Freischalten dieses Dienstes ein falsches Passwort angegeben.
Abhilfe:	Verwenden Sie das richtige Passwort oder einen nicht gesperrten Dienst.

0910_{hex} (USER FAIL)

Bedeutung:	Die Anschaltbaugruppe kann den aufgerufenen Dienst nicht bearbeiten.
Ursache:	Der aufgerufene Dienst ist gesperrt.
Abhilfe:	<ul style="list-style-type: none">– Schalten Sie den Dienst vor dem Aufruf frei– Verwenden Sie einen nicht gesperrten Dienst.
Add_Error_Info:	Code des gesperrten Dienstes

0911_{hex} (USER FAIL)

Bedeutung:	Die Anschaltbaugruppe kann den angegebenen Dienst nicht freischalten.
Ursache:	Der Dienst ist bereits freigeschaltet.
Abhilfe:	Rufen Sie den Dienst auf.
Add_Error_Info:	Code des bereits freigeschalteten Dienstes

0912_{hex} (USER FAIL)

Bedeutung:	Die Anschaltbaugruppe kann den aufgerufenen Dienst nicht bearbeiten.
Ursache:	Der Dienst ist geschützt. Er kann nur mit Passwort freigeschaltet werden.
Abhilfe:	<ul style="list-style-type: none">– Schalten Sie den Dienst vor dem Aufruf frei– Verwenden Sie einen nicht geschützten Dienst.

0913_{hex} (USER FAIL)

Bedeutung:	Die Anschaltbaugruppe kann den aufgerufenen Dienst nicht bearbeiten.
Ursache:	Der Dienst wird für diese Anschaltbaugruppe nicht unterstützt.
Abhilfe:	Verwenden Sie einen Dienst, der von dieser Anschaltbaugruppe unterstützt wird.

0914_{hex}

- Bedeutung:** Der Sendepuffer ist voll, dadurch können keine Meldungen mehr an den Host weitergeleitet werden.
- Ursache:** Es sind zuviele Meldungen im Sendepuffer, weil der Host die anstehenden Meldungen nicht übernimmt, oder weil in kurzer Zeit sehr viele Meldungen generiert worden sind. Dies kann z. B. bei einem Wackelkontakt in der Peripheriespannung der INTERBUS-Teilnehmer auftreten.
- Abhilfe:** Stellen Sie sicher, dass der Host die anstehenden Meldungen übernimmt.

0915_{hex}

- Bedeutung:** Der Empfangspuffer ist voll, dadurch können keine Meldungen mehr von dem Host abgegeben werden.
- Ursache:** Es sind zuviele Meldungen im Empfangspuffer, weil die Anschaltbaugruppe die anstehenden Meldungen nicht übernimmt, oder weil in kurzer Zeit sehr viele Meldungen vom Host generiert worden sind.
- Abhilfe:** Stellen Sie sicher, dass die Anschaltbaugruppe die anstehenden Meldungen übernimmt.

0918_{hex} (USER FAIL)

- Bedeutung:** Unbekannter Dienst-Code wurde aufgerufen.
- Abhilfe:** Prüfen Sie den letzten Aufruf.
- Add_Error_Info:** Unbekannter Dienst-Code

0919_{hex}

- Bedeutung:** Der Dienst überschreitet die maximal zulässige Mailbox-Größe.
- Ursache:** Sie haben einen Dienst aufgerufen, der länger als 1024 Byte ist.
- Abhilfe:** Achten Sie darauf, dass Sie Dienste verwenden, die inklusive aller Parameter nicht länger als 1024 Byte sind. Zur Übergabe eines größeren Datensatzes können Sie einige Dienste mehrmals aufrufen, um den Datensatz in mehreren Schritten zu übergeben.

091A_{hex} (USER FAIL)

Bedeutung: Sie haben für den Parameter *Action_Index* einen unbekanntem Wert angegeben.

Add_Error_Info: Parameter *Action_Index*

091B_{hex} (USER FAIL)

Bedeutung: Sie haben für den Parameter *Action_Index* einen Wert angegeben, der bereits an anderer Stelle benutzt wird.

Add_Error_Info: Parameter *Action_Index*

091C_{hex}

Bedeutung: Falscher Request aufgetreten.

Ursache: Unerwartete Überschneidung im Action-Handler.

Abhilfe: Setzen Sie sich mit Phoenix Contact in Verbindung.

091D_{hex}

Ursache: Unerwartete Überschneidung im Action-Handler.

Abhilfe: Setzen Sie sich mit Phoenix Contact in Verbindung.

091E_{hex}

Ursache: Unerwartete Überschneidung im Action-Handler.

Abhilfe: Setzen Sie sich mit Phoenix Contact in Verbindung.

091F_{hex}

Bedeutung: Ein interner Konflikt ist durch einen unbekanntem Action-Index aufgetreten.

Ursache: Bei der Projektierung des Signal-Handlers konnte der gewünschte Action-Index oder das gewünschte Action-Objekt nicht gefunden werden.

Abhilfe: Erstellen Sie den gewünschten Action-Index bzw. das gewünschte Action-Objekt vor der Projektierung des Signal-Handlers.

0920_{hex} (USER FAIL)

Ursache: Sie haben für den Parameter *Signal_Index* einen Wert angegeben, der bereits an anderer Stelle benutzt wird.

Add_Error_Info: Parameter *Signal_Index*

0921_{hex}

Bedeutung: Zustandskonflikt im Signal-Handler.

Ursache:

1. Zuwenig Speicher auf der Hardware vorhanden.
Sie haben zu viele Signale ausgelöst.
2. Ein Programm blockiert PCP-Verbindung. PCP-Verbindung kann nicht mehr verwendet werden.

Abhilfe:

1. Sorgen Sie dafür, dass nicht so viele Signale ausgelöst werden.
2. Führen Sie einen Warmstart der Anschaltbaugruppe mit dem Dienst „Reset_Controller_Board“ (0956hex) durch.

0922_{hex}

Bedeutung: Zustandskonflikt im Action-Handler.

Ursache:

- Zuwenig Speicher auf der Hardware vorhanden.
- Sie haben zu viele Actions ausgelöst.

Abhilfe: Lösen Sie nicht so viele Actions aus.

0924_{hex}

Bedeutung: Zustandskonflikt im Event-Handler.

Ursache:

- Zuwenig Speicher auf der Hardware vorhanden.
- Zuviele Events vom Anwender ausgelöst.

Abhilfe: Sorgen Sie dafür, dass nicht zuviele Events ausgelöst werden.

0925_{hex} (USER FAIL)

Ursache: Sie haben für den Parameter *Event_Index* einen nicht definierten Wert angegeben.

Add_Error_Info: Parameter *Event_Index*

0926_{hex} (USER FAIL)

Ursache: Sie haben für den Parameter *Event_Index* einen Wert angegeben, der bereits an anderer Stelle benutzt wird.

Add_Error_Info: Parameter *Event_Index*

0928_{hex} (USER FAIL)

Bedeutung: Die Anschaltbaugruppe kann den aufgerufenen Dienst nicht bearbeiten.

Ursache: Sie haben einen exklusiv nutzbaren Dienst aufgerufen, ohne das Recht dazu zu haben.

Abhilfe:

- Ändern Sie (falls zulässig) mit dem Dienst „Change_Exclusive_Rights“ (014F_{hex}) die Vergabe des Rechts, exklusiv nutzbare Dienste aufzurufen
- Verwenden Sie einen nicht exklusiv nutzbaren Dienst.

0929_{hex}

Bedeutung: Eine Meldung kann nicht weitergereicht werden.

Ursache: Die angegebene Bitnummer der Zieladresse ist unbekannt.

Abhilfe: Wählen Sie eine erlaubte Bitnummer aus und senden Sie erneut die Meldung an das entsprechende Ziel:
 Bitnummer 0: Diagnose-Anzeige auf der Frontblende (falls vorhanden).
 Bitnummer 1: Standard-Signal-Interface (SSGI).

Add_Error_Info: Falsche Bitnummer

092A_{hex}

Bedeutung: Nicht zugelassener oder unbekannter Indication-Code

Ursache: Es wurde als Parameter ein nicht erlaubter Indication-Code eingesetzt.

Abhilfe: Geben Sie den Code der Fehleranzeige (Indication) an, die Sie freischalten oder sperren möchten, z. B. 4341_{hex} für die Fehleranzeige „Fault_Indication“.

Add_Error_Info: Falscher oder unerlaubter Indication-Code

092B_{hex} (USER FAIL)

- Ursache:
- Ein Teilnehmer hat eine ihm nicht zugeordnete Kommunikationsreferenz benutzt.
 - Über die V.24-Schnittstelle wurde ein Kanal geöffnet, auf den nicht zugegriffen werden kann.
 - Während ein PCP-Kanal geöffnet war, wurde die Schnittstelle geändert.

092D_{hex} (USER FAIL)

- Bedeutung: Die Anschaltbaugruppe hat einen PCP-Dienst abgewiesen.
- Ursache: Ein anderer PCP-Dienst wird bereits auf der Kommunikationsreferenz mit derselben Invoke-ID bearbeitet.
- Abhilfe: Warten Sie auf die PCP-Confirmation des laufenden Dienstes und setzen Sie erst dann den neuen Dienst ab.

092E_{hex}

- Ursache: Auf der Anschaltbaugruppe ist zu wenig Arbeitsspeicher frei, um das Standard-Signal-Interface (SSGI) zu initialisieren.
- Abhilfe: Führen Sie mit dem Dienst „Reset_Controller_Board“ (0956_{hex}) einen Warmstart der Anschaltbaugruppe durch und versuchen Sie es erneut.

0930_{hex} (USER FAIL)

- Bedeutung: Die Anschaltbaugruppe kann den Dienst „Set_Value“ (0750_{hex}) nicht bearbeiten.
- Ursache: Sie haben beim Aufruf des Dienstes falsche Parameterwerte angegeben.
- Abhilfe: Prüfen Sie den Aufruf.

0931_{hex} (USER FAIL)

- Bedeutung: Die Anschaltbaugruppe konnte den zuletzt aufgerufenen Dienst nicht bearbeiten.
- Ursache: Sie haben für den Parameter *Variable_ID* einen nicht definierten Wert benutzt.
- Abhilfe: Prüfen Sie den Parameter *Variable_ID*.

0932_{hex} (USER FAIL)

- Bedeutung:** Der angegebene MPM-Teilnehmer konnte die exklusiven Rechte nicht freigeben.
- Ursache:** Der angegebene MPM-Teilnehmer besaß die exklusiven Rechte nicht.
- Abhilfe:** Wenden Sie den Dienst „Change_Exclusive_Rights“ (014F_{hex}) zur Freigabe der exklusiven Rechte nur auf den MPM-Teilnehmer an, der sie besitzt.

0933_{hex} (USER FAIL)

- Bedeutung:** Die Anforderung der exklusiven Rechte wurde abgelehnt.
- Ursache:** Die exklusiven Rechte sind noch im Besitz eines anderen MPM-Teilnehmers.
- Abhilfe:** Wenden Sie den Dienst „Change_Exclusive_Rights“ (014F_{hex}) zur Freigabe der exklusiven Rechte nur an, wenn kein MPM-Teilnehmer diese Rechte bereits besitzt.

0934_{hex} (USER FAIL)

- Bedeutung:** Die Anforderung der exklusiven Rechte wurde abgelehnt.
- Ursache:** Der MPM-Teilnehmer besitzt die exklusiven Rechte bereits.

0935_{hex} (USER FAIL)

- Bedeutung:** Die angegebene Kommunikationsreferenz (CR) liegt außerhalb des zulässigen Wertebereichs.
- Abhilfe:** Vergeben Sie Kommunikationsreferenzen im Bereich von $2 \leq CR \leq 127$.

0936_{hex}

- Bedeutung:** Der Variablen_ID-Code ist falsch.
- Ursache:** Es wurde ein „Set_Value“ oder „Read_Value“ Dienst mit einem unbekanntem Variablen_ID-Code abgesendet.
- Abhilfe:** Senden Sie den Dienst mit dem richtigen ID-Code erneut ab.
- Add_Error_Info:** Falscher Variablen_ID-Code

0937_{hex}

Bedeutung: Die Variablen_ID-Komponente ist falsch.
Ursache: Es wurde ein „Set_Value“ oder „Read_Value“ Dienst mit einer unbekanntem Variablen_ID-Komponente abgesendet.
Abhilfe: Senden Sie den Dienst mit dem richtiger Variablen_ID-Komponenten erneut ab.
Add_Error_Info: Fehlerhafte Variablen_ID-Komponente

0938_{hex} (USER FAIL)

Bedeutung: Sie haben für den Parameter *Variable_ID* einen Wert benutzt, der für die Firmware reserviert ist.
Add_Error_Info: Reservierter Parameter *Variable_ID*

0939_{hex} (USER FAIL)

Bedeutung: Sie haben für den Parameter *Variable_ID* einen Wert benutzt, der nicht freigegeben ist.
Add_Error_Info: Nicht freigeschalteter Parameter *Variable_ID*

093A_{hex} (USER FAIL)

Bedeutung: Die Anschaltbaugruppe konnte den Dienst „Set_Value“ (0750_{hex}) nicht bearbeiten.
Ursache: Sie haben für den Parameter *Variable_ID* einen falschen Wert angegeben. In den Bits 8 bis 11 des Parameters *Variable_ID* wurde die Länge der Variablen in Worten nicht korrekt angegeben.
Abhilfe: Prüfen Sie den Aufruf dieses Dienstes.
Add_Error_Info: Falsche *Variable_ID*

093B_{hex} (USER FAIL)

- Bedeutung: Die Anschaltbaugruppe konnte den zuletzt aufgerufenen Dienst nicht bearbeiten.
- Ursache: Die im Parameter *Variable_ID* kodierte Variablen-Länge stimmt nicht mit der Länge der angegebenen Variable überein.
- Abhilfe: Prüfen Sie den Aufruf des zuletzt aufgerufenen Dienstes.

093C_{hex} (USER FAIL)

- Ursache: Sie haben ein Signal-Handle-Objekt nicht richtig definiert.
- Abhilfe: Prüfen Sie die Definition des Signal-Objekts.

093D_{hex} (USER FAIL)

- Ursache: Sie haben im Signal-Objekt eine ungültige Bitnummer angegeben.
- Abhilfe: Definieren Sie das Signal-Objekt richtig.
- Add_Error_Info: Ungültige Bitnummer

093E_{hex} (USER FAIL)

- Ursache: Sie haben die maximal zulässige Anzahl der Write-Parameter überschritten.
- Abhilfe: Prüfen Sie den Dienstaufruf.

093F_{hex} (USER FAIL)

- Bedeutung: Die maximal zulässige Anzahl der Read-Parameter wurde überschritten.
- Ursache: Sie haben zu viele Parameter oder falsche Parameter angegeben.
- Abhilfe: Prüfen Sie den Dienstaufruf.

0940_{hex} (USER FAIL)

- Bedeutung: Bei residenten Actions dürfen keine Write-Parameter-Adressen angegeben werden.
- Ursache: Sie haben für eine residente Action Write-Parameter-Adressen angegeben.
- Abhilfe: Prüfen Sie den Dienstaufruf.

0941_{hex}

- Bedeutung: Die Anschaltbaugruppe konnte die Boot-Sequenz nicht fehlerfrei abarbeiten.
- Ursache: Fehler in einem Dienstaufruf
- Abhilfe: Prüfen Sie den Aufruf des im Parameter *Add_Error_Info* angegebenen Dienstes.
- Add_Error_Info: Code des Dienstes, bei dem der Fehler aufgetreten ist

0942_{hex} (USER FAIL)

- Bedeutung: Die Anschaltbaugruppe konnte den Dienst „Clear_Parameterization_Memory“ (0159_{hex}) nicht bearbeiten.
- Ursache: Sie haben für den Parameter *Action_Flag* einen falschen Wert angegeben.
- Abhilfe: Prüfen Sie den Aufruf dieses Dienstes.
- Add_Error_Info: Falscher Parameterwert

0943_{hex} (USER FAIL)

- Bedeutung: Die Anschaltbaugruppe konnte den Dienst „Clear_Parameterization_Memory“ (0159_{hex}) nicht fehlerfrei bearbeiten.
- Ursache: Sie haben über den Parameter *Area_Flag* ein falsches Adressfeld angegeben.
- Abhilfe: Prüfen Sie den Aufruf dieses Dienstes.
- Add_Error_Info: Falscher Parameterwert

0944_{hex} (USER FAIL)

- Bedeutung:** Die Anschaltbaugruppe konnte den Dienst „Clear_Parameterization_Memory“ (0159_{hex}) nicht bearbeiten.
- Ursache:** Sie haben ein nicht definiertes *Action_Flag* angegeben.
- Abhilfe:** Prüfen Sie den Aufruf dieses Dienstes.

0945_{hex} (USER FAIL)

- Bedeutung:** Die Anschaltbaugruppe konnte den Dienst „Clear_Parameterization_Memory“ (0159_{hex}) nicht bearbeiten. Der Zugriff auf den Parametrierungsspeicher wurde abgewiesen.
- Ursache:** Der Parametrierungsspeicher ist nicht vorhanden oder nicht richtig gesteckt.
- Abhilfe:** Prüfen Sie den Sitz des Parametrierungsspeichers.

0946_{hex} (USER FAIL)

- Bedeutung:** Der Zugriff auf den Parametrierungsspeicher wurde abgewiesen.
- Ursache:** Der Parametrierungsspeicher ist schreibgeschützt.
- Abhilfe:** Deaktivieren Sie den Schreibschutz des Parametrierungsspeichers und versuchen Sie es erneut.

0947_{hex} (USER FAIL)

- Ursache:** Sie haben ein Action-Objekt nicht richtig definiert.
- Abhilfe:** Prüfen Sie die Definitionen der Action-Objekte.
- Add_Error_Info:** Action-Index des fehlerhaften Action-Objekts

0948_{hex} (USER FAIL)

- Ursache:** Sie haben ein Signal-Objekt nicht richtig definiert.
- Abhilfe:** Prüfen Sie die Definitionen der Signal-Objekte.
- Add_Error_Info:** Signal-Index des fehlerhaften Signal-Objekts

0949_{hex} (USER FAIL)

Ursache: Sie haben ein Event-Objekt nicht richtig definiert.
Abhilfe: Prüfen Sie die Definitionen der Event-Objekte.
Add_Error_Info: Event-Index des fehlerhaften Event-Objekts

0950_{hex}

Bedeutung: Die Anschaltbaugruppe kann die Informationen nicht drucken.
Ursache: Die Diagnose-Schnittstelle der Anschaltbaugruppe ist nicht im Druckmodus.
Abhilfe: Schalten Sie die Diagnose-Schnittstelle mit dem Dienst „Set_Value“ (0750_{hex}) in den Druckmodus.

0951_{hex}

Bedeutung: Die Anschaltbaugruppe kann die Informationen nicht drucken.
Ursache: Der Pufferspeicher (Buffer) der Diagnose-Schnittstelle ist voll.
Abhilfe: Warten Sie einen Moment und versuchen Sie es erneut.

0952_{hex} (USER FAIL)

Bedeutung: Sie haben bei der Eingabe für den Parameter *Variable_ID* einen nicht definierten Wert benutzt.
Abhilfe: Prüfen Sie die Definitionen der *Variable_ID* und versuchen Sie es erneut.

0953_{hex} (USER FAIL)

Bedeutung: Die Anschaltbaugruppe konnte den zuletzt aufgerufenen Dienst nicht bearbeiten.
Ursache: Sie haben einen unzulässigen Parameter angegeben.
Abhilfe: Prüfen Sie den Aufruf dieses Dienstes.

0954_{hex} (USER FAIL)

- Bedeutung:** Die Anschaltbaugruppe konnte den zuletzt aufgerufenen Dienst für die Diagnose-Schnittstelle nicht bearbeiten.
- Ursache:** Sie haben für den Parameter *Parameter_Count* einen falschen Wert angegeben.
- Abhilfe:** Prüfen Sie den Aufruf dieses Dienstes. Der Parameter *Parameter_Count* muss der Anzahl der nachfolgenden Worte entsprechen.

0955_{hex}

- Bedeutung:** Die Nachricht wurde nicht ausgelesen bzw. gesendet.
- Ursache:** Die Nachricht überschreitet die maximal zulässige Mailbox-Größe von 1024 Byte der Diagnose-Schnittstelle.
- Abhilfe:** Falls der Fehler-Code nach einem Request auftritt, können Sie die Nachricht in mehreren Schritten auslesen. Einige Dienste lassen sich zum Auslesen größerer Datensätze mehrmals aufrufen. Geben Sie beim zweiten Aufruf einen Adress-Offset an.

0956_{hex}

- Bedeutung:** Die Nachricht wurde nicht ausgelesen bzw. gesendet.
- Ursache:** Beim Senden von PCP-Diensten über das SSGI wurde in eine SSGI-Box ein PCP-Dienst mit einer Kommunikationsreferenz geschrieben, welche nicht der SSGI-Box entspricht.
- Abhilfe:** Schreiben Sie einen PCP-Dienst nur in eine SSGI-Box, deren Nummer der Kommunikationsreferenz entspricht.

0957_{hex}

- Bedeutung:** Alle Remote-PCP-Dienste auf der abgebauten Verbindung sind abgebrochen worden.
- Ursache:** Die Verbindung ist durch einen PMS- oder einen PNM7-Abort auf einer Kommunikationsreferenz abgebrochen worden.
- Abhilfe:** Prüfen Sie Ihr Anwendungsprogramm und starten Sie es erneut.

0960_{hex}

- Bedeutung: Der verwendete Parameter *Variablen_ID* für das Diagnose-Display ist ungültig.
- Abhilfe: Setzen Sie den richtigen Wert für den Parameter *Variablen_ID* ein.
- Add_Error_Info: Ungültiger Parameter *Variablen_ID*

0970_{hex}

- Bedeutung: Bei der Kommunikation mit der Anschaltbaugruppe (z. B. über IB-Loader) ist ein Fehler im IEC-61131-Laufzeitsystem der Anschaltbaugruppe aufgetreten.
- Abhilfe: Schicken Sie die Dienste langsamer ab, d. h. bauen Sie zwischen den Diensten eine Wartezeit ein.

09D0_{hex}

- Bedeutung: Sie verwenden einen Parametrierungsspeichertyp, der von Ihrer Firmware nicht unterstützt wird.
- Abhilfe: Tauschen Sie den vorhandenen gegen einen korrekten Parametrierungsspeicher.

09D1_{hex}

- Bedeutung: Der von Ihnen verwendete Parametrierungsspeicher ist defekt.
- Abhilfe: Tauschen Sie den vorhandenen gegen einen intakten Parametrierungsspeicher.

09D2_{hex}

- Bedeutung: Sie verwenden einen Parametrierungsspeichertyp, der von Ihrer Firmware nicht unterstützt wird.
- Abhilfe: Tauschen Sie den vorhandenen gegen einen korrekten Parametrierungsspeicher.

09D3_{hex}

Bedeutung: Die Anzahl der geöffneten Dateien auf dem Parametrierungsspeicher kann nicht mehr verwaltet werden.

09D4_{hex}

Bedeutung: Sie haben versucht eine Datei auf dem Parametrierungsspeicher zu speichern, deren Name zuviele Zeichen hat.

Abhilfe: Benennen Sie die Datei um.

09D5_{hex}

Bedeutung: Sie haben die Dateizugriffsrechte auf eine geöffnete Datei im Parametrierungsspeicher der Anschaltbaugruppe verletzt.

Ursache: Sie haben z. B. versucht in eine Datei zu schreiben, die nur zum Lesen geöffnet war.

Abhilfe: Beachten Sie die Zugriffsrechte beim Umgang mit Dateien.

09D6_{hex}

Bedeutung: Auf dem Parametrierungsspeicher der Anschaltbaugruppe ist nicht mehr genügend Speicherplatz vorhanden.

Abhilfe:

- Löschen Sie einige Dateien.
- Sichern Sie die Dateien und formatieren Sie den Parametrierungsspeicher neu.

09D8_{hex}

Bedeutung: Auf dem Parametrierungsspeicher ist zu wenig Speicherplatz vorhanden.

Ursache: Z. B. kann der Parametrierungsspeicher zu stark fragmentiert sein.

Abhilfe: Sichern Sie die Dateien und formatieren Sie den Parametrierungsspeicher neu.

09D9_{hex}

Bedeutung: Die gewünschte Datei ist auf dem Parametrierungsspeicher der Anschaltbaugruppe nicht vorhanden oder kann nicht gefunden werden.

09DA_{hex}

Bedeutung: Die gewünschte Datei konnte nicht vom Parametrierungsspeicher der Anschaltbaugruppe gelesen werden.

Ursache: Während des Lesens der Datei ist ein Fehler aufgetreten.

09DB_{hex}

Bedeutung: Eine Datei konnte nicht auf dem Parametrierungsspeicher der Anschaltbaugruppe geschrieben werden.

Ursache: Während des Schreibens der Datei ist ein Fehler aufgetreten.

09DC_{hex}

Bedeutung: Auf die gewünschte Datei vom Parametrierungsspeicher der Anschaltbaugruppe konnte nicht zugegriffen werden.

Ursache: Während des Dateizugriffs mit dem Dienst „File_Seek“ (015F_{hex}) ist ein Fehler aufgetreten. Der Dateizeiger konnte nicht richtig positioniert werden.

09DD_{hex}

Bedeutung: Eine Datei auf dem Parametrierungsspeicher der Anschaltbaugruppe konnte nicht geschlossen werden.

Ursache: Während des Schließens der Datei ist ein Fehler aufgetreten.

09DE_{hex}

Bedeutung: Eine Datei auf dem Parametrierungsspeicher der Anschaltbaugruppe konnte nicht geöffnet werden.

Ursache: Während des Öffnens der Datei ist ein Fehler aufgetreten da:

- kein Parametrierungsspeicher gesteckt ist,
- die Datei defekt ist,
- die Datei nicht vorhanden ist oder
- der Parametrierungsspeicher schreibgeschützt oder defekt ist.

09DF_{hex}

Bedeutung: Die gewünschte Datei vom Parametrierungsspeicher der Anschaltbaugruppe konnte nicht gelöscht werden.

Ursache: Während des Löschens der Datei ist ein Fehler aufgetreten.

09F0_{hex}

Bedeutung: Auf dem Parametrierungsspeicher der Anschaltbaugruppe kann nur eine maximale Anzahl Dateien verwaltet werden. Diese Anzahl wurde überschritten.

Abhilfe:

- Löschen Sie einige Dateien.
- Sichern Sie die Dateien und formatieren Sie den Parametrierungsspeicher neu.

09F1_{hex}

Bedeutung: Die auf dem Parametrierungsspeicher der Anschaltbaugruppe vorhandene Dateistruktur kann nicht gelesen werden.

Ursache: Der Parametrierungsspeicher wurde mit einer älteren Firmware-Version geschrieben.

Abhilfe: Formatieren Sie den Parametrierungsspeicher neu.

09F2_{hex}

Bedeutung: Auf dem Parametrierungsspeicher der Anschaltbaugruppe kann eine vorhandene Datei nicht bearbeitet werden, da der File-Handle falsch oder nicht gültig ist.

Abhilfe: Geben Sie den richtigen File-Handle für die Datei ein, die bearbeitet werden soll.

09FC_{hex}

Bedeutung: Auf dem Parametrierungsspeicher kann nicht geschrieben werden, dadurch wurde ein Timeout ausgelöst.

Ursache: Der Parametrierungsspeicher ist nicht programmierbar.

Abhilfe: Tauschen Sie den Parametrierungsspeicher aus.

0A02_{hex} (USER FAIL)

Bedeutung:	Die Anschaltbaugruppe konnte den zuletzt aufgerufenen Dienst nicht bearbeiten.
Ursache:	Sie haben einen Dienst abgesetzt, der im aktuellen Zustand (Status) der Anschaltbaugruppe nicht zulässig ist. Beispielsweise dürfen Sie nicht den Dienst „Start_Data_Transfer“ (0701 _{hex}) absetzen, wenn die Anschaltbaugruppe im Zustand <i>Ready</i> ist. Zum Starten des Datentransfers muss die Anschaltbaugruppe im Zustand <i>Active</i> sein.
Abhilfe:	Setzen Sie die Anschaltbaugruppe vor Aufruf des gewünschten Dienstes in den erforderlichen Zustand.
Add_Error_Info:	Aktueller Zustand der Anschaltbaugruppe:
	0001 _{hex} Ready (Anzeige: <i>RDY</i>)
	0002 _{hex} Parametrierung nicht abgeschlossen (Parameterization not Ready)
	0004 _{hex} Laden der Konfiguration (Loading CFG)
	0008 _{hex} Laden des Prozessdatenbeschreibungsliste (Loading PDDL)
	0010 _{hex} Laden der Prozessdatenreferenzliste (Loading PDRL)
	0020 _{hex} Parametrierung abgeschlossen (Parameterization Ready)
	0040 _{hex} Die Anschaltbaugruppe fährt sporadisch ID-Zyklen (Anzeige: <i>ACTV</i>).
	0080 _{hex} Die Anschaltbaugruppe fährt Datenzyklen (Anzeige: <i>RUN</i>).
	0100 _{hex} Bus-Fail
	0200 _{hex} Die Anschaltbaugruppe sucht den Fehler (Anzeige: <i>Look For Fail</i>)

0A03_{hex}

Bedeutung:	Auf der Anschaltbaugruppe ist zu wenig Arbeitsspeicher frei.
Ursache:	Z. B. kann der Arbeitsspeicher zu stark fragmentiert sein.
Abhilfe:	Führen Sie mit dem Dienst „Reset_Controller_Board“ (0956 _{hex}) einen Warmstart der Anschaltbaugruppe durch und versuchen Sie es erneut.

0A04_{hex} (USER FAIL)

Bedeutung:	Die Anschaltbaugruppe konnte den zuletzt aufgerufenen Dienst nicht bearbeiten.
Ursache:	Sie haben widersprüchliche Parameter angegeben.
Abhilfe:	Prüfen Sie den Aufruf dieses Dienstes.
Add_Error_Info:	Nummer des widersprüchlichen Parameters

0A05_{hex} (USER FAIL)

Bedeutung:	Die Anschaltbaugruppe konnte den zuletzt aufgerufenen Dienst nicht bearbeiten.
Ursache:	Sie haben einen unzulässigen Parameter angegeben.
Abhilfe:	Prüfen Sie den Aufruf dieses Dienstes.
Add_Error_Info:	Nummer des unzulässigen Parameters

0A06_{hex} (USER FAIL)

Bedeutung:	Die Anschaltbaugruppe konnte den Zugriff nicht durchführen.
Ursache:	Sie haben einen Zugriff auf ein Objekt versucht. Dieser Zugriff wird vom System-Management für das Objekt nicht unterstützt, z. B. ein Schreibzugriff auf ein Objekt, für das nur Lesezugriffe erlaubt sind.

0A07_{hex} (USER FAIL)

Bedeutung:	Die Anschaltbaugruppe konnte den Zugriff nicht durchführen.
Ursache:	Sie haben einen Zugriff auf ein Objekt ausgeführt, das nicht existiert.

0A08_{hex} (USER FAIL)

Bedeutung: Sie haben einen Dienst aufgerufen, der mit einem anderen, zuvor aufgerufenen Dienst in Bearbeitungskonflikt gerät.

Ursache: Die Firmware kann nicht zwei Dienste gleichzeitig bearbeiten.

Abhilfe: Warten Sie die Bearbeitung des zuvor aufgerufenen Dienstes ab, und versuchen Sie es danach erneut.

Add_Error_Info: Code des zuvor aufgerufenen Dienstes

0A10_{hex} (USER FAIL)

Bedeutung: Die Anschaltbaugruppe kann den Dienst „Set_Value“ (0750_{hex}) nicht bearbeiten.

Ursache: Sie haben für den Parameter *Variable_ID* einen falschen Wert angegeben.

Add_Error_Info: Nummer der falschen *Variable_ID*

0A11_{hex} (USER FAIL)

Bedeutung: Die Anschaltbaugruppe kann den Dienst „Set_Value“ (0750_{hex}) nicht bearbeiten.

Ursache: Sie haben bei der Angabe eines Parameters den zulässigen Wertebereich überschritten.

Abhilfe: Prüfen Sie die Parameter dieses Dienstes.

0A12_{hex} (USER FAIL)

Ursache: Der Parameter *Device_Level* überschreitet den Wertebereich.

Abhilfe: Prüfen Sie den Parameter *Device_Level*. Zulässig sind Werte von 00_{hex} bis 0F_{hex} (entspricht 0 bis 15_{dez}).

Add_Error_Info: Position im Konfigurationsrahmen

0A14_{hex} (USER FAIL)

- Ursache: Sie haben für den Parameter *Error_Char_Flag* einen unzulässigen Wert angegeben.
- Abhilfe: Prüfen Sie den Parameter *Error_Char_Flag*. Zulässig sind nur die beiden Werte 0000_{hex} und 0001_{hex}.
- Add_Error_Info: Position im Konfigurationsrahmen

0A15_{hex} (USER FAIL)

- Ursache: Sie haben für den Parameter *Position* einen unzulässigen Wert angegeben.
- Abhilfe: Prüfen Sie den Parameter *Position*. Zulässig sind Werte von 00_{hex} bis 3F_{hex} (0_{dez} bis 63_{dez}).
- Add_Error_Info: Position im Konfigurationsrahmen

0A16_{hex} (USER FAIL)

- Ursache: Sie haben für die INTERBUS-Teilnehmernummer (Segment . Position) einen unzulässigen Wert angegeben.
- Abhilfe: Prüfen Sie die INTERBUS-Teilnehmernummern:
Zulässig für *Segment* sind Werte von 01_{hex} bis FF_{hex} (1_{dez} bis 255_{dez}).
Zulässig für *Position* sind Werte von 00_{hex} bis 3F_{hex} (0_{dez} bis 63_{dez}).
- Add_Error_Info: Unzulässige INTERBUS-Teilnehmernummer

0A17_{hex} (USER FAIL)

- Ursache: Sie haben einen unbekanntem ID-Code vergeben.
- Abhilfe: Prüfen Sie die Parameter des Konfigurationsrahmens.
- Add_Error_Info: Position im Konfigurationsrahmen

0A18_{hex} (USER FAIL)

Ursache: Sie haben im Parameter *Used_Attributes* ein unzulässiges Bit aktiviert.
Abhilfe: Prüfen Sie den Parameter *Used_Attributes* des entsprechenden Dienstes.
Add_Error_Info: Fehlerhafter Parameter *Used_Attributes*

0A19_{hex} (USER FAIL)

Bedeutung: Sie haben beim Zugriff auf einen Konfigurationsrahmen den maximalen Busaufbau überschritten.
Abhilfe: Prüfen Sie den letzten Zugriff auf den Konfigurationsrahmen.
Add_Error_Info: Anzahl der INTERBUS-Teilnehmer

0A1A_{hex} (USER FAIL)

Bedeutung: Die angegebene *Frame_Reference* existiert nicht oder es konnte keine *Frame_Reference* zugeordnet werden.
Abhilfe: Prüfen Sie den Parameter *Frame_Reference*.
Add_Error_Info: Fehlerhafte *Frame_Reference* (falls vorgegeben)

0A1B_{hex} (USER FAIL)

Bedeutung: Der angegebene Konfigurationsrahmen konnte nicht gelöscht werden.
Ursache: Der angegebene Konfigurationsrahmen ist zur Zeit eingeschaltet. Es ist nicht möglich, den aktuell eingeschalteten Konfigurationsrahmen zu löschen.
Abhilfe: Falls Sie wirklich den aktuell eingeschalteten Konfigurationsrahmen löschen möchten, müssen Sie ihn zuvor mit dem Dienst „Deactivate_Configuration“ (0712_{hex}) deaktivieren.
Add_Error_Info: Nummer des Konfigurationsrahmens

0A1C_{hex} (USER FAIL)

Ursache: Sie haben die zulässige Anzahl der vorgegebenen oder angeschlossenen INTERBUS-Teilnehmer überschritten. Maximal zulässig sind 255 INTERBUS-Teilnehmer.
Add_Error_Info: Anzahl der vorgegebenen oder angeschlossenen INTERBUS-Teilnehmer

0A1D_{hex} (USER FAIL)

- Bedeutung:** Der angegebene Konfigurationsrahmen konnte nicht überschrieben werden.
- Ursache:** Sie haben versucht einen bestehenden Konfigurationsrahmen mit neuen Konfigurationsdaten zu überschreiben. Dabei stimmte die Größe (Anzahl der INTERBUS-Teilnehmer) nicht mit dem bestehenden Rahmen überein. Wenn Sie einen Konfigurationsrahmen anderer Größe erstellen wollen, müssen Sie diesen unter einer neuen, noch unbenutzten *Frame_Reference* anlegen.
- Abhilfe:** Vergleichen Sie den bestehenden Konfigurationsrahmen mit den neuen Konfigurationsdaten.

0A1E_{hex} (USER FAIL)

- Bedeutung:** Die neue Extension konnte dem Konfigurationsrahmen nicht zugewiesen werden.
- Ursache:** Sie haben versucht, einem bestehenden Konfigurationsrahmen eine neue Extension zuzuweisen, die größer ist als der im Konfigurationsrahmen für die Extension vorgesehene Platz.
- Abhilfe:** Passen Sie die Größe der neuen Extension dem vorgesehenen Platz an.

0A1F_{hex} (USER FAIL)

- Bedeutung:** Der angegebene Konfigurationsrahmen konnte nicht deaktiviert werden.
- Ursache:** Der angegebene Konfigurationsrahmen war bereits inaktiv.
- Add_Error_Info:** Parameter *Frame_Reference*

0A20_{hex} (USER FAIL)

- Bedeutung:** Der physikalisch erste INTERBUS-Teilnehmer im INTERBUS-Netzwerk wurde nicht abgeschaltet.
- Ursache:** Der physikalisch erste INTERBUS-Teilnehmer ist nicht abschaltbar.

0A21_{hex} (USER FAIL)

Ursache: Sie haben eine INTERBUS-Teilnehmernummer mehrfach vergeben.
Abhilfe: Prüfen Sie die INTERBUS-Teilnehmernummern im Konfigurationsrahmen.
Add_Error_Info: Position im Konfigurationsrahmen

0A22_{hex} (USER FAIL)

Ursache: Sie haben widersprüchliche INTERBUS-Teilnehmernummern vergeben.
Abhilfe: Prüfen Sie die INTERBUS-Teilnehmernummern im Konfigurationsrahmen.
Add_Error_Info: Position im Konfigurationsrahmen

0A23_{hex} (USER FAIL)

Ursache: Sie haben widersprüchliche INTERBUS-Teilnehmerebenen angegeben.
Abhilfe: Prüfen Sie den Konfigurationsrahmen.
Add_Error_Info: Position im Konfigurationsrahmen

0A24_{hex} (USER FAIL)

Bedeutung: Innerhalb eines Lokalbus-Segments haben die angeschlossenen Teilnehmer unterschiedliche Betriebsstati.
Ursache: Beim Dienst „Control_Active_Configuration“ (0713_{hex}) wurden nicht alle Teilnehmer an- oder ausgeschaltet.
Abhilfe: Schalten Sie alle Teilnehmer in dem Lokalbus-Segment ein oder aus.
Add_Error_Info: Position im Konfigurationsrahmen

0A25_{hex} (USER FAIL)

Bedeutung: Die Anschaltbaugruppe hat die Gruppensegmentdefinition, die von Ihnen vorgegeben wurde, nicht akzeptiert.
Ursache: Sie haben für INTERBUS-Teilnehmer eines Bussegments unterschiedliche Gruppennummern vergeben.
Abhilfe: Vergeben Sie an alle INTERBUS-Teilnehmer eines Bussegments grundsätzlich die gleiche Gruppennummer.
Add_Error_Info: Position im Konfigurationsrahmen

0A26_{hex} (USER FAIL)

Bedeutung:	Die Anschaltbaugruppe hat die Definition einer alternativen Gruppe nicht akzeptiert.
Ursache:	Sie haben unzulässige alternative Gruppennummern vergeben.
Add_Error_Info:	Position im Konfigurationsrahmen

0A27_{hex}

Bedeutung:	Die Anschaltbaugruppe hat die Definition einer alternativen Gruppe nicht akzeptiert.
Ursache:	Der erste Teilnehmer in der alternativen Gruppe ist keine Busklemme.
Add_Error_Info:	Position im Konfigurationsrahmen

0A28_{hex} (USER FAIL)

Bedeutung:	Die Anschaltbaugruppe konnte die angegebenen Gruppen nicht aktivieren.
Ursache:	Sie haben versucht, alternative Gruppen gleichzeitig aktiv zu schalten.
Add_Error_Info:	Position im Konfigurationsrahmen

0A29_{hex}

Bedeutung:	Beim Ein- oder Ausschalten von INTERBUS-Teilnehmern kam es zu einem Konflikt der gegenseitigen Gruppenabhängigkeiten.
Ursache:	Sie haben mit dem Dienst „Control_Active_Configuration“ (0713 _{hex}) voneinander abhängige Gruppen widersprüchlich geschaltet.
Add_Error_Info:	Position im Konfigurationsrahmen

0A2A_{hex}

Bedeutung:	Konflikt durch voneinander abhängige INTERBUS-Teilnehmer (aktiv - inaktiv).
Ursache:	Sie haben mit dem Dienst „Control_Active_Configuration“ (0713 _{hex}) voneinander abhängige INTERBUS-Teilnehmer widersprüchlich geschaltet.
Add_Error_Info:	Position im Konfigurationsrahmen

0A2B_{hex} (USER FAIL)

Bedeutung:	Zustandskonflikt innerhalb einer Gruppe
Ursache:	Sie haben INTERBUS-Teilnehmern, die zu einer Gruppe gehören, unterschiedliche Betriebsstati zugewiesen.
Abhilfe:	Weisen Sie den INTERBUS-Teilnehmern einer Gruppe grundsätzlich die gleichen Betriebsstati zu.
Add_Error_Info:	Position im Konfigurationsrahmen

0A2C_{hex} (USER FAIL)

Ursache:	Sie haben eine INTERBUS-Teilnehmernummer angegeben, die nicht existiert.
Add_Error_Info:	Nicht existente INTERBUS-Teilnehmernummer

0A2D_{hex} (USER FAIL)

Bedeutung:	Zu viele PCP-Teilnehmer.
Ursache:	<ul style="list-style-type: none">– Sie haben mehr PCP-Teilnehmer angeschlossen als zulässig sind.– Sie haben mehr PCP-Teilnehmer projiziert als zulässig sind.
Abhilfe:	Verringern Sie die Anzahl der angeschlossenen oder projizierten PCP-Teilnehmer. Maximal sind 126 PCP-Teilnehmer zulässig, abhängig von der eingesetzten Anschaltbaugruppe und der Firmware-Version.

0A2E_{hex} (USER FAIL)

Bedeutung:	Die zulässige Anzahl der internen indirekten Adresslisten-Einträge wurde überschritten. Die Speichergrenze in der Firmware wurde erreicht.
Ursache:	Sie haben sehr viele Module, die nur ein Byte oder ein Nibble Adressraum im Datenring beanspruchen.
Abhilfe:	<ul style="list-style-type: none">– Verringern Sie die Anzahl der Module, die nur ein Byte oder ein Nibble Adressraum beanspruchen. Maximal zulässig sind 384 interne indirekte Adresslisten-Einträge.– Ordnen Sie die Module so an, dass die Teilnehmer, die weniger als 1 Wort Adressraum benötigen, hintereinanderliegen.

0A2F_{hex} (USER FAIL)

Bedeutung:	Die Anschaltbaugruppe konnte den Dienst „Initiate_Load_Configuration“ (0306 _{hex}) nicht ausführen.
Ursache:	Sie haben die Anzahl der INTERBUS-Teilnehmer mit 0 angegeben.
Abhilfe:	Geben Sie die korrekte INTERBUS-Teilnehmeranzahl an.

0A30_{hex} (USER FAIL)

Bedeutung:	Fehlerhafter Eintrag in der Prozessdatenbeschreibungsliste (PDDL)
Abhilfe:	Prüfen Sie die Prozessdatenbeschreibungsliste.

0A31_{hex} (USER FAIL)

Bedeutung:	Die Anschaltbaugruppe konnte den Dienst „Put_Process_Data_Description_List“ (0321 _{hex}) oder den Dienst „Get_Process_Data_Description_List“ (0323 _{hex}) nicht bearbeiten.
Ursache:	Sie haben eine INTERBUS-Teilnehmernummer verwendet, die nicht existiert.
Add_Error_Info:	Nicht existente INTERBUS-Teilnehmernummer im Format [RRLL]. RR = Fernbus-Segmentnummer LL = Lokalbus-Segmentnummer

0A32_{hex} (USER FAIL)

Bedeutung:	Die Anschaltbaugruppe konnte den Dienst „Put_Process_Data_Description_List“ (0321 _{hex}) nicht bearbeiten.
Ursache:	Sie haben in einer Prozessdatenbeschreibung für den Parameter <i>PDD_Index</i> einen unzulässigen Wert angegeben.
Abhilfe:	Vergeben Sie für den Parameter <i>PDD_Index</i> Werte im Bereich von 0000 _{hex} bis 7FFF _{hex} , mit Ausnahme der Werte 6010 _{hex} und 6011 _{hex} , die für Default-Prozessdatenbeschreibungen reserviert sind. Jeder <i>PDD_Index</i> darf nur einmal vergeben werden.
Add_Error_Info:	Index der Prozessdatenbeschreibung

0A33_{hex} (USER FAIL)

Bedeutung:	Für einen INTERBUS-Teilnehmer wurde eine falsche Datenrichtung angegeben.
Ursache:	<ul style="list-style-type: none">– Sie haben für einen INTERBUS-Teilnehmer, der nur Eingangs-Prozessdaten besitzt, eine Prozessdatenbeschreibung für Ausgangs-Prozessdaten definiert.– Sie haben für einen INTERBUS-Teilnehmer, der nur Ausgangs-Prozessdaten besitzt, eine Prozessdatenbeschreibung für Eingangs-Prozessdaten definiert.
Abhilfe:	Legen Sie mit dem Parameter <i>Data_Direction</i> die Datenrichtung der Prozessdatenbeschreibungen folgendermaßen fest: 0C _{hex} für Eingangs-Prozessdaten (IN-PDD) 0D _{hex} für Ausgangs-Prozessdaten (OUT-PDD)
Add_Error_Info:	Index der Prozessdatenbeschreibung

0A34_{hex} (USER FAIL)

Ursache:	Sie haben bei der Beschreibung eines Prozessdatums den internen Adressraum eines INTERBUS-Teilnehmers überschritten.
Add_Error_Info:	Index des Prozessdaten-Objektes

0A35_{hex} (USER FAIL)

Ursache:	Sie haben bei der Beschreibung eines Prozessdatums einen Datentyp und eine Datenlänge angegeben, die nicht zueinander passen.
Abhilfe:	Prüfen Sie die Prozessdatenbeschreibung.
Add_Error_Info:	Index der Prozessdatenbeschreibung

0A36_{hex} (USER FAIL)

Bedeutung:	Fehler bei der Definition eines Prozessdatums
Ursache:	Sie haben mit dem Dienst „Put_Process_Data_Description_List“ (0321 _{hex}) ein Bitstring-Prozessdatum definiert, das über eine Byte-Grenze hinausgeht.
Abhilfe:	Definieren Sie nur Bitstring-Prozessdaten, die die Byte-Grenzen nicht überschreiten und somit maximal 8 Bit lang sind.
Add_Error_Info:	Index der Prozessdatenbeschreibung

0A40_{hex} (USER FAIL)

Bedeutung:	Ein Eintrag in der Prozessdatenreferenzliste (PDRL) ist fehlerhaft.
Abhilfe:	Prüfen Sie die Prozessdatenreferenzliste.
Add_Error_Info:	<i>PDRL_Index</i> des falschen PDRL-Eintrages

0A41_{hex} (USER FAIL)

Bedeutung:	Fehlerhafter Eintrag in der Prozessdatenreferenzliste.
Ursache:	Sie haben als Ziel kein Ausgangs-Prozessdatum angegeben.
Abhilfe:	Geben Sie als Ziel ein Ausgangs-Prozessdatum an.
Add_Error_Info:	<i>PDRL_Index</i> des falschen PDRL-Eintrages

0A42_{hex} (USER FAIL)

Bedeutung:	Fehlerhafter Eintrag in der Prozessdatenreferenzliste.
Ursache:	Sie haben als Quelle kein Eingangs-Prozessdatum angegeben.
Abhilfe:	Geben Sie als Quelle ein Eingangs-Prozessdatum an.
Add_Error_Info:	<i>PDRL_Index</i> des falschen PDRL-Eintrages

0A43_{hex} (USER FAIL)

Bedeutung: Ein Eintrag in der Prozessdatenreferenzliste ist fehlerhaft.
Ursache: Sie haben eine INTERBUS-Teilnehmernummer angegeben, die nicht existiert.
Abhilfe: Prüfen Sie die Prozessdatenreferenzliste.
Add_Error_Info: *PDRL_Index* des falschen PDRL-Eintrages

0A44_{hex} (USER FAIL)

Bedeutung: Ein Eintrag in der Prozessdatenreferenzliste ist fehlerhaft.
Ursache: Sie haben für ein Prozessdatum eine Datenkonsistenz angegeben, die nicht zu dessen Länge passt.
Abhilfe: Prüfen Sie die Prozessdatenreferenzliste.
Add_Error_Info: *PDRL_Index* des falschen PDRL-Eintrages

0A45_{hex} (USER FAIL)

Bedeutung: Ein Eintrag in der Prozessdatenreferenzliste ist fehlerhaft.
Ursache: Sie haben für den Parameter *PDD_Index* einen Wert angegeben, der nicht existiert.
Abhilfe: Prüfen Sie die Prozessdatenreferenzliste.
Add_Error_Info: *PDRL_Index* des falschen PDRL-Eintrages

0A46_{hex} (USER FAIL)

Bedeutung: Adressierungsfehler
Ursache: Sie haben beim Dienst „Compact_Load_Process_Data_Reference_List“ (0328_{hex}) oder „Load_Process_Data_Reference_List“ (0325_{hex}) für den Parameter *Address_Direction* einen falschen Wert angegeben.
Abhilfe: Vergeben Sie
1000_{hex} für eine Eingangsadressliste oder
2000_{hex} für eine Ausgangsadressliste.
Add_Error_Info: *PDRL_Index* des falschen PDRL-Eintrages

0A47_{hex} (USER FAIL)

Bedeutung:	Ein Eintrag in der Prozessdatenreferenzliste ist fehlerhaft.
Ursache:	Sie haben für den Parameter <i>PDRL_Index</i> einen fehlerhaften oder unbekanntenen Wert angegeben.
Abhilfe:	Prüfen Sie die Prozessdatenreferenzliste.
Add_Error_Info:	<i>PDRL_Index</i> des falschen PDRL-Eintrages

0A48_{hex} (USER FAIL)

Bedeutung:	Durch die Definition eines Durchgangs können Sie ein Eingangs-Prozessdatum direkt auf ein Ausgangs-Prozessdatum gleicher Prozessdatenlänge abbilden. Einer dieser Einträge in der Prozessdatenreferenzliste ist nicht korrekt.
Ursache:	Sie haben versucht: <ul style="list-style-type: none">– ein Eingangs-Prozessdatum auf ein anderes Eingangs-Prozessdatum abzubilden,– ein Ausgangs-Prozessdatum auf ein anderes Ausgangs-Prozessdatum abzubilden oder– ein Eingangs-Prozessdatum auf ein Ausgangs-Prozessdatum mit unterschiedlicher Prozessdatenlänge abzubilden.
Abhilfe:	Prüfen Sie die Durchgangsdefinitionen.
Add_Error_Info:	<i>PDRL_Index</i> des falschen PDRL-Eintrages

0A49_{hex} (USER FAIL)

Bedeutung:	Die Anschaltbaugruppe kann den angeschlossenen Busaufbau nicht in Betrieb nehmen.
Ursache:	Sie haben beim Dienst „Compact_Load_Process_Data_Reference_List“ (0328 _{hex}) mit dem Parameter <i>Entry_Count</i> eine andere INTERBUS-Teilnehmeranzahl angegeben, als im angeschlossenen Busaufbau vorhanden ist.
Abhilfe:	Prüfen Sie mit dem Dienst „Compact_Read_Process_Data_Reference_List“ (0329 _{hex}) die Zuordnung der Prozessdaten und vergleichen Sie diese mit der INTERBUS-Teilnehmeranzahl Ihres angeschlossenen Busaufbaus.
Add_Error_Info:	Kennung, ob IN- oder OUT-Liste

0A4A_{hex} (USER FAIL)

- Bedeutung:** Die Anschaltbaugruppe konnte die Prozessdatenreferenzliste (PDRL) nicht auslesen.
- Ursache:** Sie haben versucht, eine mit dem Dienst „Load_Process_Data_Reference_List“ (0325_{hex}) geladene PDRL mit dem Dienst „Compact_Read_Process_Data_Reference_List“ (0329_{hex}) auszulesen.
- Abhilfe:** Lesen Sie PDRLs, die mit dem Dienst „Load_Process_Data_Reference_List“ (0325_{hex}) geladen wurden, mit dem Dienst „Read_Process_Data_Reference_List“ (0327_{hex}) aus.

0A4B_{hex} (USER FAIL)

- Bedeutung:** Fehler bei der Zuweisung eines Prozessdatums
- Ursache:** Sie haben mit dem Dienst „Load_Process_Data_Reference_List“ (0325_{hex}) ein Bitstring-Prozessdatum so zugewiesen, dass es über eine Byte-Grenze hinausgeht.
- Abhilfe:** Verschieben Sie das Bitstring-Prozessdatum bis es die Byte-Grenze nicht überschreitet.
- Add_Error_Info:** Bit-Position des fehlerhaft zugewiesenen Bitstring-Prozessdatums im MPM.

0A4C_{hex} (USER FAIL)

- Bedeutung:** Fehler bei der Zuordnung eines Prozessdatums.
- Ursache:** Sie haben einem Prozessdatum mit einer Datenkonsistenz von 16, 32 oder 64 Bit eine ungerade MPM-Adresse zugeordnet. Dies ist aber nur für Prozessdaten mit einer Datenkonsistenz von 8 Bit erlaubt.
- Abhilfe:** Weisen Sie Prozessdaten mit einer Datenkonsistenz von 16, 32 oder 64 Bit grundsätzlich gerade MPM-Adressen zu.
- Add_Error_Info:** *PDRL_Index* des falschen PDRL-Eintrages

0A50_{hex} (USER FAIL)

Bedeutung:	Fehler bei der Zuordnung eines Prozessdatums.
Ursache:	Eine Ausgangs-Prozessdatenbeschreibung oder eine Eingangs-Hostadresse wurde mehrfach vergeben.
Abhilfe:	Weisen Sie jeder Ausgangs-Prozessdatenbeschreibung grundsätzlich nur eine Hostadresse und jeder Eingangs-Hostadresse nur eine Prozessdatenbeschreibung zu.
Add_Error_Info:	<i>PDRL_Index</i> des falschen PDRL-Eintrages

**Zusatzinfo für Systemkopplerkarten:**

Wenn ein großer Busaufbau (Systemkopplerkarte und Anzahl Teilnehmer > 225, alle mit 32 Bit DIO) mit dem Dienst *Create_Configuration* (0710_{hex}) eingelesen wird, dann werden die Default-E/A-Daten im Adressbereich 0000_{hex} bis 03FF_{hex} abgelegt. Dies bedeutet einen Konflikt mit der Systemkoppleradresse 0380_{hex}.

Abhilfe:	Legen Sie die Adresse des Systemkopplers auf eine Adresse die hinter dem Adressbereich der Default-E/A-Daten liegt.
----------	---

0A51_{hex} (USER FAIL)

Bedeutung:	Die Anschaltbaugruppe hat den Parameter <i>Frame_Reference</i> nicht akzeptiert.
Ursache:	Der angegebene Wert ist für den Parameter <i>Frame_Reference</i> nicht zulässig.
Abhilfe:	Ändern Sie den Wert des Parameters <i>Frame_Reference</i> . Zulässiger Wertebereich: $1 \leq \textit{Frame_Reference} \leq 254$

0A52_{hex} (USER FAIL)

Ursache:	Sie haben versucht, einen aktiven Teilnehmer zu überbrücken.
Abhilfe:	Schalten Sie den Teilnehmer zuvor mit „ <i>Control_Active_Configuration</i> “ (0713 _{hex}) ab.
Add_Error_Info:	Position im Konfigurationsrahmen

0A53_{hex} (USER FAIL)

- Bedeutung:** Die maximale Anzahl der Fernbus-Teilnehmer wurde überschritten.
- Ursache:** Die maximale Anzahl der zugelassenen Fernbus-Teilnehmer wurde durch den Anwender mit dem Dienst „Load_Configuration“ (0307_{hex}) oder beim automatischen Einlesen der angeschlossenen Teilnehmer mit dem Dienst „Create_Configuration“ (0710_{hex}) überschritten.
- Abhilfe:** Verringern Sie die Anzahl der Fernbus-Teilnehmer auf weniger als 254.

0A54_{hex} (USER FAIL)

- Bedeutung:** Die maximale Anzahl der E/A-Punkte wurde überschritten.
- Ursache:** Die maximale Anzahl der zugelassenen E/A-Punkte wurde durch den Anwender mit dem Dienst „Load_Configuration“ (0307_{hex}) oder beim automatischen Einlesen der angeschlossenen Teilnehmer mit dem Dienst „Create_Configuration“ (0710_{hex}) überschritten.
- Abhilfe:** Verringern Sie die Anzahl der E/A-Punkte auf das Maximum der digitalen Ein- und Ausgänge jeweils für den Ein- und Ausgangsbereich. Je nach Typ der Anschaltbaugruppe dürfen Sie bis zu 8192 E/A-Punkte verwenden. Die genaue Anzahl entnehmen Sie bitte der Dokumentation Ihrer Anschaltbaugruppe.

0A55_{hex}

- Bedeutung:** Die Konfigurationsdaten, die mit dem Dienst „Compare_Configuration“ (0317_{hex}) zur Anschaltbaugruppe übertragen wurden, stimmen nicht mit den Daten des Konfigurationsrahmens überein, der durch Parameter *Frame_Reference* angegeben wurde.
- Ursache:** Die durch den Parameter *Add_Error_Info* bezeichnete Teilnehmernummer (*Segment/Position*) ist unterschiedlich eingetragen worden.
- Abhilfe:** Prüfen Sie die Liste der übertragenen Konfigurationsdaten mit den angegebenen Konfigurationsrahmen.
- Add_Error_Info:** Zeilennummer des Konfigurationsrahmens, die nicht mit den übertragenen Konfigurationsdaten übereinstimmt.

0A56_{hex}

Bedeutung:	Die Konfigurationsdaten, die mit dem Dienst „Compare_Configuration“ (0317 _{hex}) zur Anschaltbaugruppe übertragen wurden, stimmen nicht mit den Daten des Konfigurationsrahmens überein, der durch Parameter <i>Frame_Reference</i> angegeben wurde.
Ursache:	Der durch den Parameter <i>Add_Error_Info</i> bezeichnete Teilnehmer-Code (Längen- und ID-Code) ist unterschiedlich eingetragen worden.
Abhilfe:	Prüfen Sie die Liste der übertragenen Konfigurationsdaten mit den angegebenen Konfigurationsrahmen.
Add_Error_Info:	Zeilennummer des Konfigurationsrahmens, die nicht mit den übertragenen Konfigurationsdaten übereinstimmt.

0A57_{hex}

Bedeutung:	Die Konfigurationsdaten, die mit dem Dienst „Compare_Configuration“ (0317 _{hex}) zur Anschaltbaugruppe übertragen wurden, stimmen nicht mit den Daten des Konfigurationsrahmens überein, der durch Parameter <i>Frame_Reference</i> angegeben wurde.
Ursache:	Die durch den Parameter <i>Add_Error_Info</i> bezeichnete Teilnehmerebene ist unterschiedlich eingetragen worden.
Abhilfe:	Prüfen Sie die Liste der übertragenen Konfigurationsdaten mit den angegebenen Konfigurationsrahmen.
Add_Error_Info:	Zeilennummer des Konfigurationsrahmens, die nicht mit den übertragenen Konfigurationsdaten übereinstimmt.

0A58_{hex}

Bedeutung:	Die Konfigurationsdaten, die mit dem Dienst „Compare_Configuration“ (0317 _{hex}) zur Anschaltbaugruppe übertragen wurden, stimmen nicht mit den Daten des Konfigurationsrahmens überein, der durch Parameter <i>Frame_Reference</i> angegeben wurde.
Ursache:	Die durch den Parameter <i>Add_Error_Info</i> bezeichnete Gruppennummer (Gruppe und Alternative) ist unterschiedlich eingetragen worden.
Abhilfe:	Prüfen Sie die Listen der übertragenen Konfigurationsdaten mit den angegebenen Konfigurationsrahmen.
Add_Error_Info:	Zeilennummer des Konfigurationsrahmens, die nicht mit den übertragenen Konfigurationsdaten übereinstimmt.

0A59_{hex}

Bedeutung: Der lokale Systemkoppler darf nicht geschaltet werden.
Ursache: Sie haben versucht den Slave-Teil des lokalen Systemkopplers zu schalten.

0A5A_{hex}

Bedeutung: Über den Dienst „Control_Active_Configuration“ (0713_{hex}) wurde versucht den aktiven Teilnehmer einzuschalten.
Ursache: Der aktive Teilnehmer kann mit dem angegebenen Dienst nicht eingeschaltet werden.
Add_Error_Info: Zeilennummer des Konfigurationsrahmens

0A5B_{hex}

Bedeutung: Die Konfigurationsdaten, die mit dem Dienst „Compare_Configuration“ (0317_{hex}) zur Anschaltbaugruppe übertragen wurden, stimmen nicht mit den Daten des Konfigurationsrahmens überein, der durch den Parameter *Frame_Reference* angegeben wurde.
Ursache: Der durch den Parameter *Add_Error_Info* bezeichnete Teilnehmer, der rückwirkungsfrei geschaltet werden soll, ist unterschiedlich eingetragen worden.
Abhilfe: Vergleichen Sie die Liste der übertragenen Konfigurationsdaten mit dem angegebenen Konfigurationsrahmen.
Add_Error_Info: Zeilennummer des Konfigurationsrahmens, die nicht mit den übertragenen Konfigurationsdaten übereinstimmt.

0A5C_{hex}

Bedeutung: Über den Dienst „Create_Configuration“ (0710_{hex}) wurde versucht einen Lokalbus-Aufbau einzulesen.
Ursache: Der eingelesene Lokalbus enthält zu viele Busteilnehmer (>= 63).
Abhilfe: Überprüfen Sie ihren Lokalbus-Aufbau.

0A60_{hex} (USER FAIL)

Bedeutung:	Die Anschaltbaugruppe konnte keinen Konfigurationsrahmen zuordnen.
Ursache:	Sie haben für den Parameter <i>Frame_Reference</i> einen Wert angegeben, unter dem noch kein Konfigurationsrahmen angelegt ist.
Abhilfe:	Beachten Sie die notwendige Reihenfolge der Dienste. Legen Sie zuerst den gewünschten Konfigurationsrahmen an, bevor Sie darauf zugreifen.

0A63_{hex} (USER FAIL)

Bedeutung:	Ein Diagnoseregister liegt im falschen Adressbereich.
Ursache:	Sie haben das Diagnose-Status- oder das Diagnose-Parameterregister in den Ausgangsadressbereich gelegt.
Abhilfe:	Legen Sie diese Register grundsätzlich in den Eingangsadressbereich.
Add_Error_Info:	<i>Variable_ID</i> des falsch adressierten Registers

0A64_{hex} (USER FAIL)

Bedeutung:	Adresskonflikt zwischen einem Diagnose-Register und einem Eingangs-Prozessdatum
Ursache:	Das Diagnose-Status- oder das Diagnose-Parameterregister liegt in einem Adressbereich, der von der Anschaltbaugruppe für Eingangs-Prozessdaten beansprucht wird. Dies geschieht z. B. nach Aufruf des Dienstes „Create_ Configuration“ (0710 _{hex}).
Abhilfe:	Legen Sie das Diagnose-Status- und das Diagnose-Parameterregister mit dem Dienst „Set_Value“ (0750 _{hex}) in Adressbereiche, die nicht von Eingangs-Prozessdaten belegt sind.
Add_Error_Info:	<i>Variable_ID</i> des Registers, das den Adresskonflikt verursacht.

0A65_{hex} (USER FAIL)

Bedeutung:	Fehler bei der Zuordnung eines Registers
Ursache:	Sie haben einem Register mit einer Datenkonsistenz von 16, 32 oder 64 Bit eine ungerade MPM-Adresse zugeordnet. Dies ist aber nur für Register mit einer Datenkonsistenz von 8 Bit erlaubt.
Abhilfe:	Weisen Sie Registern mit einer Datenkonsistenz von 16, 32 oder 64 Bit grundsätzlich gerade MPM-Adressen zu.
Add_Error_Info:	<i>Variable_ID</i> des Registers, dem die falsche Byte-Adresse zugeordnet ist

0A70_{hex}

Bedeutung:	Die Anschaltbaugruppe konnte den Dienst „Get_Diag_Info“ (032B _{hex}) nicht bearbeiten.
Ursache:	Sie haben versucht mit dem Parameter <i>Diag_Info_Attr</i> ein reserviertes Attribut freizuschalten.

0A80_{hex}

Bedeutung:	Die Konfiguration wird abgelehnt, der Bus kann nicht in Betrieb genommen werden.
Ursache:	Sie haben einen Bus, der aus SUPI-3-Teilnehmern besteht, als rückwirkungsfrei abschaltbar parametrieren. Die Busklemme, deren abzweigende Stich-Schnittstelle als rückwirkungsfrei abschaltbar parametrieren wurde, enthält keinen OPC-Chip.
Abhilfe:	Verwenden Sie als Busklemme nur Teilnehmer mit SUPI-3-OPC.
Add_Error_Info:	Zeilennummer im Konfigurationsrahmen

0AFB_{hex}

Bedeutung:	Die Anschaltbaugruppe hat die angegebene Datenkonsistenz nicht akzeptiert.
Ursache:	Sie haben beim Festlegen der Prozessdatenreferenzen einer Speicherzelle im MPM zwei verschiedene Datenkonsistenzen zugewiesen.
Abhilfe:	Prüfen Sie die Zuweisungen der Datenkonsistenzen.

0AFC_{hex}

Bedeutung: Auf der Anschaltbaugruppe ist ein Hardware-Fehler aufgetreten.
Abhilfe: Tauschen Sie die Anschaltbaugruppe aus.

0AFD_{hex}

Bedeutung: Der Inhalt des Adressdecoder-EEPROMs hat sich verändert.
Abhilfe: Starten Sie die Anschaltbaugruppe mit Power-Up/Reset und die Steuerung mit Neustart.

0AFE_{hex}

Bedeutung: Beim Einlesen der angeschlossenen Busstruktur wurde eine Adressüberschneidung festgestellt.
Ursache: Die eingelesenen Adressen der Busteilnehmer überlagern die Adresse des Kommunikationsregisters.
Abhilfe:

- Stellen Sie die Adresse für das Kommunikationsregister in CMD so ein, wie es auf der Anschaltbaugruppe eingestellt ist.
- Stellen Sie bei der Anschaltbaugruppe eine höhere Adresse für das Kommunikationsregister ein, die nicht bei der Adressierung vergeben wird oder ändern Sie die Adressen der Busteilnehmer.
- Legen Sie zum Einlesen des Konfigurationsrahmens mit dem Software-Tool CMD das Kommunikationsregister auf der Anschaltbaugruppe in einen höheren Bereich.

Besonderheit: Ist nur bei Siemens S5 oder Bosch SPS relevant.

0B02_{hex} (USER FAIL)

- Ursache:
1. In der bussynchronen Betriebsart ist entweder
 - keine Buszykluszeit eingestellt oder
 - die eingestellte Buszykluszeit ist zu klein.
 2. Bei einer Firmware bis Version 4.15 kann sein, dass
 - ein leerer Konfigurationsrahmen vorliegt oder
 - der erste Teilnehmer hinter der Anschaltbaugruppe abgeschaltet ist.
- Abhilfe:
1. Stellen Sie eine korrekte Buszykluszeit ein.
 2. – Aktivieren Sie einen korrekten Konfigurationsrahmen oder
 - schalten Sie den ersten Teilnehmer ein.

0B00_{hex}, 0B01_{hex}, 0B03_{hex}

- Bedeutung: Auf der Anschaltbaugruppe ist ein Firmware-Fehler aufgetreten.
- Abhilfe: Tauschen Sie die Anschaltbaugruppe aus.

0B80_{hex} (USER FAIL)

- Bedeutung: Die Anschaltbaugruppe kann den Dienst „Set_Value“ (0750_{hex}) bzw. „Read_Value“ (0351_{hex}) nicht bearbeiten.
- Ursache: Sie haben für den Parameter *Variable_ID* einen unzulässigen Wert angegeben.
- Abhilfe: Verwenden Sie nur die in der Beschreibung des Dienstes „Set_Value“ (0750_{hex}) bzw. „Read_Value“ (0351_{hex}) angegebenen Werte.

0B81_{hex} (USER FAIL)

- Bedeutung: Die Anschaltbaugruppe kann den Dienst „Set_Value“ (0750_{hex}) nicht bearbeiten.
- Ursache: Sie haben bei der Angabe eines Parameters den zulässigen Wertebereich überschritten.
- Abhilfe: Verwenden Sie nur die in der Beschreibung des Dienstes „Set_Value“ (0750_{hex}) angegebenen Werte.

0B83_{hex} (USER FAIL)

- Bedeutung:** Die Anschaltbaugruppe kann den Dienst „Control_Device_Function“ (0714_{hex}) nicht bearbeiten.
- Ursache:** Sie haben für den Parameter *Device_Function* einen unzulässigen Wert angegeben.
- Abhilfe:** Verwenden Sie nur die in der Beschreibung des Dienstes „Control_Device_Function“ (0714_{hex}) angegebenen Werte.

0B84_{hex} (USER FAIL)

- Bedeutung:** Die Anschaltbaugruppe kann den Dienst „Control_Device_Function“ (0714_{hex}) nicht bearbeiten.
- Ursache:** In der Liste der physikalischen INTERBUS-Teilnehmerpositionen wurde ein Fehler festgestellt.
- Abhilfe:** Prüfen Sie die Liste der physikalischen INTERBUS-Teilnehmerpositionen.

0B85_{hex} (USER FAIL)

- Bedeutung:** Die Konfiguration kann nicht erzeugt werden.
- Ursache:** In der Liste der aktiven Konfiguration wurde ein Fehler festgestellt.
- Abhilfe:** Prüfen Sie die Liste der aktiven Konfiguration.

0B86_{hex} (USER FAIL)

- Bedeutung:** Die Anschaltbaugruppe kann den Dienst „Control_Device_Function“ (0714_{hex}) nicht bearbeiten.
- Ursache:** Sie haben versucht, diesen Dienst auf einen INTERBUS-Teilnehmer ohne Kommandoregister anzuwenden.

0B87_{hex} (USER FAIL)

- Bedeutung:** Die Anschaltbaugruppe kann den Dienst „Control_Device_Function“ (0714_{hex}) nicht bearbeiten.
- Ursache:** Sie haben für den Parameter *Entry_Count* einen höheren Wert angegeben als INTERBUS-Teilnehmer vorhanden sind.
- Abhilfe:** Prüfen Sie den Dienstauf Ruf.

0B88_{hex}

- Bedeutung: Die Anschaltbaugruppe kann den Dienst „Control_Device_Function“ (0714_{hex}) nicht bearbeiten.
- Ursache: Firmware-Fehler auf der Anschaltbaugruppe
- Abhilfe: Tauschen Sie die Anschaltbaugruppe aus.

0B8A_{hex} (USER FAIL)

- Bedeutung: Nach Aktivieren einer neuen oder geänderten Konfiguration ging die Anschaltbaugruppe in den Stopp-Zustand.
- Ursache: Sie haben den Dienst „Alarm_Stop“ (1303_{hex}) zu früh gesendet.
- Abhilfe: Setzen Sie den Dienst „Alarm_Stop“ (1303_{hex}) später ab.

0B8C_{hex} (USER FAIL)

- Bedeutung: Die Anschaltbaugruppe kann den Konfigurationsrahmen nicht aktivieren.
- Ursache: Sie haben die maximal zulässige Anzahl von E/A-Bits überschritten.
- Abhilfe: Verringern Sie die Anzahl der INTERBUS-Teilnehmer. Je nach Typ der Anschaltbaugruppe dürfen Sie bis zu 8192 E/A-Punkte verwenden. Die genaue Anzahl entnehmen Sie bitte der Dokumentation Ihrer Anschaltbaugruppe.

0B8D_{hex} (USER FAIL)

- Bedeutung: Es wurde unzulässigerweise ein Synchroninterrupt ausgelöst.
- Ursache: Sie haben im MPM einen Synchroninterrupt ausgelöst, obwohl die Anschaltbaugruppe in der asynchronen Betriebsart arbeitet oder der Bus nicht gestartet war.
- Abhilfe: Lösen Sie Synchroninterrupts nur aus, wenn die Anschaltbaugruppe in einer synchronen Betriebsart arbeitet und der Bus läuft.

0B8E_{hex} (USER FAIL)

- Bedeutung:** Unerwarteter MPM Synchron-Interrupt (Diagonal-Handshake-Interrupt) ausgelöst.
- Ursache:** Sie haben im MPM einen Diagonal-Handshake-Interrupt, welcher für die synchrone Betriebsarten benötigt wird, ausgelöst, obwohl die Anschaltbaugruppe in der asynchronen Betriebsart arbeitet oder der Bus nicht gestartet war.
- Abhilfe:** Lösen Sie Synchroninterrupts nur aus, wenn die Anschaltbaugruppe in einer synchronen Betriebsart arbeitet und der Bus läuft.

0B8F_{hex} (USER FAIL)

- Bedeutung:** Überlauf beim Eingangs-FIFOs des INTERBUS-Protokoll-Masterchips (IPMS).
- Ursache:**
1. Sie haben auf einen Datenbereich mit einer Datenbreite zugegriffen, die geringer ist als die Datenkonsistenz, die Sie für diesen Bereich definiert haben.
 2. Die Übertragungsqualität sinkt, da der Bus schlecht installiert wurde.
- Abhilfe:**
1. Greifen Sie auf diesen Datenbereich nur mit der Datenbreite zu, die der vollen für diesen Datenbereich definierten Datenkonsistenz entspricht.
 2. Prüfen Sie die Übertragungsstrecke.

0B90_{hex}

- Bedeutung:** Der letzte INTERBUS-Zyklus wurde abgebrochen.
- Ursache:** Es ist ein Hardware-Fehler auf der Anschaltbaugruppe aufgetreten.
- Abhilfe:** Tauschen Sie die Anschaltbaugruppe aus.

0B91_{hex} (USER FAIL)

- Bedeutung:** In einem Buszustand ungleich ACTIVE oder RUN hat ein MPM-Teilnehmer einen Timeout verursacht.
- Ursache:** Es liegt ein Datenkonsistenz-Fehler vor, der entstehen kann, wenn aus dem MPM nicht alle Bit übernommen wurden.

0B92_{hex} (USER FAIL)

Bedeutung: Ein READ_PD-Dienst konnte nicht zu Ende bearbeitet werden.

Ursache: Es konnte innerhalb einer Timeout-Zeit kein Datenzyklus abgearbeitet werden, z. B. weil der Bus nicht im RUN ist oder während des Dienstes ein Busfehler aufgetreten ist.

Abhilfe: Bringen Sie den Bus wieder in den RUN-Status und rufen Sie anschließend den READ-PD-Dienst erneut auf.

0B93_{hex} (USER FAIL)

Bedeutung: Auf dem Teilnehmer wurden nach einem E/A-Timeout die Ausgänge zurückgesetzt.

Ursache: Der INTERBUS-Protokoll-Chip konnte in einer festgesetzten Zeit keine E/A-Daten verarbeiten. Mögliche Ursache könnte ein defekter Protokoll-Chip sein.

Abhilfe: Starten Sie das System neu.

Add_Error_Info: Teilnehmernummer (Segment . Position) des INTERBUS-Teilnehmers

0B94_{hex}

Bedeutung: Möglicherweise treten falsche Diagnose-Anzeigen auf.

Ursache: An einer Busklemme mit Lokalbus-Diagnose befindet sich ein Lokalbus-Teilnehmer, der die Lokalbus-Diagnose nicht unterstützt.

Abhilfe: Tauschen Sie den Lokalbus-Teilnehmer aus.

Add_Error_Info: Physikalische Position des ersten Lokalbus-Teilnehmers ohne Lokalbus-Diagnose

0B97_{hex}

Bedeutung: Für die programmsynchrone oder bussynchrone Betriebsart muss eine Scantime (mittlere PD-Zykluszeit) vorgegeben werden.

Ursache: Die Scantime wurde nicht vorgegeben.

Abhilfe: Geben Sie einen Wert für die Scantime ein.

0BB1_{hex} (PF)

Bedeutung: Der angegebene INTERBUS-Teilnehmer meldet einen Peripheriefehler.
Abhilfe: Prüfen Sie den angegebenen INTERBUS-Teilnehmer.
Add_Error_Info: Teilnehmernummer (Segment . Position) des INTERBUS-Teilnehmers

0BB2_{hex}

Bedeutung: Rekonfigurationsanforderung des angegebenen INTERBUS-Teilnehmers
Ursache: Auf dem angegebenen INTERBUS-Teilnehmer wurde der Rekonfigurationstaster gedrückt.
Add_Error_Info: Teilnehmernummer (Segment . Position) des INTERBUS-Teilnehmers

0BB4_{hex}

Bedeutung: Der Mikroprozessor des angegebenen Teilnehmers hat einen Reset durchgeführt.
Abhilfe: Prüfen Sie diesen INTERBUS-Teilnehmer.
Add_Error_Info: Fehlerort (Segment . Position)

0BB5_{hex}

Bedeutung: Die Übertragungsqualität im Datenhinweg der ankommenden Busschnittstelle (IN) des angegebenen Teilnehmers hat sich verschlechtert.
Abhilfe: Prüfen Sie die Datenleitungen der ankommenden Busschnittstelle.
Add_Error_Info: Teilnehmernummer (Segment . Position) des INTERBUS-Teilnehmers

0BB6_{hex}

Bedeutung: Die Übertragungsqualität im Datenrückweg der ankommenden Busschnittstelle (IN) des angegebenen Teilnehmers hat sich verschlechtert.
Abhilfe: Prüfen Sie die Datenleitungen der ankommenden Busschnittstelle.
Add_Error_Info: Teilnehmernummer (Segment . Position) des INTERBUS-Teilnehmers

0BB7_{hex}

Bedeutung: Der angegebene Teilnehmer hat einen E/A-Timeout gemeldet und seine Ausgänge, falls vorhanden, zurückgesetzt. Er verhält sich wie bei einem Bus-Reset.

Add_Error_Info: Teilnehmernummer (Segment . Position) des INTERBUS-Teilnehmers

0BC0_{hex}

Bedeutung: Systemkoppler meldet einen Fehler.

Ursachen:

- Der unterlagerte Bus ist durch einen Busfehler in den STOP-Zustand gegangen.
- Spannungs-Reset des Systemkoppler-Masterteils

Add_Error_Info: Teilnehmernummer (Segment . Position) des INTERBUS-Teilnehmers

0BC2_{hex}

Bedeutung: Die Spannung für die Initiatoren ist zu gering.

Ursache:

1. Zu viele Teilnehmer sind im Installationslokalbus-Ring projektiert.
2. Die Gesamtausdehnung des Installationslokalbus-Rings ist zu groß.

Abhilfe:

1. Prüfen Sie die Projektierung und verringern Sie die Anzahl der angeschlossenen Teilnehmer.
2. Verkleinern Sie die Ausdehnung des Installationslokalbus.

Add_Error_Info: Teilnehmernummer (Segment . Position) des INTERBUS-Teilnehmers

0BC3_{hex}

Bedeutung:	Die Temperatur des INTERBUS-Protokoll-Chips ist zu hoch.
Ursache:	Die Umgebungstemperatur ist zu hoch.
Abhilfe:	<ul style="list-style-type: none">– Verringern Sie die Umgebungstemperatur.– Sorgen Sie für eine Fremdbelüftung des Teilnehmers.
Add_Error_Info:	Teilnehmernummer (Segment . Position) des INTERBUS-Teilnehmers

0BC4_{hex}

Bedeutung:	Die interne Stromquelle eines Loop-Teilnehmers ist überlastet.
Ursache:	Die angeschlossene Last durch Sensoren und/oder Aktoren ist zu hoch.
Abhilfe:	<ul style="list-style-type: none">– Prüfen Sie die Anzahl der angeschlossenen Sensoren/Aktoren.– Prüfen Sie die Sensoren/Aktoren auf Fehler.
Add_Error_Info:	Teilnehmernummer (Segment . Position) des INTERBUS-Teilnehmers

0BC5_{hex}

Bedeutung:	Der erlaubte Ausgangsstrom des Leistungstreibers wurde überschritten.
Ursache:	Die angeschlossene Last durch Sensoren und/oder Aktoren ist zu hoch.
Abhilfe:	<ul style="list-style-type: none">– Prüfen Sie die Anzahl der angeschlossenen Sensoren/Aktoren.– Prüfen Sie die Sensoren/Aktoren auf Fehler.
Add_Error_Info:	Teilnehmernummer (Segment . Position) des INTERBUS-Teilnehmers

0BD0_{hex}

Bedeutung:	Die Anschaltbaugruppe konnte die Konfiguration nicht in Betrieb nehmen.
Ursache:	Es ist kein Busaufbau angeschlossen.
Abhilfe:	Schliessen Sie einen Busaufbau an.

0BD1_{hex}

Bedeutung:	Ein Datenzyklus hat die vorgegebene PD-Zykluszeit überschritten.
------------	--

Die Anschaltbaugruppe meldet diesen Umstand nur, wenn dies zuvor mit dem Dienst „Set_Value“ (0750_{hex}) aktiviert wurde.

Ursache: Innerhalb der vorgegebenen PD-Zykluszeit konnte kein Update der Prozessdaten vorgenommen werden.

Abhilfe: Prüfen Sie Ihre Anlage oder setzen Sie die Vorgabe für die PD-Zykluszeit mit dem Dienst „Set_Value“ (0750_{hex}) herauf.

0BD2_{hex}

Bedeutung: Die Buswarnzeit ist abgelaufen.

Ursache: Innerhalb der von Ihnen mit dem Dienst „Set_Value“ (0750_{hex}) vorgegebenen Buswarnzeit konnte kein Datenzyklus übertragen werden.

Abhilfe:

- Prüfen Sie Ihre Anlage oder
- setzen Sie die Buswarnzeit mit dem Dienst „Set_Value“ (0750_{hex}) auf einen höheren Wert.

0BD4_{hex}

Bedeutung: Es ist ein einzelner Busfehler aufgetreten.
Die Anschaltbaugruppe meldet diesen Umstand nur, wenn dies zuvor aktiviert wurde.

0BD5_{hex}

Bedeutung: Die abzweigende Stich-Schnittstelle des in der Additional-Info angegebenen Teilnehmers hat aufgrund eines Fehlers im angeschlossenen Segment abgeschaltet.

Ursache: Im an die abzweigende Stich-Schnittstelle des Teilnehmers angeschlossenen Bussegment ist ein Fehler aufgetreten.

Abhilfe: Überprüfen Sie ihren Busaufbau

Add_Error_Info: Teilnehmernummer des abgeschalteten INTERBUS-Teilnehmers

0BDB_{hex}

Bedeutung: Die Prozessvorverarbeitungs-Task ist nicht mehr im Zustand RUN. Der Bus wurde gestoppt, damit keine unvorverarbeiteten Daten übertragen werden.

Ursache: Fehler im Vorverarbeitungsprogramm (z. B. Division durch Null) oder bewusstes Stoppen der Vorverarbeitung durch den Anwender.

0BDC_{hex}

Bedeutung: Im Zustand ACTIVE oder im Zustand RUN wurde von der Anschaltbaugruppe ein Alarm-Stopp generiert.

Ursache: Der Alarm-Stopp wurde aufgrund eines SYSFAIL im Host-System generiert.

0BDD_{hex}

Bedeutung: Der Bus wurde angehalten, da vom Anwender ein MPM-Timeout verursacht wurde. Es könnten sonst inkonsistente Daten übertragen werden.

Ursache: Es könnte z.B. durch einen Byte-Zugriff auf eine 16-Bit verriegelte MPM-Adresse verursacht worden sein.

Abhilfe: Greifen Sie nur auf Datenbereiche zu, die der vollen für diesen Datenbereich definierten Datenkonsistenz entspricht.

0BDE_{hex}

Bedeutung: In der Betriebsart *bussynchron* ist ein Synchronisationsfehler aufgetreten.

Abhilfe:

- Prüfen Sie Ihre Anlage oder
- setzen Sie die PD-Zykluszeit mit dem Dienst „Set_Value“ (0750_{hex}) auf einen höheren Wert.

0BDF_{hex} (LOOK FOR FAIL)

Bedeutung: Die Anschaltbaugruppe hat die Datenübertragung unterbrochen und sucht den Fehlerort und die Fehlerursache.

Ursache: Es ist ein Busfehler aufgetreten.

Abhilfe: Warten Sie die Fehlersuche ab. Die Anschaltbaugruppe wird Ihnen das Ergebnis mitteilen.

0BE0_{hex}

Bedeutung: Die Fehlerlokalisierung konnte nicht beendet werden.

Ursache: Während der Fehlerlokalisierung (Look for Fail) wurde durch den Anwender der Dienst „Alarm_Stop“ (1303_{hex}) abgesetzt.

3.2 Fehler-Codes zu allgemeinen Busfehlern (BUS FAIL)

Den allgemeinen Busfehlern, denen ein Fehlerort oder -bereich nicht eindeutig zugeordnet werden kann, sind die Fehler-Codes **0BE1_{hex}** bis **0BEA_{hex}** zugeordnet. Das Display der Anschaltbaugruppe zeigt hierbei nur die letzten beiden Stellen **E1_{hex}** bis **EA_{hex}** an.



Beachten Sie, dass nach Auftreten dieser Fehler und anschließendem Auslesen der Diagnose-Informationen aus den Diagnose-Parameterregistern mit dem Dienst „Read_Value“, im Diagnose-Parameterregister der Fehler-Code und im erweiterten Diagnose-Parameterregister der Wert **0000_{hex}** abgebildet ist.

E1_{hex} bzw. **0BE1_{hex}** (BUS FAIL)

Bedeutung:	Es ist ein schwerwiegender Fehler aufgetreten, der zum Abschalten des Bussystems geführt hat. Bei der anschließenden Untersuchung der augenblicklichen Konfiguration konnte aber kein Fehler gefunden werden. Dies deutet auf eine kurzzeitig auftretende Fehlerursache hin.
Ursache:	Die Störung tritt aufgrund <ul style="list-style-type: none"> – von Installationsfehlern – des Defekts eines INTERBUS-Teilnehmers auf.
Abhilfe:	Prüfen Sie die Anlage auf: <ul style="list-style-type: none"> – fehlende oder nicht korrekte Schirmung der Buskabel (Steckverbinder), – fehlende oder nicht korrekte Erdung/Potentialausgleich, – defekte Verbindungen im Steckverbinder (Wackelkontakt, kalte Lötstelle), – Leitungsbrüche in der Fern- und Lokalbusverkabelung, – Spannungseinbrüche auf der Logikversorgung der Fernbus-Teilnehmer. <p>Sind Einträge in den Top-Ten vorhanden, so können diese Hinweise auf den Fehlerort geben. Die Top-Ten können Sie über die Diagnose-Anzeige oder über den Dienst „Get_Diag_Info“ (032B_{hex}) abfragen.</p>



Beachten Sie, dass nach Auftreten der Fehler $E0_{\text{hex}}$ und $E1_{\text{hex}}$ und anschließendem Auslesen der Diagnose-Informationen aus den Diagnose-Parameterregistern mit dem Dienst „Read_Value“, im Diagnose-Parameterregister der Fehler-Code und im erweiterten Diagnose-Parameterregister der Fehlerort abgebildet ist.

$E2_{\text{hex}}$ bzw. $0BE2_{\text{hex}}$ (BUS FAIL)

- Bedeutung:** Die Anschaltbaugruppe hat Konfigurationsänderungen erkannt, die den weiteren Betrieb des Datenverkehrs über den Bus nicht zulassen.
- Ursache:**
- Die maximal zulässige Anzahl der INTERBUS-Worte wurde überschritten.
 - Die maximal zulässige Anzahl der INTERBUS-Teilnehmer wurde überschritten.

$E4_{\text{hex}}$ bzw. $0BE4_{\text{hex}}$ (BUS FAIL)

- Bedeutung:** Während aufgrund des Dienstes „Create_Configuration“ (0710_{hex}) die Buskonfiguration erfasst wurde, ist ein schwerwiegender Fehler aufgetreten. Der Fehler führte zum Abschalten des Bussystems, wobei der Fehlerort nicht ermittelt werden konnte. Dies deutet auf eine kurzzeitig auftretende Fehlerursache hin. Die Fehlerrate kann dabei sehr hoch sein.
- Ursache:** Die Störung tritt aufgrund
- von Installationsfehlern oder
 - des Defekts eines INTERBUS-Teilnehmers
- auf.
- Abhilfe:** Prüfen Sie die Anlage auf:
- fehlende oder nicht korrekte Schirmung der Buskabel (Steckverbinder),
 - fehlende oder nicht korrekte Erdung/Potentialausgleich,
 - defekte Verbindungen im Steckverbinder (Wackelkontakt, kalte Lötstelle),
 - Leitungsbrüche in der Fern- und Lokalbusverkabelung,
 - Spannungseinbrüche auf der Logikversorgung der Fernbus-Teilnehmer.

Die Diagnose-Möglichkeiten der Anschaltbaugruppe werden verbessert, wenn ihr die Konfiguration des Bussystems bekannt ist. Aus diesem Grund ist es besser, die gewünschte Konfiguration in die Anschaltbaugruppe mit

den Diensten „Load_Configuration“ (0307_{hex}) oder „Complete_Load_Configuration“ (030A_{hex}) zu laden, und dann mit dem Dienst „Activate_Configuration“ (0711_{hex}) zu aktivieren.

E6_{hex} bzw. 0BE6_{hex} (BUS FAIL)

Bedeutung: Es ist ein schwerwiegender Fehler aufgetreten, der zum Abschalten des Bussystems geführt hat. Bei der anschließenden Untersuchung der augenblicklichen Konfiguration konnte aber kein Fehler gefunden werden. Dies deutet auf eine kurzzeitig auftretende Fehlerursache hin. Von dem Fehler waren nur Datenzyklen und keine ID-Zyklen betroffen.

Ursache: Die Störung tritt aufgrund

- von Installationsfehlern,
- des Defekts eines INTERBUS-Teilnehmers oder
- beim Überbrücken von Teilnehmern mit einer Datenbreite kleiner 8 Bit (bedingt durch den Transport der Daten dieser Teilnehmer in den Speicher der Anschaltbaugruppe)

auf.

Abhilfe: Prüfen Sie die Anlage auf:

- fehlende oder nicht korrekte Schirmung der Buskabel (Steckverbinder),
- fehlende oder nicht korrekte Erdung/Potentialausgleich,
- defekte Verbindungen im Steckverbinder (Wackelkontakt, kalte Lötstelle),
- Leitungsbrüche in der Fern- und Lokalbusverkabelung,
- Spannungseinbrüche auf der Logikversorgung der Fernbus-Teilnehmer.

Sind Einträge in den Top-Ten vorhanden, so können diese Hinweise auf den Fehlerort geben. Die Top-Ten können Sie über die Diagnose-Anzeige oder den Dienst „Get_Diag_Info“ (032B_{hex}) abfragen.

E7_{hex} bzw. 0BE7_{hex} (BUS FAIL)

Bedeutung: Die Anschaltbaugruppe konnte bei der Bearbeitung des Dienstes

- „Activate_Configuration“ (0711_{hex}) oder
- „Control_Active_Configuration“ (0713_{hex})

die Konfiguration nicht aktivieren. Der Fehlerort konnte nicht ermittelt werden.

Ursache:

Die Störung tritt aufgrund

- von Installationsfehlern
- des Defekts eines INTERBUS-Teilnehmers

auf.

Abhilfe:

Prüfen Sie die Anlage auf:

- fehlende oder nicht korrekte Schirmung der Buskabel (Steckverbinder),
- fehlende oder nicht korrekte Erdung/Potentialausgleich,
- defekte Verbindungen im Steckverbinder (Wackelkontakt, kalte Lötstelle),
- Leitungsbrüche in der Fern- und Lokalbusverkabelung,
- Spannungseinbrüche auf der Logikversorgung der Fernbus-Teilnehmer.

E8_{hex} bzw. 0BE8_{hex} (BUS FAIL)

Bedeutung:

Es ist ein schwerwiegender Fehler aufgetreten, der zum Abschalten des Bussystems geführt hat. Bei der Untersuchung der augenblicklichen Konfiguration hat der Diagnose-Algorithmus Fehler erkannt, konnte aber keinen eindeutigen Fehlerort lokalisieren. Dies deutet auf eine kurzzeitig auftretende Fehlerursache hin. Die Fehlerrate kann hierbei sehr hoch sein.

Ursache:

Die Störung tritt aufgrund

- von Installationsfehlern oder
- des Defekts eines INTERBUS-Teilnehmers auf.

Abhilfe:

Prüfen Sie die Anlage auf:

- fehlende oder nicht korrekte Schirmung der Buskabel (Steckverbinder),
- fehlende oder nicht korrekte Erdung/Potentialausgleich,
- defekte Verbindungen im Steckverbinder (Wackelkontakt, kalte Lötstelle),
- Leitungsbrüche in der Fern- und Lokalbusverkabelung,
- Spannungseinbrüche auf der Logikversorgung der Fernbus-Teilnehmer.

Sind Einträge in den Top-Ten vorhanden, so können diese Hinweise auf den Fehlerort geben. Die Top-Ten können über die Diagnose-Anzeige oder über den Dienst „Get_Diag_Info“ (032B_{hex}) abfragen.

E9_{hex} bzw. 0BE9_{hex} (BUS FAIL)

Bedeutung: Es ist ein schwerwiegender Fehler aufgetreten, der zum Abschalten des Bussystems geführt hat. Bei der Untersuchung der augenblicklichen Konfiguration hat der Diagnose-Algorithmus Fehler erkannt, konnte aber keinen eindeutigen Fehlerort lokalisieren. Dies deutet auf eine kurzzeitig auftretende Fehlerursache hin. Die Fehlerrate kann hierbei sehr hoch sein.

Ursache: Die Störung tritt aufgrund

- von Installationsfehlern oder
- des Defekts eines INTERBUS-Teilnehmers auf.

Abhilfe: Prüfen Sie die Anlage auf:

- fehlende oder nicht korrekte Schirmung der Buskabel (Steckverbinder),
- fehlende oder nicht korrekte Erdung/Potentialausgleich,
- defekte Verbindungen im Steckverbinder (Wackelkontakt, kalte Lötstelle),
- Leitungsbrüche in der Fern- und Lokalbusverkabelung,
- Spannungseinbrüche auf der Logikversorgung der Fernbus-Teilnehmer.

Sind Einträge in den Top-Ten vorhanden, so können diese Hinweise auf den Fehlerort geben. Die Top-Ten können Sie über die Diagnose-Anzeige oder über den Dienst „Get_Diag_Info“ (032B_{hex}) abfragen.

EA_{hex} bzw. 0BEA_{hex} (BUS FAIL)

Bedeutung: Der Dienst „Control_Device_Function“ (0714_{hex}) konnte nicht ausgeführt werden.

Ursache: Ein fataler Fehler ist aufgetreten.

Abhilfe: Wiederholen Sie den Dienst, falls sich der Controller noch im Zustand RUN oder ACTIVE befindet. Ist die Diagnose aktiv, so warten Sie zunächst deren Ergebnis ab. Der danach angezeigte Busfehler kennzeichnet den Fehlerort.

0BF0_{hex} (BUS FAIL)

Bedeutung:	Die Datenübertragung ist kurzzeitig gestört worden. Daraufhin hat die Anschaltbaugruppe alle Ausgänge zurückgesetzt und die Datenübertragung gestoppt. Die Diagnose-Anzeige zeigt eine INTERBUS-Teilnehmernummer. Die Fehlerursache liegt <ul style="list-style-type: none">– im davorliegenden Bussegment eines Lokalbusses,– im davorliegenden Bussegment einer ST-Kompaktstation,– in den Bussegmenten eines davorliegenden Fernbus-Stichs (z.B. Installationsfernbus) oder– im Bussegment des angezeigten INTERBUS-Teilnehmers.
Ursache:	<ul style="list-style-type: none">– Spannungs-Reset eines INTERBUS-Teilnehmers im angegebenen Bereich.– Leitungsbruch im angegebenen Bussegment.– An einem INTERBUS-Teilnehmer im angegebenen Bereich ist die Brücke (RBST bzw. LBST) defekt, die sich im Stecker für den abgehenden Bus befindet.
Add_Error_Info:	Teilnehmernummer (Segment . Position) des INTERBUS-Teilnehmers

0BF1_{hex} (BUS FAIL)

Bedeutung:	An einer Busklemme ist die Datenübertragung gestört.
Ursache:	<ul style="list-style-type: none">– Der Stecker für den abzweigenden Fernbus-Stich ist nicht gesteckt.– Im Stecker für den abzweigenden Fernbus-Stich ist die Brücke (LBST) defekt.
Abhilfe:	Prüfen Sie den Stecker für den abzweigenden Fernbus-Stich.
Add_Error_Info:	Teilnehmernummer (Segment . Position) des INTERBUS-Teilnehmers

0BF2_{hex} (BUS FAIL)

Bedeutung:	An einer Busklemme ist die Datenübertragung gestört.
Ursache:	<ul style="list-style-type: none">– Der Stecker für den weiterführenden Fernbus ist nicht gesteckt.– Im Stecker für den weiterführenden Fernbus ist die Brücke (RBST) defekt.
Abhilfe:	Prüfen Sie den Stecker für den weiterführenden Fernbus.
Add_Error_Info:	Teilnehmernummer (Segment . Position) des INTERBUS-Teilnehmers

0BF3_{hex} (BUS FAIL)

Bedeutung:	Die Datenübertragung ist <ul style="list-style-type: none">– an einer Busklemme,– an einem Lokalbus-Teilnehmer oder– innerhalb einer IB-ST-Kompaktstation gestört.
Ursache:	Lokalbus <ul style="list-style-type: none">– Der Stecker für den abgehenden Lokalbus ist nicht gesteckt.– Die Brücke (RBST bzw. LBST) am Anschluss für den abgehenden Lokalbus ist defekt. ST-Kompaktstation <ul style="list-style-type: none">– Das ST-Kabel ist nicht gesteckt.– Die über das nächste Modul der IB-ST-Kompaktstation geführte RBST-Verbindung ist unterbrochen.
Add_Error_Info:	Teilnehmernummer (Segment . Position) des INTERBUS-Teilnehmers

0BF4_{hex} (BUS FAIL)

Bedeutung:	Übertragungsstörung (CRC-Fehler) im Datenhinweg an der ankommenden Bus-Schnittstelle (IN) des angegebenen INTERBUS-Teilnehmers
Ursache:	Übertragungsstörungen
Abhilfe:	Prüfen Sie das Segment des angegebenen INTERBUS-Teilnehmers auf: <ul style="list-style-type: none">– fehlende oder nicht korrekte Schirmung der Buskabel (Steckverbinder),– fehlende oder nicht korrekte Erdung/Potentialausgleich,– defekte Verbindungen im Steckverbinder (Wackelkontakt, kalte Lötstelle)– Spannungseinbrüche auf der Logikversorgung der Fernbus-Teilnehmer,– fehlerhafte LWL-Konfektionierung.
Add_Error_Info:	Teilnehmernummer (Segment . Position) des INTERBUS-Teilnehmers

0BF5_{hex} (BUS FAIL)

Bedeutung:	Übertragungsstörung (CRC-Fehler) im Datenrückweg an der ankommenden Bus-Schnittstelle (IN) des angegebenen INTERBUS-Teilnehmers
Abhilfe:	Prüfen Sie das Segment des angegebenen INTERBUS-Teilnehmers auf: <ul style="list-style-type: none">– fehlende oder nicht korrekte Schirmung der Buskabel (Steckverbinder),– fehlende oder nicht korrekte Erdung/Potentialausgleich,– defekte Verbindungen im Steckverbinder (Wackelkontakt, kalte Lötstelle)– Spannungseinbrüche auf der Logikversorgung der Fernbus-Teilnehmer,– fehlerhafte LWL-Konfektionierung.
Add_Error_Info:	Teilnehmernummer (Segment . Position) des INTERBUS-Teilnehmers

0BF6_{hex} (BUS FAIL)

Bedeutung:	Die Datenübertragung ist kurzzeitig gestört worden. Daraufhin hat die Anschaltbaugruppe alle Ausgänge zurückgesetzt und die Datenübertragung gestoppt. Die Diagnose-Anzeige zeigt eine INTERBUS-Teilnehmernummer. Die Fehlerursache liegt <ul style="list-style-type: none">– im davorliegenden Bussegment eines Lokalbusses,– im davorliegenden Bussegment einer ST-Kompaktstation,– in den Bussegmenten eines davorliegenden Fernbus-Stichs (z. B. Installationsfernbus) oder– im Bussegment des angezeigten INTERBUS-Teilnehmers.
Ursache:	Mögliche Ursachen sind: <ul style="list-style-type: none">– Spannungs-Reset eines INTERBUS-Teilnehmers im angegebenen Bereich– Leitungsbruch im angegebenen Bussegment– An einem INTERBUS-Teilnehmer im angegebenen Bereich ist die Brücke (RBST bzw. LBST) defekt, die sich im Stecker für den abgehenden Bus befindet.
Add_Error_Info:	Teilnehmernummer (Segment . Position) des INTERBUS-Teilnehmers

0BF8_{hex} (BUS FAIL)

Bedeutung: Mehrfachfehler beim Erfassen von Peripheriedaten am angegebenen Teilnehmer aufgetreten. Der Fehler konnte nicht eindeutig lokalisiert werden.

Ursache: Die Störung tritt aufgrund

- von Installationsfehlern
- des Defekts eines INTERBUS-Teilnehmers

auf.

Mögliche Fehlerorte:

- Angegebene Teilnehmer,
- der davorliegende komplette Bus oder
- alle Teilnehmer, die an der OUT2-Schnittstelle des angegebenen Teilnehmers angeschlossen sind.

Abhilfe: Prüfen Sie die Anlage auf:

- fehlende oder nicht korrekte Schirmung der Buskabel (Steckverbinder),
- fehlende oder nicht korrekte Erdung/Potentialausgleich,
- defekte Verbindungen im Steckverbinder (Wackelkontakt, kalte Lötstelle),
- Leitungsbrüche in der Fern- und Lokalbusverkabelung,
- Spannungseinbrüche auf der Logikversorgung der Fernbus-Teilnehmer.

Sind Einträge in den Top-Ten vorhanden, so können diese Hinweise auf den Fehlerort geben. Die Top-Ten können Sie über die Diagnose-Anzeige oder über den Dienst „Get_Diag_Info“ (032B_{hex}) abfragen.

Add_Error_Info: Teilnehmernummer (Segment . Position) des INTERBUS-Teilnehmers

0BF9_{hex} (BUS FAIL)

Bedeutung: Ein Mehrfachfehler ist bei der schnellen Diagnose am angegebenen Teilnehmer aufgetreten. Der Fehler konnte nicht eindeutig lokalisiert werden.

Ursache: Die Störung tritt aufgrund

- von Installationsfehlern
- des Defekts eines INTERBUS-Teilnehmers

auf.

Mögliche Fehlerorte:

- Angegebene Teilnehmer,
- der davorliegende komplette Bus oder
- alle Teilnehmer, die an OUT2 des angegebenen Teilnehmers angeschlossen sind.

Abhilfe:

Prüfen Sie die Anlage auf:

- fehlende oder nicht korrekte Schirmung der Buskabel (Steckverbinder),
- fehlende oder nicht korrekte Erdung/Potentialausgleich,
- defekte Verbindungen im Steckverbinder (Wackelkontakt, kalte Lötstelle),
- Leitungsbrüche in der Fern- und Lokalbusverkabelung,
- Spannungseinbrüche auf der Logikversorgung der Fernbus-Teilnehmer.

Sind Einträge in den Top-Ten vorhanden, so können diese Hinweise auf den Fehlerort geben. Die Top-Ten können Sie über die Diagnose-Anzeige oder über den Dienst „Get_Diag_Info“ (032B_{hex}) abfragen.

Add_Error_Info:

Teilnehmernummer (Segment . Position) des INTERBUS-Teilnehmers

0BFA_{hex} (BUS FAIL)

Bedeutung:

Mehrfachfehler beim Aufschalten oder bei der Dauerdiagnose am angegebenen Teilnehmer aufgetreten.

Ursache:

Die Störung tritt aufgrund

- von Installationsfehlern
- des Defekts eines INTERBUS-Teilnehmers

auf.

Mögliche Fehlerorte:

- Angegebene Teilnehmer,
- der komplette Bus der davor liegt oder
- alle Teilnehmer, die an OUT2 des angegebenen Teilnehmers angeschlossen sind.

Abhilfe:

Prüfen Sie die Anlage auf:

- fehlende oder nicht korrekte Schirmung der Buskabel (Steckverbinder),
- fehlende oder nicht korrekte Erdung/Potentialausgleich,

- defekte Verbindungen im Steckverbinder (Wackelkontakt, kalte Lötstelle),
- Leitungsbrüche in der Fern- und Lokabusverkabelung,
- Spannungseinbrüche auf der Logikversorgung der Fernbus-Teilnehmer.

Sind Einträge in den Top-Ten vorhanden, so können diese Hinweise auf den Fehlerort geben. Die Top-Ten können Sie über die Diagnose-Anzeige oder über den Dienst „Get_Diag_Info“ (032B_{hex}) abfragen.

Add_Error_Info: Teilnehmernummer (Segment . Position) des INTERBUS-Teilnehmers

0BFB_{hex} (BUS FAIL)

Bedeutung: Fehler durch schnelle Diagnose erkannt.

Mögliche Fehlerorte:

- Angegebene Teilnehmer,
- der davorliegende komplette Bus oder
- alle Teilnehmer, die an OUT2 des angegebenen Teilnehmers angeschlossen sind.

Abhilfe: Prüfen Sie die Anlage auf:

- fehlende oder nicht korrekte Schirmung der Buskabel (Steckverbinder),
- fehlende oder nicht korrekte Erdung/Potentialausgleich,
- defekte Verbindungen im Steckverbinder (Wackelkontakt, kalte Lötstelle),
- Leitungsbrüche in der Fern- und Lokabusverkabelung,
- Spannungseinbrüche auf der Logikversorgung der Fernbus-Teilnehmer.

Sind Einträge in den Top-Ten vorhanden, so können diese Hinweise auf den Fehlerort geben. Die Top-Ten können Sie über die Diagnose-Anzeige oder über den Dienst „Get_Diag_Info“ (032B_{hex}) abfragen.

Add_Error_Info: Teilnehmernummer (Segment . Position) des INTERBUS-Teilnehmers

0BFC_{hex} (BUS FAIL)

Bedeutung:	<p>Die Datenübertragung ist kurzzeitig gestört worden. Daraufhin hat die Anschaltbaugruppe alle Ausgänge zurückgesetzt und die Datenübertragung gestoppt. Die Diagnose-Anzeige zeigt eine INTERBUS-Teilnehmernummer. Die Fehlerursache liegt</p> <ul style="list-style-type: none">– im davorliegenden Bussegment eines Lokalbusses,– im davorliegenden Bussegment einer ST-Kompaktstation,– in den Bussegmenten eines davorliegenden Fernbus-Stichs (z. B. Installationsfernbus),– im Bussegment des angezeigten INTERBUS-Teilnehmers oder– bei den Übertragungsstrecken oder Teilnehmern in den Abzweigen vor dem angezeigten INTERBUS-Teilnehmer.
Ursache:	<ul style="list-style-type: none">– Spannungs-Reset eines INTERBUS-Teilnehmers im angegebenen Bereich.– Leitungsbruch im angegebenen Bussegment.– An einem INTERBUS-Teilnehmer im angegebenen Bereich ist die Brücke (RBST oder LBST) defekt, die sich im Stecker für den abgehenden Bus befindet.
Add_Error_Info:	Teilnehmernummer (Segment . Position) des INTERBUS-Teilnehmers

3.3 Fehler-Codes zu Fernbus- und Lokalbusfehlern

Bei lokalisierten Fern- oder Lokalbusfehlern liegt in der *Add_Error_Info* der Fehlerort kodiert vor. Die genaue Fehlerposition wird nur angegeben, wenn kein Schnittstellenfehler vorliegt. Bei einem vorliegenden Schnittstellenfehler wird nur das fehlerhafte Bussegment angegeben. Das Bit 7 gibt dabei an, ob ein Schnittstellenfehler vorliegt oder nicht. Hierdurch verändert sich auch die Bedeutung der Bits 0 bis 6. Dies führt zu drei Zuständen mit den folgend dargestellten Bitkombinationen in der *Add_Error_Info*.

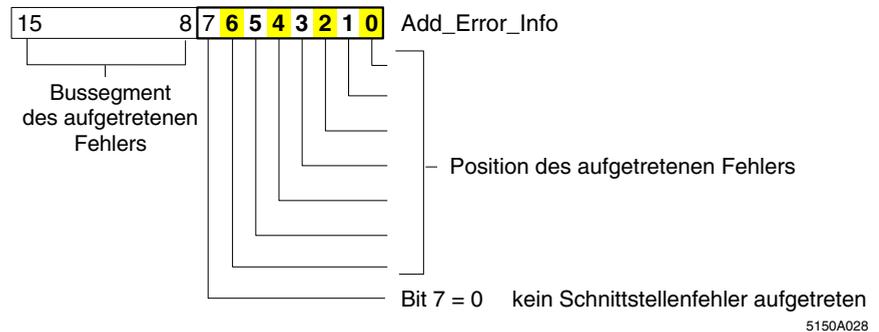


Bild 3-1 Kein Schnittstellenfehler aufgetreten

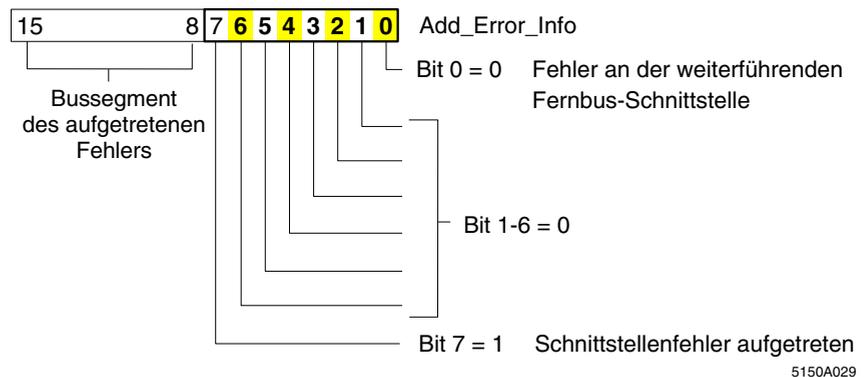


Bild 3-2 Fehler an weiterführender Fernbus-Schnittstelle

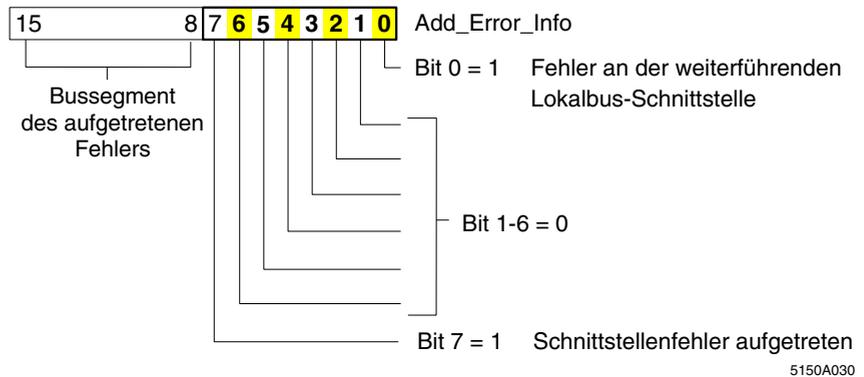


Bild 3-3 Fehler an weiterführender Lokalbus-Schnittstelle

**0C10_{hex} bis 0C13_{hex} (RB FAIL) und
0D10_{hex} bis 0D13_{hex} (LB FAIL)**

- Bedeutung:** Ein INTERBUS-Teilnehmer fehlt.
- Ursache:** Im angeschlossenen Busaufbau fehlt ein Teilnehmer, der in der aktiven Konfiguration eingetragen und nicht als abgeschaltet gekennzeichnet ist. Die aktive Konfiguration ist die Menge der am INTERBUS angeschlossenen INTERBUS-Teilnehmer, die bei Buszyklen mit ihren Daten innerhalb des Summenrahmens liegen. Die aktive Konfiguration darf sich vom angeschlossenen Busaufbau nur unterscheiden, wenn physikalisch angeschlossene Bussegmente abgeschaltet sind.
- Abhilfe:** Vergleichen Sie die aktive Konfiguration mit dem angeschlossenen Busaufbau unter Berücksichtigung von eventuell abgeschalteten Bussegmenten.
- Add_Error_Info:** Fehlerort (Segment . Position)

**0C14_{hex} bis 0C17_{hex} (RB FAIL) und
0D14_{hex} bis 0D17_{hex} (LB FAIL)**

- Bedeutung:** Mehrfacher Fehler im Segment des angegebenen INTERBUS-Teilnehmers.
- Ursache:** Übertragungsstörungen
- Abhilfe:** Prüfen Sie das Segment des angegebenen INTERBUS-Teilnehmers auf:

- fehlende oder nicht korrekte Schirmung der Buskabel (Steckverbinder),
- fehlende oder nicht korrekte Erdung/Potentialausgleich,
- defekte Verbindungen im Steckverbinder (Wackelkontakt, kalte Lötstelle)
- Spannungseinbrüche auf der Logikversorgung der Fernbus-Teilnehmer.

Add_Error_Info: Fehlerort (Segment . Position)

**0C18_{hex} bis 0C1B_{hex} (RB FAIL) und
0D18_{hex} bis 0D1B_{hex} (LB FAIL)**

Bedeutung: Mehrfacher Timeout im Segment des angegebenen INTERBUS-Teilnehmers

Ursache: Übertragungsstörungen

Abhilfe: Prüfen Sie das Segment des angegebenen INTERBUS-Teilnehmers auf:

- fehlende oder nicht korrekte Schirmung der Buskabel (Steckverbinder),
- fehlende oder nicht korrekte Erdung/Potentialausgleich,
- defekte Verbindungen im Steckverbinder (Wackelkontakt, kalte Lötstelle)
- Spannungseinbrüche auf der Logikversorgung der Fernbus-Teilnehmer.

Add_Error_Info: Fehlerort (Segment . Position)

**0C1C_{hex} bis 0C1F_{hex} (RB FAIL) und
0D1C_{hex} bis 0D1F_{hex} (LB FAIL)**

Bedeutung: Übertragungsstörung (CRC-Fehler) im Datenhinweg an der ankommenden Bus-Schnittstelle (IN) des angegebenen INTERBUS-Teilnehmers

Ursache: Übertragungsstörungen

Abhilfe: Prüfen Sie das Segment des angegebenen INTERBUS-Teilnehmers auf:

- fehlende oder nicht korrekte Schirmung der Buskabel (Steckverbinder),
- fehlende oder nicht korrekte Erdung/Potentialausgleich,
- defekte Verbindungen im Steckverbinder (Wackelkontakt, kalte Lötstelle)

- Spannungseinbrüche auf der Logikversorgung der Fernbus-Teilnehmer.

Add_Error_Info: Fehlerort (Segment . Position)

**0C20_{hex} bis 0C23_{hex} (RB FAIL) und
0D20_{hex} bis 0D23_{hex} (LB FAIL)**

Bedeutung: Die Firmware-Komponente Medium-Attachment-Unit (MAU) hat eine Unterbrechung der Datenübertragung diagnostiziert.

Ursache: Leitungsbruch im Datenhinweg an der ankommenden Bus-Schnittstelle (IN) des angegebenen INTERBUS-Teilnehmers

Abhilfe: Prüfen Sie die Kabel, Stecker und INTERBUS-Anschlüsse auf mögliche Unterbrechungen, und führen Sie gegebenenfalls eine Instandsetzung durch.

Add_Error_Info: Fehlerort (Segment . Position)

**0C24_{hex} bis 0C27_{hex} (RB FAIL) und
0D24_{hex} bis 0D27_{hex} (LB FAIL)**

Bedeutung: Übertragungsstörung (CRC-Fehler) im Datenrückweg an der ankommenden Bus-Schnittstelle (IN) des angegebenen INTERBUS-Teilnehmers

Ursache: Übertragungsstörungen

Abhilfe: Prüfen Sie das Segment des angegebenen INTERBUS-Teilnehmers auf:

- fehlende oder nicht korrekte Schirmung der Buskabel (Steckverbinder),
- fehlende oder nicht korrekte Erdung/Potentialausgleich,
- defekte Verbindungen im Steckverbinder (Wackelkontakt, kalte Lötstelle)
- Spannungseinbrüche auf der Logikversorgung der Fernbus-Teilnehmer.

Add_Error_Info: Fehlerort (Segment . Position)

**0C28_{hex} bis 0C2B_{hex} (RB FAIL) und
0D28_{hex} bis 0D2B_{hex} (LB FAIL)**

Bedeutung:	Die Medium-Attachment-Unit (MAU) hat eine Unterbrechung der Datenübertragung diagnostiziert.
Ursache:	Leistungsbruch im Datenrückweg an der ankommenden Bus-Schnittstelle (IN) des angegebenen INTERBUS-Teilnehmers.
Abhilfe:	Prüfen Sie die Kabel, Stecker und INTERBUS-Anschlüsse auf mögliche Unterbrechungen, und führen Sie gegebenenfalls eine Instandsetzung durch.
Add_Error_Info:	Fehlerort (Segment . Position)

**0C2C_{hex} bis 0C2F_{hex} (RB FAIL) und
0D2C_{hex} bis 0D2F_{hex} (LB FAIL)**

Bedeutung:	Unerwartete Änderung des Signals RBST oder LBST.
Ursache:	Im Stecker an der abgehenden Schnittstelle des vorgelagerten Busteilnehmers fehlt die Brücke oder die Brücke ist defekt (Wackelkontakt, kalte Lötstelle).
Abhilfe:	Prüfen Sie das Segment des angegebenen INTERBUS-Teilnehmers auf defekte Verbindungen im Steckverbinder (Wackelkontakt, kalte Lötstelle). Löten Sie eine Brücke ein oder stellen Sie eine einwandfreie Verbindung der bereits vorhandenen Brücke her, um ein fehlerfreies RBST- oder LBST-Signal zu erzeugen.
Add_Error_Info:	Fehlerort (Segment . Position)

0C30_{hex} bis 0C33_{hex} (RB FAIL)

Bedeutung:	Mehrfacher Fehler im Segment des angegebenen INTERBUS-Teilnehmers.
Ursache:	<ol style="list-style-type: none">1. Übertragungsstörungen2. Der angegebene INTERBUS-Teilnehmer hat als Slave-Chip einen SUPI 1, der im μP-Mode betrieben wird. Dieser Mode wird von der Firmware Ihrer Anschaltbaugruppe nicht unterstützt.
Abhilfe:	<ol style="list-style-type: none">1. Prüfen Sie das Segment des angegebenen INTERBUS-Teilnehmers auf:<ul style="list-style-type: none">– fehlende oder nicht korrekte Schirmung der Buskabel (Steckverbinder),– fehlende oder nicht korrekte Erdung/Potentialausgleich,– defekte Verbindungen im Steckverbinder (Wackelkontakt, kalte Lötstelle)– Spannungseinbrüche auf der Logikversorgung der Fernbus-Teilnehmer.2. Tauschen Sie den angegebenen Teilnehmer gegen einen Teilnehmer mit SUPI 3-Chip aus.
Add_Error_Info:	Fehlerort (Segment . Position)

**0C40_{hex} bis 0C43_{hex} (RB FAIL) und
0D40_{hex} bis 0D43_{hex} (LB FAIL)**

Bedeutung:	Der Längen-Code des angegebenen INTERBUS-Teilnehmers stimmt nicht mit dem Eintrag im Konfigurationsrahmen überein.
Abhilfe:	Passen Sie den aktiven Konfigurationsrahmen dem Busaufbau an.
Add_Error_Info:	Fehlerort (Segment . Position)

**0C44_{hex} bis 0C47_{hex} (RB FAIL) und
0D44_{hex} bis 0D47_{hex} (LB FAIL)**

Bedeutung:	Der ID-Code des angegebenen INTERBUS-Teilnehmers stimmt nicht mit dem Eintrag im Konfigurationsrahmen überein.
Abhilfe:	Passen Sie den aktiven Konfigurationsrahmen dem Busaufbau an.
Add_Error_Info:	Fehlerort (Segment . Position)

**0C48_{hex} bis 0C4B_{hex} (RB FAIL) und
0D48_{hex} bis 0D4B_{hex} (LB FAIL)**

Bedeutung:	Der Bus kann zwar eingelesen, aber nicht in Betrieb genommen werden. Es können nur ID-Zyklen aber keine Datenzyklen gefahren werden.
Ursache:	<ul style="list-style-type: none">– Das Datenregister des angegebenen INTERBUS-Teilnehmers ist unterbrochen.– Die Anzahl der Datenregister des angegebenen INTERBUS-Teilnehmers entspricht nicht dem im Konfigurationsrahmen eingetragenen Längen-Code.
Add_Error_Info:	Fehlerort (Segment . Position)

**0C4C_{hex} bis 0C4F_{hex} (RB FAIL) und
0D4C_{hex} bis 0D4F_{hex} (LB FAIL)**

Bedeutung:	Der angegebene INTERBUS-Teilnehmer hat einen unzulässigen ID-Code.
Add_Error_Info:	Fehlerort (Segment . Position)

0D50_{hex} bis 0D53_{hex} (LB FAIL)

Bedeutung:	Der angegebene INTERBUS-Teilnehmer hat den ID-Code eines Fernbus-Teilnehmers, befindet sich aber in einem Lokalbus.
Add_Error_Info:	Fehlerort (Segment . Position)

**0C54_{hex} bis 0C57_{hex} (RB FAIL) und
0D54_{hex} bis 0D57_{hex} (LB FAIL)**

Bedeutung:	Der angegebene INTERBUS-Teilnehmer hat als Slave-Chip einen SUP1 1, der im μ P-Mode betrieben wird. Dieser Mode wird von der Firmware Ihrer Anschaltbaugruppe nicht unterstützt.
Abhilfe:	Tauschen Sie den Teilnehmer gegen einen Teilnehmer mit SUP1 3-Chip aus.
Add_Error_Info:	Fehlerort (Segment . Position)

**0C58_{hex} bis 0C5B_{hex} (RB FAIL) und
0D58_{hex} bis 0D5B_{hex} (LB FAIL)**

Bedeutung:	An der weiterführenden Fernbus-Schnittstelle (OUT1) des angegebenen INTERBUS-Teilnehmers ist die Datenübertragung unterbrochen.
Ursache:	<ol style="list-style-type: none">1. Der Stecker ist nicht gesteckt.2. Die Brücke zur Steckererkennung (RBST oder LBST) ist defekt.
Abhilfe:	<ol style="list-style-type: none">1. Prüfen Sie, ob der Stecker gesteckt ist.2. Prüfen Sie, ob die Brücke zur Steckererkennung (RBST oder LBST) defekt ist.
Add_Error_Info:	Fehlerort (Segment . Position)

**0C5C_{hex} bis 0C5F_{hex} (RB FAIL) und
0D5C_{hex} bis 0D5F_{hex} (LB FAIL)**

Bedeutung:	An der abzweigenden Bus-Schnittstelle (OUT2) des angegebenen INTERBUS-Teilnehmers ist die Datenübertragung unterbrochen.
Ursache:	<ol style="list-style-type: none">1. Der Stecker ist nicht gesteckt.2. Die Brücke zur Steckererkennung (RBST oder LBST) ist defekt.
Abhilfe:	<ol style="list-style-type: none">1. Prüfen Sie, ob der Stecker gesteckt ist.2. Prüfen Sie, ob die Brücke zur Steckererkennung (RBST oder LBST) defekt ist.
Add_Error_Info:	Fehlerort (Segment . Position)

**0C60_{hex} bis 0C63_{hex} (RB FAIL) und
0D60_{hex} bis 0D63_{hex} (LB FAIL)**

Bedeutung:	An der weiterführenden Fernbus-Schnittstelle (OUT1) des angegebenen INTERBUS-Teilnehmers (mit SUPI 3) wurde kurzzeitig die Datenübertragung gestört, obwohl diese Schnittstelle nicht belegt ist.
Ursache:	Der Protokoll-Chip SUPI 3 hat einen CRC- oder MAU-Fehler diagnostiziert.
Abhilfe:	Tauschen Sie den INTERBUS-Teilnehmer aus.
Add_Error_Info:	Fehlerort (Segment . Position)

**0C64_{hex} bis 0C67_{hex} (RB FAIL und
0D64_{hex} bis 0D67_{hex} (LB FAIL)**

Bedeutung:	An der abzweigenden Bus-Schnittstelle (OUT2) des angegebenen INTERBUS-Teilnehmers (mit SUPI 3) wurde kurzzeitig die Datenübertragung gestört, obwohl diese Schnittstelle nicht belegt ist.
Ursache:	Der Protokoll-Chip SUPI 3 hat einen CRC- oder MAU-Fehler diagnostiziert.
Abhilfe:	Tauschen Sie den INTERBUS-Teilnehmer aus.
Add_Error_Info:	Fehlerort (Segment . Position)

**0C68_{hex} bis 0C6B_{hex} (RB FAIL) und
0D68_{hex} bis 0D6B_{hex} (LB FAIL)**

Bedeutung:	Der Protokoll-Chip SUPI 3 des angegebenen INTERBUS-Teilnehmer hat einen E/A-Timeout festgestellt.
Add_Error_Info:	Fehlerort (Segment . Position)

**0C6C_{hex} bis 0C6F_{hex} (RB FAIL) und
0D6C_{hex} bis 0D6F_{hex} (LB FAIL)**

Bedeutung:	Der angegebene INTERBUS-Teilnehmer hat einen Reset ausgeführt.
Ursache:	Der angegebene INTERBUS-Teilnehmer wird nur unzureichend mit Spannung versorgt oder ist defekt.
Abhilfe:	<ul style="list-style-type: none"> – Prüfen Sie diesen INTERBUS-Teilnehmer. – Prüfen Sie die Spannungsversorgung dieses INTERBUS-Teilnehmers auf Einhaltung des Nennwerts und Überschreiten des zulässigen Wechselspannungsanteils. Die Angaben können Sie dem zugehörigen Datenblatt entnehmen. – Prüfen Sie, ob das Netzteil der Busklemme überlastet ist. Die maximal zulässige Stromabgabe der Busklemme und die typische Stromaufnahme der angeschlossenen Lokalbusteilnehmer können Sie den zugehörigen Datenblättern entnehmen.
Add_Error_Info:	Fehlerort (Segment . Position)

**0C70_{hex} bis 0C73_{hex} (RB FAIL) und
0D70_{hex} bis 0D73_{hex} (LB FAIL)**

- Bedeutung:** Die Datenübertragung wurde abgebrochen. Bei einem INTERBUS-Teilnehmer, dessen Protokoll-Chip im Mikroprozessor-Mode betrieben wird, hat der Mikroprozessor den Protokoll-Chip nicht initialisiert.
- Ursache:**
1. Die Anschaltbaugruppe hat versucht den Bus schneller in den Zustand *ACTIVE* zu schalten als der Mikroprozessor des INTERBUS-Teilnehmers den Protokoll-Chip initialisieren konnte.
 2. Der INTERBUS-Teilnehmer ist defekt.
- Abhilfe:**
1. Verzögern Sie den Aufruf des Dienstes „Activate_Configuration“ (0711_{hex}) bis der Mikroprozessor den Protokoll-Chip initialisiert hat.
 2. Tauschen Sie den INTERBUS-Teilnehmer aus.
- Add_Error_Info:** Fehlerort (Segment . Position)

**0C74_{hex} bis 0C77_{hex} (RB FAIL) und
0D74_{hex} bis 0D77_{hex} (LB FAIL)**

- Bedeutung:** Die Datenübertragung wurde abgebrochen.
- Ursache:** Bei einem INTERBUS-Teilnehmer ist bei dem Protokoll-Chip eine unzulässige Betriebsart eingestellt.
- Abhilfe:**
- Stellen Sie eine gültige Betriebsart ein oder
 - tauschen Sie den Teilnehmer aus!
- Add_Error_Info:** Fehlerort (Segment . Position)

0C78_{hex} (RB FAIL) und 0D78_{hex} (LB FAIL)

- Bedeutung:** Die Datenlänge des angegebenen Teilnehmers stimmt nicht mit dem Konfigurationsrahmen überein, wenn der dynamische PCP-Kanal eingeschaltet ist.
- Add_Error_Info:** Fehlerort (Segment . Position)

0C79_{hex} (RB FAIL) und 0D79_{hex} (LB FAIL)

Bedeutung: Der ID-Code des angegebenen Teilnehmers stimmt nicht mit dem Konfigurationsrahmen überein, wenn der dynamische PCP-Kanal eingeschaltet ist.

Add_Error_Info: Fehlerort (Segment . Position)

0C7A_{hex} (RB FAIL) und 0D7A_{hex} (LB FAIL)

Bedeutung: Die Breite des dynamischen PCP-Kanals des angegebenen Teilnehmers ist nicht zulässig.

Add_Error_Info: Fehlerort (Segment . Position)

0C7B_{hex} (RB FAIL) und 0D7B_{hex} (LB FAIL)

Bedeutung: Der angegebene Teilnehmer hat einen ID-Code für Teilnehmer mit dynamischen PCP-Kanal, kann diesen Kanal aber nicht nutzen.

Ursache:

- Der angegebene Teilnehmer verfügt nicht über einen entsprechenden Protokoll-Chip.
- Die Firmware der Anschaltbaugruppe unterstützt Teilnehmer mit dynamischen PCP-Kanal noch nicht.

Add_Error_Info: Fehlerort (Segment . Position)

0C7C_{hex} (RB FAIL) und 0D7C_{hex} (LB FAIL)

Bedeutung: Beim angegebenen Teilnehmer ist der dynamische PCP-Kanal eingeschaltet, obwohl er ausgeschaltet sein sollte.

Add_Error_Info: Fehlerort (Segment . Position)

0C7D_{hex} (RB FAIL) und 0D7D_{hex} (LB FAIL)

Bedeutung: Beim angegebenen Teilnehmer ist der dynamische PCP-Kanal abgeschaltet, obwohl er eingeschaltet sein sollte.

Add_Error_Info: Fehlerort (Segment . Position)

0C7E_{hex} (RB FAIL) und 0D7E_{hex} (LB FAIL)

Bedeutung:	Der INTERBUS-Protokoll-Chip im angegebenen Teilnehmer kann nicht in der vorgegebenen Konfiguration betrieben werden.
Ursache:	Der INTERBUS-Protokoll-Chip unterstützt die hierfür erforderliche Funktionen nicht.
Abhilfe:	Tauschen Sie den Teilnehmer aus.
Add_Error_Info:	Fehlerort (Segment . Position)

**0C80_{hex} bis 0C83_{hex} (RB FAIL) und
0D80_{hex} bis 0D83_{hex} (LB FAIL)**

Bedeutung:	Mehrfacher Fehler an der weiterführenden Bus-Schnittstelle (OUT1) des angegebenen INTERBUS-Teilnehmers.
Ursache:	Defekt des an dieser Bus-Schnittstelle angeschlossenen Buskabels, des folgenden INTERBUS-Teilnehmers oder eines Teilnehmers im eventuell folgenden Lokalbus.
Abhilfe:	Prüfen Sie diesen Anlagenteil auf: <ul style="list-style-type: none">– fehlende oder nicht korrekte Schirmung der Buskabel (Steckverbinder),– fehlende oder nicht korrekte Erdung/Potentialausgleich,– defekte Verbindungen im Steckverbinder (Wackelkontakt, kalte Lötstelle),– Spannungseinbrüche auf der Logikversorgung der Fernbus-Teilnehmer,– fehlerhafte LWL-Konfektionierung.
Add_Error_Info:	Fehlerort (Segment . Position)

**0C84_{hex} bis 0C87_{hex} (RB FAIL) und
0D84_{hex} bis 0D87_{hex} (LB FAIL)**

Bedeutung:	Mehrfacher Timeout an der weiterführenden Bus-Schnittstelle (OUT1) des angegebenen INTERBUS-Teilnehmers.
Ursache:	Defekt <ul style="list-style-type: none">– des an dieser Bus-Schnittstelle angeschlossenen Buskabels,– des folgenden INTERBUS-Teilnehmers oder

- eines Teilnehmers im eventuell folgenden Lokalbus.
- Abhilfe: Prüfen Sie diesen Anlagenteil auf:
- fehlende oder nicht korrekte Schirmung der Buskabel (Steckverbinder),
 - fehlende oder nicht korrekte Erdung/Potentialausgleich,
 - defekte Verbindungen im Steckverbinder (Wackelkontakt, kalte Lötstelle),
 - Spannungseinbrüche auf der Logikversorgung der Fernbus-Teilnehmer,
 - fehlerhafte LWL-Konfektionierung.
- Add_Error_Info: Fehlerort (Segment . Position)

**0C88_{hex} bis 0C8B_{hex} (RB FAIL) und
0D88_{hex} bis 0D8B_{hex} (LB FAIL)**

- Bedeutung: An der weiterführenden Bus-Schnittstelle (OUT1) des angegebenen INTERBUS-Teilnehmers wurde ein zusätzliches Gerät gefunden.
- Ursache:
- Es ist ein INTERBUS-Teilnehmer angeschlossen, der in der aktiven Konfiguration nicht eingetragen ist.
 - Es ist ein INTERBUS-Kabel ohne weiteren INTERBUS-Teilnehmer angeschlossen.
- Abhilfe: Prüfen Sie den Busaufbau.
- Add_Error_Info: Fehlerort (Segment . Position)

**0C8C_{hex} bis 0C8F_{hex} (RB FAIL) oder
0D8C_{hex} bis 0D8F_{hex} (LB FAIL)**

- Bedeutung: Es können nur ID-Zyklen, aber keine Datenzyklen gefahren werden.
- Ursache:
- Das Datenregister des INTERBUS-Teilnehmers, der an die weiterführende Fernbus-Schnittstelle (OUT1) des angegebenen INTERBUS-Teilnehmers angeschlossen ist, ist unterbrochen.
 - Die Anzahl der Datenregister des INTERBUS-Teilnehmers, der an die weiterführende Fernbus-Schnittstelle (OUT1) des angegebenen INTERBUS-Teilnehmers angeschlossen ist, stimmt nicht mit dem Längen-Code überein.
- Add_Error_Info: Fehlerort (Segment . Position)

0C90_{hex} bis 0C93_{hex} (RB FAIL)

- Bedeutung: Der angegebene INTERBUS-Teilnehmer konnte das folgende Bussegment nicht aktivieren.
- Ursache: Der INTERBUS-Teilnehmer, der an die weiterführende Schnittstelle (OUT1) des angegebenen INTERBUS-Teilnehmers angeschlossen ist, hat einen Spannungsreset ausgeführt oder ist defekt.
- Abhilfe:
- Prüfen Sie diesen INTERBUS-Teilnehmer.
 - Prüfen Sie die Spannungsversorgung dieses INTERBUS-Teilnehmers auf Einhaltung des Nennwerts und Überschreiten des zulässigen Wechselspannungsanteils. Die Angaben können Sie dem zugehörigen Datenblatt entnehmen.
 - Prüfen Sie, ob das Netzteil dieser Busklemme überlastet ist. Die maximal zulässige Stromabgabe der Busklemme und die typische Stromaufnahme der angeschlossenen Lokalbusteilnehmer können Sie den zugehörigen Datenblättern entnehmen.
- Add_Error_Info: Fehlerort (Segment . Position)

0C94_{hex} bis 0C97_{hex} (RB FAIL)

- Bedeutung: An der weiterführender Fernbus-Schnittstelle (OUT1) des angegebenen INTERBUS-Teilnehmers wurde ein INTERBUS-Teilnehmer mit dem ID-Code eines Lokalbus-Teilnehmers gefunden.
- Abhilfe: Schließen Sie einen Fernbus-Teilnehmer an.
- Add_Error_Info: Fehlerort (Segment . Position)

**0C98_{hex} bis 0C9B_{hex} (RB FAIL) oder
0D98_{hex} bis 0D9B_{hex} (LB FAIL)**

- Bedeutung: Der INTERBUS-Teilnehmer, der an die weiterführende Fernbus-Schnittstelle (OUT1) des angegebenen INTERBUS-Teilnehmers angeschlossen ist, hat einen ungültigen ID-Code.
- Abhilfe: Tauschen Sie diesen Teilnehmer aus
- Add_Error_Info: Fehlerort (Segment . Position)

0CA0_{hex}

Bedeutung: Die Anschaltbaugruppe kann den Busaufbau nicht in Betrieb nehmen.

Ursache: Sie haben versucht, einen Bus als rückwirkungsfrei abschaltbar zu parametrieren, der nicht nur SUPI 3-Teilnehmer enthält.

Abhilfe: Verwenden Sie nur Teilnehmer mit SUPI 3.



Setzen Sie das rückwirkungsfreie Schalten nur in einem reinen SUPI 3-System ein.

0D9C_{hex} bis 0D9F_{hex} (LB FAIL)

Bedeutung: Der direkt an die Anschaltbaugruppe angeschlossene Lokalbus besteht aus mehr INTERBUS-Teilnehmern, als in der aktiven Konfiguration eingetragen sind.

Abhilfe: Prüfen Sie diesen Lokalbus.

Add_Error_Info: Fehlerort (Segment . Position)

**0CC0_{hex} bis 0CC3_{hex} (RB FAIL) und
0DC0_{hex} bis 0DC3_{hex} (LB FAIL)**

Bedeutung: Mehrfacher Fehler an der abzweigenden Bus-Schnittstelle (OUT2) des angegebenen INTERBUS-Teilnehmers.

Ursache:

- An der abzweigenden Bus-Schnittstelle (OUT 2) ist ein INTERBUS-Kabel ohne weiteren INTERBUS-Teilnehmer angeschlossen.
- Es ist ein Lokal- / Fernbuskabel defekt, das zum Lokal- / Fernbus des angegebenen INTERBUS-Teilnehmers gehört.
- Es ist ein INTERBUS-Teilnehmer defekt, der an den Lokal- / Fernbus des angegebenen INTERBUS-Teilnehmers angeschlossen ist.
- Ausfall der Spannungsversorgung für die Modulelektronik (Logikspannung U_L), die durch die Busklemme zur Verfügung gestellt wird.
- Ausfall der Spannungsversorgung für die Busklemme (Logikspannung U_L).

Abhilfe: Prüfen Sie diesen Lokal- / Fernbus.

Add_Error_Info: Fehlerort (Segment . Position)

**0CC4_{hex} bis 0CC7_{hex} (RB FAIL) und
0DC4_{hex} bis 0DC7_{hex} (LB FAIL)**

Bedeutung:	Mehrfacher Timeout an der abzweigenden Bus-Schnittstelle (OUT2) des angegebenen INTERBUS-Teilnehmers.
Ursache:	<ul style="list-style-type: none">– Es ist ein Lokal- / Fernbuskabel defekt, das zum Lokal- / Fernbus des angegebenen INTERBUS-Teilnehmers gehört.– Es ist ein INTERBUS-Teilnehmer defekt, der an den Lokal- / Fernbus des angegebenen INTERBUS-Teilnehmers angeschlossen ist.– Ausfall der Spannungsversorgung für die Modulelektronik (Logikspannung U_L), die durch die Busklemme zur Verfügung gestellt wird.– Ausfall der Spannungsversorgung für die Busklemme (Logikspannung U_L).
Abhilfe:	Prüfen Sie diesen Lokal- / Fernbus.
Add_Error_Info:	Fehlerort (Segment . Position)

**0CC8_{hex} bis 0CCB_{hex} (RB FAIL) und
0DC8_{hex} bis 0DCB_{hex} (LB FAIL)**

Bedeutung:	An der abzweigenden Bus-Schnittstelle (OUT2) des angegebenen INTERBUS-Teilnehmers wurden zusätzliche Geräte gefunden.
Ursache:	<ul style="list-style-type: none">– Es ist ein INTERBUS-Teilnehmer angeschlossen, der in der aktiven Konfiguration nicht eingetragen ist.– Es ist ein INTERBUS-Kabel ohne weiteren INTERBUS-Teilnehmer angeschlossen.
Abhilfe:	Prüfen Sie Ihr INTERBUS-Netzwerk an dem angegebenen Fehlerort.
Add_Error_Info:	Fehlerort (Segment . Position)

**0CCC_{hex} bis 0CCF_{hex} (RB FAIL) und
0DCC_{hex} bis 0DCF_{hex} (LB FAIL)**

Bedeutung:	Der Bus kann zwar eingelesen, aber nicht in Betrieb genommen werden. Es können nur ID-Zyklen, aber keine Datenzyklen gefahren werden.
Ursache:	<ol style="list-style-type: none"> 1. Das Datenregister des an OUT2 angeschlossenen INTERBUS-Teilnehmers ist unterbrochen. 2. Die Anzahl der Datenregister des INTERBUS-Teilnehmers, der an die abzweigende Bus-Schnittstelle (OUT2) des angegebenen INTERBUS-Teilnehmers angeschlossen ist, entspricht nicht dem im Konfigurationsrahmen eingetragenen Längen-Code.
Abhilfe:	<ol style="list-style-type: none"> 1. Tauschen Sie den INTERBUS-Teilnehmer, der an die abzweigende Bus-Schnittstelle (OUT2) des angegebenen INTERBUS-Teilnehmers angeschlossen ist, aus. 2. Passen Sie im Konfigurationsrahmen den Eintrag für den Längen-Code an.
Add_Error_Info:	Fehlerort (Segment . Position)

**0CDO_{hex} bis 0CD3_{hex} (RB FAIL) und
0DD0_{hex} bis 0DD3_{hex} (LB FAIL)**

Bedeutung:	Nach dem Öffnen der abzweigenden Bus-Schnittstelle (OUT2) des angegebenen INTERBUS-Teilnehmers wurden außer einer Busklemme auch noch weitere INTERBUS-Teilnehmer in den Datenring aufgenommen.
Ursache:	Der INTERBUS-Teilnehmer, der an der abzweigenden Bus-Schnittstelle (OUT2) des angegebenen INTERBUS-Teilnehmers angeschlossen ist, hat einen Spannungs-Reset ausgeführt oder ist defekt.
Abhilfe:	<ul style="list-style-type: none"> – Prüfen Sie diesen INTERBUS-Teilnehmer. – Prüfen Sie die Spannungsversorgung dieses INTERBUS-Teilnehmers auf Einhaltung des Nennwerts und Überschreiten des zulässigen Wechselspannungsanteils. Die Angaben können Sie dem zugehörigen Datenblatt entnehmen. – Prüfen Sie, ob das Netzteil dieser Busklemme überlastet ist. Die maximal zulässige Stromabgabe der Busklemme und die typische Stromaufnahme der angeschlossenen Lokalbusteilnehmer können Sie den zugehörigen Datenblättern entnehmen.
Add_Error_Info:	Fehlerort (Segment . Position)

**0CD4_{hex} bis 0CD7_{hex} (RB FAIL) und
0DD4_{hex} bis 0DD7_{hex} (LB FAIL)**

- Bedeutung: Lokalbusfehler im 8-Leiter-Lokalbus, der an den angegebenen INTERBUS-Teilnehmer angeschlossen ist.
- Ursachen:
- Es ist ein Lokalbuskabel defekt, das zum Lokalbus des angegebenen INTERBUS-Teilnehmers gehört.
 - Es ist ein INTERBUS-Teilnehmer defekt, der an den Lokalbus des angegebenen INTERBUS-Teilnehmers angeschlossen ist.
 - Ausfall der Spannungsversorgung für die Modulelektronik (Logikspannung U_L), die durch die Busklemme zur Verfügung gestellt wird.
- Abhilfe: Prüfen Sie diesen Lokalbus.
- Add_Error_Info: Fehlerort (Segment . Position)

**0CD8_{hex} bis 0CDB_{hex} (RB FAIL) und
0DD8_{hex} bis 0DDB_{hex} (LB FAIL)**

- Bedeutung: Der Lokalbus, der an die angegebenen Busklemme angeschlossen ist, besteht aus mehr Lokalbus-Teilnehmern, als in der aktiven Konfiguration eingetragen sind.
- Add_Error_Info: Fehlerort (Segment . Position)

**0CDC_{hex} bis 0CDF_{hex} (RB FAIL) und
0DDC_{hex} bis 0DDF_{hex} (LB FAIL)**

- Bedeutung: Der INTERBUS-Teilnehmer, der an der abzweigenden Bus-Schnittstelle (OUT2) des angegebenen INTERBUS-Teilnehmers angeschlossen ist, hat einen ungültigen ID-Code.
- Add_Error_Info: Fehlerort (Segment . Position)

8040_{hex}

- Bedeutung: Der angegebene INTERBUS-Teilnehmer hat einen Einzelkanalfehler gemeldet.
- Abhilfe: Prüfen Sie den Kanal des angegebenen INTERBUS-Teilnehmers.
- Add_Error_Info: INTERBUS-Teilnehmernummer (Segment . Position : Kanal)

8060_{hex}

Bedeutung:	Der angegebene INTERBUS-Teilnehmer hat einen Kurzschluss am Ausgang gemeldet.
Abhilfe:	Prüfen Sie die Ausgangsbeschaltung des angegebenen INTERBUS-Teilnehmers
Add_Error_Info:	INTERBUS-Teilnehmernummer (Segment . Position : Kanal)

8080_{hex}

Bedeutung:	Der angegebene INTERBUS-Teilnehmer hat einen Initiatorversorgungsfehler auf einer (mehreren) Gruppe(n) gemeldet.
Abhilfe:	Prüfen Sie die Gruppe(n) des angegebenen INTERBUS-Teilnehmer.
Add_Error_Info:	INTERBUS-Teilnehmernummer (Segment . Position)

80A0_{hex}

Bedeutung:	Der angegebene INTERBUS-Teilnehmer hat einen Spannungsversorgungsfehler auf einer (mehreren) Gruppe(n) gemeldet.
Abhilfe:	Prüfen Sie die Gruppe(n) des angegebenen INTERBUS-Teilnehmer.
Add_Error_Info:	INTERBUS-Teilnehmernummer (Segment . Position)

80B1_{hex}

Bedeutung:	Der angegebene INTERBUS-Teilnehmer hat einen Konfigurationsfehler gemeldet.
Abhilfe:	Prüfen Sie die Parameter des angegebenen INTERBUS-Teilnehmers.
Add_Error_Info:	INTERBUS-Teilnehmernummer (Segment . Position)

80B2_{hex}

Bedeutung:	Der angegebene INTERBUS-Teilnehmer hat einen Fehler in der Peripherieelektronik gemeldet.
Abhilfe:	Prüfen Sie die am angegebenen INTERBUS-Teilnehmers angeschlossenen Sensoren und Aktoren.
Add_Error_Info:	INTERBUS-Teilnehmernummer (Segment . Position)

80B4_{hex}

- Bedeutung:** Der angegebene INTERBUS-Teilnehmer hat eine Temperaturüberschreitung gemeldet.
- Abhilfe:** Prüfen Sie die am angegebenen INTERBUS-Teilnehmers angeschlossenen Sensoren und Aktoren.
- Add_Error_Info:** INTERBUS-Teilnehmernummer (Segment . Position)

8400_{hex}

- Bedeutung:** Der angegebene INTERBUS-Teilnehmer hat einen Fehler an der weiterführenden Schnittstelle (OUT2) gemeldet.
- Ursache:** Im unterlagerten Installationslokalbus liegt ein Fehler vor.
- Abhilfe:** Prüfen Sie die am angegebenen INTERBUS-Teilnehmer angeschlossenen Module. Sie finden den fehlerverursachenden Installationslokalbus-Teilnehmer indem Sie, beginnend beim letzten Modul die angegebene Anzahl an Teilnehmern abzählen.
- Add_Error_Info:** INTERBUS-Teilnehmernummer (Segment . Position)

3.4 Fehler-Codes zu Systemfehlern

0Fxx_{hex} (xx = beliebig)

Bedeutung: Auf der Anschaltbaugruppe ist ein Systemfehler (z. B. Hardware- oder Firmware-Fehler) aufgetreten.

Abhilfe: Tauschen Sie die Anschaltbaugruppe aus.

0FA4_{hex} (CTRL)

Bedeutung: Das Schreiben des Parametrierungsspeichers wurde abgebrochen.

Ursache: Auf dem Parametrierungsspeicher wurde ein Checksummenfehler festgestellt.

Abhilfe: Formatieren Sie den Parametrierungsspeicher mit einer Firmware-Version ≥ 4.12 .

0FC3_{hex} (CTRL)

Bedeutung: Unverträglichkeit mit dem Funktionsbaustein FB 14 beim automatischen Quittieren von Peripheriefehlern (PF-Fehlern) bei einer Firmware-Version ≤ 4.15

1010_{hex}

Ursache:

1. CPU ist nicht aufgesteckt.
2. Auf der Anschaltbaugruppe ist ein Firmware-Fehler aufgetreten.

Abhilfe:

1. Betreiben Sie Ihre Anschaltbaugruppe mit CPU.
2. Tauschen Sie die Anschaltbaugruppe aus.

1011_{hex}

Bedeutung: Auf der Anschaltbaugruppe ist ein Hardware-Fehler aufgetreten.

Abhilfe: Tauschen Sie die Anschaltbaugruppe aus.

1012_{hex}

- Bedeutung: In der Boot-Firmware liegt ein Fehler vor.
- Ursache:
1. Firmware hat einen Fehler.
 2. Beim Firmware Update ist der Download der neuen Firmware fehlgeschlagen.
 3. Es liegt ein Hardwarefehler vor.
- Abhilfe:
1. Firmware neu downloaden. Wird danach weiterhin die Fehlermeldung generiert, liegt ein Hardware-Fehler vor. Setzen Sie sich in diesem Fall mit Phoenix Contact in Verbindung.
 2. Führen Sie den Download erneut durch.
 3. Tauschen Sie die Anschaltbaugruppe aus.

1013_{hex}

- Bedeutung: In der Firmware liegt ein Fehler vor.
- Ursache:
1. Firmware hat einen Fehler.
 2. Beim Firmware-Update ist der Download der neuen Firmware fehlgeschlagen.
 3. Es liegt ein Hardware-Fehler vor.
- Abhilfe:
1. Firmware neu downloaden. Wird danach weiterhin die Fehlermeldung generiert, liegt ein Hardware-Fehler vor. Setzen Sie sich in diesem Fall mit Phoenix Contact in Verbindung.
 2. Führen Sie den Download erneut durch.
 3. Tauschen Sie die Anschaltbaugruppe aus.

1019_{hex} bis 101E_{hex} (MPM Manager Error)

- Bedeutung: Auf der Anschaltbaugruppe ist beim Zugriff auf das MPM ein Fehler aufgetreten.
- Abhilfe: Tauschen Sie die Anschaltbaugruppe aus.

1020_{hex} bis 1025_{hex} (Flash EEPROM Error)

- Bedeutung: Auf der Anschaltbaugruppe ist beim Zugriff auf den Parametrierungsspeicher ein Fehler aufgetreten.
- Abhilfe: Tauschen Sie die Anschaltbaugruppe aus.

1030_{hex} bis 1036_{hex} (Power on Selftest Error)

- Ursache:
1. Die SPS läuft, während die Anschaltbaugruppe zurückgesetzt wird.
 2. Auf der Anschaltbaugruppe ist während des Selbsttests ein Hardware-Fehler aufgetreten.
- Abhilfe:
1. Stoppen Sie erst die SPS bevor Sie die Anschaltbaugruppe zurücksetzen.
 2. Tauschen Sie die Anschaltbaugruppe aus.

1051_{hex} bis 1055_{hex} (RS-232-Error)

- Bedeutung:
- Auf der Anschaltbaugruppe ist beim Zugriff auf die Diagnose-Schnittstelle ein Fehler aufgetreten.
- Abhilfe:
- Tauschen Sie die Anschaltbaugruppe aus.

1056_{hex}

- Bedeutung:
- Der Firmware-Download wurde abgebrochen.
- Ursache:
- Zeitüberschreitung am Übertragungsprotokoll oder an der Diagnose-Schnittstelle
- Abhilfe:
- Prüfen Sie die Verbindung der Diagnose-Schnittstelle und des RS-232-Kabels und starten Sie danach die Übertragung erneut.

1057_{hex}

- Bedeutung:
- Der Firmware-Download wurde abgebrochen.
- Ursache:
- Das Programm hat einen Restart ausgelöst.
- Abhilfe:
- Führen Sie den Firmware-Download erneut durch.

1101_{hex} (Host Adaption Error)

- Bedeutung:
- Auf der Anschaltbaugruppe ist im hostspezifischen Teil der Firmware ein Fehler aufgetreten.
- Abhilfe:
- Tauschen Sie die Anschaltbaugruppe aus.

6342_{hex} (Bus-Error-Indication)

- Bedeutung: Im laufenden Betrieb des INTERBUS wurde von der Anschaltbaugruppe ein Fehler im angeschlossenen Bussystem erkannt.
- Ursache: Der Bus kann nicht weiter betrieben werden. Die Anschaltbaugruppe sucht den Fehlerort.
- Abhilfe: Lesen Sie die Fehlerursachen mit dem Dienst „Get_Error_Info“ (0316_{hex}) aus.

3.5 Fehler-Codes zu Anschaltbaugruppen mit Slaveteil

1210_{hex}

Bedeutung:	Es ist ein Firmware-Fehler aufgetreten.
Ursache:	Im INTERBUS-Ring befindet sich ein fehlerhafter Teilnehmer.
Abhilfe:	Tauschen Sie den Teilnehmer aus.
Add_Error_Info:	Anzahl der fehlerhaften Teilnehmer

1211_{hex}

Bedeutung:	Slave-Typ ist falsch eingestellt.
Ursache:	<ol style="list-style-type: none">1. Unerlaubte Slave-Nummer eingestellt.2. Es liegt ein Hardware-Fehler vor.
Abhilfe:	<ol style="list-style-type: none">1. Setzen Sie die richtige Slave-Nummer ein (0 oder 1).2. Tauschen Sie den Teilnehmer aus.
Add_Error_Info:	Fehlerhafte Slave-Nummer

1212_{hex}

Bedeutung:	Es ist ein Hardware-Fehler aufgetreten.
Ursache:	Im INTERBUS-Ring befindet sich ein Teilnehmer mit einem falschen Protokoll-Chip. Es sind nur Teilnehmer mit dem Protokoll-Chip SUPI 3 und höher erlaubt.
Abhilfe:	Entfernen Sie den Teilnehmer mit dem falschen Protokoll-Chip.

1213_{hex}

Ursache:	Der Slave wurde mit einer unerlaubten Anzahl von Worten initialisiert.
Abhilfe:	Verringern Sie die Anzahl der Worte, und initialisieren Sie den Slave erneut.

1215_{hex}

- Bedeutung: Fehler im Mikroprozessor-Mode
- Ursache:
1. Es wurden Datenlängen verwendet, die im Mikroprozessor-Mode nicht erlaubt sind (z. B. null Worte).
 2. Der Slave-Teil des Systemkopplers hat keinen korrekten ID-Code
- Abhilfe:
1. Ziehen Sie die externe Spannungsversorgung des Systemkopplers kurzzeitig ab, um den Slave-Teil des Systemkopplers neu zu initialisieren.
 2. Geben Sie einen korrekten ID-Code für den Slave-Teil ein.

1217_{hex}

- Bedeutung: Fehler bei der Initialisierung
- Ursache: Der INTERBUS-Protokoll-Chip wurde mehrfach initialisiert. Der INTERBUS-Protokoll-Chip SUP1 3 kann nur einmal initialisiert werden, wenn hardwaremäßig der ID-Code *NOT-READY* anliegt. Bei anderen ID-Codes kann der SUP1 3 mehrmals initialisiert werden.
- Abhilfe:
- Stellen Sie sicher, dass der Protokoll-Chip nur einmal initialisiert wird.
 - Schalten Sie den INTERBUS-Protokoll-Chip spannungsfrei, und versuchen Sie es erneut.

3.6 Fehler-Codes zu Anschaltbaugruppen mit COP-Karte

Fehler-Codes zu Anwender-Fehlern

1402_{hex}

Bedeutung:	Die Coprozessorkarte konnte den zuletzt aufgerufenen Dienst nicht bearbeiten.
Ursache:	Im Bootloader trat ein Zustandskonflikt auf. Z. B. wurde bei einer geöffneten Datei der Dienst „PC104_Download_Open_File_Request“ (0291 _{hex}) oder der Dienst „PC104_Download_Terminate_Request“ (0294 _{hex}) abgesetzt. Oder es wurde bei einer nicht geöffneten Datei der Dienst „PC104_Download_Send_File_Request“ (0292 _{hex}) oder der Dienst „PC104_Download_Close_File_Request“ (0293 _{hex}) abgesetzt.
Abhilfe:	Prüfen Sie den letzten und die vorherigen Dienstaufrufe.
Add_Error_Info:	Dienst, bei dem der Zustandskonflikt auftrat: 0001 _{hex} „PC104_Download_Initiate_Request“ (0290 _{hex}) 0002 _{hex} „PC104_Download_Open_File_Request“ (0291 _{hex}) 0003 _{hex} „PC104_Download_Send_File_Request“ (0292 _{hex}) 0004 _{hex} „PC104_Download_Close_File_Request“ (0293 _{hex}) 0005 _{hex} „PC104_Download_Terminate_Request“ (0294 _{hex})

1410_{hex}

Bedeutung:	Beim Öffnen einer Datei mit dem Dienst „PC104_Download_Open_File_Request“ (0291 _{hex}) trat ein Fehler auf.
Add_Error_Info:	Kennung, bei welcher Datei der Fehler auftrat: 0001 _{hex} Datei „bootld.ini“ 0002 _{hex} andere Datei

1411_{hex}

Bedeutung:	Beim Schreiben einer Datei mit dem Dienst „PC104_Download_Send_File_Request“ (0292 _{hex}) trat ein Fehler auf.
Add_Error_Info:	Immer 0001 _{hex}

1412_{hex}

Bedeutung: Beim Schließen einer Datei mit dem Dienst „PC104_Download_Close_File_Request“ (0293_{hex}) trat ein Fehler auf.

Add_Error_Info: Kennung, bei welcher Datei der Fehler auftrat:
0001_{hex} Datei „bootld.ini“
0002_{hex} andere Datei

1413_{hex}

Bedeutung: Beim Öffnen der Datei „bootld.ini“ trat ein Fehler auf.

Add_Error_Info: Immer 0001_{hex}

1414_{hex}

Bedeutung: Sie haben versucht mit dem Dienst „PC104_Download_Open_File_Request“ (0291_{hex}) eine Datei zu öffnen, die ungültig ist. Bei einem Firmware-Download muss als erstes immer die Datei „bootld.ini“ geöffnet werden. In dieser Datei müssen die Namen aller weiteren zu öffnenden Dateien eingetragen sein.

Abhilfe: Prüfen Sie die Reihenfolge der Dateien beim Download und die Einträge in der Datei „bootld.ini“.

Add_Error_Info: 0001_{hex} Erste Datei ist nicht die Datei „bootld.ini“.
0002_{hex} Datei ist eine Systemdatei.
0003_{hex} Dateiname ist nicht in Datei „bootld.ini“ eingetragen.

1420_{hex}

Bedeutung: Der Download der Firmware wurde abgebrochen.

Ursache: Der Download wurde durch Tastendruck auf PC-Tastatur abgebrochen.

Add_Error_Info: Abbruchposition im Bootloader:
0001_{hex} oder 0002_{hex}

1421_{hex}

Bedeutung: Der Download der Firmware wurde abgebrochen.

Ursache: Beim Download der Firmware ist ein Fehler aufgetreten.

Add_Error_Info: Immer 0000_{hex}

1422_{hex}

Bedeutung: Der Bootvorgang wurde abgebrochen.
Ursache: Die Baugruppe enthält keine Haupt-Firmware.
Add_Error_Info: Immer 0000_{hex}

1430_{hex}

Bedeutung: Die Checksummenüberprüfung wurde abgebrochen.
Ursache: Während der Checksummenüberprüfung konnte eine Datei nicht geöffnet werden.
Add_Error_Info: Position in der Firmware
0001_{hex} oder 0002_{hex}

1431_{hex}

Bedeutung: Die Checksummenüberprüfung wurde abgebrochen.
Ursache: Während der Checksummenüberprüfung trat beim Lesen einer Datei ein Fehler auf.
Add_Error_Info: Immer 0001_{hex}

1433_{hex}

Bedeutung: Die Checksumme der Boot-Firmware (DOS + Bootloader) ist falsch.
Add_Error_Info: Richtige Checksumme

1434_{hex}

Bedeutung: Die Checksumme der Haupt-Firmware ist falsch.
Add_Error_Info: Richtige Checksumme

1435_{hex}

Bedeutung: Der Bootvorgang wurde abgebrochen.
Ursache: Während der Checksummenüberprüfung wurde keine Haupt-Firmware gefunden.
Add_Error_Info: Immer 0001_{hex}

2002_{hex}

Bedeutung: Die Coprozessorkarte konnte den zuletzt aufgerufenen Dienst nicht bearbeiten.
Ursache: In der Firmware trat ein Zustandskonflikt auf.
Z. B. wurde bei einer geöffneten Datei der Dienst „PC104_Download_Open_File_Request“ (0291_{hex}) oder der Dienst „PC104_Download_Terminate_Request“ (0294_{hex}) abgesetzt. Oder es wurde bei einer nicht geöffneten Datei der Dienst „PC104_Download_Send_File_Request“ (0292_{hex}) oder der Dienst „PC104_Download_Close_File_Request“ (0293_{hex}) abgesetzt.

Abhilfe: Prüfen Sie den letzten und die vorherigen Dienstaufrufe.

Add_Error_Info: Dienst, bei dem der Zustandskonflikt auftrat:
0001_{hex} „PC104_Download_Initiate_Request“ (0290_{hex})
0002_{hex} „PC104_Download_Open_File_Request“ (0291_{hex})
0003_{hex} „PC104_Download_Send_File_Request“ (0292_{hex})
0004_{hex} „PC104_Download_Close_File_Request“ (0293_{hex})
0005_{hex} „PC104_Download_Terminate_Request“ (0294_{hex})

2010_{hex}

Bedeutung: Beim Öffnen einer Datei mit dem Dienst „PC104_Download_Open_File_Request“ (0291_{hex}) trat ein Fehler auf.

Add_Error_Info: Kennung, bei welcher Datei der Fehler auftrat:
0001_{hex} Datei „bootld.ini“
0002_{hex} andere Datei

2011_{hex}

- Bedeutung:** Beim Schreiben einer Datei mit dem Dienst „PC104_Download_Send_File_Request“ (0292_{hex}) oder mit dem Dienst „PC104_File_Transfer_Write_Request“ (02B6_{hex}) trat ein Fehler auf.
- Add_Error_Info:** Bei Fehler beim Dienst „PC104_Download_Send_File_Request“ (0292_{hex}) immer 0001_{hex}.
Bei Fehler beim Dienst „PC104_File_Transfer_Write_Request“ (02B6_{hex}) der Fehler-Code des zuständigen File System Treibers.

2012_{hex}

- Bedeutung:** Beim Schließen einer Datei mit dem Dienst „PC104_Download_Close_File_Request“ (0293_{hex}) trat ein Fehler auf.
- Add_Error_Info:** Kennung, bei welcher Datei der Fehler auftrat:
0001_{hex} Datei „bootld.ini“
0002_{hex} andere Datei

2013_{hex}

- Bedeutung:** Beim Öffnen der Datei „bootld.ini“ trat ein Fehler auf.
- Add_Error_Info:** Immer 0001_{hex}

2014_{hex}

- Bedeutung:** Es wurde versucht mit dem Dienst „PC104_Download_Open_File_Request“ (0291_{hex}) eine Datei zu öffnen, die ungültig ist. Bei einem Firmware-Download muss als erstes immer die Datei „bootld.ini“ geöffnet werden. In dieser Datei müssen alle weiteren zu öffnenden Dateien eingetragen sein.
- Abhilfe:** Prüfen Sie die Reihenfolge der Dateien beim Download und die Einträge in der Datei „bootld.ini“.
- Add_Error_Info:** 0001_{hex} Erste Datei ist nicht die Datei „bootld.ini“.
0002_{hex} Datei ist eine Systemdatei.
0003_{hex} Dateiname ist nicht in Datei „bootld.ini“ eingetragen.

2015_{hex}

- Bedeutung:** Fehler beim Löschen einer Datei, z. B. beim Löschen des Bootprojektes.

2020_{hex}

Bedeutung: Sie haben einen unbekanntem Message-Code verwendet.
Add_Error_Info: Unbekannter Message-Code

2021_{hex}

Bedeutung: Zugriffsverletzung!
Ursache: Z. B. Zugriff auf eine geschützte Variable mit falschem Passwort.
Abhilfe: Verwenden Sie das richtige Passwort.

2022_{hex}

Bedeutung: Beim Schreiben in eine INI-Datei ist ein Fehler aufgetreten.
Ursache: Es wurde versucht eine nicht vorhandene Variable zu löschen.

2023_{hex}

Bedeutung: Beim Lesen aus einer INI-Datei ist ein Fehler aufgetreten.
Ursache: Es wurde versucht eine nicht vorhandene Variable zu lesen.

2024_{hex}

Bedeutung: Beim Öffnen einer Datei ist ein Fehler aufgetreten.
Add_Error_Info: Fehler-Code des zuständigen File System Treibers.

2025_{hex}

Bedeutung: Beim Lesen einer Datei ist ein Fehler aufgetreten.
Add_Error_Info: Fehler-Code des zuständigen File System Treibers.

2026_{hex}

Bedeutung: Beim Schließen einer Datei ist ein Fehler aufgetreten.
Add_Error_Info: Fehler-Code des zuständigen File System Treibers.

2027_{hex}

Bedeutung: Bei einem Dienst für den File-Transfer ist ein ungültiger File Handle aufgetreten.

2028_{hex}

Bedeutung: Beim Dienst „PCP_Read_With_Name_Request“ (0098_{hex}) oder beim Dienst „PCP_Write_With_Name_Request“ (0097_{hex}) wurde eine ungültige Kommunikationsreferenz verwendet.

Add_Error_Info: Fehlerhafte Kommunikationsreferenz

2029_{hex}

Bedeutung: Beim Dienst „PC104_File_Transfer_ioctl_Request“ (02B8_{hex}) ist ein Fehler aufgetreten.

Add_Error_Info: Fehler-Codes der VxWorks-Funktion ioctl()

202A_{hex}

Bedeutung: Beim Dienst „PC104_File_Transfer_ioctl_Request“ (02B8_{hex}) ist die Anzahl der folgenden Bytes (Parameter *No_of_Bytes*) zu groß.

Add_Error_Info: Parameter *No_of_Bytes*

202B_{hex}

Bedeutung: Der Dienst „PC104_File_Transfer_ioctl_Request“ (02B8_{hex}) enthält eine nicht erlaubte Funktion in Parameter *Function*.

Add_Error_Info: Parameter *Function*

202C_{hex}

Bedeutung: Mit dem Dienst „PC104_File_Transfer_Open_Request“ (02B4_{hex}) konnte eine Datei nicht geöffnet werden.

Ursache: Die maximale Anzahl an offenen Dateien ist erreicht.

Abhilfe: Schließen Sie mindestens eine der offenen Dateien.

202D_{hex}

Bedeutung:	Ein Eintrag in der SVC-Datei ist fehlerhaft.
Ursache:	Zwischen zwei Rauten ('#' -Zeichen) steht weder das Schlüsselwort „CMD“ noch eine hexadezimale Zahl (0xXXXX).
Abhilfe:	Prüfen Sie die SVC-Datei.
Add_Error_Info:	Zeilennummer mit fehlerhaften Eintrag

202E_{hex}

Bedeutung:	Ein Eintrag in der SVC-Datei ist fehlerhaft.
Ursache:	Die SVC-Datei ist nicht logisch aufgebaut.
Abhilfe:	Prüfen Sie die SVC-Datei.
Add_Error_Info:	Zeilennummer mit fehlerhaften Eintrag

202F_{hex}

Bedeutung:	In einer SVC-Datei wird bei einem Dienst die maximale Anzahl an Parametern überschritten.
Abhilfe:	Prüfen Sie die SVC-Datei und reduzieren Sie die Parameteranzahl.
Add_Error_Info:	Zeilennummer mit fehlerhaften Eintrag

2030_{hex}

Bedeutung:	Eine negative Dienstbestätigung (Confirmation) wurde empfangen.
------------	---

Fehler-Codes zu Warnungen

2111_{hex}

Bedeutung:	Die Batteriespannung ist ausgefallen. Bei Netzausfall können Datum und Uhrzeit der Echtzeituhr sowie Retain-Daten verloren gehen.
Abhilfe:	Wechseln Sie die Batterie.

2112_{hex}

Bedeutung: Die IP-Adresse wurde noch nicht parametrierung oder sie ist gleich „0.0.0.0“.
Abhilfe: Geben Sie eine IP-Adresse ein.

2113_{hex}

Bedeutung: Die Datei „vxwusr.ini“ ist defekt.
Ursache: Beispielsweise:
– Fehler in der Checksumme oder
– keine Checksumme vorhanden.

Fehler-Codes zu Laufzeitfehlern

Fehler, die zur Laufzeit des IEC-61131-Laufzeitsystems auftreten, werden in der Fehlerklasse 22xx_{hex} beschrieben.

2211_{hex}

Bedeutung: Stack-Überlauf in einer Task vom IEC-61131-Laufzeitsystems.
Abhilfe: Vergrößern Sie den Stack.

2212_{hex}

Bedeutung: In einer User Task des IEC-61131-Laufzeitsystems wurde der Speicherbereich bei einem Array überschritten.
Abhilfe: Prüfen Sie den Zugriff auf das Array und vergrößern Sie eventuell die Dimension des Arrays.

2214_{hex}

Bedeutung: In einer User Task vom IEC-61131-Laufzeitsystem wurde durch Null dividiert.
Abhilfe: Prüfen Sie die User Task.

2216_{hex}

Bedeutung: Bei einer Fließkomma-Berechnung in einer User Task vom IEC-61131-Laufzeitsystem wurde der Wertebereich überschritten.

Abhilfe: Prüfen Sie die User Task.

2217_{hex}

Bedeutung: Die Task Watchdog einer Task vom IEC-61131-Laufzeitsystem greift ein, weil die Ausführungszeit zu lang ist.

Abhilfe: Prüfen Sie die Task vom IEC-61131-Laufzeitsystem.

2218_{hex}

Bedeutung: Die Laufzeit einer Task vom IEC-61131-Laufzeitsystem ist zu lang. Tasks mit niedriger Priorität (z. B. Kommunikationstasks) bekommen keine Prozessorzeit mehr.

Abhilfe: Prüfen Sie die Task.

2219_{hex}

Bedeutung: Sie haben einen Funktionsbaustein aufgerufen, der in der Firmware oder im IEC-61131-Laufzeitsystem nicht existiert.

221A_{hex}

Bedeutung: Ein Programm des IEC-61131-Laufzeitsystem wurde aus dem Programm heraus gestoppt.

221B_{hex}

Bedeutung: Ein unerwarteter Breakpoint ist aufgetreten.

221C_{hex}

Bedeutung: Eine interne Exception ist aufgetreten.

221D_{hex}

Bedeutung: Im Programm ist ein String-Fehler aufgetreten.

221E_{hex}

Bedeutung: – In einer User Task vom IEC-61131-Laufzeitsystem wurde durch Null dividiert oder
– die Task Watchdog einer Task vom IEC-61131-Laufzeitsystem greift ein, weil die Ausführungszeit zu lang ist.

Abhilfe: Prüfen Sie die Task.

Fehler-Codes zu fatalen Fehler

Fatale Fehler werden in den Fehlerklassen 23xx_{hex}, 24xx_{hex} und 26xx_{hex} beschrieben. Setzen Sie sich bitte mit Phoenix Contact in Verbindung, wenn fatale Fehler öfter auftreten.

Die Fehlerklasse 23xx_{hex} enthält Fehler, die im DDI (Device Driver Interface) auftreten. Das niederwertige Byte dieses Fehler-Codes ist der Fehler-Code des DDI. Für Informationen über den Fehler-Code des DDI lesen Sie bitte das Treiber-Referenzhandbuch für PC-Anschaltbaugruppen (Art.-Nr. 27 45 12 7)

Die Fehlerklasse 24xx_{hex} enthält fatale Fehler, die bei der Firmware auftreten.

2410_{hex}

Bedeutung: Der Software-Watchdog des COP hat eingegriffen.

2411_{hex}

Bedeutung: Beim Initialisieren des IEC-61131-Laufzeitsystems ist ein Fehler aufgetreten.

2412_{hex}

Bedeutung: Beim Initialisieren der seriellen Schnittstelle COM1 für das IEC-61131-Laufzeitsystem ist ein Fehler aufgetreten.

2413_{hex}

Bedeutung: Beim Initialisieren der seriellen Schnittstelle COM2 für das IEC-61131-Laufzeitsystem ist ein Fehler aufgetreten.

2414_{hex}

Bedeutung: Beim Initialisieren der Ethernet-Verbindung für das IEC-61131-Laufzeitsystem ist ein Fehler aufgetreten.

2415_{hex}

Bedeutung: Beim Initialisieren der Kommunikation zwischen dem IEC-61131-Laufzeitsystem und dem Message Handler ist ein Fehler aufgetreten.

2416_{hex}

Bedeutung: Beim Initialisieren des IEC-61131-Laufzeitsystem-Treibers „IBS_IO“ ist ein Fehler aufgetreten.

2417_{hex}

Bedeutung: Beim Initialisieren des IEC-61131-Laufzeitsystem-Treibers „PSTD_IO“ ist ein Fehler aufgetreten.

2418_{hex}

Bedeutung: Beim Initialisieren des IEC-61131-Laufzeitsystem-Exception Handlers ist ein Fehler aufgetreten.

2419_{hex}

Bedeutung: Es steht kein Speicher für die Systemmerker des IEC-61131-Laufzeitsystems zur Verfügung.

241A_{hex}

Bedeutung: Beim Anmelden des Speichers für die Systemmerker an das IEC-61131-Laufzeitsystem ist ein Fehler aufgetreten.

2420_{hex}

Bedeutung: Bei Geräten, bei denen der Flash Speicher in zwei Laufwerke aufgeteilt sein muss, haben die Laufwerke entweder die falsche Größe, oder das zweite Laufwerk ist nicht vorhanden.

2421_{hex}

Bedeutung: Die Datei „vxworks.ini“ ist defekt.

2422_{hex}

Bedeutung: Fehler im IB-Loader

Ursache: Die angegebene SVC-Datei kann nicht gelesen werden.

2423_{hex}

Bedeutung: Fehler im IB-Loader

Ursache: Die Timeout-Zeit ist abgelaufen und es wurde keine Dienstbestätigung (Confirmation) empfangen.

2425_{hex}

Bedeutung: Fehler im IB-Loader

Ursache: Es wurde eine Dienstmeldung (Indication) empfangen obwohl eine Dienstquittung (Confirmation) erwartet wurde.

2426_{hex}

Bedeutung: Fehler im IB-Loader

Ursache: Beim Warten auf eine Dienstquittung (Confirmation) wurde eine ungültige Meldung (weder Dienstmeldung noch Dienstquittung) empfangen.

2427_{hex}

Bedeutung: Fehler im IB-Loader

Ursache: Die Timeout ist abgelaufen, bevor das Busy-Flag von IEC-61131-Laufzeitsystem gesetzt wurde.

2428_{hex}

Bedeutung:	Fehler im IB-Loader
Ursache:	Beim Öffnen eines Datenkanals zu einem Node ist ein Fehler aufgetreten. Lesen Sie für weitere Informationen den Abschnitt über die DDI-Routine DDI_DevOpenNode() im Treiber-Referenzhandbuch für PC-Anschaltbaugruppen (Art.-Nr. 27 45 12 7).
Add_Error_Info	Fehler-Code der DDI-Routine DDI_DevOpenNode()

2429_{hex}

Bedeutung:	Fehler im IB-Loader
Ursache:	Beim Senden einer Meldung oder Kommandos zu einer Mailbox ist ein Fehler aufgetreten. Lesen Sie für weitere Informationen den Abschnitt über die DDI-Routine DDI_MXI_SndMessage() im Treiber-Referenzhandbuch für PC-Anschaltbaugruppen (Art.-Nr. 27 45 12 7).
Add_Error_Info	Fehler-Code der DDI-Routine DDI_MXI_SndMessage()

242A_{hex}

Bedeutung:	Fehler im IB-Loader
Ursache:	Beim Empfangen einer Meldung oder Kommandos aus einer Mailbox ist ein Fehler aufgetreten. Lesen Sie für weitere Informationen den Abschnitt über die DDI-Routine DDI_MXI_RcvMessage() im Treiber-Referenzhandbuch für PC-Anschaltbaugruppen (Art.-Nr. 27 45 12 7).
Add_Error_Info	Fehler-Code der DDI-Routine DDI_MXI_RcvMessage()

242B_{hex}

Bedeutung:	Fehler im IB-Loader
Ursache:	Beim Aktivieren des Notification-Modes für einen Datenkanal ist ein Fehler aufgetreten. Lesen Sie für weitere Informationen den Abschnitt über die DDI-Routine DDI_SetMsgNotification() im Treiber-Referenzhandbuch für PC-Anschaltbaugruppen (Art.-Nr. 27 45 12 7).
Add_Error_Info	Fehler-Code der DDI-Routine DDI_SetMsgNotification()

26xx_{hex}

Bedeutung:

Beim Prozessor ist eine Exception aufgetreten. Das niederwertige Byte enthält die Vektor-Nummer der Exception.

A 1 Abbildungsverzeichnis

Kapitel 1

Bild 1-1:	Kommunikationsmechanismen im MPM	1-12
Bild 1-2:	Diagnose-Statusregister	1-16
Bild 1-3:	Inhalt des Diagnose-Parameterregisters (Beispiel) ...	1-18
Bild 1-4:	Zuordnung von häufig genutzten Funktionen im Standardfunktions-Startregister	1-19
Bild 1-5:	Ablauf einer Funktionsausführung ohne Parameterübergabe	1-20
Bild 1-6:	Ablauf einer Funktionsausführung mit Parameterübergabe	1-21
Bild 1-7:	Die Zustandsmaschine	1-22
Bild 1-8:	Dienstgruppenübersicht	1-25
Bild 1-9:	Busaufbau (angeschlossene und aktive Konfiguration)	1-29
Bild 1-10:	Aufbau der Teilenummern	1-32
Bild 1-11:	Teilnehmerorientierte und listenorientierte Übertragung	1-33
Bild 1-12:	Busaufbau (Gruppen und Alternativen)	1-34
Bild 1-13:	Zusammensetzung des Längen-Codes	1-36
Bild 1-14:	Prozessdatenkanal eines Frequenzumrichters	1-40
Bild 1-15:	Broadcast	1-41
Bild 1-16:	Durchgangsverbindung	1-41
Bild 1-17:	Dienste zur Konfiguration der Anschaltbaugruppe	1-43
Bild 1-18:	Laden eines Konfigurationsrahmens	1-47
Bild 1-19:	Laden der Prozessdatenbeschreibungsliste	1-48
Bild 1-20:	Dienstaufbau Prozessdatenbeschreibungsliste	1-50
Bild 1-21:	Byte-IN- und OUT-Prozessdaten	1-51
Bild 1-22:	Laden der Prozessdatenreferenzliste	1-52

Bild 1-23:	Die Prozessdatenzuordnung	1-53
Bild 1-24:	Struktur des Dienstes „Load_Process_Data_Reference_List“	1-54
Bild 1-25:	Dienste zur Systemzustandssteuerung	1-58

Kapitel 3

Bild 3-1:	Kein Schnittstellenfehler aufgetreten	3-67
Bild 3-2:	Fehler an weiterführender Fernbus-Schnittstelle	3-67
Bild 3-3:	Fehler an weiterführender Lokalbus-Schnittstelle	3-68

A 2 Tabellenverzeichnis

Kapitel 1

Tabelle 1-1:	Gegenüberstellung von G3-Diensten und G4-Diensten	1-8
Tabelle 1-2:	Fehler mit Busabschaltung	1-17
Tabelle 1-3:	Fehler ohne Busabschaltung	1-17
Tabelle 1-4:	Prinzipieller Aufbau eines Konfigurationsrahmens	1-28
Tabelle 1-5:	Konfigurationsrahmen.....	1-29
Tabelle 1-6:	Konfigurationsrahmen.....	1-35
Tabelle 1-7:	Parameter der Prozessdatenreferenzliste	1-54

Kapitel 2

Tabelle 2-1:	Übersicht der Dienste (sortiert nach Kommando-Codes).....	2-5
Tabelle 2-2:	Selbständige Fehleranzeigen (Indications).....	2-7
Tabelle 2-3:	Systemparameter	2-18

A 3 Erklärung der Abkürzungen

CR	Communication Reference Kommunikationsreferenz
CRC	Cyclic Redundancy Check Zyklische Redundanzprüfung
CRL	Communication Relationship List Kommunikationsbeziehungsliste
DDI	Device-Driver-Interface
DPM	Dual-Port-Memory
DTA	Data-Transmission-Area
DTI	Data-Transmission-Interface Schnittstelle zum DTA
G4	Generation 4
IB, IBS	INTERBUS
IN-PD	IN Process Data Eingangs-Prozessdaten
MAU	Medium Attachment Unit
MMS	Manufacturing Message Specification
MPM	Multi-Port-Memory
MXA	Mailbox Area
MXI	Mailbox-Interface Schnittstelle zum MXA
LB	Local Bus Lokalbus

Erklärung der Abkürzungen

OD	Object Dictionary Objektverzeichnis
OUT-PD	OUT Process Data Ausgangs-Prozessdaten
PCP	Peripherals Communication Protocol
PD	Process Data Prozessdaten
PDD	Process Data Description Prozessdatenbeschreibung
PDDL	Process Data Description List Prozessdatenbeschreibungsliste
PDR	Process Data Reference Prozessdatenreferenz
PDRL	Process Data Reference List Prozessdatenreferenzliste
RB	Remote Bus Fernbus
SGA	Signal Area
SGI	Signal Interface Schnittstelle zum SGA
SSGI	Standard-Signal-Interface
XSGI	Extended Signal Interface Erweitertes Signal-Interface

A 4 Erklärung der Fachwörter

Action Object	Ein Action Object (Aktions-Objekt) ist ein mit Parametern belegter Dienst oder eine vordefinierte Dienstsequenz.
Anschaltbaugruppe	Die Anschaltbaugruppe verbindet speicherprogrammierbare Steuerungen (SPS) oder Computer-Systeme (PC, VMEbus usw.) mit dem Sensor-/Aktorbus INTERBUS. Sie übernimmt im INTERBUS-System die Master-Funktion. Sie steuert den Datenverkehr auf dem INTERBUS unabhängig vom Steuerungs- oder Rechnersystem, in dem sie eingebaut ist.
aktive Konfiguration	Die aktive Konfiguration ist die Parametrierung, mit der die Anschaltbaugruppe den aktuellen Busaufbau betreibt (der Bus ist im Zustand ACTIVE oder RUN) wobei der vollständige Busaufbau bekannt ist.
aktueller Busaufbau	Der aktuelle Busaufbau ist der physikalische Busaufbau der momentan von der Anschaltbaugruppe betrieben wird.
Alternative	Eine Alternative ist ein zusammenhängender Konfigurationsteil, der alternativ zu anderen Konfigurationsteilen an eine bestimmte Busklemme angeschlossen werden kann. Sie wird in das niederwertige Byte der Gruppennummer eingetragen (siehe auch Gruppe).
Ausgangs-Prozessdaten	Die Ausgangs-Prozessdaten sind die Teilmenge der Ausgangsdaten, die zyklisch von der INTERBUS-Anschaltbaugruppe zu den INTERBUS-Teilnehmern übertragen werden.
Bussegment	Ein Bussegment besteht aus einem Fernbus-Teilnehmer einschließlich der daran angeschlossenen E/A-Module. Das davorliegende Kabel gehört mit zum Segment.
Busteilnehmer	→ INTERBUS-Teilnehmer
Client	Ein Client ist ein Kommunikationsteilnehmer, der einen Dienst von einem Server anfordert.
Client-Server-Modell	Dieses Modell definiert die Kommunikationsmechanismen zwischen einem Dienstanforderer (Client) und einem Diensterbringer (Server). Durch diese Kommunikationsmechanismen kann der Client die Funktionen des Servers nutzen. Mit Kommunikationsdiensten kann auf die Funktion des Servers zugegriffen werden.

Confirmation	Eine Confirmation ist eine Dienstanwort des Server auf einen Request des Client. Die Confirmation wird vom Server als Response gesendet.
Data Transmission Area	Datenbereich im Multi-Port-Memory zur Kommunikation zwischen einem Host und einer Anschaltbaugruppe.
Eingangs-Prozessdaten	Die Eingangs-Prozessdaten sind die Teilmenge der Eingangsdaten, die zyklisch von den INTERBUS-Teilnehmern zur INTERBUS-Anschaltbaugruppe übertragen werden.
Event Object	Ein Event Object (Ereignis) ist eine mit Parametern belegte Dienstfolge, die durch die Dienstprimitive „Indication“ gestartet wird. Die Indication, die den Start des Ereignisses veranlasst, muss vom Anwender bestimmt werden.
Gruppe	INTERBUS-Teilnehmer können vom Anwender zu Gruppen zusammengefasst werden. Jeder Teilnehmer darf nur einer Gruppe zugeordnet werden. Diese Gruppen können programmgesteuert ab- und wieder zugeschaltet werden. Beim Abschalten einer Gruppe wird immer der komplette Lokalbus, in dem diese Gruppe vorkommt, abgeschaltet. Ab Firmware-Generation 4 ist auch das Ab- und Zuschalten von Fernbus-Teilnehmern möglich. Dazu ist aber nicht mehr die Gruppeneffinition notwendig, da beim Schalten von Teilnehmern nur noch deren logische Teilnehmernummer anzugeben ist.
Gruppennummer	Teilnehmer können unter der Gruppennummer zu einer Gruppe zusammengefasst werden. Die Gruppennummer besteht aus der Gruppe (höherwertiges Byte) und der Alternative (niederwertiges Byte).
Host	Host ist die Bezeichnung für das Steuerungs- oder Rechnersystem, in das die Anschaltbaugruppe integriert wird.
ID-Code	Jeder INTERBUS-Teilnehmer besitzt einen ID-Code (Identifikations-Code), damit er durch die Anschaltbaugruppe identifiziert werden kann. Der ID-Code gibt die Art des Teilnehmers an. Er gibt Auskunft darüber, ob es sich um ein analoges bzw. digitales Modul oder eine Busklemme handelt, ob es ein Ein- oder Ausgabemodul ist und ob es sich um einen PCP-Teilnehmer handelt. Er belegt das niederwertige Byte des Teilnehmer-Codes.
Indication	Eine Indication ist eine vom Server empfangene Dienstanforderung auf einen Request des Client. Der Server antwortet auf die Indication mit einer Response.

INTERBUS	Der INTERBUS ist ein nach IEC 61158 genormter Feldbus zur seriellen Übertragung von Daten auf der Sensor-/Aktorebene.
INTERBUS-Gerät	Alle komplexen technischen Einrichtungen, die für die Datenübertragung über INTERBUS verwendet werden. Ein INTERBUS-Gerät kann mehrere INTERBUS-Teilnehmer enthalten.
INTERBUS-Teilnehmer	Ein INTERBUS-Teilnehmer ist Teil eines INTERBUS-Gerätes und am Datenaustausch im INTERBUS-System beteiligt. Jeder Teilnehmer hat genau einen Protokoll-Chip. Die Teilnehmer werden durch den Teilnehmer-Code charakterisiert. Es gibt auch INTERBUS-Geräte, die mehrere Teilnehmer enthalten.
Kommunikationsbeziehung	Die Kommunikationsbeziehung stellt bei PCP-Kommunikation die logische Verbindung zwischen zwei Teilnehmern her. Voraussetzung dafür ist die physikalische Möglichkeit zur Kommunikation, das heißt die beiden Teilnehmer müssen über das Netz miteinander verbunden und PCP-fähig sein. Für jede Kommunikationsbeziehung sind in der Kommunikationsbeziehungsliste Angaben abgelegt.
Kommunikationsbeziehungsliste	Eine Kommunikationsbeziehungsliste ist eine Liste für die PCP-Kommunikation, in der die Verbindungsparameter der Kommunikationsbeziehung zwischen zwei Teilnehmern abgelegt sind. Bei einem Verbindungsaufbau wird geprüft, ob die Verbindungsparameter in den Kommunikationsbeziehungslisten der beiden Teilnehmer zusammenpassen. Die relevanten Verbindungsparameter sind die Sende- und Empfangspuffergröße sowie die unterstützten PCP-Dienste. Anstelle von Verbindungsparametern spricht man auch von passenden Kontextbedingungen. Die Kommunikationsbeziehungsliste eines Teilnehmers enthält die Beschreibung aller Kommunikationsbeziehungen dieses Teilnehmers unabhängig vom Zeitpunkt der Nutzung.
Kommunikationsreferenz (CR)	Die Kommunikationsreferenz ist eine Nummer, die jedem PCP-Teilnehmer zugeordnet wird. Sie bezeichnet die Adresse der logischen Verbindung. Die INTERBUS-Anschaltbaugruppe hat immer die Kommunikationsreferenz 1. Der Anwender kann die Kommunikationsreferenzen ab 2 aufeinanderfolgend vergeben.
Konfigurationsrahmen	Der Konfigurationsrahmen enthält die gesamte Parametrierung der Anschaltbaugruppe inklusive der Gruppen und Alternativen. Der Konfigurationsrahmen enthält alle Teilnehmer des vollständigen Busaufbaus.

Längen-Code	Der Längen-Code gibt die Anzahl und die Darstellungsform der Prozessdaten (Bit, Nibble, Byte, Wort) an. Er belegt das höherwertige Byte des Teilnehmer-Codes.
Mailbox Area	Mailbox-Bereich im Multi-Port-Memory zur Kommunikation zwischen einem Host und einer Anschaltbaugruppe.
Manufacturing Message Specification (MMS)	ISO-Standard von Kommunikationsdiensten, mit denen administrative Aufgaben, Identifikations- und Statusabfragen, kommunikationsbezogene Aktivitäten sowie die Übertragung von Nutzdaten abgewickelt werden. MMS wurde für Netzwerke entwickelt, die hierarchisch über der Sensor-/Aktor-Ebene angeordnet sind.
Objektverzeichnis	Das Objektverzeichnis ist ein Verzeichnis aller Kommunikationsobjekte eines Teilnehmers, das sämtliche Angaben über jedes einzelne Objekt enthält. Über den Index, der jedem Objekt zugeordnet ist, können andere Teilnehmer auf ein Objekt zugreifen.
Peripherals Communication Protocol (PCP)	Das Peripherals Communication Protocol (PCP) gehört zum INTERBUS-Protokoll und steuert die Übertragung von Parameterdaten. Dazu stehen spezielle PCP-Dienste zur Verfügung.
PCP-Dienst	Ein PCP-Dienst ist ein Dienst zum Verbindungsaufbau und -abbau sowie zum Datenaustausch zwischen zwei Teilnehmern.
PCP-Objekt	Ein PCP-Objekt ist ein strukturierter Speicherbereich für Daten, die zwischen zwei Teilnehmern ausgetauscht werden, z. B. Messwerte, Programmteile, Geräteparameter usw. Das Kommunikationsobjekt ist im Objektverzeichnis eines Teilnehmers beschrieben. Auf die PCP-Objekte kann von anderen Kommunikationsteilnehmern zugegriffen werden.
Peripherals Message Specification (PMS)	Peripherals Message Specification (PMS) ist eine speziell auf den Sensor-/Aktorbereich zugeschnittene Untermenge der MMS-Kommunikationsdienste.
physikalische Teilnehmerposition	Die physikalische Teilnehmerposition (physical device position) ist die Position des Busteilnehmers im Summenrahmen. Dem ersten Busteilnehmer wird die Position „0“ zugewiesen. Die physikalische Teilnehmerposition entspricht der physikalischen Gerätenummer, wenn der gesamte Konfigurationsrahmen aktiv ist.
Protokoll	Ein Protokoll ist ein Satz von Konventionen für die Kommunikation zwischen Geräten oder Prozessen.

Prozessdaten	Prozessdaten sind Eingangs- und Ausgangsinformationen von und zu INTERBUS-Teilnehmern, die sich laufend ändern und ständig aktualisiert werden müssen. Sie werden in jedem Buszyklus über den Prozessdatenkanal übertragen (siehe auch Parameterdaten).
Prozesseingangsdaten	→ Eingangs-Prozessdaten
Prozessausgangsdaten	→ Ausgangs-Prozessdaten
Request	Ein Request ist ein Dienstauftrag des Client beim Server. Der Client erhält als Antwort eine Confirmation. Der Request wird vom Server als Indication empfangen.
Response	Eine Response ist eine Dienstantwort des Servers auf eine Indication des Client. Die Dienstantwort wird vom Client als Confirmation empfangen.
Server	Ein Server ist ein Kommunikationsteilnehmer, der auf einen Dienst von einem Client antwortet. Der Server stellt seine Objekte einem anderen Teilnehmer über einen Dienst zur Verfügung.
Signal Area	Datenbereich im Multi-Port-Memory zur Kommunikation zwischen einem Host und einer Anschaltbaugruppe.
Signal Object	Die Signal Objects (Signalobjekte) enthalten die Parametersätze für Dienstfolgen.
Slave	Ein Slave ist ein Teilnehmer im Netz der nur nach Ansprache durch den Master am Datenaustausch teilnehmen kann.
Standard-Signal-Interface	Das Standard-Signal-Interface ist eine Schnittstelle für den vereinfachten Zugriff auf Dienste (speziell für SPSSen).
Summenrahmen	Der Summenrahmen ist ein Übertragungsprotokoll, bei dem alle physikalischen INTERBUS-Teilnehmer wie ein einziger logischer Teilnehmer behandelt werden. Sämtliche Prozessdaten werden in einem Zyklus gleichzeitig von allen Teilnehmern übernommen und an alle Teilnehmer übertragen. Anhand der Lage der Informationen im Summenrahmen kann jeder INTERBUS-Teilnehmer die Daten übernehmen, die für ihn bestimmt sind.
Teilnehmer	→ INTERBUS-Teilnehmer

Teilnehmer-Code	Der Teilnehmer-Code ist ein Datenwort zur Kennzeichnung der Eigenschaften eines INTERBUS-Teilnehmers. Er besteht aus Längen-Code (höherwertiges Byte) und ID-Code (niederwertiges Byte).
Teilnehmernummer	Beim INTERBUS gibt es logische Teilnehmernummern und physikalische Teilnehmernummern.
Teilnehmernummer, logische	Jeder INTERBUS-Teilnehmer eines Konfigurationsrahmens bekommt eine eindeutige logische Teilnehmernummer zugewiesen. Diese Teilnehmernummer wird in der Form „Segment.Position“ (Seg.Pos) angegeben. Die logische Teilnehmernummer 0.0 ist für die Anschaltbaugruppe reserviert. Die Nummern „1.0“ bis „254.254“ können vergeben werden. Jeder Fernbus-Teilnehmer erhält die Positionsnummer 0. Jeder Lokalbus-Teilnehmer hat die Segmentnummer des zugeordneten Fernbus-Teilnehmers. Die logische Teilnehmernummer belegt ein Datenwort (16 Bit). Die Bussegmentnummer wird auf die Bits 8 ... 15 abgebildet, die Teilnehmerposition auf die Bits 0 ... 7.
Teilnehmernummer, physikalische	Die physikalische Teilnehmernummer kennzeichnet die durch den physikalischen Aufbau des Bussystems festgelegte Reihenfolge der Teilnehmer. Sie wird von 1 aufsteigend lückenlos bis 512 vergeben. Die physikalische Teilnehmernummer ist unabhängig davon, ob einzelne Bussegmente abgeschaltet sind oder nicht.

A 5 Stichwortverzeichnis

A

Action-Object-Verwaltung	1-26
Alarm_Stop	2-100
Allgemeine Funktionen.....	1-26
Alternative Teilnehmer	1-34
Anlaufverhalten	1-7
Anschaltbaugruppe zurücksetzen	2-100, 2-110
Anwenderfehler	1-55
Auslesen	
Diagnose-Informationen	2-124
Dienstfolge.....	2-139
Ereignisbeschreibung	2-169
Parametersatz	2-154
Prozessdatenbeschreibung	2-76
Prozessdatenreferenz.....	2-88
Statusinformationen.....	2-116
Automatische Adresszuweisung	1-42

B

Betriebsart	
Asynchron.....	1-13
Synchron.....	1-13
Broadcast-Funktion	1-40
Bus_Error-Fehleranzeige	2-207
Busabschaltzeit	2-21
Busebene	1-27
Buseinzelfehler.....	1-56
Busfehler	1-56
Busqualität	1-56
Bussteuerung	1-26, 1-57

Buswarnzeit	2-21
-------------------	------

D

Data Transmission Area	1-12
Datei	
lesen	2-195
löschen	2-186, 2-198
öffnen.....	2-179
Datei schließen	2-182
Datei schreiben	2-189
Datenkonsistenz	2-92
Datentransfer starten	2-98
Datentransfer stoppen	2-102
Deaktivieren von Konfigurationsrahmen	2-66
Device Driver Interface	1-13
Device_Fail-Fehleranzeige	2-206
Diagnose	
Allgemein.....	1-56
Behandlung	1-26
Register	1-15
Diagnose steuern.....	2-131
Diagnose-Informationen auslesen	2-124
Diagnose-Register aktualisieren	2-111
Dienstfolge	
auslesen	2-139
löschen	2-142
übertragen	2-174
zusammenstellen.....	2-135
Dual-Port-Memory.....	1-12
Durchgangsverbindung.....	1-41

E

Ereignisbeschreibung	
auslesen	2-169
löschen	2-172
projektieren	2-162
Erzeugen von Konfigurationsrahmen....	2-62
Exklusiver Dienst.....	2-13

F

Fault-Fehleranzeige	2-204, 2-205
Fehleranzeige	
Bus_Error.....	2-207
Device_Fail.....	2-206
Fault.....	2-204, 2-205
Fehlerbehandlung	1-26
Fehler-Codes anzeigen.....	2-204, 2-205
Fehlerort auslesen	2-113
Fehlerursache auslesen.....	2-113
Fehlerverwaltung.....	1-55
Formatieren Parametrierungsspeicher	2-176
Freischalten einer Schnittstelle	2-15
Funktion definieren.....	2-133

G

Gerätstatusmeldungen.....	1-55
Gruppennummern, logische.....	1-28, 1-34
Gruppenzugehörigkeit.....	1-34

I

ID-Code.....	1-27
Initial-Zustand.....	1-23

K

Kommunikationsbeziehungsliste.....	1-30
------------------------------------	------

Konfiguration

aktiv	1-28
angeschlossen.....	1-28
Dienste	1-43
Zustandsbeschreibung	1-43
Konfigurationsrahmen.....	1-27
aktivieren	2-62
auslesen	2-38
automatisches Laden	1-45
Configuration_Entry.....	2-33, 2-44
deaktivieren	2-66
erzeugen.....	2-62
laden.....	1-43
löschen	2-60
manuell erzeugen	1-46
übertragen	2-31, 2-54
vergleichen	2-51
Verwaltung.....	1-25, 1-45
Verzeichnis lesen	2-57

L

Laden einer Prozessdatenreferenz.....	2-91
Längen-Code	1-27
Lesen einer Datei.....	2-195
Logische Gruppennummern	1-34
Löschen	
Datei	2-186, 2-198
Dienstfolge.....	2-142
Ereignisbeschreibung	2-172
Konfigurationsrahmen	2-60
Parametersatz	2-158

M

Mailbox Area.....	1-12
-------------------	------

Mailbox-Interface.....	1-13	Prozessdaten	1-39
Multi-Port-Memory.....	1-12	anwenderdefiniert	1-39
N		Kanal	1-39
Nodes.....	1-12	Objekte	1-39
O		Referenzliste.....	1-39
Öffnen einer Datei	2-179	Verknüpfung	1-23, 1-39
P		Verwaltung.....	1-26, 1-39, 1-48
Parametersatz.....	2-133	Zuordnung	1-39
auslesen	2-154	Prozessdatenbeschreibung	1-50, 2-71
löschen	2-158	auslesen	2-76
projektieren.....	2-146	definieren.....	2-70
übertragen	2-174	Prozessdatenbeschreibungsliste	1-23
Parametrierung der Anschaltbaugruppe	2-11	definieren.....	1-50
Parametrierungsphase.....	1-23, 2-11	laden.....	1-44
Parametrierungsspeicher		Prozessdaten-Interface.....	1-13
Datei lesen.....	2-195	Prozessdatenreferenz	
Datei löschen	2-186, 2-198	Ausgangsdaten.....	2-83
Datei öffnen	2-179	auslesen	2-88, 2-95
Datei schließen.....	2-182	Durchgangsdaten	2-83
Datei schreiben.....	2-189	Eingangsdaten.....	2-83
Datei-Zeiger positionieren.....	2-192	festlegen	2-82
formatieren.....	2-176	laden.....	2-91
PD-Objekte.....	1-39	Prozessdatenreferenzliste	1-23
Positionieren von Datei-Zeiger.....	2-192	erstellen.....	1-44
PowerOn-Selbsttest	1-7	laden.....	1-52
Projektieren		Parameter.....	1-54
Ereignisbeschreibung	2-162	Regeln beim Eintragen	1-53
Parametersatz	2-146	R	
		Request.....	1-14
		Reset.....	2-100, 2-110
		Response.....	1-14

S

Schalten von Teilnehmer	2-104
Schließen einer Datei	2-182
Schnittstellen sperren/freischalten	2-15
Schreiben einer Datei	2-189
Signal Area	1-12
Signal-Object-Verwaltung	1-26
Sperren einer Schnittstelle	2-15
Standardfunktionsregister	1-18
Starten des Datentransfers	2-98
Statusinformationen auslesen	2-116
Steuern	
Diagnose	2-131
Teilnehmer	2-107
Stoppen des Datentransfers	2-102
Systemfehler	1-55
Systemparameter	
auslesen	2-25
Liste	2-18
setzen	2-23
Systemzustandssteuerung	1-57

T

Teilnehmer

Alternativ	1-34
an-/abschalten	2-104
Fehlermeldung anzeigen	2-206
steuern	2-107

Teilnehmernummern

Aufbau	1-32
logische	1-31
physikalische	1-31

V

Versionsinformationen	2-120
Verwalten	
Konfigurationsrahmen	1-25
Prozessdaten	1-26

Z

Zuordnen von Prozessdaten	1-39
Zustandsmaschine	1-22
Zustandssteuerung	1-57
Zykluszeit	2-21

Ihre Meinung interessiert uns!

Geben Sie uns die Möglichkeit, Ihre Anregungen, Wünsche und Kritikpunkte zum vorliegenden Handbuch zu erfahren.

Jeder noch so kleine Hinweis oder Kommentar wird von uns bearbeitet und wird wenn möglich in die Dokumentationen aufgenommen.

Faxen Sie uns deshalb den Vordruck auf der folgenden Seite ausgefüllt zu oder schicken Sie Ihre Anmerkungen, Verbesserungsvorschläge etc. an die folgende Adresse:

Phoenix Contact GmbH & Co. KG
Marketing Services
Dokumentation INTERBUS
32823 Blomberg
DEUTSCHLAND

Telefon +49 - (0) 52 35 - 3-00

Telefax +49 - (0) 52 35 - 3-4 20 66

E-Mail tecdoc@phoenixcontact.com

Fax-Antwort

Phoenix Contact GmbH & Co. KG
Marketing Services
Dokumentation INTERBUS

Datum: _____

Fax.-Nr.: +49 - (0) 52 35 - 3-4 20 66

Absender:

Firma: _____ Name: _____

Abteilung: _____

Straße: _____ Funktion: _____

Ort: _____ Tel.: _____

Fax: _____

Angaben zum Handbuch:

Bezeichnung: _____ Revision: _____ Art.-Nr.: _____

Meine Meinung zum Handbuch

Gestaltung

	Ja	zum Teil	Nein
Ist das Inhaltsverzeichnis übersichtlich gestaltet?	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Sind die Bilder/Grafiken verständlich/aussagekräftig?	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Sind die Texterklärungen zu den Bildern ausreichend?	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Entspricht die Qualität der Bilder Ihren Erwartungen?	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Fördert die Seitengestaltung die schnelle Infosuche?	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

Inhalt

	Ja	zum Teil	Nein
Sind die Formulierungen/Fachbegriffe verständlich?	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Sind die Verzeichniseinträge verständlich/aussagekräftig?	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Sind die Beispiele praxisgerecht?	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Ist das Handbuch gut zu handhaben?	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Fehlen wichtige Informationen? Wenn ja, welche?	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

Kommentar:
