

Profil : Sensor/Aktor

Profil-Nummer : 12

Datum : 6. November 1997

Herausgeber : INTERBUS Club Deutschland e.V.  
Geschäftsstelle  
Postfach 1108, D-32817 Blomberg  
Telefon : (0 52 35) 34 21 00  
Fax : (0 52 35) 34 12 34

Artikel-Nr. : 9318081

Copyright by INTERBUS Club Deutschland e.V.

Alle Abbildungen und Beschreibungen wurden nach bestem Wissen erstellt und geprüft, befreien den Anwender jedoch nicht von eigenen Prüfungen und Versuchen. Änderungen jeder Art, insbesondere soweit Sie sich aus technischen Fortschritt ergeben bleiben vorbehalten. Für Fehlhandlungen und Schäden, die durch Mißachtung der in diesem Profil enthaltenen Informationen entstehen, übernimmt der INTERBUS Club Deutschland e.V. keine Haftung. Dieses Profil, einschließlich aller darin enthaltenen Abbildungen, ist urheberrechtlich geschützt. Jede Drittverwendung dieses Profils, die von den urheberrechtlichen Bestimmungen abweicht, ist verboten.

Änderungen vorbehalten

## Inhaltsverzeichnis

	<b>Vorwort</b> .....	3
	<b>Einleitung</b> .....	4
<b>1.</b>	<b>Anwendungsbereich</b> .....	6
<b>2.</b>	<b>Referenzen</b> .....	6
<b>3.</b>	<b>Begriffe</b> .....	6
<b>4.</b>	<b>Symbole und Abkürzungen</b> .....	8
<b>5.</b>	<b>Gerätecharakterisierung</b> .....	9
<b>6.</b>	<b>Anwendung und Geräteeigenschaften</b> .....	9
6.1.	Allgemeines .....	9
6.2.	Diagnosefunktionen .....	11
6.3.	Kommunikationsfunktionen .....	11
6.3.1.	Prozeßdatenkanal .....	11
6.3.2.	Parameter-Kanal .....	13
6.3.3.	Verbindungsaufbau .....	14
6.3.4.	Verbindungsabbau .....	17
6.3.5.	Verbindungabbauen .....	18
6.3.6.	Identifikation .....	20
6.3.7.	Kommunikationsobjekt-Liste-Lesen .....	21
6.3.8.	Status-Funktion .....	23
6.3.9.	Read-Funktion .....	24
6.3.10.	Write-Funktion .....	27
6.3.10.1.	Einzelparametrierung .....	29
6.3.10.2.	Blockparametrierung .....	29
6.3.11.	Prozeßdatensteuerung .....	33
6.3.12.	Prozeßdaten-Überwachung .....	43
6.3.13.	Kommunikations-Überwachung .....	46
6.4.	Sensor/Aktor-Funktion .....	48
6.4.1.	Funktionsgruppen-Beschreibung .....	48
6.4.2.	Geräteinformation .....	51
6.4.3.	Störungsfunktion .....	57
<b>7.</b>	<b>Datenstrukturen</b> .....	60
7.1.	Datentypen .....	62
7.1.1.	Boolean .....	62
7.1.2.	Integer .....	62
7.1.3.	Unsigned .....	63
7.1.4.	Floating-Point .....	64
7.1.5.	Visible-String .....	64
7.1.6.	Octet-String .....	65
7.1.7.	Date .....	65
7.1.8.	Time-Of-Day .....	66
7.1.9.	Time-Difference .....	67
7.1.10.	Bit-String .....	67
7.2.	Anwendungsdaten .....	69
7.2.1.	Parameterbeschreibungsdaten .....	69
7.2.1.1.	Parameterbeschreibungsdaten der Geräteparameter .....	73
7.2.2.	Grenzwerte der Parameterbeschreibungsdaten .....	74

7.2.3.	Defaulteinstellung der Parameterbeschreibungsdaten .....	74
7.2.4.	Auswahlcode .....	74
<b>8.</b>	<b>Betriebsphasen der Anwendung</b> .....	<b>74</b>
8.1.	Anlauf/Abbruch.....	75
8.2.	Inbetriebnahmephase und Projektierungsphase.....	76
<b>9.</b>	<b>Kommunikationsprofil</b> .....	<b>76</b>
9.1.	Schicht 1.....	76
9.1.1.	Installationsfernbus-Schnittstelle.....	76
9.1.2.	Remotebus-Schnittstelle .....	76
9.1.3.	Lokalbus-Schnittstelle.....	77
9.2.	Schicht 2.....	77
9.2.1.	Konfiguration der INTERBUS-Register .....	77
9.2.2.	Identifikation der INTERBUS Teilnehmer.....	78
9.3.	Schicht 7.....	81

## Vorwort

Im Rahmen der Fabrikautomatisierung werden in der industriellen Sensorik und Aktorik immer mehr leistungsfähigere und flexiblere Systeme benötigt. Digitalisierte Sensoren und Aktoren können diese Anforderungen erfüllen. Ihre volle Integration in komplexe Fertigungsabläufe setzt aber offene und standardisierte Kommunikationsfähigkeit voraus.

Der Grundgedanke von offenen Systemen ist, den Informationsaustausch zwischen Anwendungsfunktionen zu ermöglichen, die auf Geräten unterschiedlicher Hersteller implementiert sind.

Hierzu gehören festgelegte Anwendungsfunktionen, eine einheitliche Anwenderschnittstelle zur Kommunikation und ein einheitliches Übertragungsmedium.

Der INTERBUS Club Deutschland e.V. hat sich zur Aufgabe gemacht, die wichtigsten Sensor/Aktor Gerätefunktionen zu standardisieren und in diesem Profil für Sensoren u. Aktoren zusammenzufassen.

Um die Sensor/Aktor Gerätefunktionen unabhängig vom Kommunikationsmedium definieren zu können, wurde eine international anerkannte und standardisierte Anwenderschnittstelle DIN 19 245 Teil 2 zur Kommunikation verwendet. Damit wurde eine Durchgängigkeit zu MMS geschaffen.

Als Übertragungsmedium wurde das INTERBUS-System ausgewählt, das die Anforderungen der Sensorik und Aktorik bezüglich Echtzeitverhalten und standardisierter Anwenderschnittstelle erfüllt.

Das Profil für Sensoren u. Aktoren richtet sich an den Nutzer und Gerätehersteller von Sensoren u. Aktoren die am Sensor-Aktorbus betrieben werden sollen.

Diese Profil-Definition ist für den Anwender eine sinnvolle Ergänzung zur standardisierten Kommunikation und bringt eine allgemeingültige Absprache über Dateninhalte und Geräteverhalten. Diese Funktionsfestlegungen vereinheitlichen wesentliche Sensor- u. Aktor Geräteparameter. Hierdurch zeigen Geräte verschiedener Hersteller, bei Verwendung dieser Standardparameter, ein gleiches Verhalten am Kommunikationsmedium.

Zur Konformitätsprüfung und Zertifizierung von Produkten mit dem Sensor/Aktor-Profil wird eine unabhängige Sachverständigenstelle eingerichtet.

Aufgrund von weiterführenden Standardisierungsarbeiten ist mit weiteren Ergänzungen zu rechnen.

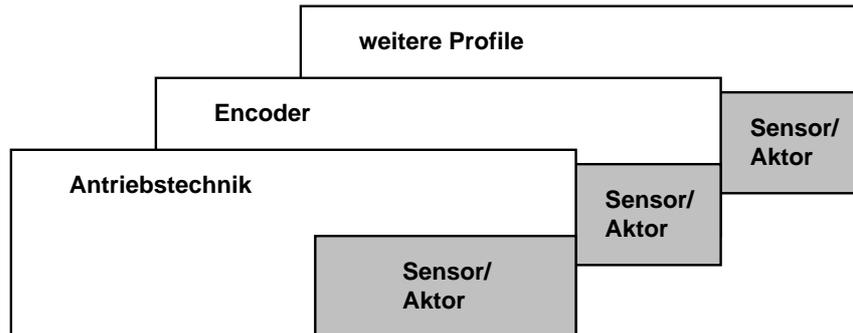
Verfasser:

Herr Krumsiek

Phoenix Contact, Blomberg

## Einleitung

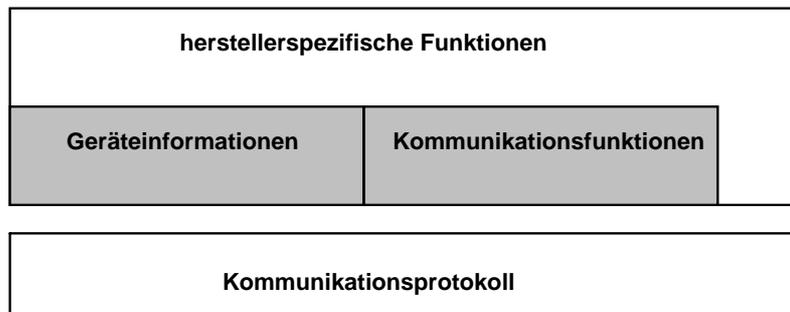
Dieses Profil beinhaltet die Grundfunktionen, die jedes Sensor- und Aktor-Gerät einem Nutzer zur Verfügung stellen muß. Im wesentlichen sind das Informationen über die Nutzung des Prozeßdatenkanals, Kommunikationsfunktionen und Geräteinformationen. Das Sensor/Aktor-Profil ist die Basis für alle Geräte mit Server Funktionalität. Die Profile Antriebstechnik und Encoder basieren bereits auf dem Sensor/Aktor-Profil. Die Indizes der Geräte-Parameter eines Sensor/Aktor-Gerätes liegen im Bereich 6000 - 603F hex, d.h. dieser Bereich ist für Indizes der Geräte-Parameter anderer Profile, die auf dieses Sensor/Aktor-Profil aufsetzen, gesperrt. Die Zusammenhänge werden in Bild 1 gezeigt.



**Bild 1: Sensor/Aktor-Profil als Basis-Profil**

### Sensor/Aktor-Funktionen

Dieses Profil definiert Anwendungsfunktionen von Sensoren und Aktoren. Die Anwendungsfunktionen teilen sich, wie in Bild 2 angegeben, in Gerätefunktionen, Kommunikationsfunktionen und Geräteinformationen. Außerdem werden Freiräume für die herstellereigenspezifischen Funktionen definiert.



**Bild 2: Anwendungsfunktionen von Sensoren und Aktoren**

Jede Anwendungsfunktion ist mit Hilfe eines Funktionsblocks beschrieben.

### Kommunikationsfunktionen

Die Kommunikationsfunktionen dienen zur Abwicklung der Dienstauführung und legen die Fehlermeldungen fest.

### Geräteinformationen

Die Geräteinformation ist eine Funktion, die die Ablage von allgemeingültigen Geräteinformationen in einen nichtflüchtigen Speicher festlegt.

**Herstellerspezifische Funktionen**

Die frei definierbaren, herstellerspezifischen Funktionen können die standardisierten Funktionen aller anderen Funktionsbereiche benutzen.

Diese Anwendungsfunktionen sind vom Geräteanwender durch Geräteparameter für seine Aufgabe parametrierbar.

Diese allgemeingültigen Parameter sind damit für alle Geräte, die dem Sensor/Aktor Profil, und allen darauf aufsetzenden Profilen entsprechen, festgelegt.

Die Parameternummer (Index), mit der diese Geräte-Parameter adressiert werden, liegen im Bereich 6000 - 603f hex, d.h. dieser Bereich ist für Geräte-Parameter aller Profile, die auf dieses Sensor/Aktor-Profil aufsetzen, reserviert.

## 1. Anwendungsbereich

Diese Profil legt Grundfunktionen fest, die in jedem Profil für INTERBUS Teilnehmer genutzt werden.

## 2. Referenzen

Das Anwendungsprotokoll und die Datenstrukturen entsprechen der INTERBUS Club Richtlinie.

Die Anwendungsschnittstelle zur Kommunikation über den INTERBUS Parameter-Kanal entsprechen der INTERBUS Club Richtlinie.

Die Festlegungen zur Datenübertragung über den Prozeßdatenkanal beruhen auf der INTERBUS Club Richtlinie und dem Entwurf DIN 19258.

## 3. Begriffe

### Geräteprofil

Das Geräteprofil legt - die über die Kommunikation - sichtbaren Anwendungsfunktionen fest. Die Anwendungsfunktionen werden durch folgende Festlegungen auf die Kommunikation abgebildet:

- Durch das Kommunikationsprofil,
- durch die Interaktionen zwischen den Anwendungsfunktionen, soweit sie über das Kommunikationssystem ausgeführt werden, sowie
- durch die genutzten Kommunikationsdienste und der damit manipulierbaren Kommunikationsobjekte.

Das Ergebnis der Abbildung ist das sichtbare Verhalten der Anwendung. Die Festlegungen eines Anwendungsprofils ermöglichen die Interoperabilität in einem Anwendungsfeld. Die Prämisse hierfür ist, daß die genutzten Geräteeigenschaften dieses zulassen.

Weiterhin werden Eigenschaften der Geräte festgelegt, die für den Nutzer von Bedeutung sind.

Es wird unterschieden zwischen Pflichtfunktionen (mandatory), optionalen und herstellerspezifischen Gerätefunktionen sowie Parametern.

Beschränkt sich der Anwender auf die Pflichtfunktionen oder Parameter, ist eine Austauschbarkeit der Geräte möglich - wenn die genutzten Geräteeigenschaften und -einstellungen dieses zulassen. In Bezug auf die Kommunikation sind die Geräte - unabhängig von der Funktion - bei gleichen Parametern immer austauschbar.

### Kommunikationsprofil

Im Kommunikationsprofil werden, die in der Spezifikation des Übertragungsmediums enthaltenen Freiheitsgrade, anwendungs- oder gerätegruppenspezifisch eingeschränkt bzw. klassifiziert. Im Kommunikationsprofil werden Kommunikations-Dienste und -Parameter festgelegt, die in der Spezifikation als optional gekennzeichnet sind. Alle im Kommunikationsprofil nicht genannten optionalen Funktionen und Parameter bleiben optional. Mandatory-Dienste und -Parameter sind auch ohne Nennung im Profil bindend. In dem Profil werden weiterhin Wertebereiche von Attributen und Parametern eingegrenzt bzw. festgelegt.

### Kommunikationsinterface

Das Kommunikationsinterface setzt sich aus einem Prozeßdatenkanal und einem Parameter-Kanal zusammen.

### Prozeßdatenkanal

Der Prozeßdatenkanal dient zur schnellen Übertragung von Prozeßdaten. Über den Prozeßdatenkanal werden Daten unquittiert und äquidistant übertragen. Prozeßdaten können gelesen und geschrieben werden.

Die Richtungsangabe der Prozeßdaten wird vom Bus gesehen; d.h.,

- Prozeßausgangsdaten sind Daten, die vom Steuerungssystem zum Gerät übertragen werden. Dieses läßt diese Daten aus dem Prozeßdatenkanal und gibt sie je nach Funktion an den Prozeß aus.
- Prozeßeingangsdaten sind Daten, die vom Gerät zum Steuerungssystem übertragen werden. Das Gerät schreibt diese Daten in den Prozeßdatenkanal und überträgt sie damit zum Steuerungssystem.

### **Prozeßdatenbeschreibung**

Falls das Gerät parallel mehrere Prozeßdaten überträgt oder empfängt, muß die Struktur und Bedeutung dieser Prozeßdaten durch den Hersteller beschrieben werden.

### **Parameter-Kanal**

Über den Parameter-Kanal kann auf alle Kommunikationsobjekte zugegriffen werden. Die Parameterkanal-Dienste erlauben einen quitierten Zugriff auf die Geräteparameter, d.h. der Zugriff auf einen Geräteparameter wird vom Gerät bestätigt.

### **VFD-Objekt**

Das Virtual Field Device (VFD) ist ein abstraktes Modell zur Beschreibung der Daten und des Verhaltens eines Automatisierungsgerätes aus der Sicht seines Kommunikationspartners. Die Basis des VFD-Modells ist das VFD-Objekt. Das VFD-Objekt beinhaltet alle Objekte und Objektbeschreibungen, die von einem Kommunikationspartner durch einen Services genutzt werden können. Die Objektbeschreibungen sind in einem Objektverzeichnis hinterlegt. Je VFD gibt es genau ein Objektverzeichnis.

### **Kommunikationsreferenz**

Jede Kommunikationsbeziehung zwischen zwei Geräten wird projiziert, unabhängig vom Zeitpunkt ihrer Nutzung. Die Projektierung wird in jedem Busteilnehmer in einer Kommunikationsbeziehungsliste (KBL) gespeichert. Ein Anwendungsprozeß identifiziert die Kommunikationsbeziehung über eine lokale Kommunikationsreferenz. Diese dient also zum Adressieren des Kommunikationspartners.

### **Fehlermeldung**

Die Fehlermeldung wird zurückgegeben, wenn ein Service nicht ausgeführt werden konnte.

### **Index, Subindex**

Der Index dient dazu, einen Parameter (Kommunikationsobjekt) zu adressieren. Der Subindex adressiert innerhalb eines Parameters, der als Struktur angelegt ist, einen Subparameter (Element eines Kommunikationsobjektes).

### **Geräteparameter**

Für alle Geräteparameter sind Defaultwerte in diesem Profil festgelegt.

### **Ersatzwerte**

Wenn die optionalen Kommunikationsobjekte nicht implementiert sind, verhält sich das Gerät, adäquat zu dem, für diesen Parameter definierten Ersatzwert.

### **Pflichtbereich**

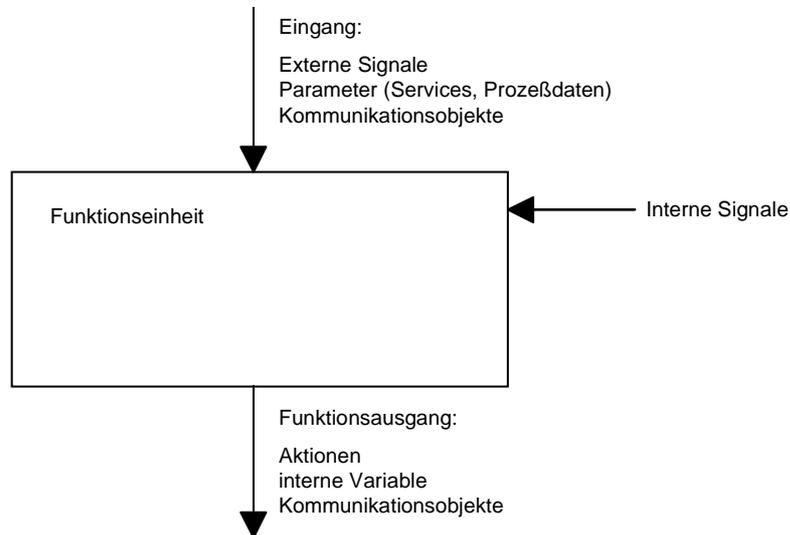
Der Pflichtbereich ist der Wertebereich, auf den ein Parameter, falls er implementiert ist, in jedem Fall parametrisiert werden kann.

### **Zustandsmaschine**

In diesem Profil sind einige Funktionen mit Hilfe einer Zustandsmaschine beschrieben. Ein Zustand repräsentiert ein bestimmtes internes und externes Verhalten. Er kann nur durch definierte Ereignisse verlassen werden. Den Ereignissen sind entsprechende Zustandsübergänge zugeordnet. In einem Übergang können Aktionen ausgeführt werden. In diesem wird das Zustandsverhalten geändert. Mit Beendigung des Übergangs wird der aktuelle Zustand auf den Folgezustand geändert.

### Definition der Funktionseinheiten

Die Gerätefunktion wird durch eine Funktionseinheit beschrieben. Die Funktion wird durch die Eingänge gesteuert und parametrisiert. Außerdem können interne Signale auf die Funktion einwirken oder die Funktion wird über interne Parameter beeinflusst. Der Ausgang der Funktion kann auf die Eingänge von anderen Funktionen geschaltet oder über den Bus zugänglich gemacht werden.



**Bild 3: Funktionseinheit**

## 4. Symbole und Abkürzungen

### Netzwerkspezifische Abkürzungen

ALI	Application Layer Interface
IB-S	INTERBUS
KBL	Kommunikationsbeziehungsliste
KBL-Header	Kopf der Kommunikationsbeziehungsliste
KR	Kommunikationsreferenz
m	mandatory
MAP	Manufacturing Automation Protocol
o	optional
OV	Objektverzeichnis
PCP	Peripherals Communication Protocol
PD	Prozeßdaten
PMS	Peripherals Message Specification
S-OV	Statisches-Objektverzeichnis
ST-OV	Statisches Typverzeichnis
VFD	Virtual Field Device
.con	Confirmation-Primitive
.ind	Indication-Primitive
.req	Request-Primitive
.res	Response-Primitive

## 5. Gerätecharakterisierung

Die Verbindung vom Automatisierungsgerät (SPS, Prozeßrechner) zum Prozeß wird durch einen Sensor oder Aktor gebildet. Vereinfacht ausgedrückt - setzt der Sensor physikalische Größen aus dem Prozeß in elektrische Signale für das Automatisierungsgerät um. Der Aktor setzt die elektrischen Signale des Automatisierungsgerätes in eine physikalische Größe für den Prozeß um.

Sensoren und Aktoren können aktive oder passive Teilnehmer am Bus sein.

Bezüglich Funktion und Preis verlangt der Markt der allgemeinen Sensorik und Aktorik ein großes Spektrum an unterschiedlichen Geräten. Durch die offene Struktur des Sensor/Aktor-Profiles wird die Abdeckung der unterschiedlichen Funktionen erreicht, und es ist deshalb keine Einteilung der Geräte in Geräteklassen erforderlich.

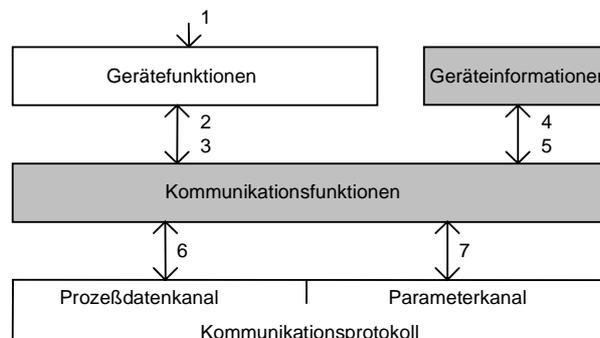
Die Gerätefunktionen und Parameter werden unterschieden in Pflicht-, optionale und herstellerspezifische. Beschränkt sich der Anwender auf die Pflicht-Funktionen oder Pflicht-Parameter, ist eine Austauschbarkeit der Sensoren und Aktoren möglich.

Im Bezug auf die Kommunikation sind die Geräte - unabhängig von der Funktion - bei gleichen Parametern immer austauschbar.

## 6. Anwendung und Geräteeigenschaften

### 6.1. Allgemeines

In diesem Kapitel wird die gesamte Anwendung aus der Kommunikationssicht beschrieben. Die Anwendung ist in folgende Funktionsblöcke aufgeteilt:



**Bild 4: Funktionsblöcke einer Anwendung**

#### Kommunikationsfunktion

Die Kommunikationsfunktion führt alle kommunikationsspezifischen Funktionen aus.

#### Gerätefunktion

Die Gerätefunktion führt alle gerätespezifischen Funktionen aus.

#### Geräteinformation

Die Geräteinformation verwaltet Informationen über das Gerät in einen nichtflüchtigen Speicher.

### Kommunikationslayer

Der Kommunikationslayer beinhaltet eine Schicht 7 entsprechend der DIN 19245 Teil 2 und einer Schicht 2 entsprechend der INTERBUS Spezifikation (siehe Gliederungspunkte 5.2. - 5.7.).

### Interaktionen zwischen den Funktionsblöcken

- 1 Prozeßgrößen
- 2 Prozeßdaten vom überlagertem Steuerungssystem zu den Gerätefunktionen
- 3 Prozeßdaten von den Gerätefunktionen zum überlagertem Steuerungssystem
- 4 Speicherung von Geräteinformationen
- 5 Auslesen der Geräteinformationen
- 6 Abbildung auf den Prozeßdatenkanal
- 7 Abbildung auf den Parameter-Kanal

### Funktionsgruppen

Folgende Funktionsgruppen sind für dieses Profil definiert.

Funktionsgruppe		Bedeutung
Profil-Gruppe	Funktionskennungen	
1	1	Kommunikationsfunktionen
1	2	Geräteinformationen

## 6.2. Diagnosefunktionen

Die Diagnoseanzeige signalisiert den Zustand des Netzwerks und wird durch LED's an den Busteilnehmern realisiert. Diese LED's müssen sichtbar an dem Gerät angebracht sein.

Kennzeichnung am Gerät	Name	Farbe	Funktion
RC	Remote bus Check	grün	Überwachung der Funktion des ankommenden Busses
RD	Remote bus Disable	rot	Abschaltung des weiterführenden Fibussegments
BA	Bus Active	grün	Anzeige der Busaktivität (ID-Zyklen und Datenzyklen)
TR	Transmit / Receive	grün	Anzeige einer Datenübertragung über den Parameterkanal

Anmerkung die LED „TR“ ist nur erforderlich, wenn der Parameter-Kanal implementiert ist

## 6.3. Kommunikationsfunktionen

### 6.3.1. Prozeßdatenkanal

Über einen Prozeßdatenkanal können parallel mehrere Prozeßdaten übertragen werden (siehe Beispiel). In einem solchen Fall muß dem Anwender die Struktur dieses Prozeßdatenkanals bekannt sein. Die Information über die Struktur muß der Gerätehersteller in Form einer Prozeßdatenkanal-Beschreibung liefern.

Beispiel: Prozeßdatenkanal eines Antriebs

	Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5
IN	Control word		Setpoint		Setpoint factor	
OUT	Status word		Actual value		....	

Eine Prozeßdatenkanal-Beschreibung wird als Prozeßdaten-Objektverzeichnis realisiert. Dieses Prozeßdaten-Objektverzeichnis enthält für jedes Prozeßdatum eine Prozeßdaten-Objektbeschreibung. Ein Prozeßdaten-Objektverzeichnis beinhaltet folgende Attribute.

### Index

Der Index ist die logische Adresse des Prozeßdatums innerhalb des Gerätes. Der Index 0000hex ist für den Header des Prozeßdaten-Objektverzeichnisses reserviert. Das Prozeßdatum mit dem Index 0001hex beschreibt den Prozeßeingangsdatenkanal. Das Prozeßdatum mit dem Index 0002hex beschreibt den Prozeßausgangsdatenkanal.

### Objekt Code

Dieses Attribut kennzeichnet das Prozeßdatum als Eingangsdaten oder Ausgangsdaten

### Datentyp

Dieses Attribut definiert, wie das Prozeßdatum zu interpretieren ist.

### Länge

Dieses Attribut beschreibt die Länge des Prozeßdatums in Byte.

### Interne Adresse

Dieses Attribut gibt die Position des Prozeßdatums innerhalb des Prozeßdatenkanals des Gerätes an.

### Extension

enthält zusätzliche Informationen (z.B. Name des Prozeßdatums).

**Beispiel für einen Antrieb:**

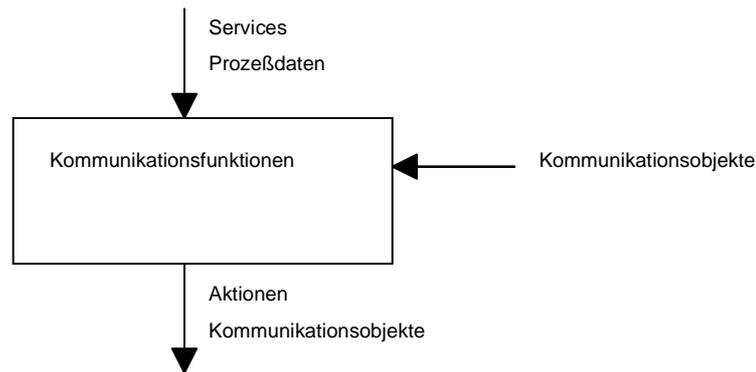
Prozeßdatenkanal-Beschreibung

<b>Prozeßdatenname</b>	<b>Index</b>	<b>Objekt Code</b>	<b>Datentyp</b>	<b>Länge</b>	<b>Interne Adr.</b>	<b>Extension</b>
Eingangsdaten	0001	PD-IN	Octet String	6	0.0	
Ausgangsdaten	0002	PD-OUT	Octet String	6	0.0	
Statuswort	6041	PD-IN	Octet String	2	0.0	
Steuerwort	6040	PD-OUT	Octet String	2	0.0	
Istwert	6044	PD-IN	Integer16	2	2.0	
Sollwert	6042	PD-OUT	Integer16	2	2.0	
Störungscode	603F	PD-IN	Octet String	2	4.0	
Sollwert-Faktor	604B	PD-OUT	Integer16	2	4.0	

### 6.3.2. Parameter-Kanal

Die Kommunikationsfunktion (siehe Bild 5) setzt sich aus Subfunktionen zusammen und beinhaltet folgende kommunikationsspezifischen Funktionen:

- Ausführung der Services;
- Steuerung der Kommunikation;
- Abbildung der Prozeßdaten auf die Kommunikationsobjekte;
- Prozeßdatenüberwachung;
- Kommunikationsüberwachung.



**Bild 5: Kommunikationsfunktionen**

#### **Services**

Folgende Services werden durch die Kommunikationsfunktion ausgeführt:

- Initiate (Verbindung aufbauen);
- Abort (Verbindung abbauen);
- Status (Gerätestatus lesen);
- Identify (Lesen von Herstellername, Typ und Version);
- Get-OV (Lesen der Objektbeschreibungen);
- Read (Lesen von Kommunikationsobjekten);
- Write (Schreiben von Kommunikationsobjekten).

#### **Prozeßdaten**

Prozeßdaten sind die Daten, die über den Prozeßdatenkanal übertragen und auf Kommunikationsobjekte abgebildet werden. Die Abbildung der Prozeßdaten auf die Kommunikationsobjekte erfolgt mit der Funktion Prozeßdatensteuerung.

#### **Aktionen**

Die Aktionen, die von der kommunikationsspezifischen Funktion ausgelöst werden können, sind parametrierbar. Sie legen im wesentlichen das Verhalten des Gerätes beim Kommunikationsausfall fest.

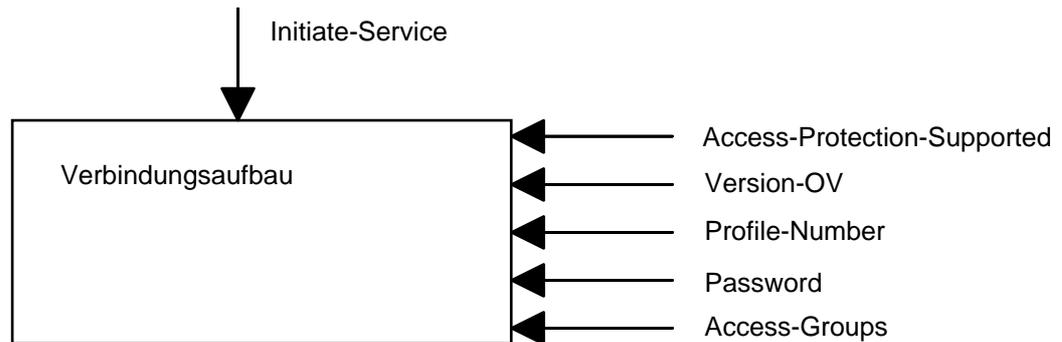
#### **Kommunikationsobjekte**

Kommunikationsobjekte sind über den Bus mit Hilfe von Services manipulierbar. Die Kommunikationsobjekte werden auf die realen Parameter im Gerät abgebildet. Die Daten der Kommunikationsobjekte können über Services oder Prozeßdaten übertragen werden. Alle Kommunikationsobjekte, die über Read und Write Services erreichbar sind, werden im Objektverzeichnis beschrieben.

Der Zugriff auf nicht implementierte optionale Kommunikationsobjekte wird immer negativ quittiert. (Fehlermeldung: Error-Class = Access (6) Error-Code = Objekt-Access-Unsupported (6))

### 6.3.3. Verbindungsaufbau

Die in Bild 6 dargestellte Funktion Verbindungsaufbau wird durchgeführt, wenn eine Aufforderung zum Verbindungsaufbau (Initiate-Service Indication) vom Bus empfangen wird. Die Verbindung wird nur aufgebaut, wenn das Objektverzeichnis und VFD-Objekt vorhanden sind.



**Bild 6: Verbindungsaufbau**

#### **Initiate-Service**

Der Service Initiate ist die Anforderung für einen Verbindungsaufbau. Die Service-Parameter müssen vom Initiator des Verbindungsaufbaus mit folgenden Werten gesetzt werden:

##### *Version-OV*

Dieser Service-Parameter hat in diesem Profil keine Bedeutung.

##### *Profile-Number*

Dieser Service-Parameter hat in diesem Profil keine Bedeutung.

##### *Access-Protection-Supported*

Dieser Service-Parameter hat in diesem Profil keine Bedeutung.

##### *Password*

Dieser Parameter enthält das Paßwort, das auf dieser Kommunikationsbeziehung für alle Zugriffe auf die Kommunikationsobjekte des Gerätes gilt. Für den Zugriff auf die Kommunikationsobjekte, die in diesem Profil definiert sind, ist kein Zugriffsschutz für das Paßwort vorgesehen. Damit wird dieser Wert in dem Gerät nicht berücksichtigt. Falls herstellerspezifische Kommunikationsobjekte mit einem Zugriffsschutz definiert sind, muß die Kommunikationsverbindung mit dem entsprechenden Paßwort (siehe Herstellerangabe) aufgebaut werden.

##### *Access-Groups*

Dieser Parameter enthält eine Zuordnung des Gerätenutzers zu bestimmten Zugriffsgruppen. Diese Zuordnung gilt auf dieser Kommunikationsbeziehung für alle Zugriffe auf die Kommunikationsobjekte des Gerätes. Für den Zugriff auf die Kommunikationsobjekte, die in diesem Profil definiert sind, ist kein Zugriffsschutz für die Zugriffsgruppen vorgesehen. Damit wird dieser Wert in dem Gerät nicht berücksichtigt. Falls herstellerspezifische Kommunikationsobjekte mit einem Zugriffsschutz über Zugriffsgruppen definiert sind, muß die Kommunikationsverbindung mit dem entsprechenden Wert für den Service-Parameter Access-Groups (siehe Herstellerangabe) aufgebaut werden.

Folgende Service-Parameter werden vom Gerät bei der Beantwortung des Initiate-Services zurückgegeben:

**Access-Protection-Supported**

Dieser Service-Parameter gibt an, ob der Zugriffsschutz durchgeführt wird. Der Service-Parameter ist auf den Wert TRUE zu setzen.

**Version-OV**

Der Wert für die Version des Objektverzeichnisses ist herstellerspezifisch zu vergeben. Der Wert sollte bei jeder Änderung des Objektverzeichnisses geändert werden.

**Profile-Number**

Der Wert des Service-Parameters Profile-Number ist für dieses Profil 0012 hex. Falls weiterführende Profile, die auf diesem Profil basieren, unterstützt werden, wird die entsprechende Profile-Number zurückgegeben. Die Profilnummer setzt sich aus den Komponenten Profil-Gruppe und Profilversion zusammen. Die Profilnummer kennzeichnet, welchem Profil das Gerät entspricht. Ein Gerät darf die entsprechende Profilnummer nur melden, wenn alle in dem Profil vorgeschriebenen Funktionen zur Verfügung stehen.

b15	b12	b11	b4	b3	b0
Profile group			Profile version		

*Profil-Gruppe*

Dieser Parameter gibt die Profilgruppe an, in der diese Funktionsgruppe beschrieben ist. Für ein DRIVECOM-Gerät muß z.B. mindestens eine Funktion aus der Profilgruppe 2 und 1 vorhanden sein.

Wertebereich : 0-FF hex

Profilgruppe	Bedeutung	
0	kein Profil	
1	Sensor/Aktor	
2	DRIVECOM	
3	reserve	
4	Anschaltbaugruppen	
5	reserve	
6	reserve	
7	Encoder	
8	Prozeßregler	
9	Robotersteuerungen	
A	Schraubersteuerungen	
B	ISO-Ventile	
C	Schweißsteuerungen	
D	Bedien/Anzeigeräte	

*Profil-Version*

Dieser Parameter enthält die Profil-Versionskennung der entsprechenden Profil-Gruppe.

Wertebereich : 0-FF hex

**Password**

Der Wert des Service-Parameters Password ist für dieses Profil 0 hex.

**Access-Groups**

Der Wert des Service-Parameters Access-Groups ist für dieses Profil 0 hex.

**Fehlermeldung**

Der Service-Parameter Error-Type setzt sich aus folgenden Parametern zusammen:

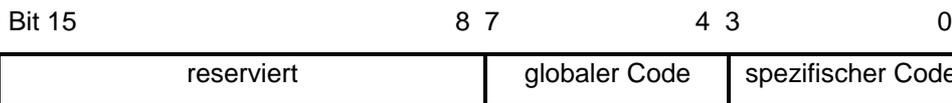
- Error-Class;
- Error-Code;
- Additional-Code.

Error-Class und Error-Code

Error-Class	Error-Code	Bedeutung
0 initiate	0	nicht zugelassen
0 initiate	1 max-pdu-size-unsufficient	die PDU-Länge ist nicht ausreichend
0 initiate	2 feature-not-supported	der gewünschte Service wird nicht unterstützt
0 initiate	3 version-ov-incompatible	die Versionen der Objektverzeichnisse sind unverträglich
0 initiate	4 User-Initiate-Denied	der PMS Anwender lehnt den Initiate ab
0 initiate	5 password-error	es ist bereits eine Verbindung mit dem Paßwort aufgebaut
0 initiate	6 profil-number-incompatible	das Profil des Client wird nicht unterstützt
0 initiate	7 other	der Fehler läßt sich keinem Code zuordnen

Additional-Code:

2 Octets mit folgendem Aufbau:



Die Bits 8 bis 15 sind mit dem Wert 0 zu besetzen.

globaler Code	spezifischer Code	Bedeutung
0	0	keine genauere Angabe des Fehlergrundes
1	0	noch kein Objektverzeichnis vorhanden
2	0	noch kein VFD-Objekt vorhanden

Nicht aufgeführte Codes sind reserviert.

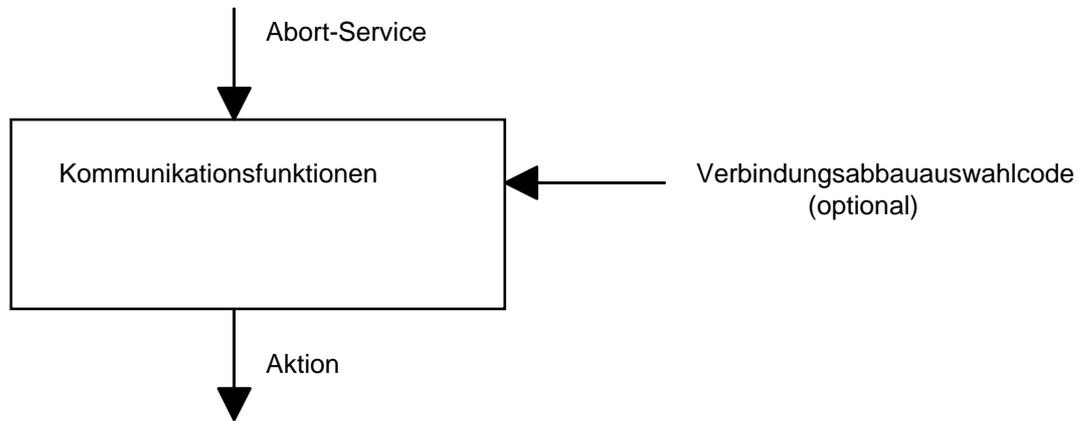
**Abbildung der Gerätefunktion auf die Kommunikation**

Die Initiate Response wird mit folgenden Service Parametern zurückgegeben.

Initiate.res-Parameter	Wert
Invoke-ID	Invoke-ID aus Initiate.ind
Kommunikationsreferenz	Kommunikationsreferenz aus Initiate.ind
Version-OV	herstellerspezifisch
Profile-Number	0012 hex (falls nur das Sensor/Aktor-Profil unterstützt wird)
Access-Protection-Supported	TRUE
Password	herstellerspezifisch
Access-Groups	herstellerspezifisch

### 6.3.4. Verbindungsabbau

Die in Bild 7 dargestellte Funktion Verbindungsabbau wird aufgerufen, wenn ein Abort-Service vom Bus empfangen wird.



**Bild 7: Verbindungsabbau**

#### 'Verbindungsabbau-Auswahlcode'

Der 'Verbindungsabbau-Auswahlcode' definiert, welche Funktion ausgeführt wird, wenn die Verbindung abgebaut wird. Falls das Objekt Auswahlcode nicht vorhanden ist, wird keine Aktion ausgeführt.

Auswahlcode	Bedeutung der Auswahl-Funktion
-32768 ... -1	herstellerspezifisch
0	keine Aktion
1	Störung
2 ... 32767	reserviert für Profile

Objektklasse	optional
Zugriff	nur lesbar
Prozeßdaten-Abbildung	nicht möglich
Einheit	-
Wertebereich	Integer16
Pflichtbereich	0 ( keine Aktion )
Ersatzwert	0 ( keine Aktion )

#### Aktion

Es wird die Funktion ausgelöst, die im 'Verbindungsabbau-Auswahlcode' definiert ist.

#### Fehlermeldung

Ja, siehe Read- oder Write-Funktion.

**Abbildung der Gerätefunktion auf die Kommunikation**

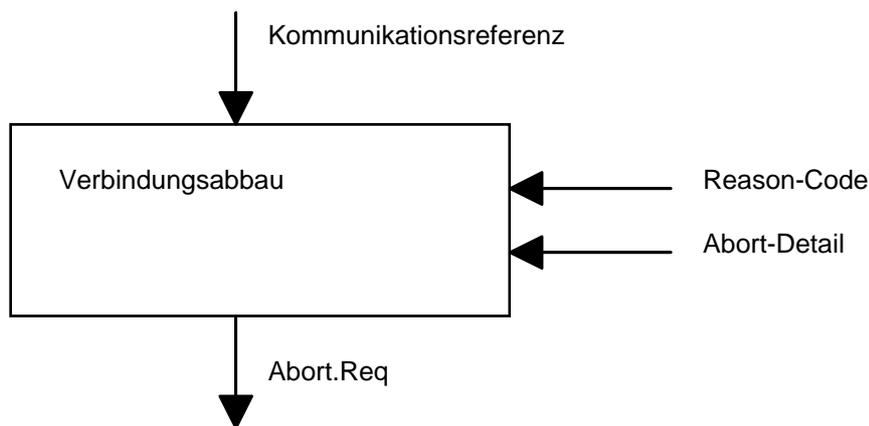
Objektbeschreibung: 'Verbindungsabbau-Auswahlcode'

**Tabelle 1: Objektbeschreibung 'Verbindungsabbau-Auswahlcode'**

Objektattribut	Wert hex	Bedeutung
Index	xxxx	Verbindungsabbau-Auswahlcode
Variable-Name	-	nicht vorhanden
Object-Code	07	Simple-Variable
Data-Type-Index	03	Integer16
Length	02	2 Byte
Password	00	kein Paßwort
Access-Groups	00	keine Zugriffsgruppen
Access-Rights	0300	Read-All, Write-All
Local-Address	xxxx	herstellerspezifisch
Extension	-	nicht vorhanden

6.3.5. Verbindungabbauen

Die in Bild 8 dargestellte Funktion Verbindungabbauen führt den Verbindungsabbau durch. Die Funktion muß z.B. benutzt werden, wenn das Objektverzeichnis oder das VFD-Objekt des Gerätes lokal verändert werden soll.



**Bild 8: Verbindungabbauen**

**Kommunikationsreferenz**

Der Service-Parameter enthält die Kommunikationsreferenz der Kommunikationsbeziehung, auf der diese Verbindung abgebaut werden soll.

**Reason-Code**

Der Reason-Code gibt den Grund für den Verbindungsabbau an. Folgende Gründe sind möglich:

- Disconnect:  
Der Client baut die Verbindung ab.
- Version-OV-incompatible:  
Falls das Gerät das Objektverzeichnis lokal ändern will. In diesem Fall sind die Versionen der Objektverzeichnisse (Source-OV und Remote-OV) der beiden Kommunikationspartner unverträglich.

- Password-Error:  
Es ist bereits eine Kommunikationsbeziehung mit dem gleichen Paßwort aufgebaut.
- Profil-Number-incompatible:  
Das Profil des Servers wird nicht unterstützt.
- Limited-Services-Permitted:  
Das Gerät hat den Logical-Status "Begrenzte Anzahl von Services" eingenommen.

Reason-Code	Bedeutung
1	Disconnect
2	Version-OV-incompatible
3	Password-Error
4	Profil-Number-incompatible
5	Limited-Services-Permitted

### Abort-Detail

Dieser Parameter enthält zusätzliche Informationen über den Abbruchgrund.

Abort-Detail	Bedeutung
-127 ... -1	herstellerspezifisch
0	kein Abort-Detail
1 ... 128	reserviert für zukünftige Profil Versionen

### Abort.Req

Mit dem Abort-Service Request wird der Verbindungsabbau an den Kommunikationspartner übermittelt.

### Fehlerbehandlung

Keine

### Abbildung der Gerätefunktion auf die Kommunikation

Folgende Service-Parameter werden beim Verbindungsabbruch angegeben:

Abort-Identifizier = 0 (User)

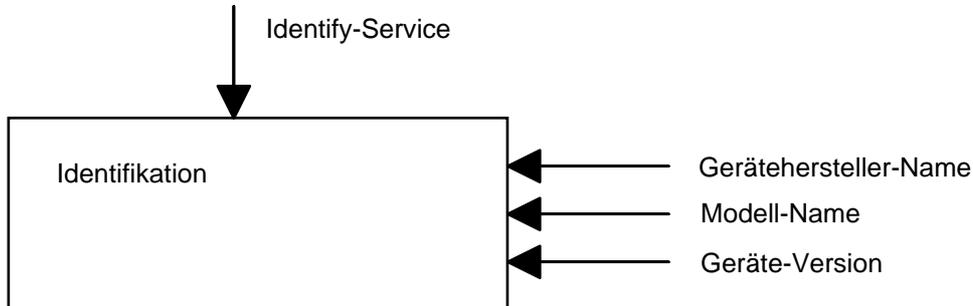
Reasoncode:

Reasoncode	Bedeutung
1	Disconnect
2	Version-OV-Incompatibe

Abort-Detail	Bedeutung
-127 ... -1	herstellerspezifisch
0	kein Abort-Detail
1 ... 128	reserviert für nächste Profil Versionen

### 6.3.6. Identifikation

Die Geräteidentifikation wird selbständig von dem Funktionsblock 'Identifikation' (siehe Bild 9) durchgeführt.



**Bild 9: Identifikation**

#### **Gerätehersteller-Name**

Dieser Parameter enthält den Namen des Geräteherstellers, dargestellt durch max. 16 ASCII-Zeichen.

#### **Modell-Name**

Dieser Parameter enthält den Modellnamen des Gerätes, dargestellt durch max. 16 ASCII-Zeichen.

#### **Geräte-Version**

Dieser Parameter enthält die Version des Gerätes, dargestellt durch max. 16 ASCII-Zeichen.

#### **Fehlermeldung**

keine

#### **Abbildung der Gerätefunktion auf die Kommunikation**

#### **Kommunikationsreferenz**

Die Service.res wird mit der Kommunikationsreferenz aus der Service.ind aufgerufen.

#### **Invoke-ID**

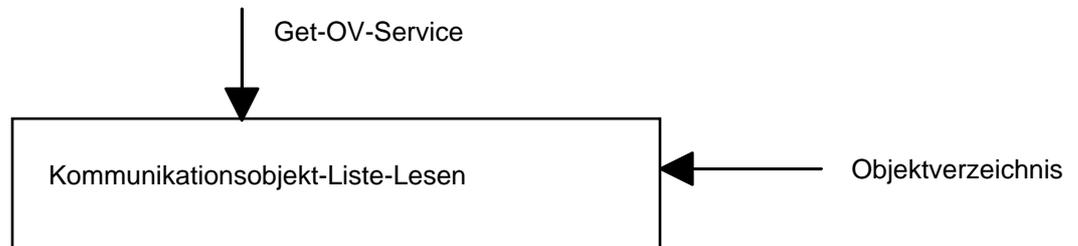
Die Service.res wird mit der Invoke-ID aus der Service.ind aufgerufen.

VFD-Objekt:

Attribute	Bedeutung	Länge
Vendor-Name	Gerätehersteller-Name	16 Byte
Model-Name	Geräte-Name	16 Byte
Revision	Geräte-Version	16 Byte

### 6.3.7. Kommunikationsobjekt-Liste-Lesen

Mit dieser in Bild 10 dargestellten Funktion können alle Beschreibungen der Geräteparameter ausgelesen werden. Das Gerät stellt die Daten für das Objektverzeichnis zur Verfügung.



**Bild 10: Kommunikationsobjekt-Liste-Lesen**

#### Get-OV-Dienst

Mit diesem Dienst wird das Objektverzeichnis ausgelesen.

#### Objektverzeichnis

Das Objektverzeichnis enthält die Beschreibung aller Kommunikationsobjekte.

#### Fehlermeldung

Der Service-Parameter Error-Type setzt sich aus folgenden Parametern zusammen:

- Error-Class;
- Error-Code;
- Additional-Code.

Der Parameter Additional-Code hat den Wert Null.

Error-Class und Error-Code:

Error-Class	Error-Code	Bedeutung
5 Service	4 Parameter-Inkonsistent	Zugriff über Namen mit Länge Null
5 Service	5 Illegal-Parameter	Der Service-Parameter 'Access-Specification' hat einen unzulässigen Wert.
6 Access	7 Objekt-Non-Existent	Unter diesem Index existiert kein Objekt.
6 Access	8 Type-Conflict	Die Daten des Parameters 'Name' sind nicht vom Datentyp Visible-String.
6 Access	9 Name-Access-Unsupported	Zugriff über Namen, obwohl im OV die Symbollänge mit Null spezifiziert ist

## **Abbildung der Gerätefunktion auf die Kommunikation**

### **Kommunikationsreferenz**

Die Service.res wird mit der Kommunikationsreferenz aus der Service.ind aufgerufen.

### **Invoke-ID**

Die Service.res wird mit der Invoke-ID aus der Service.ind aufgerufen.

### **Index**

Der Parameter Index definiert das Objekt, dessen Objektbeschreibung ausgelesen werden soll.

### **List of Objektbeschreibungen**

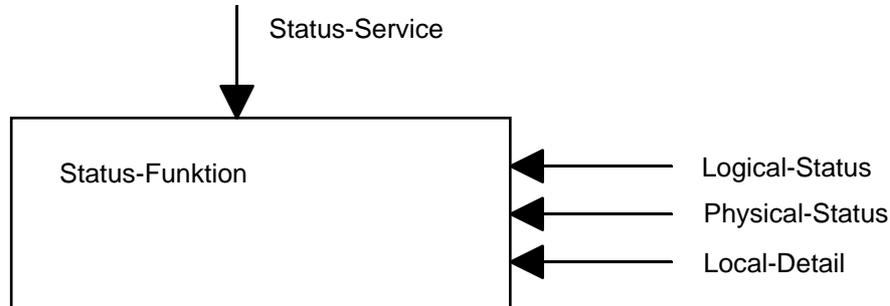
Dieser Serviceparameter enthält die Objektbeschreibung des mit dem Index definierten Objektes.

### **Error-Type**

Der Parameter Error-Type enthält den Fehlergrund.

### 6.3.8. Status-Funktion

Der Funktionsblock Status (siehe Bild 11) führt die Statusabfrage durch. Der Status gibt den Gerätezustand an.



**Bild 11: Status-Funktion**

#### Logical-Status

Dieser Parameter definiert die Kommunikationsfähigkeit des Gerätes.

	Logical-Status
0	kommunikationsbereit
2	Begrenzte Anzahl Services

#### Physical-Status

Dieser Parameter gibt einen groben Überblick über den Betriebszustand des Gerätes.

	Physical-Status
0	betriebsbereit
1	teilweise betriebsbereit
2	nicht betriebsbereit
3	Wartung erforderlich

#### Local-Detail

Dieser Parameter gibt den lokalen Status der Anwendung und des Gerätes an. Die Bedeutung der einzelnen Bits wird in Profilen festgelegt. Der Datentyp ist Bit-String mit der Länge 24 Bit.

#### Fehlermeldung

keine

#### Abbildung der Gerätefunktion auf die Kommunikation

#### Kommunikationsreferenz

Die Service.res wird mit der Kommunikationsreferenz aus der Service.ind aufgerufen.

#### Invoke-ID

Die Service.res wird mit der Invoke-ID aus der Service.ind aufgerufen.

VFD-Objekt:

Attribute	Bedeutung
Logical-Status	Zustand der Kommunikation
Physical-Status	Betriebszustand des Gerätes
Local-Detail	lokaler Status der Anwendung profilspezifisch

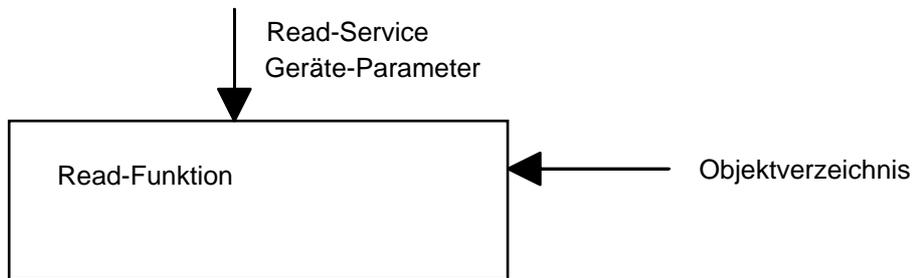
Logical-Status

Dieses Attribut enthält eine Angabe über den Zustand der Kommunikation des Gerätes:

- 0 <=> Kommunikationsbereit
- 2 <=> Begrenzte Anzahl von Services
- 4 <=> OV-LOADING-NON-INTERACTING
- 5 <=> OV-LOADING-INTERACTING

### 6.3.9. Read-Funktion

Mit dieser in Bild 12 dargestellten Funktion wird der Lesezugriff auf die Kommunikationsobjekte durchgeführt. Die beim Read-Service übermittelten Daten werden aus dem entsprechendem Geräte-Parameter gelesen. Die Zuordnung von dem Geräte-Parameter zu dem Kommunikationsobjekt ist dem Objektverzeichnis zu entnehmen.



**Bild 12: Read-Funktion**

#### **Read-Service**

Der Read-Service übermittelt die Daten, die aus einem Kommunikationsobjekt gelesen werden sollen.

#### **Geräte-Parameter**

Die Daten, die mit dem Read-Service übermittleit werden, werden aus den Geräte-Parametern gelesen.

## Fehlermeldung

Der Service-Parameter Error-Type setzt sich aus folgenden Parametern zusammen:

- Error-Class;
- Error-Code;
- Additional-Code.

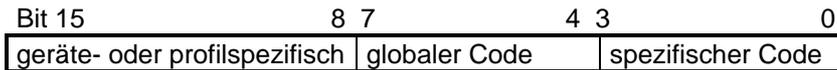
Error-Class und Error-Code:

<b>Error-Class</b>	<b>Error-Code</b>	<b>Bedeutung</b>
6 Access	2 Hardware-Fault	Der Zugriff auf das Objekt ist aufgrund eines Hardwarefehlers fehlgeschlagen. Genauere Angabe des Grundes im Additional-Code (glob.= 8 u. 9)
6 Access	3 Object-Access-Denied	Die Zugriffsrechte des Client sind nicht ausreichend
6 Access	5 Object-Attribute-Inconsistent	Ein Service-Parameter hat einen unzulässigen Wert angenommen. Genauere Angabe des Grundes im Additional-Code (glob.=1)
6 Access	6 Object-Access-Unsupported	Das Objekt ist kein Variable-Access Objekt
6 Access	7 Object-Non-Existent	Unter diesem Index existiert kein Objekt
8 Other	0 Other	Der Service wurde nicht ausgeführt. Genauere Angabe des Grundes im Additional-Code

**Additional-Code:**

Der Additional-Code setzt sich aus einem globalen, spezifischen und geräte- oder profilspezifischen Anteil zusammen. Die Angabe eines globalen oder spezifischen Codes ist optional. Das bedeutet: ist keine genaue Angabe des Fehlergrundes durch die Anwendung möglich, so muß der Additional-Code = 0000 hex ausgegeben werden. Der spezifische Code enthält eine genauere Beschreibung des im globalen Codes definierten Fehlergrundes. Entspricht der Fehlergrund nicht sinngemäß einem spezifischen Code, dann muß der spezifische Code = 0 angegeben werden. Der geräte- oder profilspezifische Code (Bit 8 bis 15) wird durch den Gerätehersteller oder andere Profile festgelegt.

2 Octets mit folgendem Aufbau:



Die Werte für die Bits 8 bis 15 sind reserviert und zur Zeit mit dem Wert 0 besetzt.

globaler Code [hex]	spezif. Code [hex]	Bedeutung
0	0	keine genauere Angabe des Fehlergrundes
1	0	Service-Parameter mit unzulässigem Wert
1	1	Subindex nicht vorhanden
2	0	Dienst kann momentan nicht ausgeführt werden
2	1	Dienst kann momentan nicht ausgeführt werden wegen Lokal-Steuerung
2	2	Dienst kann im momentanen Gerätezustand (Gerätesteuerung) nicht ausgeführt werden.
2	3	Dienst kann momentan nicht ausgeführt werden weil kein Objektverzeichnis vorhanden ist
3	0	Wertebereich eines Parameters ist verlassen (der Server will den Wert nicht liefern)
8	0	Hardware Fehler
8	1	Anwendung ausgefallen

Nicht aufgeführte Codes sind reserviert.

**Abbildung der Gerätefunktion auf die Kommunikation**

**Kommunikationsreferenz**

Die Service.res wird mit der Kommunikationsreferenz aus der Service.ind aufgerufen.

**Invoke-ID**

Die Service.res wird mit der Invoke-ID aus der Service.ind aufgerufen.

**Index**

Der Parameter Index definiert das zu lesende Objekt des Servers.

**Data**

Die Daten werden dem Client in dem Parameter 'Data' der Read.res übergeben.

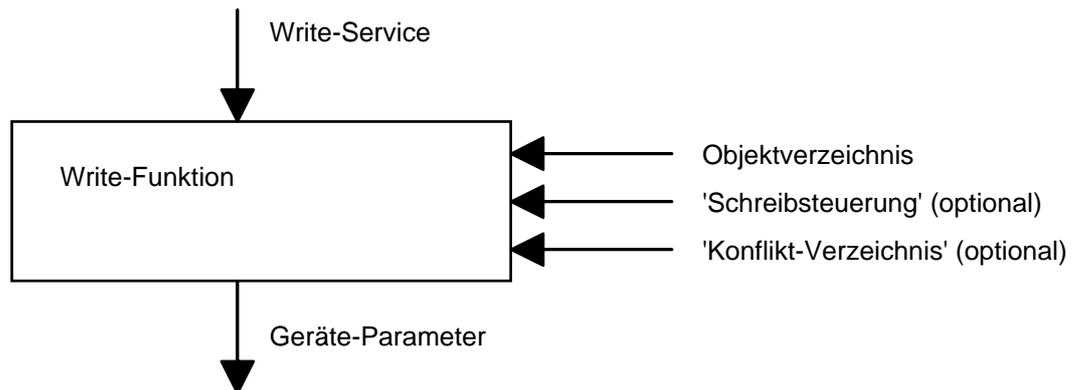
**Error-Type**

Der Parameter Error-Type enthält den Fehlergrund.

### 6.3.10. Write-Funktion

Mit dieser in Bild 13 dargestellten Funktion werden die mit dem Write-Service übermittelten Daten auf einen Geräte-Parameter geschrieben. Die Zuordnung Geräte-Parameter zu dem Kommunikationsobjekt ist dem Objektverzeichnis zu entnehmen. Ein Schreibzugriff mit einem Write-Service auf den Geräte-Parameter kann herstellerspezifisch mit der entsprechenden Fehlermeldung negativ quittiert werden.

Es gibt die beiden Funktionsausführungen Einzelparametrierung und Blockparametrierung (siehe folgende Gliederungspunkte). Die Blockparametrierung wird zur Parametrierung von untereinander abhängigen Parametern genutzt.



**Bild 13: Write-Funktion**

#### **Write-Service**

Der Write-Service übermittelt die Daten, die auf ein Kommunikationsobjekt geschrieben werden sollen. Die Datenlänge muß entsprechend der Angabe im Objektverzeichnis überprüft werden. Falls die Datenlänge nicht übereinstimmt, wird der Dienst negativ quittiert.

#### **Geräte-Parameter**

Die mit dem Write-Service übermittelten Daten werden auf die Geräteparameter geschrieben.

#### **Fehlermeldung**

Der Service-Parameter Error-Type setzt sich aus folgenden Parametern zusammen:

- Error-Class;
- Error-Code;
- Additional-Code.

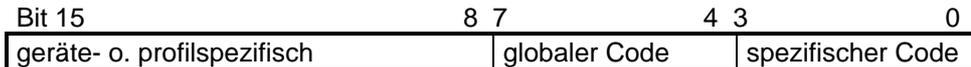
Error-Class und Error-Code:

Error-Class	Error-Code	Bedeutung
6 Access	2 Hardware-Fault	Der Zugriff auf das Objekt ist aufgrund eines Hardwarefehlers fehlgeschlagen. Genauere Angabe des Grundes im Additional-Code (glob.= 8 u. 9)
6 Access	3 Objekt-Access-Denied	Die Zugriffsrechte des Client sind nicht ausreichend
6 Access	5 Object-Attribute-Inconsistent	Ein Service-Parameter hat einen unzulässigen Wert angenommen. Genauere Angabe des Grundes im Additional-Code (glob.=1)
6 Access	6 Objekt-Access-Unsupported	Das Objekt ist kein Variable-Access Objekt
6 Access	7 Objekt-Non-Existent	Unter diesem Index existiert kein Objekt
6 Access	8 Type-Conflict	Die Daten entsprechen nicht dem Datentyp des Objektes
8 Other	0 Other	Der Service wurde nicht ausgeführt. Genauere Angabe des Grundes im Additional-Code

Additional-Code:

Der Additional-Code setzt sich aus einem globalen, spezifischen und geräte- oder profilspezifischen Anteil zusammen. Die Angabe eines globalen oder spezifischen Codes ist optional. Das bedeutet: ist keine genaue Angabe des Fehlergrundes durch die Anwendung möglich, so muß der Additional-Code = 0000 hex ausgegeben werden. Der spezifische Code enthält eine genauere Beschreibung des im globalen Codes definierten Fehlergrundes. Entspricht der Fehlergrund nicht sinngemäß einem spezifischen Code, dann muß der spezifische Code = 0 angegeben werden. Der geräte- oder profilspezifische Code (Bit 8 bis 15) wird durch den Gerätehersteller oder andere Profile festgelegt.

2 Octets mit folgendem Aufbau:



Die Werte für die Bits 8 bis 15 sind reserviert und zur Zeit mit dem Wert 0 besetzt.

globaler Code [hex]	spezif. Code [hex]	Bedeutung
0	0	keine genauere Angabe des Fehlergrundes
1	0	Service-Parameter mit unzulässigem Wert
1	1	Subindex nicht vorhanden
2	0	Dienst kann momentan nicht ausgeführt werden
2	1	Dienst kann momentan nicht ausgeführt werden wegen Lokal-Steuerung
2	2	Dienst kann im momentanen Gerätezustand (Gerätesteuerung) nicht ausgeführt werden.
2	3	Dienst kann momentan nicht ausgeführt werden, weil kein Objektverzeichnis vorhanden ist
3	0	Wertebereich eines Parameters ist verlassen
3	1	Wert des Parameters zu groß
3	2	Wert des Parameters zu klein
3	3	reserviert (wegen DRIVECOM Profil 21)
3	4	reserviert (wegen DRIVECOM Profil 21)
3	5	reserviert (wegen DRIVECOM Profil 21)
4	0	Kollision mit anderen Werten
4	1	Kommunikationsobjekt kann nicht auf Prozeßdaten abgebildet werden
4	2	Prozeßdaten-Länge überschritten
8	0	Hardware Fehler
8	1	Anwendung ausgefallen

Nicht aufgeführte Codes sind reserviert.

#### 6.3.10.1. Einzelparametrierung

Im Zustand Einzelparametrierung wird der geschriebene Parameterinhalt auf Grenzen und Konfliktfreiheit mit anderen Parameterinhalten geprüft. Nur bei positiver Quittierung wird der geschriebene Parameterinhalt wirksam.

#### 6.3.10.2. Blockparametrierung

Abhängigkeiten zwischen Parameterinhalten können bei Änderungen zu zeitweiligen Unverträglichkeiten führen, die eine bestimmte Reihenfolge oder sogar eine wechselseitige schrittweise Änderung verlangen würden (z.B. Fehlermeldung 'Kollision mit anderen Werten'). Für diesen Fall kann die 'Blockparametrierung' aktiviert und damit die Bedeutung der pos. Quittung geändert werden. Die pos. Quittung bedeutet, daß der Parameterinhalt angenommen, aber nicht wirksam wird.

Die Blockparametrierung bezieht sich auf die vom Hersteller definierten abhängigen Parameter. Die Prüfungen können sofort oder laufend - und - müssen mindestens beim Zurückschalten in die Einzelparametrierung durchgeführt werden. Dazu werden mindestens die abhängigen Parameter zwischengespeichert.

Fehler, die durch Verwendung der Blockparametrierung nicht auszuschließen sind, können auch zu einer sofortigen neg. Quittung führen.

Die Abhängigkeit von Parametern wird durch den Hersteller definiert und dem Anwender dokumentiert. Die Abhängigkeit kann aufgrund physikalischer Zusammenhänge bestehen oder wird dynamisch durch den Ein- und Austritt aus der Blockparametrierung definiert.

Die Übergänge zwischen Einzelparametrierung und Blockparametrierung werden durch den Parameter 'Schreibsteuerung' ausgelöst. Beim Verlassen der Blockparametrierung muß Verträglichkeit erreicht sein, damit die geänderten Parameter vollständig übernommen werden und der Zugriff positiv quittiert wird. Im Gegensatz dazu wird bei Zustandsänderung von Block- zu Einzelparametrierung und bestehender Unverträglichkeit der

Schreibzugriff auf den Parameter 'Schreibsteuerung' negativ quittiert. In diesem Fall behalten die unverträglichen Parameter den vor der Blockparametrierung gültigen Wert.

Ein Read-Zugriff während der Blockparametrierung liefert bei den abhängigen Parametern den Wert vor der Blockparametrierung.

**'Schreibsteuerung'**

Die Übergänge zwischen Einzelparametrierung und Blockparametrierung werden durch diesen Parameter ausgelöst.

False : Einzelparametrierung

True : Blockparametrierung

Folgende Aktionen werden bei einer Änderung des Parameterinhalts durchgeführt:

False -> True:

Wechsel von der Einzelparametrierung zu der Blockparametrierung und Aktualisierung des Parameters 'Konflikt-Verzeichnis'.

True -> False:

Wechsel von der Blockparametrierung zu der Einzelparametrierung.

Überprüfung auf Verträglichkeit:

- Wenn diese gegeben ist, werden die Parameterinhalte übernommen und der Schreibzugriff auf den Parameter 'Schreibsteuerung' positiv quittiert.
  
- Bei einer Unverträglichkeit bleiben die alten Inhalte der an der Kollision beteiligten Parameter wirksam. Im 'Konflikt-Verzeichnis' werden die Indices und die Fehlermeldung der am Konflikt beteiligten Parameter hinterlegt. Der Schreibzugriff auf den Parameter 'Schreibsteuerung' wird negativ quittiert.

Objektklasse:	optional
Zugriff:	schreib und lesbar
Prozeßdaten-Abbildung:	nicht möglich
Einheit:	-
Wertebereich:	Boolean
Pflichtbereich:	True, False
Ersatzwert:	False

**'Konflikt-Verzeichnis'**

Der Parameter enthält die Indizes und die Fehlermeldungen (Add.-Code) der am Konflikt beteiligten Parameter. Das 'Konflikt-Verzeichnis' muß bei negativer Quittierung von 'Blockschreib-Steuerung', kann aber auch nach jedem Schreibzugriff, innerhalb und/oder außerhalb des Zustands 'Blockparametrierung', aktualisiert werden.

Konflikt-Verzeichnis
Index
Add.-Code
Index
Add.-Code
...
...

Objektklasse:	optional
Zugriff:	lesbar
Prozeßdaten-Abbildung:	nicht möglich
Einheit:	-
Wertebereich:	Unsigned 16
Pflichtbereich:	-
Ersatzwert:	-

**Abbildung der Gerätefunktion auf die Kommunikation****Kommunikationsreferenz**

Die Service.res wird mit der Kommunikationsreferenz aus der Service.ind aufgerufen.

**Invoke-ID**

Die Service.res wird mit der Invoke-ID aus der Service.ind aufgerufen.

**Index**

Der Parameter Index definiert das zu schreibende Objekt des Servers.

**Data**

Die Daten werden vom Client in dem Parameter 'Data' der Service.ind übergeben.

**Error-Type**

Der Parameter Error-Type enthält den Fehlergrund.

Objektbeschreibung: 'Schreibsteuerung'

**Tabelle 2: Objektbeschreibung 'Schreibsteuerung'**

Objektattribut	Wert hex	Bedeutung
Index	600A	Schreibsteuerung
Variable-Name	-	nicht vorhanden
Object-Code	07	Simple-Variable
Data-Type-Index	01	Boolean
Length	01	1 Byte
Password	00	kein Paßwort
Access-Groups	00	keine Zugriffsgruppen
Access-Rights	0300	Read-All, Write-All
Local-Address	xxxx	herstellerspezifisch
Extension	-	nicht vorhanden

Objektbeschreibung: 'Konflikt-Verzeichnis'

**Tabelle 3: Objektbeschreibung 'Konflikt-Verzeichnis'**

Objektattribut	Wert hex	Bedeutung
Index	604C	Konflikt-Verzeichnis
Variable-Name	-	nicht vorhanden
Object-Code	08	Array
Number-Of-Elements	xx	n Elemente
Data-Type-Index	05	Octet-String
Length	02	2 Byte
Password	00	kein Paßwort
Access-Groups	00	keine Zugriffsgruppen
Access-Rights	0001	Read-All
Local-Address	xxxx	herstellerspezifisch
Extension	-	nicht vorhanden

Wertebeschreibung

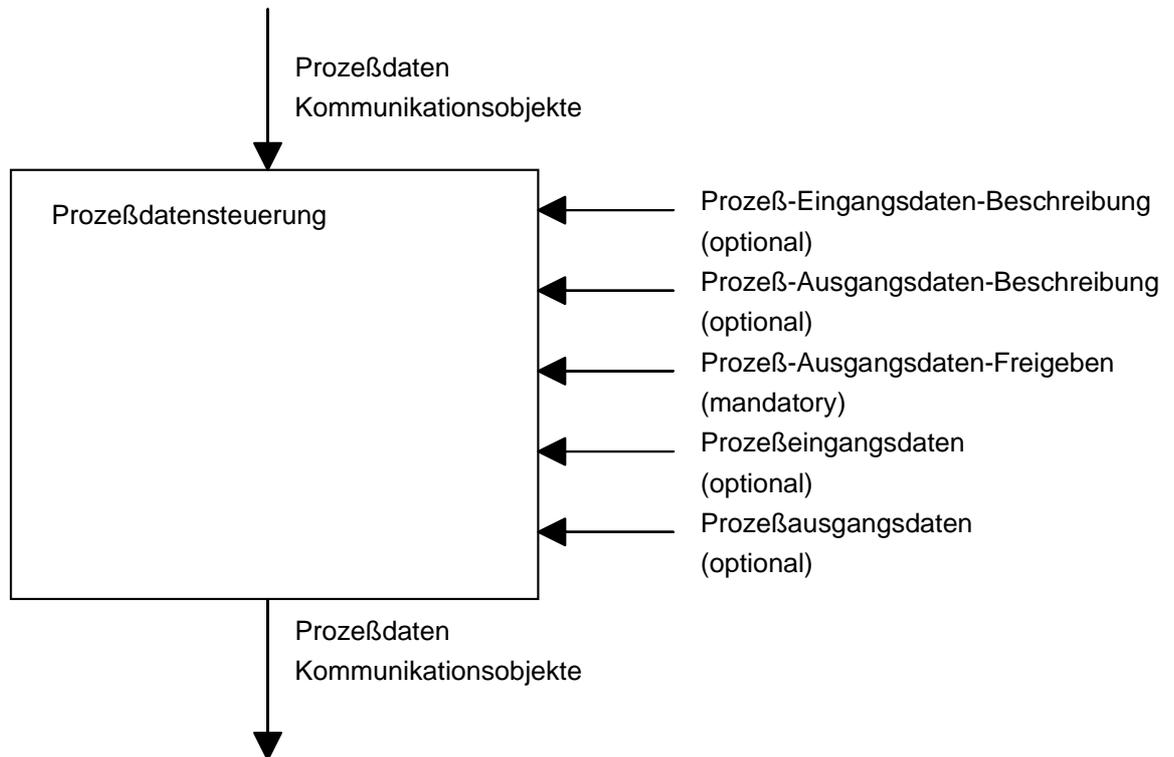
Subindex (hex)	Bedeutung
01	Index
02	Add.-Code
03	Index
04	Add.-Code
x	...

### 6.3.11. Prozeßdatensteuerung

Mit dieser Funktion werden die Prozeßdaten, die über den Prozeßdatenkanal übertragen werden, auf die Kommunikationsobjekte abgebildet.

Der Prozeßdatenkanal kann in einer Breite von 0 bis 26 Worten implementiert werden (siehe INTERBUS Management Dokumentation). Jedes Byte kann vom Server-Gerät gelesen und geschrieben werden.

Die Daten, die das Server-Gerät aus dem Prozeßdatenkanal liest, werden als Prozeßausgangsdaten bezeichnet. Die Daten, die das Gerät in den Prozeßdatenkanal schreibt, werden als Prozeßeingangsdaten bezeichnet.

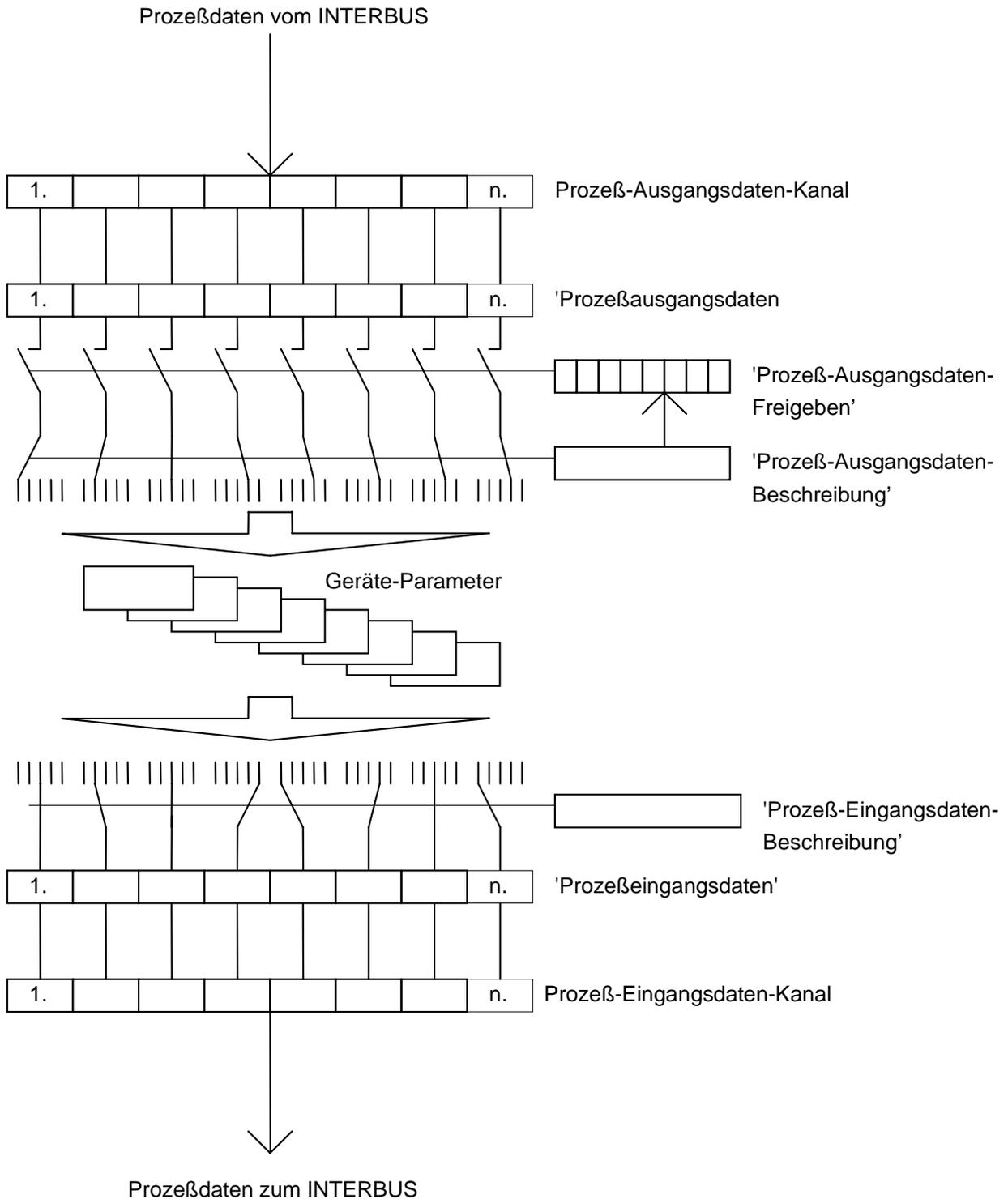


Durch die Abbildung der Kommunikationsobjekte auf die Prozeßeingangsdaten werden die entsprechenden Server-Geräte-Parameter über den Prozeßdatenkanal zyklisch ausgelesen.

Die Prozeßausgangsdaten, die auf ein Kommunikationsobjekt abgebildet sind, werden zyklisch auf die Geräte-Parameter übertragen.

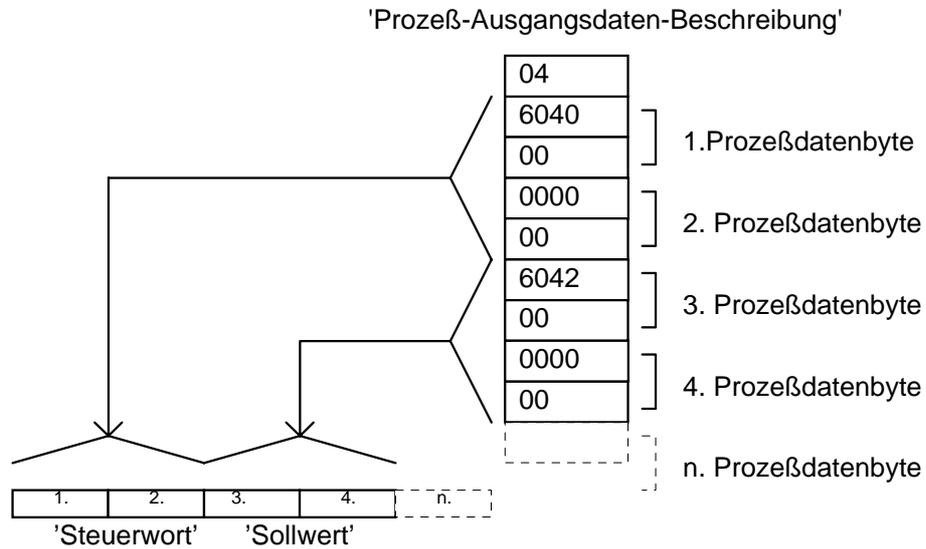
Die Zuordnung von Prozeßdaten zu bestimmten Kommunikationsobjekten kann parametrierbar sein. Hierzu dienen die Kommunikationsobjekte 'Prozeß-Eingangsdaten-Beschreibung' und 'Prozeß-Ausgangsdaten-Beschreibung'. Die Prozeßausgangsdaten können mit dem Parameter 'Prozeß-Ausgangsdaten-Freigeben' freigegeben und gesperrt werden.

**Funktionsweise:**



Die Funktionalität der Prozeßdatensteuerung kann mit Hilfe von Schaltern verdeutlicht werden. Der Parameter 'Prozeß-Eingangsdaten-Beschreibung' (Eingangsdaten für den Master) bestimmt, welches Geräte-Parameter auf welches Byte im INTERBUS-Prozeßdatenkanal abgebildet wird. Der Parameter 'Prozeß-Ausgangsdaten-Beschreibung' (Ausgangsdaten für den Master) bestimmt, welche Bytes der INTERBUS-Prozeßdaten auf welche Geräte-Parameter abgebildet wird.

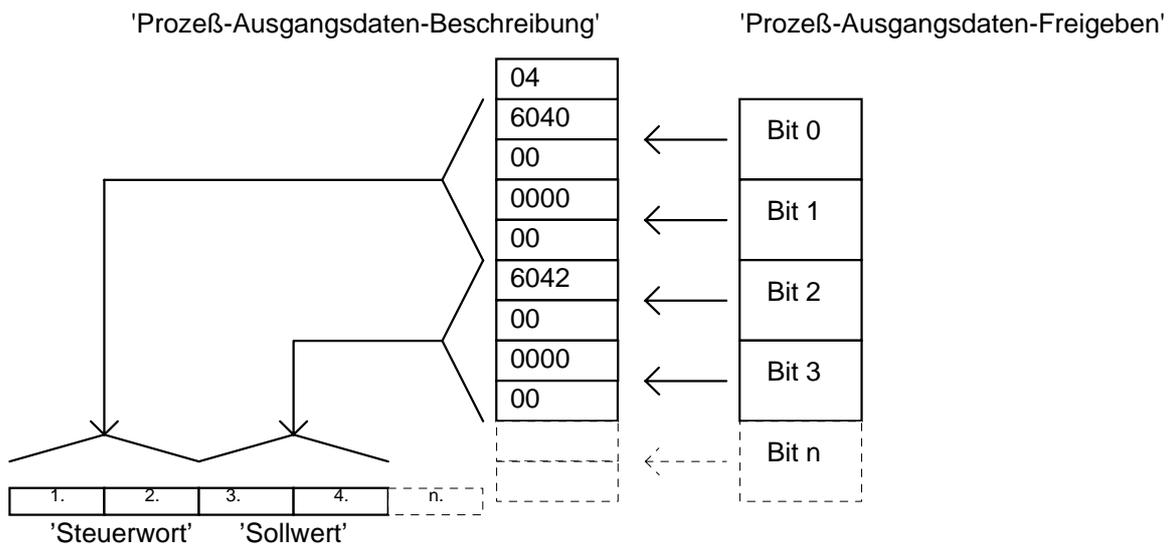
Die Zuordnung der Geräteparameter zu den Bytes des Prozeßdatenkanals erfolgt durch die Parameterkennung (Index, Subindex). Die Größe der Parameter 'Prozeß-Ausgangsdaten-Beschreibung' und 'Prozeß-Eingangsdaten-Beschreibung' - ist abhängig - von der Länge des Prozeßdatenkanals (durch die Hardware definiert). Im ersten Element der beiden Parameter ist die Länge des Prozeßdatenkanals in Bytes angegeben. Es können auf dem Prozeßdatenkanal Parameter - unterschiedlichste Datentypen (und damit Länge) abgebildet werden.



**ANMERKUNG:**

Die Parameterkennung 0000,xx ist zulässig und wird als 'nicht belegt' interpretiert.

Für eine Umkonfiguration der Prozeßausgangsdaten ist es notwendig, daß die Verbindung zwischen den INTERBUS-Prozeßdaten und dem Geräte-Parameter - definiert - unterbrochen werden kann. Hierzu dient der Parameter 'Prozeß-Ausgangsdaten-Freigeben' .



Ein Bit des Parameters 'Prozeß-Ausgangsdaten-Freigeben' ist einem Byte des Prozeßausgangsdatenkanals zugeordnet. Hierbei gilt folgende Zuordnung:

- Bit = 0            entsprechende Parameterabbildung ist gesperrt
- Bit = 1            entsprechende Parameterabbildung ist freigegeben.

Belegt ein Parameter mehrere Bytes, so gilt das Bit, daß dem ersten Byte des Parameters zugeordnet ist. Dieses entspricht der Position des Indexeintrages in der zugehörigen Prozeß-Ausgangs- bzw. Prozeß-Eingangsdatenbeschreibung. Die evtl. weiter zugehörigen Bits sind nicht relevant.

Durch das Beschreiben des Parameters 'Prozeß-Ausgangsdaten-Beschreibung' wird automatisch die Verbindung zwischen Prozeßausgangsdaten und dem entsprechendem Geräteparameter unterbrochen. Dieses geschieht durch Rücksetzen der entsprechenden Steuerbits im Parameter 'Prozeß-Ausgangsdaten-Freigeben'.

**Beispiel 1:**

Wird die Abbildung des ersten Parameters in dem Parameter 'Prozeß-Ausgangsdaten-Beschreibung' verändert (über Subindex<>0 des 'Prozeß-Ausgangsdaten-Beschreibung'), so wird das Bit 0 zurückgesetzt. Die restlichen Bits werden nicht geändert.

**Beispiel 2:**

Wird die Abbildung aller Parameter in dem Parameter 'Prozeß-Ausgangsdaten-Beschreibung' verändert (über Subindex=0 des 'Prozeß-Ausgangsdaten-Beschreibung'), so werden alle Bits zurückgesetzt.

Beispielhafte Belegung des Prozeßdatenkanals mit den folgenden Parametern:

- 1. Parameter A                    Datentyp :    Unsigned8
- 2. Parameter B                    Datentyp :    Unsigned8
- 3. nicht belegt
- 4. Parameter C                    Datentyp :    Unsigned8

Subindex des Parameters 'Prozeß-Eingangsdaten- Beschreibung	Bedeutung der Parameter-Elemente	Bytenummer im Prozeßdatenkanal
1	Länge des Prozeßdatenkanals	-
2	Index eines Geräte-Parameters A (8Bit-Parameter)	1
3	Subindex eines Geräte-Parameters A	1
4	Index eines Geräte-Parameters B (8Bit-Parameter)	2
5	Subindex eines Geräte-Parameters B	2
6	nicht belegt. Index = 0000	3
7	nicht belegt. Subindex =00	3
8	Index eines Geräte-Parameters C (8Bit-Parameter)	4
9	Subindex eines Geräte-Parameters C	4
...		...

Beispielhafte Belegung des Prozeßdatenkanals mit den folgenden Parametern:

1. Parameter A            Datentyp :    Unsigned16
2. Parameter B            Datentyp :    Unsigned8
3. Parameter C            Datentyp :    Unsigned8

Subindex des Parameters 'Prozeß-Eingangsdaten- Beschreibung	Bedeutung der Parameter-Elemente	Bytenummer im Prozeßdatenkanal
1	Länge des Prozeßdatenkanals	-
2	Index eines Geräte-Parameters A (16-Bit-Parameter)	1
3	Subindex eines Geräte-Parameters A	1
4	0000 (Durch Parameter A mitbelegt)	2
5	00 (Durch Parameter A mitbelegt)	2
6	Index eines Geräte-Parameters B (8Bit-Parameter)	3
7	Subindex eines Geräte-Parameters B	3
8	Index eines Geräte-Parameters C (8Bit-Parameter)	4
9	Subindex eines Geräte-Parameters C	4
...		...

Soll die Abbildung eines Parameters auf dem Prozeßdatenkanal durch einen Parameter mit einer größeren Datenbreite ersetzt werden, so müssen erst die Abbildungen der darauf folgenden Prozeßdatenkanal-Bytes gelöscht werden. Das Löschen wird durch das Beschreiben der entsprechenden Parameterkennung mit 0000,00 durchgeführt. Kann die Abbildung nicht durchgeführt werden, so wird der Dienst negativ quittiert.

**'Prozeß-Eingangsdaten-Beschreibung'**

Dieser Parameter enthält die Daten, die definieren, welche Prozeßeingangsdaten auf welche Kommunikationsobjekte abgebildet sind. Kommunikationsobjekte, die auf Prozeßeingangsdaten abgebildet werden können, sind in den jeweiligen Parameterbeschreibungen gekennzeichnet. Falls die Eingetragenen Indices mit den Subindices nicht vereinbar sind, wird das Beschreiben des Parameters nicht durchgeführt und eine Fehlermeldung abgesetzt.

Objektklasse	optional
Zugriff	schreib und lesbar
Prozeßdaten-Abbildung	nicht möglich
Einheit	-
Wertebereich	
Subindex 1:	Unsigned8
Subindex 2:	Unsigned16
Subindex 3:	Unsigned8
Subindex 4:	Unsigned16
Subindex 5:	Unsigned8
....	
Subindex n:	Unsigned16
Subindex n+1:	Unsigned8
Pflichtbereich	
Subindex 1:	4
Subindex 2:	6041
Subindex 3:	00
Subindex 4:	0000
Subindex 5:	00
Subindex 6:	6044
Subindex 7:	00
Subindex 8:	0000
Subindex 9:	00
Ersatzwert	siehe Pflichtbereich

**'Prozeß-Ausgangsdaten-Beschreibung'**

Dieser Parameter enthält die Daten, die definieren, auf welche Kommunikationsobjekte die Prozeßausgangsdaten abgebildet sind. Durch das Beschreiben dieses Parameters wird die Abbildung der Prozeßausgangsdaten auf das Kommunikationsobjekt unterbrochen. Kommunikationsobjekte, die auf Prozeßausgangsdaten abgebildet werden können, sind in den jeweiligen Parameterbeschreibungen gekennzeichnet. Falls die Eingetragenen Indices mit den Subindices nicht vereinbar sind, wird das Beschreiben des Parameters nicht durchgeführt und eine Fehlermeldung abgesetzt. Ein Schreibvorgang auf diesen Parameter bewirkt das Löschen der entsprechenden Bits im Parameter 'Prozeß-Ausgangsdaten-Freigeben.

Objektklasse	optional
Zugriff	schreib und lesbar
Prozeßdaten-Abbildung	nicht möglich
Einheit	-
Wertebereich	
Subindex 1:	Unsigned8
Subindex 2:	Unsigned16
Subindex 3:	Unsigned8
Subindex 4:	Unsigned16
Subindex 5:	Unsigned8
....	
Subindex n:	Unsigned16
Subindex n+1:	Unsigned8
Pflichtbereich	
Subindex 1:	4
Subindex 2:	6040
Subindex 3:	00
Subindex 4:	0000
Subindex 5:	00
Subindex 6:	6042
Subindex 7:	00
Subindex 8:	0000
Subindex 9:	00
Ersatzwert	siehe Pflichtbereich

**'Prozeß-Ausgangsdaten-Freigeben'**

Ein Bit des Parameters 'Prozeß-Ausgangsdaten-Freigeben' ist einem Byte des Prozeßausgangsdatenkanals zugeordnet. Hierbei gilt folgende Zuordnung:

Bit = 0           entsprechender Prozeßdatenwert ist gesperrt

Bit = 1           entsprechender Prozeßdatenwert ist freigegeben.

Belegt ein Parameter mehrere Bytes, so gilt der logische Zustand sämtlicher zugehöriger Bits:

TRUE = freigegeben

FALSE = gesperrt

Objektklasse	mandatory
Zugriff	schreib und lesbar
Prozeßdaten-Abbildung	nicht möglich
Einheit	-
Wertebereich	-
Pflichtbereich	0, FFhex
:	
Ersatzwert	siehe Pflichtbereich

**'Prozeßeingangsdaten'**

Dieser Parameter enthält die Daten die über den Prozeßdatenkanal an das Hostsystem übertragen werden. Die Größe des Parameters ist abhängig von der Länge des Prozeßdatenkanals.

Objektklasse	optional
Zugriff	schreib und lesbar
Prozeßdaten-Abbildung	nicht möglich
Einheit	-
Wertebereich	-
Pflichtbereich	-
Ersatzwert	-

**'Prozeßausgangsdaten'**

Dieser enthält die Daten die über den Prozeßdatenkanal an das Gerät übertragen werden. Die Größe des Parameters ist abhängig von der Länge des Prozeßdatenkanals.

Objektklasse	optional
Zugriff	nur lesbar
Prozeßdaten-Abbildung	nicht möglich
Einheit	-
Wertebereich	-
Pflichtbereich	-
Ersatzwert	-

**Abbildung der Gerätefunktion auf die Kommunikation**

Objektbeschreibung: 'Prozeß-Ausgangsdaten-Freigeben'

**Tabelle 4: Objektbeschreibung 'Prozeß-Eingangsdaten-Beschreibung'**

Objektattribut	Wert hex	Bedeutung
Index	6000	Prozeß-Eingangsdaten-Beschreibung
Variable-Name	-	nicht vorhanden
Object-Code	09	Record
Data-Type-Index	20	PDB-Struktur
Password	00	kein Paßwort
Access-Groups	00	keine Zugriffsgruppen
Access-Rights	0300	Read-All, Write-All
Local-Address	xxxx	herstellerspezifisch
Extension	-	nicht vorhanden

Objektbeschreibung: 'Prozeß-Ausgangsdaten-Beschreibung'

**Tabelle 5: Objektbeschreibung 'Prozeß-Ausgangsdaten-Beschreibung'**

Objektattribut	Wert hex	Bedeutung
Index	6001	Prozeß-Ausgangsdaten-Beschreibung
Variable-Name	-	nicht vorhanden
Object-Code	09	Record
Data-Type-Index	20	PDB-Struktur
Password	00	kein Paßwort
Access-Groups	00	keine Zugriffsgruppen
Access-Rights	0300	Read-All, Write-All
Local-Address	xxxx	herstellerspezifisch
Extension	-	nicht vorhanden

Objektbeschreibung: 'Prozeß-Ausgangsdaten-Freigeben'

**Tabelle 6: Objektbeschreibung 'Prozeß-Ausgangsdaten-Beschreibung'**

Objektattribut	Wert hex	Bedeutung
Index	6002	Prozeß-Ausgangsdaten-Freigeben <sup>1</sup>
Variable-Name	-	nicht vorhanden
Object-Code	07	Simple-Variable
Data-Type-Index	0A	Octet-String
Length	0n	n Byte (herstellerspezifisch)
Password	00	kein Paßwort
Access-Groups	00	keine Zugriffsgruppen
Access-Rights	0300	Read-All, Write-All
Local-Address	xxxx	herstellerspezifisch
Extension	-	nicht vorhanden

Objektbeschreibung: 'Prozeßeingangsdaten'

**Tabelle 7: Objektbeschreibung 'Prozeßeingangsdaten'**

Objektattribut	Wert hex	Bedeutung
Index	6010	Prozeßeingangsdaten
Variable-Name	-	nicht vorhanden
Object-Code	07	Simple-Variable
Data-Type-Index	0A	Octet-String
Length	n	n Byte
Password	00	kein Paßwort
Access-Groups	00	keine Zugriffsgruppen
Access-Rights	0300	Read-All, Write-All
Local-Address	xxxx	herstellerspezifisch
Extension	-	nicht vorhanden

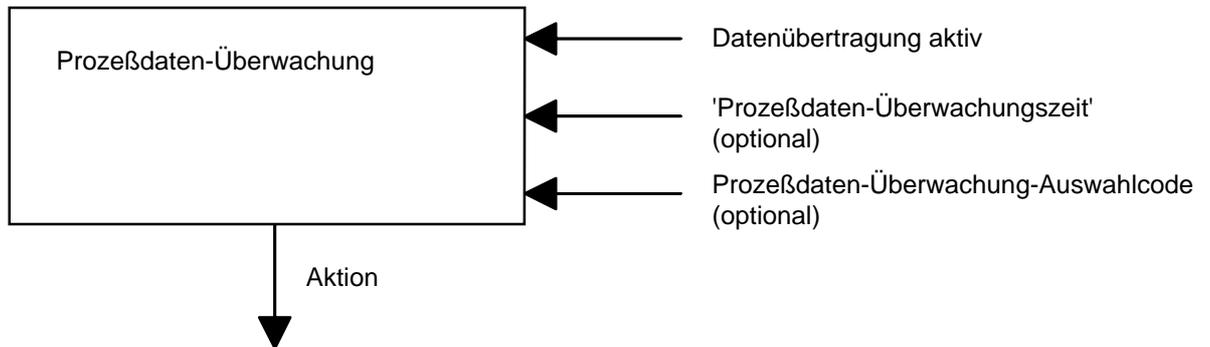
Objektbeschreibung: 'Prozeßausgangsdaten'

**Tabelle 8: Objektbeschreibung 'Prozeßausgangsdaten'**

Objektattribut	Wert hex	Bedeutung
Index	6011	Prozeßausgangsdaten
Variable-Name	-	nicht vorhanden
Object-Code	07	Simple-Variable
Data-Type-Index	0A	Octet-String
Length	n	n Byte
Password	00	kein Paßwort
Access-Groups	00	keine Zugriffsgruppen
Access-Rights	0300	Read-All, Write-All
Local-Address	xxxx	herstellerspezifisch
Extension	-	nicht vorhanden

### 6.3.12. Prozeßdaten-Überwachung

Mit der Prozeßdaten-Überwachung (siehe Bild 14) wird die Datenübertragung über den Prozeßdatenkanal überwacht. Falls die Datenübertragung länger als die eingestellte Überwachungszeit inaktiv ist, wird eine parametrierbare Aktion ausgelöst.



**Bild 14: Prozeßdaten-Überwachung**

#### Datenübertragung aktiv

Dieses interne Signal zeigt an, daß neue Prozeßdaten übertragen wurden.

#### 'Prozeßdaten-Überwachungszeit'

Der Parameter 'Prozeßdaten-Überwachungszeit' legt die Zeit fest, die maximal verstreichen darf, bis neue Prozeßdaten über den Prozeßdatenkanal übertragen werden. Die Werte der Überwachungszeit werden in ms angegeben. Der Wert 65535 (FFFF hex) schaltet die Prozeßdaten-Überwachung aus. Der Wertebereich kann vom Hersteller gerätespezifisch eingeschränkt werden.

Objektklasse:	optional
Zugriff:	schreib und lesbar
Prozeßdaten-Abbildung:	nicht möglich
Einheit:	ms
Wertebereich:	0 bis 65534 ms und 65535 = ausgeschaltet
Pflichtbereich:	65535 ( ausgeschaltet )
Ersatzwert:	65535 ( ausgeschaltet )

**'Prozeßdaten-Überwachung-Auswahlcode'**

Der 'Prozeßdaten-Überwachung-Auswahlcode' legt die Funktion fest, die ausgelöst wird, wenn keine neuen Prozeßdaten innerhalb der Überwachungszeit übertragen werden. Falls das Kommunikationsobjekt 'Prozeßdaten-Überwachung-Auswahlcode' nicht vorhanden ist, wird keine Aktion ausgeführt.

Auswahlcode	Bedeutung der Auswahl-Funktion
-32768 ... -1	herstellerspezifisch
0	keine Aktion
1 ... 32767	reserviert für weitere Profile

Objektklasse:	optional
Zugriff:	schreib und lesbar
Prozeßdaten-Abbildung:	nicht möglich
Einheit:	ms
Wertebereich:	Integer16
Pflichtbereich:	0 ( keine Aktion )
Ersatzwert:	0 ( keine Aktion )

**Aktion**

Es wird die Funktion ausgelöst, die im 'Prozeßdaten-Überwachung-Auswahlcode' definiert ist.

**Fehlermeldung**

Ja, siehe Read- oder Write-Funktion.

**Abbildung der Gerätefunktion auf die Kommunikation**

Objektbeschreibung: 'Prozeßdaten-Überwachungszeit'

**Tabelle 9: Objektbeschreibung 'Prozeßdaten-Überwachungszeit'**

Objektattribut	Wert hex	Bedeutung
Index	6003	Prozeßdaten-Überwachungszeit
Variable-Name	-	nicht vorhanden
Object-Code	07	Simple-Variable
Data-Type-Index	06	Unsigned16
Length	02	2 Byte
Password	00	kein Paßwort
Access-Groups	00	keine Zugriffsgruppen
Access-Rights	0300	Read-All, Write-All
Local-Address	xxxx	herstellerspezifisch
Extension	-	nicht vorhanden

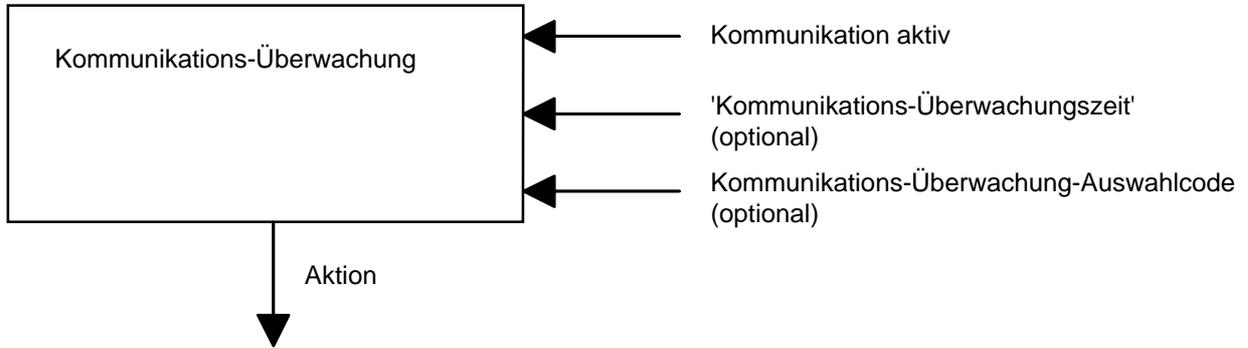
Objektbeschreibung: 'Prozeßdaten-Überwachung-Auswahlcode'

**Tabelle 10: Objektbeschreibung 'Prozeßdaten-Überwachung-Auswahlcode'**

Objektattribut	Wert hex	Bedeutung
Index	6004	Prozeßdaten-Überwachung-Auswahlcode
Variable-Name	-	nicht vorhanden
Object-Code	07	Simple-Variable
Data-Type-Index	03	Integer16
Length	02	2 Byte
Password	00	kein Paßwort
Access-Groups	00	keine Zugriffsgruppen
Access-Rights	0300	Read-All, Write-All
Local-Address	xxxx	herstellerspezifisch
Extension	-	nicht vorhanden

### 6.3.13. Kommunikations-Überwachung

Mit der Kommunikations-Überwachung (siehe Bild 15) wird die Datenübertragung über den Kommunikationskanal überwacht. Falls die Datenübertragung länger als die eingestellte Überwachungszeit inaktiv ist, wird eine parametrierbare Aktion ausgelöst. Die Überwachungszeit sollte an die Busübertragungszeit angepaßt werden.



**Bild 15: Kommunikations-Überwachung**

#### **Kommunikation aktiv**

Kommunikationsaktionen über den Kommunikationskanal.

#### **'Kommunikations-Überwachungszeit'**

Der Parameter 'Kommunikations-Überwachungszeit' legt die Zeit fest, die maximal verstreichen darf, bis neue Daten über den Kommunikationskanal übertragen werden. Die Werte der Überwachungszeit werden in ms angegeben. Der Wert 65535 (FFFF hex) schaltet die Kommunikations-Überwachung aus. Der Wertebereich kann vom Hersteller gerätespezifisch eingeschränkt werden.

Objektklasse:	optional
Zugriff:	schreib und lesbar
Prozeßdaten-Abbildung:	nicht möglich
Einheit:	ms
Wertebereich:	0 bis 65534 ms und 65535 = ausgeschaltet
Pflichtbereich:	65535 ( ausgeschaltet )
Ersatzwert:	65535 ( ausgeschaltet )

**'Kommunikations-Überwachung-Auswahlcode'**

Der 'Kommunikations-Überwachungs-Auswahlcode' legt die Funktion fest, die ausgelöst wird, wenn keine neuen Daten innerhalb der Kommunikations-Überwachungszeit über den Kommunikationskanal übertragen werden.

Auswahlcode	Bedeutung der Auswahl-Funktion
-32768 ... -1	herstellerspezifisch
0	keine Aktion
1 ... 32767	reserviert für weitere Profile

Objektklasse:	optional
Zugriff:	schreib und lesbar
Prozeßdaten-Abbildung:	nicht möglich
Einheit:	ms
Wertebereich:	Integer16
Pflichtbereich:	0 ( keine Aktion )
Ersatzwert:	0 ( keine Aktion )

**Aktion**

Es wird die Funktion ausgelöst, die im 'Kommunikations-Überwachung-Auswahlcode' definiert ist.

**Fehlermeldung**

Ja, siehe Read- oder Write-Funktion.

**Abbildung der Gerätefunktion auf die Kommunikation**

Objektbeschreibung: 'Kommunikations-Überwachungszeit'

**Tabelle 11: Objektbeschreibung 'Kommunikations-Überwachungszeit'**

Objektattribut	Wert hex	Bedeutung
Index	6005	Kommunikations-Überwachungszeit
Variable-Name	-	nicht vorhanden
Object-Code	07	Simple-Variable
Data-Type-Index	06	Unsigned16
Length	02	2 Byte
Password	00	kein Paßwort
Access-Groups	00	keine Zugriffsgruppen
Access-Rights	0300	Read-All, Write-All
Local-Address	xxxx	herstellerspezifisch
Extension	-	nicht vorhanden

Objektbeschreibung: 'Kommunikations-Überwachung-Auswahlcode'

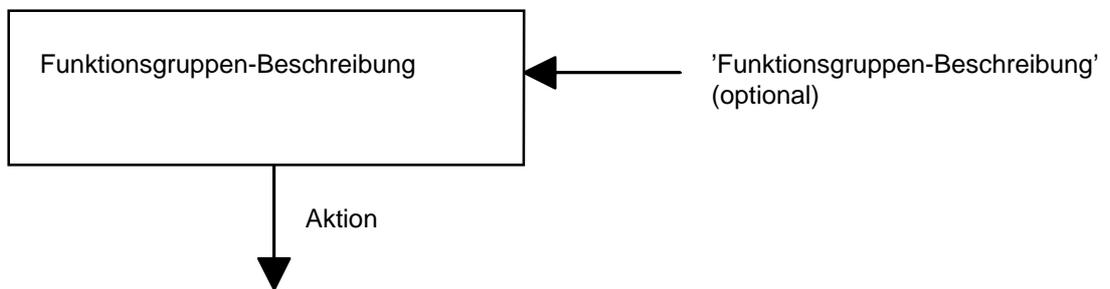
**Tabelle 12: Objektbeschreibung 'Kommunikations-Überwachung-Auswahlcode'**

Objektattribut	Wert hex	Bedeutung
Index	6006	Kommunikations-Überwachung-Auswahlcode
Variable-Name	-	nicht vorhanden
Object-Code	07	Simple-Variable
Data-Type-Index	03	Integer16
Length	02	2 Byte
Password	00	kein Paßwort
Access-Groups	00	keine Zugriffsgruppen
Access-Rights	0300	Read-All, Write-All
Local-Address	xxx	herstellerspezifisch
Extension	-	nicht vorhanden

### 6.4. Sensor/Aktor-Funktion

#### 6.4.1. Funktionsgruppen-Beschreibung

Die Funktionsgruppen-Beschreibung (siehe Bild 16) enthält Informationen über die Gerätefunktionsgruppen, die aktuell von dem Gerät unterstützt werden.



**Bild 16: Funktionsgruppen-Beschreibung**

#### 'Funktionsgruppen-Beschreibung'

Dieser Parameter enthält Informationen über die Funktionsgruppen des Gerätes. Der Parameter ist ein Feld mit 4 \* n Einträgen. Wertebereich von n (max.-PDU-Länge).

Attribut	Wert
Index, Name	600F, Funktionsgruppen-Beschreibung
Objektklasse	optional
Zugriff	nur lesbar
Prozeßdaten-Abbildung	herstellerspezifisch
Einheit	-
Wertebereich	-
Pflichtbereich	-
Defaultwert	-
Ersatzwert	-

Beispiel für ein DRIVECOM Gerät:

	Profil-Nummer		Funktionsgruppen-Nummer	
	Profil-Gruppe	Version	Funktionsgr.-Kennung	Version
1210	Sensor/Aktor	2	Kommunikationsfunktion	0
2210	DRIVECOM	1	Zustandsm. Gerätest.	x
22x0	DRIVECOM	x	eine Anwendungsfunktion	x

### Profil-Nummer

In diesem Parameter wird gekennzeichnet, in welchem Profil die in diesem Gerät implementierte Funktion beschrieben ist. Die Bedeutung der Profilnummer ist im Gliederungspunkt 6.3.3. Verbindungsaufbau festgelegt.

### Funktionsgruppenkennung:

Dieser Parameter kennzeichnet eine Funktionsgruppe innerhalb einer Profil-Gruppe.

Funktionsgruppenkennung für das Sensor/Aktor-Profil:

Funktionsgruppenkennung	Bedeutung
0	reserviert
1	Kommunikationsfunktionen
2	Sensor/Aktor-Funktionen

Für das DRIVECOM-Profil sind z.B. folgende Funktionsgruppen definiert.

Funktionsgruppenkenn.	Bedeutung
0	reserviert
1	Zustandsmaschine Gerätesteuerung
2	Allgemeine Funktionen
3	Betriebsarten-Funktionen
4	Geschwindigkeits-Funktionen 1(alte)
5	Geschwindigkeits-Funktionen 2(neu)
6	Lage-Funktionen
7	Momenten-Funktionen
8	Sollwertgenerator
9	Faktor-Funktionen
10	Programm-Funktionen

**Funktionsversion:**

Dieser Parameter kennzeichnet die Version einer Geräte-Funktionsgruppe.

Wertebereich: 0-FF hex

BEISPIEL 1:

	Profil-Nummer		Funktionsgruppen-Nummer	
	Profil-Gruppe	Version	Funktionsgr.-Kennung	Version
01 02 01 00	Sensor/Aktor	2	Kommunikationsfunktion	0
01 02 01 00	Sensor/Aktor	2	Kommunikationsfunktion	0
01 02 02 00	Sensor/Aktor	2	Sensor/Aktor-Funktionen	0
02 01 00 00	DRIVECOM 21	1	-	-
02 02 01 00	DRIVECOM	2	Gerätesteuerung	0
02 02 02 00	DRIVECOM	2	Geschw.Funktion alt	0
02 02 03 00	DRIVECOM	2	Geschw.Funktion neu	0
02 02 04 00	DRIVECOM	2	Lagefunktionen	0
02 02 05 00	DRIVECOM	2	Momentenfunktionen	0

Fehlermeldung: Ja, siehe Read- oder Write-Funktion.

**Fehlermeldung**

Ja, siehe Read- oder Write-Funktion.

**Abbildung der Gerätefunktion auf die Kommunikation**

Objektbeschreibung: 'Funktionsgruppen-Beschreibung'

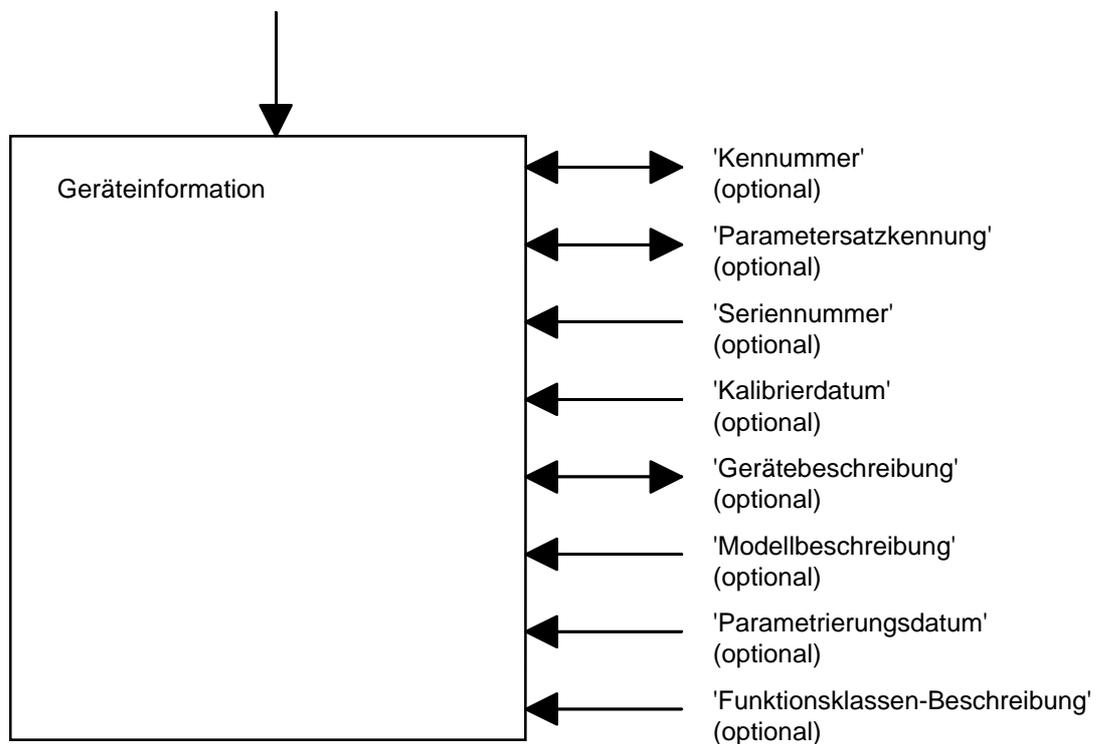
**Tabelle 13: Objektbeschreibung 'Funktionsgruppen-Beschreibung'**

Objektattribut	Wert	Bedeutung
Index	600F	Funktionsgruppen-Beschreibung
Variable-Name	-	nicht vorhanden
Object-Code	08	Array
Number of Elements	n	n Elemente
Data-Type-Index	A	Octet-String
Length	4	4 Byte
Password	00	kein Paßwort
Access-Groups	00	keine Zugriffsgruppen
Access-Rights	0001	Read-All
Local-Address	xxxx	herstellerspezifisch
Extension	-	nicht vorhanden

## 6.4.2. Geräteinformation

Mit dieser in Bild 17 dargestellten Funktion werden folgende Geräteinformationen verwaltet:

- Kennnummer;
- Parametersatzkennung;
- Seriennummer;
- Kalibrierdatum;
- Gerätebeschreibung;
- Modellbeschreibung;
- Parametrierungsdatum;
- Funktionsklassen-Beschreibung.



**Bild 17: Geräteinformation**

**'Kennnummer'**

In dieser Funktion kann der Anwender eine Kennnummer nichtflüchtig in dem Gerät hinterlegen. Der Wert ist frei wählbar. Es kann daher im Bus jede Nummer unter Umständen mehrfach vergeben sein. Durch geeignete Vergabe der Kennnummern im Bus kann z.B. der Austausch eines Gerätes oder das Vertauschen von Busanschlüssen erkannt werden.

Objektklasse:	optional
Zugriff:	schreib und lesbar
Prozeßdaten-Abbildung:	nicht möglich
Einheit:	keine
Wertebereich:	Unsigned16
Pflichtbereich:	-
Ersatzwert:	-

**'Parametersatzkennung'**

Dieser Parameter dient zur Kennzeichnung des aktuell wirksamen Geräte-Parametersatzes. Kann das Gerät die empfangenen Geräte-Parameterwerte nicht netzausfallsicher speichern, so wird nach dem Einschalten der Netzspannung diese Parametersatzkennung vom Gerät selbständig auf 0 gesetzt. Diese Information kann der Gerätenutzer auswerten und entsprechend neu initialisieren.

0	Der Parametersatz des Gerätes wurde noch nicht über den Bus initialisiert.
1-254	Der Parametersatz des Gerätes wurde über den Bus initialisiert. Zur Kennung hat er die frei wählbare Kenn-Nr. 1-254 erhalten.
255	Das Gerät wurde oder ist auf Local-Mode geschaltet und es ist nicht sichergestellt, daß der zuvor geladene Parametersatz unverändert ist.

Der Gerätenutzer kann alle Werte zwischen 0 und 255 eintragen.

Objektklasse:	optional
Zugriff:	schreib und lesbar
Prozeßdaten-Abbildung:	nicht möglich
Einheit:	keine
Wertebereich:	Unsigned8
Pflichtbereich:	-
Ersatzwert:	-

**'Seriennummer'**

Dieser Parameter enthält die Seriennummer. Dieser kann nur gelesen werden.

Objektklasse:	optional
Zugriff:	nur lesbar
Prozeßdaten-Abbildung:	nicht möglich
Einheit:	keine
Wertebereich:	Unsigned32
Pflichtbereich:	-
Ersatzwert:	-

**'Kalibrierdatum'**

Dieser Parameter enthält das Kalibrierdatum. Dieser kann nur gelesen werden.

Objektklasse:	optional
Zugriff:	nur lesbar
Prozeßdaten-Abbildung:	nicht möglich
Einheit:	keine
Wertebereich:	siehe Datentyp 'Date'
Pflichtbereich:	-
Ersatzwert:	-

**'Gerätebeschreibung'**

Dieser Parameter enthält einen Text, der vom Gerätebenutzer in diesem hinterlegt wurde. Die Gerätebeschreibung wird in einen nichtflüchtigen Speicher abgelegt. Hier kann z.B. der Gerätebenutzer eine Beschreibung über die Gerätebenutzung in der Anlage hinterlegen. Der Text hat eine feste Länge von 64 Zeichen.

Objektklasse:	optional
Zugriff:	schreib und lesbar
Prozeßdaten-Abbildung:	nicht möglich
Einheit:	keine
Wertebereich:	siehe Datentyp 'Visible-String'
Pflichtbereich:	-
Ersatzwert:	-

**'Modellbeschreibung'**

Dieser Parameter enthält einen Text, der vom Gerätehersteller in diesem hinterlegt wurde. Hier kann z.B. eine kurze Gerätebeschreibung enthalten sein. Der Text hat eine feste Länge von 64 Zeichen und kann nur ausgelesen werden.

Objektklasse:	optional
Zugriff:	nur lesbar
Prozeßdaten-Abbildung:	nicht möglich
Einheit:	keine
Wertebereich:	siehe Datentyp 'Visible-String'
Pflichtbereich:	-
Ersatzwert:	-

**'Parametrierungsdatum'**

Dieser Parameter enthält das Datum der letzten Änderung eines beliebigen Geräteparameters.

Objektklasse:	optional
Zugriff:	nur lesbar
Prozeßdaten-Abbildung:	nicht möglich
Einheit:	keine
Wertebereich:	siehe Datentyp 'Date'
Pflichtbereich:	-
Ersatzwert:	-

**Abbildung der Gerätefunktion auf die Kommunikation**

Objektbeschreibung: 'Kennnummer'

**Tabelle 14: Objektbeschreibung 'Kennnummer'**

Objektattribut	Wert hex	Bedeutung
Index	6008	Kennnummer
Variable-Name	-	nicht vorhanden
Object-Code	07	Simple-Variable
Data-Type-Index	06	Unsigned16
Length	02	2 Byte
Password	00	kein Paßwort
Access-Groups	00	keine Zugriffsgruppen
Access-Rights	0300	Read-All, Write-All
Local-Address	xxxx	herstellerspezifisch
Extension	-	nicht vorhanden

Objektbeschreibung: 'Parametersatzkennung'

**Tabelle 15: Objektbeschreibung 'Parametersatzkennung'**

Objektattribut	Wert hex	Bedeutung
Index	6009	Parametersatzkennung
Variable-Name	-	nicht vorhanden
Object-Code	07	Simple-Variable
Data-Type-Index	05	Unsigned8
Length	01	1 Byte
Password	00	kein Paßwort
Access-Groups	00	keine Zugriffsgruppen
Access-Rights	0300	Read-All, Write-All
Local-Address	xxxx	herstellerspezifisch
Extension	-	nicht vorhanden

Objektbeschreibung: 'Seriennummer'

**Tabelle 16: Objektbeschreibung 'Seriennummer'**

Objektattribut	Wert hex	Bedeutung
Index	600A	Seriennummer
Variable-Name	-	nicht vorhanden
Object-Code	07	Simple-Variable
Data-Type-Index	07	Unsigned32
Length	04	4 Byte
Password	00	kein Paßwort
Access-Groups	00	keine Zugriffsgruppen
Access-Rights	0001	Read-All
Local-Address	xxxx	herstellerspezifisch
Extension	-	nicht vorhanden

Objektbeschreibung: 'Kalibrierdatum'

**Tabelle 17: Objektbeschreibung 'Kalibrierdatum'**

Objektattribut	Wert hex	Bedeutung
Index	600B	Kalibrierdatum
Variable-Name	-	nicht vorhanden
Object-Code	07	Simple-Variable
Data-Type-Index	0B	Date
Length	07	7 Byte
Password	00	kein Paßwort
Access-Groups	00	keine Zugriffsgruppen
Access-Rights	0001	Read-All
Local-Address	xxxx	herstellerspezifisch
Extension	-	nicht vorhanden

Objektbeschreibung: 'Gerätebeschreibung'

**Tabelle 18: Objektbeschreibung 'Gerätebeschreibung'**

Objektattribut	Wert hex	Bedeutung
Index	600C	Gerätebeschreibung
Variable-Name	-	nicht vorhanden
Object-Code	07	Simple-Variable
Data-Type-Index	09	Visible-String
Length	40	64 Byte
Password	00	kein Paßwort
Access-Groups	00	keine Zugriffsgruppen
Access-Rights	0300	Read-All, Write-All
Local-Address	xxxx	herstellerspezifisch
Extension	-	nicht vorhanden

Objektbeschreibung: 'Modellbeschreibung'

**Tabelle 19: Objektbeschreibung 'Modellbeschreibung'**

Objektattribut	Wert hex	Bedeutung
Index	600D	Modellbeschreibung
Variable-Name	-	nicht vorhanden
Object-Code	07	Simple-Variable
Data-Type-Index	09	Visible-String
Length	40	64 Byte
Password	00	kein Paßwort
Access-Groups	00	keine Zugriffsgruppen
Access-Rights	0001	Read-All
Local-Address	xxxx	herstellerspezifisch
Extension	-	nicht vorhanden

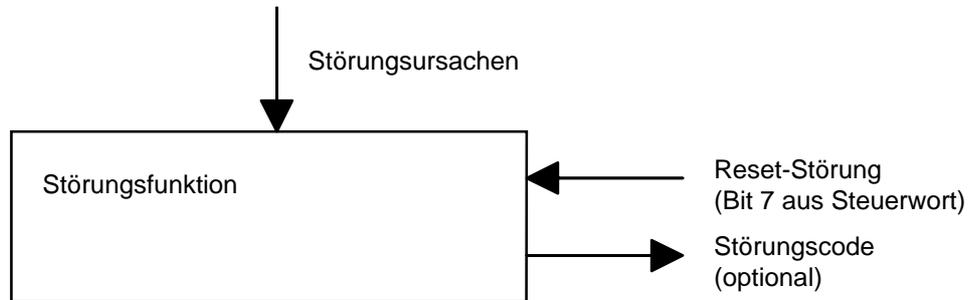
Objektbeschreibung: 'Parametrierungsdatum'

**Tabelle 20: Objektbeschreibung 'Parametrierungsdatum'**

Objektattribut	Wert hex	Bedeutung
Index	600E	Parametrierungsdatum
Variable-Name	-	nicht vorhanden
Object-Code	07	Simple-Variable
Data-Type-Index	0B	Date
Length	07	7 Byte
Password	00	kein Paßwort
Access-Groups	00	keine Zugriffsgruppen
Access-Rights	0001	Read-All
Local-Address	xxxx	herstellerspezifisch
Extension	-	nicht vorhanden

6.4.3. Störungsfunktion

Die Störungsfunktion (siehe Bild18) verwaltet den Parameter 'Störungscode'. Dieser wird durch eine Regler-Störung auf den entsprechenden Wert gesetzt (siehe Störungsliste). Durch die Aktion Störungs-Reset der Gerätesteuerung wird der Parameter auf den Wert 0 gesetzt.



**Bild 18: Störungsfunktion**

**'Störungscode'**

Der 'Störungscode' wird als ein Octet-String von der Länge 2 Byte dargestellt. Die Codierung erfolgt hierarchisch von einer groben zu einer immer feiner werdenden Unterscheidung.

Bit	Gruppierung
15 ... 12	Hauptgruppen
11 ... 8	Untergruppen
7 ... 0	Details

Ist der Regler im Zustand Störung, so enthält der Parameter einen Wert ungleich 0. Befindet sich der Regler nicht im Zustand Störung, dann enthält dieser den Wert 0.

Gesetz dem Fall, daß genau eine Störungsursache vorliegt, dann kann der dieser Ursache zugeordnete Wert im Parameter 'Störungscode' so lange unverändert ausgelesen werden, bis der Zustand Störung verlassen wird. Der Zustand Störung wird dann verlassen, wenn die Störungsursache beseitigt ist und der Befehl Störung-Reset gegeben wird.

Liegen mehrere Störungsursachen gleichzeitig vor, so wird im Parameter 'Störungscode' eine davon angezeigt. Wird nur die angezeigte beseitigt und der Befehl Reset-Störung gegeben, so wird der Zustand Störung aufgrund der übrigen vorliegenden nicht verlassen. Im Objekt 'Störungscode' wird dann eine dieser Störungsursachen angezeigt.

Objektklasse:	optional
Zugriff:	nur lesbar
Prozeßdaten-Abbildung:	nicht möglich
Einheit:	keine
Wertebereich:	0 bis 65535
Pflichtbereich:	-
Ersatzwert:	-

Tabelle 21: StörungsCodes und Störungsursachen

Code hex	Bedeutung
<b>0000</b>	<b>keine Störung</b>
<b>1000</b>	<b>Störung allgemein</b>
<b>2000</b>	<b>Strom</b>
2100	Strom geräteeingangseitig
2200	Strom geräteintern
2300	Strom geräteausgangseitig
<b>3000</b>	<b>Spannung</b>
3100	Netzspannung
3200	Spannung geräteintern
3300	Ausgangsspannung
3400	Versorgungsspannung
<b>4000</b>	<b>Temperatur</b>
4100	Temperatur Umgebung
4200	Temperatur Gerät
4300	Temperatur extern
4400	Temperatur Versorgung
<b>5000</b>	<b>Geräte-Hardware (nur innerhalb des Gerätegehäuses)</b>
5100	Versorgung
5200	Steuerung
5300	Bedieneinheit
<b>6000</b>	<b>Geräte-Software</b>
6100	Interne Software
6200	Anwender-Software
6300	Datensatz

fortgesetzt

**Tabelle 21: StörungsCodes und Störungsursachen (abgeschlossen)**

Code hex	Bedeutung
<b>7000</b>	<b>Zusatzbaugruppen</b>
7100	Leistung
7200	Meßschaltung
7300	Sensor
7400	Rechenschaltung
7500	Kommunikation
7600	Datenspeicher
<b>8000</b>	<b>Überwachung</b>
8100	Kommunikation
8110	Prozeßdaten-Überwachung
8120	Host-Überwachung
8200	Regelung
<b>9000</b>	<b>Externe Störung</b>
<b>F000</b>	<b>Zusatzfunktionen</b>

Nicht aufgeführte Codes sind reserviert.

ANMERKUNG: Weitere StörungsCodes sind in den Profilen definiert.

#### Fehlermeldung

Ja, siehe Read- oder Write-Funktion.

#### Abbildung der Gerätefunktion auf die Kommunikation

Objektbeschreibung: 'Störungscode'

**Tabelle 22: Objektbeschreibung 'Störungscode'**

Objektattribut	Wert hex	Bedeutung
Index	603F	Störungscode
Variable-Name	-	nicht vorhanden
Object-Code	07	Simple-Variable
Data-Type-Index	0A	Octet-String
Length	02	2 Byte
Password	00	kein Paßwort
Access-Groups	00	keine Zugriffsgruppen
Access-Rights	0001	Read-All
Local-Address	xxxx	herstellerspezifisch
Extension	-	nicht vorhanden

**7. Datenstrukturen**

In diesem Kapitel sind die Datenstrukturen aller Anwender-Daten zusammengefaßt.

Die Parameter eines Sensors oder Aktors werden in einem Objektverzeichnis hinterlegt. Dieses Objektverzeichnis dient zur Beschreibung der Parameter. Es enthält Angaben zum Index, Datentyp, Objekttyp, zu den Zugriffsrechten usw.. Der Index dient zur Adressierung des Parameters beim Schreiben und Lesen. Dieses Objektverzeichnis kann mit der Funktion 'Kommunikationsobjekt-Liste-Lesen' ausgelesen werden.

**Aufbau des Objektverzeichnisses**

**Tabelle 23: Aufbau des Objektverzeichnisses**

Index	Objektverzeichnis
0000	OV-Objektbeschreibung
0001 001F	(DIN 19245/Teil 2 ) Statisches Typverzeichnis
0020 003F	(Profile) Statisches Typverzeichnis
0040 005F	(frei für Hersteller) Statisches Typverzeichnis
...	...
2000 5FFF	( frei für Hersteller ) Statisches Objektverzeichnis
6000 603F	( Geräte entsprechend Sensor/Aktor-Profil ) Statisches Objektverzeichnis
6040 9FFF	( Profile ) Statisches Objektverzeichnis
A000 BFFF	( frei für Hersteller ) Dynamisches Variablenlisten-Verzeichnis
C000 DFFF	( Profile ) Dynamisches Variablenlisten-Verzeichnis
E000 EFFF	( frei für Hersteller ) Dynamisches Program-Invocation-Verzeichnis
F000 FFFF	( Profile ) Dynamisches Program-Invocation-Verzeichnis

Falls ein Gerät Funktionen unterstützt, die in weiteren Profilversionen oder anderen Profilen definiert sind, erscheinen in den Profil-Bereichen die entsprechenden Objekte.

**Objektbeschreibung der Null-Objekte**

Diese Objektbeschreibung wird unter den Indices hinterlegt, für die kein Objekt vorgesehen ist (z.B. nicht unterstützte optionale Objekte).

**Abbildung der Gerätefunktion auf die Kommunikation**

Objektbeschreibung: 'Null-Objekt'

**Tabelle 24: Objektbeschreibung 'Null-Objekt'**

Objektattribut	Wert hex	Bedeutung
Index	xxxx	Null-Objekt
Object-Code	00	Null-Objekt

Die Liste aller - über die Kommunikation - ansprechbaren Parameter.

**Tabelle 25: Liste der über die Kommunikation ansprechbaren Parameter**

Index	Typ	Objekt	Parameter-Name	m/o
6000	PDB-Struk	Record	PE-Daten-Beschreibung	o
6001	PDB-Struk	Record	PA-Daten-Beschreibung	o
6002	Octet-String	Var	PA-Daten-Freigeben	m
6003	Unsigned16	Var	PD-Überwachungszeit	o
6004	Integer16	Var	PD-Überwachung-Auswahlcode	o
6005	Unsigned16	Var	K-Überwachungszeit	o
6006	Integer16	Var	K-Überwachung-Auswahlcode	o
6007	Integer16	Var	Verbindungsabbau-Auswahlcode	o
6008	Unsigned16	Var	Kennnummer	o
6009	Unsigned8	Var	Parametersatzkennung	o
600A	Unsigned32	Var	Seriennummer	o
600B	Date	Var	Kalibrierdatum	o
600C	Visible-Str.	Var	Gerätebeschreibung	o
600D	Visible-Str.	Var	Modellbeschreibung	o
600E	Date	Var	Parametrierungsdatum	o
600F	Octet-String	Array	Funktionsgruppen-Beschreibung	m
6010	Octet-String	Var	Prozeßeingangsdaten	m
6011	Octet-String	Var	Prozeßausgangsdaten	m
6012	Boolean	Var	Schreibsteuerung	o
6013	Unsigned 16	Array	Konflikt-Verzeichnis	o
6014		Null		
....				
603E		Null		
603F	Octet-String	Var	Störungscode	o

m = mandatory o = optional

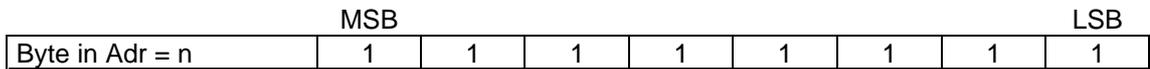
### 7.1. Datentypen

#### 7.1.1. Boolean

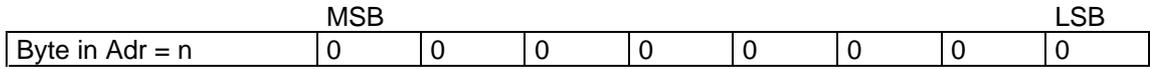
Darstellung der Werte TRUE oder FALSE in einem Octet.

Notation: Boolean  
 Wertebereich: TRUE oder FALSE  
 Codierung: FALSE wird durch den Wert 0,  
 TRUE durch den Wert FF hex repräsentiert.

#### Darstellung des Wertes TRUE am Kommunikationsinterface



#### Darstellung des Wertes FALSE am Kommunikationsinterface



#### 7.1.2. Integer

Integer-Werte sind vorzeichenbehaftete Größen.

Notation: Integer8, Integer16, Integer32

Datentyp	Wertebereich	Länge
Integer8	-128 _ i _ 127	1 Octet
Integer16	-32768 _ i _ 32767	2 Octet
	31 _ i _ 31	
Integer32	-2 <sup>31</sup> _ i _ 2 <sup>31</sup> - 1	4 Octet

Codierung: Zweierkomplement-Darstellung  
 VZ = 0: positive Zahlen einschließlich Null  
 VZ = 1: negative Zahlen

**Darstellung eines Integer8 am Kommunikationsinterface**

	MSB							LSB
Byte in Adr = n	VZ	26	25	24	23	22	21	20

**Darstellung eines Integer16 am Kommunikationsinterface**

	MSB							LSB
Byte in Adr = n	VZ	214	213	212	211	210	209	208
Byte in Adr = n+1	207	206	205	204	203	202	201	200

**Darstellung eines Integer32 am Kommunikationsinterface**

	MSB							LSB
Byte in Adr = n	VZ	230	229	228	227	226	225	224
Byte in Adr = n+1	223	222	221	220	219	218	217	216
Byte in Adr = n+2	215	214	213	212	211	210	209	208
Byte in Adr = n+3	207	206	205	204	203	202	201	200

## 7.1.3. Unsigned

Unsigned-Werte sind vorzeichenlose Größen.

Notation: Unsigned8, Unsigned16, Unsigned32

Wertebereich:	Datentyp	Wertebereich	Länge
	Unsigned8	$0 \leq i \leq 255$	1 Octet
	Unsigned16	$0 \leq i \leq 65535$	2 Octet
	Unsigned32	$0 \leq i \leq 4294967295$	4 Octet

Codierung: Binär

**Darstellung eines Unsigned8 am Kommunikationsinterface**

	MSB							LSB
Byte in Adr = n	27	26	25	24	23	22	21	20

**Darstellung eines Unsigned16 am Kommunikationsinterface**

	MSB						LSB	
Byte in Adr = n	215	214	213	212	211	210	29	28
Byte in Adr = n+1	27	26	25	24	23	22	21	20

**Darstellung eines Unsigned32 am Kommunikationsinterface**

	MSB						LSB	
Byte in Adr = n	231	230	229	228	227	226	225	224
Byte in Adr = n+1	223	222	221	220	219	218	217	216
Byte in Adr = n+2	215	214	213	212	211	210	29	28
Byte in Adr = n+3	27	26	25	24	23	22	21	20

7.1.4. Floating-Point

Gleitpunktzahl

Notation: Floating-Point (4 Octets)  
 Wertebereich: siehe IEEE Std 754 Short Real Number (32 Bits)  
 Codierung: siehe IEEE Std 754 Short Real Number (32 Bits)

	MSB						LSB		
Byte in Adr = n	VZ	27	26	25	24	23	22	21	Exponent (E)
Byte in Adr = n+1	20	2-1	2-2	2-3	2-4	2-5	2-6	2-7	
Byte in Adr = n+2	2-8	2-9	2-10	2-11	2-12	2-13	2-14	2-15	Fraction (F)
Byte in Adr = n+3	2-16	2-17	2-18	2-19	2-20	2-21	21-22	2-23	

7.1.5. Visible-String

Notation: Visible-String  
 Wertebereich: siehe ISO 646 und ISO 2375: Defining registration number 2 + SPACE  
 Codierung: siehe ISO 646

**Darstellung eines Visible-String am Kommunikationsinterface**

	MSB	LSB
Byte in Adr = n	erstes Zeichen	
Byte in Adr = n+1	zweites Zeichen	
Byte in Adr = n+2	drittes Zeichen	
Byte in Adr = n+3	viertes Zeichen	
usw.	usw.	

## 7.1.6. Octet-String

Notation: Octet-String

Binär

Codierung:

**Darstellung eines Octet-String am Kommunikationsinterface**

	MSB	LSB
Byte in Adr = n	erstes Octet	
Byte in Adr = n+1	zweites Octet	
Byte in Adr = n+2	drittes Octet	
Byte in Adr = n+3	viertes Octet	
usw.	usw.	

## 7.1.7. Date

Der Datentyp Date besteht aus einem Kalenderdatum und einer Zeitangabe.

Notation: Datum/Zeit

Wertebereich: 0 ms bis 99 Jahre

Codierung: In 7 Octets

Parameter	Wertebereich	Bedeutung des Parameters
ms	0...59 999	Millisekunden
min	0...59	Minuten
SU	0,1	0: Standard-Zeit, 1: Sommer-Zeit
RSV		Reserve
h	0...23	Stunden
Wo-tag	1...7	Wochentag: 1 =Montag, 7 =Sonntag
Mo-tag	1...31	Tag des Monats
Monat	1...12	Monat
Jahr	0...99	Jahreszahl (ohne das Jahrhundert)

**Darstellung eines Date am Kommunikationsinterface**

	MSB				LSB				
Byte in Adr = n	215	214	213	212	211	210	29	28	ms (0..59999)
Byte in Adr = n+1	27	26	25	24	23	22	21	20	
Byte in Adr = n+2	reserve		25	24	23	22	21	20	min (0..59)
Byte in Adr = n+3	SU	reserve		24	23	22	21	20	h (0..23)
Byte in Adr = n+4	Wo-Tag 22 21 20			Mo-Tag 24 23 22 21 20					Wo-Tag(1..7) Mo-Tag (1...31)
Byte in Adr = n+5	reserve		25	24	23	22	21	20	Monat (0...12)
Byte in Adr = n+6	reser- ve	26	25	24	23	22	21	20	Jahr (0...99)

7.1.8. Time-Of-Day

Der Datentyp Time-Of-Day besteht aus einer Zeitangabe und einer optionalen Datumsangabe. Die Zeit wird in Millisekunden seit Mitternacht angegeben. Um Mitternacht beginnt die Zählung mit dem Wert Null. Das Datum wird in Tagen relativ zum 1.Januar 1984 dargestellt. Am 1. Januar 1984 beginnt die Datumsangabe mit dem Wert Null.

Notation: Time-Of-Day

Wertebereich: 0 \_ i \_ ( 228 -1 ) ms

0 \_ i \_ ( 216 -1 ) Tage

Codierung: Die Zeit wird als Binärwert durch 32 Bit (4 Octet) dargestellt, wobei die ersten vier (MSB) Bits immer den Wert Null haben. Die (optionale) Datumsangabe ist als Binärwert in 16 Bit (2 Octets) codiert. Time-Of-Day ist somit ein aus 4 oder 6 Octets bestehender Octet-String.

**Darstellung eines Time-Of-Day am Kommunikationsinterface**

	MSB				LSB				
Byte in Adr = n	0	0	0	0	227	226	225	224	Anzahl der Millisekunden seit Mitternacht
Byte in Adr = n+1	223	222	221	220	219	218	217	216	
Byte in Adr = n+2	215	214	213	212	211	210	29	28	
Byte in Adr = n+3	27	26	25	24	23	22	21	20	
Byte in Adr = n+4	215	214	213	212	211	210	29	28	Anzahl der Tage seit 1.1.84 (optional)
Byte in Adr = n+5	27	26	25	24	23	22	21	20	

## 7.1.9. Time-Difference

Der Datentyp Time-Difference besteht aus einer Zeitangabe in Millisekunden und einer optionalen Tagesanzahl. Der Aufbau entspricht dem Datentyp Time-Of-Day, gibt aber im Gegensatz zu diesem eine Zeitdifferenz an.

Notation: Time-Difference

Wertebereich: 0 \_ i \_ ( 232 -1 ) ms

0 \_ i \_ ( 216 -1 ) Tage

Codierung: Die Zeit wird als Binärwert durch 32 Bit (4 Octets) dargestellt, wobei die ersten vier (MSB) Bits immer den Wert Null haben. Die (optionale) Tagesangabe ist als Binärwert in 16 Bit (2 Octets) codiert. Time-Difference ist somit ein aus 4 oder 6 Octets bestehender Octet-String.

**Darstellung eines Time-Difference am Kommunikationsinterface**

	MSB				LSB				
Byte in Adr = n	0	0	0	0	227	226	225	224	Anzahl der Millisekunden
Byte in Adr = n+1	223	222	221	220	219	218	217	216	
Byte in Adr = n+2	215	214	213	212	211	210	209	208	
Byte in Adr = n+3	27	26	25	24	23	22	21	20	
Byte in Adr = n+4	215	214	213	212	211	210	209	208	Anzahl der Tage (optional)
Byte in Adr = n+5	27	26	25	24	23	22	21	20	

## 7.1.10. Bit-String

Im nachfolgenden Bild ist das Numerierungsschema für die einzelnen Bits des Datentyps Bit-String dargestellt.

Als Länge des Bit-Strings (in Bits) sind nur ganzzahlige positive Vielfache der Zahl 8 zugelassen.

Notation: Bit-String

Wertebereich: Binär

**Darstellung eines Bit-String am Kommunikationsinterface**

	MSB				LSB			
Byte in Adr = n	0	1	2	3	4	5	6	7
Byte in Adr = n+1	8	9	10	11	12	13	14	15
Byte in Adr = n+2				usw.				
usw.				usw.				

**Abbildung der Gerätefunktion auf die Kommunikation**

Datentypen sind als Objekte im Statischen Typverzeichnis hinterlegt.

**Statisches-Typverzeichnis**

Index (hex)	Object-Code	Description	
		Symbol-Lenght	Symbol
1	5	7	Boolean
2	5	8	Integer8
3	5	9	Integer16
4	5	9	Integer32
5	5	9	Unsigned8
6	5	10	Unsigned16
7	5	10	Unsigned32
8	5	14	Floating-Point
9	5	14	Visible-String
A	5	12	Octet-String
B	5	4	Date
C	5	11	Time-Of-Day
D	5	15	Time-Difference
E	5	10	Bit-String

Für nicht implementierte Datentypen wird an der entsprechenden Stelle ein Null-Objekt angelegt.

**'Prozeßdatenbeschreibung-Struktur'**

Je nach Implementierung des Prozeßdatenkanals wird die Datentyp-Strukturbeschreibung 'Prozeßdatenbeschreibung-Struktur' angelegt. Die Datentyp-Strukturbeschreibung enthält grundsätzlich die ersten beiden Elemente - und - dann für jedes Byte des Prozeßdatenkanals jeweils 2 Elemente für Index und Subindex für jeweils Eingangs- oder Ausgangsdaten.

Subindex	Bedeutung	Datentyp
1	Prozeßdaten-Länge	Unsigned8
2	1-Index-Prozeß-Ein/Ausgangsdaten	Unsigned16
3	1-Subindex-Prozeß-Ein/Ausgangsdaten	Unsigned8
...		
52	1-Index-Prozeß-Ein/Ausgangsdaten	Unsigned16
53	1-Subindex-Prozeß-Ein/Ausgangsdaten	Unsigned8

## Abbildung der Gerätefunktion auf die Kommunikation

Objektbeschreibung: 'Prozeßdatenbeschreibung-Struktur'

**Tabelle 26: Objektbeschreibung 'Prozeßdatenbeschreibung-Struktur'**

Objektattribut	Wert	Bedeutung
Index	0020	Prozeßdatenbeschreibung-Struktur
Object-Code	06	Datentyp-Strukturbeschreibung
Number-Of-Elements	xx	n Byte (je nach Impl. der Busanschaltung, herstellerspezifisch)
Data-Type-Index	0005	Unsigned8
Length	1	1 Byte
Data-Type-Index	0006	Unsigned16
Length	2	2 Byte
Data-Type-Index	0005	Unsigned8
Length	1	1 Byte
...	...	...
Data-Type-Index	0006	Unsigned16
Length	2	2 Byte
Data-Type-Index	0005	Unsigned8
Length	1	1 Byte

## 7.2. Anwendungsdaten

In diesem Kapitel wird die Struktur von Anwendungsdaten des Kommunikationsteilnehmers beschrieben:

- Parameterbeschreibungsdaten;
- Parameterbeschreibungsdaten der Geräteparameter;
- Grenzwerte der Parameterbeschreibungsdaten;
- Defaulteinstellung der Parameterbeschreibungsdate.

### 7.2.1. Parameterbeschreibungsdaten

Die Parameterbeschreibungsdaten beschreiben einen Geräteparameter, der in einer Gerätefunktion eine definierte Geräteaktion auslöst.

Die Parameterbeschreibungsdaten setzen sich aus folgenden Daten zusammen:

- Modultyp;
- Größenindex;
- Einheitenindex;
- Offset ( Meßbereichsverschiebung );
- Auflösung ( Meßbereich / Zahlenbereich ;
- Min- / Max-Werte.

#### Modultyp

Dieser Parameter macht eine Aussage darüber, wie z.B. der Parameterwert codiert ist.

#### Größenindex

Der Größenindex gibt codiert die physikalische Größe des Meßwertes, Meßbereichs und Meßbereichsanfangs an. Der Wertebereich ist 0 bis 256. Die Codierung ist der Tabelle zu entnehmen.

### Einheitenindex

Der Einheitenindex gibt die Maßeinheit des Meßwertes, Meßbereichs und Meßbereichsanfangs an. Der Einheitenindex hat den Wertebereich - 127 bis 128. Der Wert des Einheitenindex entspricht dabei der jeweiligen Zehnerpotenz der Standardeinheit, also

Einheitenindex 0 für  $10^0$

Einheitenindex 3 für  $10^3$

Einheitenindex -3 für  $10^{-3}$

usw.

Die aufgelisteten Einheitenindizes für SI-kompatible Einheiten (Einheitenindex < 64) sollen nur als Beispiel dienen. Die Einheitenindizes für weitere SI-kompatible Vorsätze (Pico usw.) ergeben sich analog. Einheitenindizes die größer als +64 sind, haben eine Sonderbedeutung, die aus der Tabelle ermittelt werden müssen. Bei diesen Einheiten handelt es sich z.B. um die Einheiten Tag, Stunde, Minute oder um nicht SI-kompatible Einheiten, wie Fahrenheit.

### Offset (Meßbereichsverschiebung)

Dieser Subparameter gibt die Meßbereichsverschiebung bezogen auf den Nullpunkt an. Der Wertebereich ist Integer16.

### Auflösung

Dieser Subparameter gibt die Auflösung des Meßwertes an. Der Parameter berechnet sich nach der Formel:

$$A = \frac{MB}{Z}$$

Dabei ist:

A die Auflösung;

MB der Meßbereich;

Z der Zahlenbereich.

Der physikalische Meßwert ergibt sich nach folgender Formel:

$$PM = M \times A$$

Dabei ist:

PM der physikalische Meßwert;

M der Meßwert;

A die Auflösung.

Beispiel 1:

$$A = \frac{16 \text{ V}}{65\,565}$$

$$PM = 32\,782 \times \frac{16 \text{ V}}{65\,565} = 8 \text{ V}$$

Beispiel 2:

$$A = \frac{100 \text{ V}}{10\,000}$$

$$PM = 1\,000 \times \frac{100 \text{ V}}{10\,000} = 10 \text{ V}$$

### Min- / Max-Werte

Diese beiden Parameter geben den maximalen und minimalen Wert an.

Subparameter	Initialisierungsdaten	Bedeutung
Modultyp	12 Bit	12 Bit Wandler
Größenindex	22	Ampere
Einheitenindex	-3	10-3
Meßbereichsverschiebung	4	4 mA
Auflösung	16 / 65565	4-20 mA (16 mA)

Tabelle 27: Größenindex und Einheitenindex

Physikalische Größe	Größenindex	Einheit	Einheitenindex
	0	dimensionslos	0
Länge	1	Meter	0
		Millimeter	-3
		Kilometer	3
		Mikrometer	-6
Fläche	2	Quadratmeter	0
		Quadratmillimeter	-6
		Quadratkilometer	6
Volumen	3	Kubikmeter	0
		Liter	-3
		Milliliter	-6
Zeit	4	Sekunde	0
		Minute	70
		Stunde	74
		Tag	77
		Halbwellen	78
		Millisekunde	-3
		Mikrosekunde	-6
Kraft	??????	Newton	0
		Kilonewton	-3
Wirkleistung	9	Watt	0
		Kilowatt	3
		Megawatt	6
		Milliwatt	-3
Scheinleistung	10	Voltampere	0
		Kilovoltampere	3
		Megavoltampere	6
Drehzahl	11	1/Sekunde	0
		1/Minute	73
		1/Stunde	74
Winkel	12	Radian	0
		Sekunde	75
		Minute	76
		(Alt-)Grad	77
		Neugrad	78
Geschwindigkeit	13	Meter/Sekunde	0
		Millimeter/Sek.	-3
		Millimeter/Min.	79
		Meter/Min.	80
		Kilometer/Min.	81
		Millimeter/Std.	82
		Meter/Std.	83
		Kilometer/Std.	84

fortgesetzt

**Tabelle 27: Größenindex und Einheitenindex (abgeschlossen)**

Physikalische Größe	Größenindex	Einheit	Einheitenindex
Drehmoment	16	Newtonmeter	0
		Kilonewtonmeter	3
		Meganewtonmeter	6
Temperatur	17	Kelvin	0
		Grad Celsius	94
		Grad Fahrenheit	95
Elektrische Spannung	21	Volt	0
		Kilovolt	3
		Millivolt	-3
		10 Millivolt	-2
		Mikrovolt	-6
Elektrischer Strom	22	Ampere	0
		Milliampere	-3
		Kiloampere	3
		10 Ampere	1
		Mikroampere	-6
Verhältnis	24	Prozent	0
		0.01 Prozent	-2
Frequenz	28	Hertz	0
		Kilohertz	3
		Megahertz	6
		Gigahertz	9
Schritte	32	Schritte	0
		0.01 Schritte	-2
Encoderauflösung	33	Schritte/Umdrehung	0
Durchfluß	34	Kubikmeter/Sekunde	0
		Liter/Sekunde	-3
		Milliliter/Sekunde	-6
		Kubikmeter/Minute	85
		Liter/Minute	86
		Milliliter/Minute	87
Beschleunigung	35	1/Sekunde <sup>2</sup>	0

#### 7.2.1.1. Parameterbeschreibungsdaten der Geräteparameter

Die Parameterbeschreibungsdaten entsprechen der vorgegebenen Struktur im INTERBUS Sensor/Aktor Profil. Die Parameter einer Geräte-Funktion werden mit einer Parameterbeschreibung beschrieben. Die Parameter und die Parameterbeschreibung werden als Geräteparameter in dem Gerät hinterlegt.

Die Zuordnung der Parameterbeschreibungsdaten zu den Geräteparametern wird folgendermaßen dargestellt:

Parameter	Geräteparameter
Parameterbeschreibung	Geräteparameter
"	"
"	"

### 7.2.2. Grenzwerte der Parameterbeschreibungsdaten

Die Grenzwerte der Parameterbeschreibungsdaten geben an, auf welche Werte ein Einstellparameter eingestellt werden kann. Diese Daten müssen im Datenblatt angegeben werden.

### 7.2.3. Defaulteinstellung der Parameterbeschreibungsdaten

Die Defaulteinstellung sind die Parameterbeschreibungsdaten, auf die der Geber bei der Auslieferung eingestellt ist. Diese Daten müssen im Datenblatt angegeben sein:

- Modultyp;
- Größenindex;
- Einheitenindex;
- Offset ( Meßbereichsverschiebung );
- Auflösung ( Meßbereich / Zahlenbereich );
- Min- / Max-Werte.

### 7.2.4. Auswahlcode

Mit dem Auswahlcode wird eine Funktion aus mehreren profile- oder herstellerspezifischen Funktionen ausgewählt. Der Code für die profilespezifischen Funktionen ist ein Wert im Bereich von 0 bis 32767. Der Code für die herstellerspezifischen Funktionen ist ein Wert im Bereich von -32768 bis -1. Die Auswahl von nicht definierten Funktionen wird neg. quittiert.

Auswahlcode	Bedeutung der Auswahl-Funktion
-32768 ... -1	herstellerspezifisch
0 ... 32767	profilespezifisch

## 8. Betriebsphasen der Anwendung

In diesem Kapitel werden die möglichen Betriebsphasen des Gerätes beschrieben. Das Kapitel ist folgendermaßen aufgeteilt:

- Anlauf/Abbruch;
- Betrieb;
- Inbetriebnahmephase und Projektierungsphase.

## 8.1. Anlauf/Abbruch

### Anlauf

Nach Spannungseinschalten oder Zurücksetzen des Gerätes beginnt der Anlauf. Folgende Aktionen werden durchgeführt:

- Initialisierung der Kommunikationsschnittstelle;
- Initialisierung der Prozeßdaten;
- Initialisierung der Parameter.

### Initialisierung der Kommunikationsschnittstelle

Die Kommunikationsbeziehung wird mit folgenden Werten parametrier:

PMS-KBL-Parameter	Wert
PMS-Service-Supported	Read.ind/res,Write.ind/res ( alle anderen Services optional )
Max-Outstanding-Services	1
Max-PDU-Sending-High-Prio	0
Max-PDU-Sending-Low-Prio	herstellerspezifisch
Max-PDU-Receiving-High-Prio	0
Max-PDU-Receiving-Low-Prio	herstellerspezifisch

### Initialisierung der Prozeßdaten

Die Prozeß-Ein- und -Ausgangsdatenregister werden mit Null vorbesetzt.

### Initialisierung der Parameter

Folgende Kommunikationsobjekte müssen mit den entsprechenden gespeicherten Werten oder - falls nicht vorhanden - Ersatzwerten im Anlauf parametrier werden.

Kommunikations-Objekt	Ersatzwert
Prozeßdaten-Überwachungszeit	65535 ausgeschaltet
Prozeßdaten-Überwachungs-Auswahlcode	0 keine Reaktion
Kommunikations-Überwachungszeit	65535 ausgeschaltet
Kommunikations-Überwachungs-Auswahlcode	0 keine Reaktion
Verbindungsabbau-Auswahlcode	0 keine Reaktion

## **Abbruch**

Folgende Aktionen werden durchgeführt:

- Reset der Prozeßdaten

Wenn die Anwendungseinheit ausfällt und eine Entkopplung zwischen der Kommunikations- und Anwendungseinheit besteht, werden die Prozeßeingangsdaten auf Null gesetzt.

## **8.2. Inbetriebnahmephase und Projektierungsphase**

noch nicht definiert

## **9. Kommunikationsprofil**

### **9.1. Schicht 1**

In diesem Kapitel wird die Kopplung an das Übertragungsmedium definiert. In den weiterführenden Profilen werden die möglichen Kopplungen ausgewählt.

#### 9.1.1. Installationsfernbus-Schnittstelle

##### **Stecker**

CONINVERS-Stecker (IP65)

oder

9 polig. D-Sub Stecker (IP20), bis max. 1A

#### 9.1.2. Remotebus-Schnittstelle

##### **Übertragungsprotokoll**

2-Leiter-Fernbus

##### **Stecker**

D-Sub 9 polig. (male) für die „ankommende Schnittstelle“ zum vorigen Teilnehmer.

D-Sub 9 polig. (female) für die „weiterführende Schnittstelle“ zum nächsten Teilnehmer.

9.1.3. Lokalbus-Schnittstelle

**Übertragungsprotokoll**

8-Leiter-Lokalbus

**Stecker**

D-Sub 15 polig. (male) für die „ankommende Schnittstelle“ zum vorigen Teilnehmer.

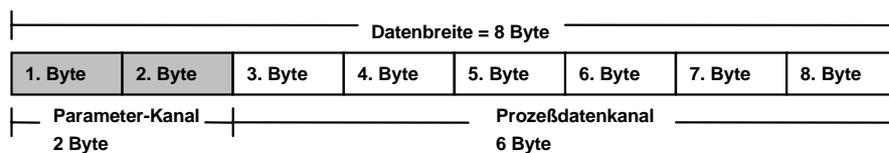
D-Sub 15 polig. (female) für die „weiterführende Schnittstelle“ zum nächsten Teilnehmer.

**9.2. Schicht 2**

In diesem Kapitel werden alle Definitionen, die die Schicht 2 betreffen, festgelegt.

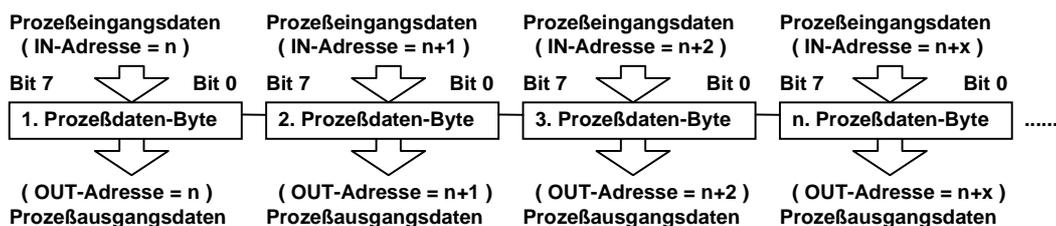
9.2.1. Konfiguration der INTERBUS-Register

Die Anordnung der Datenregister eines INTERBUS-Teilnehmers, und damit die Adressierung auf der I/O-Ebene, ist im folgenden definiert. Beispiel für einen INTERBUS Teilnehmer mit 2 Byte Parameter-Kanal:



**Bild 19: Zusammenhänge zwischen Datenbreite, Prozeßdatenkanal und Parameter-Kanal**

Die Datenbreite gibt an, wieviel Bits, der Teilnehmer im Bus belegt. Hat ein Teilnehmer z. B. 16 Bit Eingänge und 32 Bit Ausgänge, so belegt er 32 Bit (4 Bytes) im Ring (der größere Wert ist entscheidend). Die Länge des Parameter-Kanals innerhalb der Datenbreite ist im IDENT-Code festgelegt.



**Bild 20: Adressierung der Prozeßdaten**

**Prozeßdatenrichtung:**

- Prozeßeingangsdaten werden vom Gerät zum Bussystem übertragen.
- Prozeßausgangsdaten werden vom Bussystem zum Gerät übertragen.

9.2.2. Identifikation der INTERBUS Teilnehmer

Der ID-Code setzt sich folgendermaßen zusammen:

b15	b13	b12	b8	b7	b0
<b>Meldung</b>			<b>Datenbreite</b>		<b>IDENT-Code</b>

**Meldung**

Mit diesem Bit im ID-Code werden Meldungen an die Anschaltbaugruppe übertragen.

**Tabelle 28: Meldungen**

<b>b 15</b>	<b>b 14</b>	<b>b13</b>	<b>Bedeutung</b>
1	x	x	Teilnehmer-Meldung
x	1	x	CRC Fehler
x	x	1	reserviert

**Teilnehmer-Meldung**

Diese Meldung wird erzeugt, wenn der Teilnehmer erkannt hat, daß eine Peripheriestörung vorliegt. Eine Peripheriestörung kann in weiteren Profilen genauer definiert werden.

**CRC-Fehler**

Diese Meldung wird erzeugt, wenn Übertragungsstörungen erkannt wurden (vom Protokoll-Chip).

## Datenbreite

Tabelle 29: Datenbreite

Bit 12	Bit 11	Bit 10	Bit 9	Bit 8	Datenbreite
0	0	0	0	0	0
0	1	1	0	0	1 Bit
0	1	1	0	1	2 Bits
0	1	0	0	0	4 Bits
0	1	0	0	1	1 Byte
0	1	0	1	0	12 Bits
0	0	0	0	1	2 Bytes
0	1	0	1	1	3 Bytes
0	0	0	1	0	4 Bytes
0	0	0	1	1	6 Bytes
0	0	1	0	0	8 Bytes
0	0	1	0	1	10 Bytes
0	1	1	1	0	12 Bytes
0	1	1	1	1	14 Bytes
0	0	1	1	0	16 Bytes
0	0	1	1	1	18 Bytes
1	0	1	0	1	20 Bytes
1	0	1	1	0	24 Bytes
1	0	1	1	1	28 Bytes
1	0	0	1	0	32 Bytes
1	0	0	1	1	48 Bytes
1	0	0	0	1	52 Bytes
1	0	1	0	0	64 Bytes
1	0	0	0	0	reserviert
1	1	x	x	x	reserviert

x=„don't care“

**IDENT-Code**

Beschreibung der Gerätefunktion		IDENT Code (dec)	IDENT Code (hex)
<b>Buskoppler</b>			
BK mit 8-Leiter Lokabusabzweig	BK-8L-LB	52	34
BK mit 2-Leiter Lokabusabzweig	BK-2L-LB	8	08
BK mit Ein- und Ausgangsdaten und 8-Leiter Lokabus- oder 2-Leiter Fernbusabzweig	BK-I/O	11	0B
BK mit 2-Leiter Fernbusabzweig	BK-2L-RB	12	0C
<b>Remotebusteilnehmer Digital</b>			
Digital-Geräte mit Ausgangsdaten	DO	1	01
Digital-Geräte mit Eingangsdaten	DI	2	02
Digital-Geräte mit Ein- und Ausgangsdaten	DIO	3	03
Profilkonforme Digital-Geräte mit Ausgangsdaten	PROFIL DO	13	0D
Profilkonforme Digital-Geräte mit Eingangsdaten	PROFIL DI	14	0E
Profilkonforme Digital-Geräte mit Ein- und Ausgangsdaten	PROFIL DIO	47	2F
ISO-Ventilinsel	ISO-Ventilinsel	5	05
<b>Remotebusteilnehmer Analog</b>			
Analog-Geräte mit Ausgangsdaten	AO	49	31
Analog-Geräte mit Eingangsdaten	AI	50	32
Analog-Geräte mit Ein- und Ausgangsdaten	AIO	51	33
Profilkonforme Analog-Geräte mit Ausgangsdaten	PROFIL AO	53	35
Profilkonforme Analog-Geräte mit Eingangsdaten	PROFIL AI	58	3A
Profilkonforme Analog-Geräte mit Ein- und Ausgangsdaten	PROFIL AIO	59	3B
ENCOM mit Eingangsdaten	ENCOM	54	36
ENCOM mit Ein- und Ausgangsdaten	ENCOM	55	37
<b>Remotebusteilnehmer mit Parameterkanal</b>			
Geräte mit Parameterkanal (2 PCP Worte)	PA-Kanal	240	F0
Geräte mit Parameterkanal (4 PCP Worte)	PA-Kanal	241	F1
Geräte mit Parameterkanal (variabler PA-Kanal)	PA-Kanal	242	F2
Geräte mit Parameterkanal (1 PCP Wort)	PA-Kanal	243	F3
DRIVECOM (2 PCP Worte)	DRIVECOM	224	E0
DRIVECOM (4 PCP Worte)	DRIVECOM	225	E1
DRIVECOM (variabler PA-Kanal)	DRIVECOM	226	E2
DRIVECOM (1 PCP Wort)	DRIVECOM	227	E3
ENCOM (2 PCP Worte)	ENCOM	244	F4
ENCOM (4 PCP Worte)	ENCOM	245	F5
ENCOM (variabler PA-Kanal)	ENCOM	246	F6
ENCOM (1 PCP Wort)	ENCOM	247	F7
Profilkonforme Geräte (2 PCP Worte)	PROFIL PA-Kanal	228	E4
Profilkonforme Geräte (4 PCP Worte)	PROFIL PA-Kanal	229	E5
Profilkonforme Geräte (variabler PA-Kanal)	PROFIL PA-Kanal	230	E6
Profilkonforme Geräte (1 PCP Wort)	PROFIL PA-Kanal	231	E7
<b>Lokalbusteilnehmer Digital</b>			
Digital-Geräte mit Ausgangsdaten	DO	189	BD
Digital-Geräte mit Eingangsdaten	DI	190	BE
Digital-Geräte mit Ein- und Ausgangsdaten	DIO	191	BF
Profilkonforme Digital-Geräte mit Ausgangsdaten	PROFIL DO	181	B5
Profilkonforme Digital-Geräte mit Eingangsdaten	PROFIL DI	182	B6
Profilkonforme Digital-Geräte mit Ein- und Ausgangsdaten	PROFIL DIO	183	B7
Schraubersteuerungen	Schraubersteuerungen	187	BB
<b>Lokalbusteilnehmer Analog</b>			
Analog-Geräte mit Ausgangsdaten	AO	125	7D
Analog-Geräte mit Eingangsdaten	AI	126	7E
Analog-Geräte mit Ein- und Ausgangsdaten	AIO	127	7F
Profilkonforme Analog-Geräte mit Ausgangsdaten	PROFIL AO	121	79
Profilkonforme Analog-Geräte mit Eingangsdaten	PROFIL AI	122	7A
Profilkonforme Analog-Geräte mit Ein- und Ausgangsdaten	PROFIL AIO	123	7B
ENCOM mit Eingangsdaten	ENCOM	102	66
ENCOM mit Ein- und Ausgangsdaten	ENCOM	103	67
<b>Lokalbusteilnehmer mit Parameterkanal</b>			
Geräte mit Parameterkanal (2 PCP Worte)	PA-Kanal	220	DC
Geräte mit Parameterkanal (4 PCP Worte)	PA-Kanal	221	DD
Geräte mit Parameterkanal (variabler PA-Kanal)	PA-Kanal	222	DE
Geräte mit Parameterkanal (1 PCP Wort)	PA-Kanal	223	DF
DRIVECOM (2 PCP Worte)	DRIVECOM	192	C0
DRIVECOM (4 PCP Worte)	DRIVECOM	193	C1
DRIVECOM (variabler PA-Kanal)	DRIVECOM	194	C2
DRIVECOM (1 PCP Wort)	DRIVECOM	195	C3
ENCOM (2 PCP Worte)	ENCOM	212	D4
ENCOM (4 PCP Worte)	ENCOM	213	D5
ENCOM (variabler PA-Kanal)	ENCOM	214	D6
ENCOM (1 PCP Wort)	ENCOM	215	D7
Profilkonforme Geräte (2 PCP Worte)	PROFIL PA-Kanal	216	D8
Profilkonforme Geräte (4 PCP Worte)	PROFIL PA-Kanal	217	D9
Profilkonforme Geräte (variabler PA-Kanal)	PROFIL PA-Kanal	218	DA
Profilkonforme Geräte (1 PCP Wort)	PROFIL PA-Kanal	219	DB

### 9.3. Schicht 7

Folgende Dienste sind vorgeschrieben:

<b>Service</b>	<b>m/o</b>
Initiate	m
Abort	m
Reject	m
Identify	m
Status	m
Get OV	m
Get OV Long	o
Read	o
Write	o