

Basisprofil

10

V3.0

2018-04-19

09

Profil: Basis

Profil-Nummer: 10

Version: V3.0

Versionszähler: 09

Datum: 19. April 2018

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

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 technischem 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

Revisionsverzeichnis

Index	Datum	Änderungen, Ergänzungen oder Erläuterungen
1.00	14.10.2002	Erste Version
1.01	12.08.2004	<ul style="list-style-type: none"> • Ergänzung Kanalnummer im Diagnoseobjekt 0x0018 und folgende • Ergänzung der Prioritäten 81, 82, 83 „Störung ist beseitigt“ Diagnoseobjekt 0x0018 und folgende • Überarbeitung/Ergänzung der Störungscodes • Erweiterung der Handhabungsbeschreibung von Diagnosemeldungen • Objekt 0x000F DeviceProfile mandatory • Vorspann und Anhang eingefügt • Redaktionelle Überarbeitung • Einführung der Objekte <ul style="list-style-type: none"> -0x002E – CheckSum -0x002F - PDOOUT_Subst -0x0030 - PF_Code -0x0031 - PDIN_Subst -0x0032 - IBS_ID • Definition von Werten für die Objekte <ul style="list-style-type: none"> -0x0024 – IBSResetCode -0x0020 – PDTimeoutCode
1.02	28.01.2005	<ul style="list-style-type: none"> • Fehlerbereinigung (Re/Gr/Formate/etc.) • Objekt 0x0019 „ResetDiag“ Name und Bedeutung angepasst • Objekt 0x000F „DeviceProfile“ Typ geändert • Störungscodes erweitert: 2344; 2345; 341X; 342X; 5230
1.10	20.06.06	<ul style="list-style-type: none"> • Objekt 0x0018 „DiagState.Kanal“ in „DiagState.Channel/Group“ geändert, Bedeutung angepasst • Objekt 0x0018 „DiagState. MoreFollows“ Bedeutung angepasst • Objekt 0x001A „GetErrorRepMethod“, Name und Beschreibung erweitert • Objekt 0x001D „Password“ angepasst • Objekt 0x0025 „PDIN“ zusätzlicher Erläuterung • Objekt 0x0026 „PDOOUT“ zusätzlicher Erläuterung • Objekt 0x002E „Checksum“, Typ geändert UINT16 => UNIT32 • Objekt 0x0033 „DiagStateChannelNumber“ neu eingefügt • Objekt 0x0034 „DiagStateAddValue“ neu eingefügt • Objekt 0x0035 „NoOfModules“ neu eingefügt • Objekt 0x0036 „SubBusStructure“ neu eingefügt • Objekt 0x0037 „DeviceType“ neu eingefügt • Objekt 0x0038 „ObjDescrReq“ neu eingefügt • Objekt 0x0039 „ObjDescr“ neu eingefügt • Objekt 0x003A „VersionCount“ neu eingefügt • Objekt 0xE805 „ObjDescrLong“ neu eingefügt • Objekt 0xE800 „DiagStateLong“ teilweise als Verweis beschrieben und doppelte Beschreibungen zu vermeiden • Modulare Geräte sind beschrieben. Als Moduladressierung wird die Invoke-ID, jetzt „Modulnummer“, genutzt • Liste der zulässigen Kommunikations-Error Types überarbeitet und ergänzt • Datentyp „Bit-String“ eingeführt. • Kapitel „Kommunikationsobjekte“ überarbeitet, neu strukturiert und ergänzt • Kapitel Objektbeschreibung eingefügt • Einführung Error Class 8 Error Code 1 profilspezifisch

1.11	06.10.2009	<ul style="list-style-type: none"> • Additional Error Code überarbeitet • Handhabung Objekt 0x0038 „ObjDescrReq“ Objekt 0x0039 „ObjDescr“ erweitert • Error Class “8” Other, Error Code “1” Profile specific durchgängig korrigiert • Datentyp Visible String (immer “0x00” terminiert!) • Objekt 0x0037 „DeviceType“ leicht überarbeitet • Objekt 0x0008 „ProductID“ um Missverständnisse zu vermeiden in „SerialNo“ umbenannt • StörungsCodes 6800 und 6810 eingefügt A000 ff erweitert und ausführlicher erläutert • Erläuterung des Objektraums für modulare Geräte • Objekt 0x003B „PDIN_Descr“ neu eingefügt • Objekt 0x003C „PDOOUT_Descr“ neu eingefügt • Objekt 0x0019 „ResetDiag“ erweitert und konkretisiert
1.12	19.11.2010	<ul style="list-style-type: none"> • Device Family um digital/analog IN/OUT ergänzt • Liste der Datentypen vervollständigt • LanguageCode korrigiert • Länge bei Visible String korrigiert • Objekt 0x003B „PDIN_Descr“ überarbeitet, neue Typen eingefügt • Objekt 0x003C „PDOOUT_Descr“ überarbeitet, neue Typen eingefügt • Objekt 0x000C „FirmwareVersion“ und 0x000B „PChVersion“ konkretisiert um Einträge wenn keine FW/PChVersion vorhanden ist • Handhabung der Prioritäten im DiagnoseObjekt 0x0018 DiagState erläutert • Objekt 0x0019 „ResetDiag“ erweitert und konkretisiert • Objekt 0x0018.5 „ResetDiag.MoreFollows“ konkretisiert • Additional Error Codes ergänzt • Objekt 0x0037 „DeviceType“ Fehler korrigiert R/W => R, kann niemals R/W gewesen sein • Subindex „Access Rights“, „DisplayFormat“, „Resolution“ und „Offset“ in Objekt 0x0039 „ObjDescr“ eingefügt, Beschreibung ergänzt. • Größenindex für „Kraft“ und „el. Widerstand“ ergänzt • Objekt 0x0020, 0x0024, 0x0030 UINT16 => Array of UINT16 geändert und kommentiert •
2.0	07.12.2011	<ul style="list-style-type: none"> • Basisprofil „neutralisiert“ D.h. den Bezug zu INTERBUS und PCP aufgelöst. • Kommentare zu den Versions-Countern ergänzt • Objekt 0x003D „WakeUpTime“ eingeführt • ResetCode ausführlicher erläutert • Objekt 0x002D ResetParam Code “02” eingeführt • Objekt 0x0039 „ObjDescr“ inkompatibel überarbeitet (Min und Max verschoben) • Übersetzungstabelle für die Objektnamen im Anhang hinzugefügt • Beschreibung der Download-Services (Download-Write, Upload-Read) ergänzt und konkretisiert. • Objekt 0x0011 Fehler in der Stringlänge (10+1) korrigiert • Objekt 0x0005 Capabilities eingeführt • Objekt 0x0038 “ObjDescrReq” Länge korrigiert

3.0	19.04.2018	<ul style="list-style-type: none"> • Neue Additional Error Code „8.2 Hardware ist temporär gestört“ hinzugefügt • Beschreibung „Eindeutige, fortgezählte Störungsnummer“ konkretisiert • Störungscode überarbeitet und Erklärungen verbessert • Diverse neue Störungscode festgelegt u.a.A014, A025, 6340, 8F00 • Bedeutung Warnung/Störung (Alarm) konkretisiert • 0x0005 „Capabilities“ „Safety0“ „ChPDWh_0“ „SBM_0“ ergänzt • Beseitigung diverser Rechtschreib- und sprachlicher Fehler • Darstellung „N+1“ durchgängig genutzt • Schreibweise hex; 0x vereinheitlicht. • 3150 Versorgungsspannung ist verpolt • Objekt 0x0004 „DeviceFamily“ um „Safety“ ergänzt • Objekt 0x0005 „Capabilities“ Datentyp Visible String => Octet String geändert • Objekt 0x0013 „DeviceDescFile“ umbenannt in „OnBoardDeviceDescFileName“ und Beschreibung konkretisiert. • Objekt 0x0018 komplett überarbeitet • Objekt 0x0019 „ResetDiag“ Rechte geändert: W=R/W • Objekt 0x0019 „ResetDiag“ Bedeutung konkretisiert und ergänzt • Objekt 0x001A „GetErrorRepMethod“ Beschreibung Bit3 gelöscht • Objekt 0x0020 „PDTimeoutCode“ Werte hinzugefügt • Objekt 0x0024 „ResetCode“ Werte hinzugefügt • Objekt 0x002A ConflictDictionary inkompatibel geändert • Objekt 0x002D „ResetRaram“ Rechte jetzt R/W und wird mandatory , Wert 0x03 hinzugefügt • Begriffsdefinition „Ersatzwerte“ erneuert • Objekt 0x0030 „PF_Code“ Werte hinzugefügt • Objekt 0x0033 zurück gezogen • Objekt 0x0034 zurück gezogen • Objekt 0x0035 NoOfModules => Sub-BusInfo inkl. Bedeutung geändert • Objekt 0x0036 ActSubBusStructure überarbeitet • Objekt 0x0039 auf R/W_D gesetzt. Funktion erläutert. • Objekt 0x0039.10 ObjectDescr.RNR (R 2 Byte) in UINT16(Dez) geändert • Objekt 0x0039.12 ObjDescr. Access Rights Bit0-Bit3 wegen entsprechender Implementierung von Aktive = 1 auf „Aktive = 0“ geändert • Objekt 0x0039.12 ObjDescr. Access Rights Bit4 „Avoid Presentation“ ergänzt • Objekt 0x0039.13 ObjDescr.DiplayFormat: 0x08 „date“ ergänzt • Objekt 0x0039.16 ObjDescr.Symbol: Beschreibung von Bit-Strings ergänzt • Objekt 0x003A Beschreibung VersionCount ergänzt und konkretisiert • Objekte 0x003B PDIN_Descr und 0x003C PDOUT_Descr Typ „SFCH“, „Safel“, „SafeO_F“ - Channel for safe Data neu eingeführt, mandatory und Beschreibung ergänzt • Objekt 0x003D WakeUpTime Auflösung ergänzt • Objekt 0x003D „WakeUpTime“ Beschreibung konkretisiert, Legacy Information eingefügt • Objekt 0x003E „EnergyMgmt“ neu eingeführt • Objekt 0x0040 „ListOfObjToRestore“ neu eingeführt • Objekt 0x0041 „RefSubBusStructure neu eingeführt • Objekt 0x0042 „DevicesStatus neu eingeführt • Objekt 0x0043 „Sub-BusBehaviour“ neu eingeführt • Objekt 0x0044 „ControlSub-Bus“ neu eingeführt • Objekt 0x0045 „InitFWDownload“ neu eingeführt • Objekt 0x0047 „AddInfo“ neu eingeführt • Objekt 0x0047.01 „AddInfo.SafetyProtType“ neu eingeführt • Objekt 0x0047.02 „AddInfo.SafetyProtVers“ neu eingeführt • Objekt 0x0047.03 „AddInfo.LegacyInfo“ neu eingeführt
-----	------------	--

	<ul style="list-style-type: none"> • Objekt 0xE800 „DiagStateLong“: an geändertes Objekt 0x0018 angepasst. • Objekt 0xE809 „BackUpDataCompr“ eingeführt • Objekt 0xE807 „DeviceFW“ eingeführt • Objekt 0xE808 „OnBoardDeviceDescFile“ eingeführt • Objekt 0xE806 „ComplDiagState“ neu eingeführt • Kapitel „Benachrichtigung über ... des DiagState Objekts“ redaktionell überarbeitet • Kapitel „Konfliktverzeichnis (ConflictDictionary)“ redaktionell überarbeitet • Definition „Service-Parameter Error Type“ ergänzt • Übersicht über die verwendeten Datenobjekte eingefügt • Kapitel „Auswertung bei komplexen Datenobjekten“ eingefügt • Darstellungsform für alle Objekte ergänzt • Nachrichten, Informationen, Mitteilungen (nur gehend Prio 83) • Liste Device Family 0004 um Temperaturmodule ergänzt • Kapitel 8.2.11. „Erläuterung zur Handhabung des Diagnoseobjekts 0x0018 „ eingefügt. • Funktion „Busy“ eingeführt • Error Code (8) „B für Busy“ eingeführt • Beschreibung „Object Record“ und „Object Array“ geändert und Fehlerfall beschrieben • Additional Codes ergänzt (0x005x) Error Class 0x08, Error Code 0x01 • Additional Codes ergänzt (0x001D, 0x001E) • Generatorpolynome für Upload Read und Download Write definiert • Kapitel „Benachrichtigung über Auslesen (Read) des DiagState Objekts“ überarbeitet (Meldung über Bit in PD entfällt) • Definition Boolean ergänzt: „True = 0xFF, False = 0x00“ • R/WD eingeführt • Reservierte Bits/Bytes = „0“ festgelegt • „Safety-“ als Vorsatz für Gerätefamilie eingeführt • Kapitel „Hinweise zu Handhabung einzelner Parameter durch den Master“ eingeführt • Kapitel „Blockparametrierung“ konkretisiert. • Kapitel „Adressierung“ eingefügt • Kapitel „Ersatzwertverhalten/Einschaltverhalten“ eingefügt • Kapitel „Nicht flüchtige und flüchtige Parameter“ eingefügt • Kapitel „Firmware Download“ eingefügt • Kapitel „Funktionale Sicherheit – Safety“ eingefügt • Begriff „Terminal“ eingeführt • Beschreibung der Domain-Variable ergänzt • Makrodienste und Domain-Variablen/Var-Listen in Beziehung zueinander gesetzt • Länge und Objekttyp in den Tabellen vereinheitlicht • Kommunikationsprofil „636“ und „637“ ergänzt • Error Types statt Kommunikations Fehlercodes • Bei Datentyp und DatenCode 0x00 „undefined“ ergänzt • Kapitel „PDU-Länge“ eingefügt • Darstellung Additional Codes überarbeitet • Dependency Info: Erläuterung der Abhängigkeiten eingeführt • Kapitel „Parametersatzkennung“ überarbeitet
--	--

	<ul style="list-style-type: none">• 0x001E um Sub-Modul und Subindex erweitert• doppelte Beschreibung im Objekt 0x002B ParamSet entfernt• 0x002D ResetParam Reset to Factory Defaults Legacy Reset überarbeitet• 0x003E.01 EnergyMgmt.ActualMode Bereiche reserviert• 0x0039.0C „ObjDescr.Access Rights“ überarbeitet• 0x0039.07 „ObjDescr.UnitCode“ ergänzt und überarbeitet• 0x0040 „ListOfObjToRestore“ um Sub-Modul-Nr. ergänzt• 0x0048 => 0xE806 => 0xE809 Nummern getauscht• 0xE809 BackUpDataCompr Beschreibung konkretisiert• 0xC000 – 0xC07F „ProjBasProf“ eingeführt• Capabilities wieder zurück auf Visible String geändert und einige IDs gekürzt• Alignment der Objekte 0x002A, 0x0035 auf 16 Bit angepasst• Invoke-ID gelöscht• Re/Gr Wortformen korrigiert• CurrDiagState => ComplDiagState umbenannt• Basis-Profil => Basisprofil• intDeviceDescFile => OnBoardDeviceDescFile• intDeviceDescFileName => OnBoardDeviceDescFileName• Projektion des Basisprofils in die Sub-Bus-Module Adressierung geändert• 0x0035.06 "SubBusInfo.StartAddressPBP" => "SubBusInfo.SelectSubBusModule" gelöscht• 0x002A.01 ConfGrNo Konfliktgruppennummer eingefügt• Additinal Code 0x0024: „Index nicht vorhanden“ eingefügt
--	--

Inhalt

1	Vorwort	13
2	Anwendung und Geräteeigenschaften	14
2.1	Allgemeines	14
3	Allgemeines	15
4	Übersicht.....	15
4.1	Indexbereiche der Kommunikationsobjekte	15
4.2	Übersicht über die Standardobjekte	16
5	Kommunikationsobjekte	18
5.1	Datentypen	18
5.2	Datenobjekte.....	18
5.2.1	Objekt „Domain-Variable“	19
5.2.1.1	Domain-Variable - formale Beschreibung	19
5.2.2	Objekt „Simple-Variable“	20
5.2.2.1	Simple-Variable - formale Beschreibung.....	20
5.2.3	Objekt „Array“	20
5.2.3.1	Array-Variable - formale Beschreibung	20
5.2.4	Objekt „Record“	21
5.2.4.1	Record-Variable - formale Beschreibung	21
5.2.5	Objekt „Variablenliste“	22
5.2.5.1	Variablenliste - formale Beschreibung.....	22
5.2.5.2	Statische Variablenliste	22
5.2.5.3	Dynamische Variablenliste	23
5.2.5.4	Übertragungsformat Variablenlisten	23
5.2.6	Objekt „String-Variable“	24
5.2.6.1	String-Variable - formale Beschreibung	24
6	Services	25
6.1	Adressierung.....	26
6.1.1	Adressierung über Subindex gleich „0“	26
6.1.2	Adressierung über Subindex ungleich „0“	26
6.2	Datenlänge	27
6.3	Busy.....	27
6.4	Makrodienste	28
6.4.1	Download-Funktionalität mit Write-Service (Download-Write)	29
6.4.2	Upload-Funktionalität mit Read-Service (Upload-Read)	32
6.4.3	Beispiel Download Variablenliste	35
6.5	Liste der zulässigen Error Types der Kommunikation.....	36
6.5.1.1	Error Class und Error Code	36
6.5.2	Additional Code	38
7	Standardobjekte	40
7.1	Identifizierung	42
7.1.1	Gerätefamilie (DeviceFamily Index 0x0004)	48
7.1.2	Kommunikationsprofil (CommProfile Index 0x000E).....	49
7.1.3	Geräteprofil (DeviceProfile Index 0x000F)	50
7.2	Gerätediagnose	51
7.2.1	Objekte	51
7.2.1.1	Timing von Diagnosemeldungen	57
7.2.1.2	Benachrichtigung über Auslesen (Read) des Objekts DiagState.....	58
7.2.1.3	Benachrichtigung über Melden (Info Report) des Objekts „DiagState“	58
7.2.1.4	Klassifizierung von Störungen (Priorität der Meldungen).....	59
7.2.1.5	Störungscodes.....	60
7.2.2	Trace-Daten.....	67
7.3	Anwenderdatenmanagement	68
7.3.1	Prozessdatenmanagement.....	68
7.3.2	Ersatzwertverhalten/Einschaltverhalten	75
7.3.2.1	Einschaltverhalten (PowerOn behaviour):.....	75
7.3.2.2	Ersatzwertverhalten (FailSafe behaviour):	75
7.3.4	Funktionale Sicherheit – Safety.....	76
7.3.5	Parameterkanalmanagement	77
7.4	Gerätemanagement.....	78

7.4.1	Anwendungsfall Gerätetausch.....	81
7.4.2	Voneinander abhängige Parameter.....	82
7.4.2.1	Blockparametrierung	82
7.4.3	Parametersatzkennung	84
7.4.4	Flüchtige und nicht flüchtige Parameter	85
7.4.5	Datensicherung.....	85
7.4.6	Firmware-Update	86
7.4.7	Passwortschutz.....	90
7.4.8	Energiemanagement	92
7.5	Mehrsprachigkeit	94
7.6	Modulare Geräte - Subsysteme.....	96
7.6.1	Grundlagen	96
7.6.2	Parameter	96
7.6.3	Diagnose.....	97
7.7	Objektbeschreibung.....	102
7.8	OnBoard Gerätebeschreibung.....	109
8	Handhabung einzelner Parameter durch den Master	110
8.1	0x002D - ResetParam	110
8.2	Behandlung der Error Types	110
8.3	Erweiterungen – Zugriff über Subindex.....	110
9	Anhang A	113
9.1	Begriffsdefinitionen	113
9.2	Symbole und Abkürzungen	115
10	Anhang B.....	117
10.1	Übersetzungstabelle für die Objektnamen	117

1 Vorwort

Im Rahmen der Fabrikautomatisierung werden in der industriellen Sensorik und Aktorik immer mehr leistungsfähigere und flexiblere Systeme benötigt. Intelligente Feldgeräte 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 Feldgerätefunktionen zu standardisieren und in diesem Profil zusammenzufassen. Um die Feldgerätefunktionen unabhängig vom Kommunikationsmedium definieren zu können, wurde eine international anerkannte und standardisierte Anwenderschnittstelle IEC 61158 zur Kommunikation verwendet. Damit wurde eine Durchgängigkeit zu MMS (Manufacturing Message Specification ISO / IEC 9506) geschaffen.

Als Übertragungsmedium wurde kein spezielles Feldbussystem ausgewählt. Es muss lediglich die Anforderungen der Feldkommunikation bezüglich Echtzeitverhalten und standardisierter Anwenderschnittstelle erfüllen.

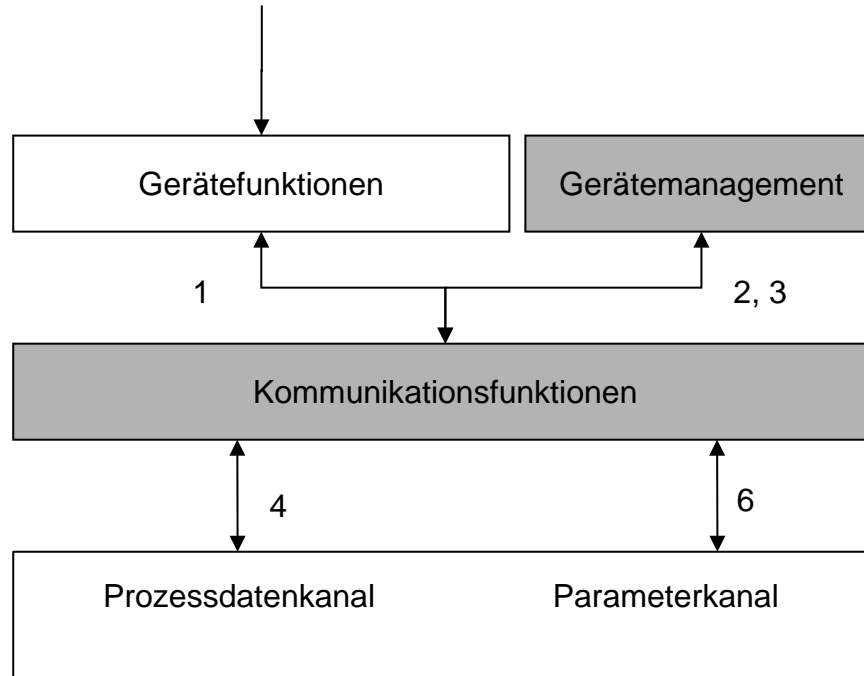
Das Basisprofil richtet sich an alle Nutzer und Gerätehersteller von Feldgeräten, die an einem Feldbus betrieben werden sollen. Diese Profil-Definition ist für den Anwender eine sinnvolle Ergänzung zur standardisierten Kommunikation und bringt eine allgemeingültige Absprache über Dateninhalte und Geräteverhalten. Diese Funktionsfestlegungen vereinheitlichen wesentliche Feldgeräteparameter. Hierdurch zeigen Geräte verschiedener Hersteller, bei Verwendung dieser Standardparameter, ein einheitliches Verhalten am Kommunikationsmedium.

Zur Konformitätsprüfung und Zertifizierung von Produkten mit dem Basisprofil wird eine unabhängige Sachverständigenstelle eingerichtet. Aufgrund von weiterführenden Standardisierungsarbeiten ist mit weiteren Ergänzungen zu rechnen.

2 Anwendung und Geräteeigenschaften

2.1 Allgemeines

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



Gerätefunktionen

Die Gerätefunktionen führen alle gerätspezifischen Funktionen aus.

Gerätemanagement

Das Gerätemanagement verwaltet Informationen, Eigenschaften und Fähigkeiten eines Geräts. Dazu kann ein nicht flüchtiger Speicher genutzt werden.

Kommunikationsfunktionen

Die Kommunikationsfunktionen führen alle kommunikationsspezifischen Funktionen aus.

Interaktionen zwischen den Funktionsblöcken

- 1 Prozessgrößen
- 2 Prozessdaten vom überlagerten Steuerungssystem zu den Gerätefunktionen
- 3 Prozessdaten von den Gerätefunktionen zum überlagerten Steuerungssystem
- 4 Speicherung von Geräteinformationen zu Management, Diagnose, Identifizierung
- 5 Auslesen der Geräteinformationen zu Management, Diagnose, Identifizierung
- 6 Abbildung auf den Prozessdatenkanal
- 7 Abbildung auf den Parameterkanal

3 Allgemeines

Zum Management und der Diagnose moderner Feldgeräte gehören neben Informationen über den augenblicklichen Zustand des Geräts auch Informationen zum Gerät selbst, historische Zustandsinformationen, Fehlerabbilder usw.

Bisher stehen nur wenige der möglichen Informationen standardmäßig zur Verfügung. Ziel dieses Dokuments ist es, ein Basisprofil zu definieren, welches ein einheitliches Management von Geräten, die mit einem Parameterkanal ausgestattet sind, ermöglicht. Dieses Basisprofil verschafft dem Gerätehersteller die Möglichkeiten, dem Endanwender (und Tools) deutlich mehr Informationen über sein Gerät und dessen Eigenschaften mitzuteilen.

Dieses „Mehr“ an Informationen erleichtert aufgrund der Vereinheitlichung insbesondere die Inbetriebnahme der Geräte. Auch im Servicefall ist jetzt die Möglichkeit vorhanden, einheitlich und damit schneller auf bestimmte Diagnoseinformationen zuzugreifen.

Profilverständnis:

„Inhaltlich gleichartige Informationen und Abläufe auch in einer einheitlichen Form darzustellen und dabei Hersteller und Anwender nicht einzuschränken, sondern Leitfaden zu sein.“

Um diesem Anspruch gerecht zu werden, werden die relevanten Informationen einheitlich in Standardobjekten hinterlegt. Es werden Mechanismen definiert, die den Zugriff auf diese Objekte regeln und für den Fehlerfall werden einheitliche Error Types festgelegt.

4 Übersicht

4.1 Indexbereiche der Kommunikationsobjekte

Die Kommunikationsobjekte im Basisprofil sind in folgende Indexbereiche eingeteilt:

Index (hex)	Name	Objekttyp
0	Objektverzeichnis	-
0001 – 0047	Allgemeingültige Standardobjekte Diese Objekte werden im Weiteren in diesem Dokument beschrieben.	Variable
0048 – 007F	Reserviert	
0080 – 5FFF	Herstellerspezifische Applikationsobjekte In diesem Bereich legt der Gerätehersteller seine gerätspezifischen Variablen-Objekte fest.	Variable
6000 – BFFF	Profilspezifische Applikationsobjekte Weitere festgelegte Applikationsobjekte sind den entsprechenden Profilen zu entnehmen (z. B. Drivecom, Sensor/Aktor usw.)	Variable
C000 – C07F	Projektion des Basisprofils für Sub-Busse	Variable
C080 – DFFF	Reserviert	
E000 – E3FF	Statische Variablenlisten - Zugriff über Makro-Dienste	VariableList
E400 – E7FE	Dynamische Variablenlisten - Zugriff über Makro-Dienste	VariableList
E7FF	Anlegen und Auslesen von Variablenlisten-Definitionen - Zugriff über Makro-Dienste	VariableListStruktur
E800 – E809	Allgemeingültige Domain-Variablen – Zugriff über Makro-Dienste Diese Objekte werden im Weiteren in diesem Dokument beschrieben.	Domain-Variable
E80A – E83F	Reservierte Domain-Variablen - Zugriff über Makro-Dienste	Domain-Variable
E840 – E8FF	Gerätspezifische Domain-Variablen - Zugriff über Makro-Dienste In diesem Bereich legt der Gerätehersteller seine gerätspezifischen Domain-Variablen fest.	Domain-Variable
E900 – FFFF	Reserviert	

4.2 Übersicht über die Standardobjekte

Index (hex)	Objektname	R/W	Länge (in Byte)	Datentyp	M/O/D
0001	VendorName	R	max. 57+1	Visible String	M
0002	VendorID	R	6+1	Visible String	O
0003	VendorText	R	max. 57+1	Visible String	O
0004	DeviceFamily	R	max. 57+1	Visible String	M
0005	Capabilities	R	N x 8	Array of Octet Strings[8]	M
0006	ProductFamily	R	max. 57+1	Visible String	O
0007	ProductName	R	max. 57+1	Visible String	M
0008	SerialNo	R	max. 57+1	Visible String	O
0009	ProductText	R	max. 57+1	Visible String	O
000A	OrderNumber	R	max. 57+1	Visible String	M
000B	HardwareVersion	R	max. 51	Record (2 Elemente)	M
000C	FirmwareVersion	R	max. 51	Record (2 Elemente)	M
000D	PChVersion	R	max. 51	Record (2 Elemente)	M
000E	CommProfile	R	max. 4+1	Visible String	M
000F	DeviceProfile	R	max. 4+1	Visible String	M
0010	Reserved				
0011	ProfileVersion	R	max. 51	Record (2 Elemente)	M
0012	VendorURL	R	max. 57+1	Visible String	O
0013	OnBoardDeviceDescFileName	R	max. 57+1	Visible String	O
0014	Location	R/W	max. 57+1	Visible String	O
0015	EquipmentIdent	R/W	max. 57+1	Visible String	O
0016	ApplDeviceAddr	R/W	2	UINT16	O
0017	Language	R/W	max. 56	Record (2 Elemente)	M
0018	DiagState	R	max. 207	Record (6 Elemente)	M
0019	ResetDiag	R/W	1	UINT8	O
001A	GetErrorRepMethod	R/W	1	UINT8	O
001B	TestMode	R/W	2	UINT16	O
001C	ControlTrace	R/W	1	UINT8	O
001D	Password	R/W	max. 40	Octet String[40]	O
001E	SetPassword	R/W	max. 45	Record (5 Elemente)	O
001F	PDTimeout	R/W	2	UINT16	O
0020	PDTimeoutCode	R/W	N x 2	Array of UINT16 (N Elemente)	O
0021	PChTimeout	R/W	2	UINT16	M
0022	PChTimeoutCode	R/W	2	UINT16	O
0023	AbortCode	R/W	2	UINT16	O
0024	ResetCode	R/W	N x 2	Array of UINT16 (N Elemente)	O
0025	PDIN	R	PD Länge	Octet String[N]	M
0026	PDOOUT	R/W	PD Länge	Octet String[N]	M
0027	GetExRight	R/W	1	UINT8	O

0028	ChangePDSet	R/W	2	UINT16	O
0029	ParamSetWriteControl	R/W	1	UINT8	O
002A	ConflictDictionary	R	N x 8	Array of Records (N x 6 Elemente)	D
002B	ParamSet	R/W	2	UINT16	O
002C	ParameterMoment	R/W	20	Record (2 Elemente)	O
002D	ResetParam	R/W	1	UINT8	M
002E	ParamHash	R	4	UINT32	O
002F	PDOOUT_Subst	R/W	PD Länge	Octet String[N]	D
0030	PF_Code	R/W	N x 2	Array of UINT16 (N Elemente)	O
0031	PDIN_Subst	R/W	PD Länge	Octet String[N]	D
0032	FieldBus_ID	R	3	Record (2 Elemente)	O
0033	DiagStateChannelNo	R	3	Record (2 Elemente)	X
0034	DiagStateAddValue	R	6	Record (2 Elemente)	X
0035	SubBusInfo	R	16	Record (6 Elemente)	O
0036	ActSubBusStructure	R/W	N x 8	Array of Records (3 Elemente)	D
0037	DeviceType	R	8	Octet String[8]	M
0038	ObjDescrReq	R/W	3	Record (2 Elemente)	D
0039	ObjDescr	R/W _D	max. 58	Record (16 Elemente)	D
003A	VersionCount	R	8	Array of UINT16 (4 Elemente)	M
003B	PDIN_Descr	R	N x 12	Array of Records (N x 3 Elemente)	M
003C	PDOOUT_Descr	R	N x 12	Array of Records (N x 3 Elemente)	M
003D	WakeUpTime	R	2	UINT16	M
003E	EnergyMgmt	R/W	5	Record (2 Elemente)	O
003F	Reserved				
0040	ListOfObjToRestore	R	N x 4	Array of Records (3 Elemente)	M
0041	RefSubBusStructure	R/W	N x 8	Array of Records (3 Elemente)	D
0042	ModuleStatus	R	N x 1	Array of Bit-Strings[8]	D
0043	SubBusBehaviour	R/W	3	Record (3 Elemente)	D
0044	SubBusControl	R/W	1	UINT8 (Hex)	D
0045	InitFWDownload	R/W	max. 58	Record (5 Elemente)	O
0047.01	AddInfo.SafetyProtType	R	8	Octet String[8]	D
0047.02	AddInfo.SafetyProtVers	R	8	Octet String[8]	D
0047.03	AddInfo.LegacyInfo	R	8	Octet String[8]	D
C000 - C07F	Projektion des Basisprofils für Sub-Busse	Def.	N	Entspricht in allen Belangen dem entsprechenden Objekt im Basisprofil.	D
E800	DiagStateLong	UR	N	Domain-Variable Record	O
E801	DiagHistory	UR	N	Domain-Variable Record	O
E802	DiagHistoryLong	UR	N	Domain-Variable Record	O
E803	TraceBuffer	UR	N	Domain-Variable Octet String	D
E804	LanguageAvailable	UR	N	Domain-Variable Record	O
E805	ObjDescrLong	UR	N	Domain-Variable Record	O
E806	ComplDiagState	UR	N	Domain-Variable Record	O
E807	Device FW	UR/DW	N	Domain-Variable Octet String	D
E808	OnBoardDeviceDescFile	UR	N	Domain-Variable Octet String	O
E809	BackUpDataCompr	UR/DW	N	Domain-Variable Octet String	O

R	= Read only,	M/D/O
W	= Write only,	M = Mandatory
R/W	= Read/Write,	O = Optional
R/W _D	= Read/Write dependent,	D = Dependent
UR	= Upload-Read,	
DW	= Download-Write,	

Details zu den Abkürzungen: siehe Anhang.

5 Kommunikationsobjekte

Die Kommunikationsobjekte werden nicht in einem Objektverzeichnis beschrieben, sondern müssen dem Anwender gewöhnlich implizit bekannt sein, also z. B. durch eine Beschreibung der Objekte im Handbuch zum entsprechenden Gerät.

Das gilt auch für mögliche Optionen/Werte. Welche Optionen/Werte eines Objekts zulässig sind, ist üblicherweise dem Handbuch zu entnehmen.

Es sind nur dann Beschreibungen/zulässige Werte definiert, wenn diese zur Laufzeit notwendig sein könnten.

5.1 Datentypen

Index of Type (dez)	Beschreibung Symbol	Anzahl Bytes
0	Undefined	N
1	Boolean*	1
2	INT8	1
3	INT16	2
4	INT32	4
5	UINT8	1
6	UINT16	2
7	UINT32	4
8	Floating-Point	4
9	Visible String (immer "0x00" terminiert!)**	1,2,3 ...
10	Octet String	1,2,3 ...
11	Date	7
12	Time-Of-Day	6
13	Time-Diff	6
14	Bit-String ***	1,2,3 ...

* True = 0xFF, False = 0x00

** Visible Strings sind zur einfacheren Verarbeitung immer 0x00 terminiert. In der Darstellung der Länge eines Objekts wird dieser Tatsache durch die Schreibweise "N+1" Rechnung getragen. "N" stellt dabei die Anzahl Byte der tatsächlich sichtbaren Zeichen dar, "+1" die notwendige 0x00-Terminierung. "max. 57+1" bedeutet, dass das Objekt maximal 57 Byte sichtbare Zeichen und die abschließende 0x00 haben kann, also eine gesamte Länge inklusive der abschließenden 0x00 von 58 Byte.

*** Bit-Strings haben immer eine Länge von $n \times 8$ Bit, wobei $n \in \mathbb{N}$ (n Element der natürlichen Zahlen).

5.2 Datenobjekte

Übersicht über die verwendeten Datenobjekte.

Objectcode	Object
0x00	Undefined
0x02	Domain-Variable
0x07	Simple-Variable
0x08	Array-Variable
0x09	Record-Variable
0x0A	Variablenliste Statisch und dynamisch
0x0B	String-Variable

5.2.1 Objekt „Domain-Variable“

Domain-Variablen sind Objekte, deren Datenlänge größer als die max. PDU-Size der einfachen Standarddienste sein kann. Das Objekt Domain-Variable dient zur Übertragung von anwenderspezifisch strukturierten Daten von meist größerem bis sehr großem Umfang (größer (max. PDU-Size) Byte). Die Objektbeschreibung muss dem Applikationsprogramm implizit bekannt sein.



Das Objekt Domain-Variable kann in der Regel nur über die Makrodienste Download-Write und Upload-Read und dann auch nur vollständig angesprochen werden. Im Objekt 0x000E „CommProfile“ ist die Eigenschaft „Up-/Download-Protokoll“ durch die Werte (634 bzw. 635) gekennzeichnet.

Die Gesamtgröße der Domain-Variable darf die maximale Nutzdatenmenge von $[PDU-Size - 8] \times 0xFFFF0$ nicht überschreiten (ca. 3,5 MB).

Im Prinzip kann jeder Objektindex eine Domain-Variable sein. Um eine einfache Unterscheidung realisieren zu können, wird den Domain-Variablen bevorzugt ein bestimmter Bereich, siehe „Indexbereiche der Kommunikationsobjekte“, zugewiesen. Damit ist dem Master wie dem Slave implizit bekannt, dass auf diese typischerweise über die Makrodienste zugegriffen wird.

Für die Umsetzung der Makrodienste kann es, je nach System, auch eigene Standarddienste, geben. Im Objekt 0x000E „CommProfile“ ist die Eigenschaft „Up-/Download-Protokoll“ durch die Werte (636 bzw. 637) gekennzeichnet.

In diesem Falle kann auf die Domain-Variablen optional mit den einfachen Standarddiensten, dann ggf. über Subindizes auch teilweise, zugegriffen werden, solange die max. PDU-Größe dieses Dienstes nicht überschritten wird.

5.2.1.1 Domain-Variable - formale Beschreibung

Object: Domain-Variable
Objectcode: 0x02
Key Attribute (M): Index
Attribute (M): Length (max. PDU-Size – 8) x 0xFFF0 Byte)
Attribute (O): Password

Services:

- (O) Read
- (O) Write
- (O) Download-Write
- (O) Upload-Read

5.2.2 Objekt „Simple-Variable“

Das Objekt Simple-Variable stellt eine einzelne, einfache Variable dar. Die Objektbeschreibung muss dem Applikationsprogramm implizit bekannt sein. Die Größe einer Simple-Variable darf die maximale Nutzdatenmenge von (PDU-Size - 6) Bytes nicht überschreiten.

5.2.2.1 Simple-Variable - formale Beschreibung

Object: Simple-Variable
Objectcode: 0x07
Key Attribute (M): Index
Attribute (M): Length (max. (PDU-Size - 6) Byte)
Attribute (O): Password

Services:
(O) Read
(O) Write
(O) Information-Report

5.2.3 Objekt „Array“

Das Objekt Array besteht aus einer Reihung von Simple-Variablen des gleichen Datentyps. Die Objektbeschreibung muss dem Applikationsprogramm implizit bekannt sein. Wenn ein Element des Objekts angesprochen werden soll, so muss im Service neben dem Index auch ein Subindex angegeben werden. Der Subindex 1 greift auf das 1. Element des Objekts zu. Der Subindex 0 adressiert das gesamte Objekt. Wenn beim Zugriff auf das Gesamtobjekt Fehler auftreten, die einem einzelnen Subindex geschuldet sind, so wird der entsprechende Error Type erzeugt und alle Daten werden als ungültig angesehen.

Es ist keine Pflicht, jedes Element eines Arrays einzeln ansprechen zu können.

Die Summe der Nutzdaten der einzelnen Elemente des Arrays darf die maximal zulässige Nutzdatenmenge von (PDU-Size - 6) Byte überschreiten. In diesem Fall ist es logischerweise Pflicht, jedes Element eines Arrays einzeln ansprechen zu können.

5.2.3.1 Array-Variable - formale Beschreibung

Object: Array-Variable
Objectcode: 0x08
Key Attribute (M): Index
Attribute (M): Length per Element (max. (PDU-Size - 6) Byte)
Attribute (M): Number-of-Elements
Attribute (O): Password

Services:
(O) Read
(O) Write
(O) Download-Write
(O) Upload-Read
(O) Information-Report

5.2.4 Objekt „Record“

Das Objekt Record besteht aus verschiedenen Elementen. Diese sind eine Reihung von Simple-Variablen unterschiedlichen Datentyps und String-Variablen. Die Objektbeschreibung muss dem Applikationsprogramm implizit bekannt sein. Das Objekt Record kann vollständig oder elementweise angesprochen werden. Soll das Objekt vollständig angesprochen werden, so wird im Service der zugehörige Index und als Subindex „0“ angegeben.

Wenn beim Zugriff auf das Gesamtobjekt Fehler auftreten, die einem einzelnen Subindex geschuldet sind, so wird die entsprechende Error Type erzeugt und alle Daten werden als ungültig angesehen.

Wenn ein Gesamtobjekt über den Subindex „0“ angesprochen wird, dann ist eine Konsistenz automatisch gewährleistet, die sonst nur mit erheblichem Aufwand zu erreichen ist. Z. B. ist es deshalb nicht sinnvoll, Objekt 0x0018 „DiagState“ in dessen einzelnen Elementen zu übertragen. Der Aufwand zur Sicherstellung der Konsistenz der einzelnen Elemente zueinander wäre zu groß.

Wenn ein Element des Objekts angesprochen werden soll, so muss im Service neben dem Index auch ein Subindex angegeben werden. Der Subindex 1 greift auf das 1. Element des Objekts zu.

Es ist keine Pflicht, jedes Element eines Record einzeln ansprechen zu können.

Die Summe der Nutzdaten der einzelnen Elemente des Record darf die maximal zulässige Nutzdatenmenge von (PDU-Size - 6) Byte überschreiten.

In diesem Fall ist es logischerweise Pflicht, jedes Element eines Record einzeln ansprechen zu können oder es handelt sich um eine Domain-Variable.

5.2.4.1 Record-Variable - formale Beschreibung

Object:	Record-Variable
Objectcode:	0x09
Key Attribute (M):	Index
Attribute (M):	Length per Element (max. (PDU-Size - 6) Byte)
Attribute (M):	List-of-Elements
Attribute (O):	Password

Services:

- (O) Read
- (O) Write
- (O) Download-Write
- (O) Upload-Read
- (O) Information-Report

5.2.5 Objekt „Variablenliste“

Eine Variablenliste ist eine spezielle Form der \Rightarrow Domain-Variable, der ein eigener Indexbereich zugewiesen wurde. Sie enthält eine Zusammenstellung einzelner Variablen. Auch hier muss die Struktur dem Applikationsprogramm implizit bekannt sein.

Das Zusammenfassen mehrerer Einzelvariablen zu einer Variablenliste ist nur für Variablen kleiner Datentypen (z. B. UINT8, Integer32 etc.) sinnvoll. Theoretisch kann die Länge einer einzelnen Variable einer Variablenliste bis zu maximal (PDU-Size – 8) Byte betragen. Auf eine solche Variable kann aber effektiver mit einem normalen Read-Dienst zugegriffen werden.

5.2.5.1 Variablenliste - formale Beschreibung

Object:	Variablenliste
Objectcode:	0x0A
Key Attribute (M):	Index
Attribute (M):	Number-Of-Elements
Attribute (M):	List of Element-Index
→ Attribute (M):	→ Index
Attribute (O):	Password

Services:

- (O) Read
- (O) Write
- (O) Download-Write
- (O) Upload-Read

5.2.5.2 Statische Variablenliste

Die Variablenlisten können statisch durch den Gerätehersteller angelegt sein. Diese Variablenlisten liegen in dem Indexbereich "statische Variablenlisten" siehe Indexbereiche der Kommunikationsobjekte.

Die Struktur der statischen Variablenliste ist über die Variable "VariableListRecord" (Index E7FF) auslesbar. Hierzu wird zunächst der Index, der zu lesenden Variablenliste auf das Objekt E7FF geschrieben. Hierbei ist der Download-Write zu verwenden, da Objekt E7FF eine Domain-Variable ist. Anschließend kann durch einen Upload-Read-Zugriff auf Objekt E7FF die „VariableListRecord“ ausgelesen werden.

Die Übertragung der „VariableListRecord“ erfolgt nach folgendem Aufbau:

Index (2 Bytes)	Bedeutung
1. Wert	Index der 1. Variable in der Liste
2. Wert	Index der 2. Variable in der Liste
...	
...	
N. Wert	Index der N. Variable in der Liste

5.2.5.3 Dynamische Variablenliste

Dynamische Variablenlisten werden über die Variable "VariableListRecord" (Index E7FF) angelegt. Diese Variablenlisten liegen in dem Indexbereich "dynamische Variablenlisten" (siehe Indexbereiche der Kommunikationsobjekte). Durch einen Download-Write-Zugriff auf Objekt E7FF wird die „VariableListRecord“ angelegt.

Die Übertragung der „VariableListRecord“ erfolgt nach folgendem Aufbau:

Index (2 Bytes)	Bedeutung
1. Wert	Index unter der die VariablenListRecord angelegt werden soll
2. Wert	Index der 1. Variable in der Liste
3. Wert	Index der 2. Variable in der Liste
...	
...	
N+1. Wert	Index der N. Variable in der Liste

5.2.5.4 Übertragungsformat Variablenlisten

Zum Schreiben/Lesen einer Variablenliste werden alle Werte der Einzelvariablen nacheinander in das Datenfeld des Download-Write/Upload-Read eingetragen.

	Bedeutung
1. Wert	Inhalt der 1. Variable in der Liste
2. Wert	Inhalt der 2. Variable in der Liste
...	
...	
N. Wert	Inhalt der N. Variable in der Liste

5.2.6 Objekt „String-Variable“

Das Objekt String-Variable stellt eine einzelne, einfache Variable dar, die durch einen bestimmten Datentyp (Octet String, Visible String oder Bit-String) charakterisiert wird. Die Objektbeschreibung des Objekts String-Variable ist statisch definiert. Durch die Objektbeschreibung des Objekts String-Variable wird die Abbildung einer String-Variablen auf eine real existierende String-Variable im Anwendersystem festgelegt. In der Beschreibung des Objekts String-Variable ist nur ein Datentyp aus Octet String, Visible String oder Bit-String zugelassen. Eine String-Variable hat den gleichen Aufbau wie eine Simple-Variable. Der Unterschied besteht in der variablen Länge der Datentypen. Im Speicher wird die Maximallänge projiziert. Übertragen wird jedoch nur die jeweils aktuelle Länge.

5.2.6.1 String-Variable - formale Beschreibung

Object: String-Variable
Objectcode: 0x0B
Key Attribute (M): Index
Attribute (M): Max Length (max. (PDU-Size - 6) Byte)
Attribute (O): Password

Services:

- (O) Read
- (O) Write
- (O) Download-Write
- (O) Upload-Read
- (O) Information-Report

6 Services

Der Zugriff auf alle einfachen Variablen, Arrays und Records (Anzahl der Daten < PDU-Size) erfolgt mit den einfachen/Standarddiensten. Diese sind:

- Read
- Write
- Information Report/Fetch.

Da diese Dienste zur Funktionalität der Parameterkanal-Implementierung gehören, werden Sie nicht an dieser Stelle beschrieben, sondern es wird auf die entsprechende Parameterkanal-Dokumentation*) verwiesen.

*) z. B. IBS PCP RE HB (5052b.pdf) und IBS PCP RR HB (5054b.pdf)

Der Zugriff auf alle Domain-Variablen, Listen, Arrays und Records (Anzahl der Daten > PDU-Size) erfolgt mit den Makro-Diensten. Diese sind:

- Upload-Read
- Download-Write

Wenn mit dem Standarddienst „Read“ auf ein Objekt zugegriffen wird, das mehr Daten enthält, als die PDU-Size erlaubt, so wird der Zugriff mit dem Error Type: Error Class: 05, Error Code: 02, AddCode 0x0018 (Service-PDU-Size - Objekt Länge für dieses Objekt passt nicht) erzeugt.

Wenn mit dem Standarddienst „Read“ auf ein Objekt zugegriffen wird, das mehr Daten enthalten könnte, als die PDU-Size erlaubt, dies aber zum aktuellen Zeitpunkt nicht tut, so ist dies fehlerfrei möglich.

Wenn mit dem Standarddienst „Write“ auf ein Objekt zugegriffen wird, das mehr Daten aufnehmen kann, als die PDU-Size erlaubt, so ist dies fehlerfrei möglich.

Wenn mit dem Makrodienst „Download-Write“ auf ein Objekt zugegriffen wird, das weniger Daten aufnehmen kann, als mit einer PDU geschrieben werden sollen, so wird der Zugriff mit dem Error Type: Error Class: 05, Error Code: 02, AddCode 0x0018 (Service-PDU-Size - Objekt Länge für dieses Objekt passt nicht) erzeugt.

Wenn mit dem Makrodienst „Upload-Read“ auf ein Objekt zugegriffen wird, das weniger Daten liefert, als die PDU-Size erlaubt, so ist dies fehlerfrei möglich.

Auf Domain-Objekte und Variablenlisten, die durch einen Objektindex im zugewiesenen Bereich explizit definiert sind, kann immer mit Makrodiensten zugegriffen werden.

6.1 Adressierung

Die Objekte eines Geräts, auf die mit den Services zugegriffen wird, werden über Index und Subindex adressiert.

Dabei sind folgende Festlegungen zu beachten:

6.1.1 Adressierung über Subindex gleich „0“

- Der Zugriff über den Subindex „0“ ist **mandatory** und steht deshalb immer zur Verfügung.
- Der Subindex „0“ adressiert das ganze Index-Objekt, inkl. aller Sub-Objekte. D. h. man kann über den Subindex „0“ im Prinzip auf alle Objekte aller Subindizes zusammen zugreifen. Die Struktur (Anzahl, leere Subindizes, Datenlänge etc.) der Sub-Objekte muss dabei implizit bekannt sein. Bei manchen Implementierungen ist dies die einzige Art, um auf die Sub-Objekte zugreifen zu können.
- Da die Summe der Daten aller Subindizes eines Index größer sein kann als die max. PDU-Size, bedeutet das aber auch, dass ggf. über den Subindex „0“ nicht auf alle Sub-Objekte gemeinsam zugegriffen werden kann.
- Im Fall des Lesens wird der Fehler Fehlertyp „PDU-Größen-Problem“ (Error Class = 0x05, Error Code = 0x02) gemeldet.
Aus diesem Grund sollten Objekte, bei denen die Summe der Datenlängen der Sub-Objekte größer als die PDU-Size ist, vermieden werden.
- Im Fall des Schreibens kann ebenfalls der Fehler Fehlertyp „PDU-Größen-Problem“ (Error Class = 0x05, Error Code = 0x02) gemeldet werden.
Der Slave kann aber auch eine beliebige kleinere Datenlänge akzeptieren, sofern dies in der Applikation sinnvoll ist.
- Die Subindizes eines Index können unterschiedliche Lese-/Schreibrechte besitzen. Solange „nur lesbare“ Objekte beim gemeinschaftlichen Zugriff mit denselben (schon vorhandenen) Werten beschrieben werden, ist dies nicht als Fehler zu bewerten und wird folglich nicht mit einem Error Type beantwortet.

6.1.2 Adressierung über Subindex ungleich „0“

- Der Zugriff über den Subindex ungleich „0“ ist optional selbst dann, wenn das Subobjekt selbst mandatory ist.
- Diese Eigenschaft steht deshalb in Abhängigkeit der durch den Slave zur Verfügung gestellten Funktionen zur Verfügung.
- Der Zugriff über einen Subindex ist immer dann zwingend notwendig, wenn die Summe der Daten aller Subindizes eines Index größer ist als die PDU-Size.
- Jedes über einen Subindex adressierte Sub-Objekt stellt im Prinzip ein eigenständiges Objekt mit eigenen Eigenschaften und eigenen Bedeutungen dar.
- Jedes über einen Subindex adressierte Sub-Objekt kann die gesamte zur Verfügung stehende Nutzdatenlänge (max. PDU-Size) nutzen.

6.2 Datenlänge

Die Anzahl der zu übertragenden Daten ist variabel – von „0“ bis „max. PDU-Size“. Sie wird mit jedem Dienst angegeben. Master und Slave teilen die Anzahl im Dienst dem jeweiligen Partner mit.

Dabei ist es dem Slave frei gestellt, eine beliebige kleinere Datenlänge zu akzeptieren, sofern dies in der Applikation sinnvoll ist.

Um eine Kompatibilität zu Vorgängergeräten, bei denen Objekte inzwischen erweitert worden sind, zu gewährleisten, ist dies sogar zwingend notwendig.

Zur Erläuterung ein Beispiel:

Ein Objekt verfügt nach einer Überarbeitung nun über eine Länge von 11 Bytes aufgeteilt in 5 Sub-Objekte. Vorher war es nur 2 Byte lang und besaß keine Subindizes. D. h. der alte Inhalt ist jetzt über den Subindex „1“ (2 Byte) ansprechbar.

Wenn jetzt 2 Byte auf den Subindex „0“ geschrieben werden, so ist dies zulässig und der Subindex „1“ wird korrekt beschrieben.

Beim Lesen über den Subindex „0“ muss der Master (die Applikation) aus Kompatibilitätsgründen das 3. und alle folgenden Bytes ignorieren.

Um die Effizienz der Übertragung nicht einzuschränken, werden, insbesondere bei Strings, immer nur so viele Bytes übertragen, wie auch tatsächlich benötigt werden/definiert sind.

6.3 Busy

Bei der Abarbeitung von Services durch einen Slave kann es einen Sonderfall geben. Z. B. können die Nutzdaten wegen ausstehender Aktionen in der Slave-Applikation noch nicht verfügbar sein.

Wenn ein Service von der Slave-Applikation in der üblicherweise erwarteten Zeit (Time-out des Dienstes) nicht beendet werden kann, so wird der Fehlertyp „Busy“ (Error Class = 0x08, Error Code = 0x0B) gemeldet.

Dem Master wird so mitgeteilt, dass die Slave-Applikation aktiv ist und der Dienst noch bearbeitet wird, jedoch zum jetzigen Zeitpunkt noch nicht beendet ist.

Im Additional Code wird die Zeit in ms angegeben, die die Bereitstellung der Ergebnisse voraussichtlich noch benötigt.

„0xFFFF“ bedeutet, die voraussichtliche Ausführungsdauer ist unbekannt.

Dieser Fehlertyp ist vor der üblichen Timeout-Zeit durch den Slave abzusetzen.

Wenn der Dienst innerhalb der in Additional Code angegebenen Zeit + dem üblichen Timeout mit gleichem Inhalt wiederholt wird, dann erfolgt vor Ablauf des üblichen Timeouts:

- entweder eine positive Response (damit ist der Dienst beendet),
- eine andere negative Response (auch damit ist der Dienst beendet) oder
- eine negative Response mit erneut diesem Error Type. Damit wird der oben beschriebene Mechanismus erneut gestartet.

Über den Report-/Fetch-Mechanismus wird das dann gültige Ergebnis spätestens nach der in Additional Code angegebenen Zeit + dem üblichen Timeout durch den Slave aktiv zum Master gemeldet.

Der übliche Timeout ist in Objekt 0x0021 „PChTimeout“ definiert.

6.4 Makrodienste

Auf die Objekte mit einer Datenlänge größer max. PDU-Size der einfachen Standarddienste und auf die Objekte Domain-Variable und Variablenliste wird mit den Makrodiensten Download-Write und Upload-Read zugegriffen.

Sind die Makrodienste Download-Write und Upload-Read implementiert, muss im Objekt 0x000E „CommProfile“ die Eigenschaft „Up-/Download-Protokoll“ (634 bzw. 635) eingetragen werden.

Für die Umsetzung der Makrodienste kann es, je nach System, auch eigene Standarddienste geben.

Im Prinzip kann jedes Objekt sowohl über die Standarddienste (solange die Datenlänge kleiner max. PDU-Size ist), als auch über die Makrodienste angesprochen werden. Um den Zugriff unterscheiden zu können, muss es eigene Standard- und Makrodienste geben. Dies ist im Objekt 0x000E „CommProfile“ mit der Eigenschaft "Up-/Download-Protokoll (636 bzw. 637) angegeben.

Die Handhabung der Makrodienste wird im Folgenden beschrieben:

6.4.1 Download-Funktionalität mit Write-Service (Download-Write)

Die Übertragung von großen Datenmengen (z. B. Applikationsprogramm, Firmware-Update, Variablenlisten usw.) wird im Gegensatz zur Standard-Parameterkanal-Implementierung (enthält Download-Services) mit Hilfe des Write-Services abgewickelt. Hierzu wird in den Nutzdaten des Write-Services ein Download-Protokoll definiert, welches eine segmentierte Übertragung größerer Datenmengen erlaubt. Wenn der Dienst nicht anderweitig eindeutig gekennzeichnet werden kann, kann der Download nur auf Domain-Variablen und Variablenlisten durchgeführt werden.

Aufbau der Download-Write-PDU

Parameter	Req/Ind	Rsp/Cnf
Kommunikationsreferenz	M	M
Modulnummer	M	M
Index	M	
Subindex	O	
Datenfeld	M	
Segmentnummer	M	
Daten	M	
Result(+)		S
Result(-)		S
Error Type		M

Kommunikationsreferenz/Slot- oder Teilnehmernummer

Die Kommunikationsreferenz adressiert den gewünschten Kommunikationspartner. („Compact“ implementierte Teilnehmer können nur mit dem Master kommunizieren (kein Peer-to-Peer), sodass die Kommunikationsreferenz immer 2 ist.)

Modulnummer

Zur Adressierung von Sub-Modulen eines modularen Geräts wird der Parameter „Modulnummer“ genutzt.
Näheres im Kapitel „Modulare Geräte“.

Index

Der Parameter Index adressiert das zu schreibende Objekt.

Subindex

Der Parameter Subindex hat für die Download-Funktionalität den Wert „0“

Datenfeld

Der Parameter Datenfeld enthält das Download-Protokoll.

Datenfeld	
Segmentnummer 2 Byte Download-Steuerung	Nutzdaten 1 ... (PDU-Size – 8) Byte Download-Datenblöcke z. B. 1 ... 56 Byte für die „Compact“-Implementierung

Segmentnummer

Dieser Parameter enthält die Nummer des zu schreibenden Datenblocks, wobei folgende Definition gilt:

Segmentnummer = 1	Einleiten einer Download-Sequenz mit dem ersten Datenblock.
Segmentnummer = 2 ... 0xFFFF0	Datenblock-Nummer N
Segmentnummer = 0xFFFFD	letzter Datenblock enthält ausschließlich <ul style="list-style-type: none"> • die Anzahl der übertragenen Datenblöcke (ohne diesen Schlussdatenblock) (2 Byte) und • das CRC16-Rest-Polynom¹⁾ (2 Byte)
Segmentnummer = 0xFFFFE	letzter Datenblock enthält ausschließlich <ul style="list-style-type: none"> • die Anzahl der übertragenen Datenblöcke (ohne diesen Schlussdatenblock) (2 Byte) und • das CRC32-Rest-Polynom¹⁾ (4 Bytes)
Segmentnummer = 0xFFFFF	letzter Datenblock enthält ausschließlich <ul style="list-style-type: none"> • die Anzahl der übertragenen Datenblöcke (ohne diesen Schlussdatenblock) (2 Byte) und kein-CRC Rest

¹⁾ Der CRC wird immer über die gesamten reinen Nutzdaten gebildet. Die Segmentnummer gehört nicht zu den Nutzdaten.

Verwendet werden folgende Generatorpolynome:

16 Bit CRC: CRC-CCITT:

$$x^{16} + x^{12} + x^5 + 1$$

Startwert: 0xFFFF

Start mit niederwertigstem Bit.

Das Ergebnis der CRC-Rechnung (Restpolynom) wird direkt übernommen.

32Bit - CRC: CRC-32 (IEEE 802.3):

$$x^{32} + x^{26} + x^{23} + x^{22} + x^{16} + x^{12} + x^{11} + x^{10} + x^8 + x^7 + x^5 + x^4 + x^2 + x + 1$$

Startwert: 0xFFFF.FFFF

Start mit niederwertigstem Bit.

Das Ergebnis der CRC-Rechnung (Restpolynom) wird zum Schluss invertiert.

Der Download-Server muss die lückenlos aufsteigende Reihenfolge der Segmentnummer kontrollieren und im Fehlerfall dem Client (typ. Feldbus Master) eine negative Response senden. Im Falle einer negativen Response kann der Client das korrekte Segment senden oder die Übertragung mit einem Schlussegment beenden. Nur ein Schlussdatenblock beendet den Download-Service.

Daten

Dieser Parameter enthält die Datenblöcke. Die Länge dieser Datenblöcke ist abhängig von der maximalen PDU-Größe. Für die PDU-Größe von 64 Bytes beträgt die maximale Datenblocklänge 56 Bytes. Die maximale Datenblocklänge muss nicht ausgenutzt werden.

Result(+)

Der Parameter Result(+) kennzeichnet ein positives Ergebnis. Nur bei einem positiven Ergebnis darf die Segmentnummer weiter gezählt werden.

Result(-)

Der Parameter Result(-) kennzeichnet ein negatives Ergebnis.

Error Type

Der Parameter Error Type enthält die Fehlerursache.

Der Inhalt des Parameter Error Type entspricht dem des Write-Service, mit folgender Definition für die spezifischen Download-Sequenzfehler:

Error Class „8“ - Other

Error Code „1“ – Profile specific

Additional Code (hex)	Bedeutung
0x00A0 Invalid segment number - Segment missing	Ungültige Segmentnummer (Segment fehlt)
0x00A1 Resource unavailable	Es sind keine Ressourcen (Speicher) für den Download mehr vorhanden
0x00A2 Invalid CRC	Falscher CRC
0x00A3 File open error	Fehler beim Öffnen der Datei (wenn File-System vorhanden)
0x00A4 File write error	Fehler beim Schreiben der Datei (wenn File-System vorhanden)
0x00A5 File close error	Fehler beim Schließen der Datei (wenn File-System vorhanden)
0x00A6 Segment missing	Es wurden weniger Datenblöcke empfangen als im letzten Segment angegeben.
0x00A7 Segment overrun	Es wurden mehr Datenblöcke empfangen als im letzten Segment angegeben.
0x00A9 Invalid segment number – double segment	Ungültige Segmentnummer (Segment doppelt, Segment wurde ignoriert)

6.4.2 Upload-Funktionalität mit Read-Service (Upload-Read)

Das Auslesen von großen Datenmengen (z. B. Applikationsprogramm, Backups, Variablenlisten usw.) wird im Gegensatz zur Standard-Parameterkanal-Implementierung (enthält Upload-Services) mit Hilfe des Read-Services abgewickelt. Hierzu wird in dem Subindex des Read-Service ein Upload-Protokoll definiert, welches eine segmentierte Übertragung größerer Datenmengen erlaubt. Wenn der Dienst nicht anderweitig eindeutig gekennzeichnet werden kann, kann der Upload auf Domain-Variable und Variablenlisten durchgeführt werden.

Aufbau der Upload-Read PDU

Parameter	Req/Ind	Rsp/Cnf
Kommunikationsreferenz	M	M
Modulnummer	M	M
Index	M	
Subindex	M	
Result(+)		S
Datenfeld		M
Segmentnummer		M
Daten		M
Result(-)		S
Error Type		M

Kommunikationsreferenz

Wie Download-Funktionalität mit Write-Service (Download-Write)

Modulnummer

Wie Download-Funktionalität mit Write-Service (Download-Write)

Index

Wie Download-Funktionalität mit Write-Service (Download-Write)

Subindex

Über einen speziellen Subindex wird eine Upload-Sequenz eingeleitet, wobei folgende Definition gilt:

- Subindex = 0xFF Einleiten einer Upload-Sequenz mit Anfordern des ersten Datenblocks
- Subindex = 0x00 Anforderung des nächsten Datenblocks
- Subindex = 0x01 Anforderung des jeweils vorigen Datenblocks (z. B. nach einem Fehler)

Result(+)

Dieser Parameter kennzeichnet ein positives Ergebnis. Nur bei einem positiven Ergebnis darf die Segmentnummer weiter gezählt werden.

Datenfeld

Der Parameter Datenfeld enthält das Upload-Protokoll.

Datenfeld	
Segmentnummer 2 Byte Upload-Steuerung	Nutzdaten 1 ... (PDU-Size – 8) Byte Download-Datenblöcke 1 ... 56 Byte für die „Compact“-Implementierung

Segmentnummer

Dieser Parameter enthält die Nummer des zu lesenden Datenblocks, wobei folgende Definition gilt:

Segmentnummer = 1	Einleiten einer Upload-Sequenz mit dem ersten Datenblock.
Segmentnummer = 2 ... 0xFFFF0 Segmentnummer = 0xFFFFD	Datenblock-Nummer N letzter Datenblock enthält ausschließlich <ul style="list-style-type: none"> die Anzahl der übertragenen Datenblöcke (ohne diesen Schlussdatenblock) (2 Byte) und das CRC16-Rest-Polynom¹⁾ (2 Byte)
Segmentnummer = 0xFFFFE	letzter Datenblock enthält ausschließlich <ul style="list-style-type: none"> die Anzahl der übertragenen Datenblöcke (ohne diesen Schlussdatenblock) (2 Byte) und das CRC32-Rest-Polynom¹⁾ (4 Byte)
Segmentnummer = 0xFFFFF	letzter Datenblock enthält ausschließlich <ul style="list-style-type: none"> die Anzahl der übertragenen Datenblöcke (ohne diesen Schlussdatenblock) (2 Byte) und kein CRC-Rest

¹⁾ Der CRC wird immer über die gesamten reinen Nutzdaten gebildet. Die Segmentnummer gehört nicht zu den Nutzdaten.

Verwendet werden folgende Generatorpolynome:

CRC-16: CRC-CCITT: $x^{16} + x^{12} + x^5 + 1$

CRC-32: IEEE 802.3: $x^{32} + x^{26} + x^{23} + x^{22} + x^{16} + x^{12} + x^{11} + x^{10} + x^8 + x^7 + x^5 + x^4 + x^2 + x + 1$

Der Upload-Client muss die lückenlos aufsteigende Reihenfolge der Segmentnummer kontrollieren. Im Fehlerfall kann der Client dann das letzte Segment mit dem Subindex 0xFE erneut anfordern oder den Upload erneut mit Subindex 0xFF einleiten.

Daten

Dieser Parameter enthält die Datenblöcke. Die Länge dieser Datenblöcke ist abhängig von der maximalen PDU-Größe. Für die PDU-Größe von 64 Bytes beträgt die maximale Datenblocklänge 56 Bytes. Die maximale Datenblocklänge muss nicht ausgenutzt werden.

Result(-)

Dieser Parameter kennzeichnet ein negatives Ergebnis.

Error Type

Der Parameter Error Type enthält die Fehlerursache.

Der Inhalt des Parameters Error Type entspricht dem des Read-Service, mit folgender Definition für die spezifischen Upload-Sequenzfehler:

Error Class "8" - Other
Error Code „1“ - Profile specific

Additional Code (hex)	Bedeutung
0x00A0 Invalid segment number - Segment missing	Upload ohne Einleitung mit Subindex == 0xFF
0x00A3 File open error	Fehler beim Öffnen der Datei (wenn File-System vorhanden)
0x00A5 File close error	Fehler beim Schließen der Datei (wenn File-System vorhanden)
0x00A8 File read error	Fehler beim Lesen der Datei (wenn File-System vorhanden)

6.4.3 Beispiel Download Variablenliste

Ein „Compact“-Gerät verfügt in diesem Beispiel unter anderen über folgende Objekte, die zu einer Variablenliste zusammengestellt werden.

Index (hex)	Subindex	Data Type	Length	Example-Value (hex)
0080	0	UINT8	1	FF
0081	7	UINT16	2	12 34
0090	3	UINT8	1	00
0091	0	Octet String	16	00 01 02 03 04 05 06 07 08 09 0A 0B 0C 0D 0E 0F
0100	0	Octet String	16	10 11 12 13 14 15 16 17 18 19 1A 1B 1C 1D 1E 1F
0101	0	Octet String	16	20 21 22 23 24 25 26 27 28 29 2A 2B 2C 2D 2E 2F
0105	0	Octet String	16	30 31 32 33 34 35 36 37 38 39 3A 3B 3C 3D 3E 3F

Die zusammengestellte Variablenliste ist in diesem Beispiel unter dem Index 0xE000 abgelegt.

Download-Write

1. Download-Write-Request/Indication

Parameter	Inhalt (hex)
Kom.Ref.	02
Modulnummer	00
Index	E0 00
Subindex	00
Segmentnr.	00 01
Daten	FF 12 34 00 00 01 02 03 04 05 06 07 08 09 0A 0B 0C 0D 0E 0F 10 11 12 13 14 15 16 17 18 19 1A 1B 1C 1D 1E 1F 20 21 22 23 24 25 26 27 28 29 2A 2B 2C 2D 2E 2F

2. Download-Write-Request/Indication

Parameter	Inhalt (hex)
Kom.Ref.	02
Modulnummer	00
Index	E0 00
Subindex	00
Segmentnr.	00 02
Daten	30 31 32 33 34 35 36 37 38 39 3A 3B 3C 3D 3E 3F

3. Download-Write-Request/Indication

Parameter	Inhalt (hex)
Kom.Ref.	02
Modulnummer	00
Index	E0 00
Subindex	00
Segmentnr.	FF FF
Daten	00 02

6.5 Liste der zulässigen Error Types der Kommunikation

Wenn beim Schreiben/Lesen der Objekte Fehler auftreten, z. B. es werden unzulässige Werte übertragen, so sind die Zugriffe mit den entsprechenden Error Types (s. u.) abzulehnen.

Kommunikations-Error Types existieren nur als negative Confirmation eines Parameterkanaldienstes (Standarddienste)

Bei der Kommunikation über den Parameterkanal auftretenden Fehlermeldungen:

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

• Error Class	1 Byte, UINT8, hexadezimale Darstellung
• Error Code	1 Byte, UINT8, hexadezimale Darstellung
• Additional Code	2 Byte, UINT16, hexadezimale Darstellung

6.5.1.1 Error Class und Error Code

Error Class	Error Code	Bedeutung
2 Application-Reference		Diese Fehlerklasse bezieht sich auf die Kommunikationsbeziehung, über die der Service abgearbeitet wird.
	1 Application unreachable	0x0201 Der Zugriff auf das Objekt ist nicht möglich, weil die Applikation nicht implementiert/nicht vorhanden ist. Typisch ist dieser Error Type, wenn über die Modulnummer auf ein nicht existierendes Modul eines modularen Geräts zugegriffen wird. Modul nicht vorhanden oder Modulnummer falsch.
	0 other	0x0200 Der Fehler in dieser Klasse lässt sich keinem der vorher angegebenen Error Types zuordnen.
5 Service		Diese Fehlerklasse wird bei einem fehlerhaften Service gemeldet. Dies betrifft alle Parameter mit Ausnahme des eigentlichen Objekts.
	1 Object-State-Conflict	0x0501 Der augenblickliche Zustand des Objekts verhindert eine Ausführung des Services.
	2 Service-PDU-Size	0x0502 PDU-Größen-Problem. z. B. sollen mehr Daten übertragen werden, als die vereinbarte max. PDU-Größe zulässt.
	3 Object-Constrain-Conflict	0x0503 Die Ausführung des Services ist augenblicklich nicht möglich.
	4 Parameter - inconsistent	0x0504 Der Service enthält inkonsistente Parameter.
	5 Illegal-Parameter	0x0505 Ein Parameter hat einen unzulässigen Wert angenommen.
	0 Other	0x0500 Der Fehler in dieser Klasse lässt sich keinem der vorher angegebenen Error Types zuordnen.
6 Access		Diese Fehlerklasse wird bei fehlerhaftem Zugriff gemeldet. Dies betrifft nur das eigentliche Objekt.
	1 Object-invalidated	0x0601 Ein Zugriff bezieht sich auf ein definiertes Objekt, das ein undefiniertes Referenzattribut hat. Das stellt einen permanenten Fehler für den Zugriff auf dieses Objekt dar.

	2 Hardware-Fault	0x0602 Der Zugriff auf das Objekt ist aufgrund eines Hardware-Fehlers fehlgeschlagen. Genauere Angabe des Grundes im Additional Code (glob.= 8 u. 9)
	3 Object-Access-Denied	0x0603 Die Zugriffsrechte des Client sind nicht ausreichend.
	4 Invalid-Address	0x0604 Der Zugriff erfolgt auf eine ungültige interne Adresse.
	5 Object-Attribute-Inconsistent	0x0605 Das Attribut hat einen unzulässigen Wert angenommen. Genauere Angabe des Grundes im Additional Code (glob.=1) z. B. sollen mehr oder weniger Nutzdatenbytes übertragen werden, als dies für ein bestimmtes Objekt zulässig ist.
	6 Object-Access-Unsupported	0x0606 Das Objekt ist kein Variable-Access-Objekt.
	7 Object-Non-Existent	0x0607 Unter diesem Index/Subindex existiert kein Objekt.
	8 Type-Conflict	0x0608 Die Daten entsprechen nicht dem Datentyp des Objekts.
	A Data-not-yet-available	0x060A Auf die Daten des Objekts kann derzeit (z. B. während einer Umparametrierung) nicht zugegriffen werden.
	0 Other	0x0600 Der Fehler in dieser Klasse lässt sich keinem der vorher angegebenen Error Types zuordnen.
8 Other (Application)	0 Other	0x0800 Der Service wurde nicht ausgeführt. Der Grund ist anwendungs-/herstellerspezifisch und betrifft nur das eigentliche Datum. Genauere Angabe des Grundes findet sich im Additional Code. z. B. kann ein bestimmter Wert eines Objekts in dieser speziellen Applikation nicht zulässig sein.
	1 Device Profile specific	0x0801 Der Service wurde nicht ausgeführt. Der Grund ist geräteprofilspezifisch (0x000F). Genauere Angabe des Grundes im Additional Code Alle in diesem Profil aufgeführten Additional Codes sind profilspezifisch.
	B Busy	0x080B Der Dienst kann von der Slave-Applikation in der üblicherweise erwarteten Zeit nicht beendet werden. Im Additional Code wird die Zeit in ms angegeben, die die Bereitstellung der Daten voraussichtlich noch benötigt. „0xFFFF“ bedeutet, die voraussichtliche Ausführungsdauer ist unbekannt.

6.5.2 Additional Code

Der Additional Code setzt sich aus einem globalen, spezifischen und gerätspezifischen Anteil zusammen. Die Angabe eines globalen oder spezifischen Codes ist optional. Das bedeutet: Wenn keine genaue Angabe des Fehlergrunds durch die Anwendung möglich ist, so wird der Additional Code = 0x0000 ausgegeben.

Der spezifische Code enthält eine genauere Beschreibung des im globalen Code definierten Fehlergrunds. Entspricht der Fehlergrund nicht sinngemäß einem spezifischen Code, dann muss der spezifische Code = 0 angegeben werden.

2 Octets mit üblicherweise folgendem Aufbau:

Bit 15	8	7	4	3	0
gerätspezifisch		globaler Code		spezifischer Code	

Die Werte für die Bits 8 bis 15 (gerätspezifischer Code) sind reserviert und zurzeit mit dem Wert 0 besetzt.

Alle in diesem Profil aufgeführten Additional Codes sind profilspezifisch.

Additional Code [hex]	Bedeutung
0000	keine genauere Angabe des Fehlergrunds
0010	Service-Parameter mit unzulässigem Wert
0011	Subindex nicht vorhanden
0012	Objekt Access ist kein Request
0013	Reserved Service Code
0014	SubSlot nicht unterstützt
0015	Objekt AccessType auf dieses Objekt nicht unterstützt
0016	Objekt Access Request Index für diesen AccessType muss 0x0000 sein
0017	Objekt Access Request Länge für diesen AccessType muss 0 sein
0018	Objektlänge für dieses Objekt passt nicht (siehe auch: Additional Code 001D und 001E)
0019	Objekt ist ReadOnly und kann nicht überschrieben werden. Error Class = 0x06; Error Code = 0x03
001A	Objekt ist WriteOnly und kann nicht gelesen werden Error Class = 0x06; Error Code = 0x03
001B	Schreib-/Lese-Dienst für dieses Objekt nicht unterstützt Error Class = 0x05; Error Code = 0x05
001C	Upload-Read oder Download-Write aufgrund der Objektlänge für den Zugriff auf das Objekt erforderlich. Error Class = 0x05; Error Code = 0x02
001D	Objektlänge für dieses Objekt passt nicht (0018) – Es wurden zu viele Daten übertragen Error Class = 0x06; Error Code = 0x05
001E	Objektlänge für dieses Objekt passt nicht (0018) – Es wurden zu wenige Daten übertragen Error Class = 0x06; Error Code = 0x05
0020	Dienst kann momentan nicht ausgeführt werden
0021	Dienst kann momentan nicht ausgeführt werden, weil die Kontrolle/Steuerung des Geräts momentan lokal erfolgt.
0022	Dienst kann im momentanen Gerätezustand (Gerätesteuerung) nicht ausgeführt werden.
0023	Dienst kann momentan nicht ausgeführt werden, weil kein Objektverzeichnis vorhanden ist
0024	Index nicht vorhanden
0030	Wertebereich eines Parameters ist verlassen (der Server kann den Wert nicht liefern) Error Class = 0x08; Error Code = 0x01
0031	Wert des Parameters zu groß Error Class = 0x08; Error Code = 0x01
0032	Wert des Parameters zu klein Error Class = 0x08; Error Code = 0x01

0040	Dependency Ignored Kollision mit anderen Werten, Abhängigkeit nicht beachtet
0041	Kommunikationsobjekt kann nicht auf Prozessdaten abgebildet werden
0042	Prozessdatenlänge überschritten
0050	FW Update: Error Class = 0x08; Error Code = 0x01 Firmware passt nicht zum Gerät – Firmware konnte vom Gerät nicht verarbeitet werden (allgemein, keine genauere Angabe). Siehe auch Kapitel „Firmware-Update“
0051	FW Update: Upload-Read oder Download-Write aufgrund der Objektlänge für den Zugriff auf das Objekt erforderlich. FW Header oder Update-Version passt nicht
0052	FW Update: Error Class = 0x08; Error Code = 0x01 Min FW-Version unterschritten – Firmware passt nicht zum Gerät (z. B. Hardware zu alt)
0053	FW Update: Error Class = 0x08; Error Code = 0x01 Signalisiert das optionale Überspringen des Downloads eines FW-Update-Blocks an das Gerät
0080	Hardware-Fehler
0081	Anwendung ausgefallen
0082	Hardware ist temporär gestört
00A0	ungültige Segmentnummer z. B. Upload ohne Einleitung mit Subindex == 0xFF
00A1	Ressource nicht verfügbar es sind keine Ressourcen (Speicher) für den Download mehr vorhanden
00A2	falscher CRC
00A3	Fehler beim Öffnen der Datei (wenn File-System vorhanden)
00A4	Fehler beim Schreiben der Datei (wenn File-System vorhanden)
00A5	Fehler beim Schließen der Datei (wenn File-System vorhanden)
00A6	Segment fehlt Es wurden weniger Datenblöcke empfangen als im letzten Segment angegeben.
00A7	Segment zu viel Es wurden mehr Datenblöcke empfangen als im letzten Segment angegeben.
00A8	Fehler beim Lesen der Datei (wenn File-System vorhanden)
00A9	ungültige Segmentnummer (Segment doppelt, Segment wurde ignoriert)
00B1	Das Passwort kann nicht ersetzt (gelöscht) werden.
00B2	Das Passwort kann nicht hinzugefügt werden (zu viele Passworte).
00B3	Das Passwort kann für die gewünschte Art des Zugriffs nicht vergeben werden.

Nicht aufgeführte Codes sind reserviert.

Sollte sich ein Hersteller mit einer Fehlermeldung nicht in dieses Schema einordnen können, wird er gebeten, sich mit dem INTERBUS Club in Verbindung zu setzen.

7 Standardobjekte

Die folgenden Objekte werden in der Form beschrieben:

Index (hex)	Eindeutiges Kennzeichen des Objekts Der Index wird für die Dienste zum Zugriff auf das Objekt genutzt.
.SubIndex (hex)	Zusätzliches eindeutiges Kennzeichen des Subobjekts Der Subindex wird für die Dienste zum Zugriff auf das Subobjekt genutzt.*)
Objektname	Name des Objekts
R/W	Zugriffsbeschränkung s. u.
Länge	Länge des Objekts in Byte
Datentyp (Darstellung)	Datentyp des Objekts, s. o. Form, in der der Wert bei einer Anzeige dargestellt wird.
Bedeutung	Erklärung zum Objektinhalt
M/D/O	Implementierungsanweisung M = Mandatory Dieses Objekt muss implementiert werden. O = Optional Dieses Objekt muss genau so implementiert werden, wenn das Gerät die entsprechende (optionale) Funktionalität unterstützt. D = Dependent Ob dieses Objekt implementiert werden muss, hängt davon ab, ob ein anderes, zugehöriges Objekt implementiert wurde oder eine andere genannte Bedingung erfüllt ist. Für Sub-Objekte gilt pauschal, dass diese gemeinsam mit dem Objekt implementiert werden müssen. Die jeweils hinreichende Bedingung ist unter „Dependency Info“ definiert.

*) Der separate Zugriff über ein Subindex größer „0“ ist optional, selbst dann, wenn das Subobjekt selbst mandatory ist.

Zugriffsbeschränkungen

R	Read only, dieses Objekt kann nur gelesen werden
W	Write only, dieses Objekt kann nur geschrieben werden
R/W	Read/Write, dieses Objekt kann gelesen und geschrieben werden
R/W _D	Read/Write, dieses Objekt kann gelesen und unter bestimmten Bedingungen auch geschrieben werden
UR	Upload-Read, dieses Objekt kann mit dem Dienst „Upload-Read“ gelesen werden. Der Standarddienst „Read“ kann nur dann erfolgreich ausgeführt werden, wenn die Anzahl der Daten < PDU-Size ist. Andernfalls wird der Error Type: Error Class: 05, Error Code: 02, AddCode 0x0018 (Service-PDU-Size - Objektlänge für dieses Objekt passt nicht) erzeugt*)
DW	Download-Write, dieses Objekt kann mit dem Dienst Download-Write geschrieben werden. Der Standarddienst „Write“ kann immer ausgeführt werden, da die Anzahl der Daten < PDU-Size ist.*)

*) siehe dazu auch Kapitel „Objekt Domain-Variable“

Um den einheitlichen Zugriff durch Tools zu gewährleisten, müssen die Objekte, wenn sie existieren, generell formal genauso implementiert werden, wie sie im Folgenden definiert sind. Der Gerätehersteller kann den Inhalt jedoch nach seinen Anforderungen anpassen. Z. B. könnte ein aus technischen Gründen nicht durch die Geräte-Firmware bestimmbarer Hardware-Stand so dargestellt werden:

000B	HardwareVersion	R	Max 51 Byte	Record (2 Elemente)	Hardware-Version (Gerät oder Kommunikationsmodul)	M
.1	• BuildDate	R	10+1 Byte	Visible String	0000-00-00	M
.2	• Version	R	max. 39+1 Byte	Visible String	Not available	M

Wenn nur einige Bits in einem Bit-String oder nur einige Bits oder Bytes in einem Octet String eine „Funktion“ haben, so sind reservierte Bits und Bytes mit „0“ definiert. Werden Werte ungleich „0“ geschrieben, ist eine Error Type zu erzeugen, um eine Inkompatibilität erkennen zu können.

7.1 Identifizierung

Die in den folgenden Objekten festgelegten Informationen beschreiben das Gerät selbst, den Hersteller und den Einsatzfall des Geräts. Bei Auslieferung müssen die eingetragenen Daten mit dem Aufdruck auf dem Gerät übereinstimmen.

Index (hex)	Objektname	R/W	Länge	Datentyp (Darstellung)	Bedeutung	M/O
0001	VendorName	R	max. 57+1 Byte	Visible String (Text)	Herstellername z. B. Phoenix Contact GmbH & Co KG	M
0002	VendorID	R	6+1 Byte	Visible String (Text)	Herstellerkennung Eindeutige Herstellerkennung gemäß Organizationally Unique Identifiers (OUI): http://standards.ieee.org/regauth/oui/index.shtml z. B. "00A045"	O
0003	VendorText	R	max. 57+1 Byte	Visible String (Text)	Herstellertext Bemerkung zum Hersteller z. B. Adresse, Branche etc.	O
0012	VendorURL	R	max. 57+1 Byte	Visible String (Text)	Hersteller-URL URL des Herstellers zur Homepage, Produktpage, Promotionpage etc. z. B. www.phoenixcontact.com	O
0007	ProductName	R	max. 57+1 Byte	Visible String (Text)	Produktname Herstellerspezifische eindeutige Produktbezeichnung z. B. IB IL 24 DO 8	M
0008	SerialNo (ProductId)	R	max. 57+1 Byte	Visible String (Text)	Seriennummer Herstellerspezifische eineindeutige Geräteerkennung z. B. 123456789	O
0009	ProductText	R	max. 57+1 Byte	Visible String (Text)	Produkttext Herstellerspezifischer Produkttext z. B. digitales Ausgabemodul	O
000A	OrderNumber	R	max. 57+1 Byte	Visible String (Text)	Artikel-Nr. Herstellerspezifische eindeutige Produkttypkennung z. B. Artikel-Nr. z. B. 2726269	M

0037	DeviceType	R	8 Byte	Octet String[8] (Hex)	<p>Gerätetyp Herstellerspezifische Geräte-/Modulidentifikation ungleich „0“ Die herstellerspezifische Geräte-/Modulidentifikation dient zusammen mit der VendorID dazu, gleichartige Geräte innerhalb einer Konfiguration austauschen und betreiben zu können. Z. B. kann ein 16-kanaliges Ausgabemodul mit Schraubanschlusstechnik durch eins mit Zugfederanschlusstechnik ersetzt werden, obwohl es nicht die gleiche Artikelnummer besitzt. Eine andere Funktionalität erfordert auch einen anderen DeviceType.</p> <p>Wenn dieses Objekt in einem modularen Gerät benutzt wird, dann kann es auch einen Platzhalter kennzeichnen. „0“ wenn Modul (noch) nicht gesteckt ist. In diesem Fall kann trotzdem eine Abbildung auf die Prozessdaten erfolgen.</p>	M
0032	FieldBus_ID	R	3 Byte	Record (2 Elemente)	Feldbusidentifikation	O
.01	<ul style="list-style-type: none"> ID-Code 	R	1 Byte	UINT8 (Dez)	ID-Code Feldbusspezifischer ID-Code (8 Bit, typ. dargestellt als dezimale Zahl)	O
.02	<ul style="list-style-type: none"> PDLength 	R	2 Byte	UINT16 (Dez)	Prozessdatenlänge Anzahl der Prozessdatenbits (Einheit Bit)	O
000B	HardwareVersion	R	Max 51 Byte	Record (2 Elemente)	<p>Hardware-Version (Gerät oder Kommunikationsmodul) <i>Bei Auslieferung müssen die eingetragenen Daten mit dem Aufdruck auf dem Gerät übereinstimmen.</i></p>	M
.01	<ul style="list-style-type: none"> BuildDate 	R	10+1 Byte	Visible String (Text)	<p>Herstelldatum Datum der Version Format YYYY-MM-DD Gemäß ISO 8601 z. B. 2002-11-29</p>	M
.02	<ul style="list-style-type: none"> Version 	R	max. 39+1 Byte	Visible String (Text)	<p>Versionskennung z. B. 4.01Beta Kunde</p>	M
000 C	FirmwareVersion	R	Max 51 Byte	Record (2 Elemente)	<p>Firmware-Version Von außen änderbare (ggf. auch updatebare) Bestandteile eines Geräts oder Kommunikationsmoduls wie z. B. Firmware Controller A, Firmware Controller B, Bootloader, FPGA Image etc. Für all diese Bestandteile gibt es nur einen einzigen anwenderrelevanten Eintrag, den der Hersteller festlegt. Herstellerspezifische Objekte, die die einzelnen Komponenten genauer bezeichnen, sind zulässig. <i>Bei Auslieferung müssen die eingetragenen Daten mit dem Aufdruck auf dem Gerät übereinstimmen. Siehe auch Begriffsdefinitionen</i></p>	M
.01	<ul style="list-style-type: none"> BuildDate 	R	10+1 Byte	Visible String (Text)	<p>Herstelldatum Datum der Version Format YYYY-MM-DD Gemäß ISO 8601 z. B. 2002-05-03 Wenn keine FW vorhanden ist, wird „0000-00-00“ eingetragen</p>	M

.02	• Version	R	max. 39+1 Byte	Visible String (Text)	Versionskennung z. B. 1.03-Kunde Wenn keine FW vorhanden ist, wird „--“ eingetragen	M
000 D	PChVersion	R	Max 51 Byte	Record (2 Elemente)	Parameter Channel Version Parameterkanalversion Implementierungsversion des Parameterkanals	M
.01	• BuildDate	R	10+1 Byte	Visible String (Text)	Herstelldatum Datum der Version Format YYYY-MM-DD Gemäß ISO 8601 z. B. 2016-12-01 Wenn kein Parameterkanal vorhanden ist, wird „0000-00-00“ eingetragen	M
.02	• Version	R	max. 39+1 Byte	Visible String (Text)	Versionskennung z. B. "PCP-Compact V1.10" Wenn kein Parameterkanal vorhanden ist, wird „--“ eingetragen	M
0005	Capabilities	R	N x 8 Bytes	Array of Visible Strings [8] (Text)	Geräteeigenschaften Eigenschaften/Funktionalitäten, über die das Gerät, neben den Basisfunktionen, verfügt. Alle im Folgenden aufgeführten Eigenschaften/Funktionalitäten sind in diesem Basisprofil definiert.	M
					<ul style="list-style-type: none"> • „Nothing“ Keine Zusatzfunktionen. Der Eintrag kann als Platzhalter auch mehrfach vorhanden sein. • „Safety0“ Der Slave unterstützt eine sichere Datenkommunikation. Diese erfolgt immer in beide Richtungen. • „Energy0“ Der Slave unterstützt das Energiemanagement. • „ChPDWh0“ Der Slave unterstützt das Ändern der Prozessdatenbreite. • „SubMa_0“ Der Slave ist ein Sub-Bus-Master. Es gibt mindestens ein weiteres Subsystem unter diesem Slave. • „FwUpdt0“ Der Slave unterstützt das Firmware-Update. <p>Der Type ist genau 8 Zeichen lang. Nicht benutzte Zeichen sind mit 0x00 aufzufüllen. Wenn der Slave das Objekt „Capabilities“ unterstützt, aber keine weitere Zusatzfunktion, ist mindestens einmal der Eintrag „Nothing“ vorhanden.</p>	
0006	ProductFamily	R	max. 57+1 Byte	Visible String (Text)	Produktfamilie Herstellerspezifische Produktfamilie z. B. Inline	O
0004	DeviceFamily	R	max. 57+1 Byte	Visible String (Text)	Gerätefamilie (z. B. I/O, Antrieb) lt. Spezifikation s. u., z. B. motion control/frequency inverters Dieses Objekt kann mehrere Einträge enthalten, die durch Komma und ein folgendes Leerzeichen zu trennen sind.	M
000E	CommProfile	R	Max. 4+1 Byte	Visible String (Text)	Kommunikationsprofil Systemspezifische Profilkennung s. u. z. B. 634	M

000F	DeviceProfile	R	Max. 4+1 Byte	Visible String (Text)	Geräteprofil Kennzeichnung des Applikationsprofils, nach dem das Gerät spezifiziert ist s. u. z. B. „10“ für das Basisprofil oder „22“ für das Drivecom-Profil	M
0011	ProfileVersion	R	Max 51 Byte	Record (2 Elemente)	Profilversion Versionsbezeichnung dieses Profils	M
.01	• BuildDate	R	10+1 Byte	Visible String (Text)	Herstelldatum „2018-04-19“	M
.02	• Version	R	max. 39+1 Byte	Visible String (Text)	Versionskennung Basisprofil V3.0	M
0013	OnBoardDeviceDesc FileName	R	max. 57+1 Byte	Visible String (Text)	Name der OnBoard Gerätebeschreibungsdatei Dateiname der internen (onBoard) Gerätebeschreibungsdatei, z. B. XYZ.xml Existiert keine onBoard-Geräte- beschreibungsdatei, existiert auch dieses Objekt nicht.	O
003A	VersionCount	R	8 Byte	Array of UINT16 (Hex) (4 Elemente)	Versionszähler Eindeutige, aufsteigende Nummerierung als Ganzzahl der Version der entsprechenden Objekte (Komponenten) mit den Indizes: <ul style="list-style-type: none"> • 0x0011 „ProfileVersion“ • 0x000D „PChVersion“ • 0x000B „HardwareVersion“ • 0x000C „FirmwareVersion“ Wenn die entsprechende Komponente geändert wird, muss der Wert in dem zugehörigen Objekt (s. o.) angepasst werden. In diesem Fall muss der dazugehörige VersionCount. Subindex um mindestens 1 erhöht werden. Es ist zulässig, dass sich zwischen zwei Versionszählerständen Lücken befinden.	M
.01	• ProfileVersion	R	2 Byte	UINT16 (Hex)	Profilversion 0x0009 für dieses Profil.	M
.02	• PChVersion	R	2 Byte	UINT16 (Hex)	Parameterkanalversion z. B. 0x0002 für die Version „Compact“- Implementierung Wenn ein Gerät keinen Parameterkanal hat, wird 0x0000 eingetragen.	M
.03	• HardwareVersion	R	2 Byte	UINT16 (Hex)	Hardware-Version z. B. 0x0002 für die Version der Hardware	M
.04	• FirmwareVersion	R	2 Byte	UINT16 (Hex)	Firmware-Version z. B. 0x0105 für die Version Firmware Wenn ein Gerät keine Firmware hat, wird 0x0000 eingetragen.	M
0047	AddInfo	R	16 Byte	Record (2 Elemente)	Zusätzliche Informationen In den folgenden Sub-Indizes sind verschiedene zusätzliche Informationen enthalten.	D

.01	<ul style="list-style-type: none"> • SafetyProtType 	R	8 Byte	Octet String[8] (Text)	<p>Safety-Protokolltyp</p> <ul style="list-style-type: none"> • „IBSSafe0“ INTERBUS-Safety • „SBT_0“ SafetyBridge-Protokoll • „PROFI_0“ PROFISafe-Protokoll • „CIP_0“ CIP-Safety-Protokoll • „FSOE_0“ FSoE-Safety-Protokoll • „OpenS_0“ Open-Safety-Protokoll <p>Der Typ ist genau 8 Zeichen lang. Nicht benutzte Zeichen sind mit 0x00 aufzufüllen.</p> <p>Dependency Info: Dieses Objekt implementiert sein, wenn das Gerät ein Safety-Protokoll unterstützt.</p>	D
.02	<ul style="list-style-type: none"> • SafetyProtVers 	R	8 Byte	Octet String[8] (Text)	<p>Safety-Protokollversion</p> <p>Für die Version stehen genau 8 Zeichen zur Verfügung. Nicht benutzte Zeichen sind mit 0x00 aufzufüllen.</p>	D
.03	<ul style="list-style-type: none"> • LegacyInfo 	R/W	8 Byte	Octet String[8] (Text)	<p>Information zu Vorgängerversionen</p> <p>Hier kann der Anwender einen herstellerspezifischen Code eintragen, der das Gerät veranlasst, sich, sofern möglich, wie eine bestimmte (Vorgänger-)Version des Geräts zu verhalten.</p> <p>Beispiel: Ein Gerät mit der Firmware-Version 1.62 kann den Wert „1.50“ eingetragen bekommen und müsste sich ab dann so verhalten, wie ein Gerät mit der Firmware-Version 1.50.</p> <p>Das Rücksetzen alle Parameter auf den Default-Wert ist damit nicht automatisch verbunden.</p> <p>Wenn die vorgegebene Kennung nicht unterstützt wird, so erfolgt eine negative Confirmation mit dem Error Type 0x08, 0x01 und AddCode z. B. 0x0032</p> <p>Dependency Info: Dieses Objekt muss implementiert sein, wenn für das Gerät eine austauschkompatible Vorgängerversion existiert, die über wenigstens ein Objekt/Sub-Objekt <u>nicht</u> verfügt, das in der aktuellen Version implementiert ist.</p>	D
0014	Location	R/W	max. 57+1 Byte	Visible String (Text)	<p>Einbauort</p> <p>Text, der vom Gerätebenutzer in diesem Parameter hinterlegt wurde. Dieser Text kennzeichnet den Einbauort des Geräts und wird in einem nichtflüchtigen Speicher abgelegt.</p> <p>z. B. Maschine 1, hinten links</p> <p>Der Inhalt wird gewöhnlich nur einmalig, während der Inbetriebnahme (Taufe), in das Gerät geschrieben.</p>	O

0015	EquipmentIdent	R/W	max. 57+1 Byte	Visible String (Text)	Betriebsmittelkennzeichen Text, der vom Gerätebenutzer in diesem Parameter hinterlegt wurde. Das Betriebsmittelkennzeichen wird in einem nichtflüchtigen Speicher abgelegt. Hier kann z. B. der Gerätebenutzer eine Beschreibung über die Gerätebenutzung in der Anlage hinterlegen. z. B. M1H747h.l. Der Inhalt wird gewöhnlich nur einmalig, während der Inbetriebnahme (Taufe), in das Gerät geschrieben.	O
0016	ApplDeviceAddr	R/W	2 Byte	UINT16 (Dez)	Applikationsspezifische Geräteadresse (Anwenderdefinierte Gerätenummer) Beliebige Kennung dieses Geräts in dieser speziellen Applikation. Diese Nummer muss nicht eindeutig sein. Die Verwaltung obliegt ausschließlich dem Endanwender. z. B. 123 Der Inhalt wird gewöhnlich nur einmalig, während der Inbetriebnahme (Taufe), in das Gerät geschrieben.	O

7.1.1 Gerätefamilie (DeviceFamily Index 0x0004)

Eindeutige Kennzeichnung der Gerätefamilie.

En	De
Actuator	Aktor
Bus Coupler	Buskoppler
Closed Loop Controller	Regler
Dosing Device	Dosiergerät
Drive	Antrieb
Drive - Frequency Inverter	Antrieb - Frequenzumrichter
Drive - Motor Starter	Antrieb - Motorschalter
Drive - Servo Amplifier	Antrieb - Servoverstärker
Drive - Stepper Motor Controller	Antrieb - Schrittmotor-Steuerungcontroller
Encoder	Encoder
Gateway	Gateway
General	Allgemeines
HMI	HMI
HMI Display	HMI-Anzeige
HMI Operator Panel	HMI-Bediengerät
Hydraulic Device	Hydraulik-Gerät
I/O	E/A
I/O analog IN / OUT	E/A analog IN / OUT
I/O analog IN	E/A analog IN
I/O analog OUT	E/A analog OUT
I/O digital IN / OUT	E/A digital IN / OUT
I/O digital IN	E/A digital IN
I/O digital OUT	E/A digital OUT
I/O Temperature Module	E/A- Temperaturmodul
I/O Function Module	E/A-Funktionsmodul
Identification System	Identifikationssystem
Media Converter active	Medienkonverter aktiv
Media Converter passive	Medienkonverter passiv
NC	NC
NC/RC	NC/RC
PC	PC
PC Board	PC-Karte
PLC	SPS
PLC board	SPS-Karte
Pneumatic Device	Pneumatik-Gerät
Positioning Controller	Positionier-Steuerung
Power Supply	Stromversorgung
Robot Control	Roboter
Safety	Funktionale Sicherheit
Sensor	Sensor
Switching Device	Schaltgerät
Technology Controller	Technologie-Steuerung
Valve	Ventil
Weighing or Batching System	Wiege- oder Dosiersystem
Welding Controller	Schweißsteuerung
Wrenching Controller	Schraubersteuerung
Safety-*)	Safety-*)

*) „Safety-“ kann als Vorsatz für jede Gerätefamilie benutzt werden.

Sollte sich ein Hersteller in dieses Schema nicht einordnen können, wird er gebeten, sich mit dem INTERBUS Club in Verbindung zu setzen.

7.1.2 Kommunikationsprofil (CommProfile Index 0x000E)

Dieser Parameter enthält eine systemspezifische Kommunikationsprofilkennung
Für die „Compact“-Implementierung kommen folgende Kommunikationsprofile in Betracht:

63	Implementierung	Kanaltyp	Geeignet für
632	Compact	Feldbus-Management, Compact-Parameterkanal	Keine zyklischen Prozessdaten, kein Download/Upload von Variablenlisten, Programmen etc.
633	Compact Prozessdaten	Feldbus-Management zyklische Prozessdaten Compact-Parameterkanal	Parametrierbare Geräte, nur Parameter, kein Download/Upload von Variablenlisten und Programmen etc.
634	Compact Prozessdaten Up-/Download- Protokoll	Feldbus-Management, zyklische Prozessdaten Compact-Parameterkanal Download, Upload	Komplexe Geräte, Download/Upload von Variablenlisten
635	Compact Up-/Download- Protokoll	Feldbus-Management, Compact-Parameterkanal Download, Upload	Komplexe Geräte, Download/Upload von Variablenlisten, keine zyklischen Prozessdaten
636	Compact Prozessdaten Explizite Up- /Download- Standarddienste	Feldbus-Management, zyklische Prozessdaten Compact-Parameterkanal Download, Upload als explizite Standarddienste	Komplexe Geräte, Download/Upload von Variablenlisten
637	Compact Explizite Up- /Download- Standarddienste	Feldbus-Management, Compact-Parameterkanal Download, Upload als explizite Standarddienste	Komplexe Geräte, Download/Upload von Variablenlisten, keine zyklischen Prozessdaten

7.1.3 Geräteprofil (DeviceProfile Index 0x000F)

Dieser Parameter enthält Informationen über das implementierte Gerätetypprofil des Geräts.

Aufbau des Parameters:

B15b12	b11b8	b7b4	b3b0
Profilgruppe			Version

DeviceProfile	Bedeutung
0000	Kein Profil
0010	Basisprofil (Grundlage aller anderen Profile)
0012	Sensor/Aktor
0020	DRIVECOM nur Prozessdaten
0021	DRIVECOM Frequenzumrichter
0022	DRIVECOM Servo
0030	Reserve
0040	Anschaltbaugruppen
0050	Reserve
0060	Reserve
0070	Encoder
0080	Prozessregler
0090	Robotersteuerungen
00A0	Schraubersteuerungen
00B0	ISO-Ventile
00C0	Schweißsteuerungen
00D0	Bedien-/Anzeigegeräte
00E0	Hydraulische Geräte
FFFF	mehr als ein Geräteprofil
Alle anderen	Reserviert

Unterstützt ein Gerät mehr als ein Geräteprofil, dann wird 0xFFFF im Parameter Geräteprofil eingetragen.

Obwohl hier als Hex-Wert dargestellt, wird der Parameter als Visible String der Länge 4+1 übertragen. Die einzelnen Stellen des Hex-Codes werden dabei als ASCII-Zeichen übertragen.

7.2 Gerätediagnose

Diese Objekte dienen dazu, der Applikation Diagnoseinformationen über den Zustand des Geräts und der evtl. angeschlossenen Peripherie mitzuteilen. Die aktuelle Diagnoseinformation des Geräts wird im Objekt „DiagState“ abgelegt. Außerdem kann diese Information in der Langform in der Domain-Variablen (Index 0xE800) abgelegt werden. Für eine Historie der Diagnoseinformationen ist die Domain-Variable „DiagHistory“ bzw. „DiagHistoryLong“ für die Langform, vorgesehen.

Eine Störung beim Zugriff auf ein Objekt (insbesondere aufgrund des übertragenen Parameterinhalts) wird mit einer negativen Service-Response gemeldet
Eine zusätzliche Meldung über das Objekt „DiagState“ 0x0018 ist nicht vorzusehen.

7.2.1 Objekte

Index (hex)	Objektname	R/W	Länge	Datentyp (Darstellung)	Bedeutung	M/O
0018	DiagState	R	Max. 207 Byte	Record 6 Elemente	Diagnosezustand Aktueller Diagnosezustand des Geräts in der Kurzform Die Änderungsgeschwindigkeit einer Diagnose sollte 1 Sekunden nicht unterschreiten.	M
.01	• Lfd.Nr.	R	2 Byte	UINT16 (Dez)	Lfd.Nr. Eindeutige, fortgezählte Störungsnummer seit dem letzten Power-Up-Reset oder HistoryReset. Die Nummer wird für jede Störung weiter gezählt, nicht für kommend und gehend separat. Ebenfalls wird die Störungsnummer auch für die Störungen hoch gezählt, die möglicherweise wegen ihrer Priorität nicht gemeldet werden konnten.	M
.02	• Priority	R	1 Byte	UINT8 (Hex)	Priorität Priorität der Störung Siehe „Klassifizierung von Meldungen“	M
.03	• Channel	R	1 Byte	UINT8 (Dez)	Kanal Kanal, an dem die Störung aufgetreten ist. „0xFF“ betrifft das ganze Gerät. Zusätzliche Informationen siehe „MoreFollows“	M
.04	• Code	R	2 Byte	UINT16 (Hex)	Störungscode s. u.	M
.05	• MoreFollows	R	1 Byte	Bit-String 8 (Bin)	Zusatzinformationen Information zur Interpretation der folgenden Daten	M
					= „0x00“ Keine weitere Informationen Bit0 = 1: Es gibt weitere Informationen zu diesem Fehler, die über das Objekt „DiagStateLong“ auslesbar sind. Bit1 : reserviert (=0) Bit2 : reserviert (=0) Bit3 : reserviert (=0) Bit4 = 1: Es liegen noch weitere, gleichzeitig aufgetretene Diagnoseereignisse vor, die über das Objekt 0xE806 „ComplDiagState“ ausgelesen werden können. Bit 5-6: reserviert (=0) Bit7 = 1: Kennzeichen für diese erweiterte Version des Objekts 0x0018	
.06	• Reserved	R	2 Byte	Octet String[2] (Hex)	Reserviert Dieser Eintrag ist mit 0x00.00 zu belegen	M

.07	• SubModNo	R	1 Byte	UINT8 (Dez)	Sub-Modul-Nummer Wenn es sich um ein modulares Gerät handelt, wird hier das betreffende Sub-Modul angegeben. Siehe Kapitel „Modulare Geräte - Subsysteme“ Wenn es sich nicht um ein modulares Gerät handelt, wird hier „0“ eingetragen.	M
.08	• FunctionGroup	R	8 Byte	Octet String[8] (Text)	Funktionsgruppe Funktionsgruppe, die eine Diagnose meldet Der Type ist genau 8 Zeichen lang. Nicht benutzte Zeichen sind mit 0x00 aufzufüllen. Die zulässigen Funktionsgruppen sind in den Objekten 0x003B.1 und 0x003C.1 aufgeführt. Wenn es mehrere Gruppen eines Typs gibt (z. B. je 4 Kanäle eines 16-kanaligen DO-Moduls), so wird die entsprechende G-Nummer angehängt. Z. B: Ein DI 32 besteht aus 4 Gruppen: DI-G1, DI-G4 Die herstellereigene Bezeichnung („Relais OUT“) wird im Diagnosetext (0x0018.11) angegeben. <u>Anmerkung:</u> „DIAG“, „CMD“, „STATUS“ etc. sind üblicherweise keine sinnvollen Funktionsgruppen	M
.09	• AddValue	R	4 Byte	Octet String[4] (Hex)	Zusatzinformation „Weiterer Wert“ zum aktuellen Diagnosezustand des Geräts. Die Nutzung ist dem Anwender in gewissen Grenzen frei gestellt: Bei Subsystemen werden hier die Subsystem-spezifischen Diagnosecodes eingetragen. Hersteller können hier auch die Legacy Codes für ihre Diagnosen eintragen, wenn es die gibt. Oder es kann ein Begleitwertwert zu einer Diagnose, wie z. B: „aktuelle Temperatur bei Grenzwertüberschreitung“, eingetragen werden. Oft lehnt man sich auch an die im PROFINET I&M Profil beschriebene Nutzung an. Legacy-Information: ehemaliges Objekt 0x0034	M
.0A	• TextLength	R	1 Byte	UINT8 (Dez)	Textlänge Länge des folgenden Diagnosetextes in Byte	M

.0B	<ul style="list-style-type: none"> Text 	R	Max 99+1 Byte	Visible String (Text)	<p>Diagnosetext</p> <p>Die aufgetretene Störung soll hier für den Anlagenbetreiber gerätspezifisch näher erläutert werden. Neben der Art des Fehlers sollte auch die betroffene Funktionsgruppe und der Kanal angegeben werden. Des Weiteren kann, wenn es einen Bezug dazu gibt, der Klemmpunkt angegeben werden. Zum Schluss sollte immer eine Handlungsoption für den Anwender angegeben werden.</p> <p><i>Beispiel:</i> „Ovrange at AI Channel 3 on Terminal point 03, 13. Check sensor signal.“</p> <p>Der Text bleibt ein Vorschlag, der nicht für eine generische Interpretation vorgesehen ist.</p> <p>Der String ist 0x00 terminiert. Default: „Status OK“</p>	M
E806	ComplDiagState	R		Array of Records	<p>Vollständiger aktueller Diagnosezustand Diagnoseinformationen des Geräts in der Kurzform für <u>alle</u> aktuell noch anstehenden Meldungen.</p> <p>Wenn das Objekt über Sub-Indizes angesprochen wird, so sind diese lückenlos aufsteigend belegt.</p>	O
.01	<ul style="list-style-type: none"> DiagState1 				Ältester Diagnosezustand in der Kurzform - Objekt 0x0018	D
	D
.N	<ul style="list-style-type: none"> DiagStateN 				Neuester Diagnosezustand in der Kurzform - Objekt 0x0018	D
E800	DiagStateLong	R		Domain- Variable Record	Diagnosezustand (Langform) Aktuelle Diagnoseinformationen des Geräts in der Langform	O
Seg.1 -a	<ul style="list-style-type: none"> Lfd.Nr. 				- Objekt 0x0018.1	D
-b	<ul style="list-style-type: none"> Priority 				- Objekt 0x0018.2	D
-c	<ul style="list-style-type: none"> Channel 				- Objekt 0x0018.3	D
-d	<ul style="list-style-type: none"> Code 				- Objekt 0x0018.4	D
-e	<ul style="list-style-type: none"> MoreFollows 				- Objekt 0x0018.5	D
-f	<ul style="list-style-type: none"> Reserved 				- Objekt 0x0018.6	D
-g	<ul style="list-style-type: none"> SubModNo 				- Objekt 0x0018.7	D
-h	<ul style="list-style-type: none"> FunctionGroup 				- Objekt 0x0018.8	D
-i	<ul style="list-style-type: none"> AddValue 				- Objekt 0x0018.9	D
-j	<ul style="list-style-type: none"> Date 	R	10+1 Byte	Visible String (Text)	<p>Datum, an dem die Störung aufgetreten ist, im Format YYYY-MM-DD. „0000-00-00“ = kein Datum vorhanden</p>	D

-k	• Time	R	8+1 Byte	Visible String (Text)	Uhrzeit, zu der die Störung aufgetreten ist, im Format hh:mm:ss. „00:00:00“ = keine Uhrzeit vorhanden	D
-l	• TimeofOperation	R	4 Byte	UINT32 (Dez)	Betriebsstundenzählerstand absolut, zu der die Störung aufgetreten ist. 0 ... 4294967295 Sekunden rund 136 Jahre = „0“ kein Betriebsstundenzähler vorhanden	D
-m	• ParamSet	R	2 Byte	UINT16 (Dez)	Aktuell gültige Parametersatznummer, bei der die Störung aufgetreten ist. „0000“ = keine spezielle Parametersatznummer vorhanden	D
-n	• TraceData	R	2 Byte	UINT16 (Dez)	Startindex evtl. zur Meldung zugehöriger Trace-Daten. = „0000“ keine Trace-Daten vorhanden	D
Seg.2 -a	• RecipientLength	R	2 Byte	UINT16 (Dez)	Länge des nachfolgenden Empfängers des Meldungstextes in Zeichen (Byte) = „0x00“ keine Angabe	D
-b	• Recipient	R	Beliebig	Visible String (Text)	Empfänger der Meldung	D
Seg.N -a	• Senderlength	R	2 Byte	UINT16 (Dez)	Länge des nachfolgenden Absenders des Meldungstextes in Zeichen (Byte) = „0x00“ keine Angabe	D
-b	• Sender	R	Beliebig	Visible String (Text)	Absender der Meldung	D
Seg. M -a	• TextLength				- Objekt 0x0018.10	D
-b	• Text				- Objekt 0x0018.11	D
E801	DiagHistory	R		Domain- Variable Record	Diagnosearchiv Diagnoseinformationen des Geräts in der Kurzform mit Historie (die älteste Information/niedrigste Störungsnummer wird zuerst übertragen) Die Struktur ist dieselbe wie die von 0x0018 DiagState	O
	• DiagState1				Ältester Diagnosezustand in der Kurzform	D
	D
	• DiagStateN				Neuester Diagnosezustand in der Kurzform	D
E802	DiagHistoryLong	R		Domain- Variable Record	Diagnosearchiv (Langform) Diagnoseinformationen des Geräts in der Langform mit Historie (die älteste Information/niedrigste Störungsnummer wird zuerst übertragen) Die Struktur ist dieselbe wie die von DiagStateLong	O
	• DiagStateLong1				Ältester Diagnosezustand in der Langform	D
	...					D
	• DiagStateLongN				Neuester Diagnosezustand in der Langform	D

0019	ResetDiag	R/W	1 Byte	UINT8 (Hex)	Diagnosemeldungen quittieren Löscht den entsprechenden Diagnosespeicher des Geräts und quittiert die Diagnosemeldung(en)	O	
		<p>= „0x00“ keine Beschränkung bei den Diagnosemeldungen, bzw. hebt vorhandene (0x06, 0x07) wieder auf</p> <p>= „0x01“ Löscht die Diagnosehistorie. Dieser Wert wird nur unterstützt, wenn das Objekt 0xE801 „DiagHistory“ bzw. 0xE802 DiagHistoryLong“ existiert.</p> <p>= „0x02“ Löscht (und quittiert), wenn möglich, alle gemeldeten und die anstehenden, noch nicht gemeldeten DiagStates (Störungen), sofern sie behoben wurden und nicht mehr aktiv sind. Das betrifft die Objekte: 0x0018, 0x0033, 0x0034, 0xE800, 0xE806.</p> <p>= „0x03“ Löscht (und quittiert) die gesamte Diagnose. Kombination aus Parameter „0x01“ und „0x02“</p> <p>= „0x04“ Löscht (und quittiert) alle schon per Information Report gemeldeten DiagStates, wenn mehrere gemeldet wurden.</p> <p>= „0x05“ Löscht (und quittiert) ausschließlich den letzten (noch anstehenden) gemeldeten DiagState. Das betrifft die Objekte: 0x0018, 0x0033, 0x0034, 0xE800 und mittelbar das Objekt 0xE806.</p> <p>= „0x06“ Löscht (und quittiert) alle Störungen (ähnlich wie „0x02“) aber hier auch dann, wenn die Ursache noch besteht und lässt keine neuen Diagnosemeldungen zu.</p> <p>= „0x07“ Löscht (und quittiert) alles und lässt keine neuen Diagnosemeldungen, auch nicht in der History, zu. Kombination aus Parameter „0x01“ und „0x06“ Die Diagnose ist damit abgeschaltet.</p> <p>sonst: reserviert.</p> <p>Nach erfolgter Aktion wird der Inhalt selbsttätig auf 0x00 zurückgesetzt. Ausnahme 0x06 und 0x07. Diese bleiben anstehen, bis ein anderer Wert geschrieben wird. Dadurch kann man erkennen, dass die Diagnosefunktion deaktiviert ist.</p> <p>Wenn die Ursachen der Störung nicht beseitigt ist und mit 0x0019 „ResetDiag“ die Störung auch nicht beseitigt werden kann, so kann diese auch nicht gelöscht (quittiert) werden (Ausnahme Datum 0x06 und 0x07). In diesem Fall ist eine negative Response mit dem ErrorType 0x08, 0x01, 0x0022 (Error Class 0x08, Error Code 0x01, Additional Code 0x0022 „Dienst kann im momentanen Gerätezustand nicht ausgeführt werden. Die Störung ist nicht beseitigt.“) zu senden.</p> <p>Wenn die durch den entsprechenden Parameter betroffenen Objekte, alle oder z.T. nicht implementiert sind, so wird der Dienst dennoch positiv quittiert.</p> <p>Wenn mit dem Objekt 0x0019 quittiert wird, obwohl keine Diagnosemeldung ansteht, so ist dennoch eine positive Response zu senden.</p>					
001A	GetErrorRepMethod	R/W	1 Byte	Bit-String 8 (Hex)	Meldemethode für Störung Legt die Methode fest, mit der eine Störung an das überlagerte System gemeldet werden soll. Das Setzen des entsprechenden Bits aktiviert die Methode	O	

		<p>Bit 0 Schaltet das Erzeugen eines Information Reports bei aufgetretener Störung mit dem Inhalt des Objekts „DiagState“ ein (=“1“) / aus (=“0“).</p> <p>Bit 1 Schaltet das Erzeugen einer expliziten Peripheriefehler-Statusmeldung bei aufgetretener Störung mit dem Inhalt des Objekts „DiagState“ ein (=“1“) / aus (=“0“).</p> <p>Bit 2 Schaltet das Erzeugen einer Prozessdaten-Statusmeldung (vom Hersteller beliebig zu wählendes Prozessdatenbit) bei aufgetretener Störung mit dem Inhalt des Objekts „DiagState“ ein (=“1“) / aus (=“0“).</p> <p>Wenn mehrere Meldemethoden gleichzeitig gewählt werden, muss die Diagnosemeldung (und die folgenden Diagnosemeldungen) im Objekt 0x0018 solange vorgehalten werden, bis sie ausgelesen wurde(n), selbst wenn ein oder mehrere andere Diagnosemeldungen bereits über den Reportmechanismus gesendet wurde(n).</p>				
0033	DiagStateChannelNo	R	3 Byte	Record (2 Elemente)	<p>Dieses Objekt soll in Zukunft nicht mehr verwendet werden. Es wurde vollständig in das Objekt 0x0018 „DiagState“ integriert.</p> <p>Zusatzinformation „Kanalnummer“ zum aktuellen Diagnosezustand des Geräts</p>	X
.01	• Lfd.Nr.	R	2 Byte	UINT16 (Dez)	- Objekt 0x0018.1	X
.02	• ChannelNo	R	1 Byte	UINT8 (Dez)	<p>Hat das Gerät zu Gruppen zusammengefasste Kanäle, kann hier der betroffene Kanal der entsprechenden Gruppe eingetragen werden.</p> <p>Wenn „DiagState“ gelesen wird, so ist dieses Objekt wenigstens für den nächsten Lesezugriff konsistent zu „DiagState“ zu halten.</p> <p>Wenn „DiagState“ als Information Report gemeldet wird und dieses Objekt dazu relevant ist, wird es ebenfalls über einen Information Report gemeldet. Zeitlich nach dem DiagState.</p>	X
0034	DiagStateAddValue	R	6 Byte	Record (2 Elemente)	<p>Dieses Objekt soll in Zukunft nicht mehr verwendet werden. Es wurde vollständig in das Objekt 0x0018 „DiagState“ integriert.</p> <p>Zusatzinformation „weiterer Wert“ zum aktuellen Diagnosezustand des Geräts</p>	X
.01	• Lfd.Nr.	R	2 Byte	UINT16 (Dez)	- Objekt 0x0018.1	X

.02	• AddValue	R	4 Byte	UINT32 (Hex)	<p>Hier kann ein Begleitwertwert zu einer Kanaldiagnose wie z.B: „aktuelle Temperatur bei Grenzwertüberschreitung“ eingetragen werden. Details: siehe PROFINET I&M Profil</p> <p>Wenn „DiagState“ gepollt wird, so ist dieses Objekt wenigstens für den nächsten Lesezugriff konsistent zu „DiagState“ zu halten.</p> <p>Wenn „DiagState“ als Information Report gemeldet wird und dieses Objekt dazu relevant ist, wird es ebenfalls über einen Information Report gemeldet. Zeitl. nach dem DiagState.</p>	X
001B	TestMode	R/W	2 Byte	UINT16 (Hex)	<p>Testbetrieb</p> <p>Mit einem herstellerspezifischen Code wird in einen Testmode geschaltet. Hierdurch werden andere Testparameter aktiviert. Mit dem Wert 0 wird wieder in den normalen Betriebsmode zurückgeschaltet = „0x0000“ normaler Betriebsmode (Default)</p>	O

Die Tiefe der Diagnosehistorie, d. h. die Anzahl der Einträge in die Diagnosearchive, ist vom Geräteentwickler entsprechend seinen Anforderungen und Ressourcen frei zu wählen.

7.2.1.1 Timing von Diagnosemeldungen

Diagnosemeldungen werden in den meisten Systemen (überlagerte Netzwerke, Steuerungen etc.) mit geringerer Priorität als die Prozessdaten behandelt. Um sicher zu stellen, dass Diagnosemeldungen Ihren Adressaten auch erreichen können und der dafür zur Verfügung gestellte Kanal nicht überlastet wird, muss eine Diagnosemeldung hinreichend lange anstehen, bevor sie als gehend gemeldet wird.

Aus diesem Grund wird eine einzuhaltende Zeit zwischen einer kommenden und einer gehenden Meldung von 1 Sekunde festgelegt.

7.2.1.2 Benachrichtigung über Auslesen (Read) des Objekts DiagState

Eine vorhandene Diagnoseinformation wird (z. B. über einen Sammelmodulfehler-Eingang (/StatErr) des Feldbus-Slave-Protokollbausteins) je nach Priorität, gemeldet. Es können auch mehrere Diagnoseinformationen gleichzeitig vorhanden sein. Der Master oder ein Software-Tool kann sich die entsprechenden Informationen über den Parameterkanal abholen.

Eine Meldung per Statusbit (z. B. Bit7 im Byte 0 des Prozessdatenkanals) kann nicht generisch bearbeitet werden und sollte deshalb vermieden werden.

Die aktuelle Diagnoseinformation in DiagState(Long) bleibt so lange bestehen, bis

- diese Information wenigstens einmal ausgelesen wurde und
- die Ursache für diese Diagnoseinformation nicht mehr vorhanden ist

oder

- sie durch eine höher priore Meldung überschrieben wird

Erst danach wird die nächste gleich- oder niederpriore Information (z. B. „Störung Nr.2“) zur Verfügung gestellt.

Das Statusbit/die Statusbits im Prozessdatenkanal oder der Modulfehleringang bleibt/bleiben so lange anstehen, bis

- jede Diagnoseinformation wenigstens einmal ausgelesen wurde und
- keine Diagnoseinformation mehr vorhanden ist.

Erst danach wird die dann aktuelle Information „Status OK“ zur Verfügung gestellt.

7.2.1.3 Benachrichtigung über Melden (Info Report) des Objekts „DiagState“

Alternativ kann das Melden von Störungen durch den Dienst „Information-Report“ (einmalig, nur das Objekt „DiagState“) dem Master gemeldet werden. Diese Funktionalität wird mit dem Objekt „GetErrorRepMethod“ ein-/ausgeschaltet. Bei aktivierter Funktion „GetErrorRepMethod“ wird jede neue Diagnoseinformation (Inhalt: Objekt 0x0018) einmalig und selbsttätig (ohne Request des Masters) dem Master gemeldet. In diesem Fall muss auf der Feldbus-Masterseite die Auswertung des Information-Reports, z. B. im System, als SPS-Funktionsbaustein oder als Hochsprachenprogramm vorhanden sein.

Die Diagnosebehandlung von Objekt 0x0018 "DiagState" (s. o.) wird dadurch nicht beeinflusst. D. h.:

Die aktuelle Diagnoseinformation in DiagState(Long) bleibt so lange bestehen, bis

- die Ursache für diese Diagnoseinformation nicht mehr vorhanden ist oder
- sie durch eine höher priore Meldung überschrieben wird.

Erst danach wird die nächste gleich- oder niederpriore Information (z. B. „Störung Nr.2“) zur Verfügung gestellt.

Das Statusbit/die Statusbits im Prozessdatenkanal oder der Modulfehleringang bleibt/bleiben so lange anstehen, bis

- keine Störungsinformation mehr vorhanden ist.

Erst danach wird die dann aktuelle Information „Status OK“ zur Verfügung gestellt.

7.2.1.4 Klassifizierung von Störungen (Priorität der Meldungen)

Unter „Störung“ werden alle Zustände einer Geräteapplikation zusammengefasst, die abweichend vom Normalzustand des Geräts sind. Welche Bedeutung dieser Zustand für die Applikation hat, wird mit der Priorität angegeben.

Prio 1 (rot) **Fehler (Alarm)**

Bedeutung: Es liegt eine Störung vor, auf die reagiert werden muss.

Eine Handlung durch den Anwender ist notwendig. Eine Störung verschwindet nicht ohne Zutun des Anwenders.

Beispiel: Ein Fehler führt zu einer Aktivität im Antrieb, aber nicht notwendigerweise zum sofortigen Stillsetzen.

Prio 2 (gelb) **Warnung**

Bedeutung: Gefahr des Auftretens eines Fehlers

Eine Warnung führt nicht zu einer Aktivität im Gerät.

Eine Handlung durch den Anwender ist nicht notwendig. Eine Warnung verschwindet ohne Zutun des Anwenders.

Beispiel: Unter- oder Überschreitung eines Grenzwerts

Prio 83 (grün) **Information**, Nachricht, Mitteilung (nur als gehende Meldung existent)

Beispiel: Allgemeine Betriebsmeldung 10.000 Betriebsstunden gelaufen

Prio 81, 82 **Aufgehoben**

Bedeutung: Die mit derselben Nr. gemeldete Störung ist beseitigt.

Nachrichten und Informationen sind benutzerdefiniert. Warnungen und Fehler können sowohl vordefiniert als auch benutzerdefiniert sein.

Eine gehende Meldung (Diagnosezustand) überschreibt die zugehörige kommende Meldung (Diagnosezustand), falls diese noch nicht vom Master abgeholt wurde. Damit wird das Kommunikationsaufkommen reduziert, ohne dass Informationen verloren gehen.

Daraus ergibt sich folgende Reihenfolge der Prioritäten:

Kodierung	Priorität	Bedeutung
0x81 – höchste Priorität	Prio 1 (rot) gehend	Fehler (Alarm) aufgehoben
0x01	Prio 1 (rot) kommend	Fehler (Alarm)
0x82	Prio 2 (gelb) gehend	Warnung aufgehoben
0x02	Prio 2 (gelb) kommend	Warnung
0x83 – niedrigste Priorität	Prio 3 (grün)	Information, Mitteilung
0x00	-	Keine Störung/Warnung/Information vorhanden

Wenn sich das Gerät im Zustand Störung befindet, so enthält der Parameter 0x0018.2 „DiagState.Priority“ einen Wert ungleich „0“.

Wenn sich das Gerät nicht im Zustand Störung befindet, dann enthält dieser Parameter den Wert „Störung gegangen“ (0x8X) oder „0“ (wenn alle gegangenen Störungen gemeldet wurden und auch keine Störung mehr vorhanden ist).

7.2.1.5 Störungscode

Ziel der Vereinheitlichung der Störungscode ist es, dem Anwender auch ohne genaue Kenntnis des Geräts, eine schnelle und einfache Orientierung für die Problembeseitigung zu geben. Der Störungscode sollte im Diagnosetext durch eine gerätspezifische Handlungsanweisung für den Anwender ergänzt werden. Dabei kann es für einen Störungscode möglicherweise auch mehrere Diagnosetexte (0x0018.6) geben.

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

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

Entsprechend (dreiteilig) ist eine Störungsmeldung auch zu interpretieren.

z.B: 0x5112:

Störung in der Geräte-Hardware (nur innerhalb des Gerätegehäuses), Störungen in der Versorgung im und durch das Gerät, hier Versorgung + 24 V

Wenn sich das Gerät im Zustand Störung befindet, so enthält der Parameter 0x0018.4 „DiagState.Code“ einen Wert ungleich „0“.

Wenn sich das Gerät nicht im Zustand Störung befindet, dann enthält dieser Parameter einen Störungscode mit dem Wert „gehend“ (0x8X) oder „0“ wenn alle gegangenen Störungen gemeldet wurden und keine Störung mehr vorhanden ist.

7.2.1.5.1 Haupt- und Untergruppen

Code (hex) Bedeutung

0000	keine Störung
1000	Störung allgemein
2000	Strom (fließt zu viel)
2100	Störung, die die Versorgung des Geräts oder Signale „geräteeingangseitig“ betrifft
2200	Störung, die interne Netzteile oder Signale „geräteintern“ betrifft
2300	Störung an der Versorgung oder an Signalen für die an das Gerät angeschlossene Peripherie (geräteausgangseitig)
3000	Spannung
3100	Störung, die die Versorgungsspannung des Geräts oder Signale „geräteeingangseitig“ betrifft
3200	Störung, die interne Netzteile oder Signale „geräteintern“ betrifft
3300	Störung an den Signalen für die an das Gerät angeschlossene Peripherie (geräteausgangseitig)
3400	Störung an der Versorgungsspannung für die Peripherie
4000	Temperatur - befindet sich nicht im zugelassenen Bereich
4100	Temperatur Umgebung nicht OK
4200	Temperatur im Gerät nicht OK
4300	Temperatur eines externen Peripheriegeräts (z. B. Antrieb) nicht OK
4400	Temperatur der Versorgungseinheit nicht OK
5000	Störung der Geräte-Hardware (nur innerhalb des Gerätegehäuses)
5100	Störung der Versorgung im Gerät und durch das Gerät
5200	Störung der Geräteapplikation
5300	Störung der Bedien- und Anzeigeeinheit des Geräts
5400	Störung im Leistungsteil des Geräts
5500	Störung der Kommunikation zur eingebauten Zusatzbaugruppe
5600	Störung im internen Datenspeicher
6000	Störung der Geräte-Software
6100	Störung der internen Software (Firmware)
6200	Störung der Anwender-Software des Geräts
6300	Anwenderparameter (Datensatz) nicht OK
6800	Gerätekonfiguration nicht OK
7000	Störung an einer Zusatzbaugruppe/an Zusatzbaugruppen
7100	Störung im Leistungsteil
7200	Störung der Messschaltung
7300	Störung des Sensors (als Bestandteil fest mit Gerät verbunden)
7400	Störung in der Rechenschaltung
7500	Störung in der Kommunikation zur Zusatzbaugruppe
7600	Störung im externen Datenspeicher
7700	Drahtbruch/Kabelfehler
8000	Überwachung der Gerätefunktion
8100	Störung in der Feldkommunikation
8200	Störung der Regelung
8300	Störung des Momentenreglers
8400	Störung des Drehzahlreglers
8500	Störung des Lagereglers
8600	Störung des Positionierreglers
8700	Störung des Gleichlaufreglers
8800	Störung des Wickelreglers
8900	Störung am externen Sensor (eigenes Gerät)
8A00	Störung des externen Aktors
8B00	Vorbeugende Wartung erforderlich (Condition Monitoring)
8F00	Safety-Störung
9000	Störungen an externen Geräten
A000	Störung in einem modularen Gerät - Subsystem
BXXX	reserviert
F000	Störung in der Zusatzfunktionen

7.2.1.5.2 Hauptgruppen mit Untergruppen und Details

Code (hex) Bedeutung

0000	keine Störung
1000	Störung allgemein Bitte im Handbuch nachsehen oder den Support des Herstellers kontaktieren.
2000	Strom (fließt zu viel)
2100	Störung, die die Versorgung des Geräts oder Signale „geräteeingangseitig“ betrifft
2110	Kurz-/Erdschluss der Geräteversorgungsspannung - allgemein
2120	Erdschluss der Geräteversorgungsspannung - allgemein
2121	Erdschluss Phase (L1 bei einem 3-phasigen Anschluss)
2122	Erdschluss Phase L2 bei einem 3-phasigen Anschluss
2123	Erdschluss Phase L3 bei einem 3-phasigen Anschluss
2126	Erdschluss VCC (Versorgungsspannung, typ. 24V)
2127	Erdschluss GND
2130	Kurzschluss der Geräteversorgungsspannung – allgemein (z. B. 24 V und GND)
2131	Kurzschluss Phasen L1-L2 bei einem 3-phasigen Anschluss
2132	Kurzschluss Phasen L2-L3 bei einem 3-phasigen Anschluss
2133	Kurzschluss Phasen L3-L1 bei einem 3-phasigen Anschluss
2140	Kurzschluss/Querschuss an der Peripherie
2141	Querschuss zwischen Signalausgängen untereinander und/oder anderen Spannungen
2142	Querschuss zwischen einem Signalausgang und einem Taktsignal
2146	Kurzschluss (Querschuss) eines Signaleingangs nach VCC
2147	Kurzschluss (Querschuss) eines Signaleingangs nach GND
2150	Offene Signalleitungen (Eingangssignal nicht angeschlossen)
2160	Signalleitungsunterbrechung/Kabelbruch (Eingangssignal)
2180	Eingang wird nicht aktiviert (schaltet nicht)
2181	Beschaltung des Eingangs (z. B. Last) fehlerhaft
2200	Störung, die interne Netzteile oder Signale „geräteintern“ betrifft
2213	Überstrom im Hochlauf
2214	Überstrom im Ablauf
2220	Dauerüberstrom einer internen Spannungsquelle
2221	Dauerüberstrom Nr. 1
2222	Dauerüberstrom Nr. 2
2230	Kurz-/Erdschluss einer internen Spannungsquelle
2240	Erdschluss
2250	Kurzschluss
2300	Störung an der Versorgung oder an Signalen für die an das Gerät angeschlossene Peripherie (geräteausgangseitig)
2310	Dauerüberstrom der Peripherieverorgung - allgemein
2311	Dauerüberstrom der Quelle Nr. 1
2312	Dauerüberstrom der Quelle Nr. 2
2320	Kurz-/Erdschluss der Peripherieverorgung - allgemein
2330	Erdschluss der Peripherieverorgungsspannung
2331	Erdschluss Phase U bei einem 3-phasigen Anschluss
2332	Erdschluss Phase V bei einem 3-phasigen Anschluss
2333	Erdschluss Phase W bei einem 3-phasigen Anschluss
2326	Erdschluss VCC (Versorgungsspannung, typ. 24 V)
2327	Erdschluss GND
2340	Kurzschluss/Überlast an der Peripherie
2341	Kurzschluss Phasen U-V bei einem 3-phasigen Anschluss
2342	Kurzschluss Phasen V-W bei einem 3-phasigen Anschluss
2343	Kurzschluss Phasen W-U bei einem 3-phasigen Anschluss
2344	Überlast eines Signalausgangs
2345	Überlast der Initiatorversorgung
2346	Kurzschluss (Querschuss) eines Signalausgangs nach VCC
2347	Kurzschluss (Querschuss) eines Signalausgangs nach GND
2350	Andauernd offene Signalleitungen (Ausgang)
2360	Signalleitungsunterbrechung/Kabelbruch (Ausgangssignal)
2370	Querschuss
2371	Querschuss zwischen Signalausgängen untereinander und/oder anderen Spannungen
2372	Querschuss zwischen einem Signalausgang und einem Taktsignal
2380	Ausgang wird nicht aktiviert (schaltet nicht)
2381	Beschaltung des Signalausgangs (z. B. Last) fehlerhaft

3000	Spannung
3100	Störung, die die Versorgungsspannung des Geräts oder Signale „geräteeingangseitig“
betrifft	
3110	Versorgungsüberspannung
3111	Netzüberspannung Phase L1
3112	Netzüberspannung Phase L2
3113	Netzüberspannung Phase L3
3120	Versorgungsunterspannung
3121	Netzunterspannung Phase L1
3122	Netzunterspannung Phase L2
3123	Netzunterspannung Phase L3
3130	Versorgungsspannungsausfall
3131	Phasenausfall L1
3132	Phasenausfall L2
3133	Phasenausfall L3
3134	Phasenfolge nicht OK
3140	Versorgungsspannungsfrequenz nicht OK
3141	Frequenz zu groß
3142	Frequenz zu klein
3150	Versorgungsspannung verpolt
3180	Störung, die den Signaleingang betrifft – allgemein
3181	Offene Signalleitung (Eingang)
3182	Signalleitungsunterbrechung/Kabelbruch (Eingangssignal)
3183	Unerwarteter Signalwechsel am Signaleingang
3184	Überspannung am Signaleingang (Spannung größer als der zulässige Bereich)
3185	Signalspannung im nicht definierten Bereich
3186	Unterspannung am Signaleingang (Spannung kleiner als der zulässige Bereich)
3200	Störung, die interne Netzteile oder Signale „geräteintern“ betrifft
3210	Überspannung geräteintern
3211	Überspannung Nr. 1
3212	Überspannung Nr. 2
3220	Unterspannung geräteintern
3221	Unterspannung Nr. 1
3222	Unterspannung Nr. 2
3230	Aufladestörung
3300	Störung an den Signalen für die an das Gerät angeschlossene Peripherie
(geräteausgangseitig)	
3310	Ausgangsüberspannung
3311	Ausgangsüberspannung Phase U
3312	Ausgangsüberspannung Phase V
3313	Ausgangsüberspannung Phase W
3320	Ankerkreis
3321	Ankerkreis unterbrochen
3330	Feldkreis
3331	Feldkreis unterbrochen
3340	Ausgangsunterspannung
3400	Störung an der Versorgungsspannung für die Peripherie
3401	Versorgungsspannung für die Peripherie - Überspannung
3402	Versorgungsspannung für die Peripherie - fehlt
3403	Versorgungsspannung für die Peripherie - Unterspannung
3410	Störung an der Initiatorversorgung - allgemein
3411	Initiatorversorgung - Unterspannung
3412	Initiatorversorgung fehlt
3413	Initiatorversorgung - Überspannung
3420	Störung an der Aktorversorgung - allgemein
3421	Aktorversorgung - Unterspannung
3422	Aktorversorgung fehlt
3423	Aktorversorgung - Überspannung
4000	Temperatur - befindet sich nicht im zugelassenen Bereich
4100	Temperatur Umgebung nicht OK
4110	Übertemperatur der Umgebung
4120	Untertemperatur der Umgebung
4130	Temperatur der Zuluft
4140	Temperatur der Abluft
4200	Temperatur im Gerät nicht OK
4210	Übertemperatur im Gerät
4220	Untertemperatur im Gerät
4300	Temperatur eines externen Peripheriegeräts (z. B. Antrieb) nicht OK
4310	Übertemperatur im externen Peripheriegerät (z. B. Antrieb)

4320	Untertemperatur im externen Peripheriegerät (z. B. Antrieb)
4400	Temperatur der Versorgungseinheit nicht OK
4410	Übertemperatur der Versorgungseinheit
4420	Untertemperatur der Versorgungseinheit
5000	Störung der Geräte-Hardware (nur innerhalb des Gerätegehäuses)
5010	Bauteilestörung
5100	Störung der Versorgung im Gerät und durch das Gerät
5110	Störung in den Netzteilen für die Niederspannung allgemein
5111	Versorgung +/- 15 V
5112	Versorgung + 24 V
5113	Versorgung + 5 V
5114	Versorgung + 3,3 V
5115	Versorgung + 2,5 V
5116	Versorgung + 1,2 V
5118	U8 = herstellerspezifisch
.... für U8 bis U15 herstellerspezifisch
511F	U15 = herstellerspezifisch
5120	Störung der Versorgung Luft
5130	Störung der Versorgung Lack
5140	Störung der Versorgung des Zwischenkreises
5150	Störung der Versorgung (Netzteil) Initiator
5151	Interner Kurzschluss
5160	Störung der Versorgung (Netzteil) der Peripherie, die durch das Gerät versorgt wird.
5200	Störung der Geräteapplikation
5210	Messschaltung gestört
5220	Rechenschaltungen gestört
5230	Kommunikation (nur geräteintern) gestört
5300	Störung der Bedien- und Anzeigeeinheit des Geräts
5400	Störung im Leistungsteil des Geräts
5410	Endstufe gestört
5420	Chopper gestört
5430	Eingangsstufen gestört
5440	Schütze/Relais gestört
5441	Kanal 1
...	...
5448	Kanal 8
5450	Sicherungen defekt
5451	S1 = L1
5452	S2 = L2
5453	S3 = L3
5454	S4 = herstellerspezifisch
.... für S5, S6, S7, S8
5459	S9 = herstellerspezifisch
5500	Störung der Kommunikation zur eingebauten Zusatzbaugruppe
5510	Schnittstelle Nr. 1
5520	Schnittstelle Nr. 2
5600	Störung im internen Datenspeicher
5610	Fest verbautes RAM defekt
5620	Fest verbautes EPROM defekt
5630	Fest verbautes EEPROM defekt
5640	Fest verbautes Flash defekt
6000	Störung der Geräte-Software
6010	Software-Reset (Watchdog) aufgetreten
6100	Störung der internen Software (Firmware)
6110	Firmware fehlt (z. B. SD-Karte fehlt)
6120	Firmware inkonsistent (z. B. SD-Karte enthält korrupte Daten)
6130	Applikation auf dem Gerät noch nicht bereit (Teilapplikation bootet noch, Selbsttests wird durchgeführt etc.)
6200	Störung der Anwender-Software des Geräts
6210	Prozessdaten-Index nicht vorhanden
6211	Variablennummer nicht vorhanden
6300	Anwenderparameter (Datensatz) nicht OK
6301	Datensatz Nr. 1
..	von 2 bis 14 entsprechend
630F	Datensatz Nr. 15
6310	Verlust der intern gespeicherten Anwenderparameter
6320	Inkonsistenz bei den intern gespeicherten Anwenderparametern (z. B. durch CRC festgestellt)
6330	Anwenderparameter/Anlaufparameter noch nicht initialisiert
6340	Voneinander abhängige Anwenderparameter/Anlaufparameter sind zueinander nicht konsistent

6800	Gerätekonfiguration nicht OK
6810	Daten in der Prozessdatenkonfiguration doppelt gemappt
7000	Störung an einer Zusatzbaugruppe/an Zusatzbaugruppen
	Fest mit dem Gerät verbunden, durch den Anwender ggf. applizierbar, zum kompletten Gerät gehörig, meist damit ausgeliefert.
7100	Störung im Leistungsteil
7110	Bremschopper defekt
7111	Ausfall Bremschopper
7112	Überstrom Bremschopper
7113	Beschaltung Bremschopper
7120	Motor defekt
7121	Motor blockiert
7122	Motor fehlt oder Kommutierung gestört
7123	Motor gekippt
7140	(durch den Anwender) tauschbares Relais/Schütz defekt
7150	(durch den Anwender) tauschbare Sicherung defekt
7200	Störung der Messschaltung
7300	Störung des Sensors (als Bestandteil fest mit Gerät verbunden)
7301	Tachogeber defekt
7302	Tachogeber verpolt
7303	Resolver 1 defekt
7304	Resolver 2 defekt
7305	Inkrementalgeber 1 defekt
7306	Inkrementalgeber 2 defekt
7307	Inkrementalgeber 3 defekt
7308	Sensor 8 defekt
...	Sensor 9 bis 14 entsprechend
730F	Sensor 15 defekt
7310	Geschwindigkeitssensor defekt
7320	Lagesensor defekt
7400	Störung in der Rechenschaltung
7500	Störung in der Kommunikation zur Zusatzbaugruppe
7501	Kurzschluss der Datensignalleitung
7502	Leitungsunterbrechung der Datensignalleitung
7503	Störung der Datensignalübertragung (z. B. durch EMV)
7510	Serielle Schnittstelle Nr. 1 defekt
7520	Serielle Schnittstelle Nr. 2 defekt
7600	Störung im externen Datenspeicher
7610	Tauschbares RAM defekt
7620	Tauschbares EPROM defekt
7630	Tauschbares EEPROM defekt
7640	Tauschbares Flash defekt
7700	Drahtbruch/Kabelfehler
7701	Kabel 1 defekt
...	Kabel 2 bis 14 entsprechend
770F	Kabel 15 defekt
7710	Drahtbruch Kabel zum Sensor
8000	Überwachung der Gerätefunktion
8100	Störung in der Feldkommunikation
8110	Prozessdatenaktualisierungs-Timeout abgelaufen
8120	Hostdatenaktualisierungs-Timeout abgelaufen
8121	PD-Kanal Handshake-Timeout abgelaufen
8150	Bufferhandling gestört
8151	Sende-Buffer allgemein
8152	Sende-Buffer voll
8153	Sende-Buffer übergelaufen
8159	Empfangs-Buffer allgemein
815A	Empfangs-Buffer voll
815B	Empfangs-Buffer übergelaufen
8200	Störung der Regelung
8210	Regelabweichung Soll > Ist, die Abweichung steht länger als eine bestimmte Zeit an (herstellerspezifisch)
8211	Max. Stellgröße erreicht/überschritten
8220	Regelabweichung Soll < Ist, die Abweichung steht länger als eine bestimmte Zeit an (herstellerspezifisch)
8221	Min. Stellgröße erreicht
...	Reserviert für profilspezifische Regelstörungen
827f	Reserviert für profilspezifische Regelstörungen
8300	Störung des Momentenreglers

8311	Übermoment
8312	Schweranlauf
8313	Stillstandsmoment
8321	Untermoment
8331	Momentenbruch
8400	Störung des Drehzahlreglers
8500	Störung des Lagereglers
8600	Störung des Positionierreglers
8611	Schleppfehler
8612	Referenzgrenze überschritten
8700	Störung des Gleichlaufreglers
8800	Störung des Wickelreglers
8900	Störung am externen Sensor (eigenes Gerät)
8910	Messbereichsüberschreitung
8920	Messbereichsunterschreitung
8930	Zusatzbeschaltung für externen Sensor defekt
8A00	Störung des externen Aktors
8B00	Vorbeugende Wartung erforderlich (Condition Monitoring)
8F00	Safety-Störung
8F01	Symmetrieverletzung (Schaltreihenfolge falsch, Diskrepanzzeitüberwachung, Antivalent, Äquivalent)
8F08	Helltestfehler (Ausgang kann nicht eingeschaltet werden, wenn er inaktiv ist)
8F09	Dunkeltestfehler (Ausgang kann nicht abgeschaltet werden, wenn er aktiv ist)
8F0F	Failure State (kritische Störung, Sicherheit nicht gegeben, das Modul wird abgeschaltet)
9000	Störungen an externen Geräten
A000	Störung in einem modularen Gerät - Subsystem
A001	Kein Modul vorhanden/Modul fehlt
A002	Falsches Modul vorhanden
A003	Ersetzt (Das erwartete Modul wurde durch ein kompatibles Modul ausgetauscht.)
A004	Im Sub-Bus mehr Module als erwartet
A010	Allgemeiner Fehler im Modul
A012	Applikation auf dem Modul nicht bereit
A013	Modul hat einen Reset ausgeführt
A014	Modulparametrierung fehlt oder ist fehlerhaft
A020	Sub-Bus-Kommunikationsfehler
A021	Sub-Bus Fehler - Timeout
A022	Multiple Übertragungsfehler im Sub-Bus
A023	Sub-Bus-I/O-Daten-Kommunikationsfehler
A024	Sub-Bus-Managementdaten-Kommunikationsfehler
A025	Ring Error – die Busstruktur „offener Ring“ ist unterbrochen
A030	Sub-Bus-Konfigurationsfehler
A031	I/O-Datenkonfiguration im Sub-Bus doppelt gemappt
A033	Die reale Prozessdatenlänge stimmt nicht mit der konfigurierten Prozessdatenlänge überein. (Ist größer)
A040	Common Errors
A041	Hardware-Fehler – Modul muss getauscht werden
A042	Firmware-Fehler – Firmware des Moduls oder ganzes Modul muss getauscht werden
A043	Sub-Bus asynchron zum überlagerten System
BXXX	reserviert
F000	Störung in der Zusatzfunktionen
F001	Verzögerung
F002	Untersynchronlauf
F003	Hubwerk
F004	Steuerung

Nicht aufgeführte Codes sind reserviert.

ANMERKUNG: Weitere StörungsCodes sind in den Profilen definiert.

Sollte sich ein Hersteller in dieses Schema der StörungsCodes nicht einordnen können, wird er gebeten, sich mit dem INTERBUS Club in Verbindung zu setzen.

7.2.2 Trace-Daten

Traces sind fortlaufende Aufnahmen von bestimmten Daten eines Geräts. Sie dienen u. a. dazu, Abläufe zu rekapitulieren, um Fehler oder Probleme zu finden oder Daten archivieren zu können.

Grundsätzlich kann es sehr unterschiedliche Traces geben. Inhalt, Struktur, Umfang und Handhabung werden vom Gerätehersteller festgelegt.

Index (hex)	Objektname	R/W	Länge	Datentyp (Darstellung)	Bedeutung	M/O
001C	ControlTrace	R/W	1 Byte	UINT8 (Hex)	Steuerung der Protokollierung Steuern eines Aufnahmevorgangs. „0x00“ Reset der Aufnahme und der dazugehörigen Parameter „0x01“ Start einer Aufnahme „0x02“ Stopp einer Aufnahme Weitere Details werden vom Hersteller festgelegt	O
E803	TraceBuffer	R	N Byte	Domain-Variable Octet String (Hex)	Puffer für die Protokollierung Beinhaltet die aufgenommenen Trace-Daten. Die Struktur wird vom Hersteller festgelegt	D

Die Trace-Daten werden generell mit dem Upload-Mechanismus übertragen. Beim Upload der Trace-Daten sollten die grundlegenden Informationen im Header enthalten sein.

7.3 Anwenderdatenmanagement

7.3.1 Prozessdatenmanagement

Dieser Abschnitt definiert das Verhalten der Prozessdaten. Dazu gehört das Verhalten bei Timeouts und Feldbus-Reset sowie das Lesen und Schreiben der Prozessdaten über den Parameterkanal.

Index (hex)	Objektname	R/W	Länge	Datentyp (Darstellung)	Bedeutung	M/O
001F	PDTimeout	R/W	2 Byte	UINT16 (Dez)	Prozessdaten-Überwachungszeit Max. zulässige Zeit in ms (z. B. 100 ms), bis zu der neue Daten über den Prozessdatenkanal übertragen worden sein müssen. Die dann ausgelöste Aktion wird im Objekt 0x0020 „PDTimeoutCode“ festgelegt. 0xFFFF = ausgeschaltet	O
0020	PDTimeoutCode	R/W	N x 2 Byte	Array of UINT16 (Hex)	<p>Prozessdaten-Überwachungscode Bei Ablaufen der Prozessdaten-Überwachungszeit wird die dem „Prozessdaten-Überwachungscode“ zugeordnete Funktion durch den Teilnehmer ausgeführt.</p> <p>Eine solche Funktion kann z. B. das definierte Herunterfahren eines Antriebs sein oder das kontinuierliche Senden von Fehlertelegrammen über eine serielle Schnittstelle.</p> <p>0x0000 Ausgabe von „0“ auf allen Ausgangsbits 0x0001 Ausgabe von „1“ auf allen Ausgangsbits 0x0002 Halten des letzten gültigen Werts 0x0003 Übernahme des Ersatzwerts aus Objekt 0x002F „PDOOUT_Subst“ 0x0004 Ausgabe des minimal zulässigen Werts 0x0005 Ausgabe des maximal zulässigen Werts 0x0010 Nur bei Sub-Bussen: Einnahme des Sub-Bus-spezifischen Verhaltens.</p> <p>0x0006 ... 00x0009 und 0x0011 ... 0x7FFF reserviert 0x8000 bis 0xFFFF herstellerspezifisch</p> <p>Da das Verhalten ggf. für jeden Kanal/jede Gruppe unterschiedlich gewünscht sein kann, ist dieser Parameter als Array angelegt. Die Anzahl der Elemente des Arrays ist durch die Anzahl der Elemente des jeweiligen Objekts PDOOUT (0x0026) / PDIN (0x0025) definiert.</p> <p>N = Anzahl der Elemente Wenn nur ein Eintrag unterstützt (geschrieben) wird, obwohl das Gerät mehrere Elemente (Kanäle/Gruppen) besitzt, so gilt der Parameter für das gesamte Gerät.</p>	O
0024	ResetCode	R/W	N x 2 Byte	Array of UINT16 (Hex)	<p>Fieldbus Reset Code Ersatzwertverhalten bei Bus-Reset (PDOOUT) Ersatzwertverhalten beim Ausbleiben von gültigen Prozessdaten</p>	O

		<p>Bei Auftreten eines Bus-Resets wird die dem „ResetCode“ zugeordnete Funktion durch den Teilnehmer ausgeführt.</p> <p>Eine solche Funktion kann z. B. das definierte Herunterfahren eines Antriebs sein oder das kontinuierliche Senden von Fehlertelegrammen über eine serielle Schnittstelle.</p> <p>0x0000 Ausgabe von „0“ auf allen Ausgangsbits 0x0001 Ausgabe von „1“ auf allen Ausgangsbits 0x0002 Halten des letzten gültigen Werts 0x0003 Übernahme des Ersatzwerts aus Objekt 0x002F „PDOOUT_Subst“ 0x0004 Ausgabe des minimal zulässigen Werts 0x0005 Ausgabe des maximal zulässigen Werts 0x0010 Nur bei Sub-Bussen: Einnahme des Sub-Bus-spezifischen Verhaltens.</p> <p>0x0006 ... 00x0009 und 0x0011 ... 0x7FFF reserviert 0x8000 bis 0xFFFF herstellerspezifisch</p> <p>Da das Verhalten ggf. für jeden Kanal/jede Gruppe unterschiedlich gewünscht sein kann, ist dieser Parameter als Array angelegt. Die Anzahl der Elemente des Arrays ist durch die Anzahl der Elemente des jeweiligen Objekts PDOOUT (0x0026) / PDIN (0x0025) definiert.</p> <p>N = Anzahl der Elemente Wenn nur ein Eintrag unterstützt (geschrieben) wird, obwohl das Gerät mehrere Elemente (Kanäle/Gruppen) besitzt, so gilt der Parameter für das gesamte Gerät.</p> <p>Siehe auch Kap. „Ersatzwertverhalten/Einschaltverhalten“</p>				
0030	PF_Code	R/W	N x 2 Byte	Array of UINT16 (Hex)	<p>Peripheral Fail Code Ersatzwertverhalten bei Peripheriefehler (PDIN) Bei Auftreten eines Fehlers an der Peripherie des Feldbus-Protokollchips, der die Bearbeitung der Eingangsprozessdaten nicht mehr zulässt, wird durch den Teilnehmer die hier festgelegte Funktion ausgeführt.</p> <p>Eine solche Funktion kann z. B. über den Eingang /StatErr oder µP WD an den Protokoll-Chips angezeigt werden.</p> <p>0x0000 Ausgabe von „0“ auf allen Eingangsbits 0x0001 Ausgabe von „1“ auf allen Eingangsbits 0x0002 Halten des letzten gültigen Werts 0x0003 Übernahme des Ersatzwerts aus Objekt 0x0031 „PDIN_Subst“ 0x0004 Ausgabe des minimal zulässigen Werts 0x0005 Ausgabe des maximal zulässigen Werts 0x0010 Nur bei Sub-Bussen: Einnahme des Sub-Bus-spezifischen Verhaltens.</p> <p>0x0006 ... 00x0009 und 0x0011 ... 0x7FFF reserviert 0x8000 bis 0xFFFF herstellerspezifisch</p> <p>Da das Verhalten ggf. für jeden Kanal/jede Gruppe unterschiedlich gewünscht sein kann, ist dieser Parameter als Array angelegt. Die Anzahl der Elemente des Arrays ist durch die Anzahl der Elemente des jeweiligen Objekts PDOOUT (0x0026) / PDIN (0x0025) definiert.</p> <p>N = Anzahl der Elemente Wenn nur ein Eintrag unterstützt (geschrieben) wird, obwohl das Gerät mehrere Elemente (Kanäle/Gruppen) besitzt, so gilt der Parameter für das gesamte Gerät.</p>	O

0025	PDIN	R	PD Länge	Octet String[N]	<p>Process Input Data Eingangsprozessdaten Abbildung der Eingangsprozessdaten (vom Teilnehmer zum Master) auf ein Objekt Die Zählung der PD-Bytes wie auch der Bytes im Octet String erfolgt mit 0 beginnend von links (oben) aufsteigend.</p> <p>Wenn die Prozessdaten strukturiert sind (z. B. mehrere Kanäle), sollte dieses Objekt ebenfalls strukturiert sein und der Zugriff auf die einzelnen Strukturelemente erfolgt über den Subindex.</p>	M
.01	● Part 1	R		dependent	Erster Teil des Prozessdatums, z. B. Wert des ersten Kanals oder der ersten Funktionsgruppe	O
.02	● Part 2	R		dependent	Zweiter Teil des Prozessdatums, z. B. Wert des zweiten Kanals oder der zweiten Funktionsgruppe	O
...	●					
.N	● Part N	R		dependent	N-ter Teil des Prozessdatums, z. B. Wert des N-ten Kanals oder der N-ten Funktionsgruppe	O
0026	PDOOUT	R/W _D	PD Länge	Octet String[N]	<p>Process Output Data Ausgangsprozessdaten Abbildung der Ausgangsprozessdaten (vom Master zum Teilnehmer) auf ein Parameterobjekt.</p> <p>Dabei sollte auf Daten zugegriffen werden, die die Zustände an der Peripherie so gut wie möglich abbilden. Das trifft auch auf den Ersatzwertefall zu. Die Zählung der PD-Bytes wie auch der Bytes im Octet String erfolgt mit 0 beginnend von links (oben) aufsteigend.</p> <p>Wenn die Prozessdaten strukturiert sind (z. B. mehrere Kanäle), sollte dieses Objekt ebenfalls strukturiert sein und der Zugriff auf die einzelnen Strukturelemente erfolgt über den Subindex. Es ist in jeden Fall darauf zu achten, dass die Datenkonsistenz eines Werts (z. B. alle Bytes eines 32 Bit eines Analogwerts) sichergestellt ist</p> <p>Wenn das Objekt „GetExRight“ nicht implementiert ist, hat das Objekt PDOOUT den Status „Nur Lesen“. Wenn das Objekt „GetExRight“ implementiert ist, hat das Objekt PDOOUT den Status „Lesen und Schreiben“. Wenn das Objekt „GetExRight“ einen Wert = „0“ hat, wird ein schreibender Zugriff auf das Objekt PDOOUT abgewiesen mit Error Class „8“ - Other Error Code „1“ – Profile specific Additional Code 0x0022 „Dienst kann im momentanen Gerätezustand nicht ausgeführt werden“.</p> <p>ACHTUNG: Bei der Möglichkeit des Zugriffs, z. B. von Tools, auf Ausgangsprozessdaten, sollte darauf hingewiesen werden, dass hier eventuelle Prozessdaten-Watchdog-Funktionalitäten außer Kraft gesetzt werden/sind!</p>	M
.01	● Part 1	R		dependent	Erster Teil des Prozessdatums, z. B. Wert des ersten Kanals oder der ersten Funktionsgruppe	O
.02	● Part 2	R		dependent	Zweiter Teil des Prozessdatums, z. B. Wert des zweiten Kanals oder der zweiten Funktionsgruppe	O
...	●					
.N	● Part N	R		dependent	N-ter Teil des Prozessdatums, z. B. Wert des N-ten Kanals oder der N-ten Funktionsgruppe	O

002F	PDOOUT_Subst	R/W	PD Länge	dependent	<p>Process Output Data Substitute Ersatzwert Ausgangsprozessdaten Ersatzwert für die Ausgangsprozessdaten (vom Master zum Teilnehmer) im Fehlerfall. Die Zählung der PD-Bytes wie auch der Bytes im Octet String erfolgt mit 0 beginnend von links (oben) aufsteigend.</p> <p>Dependency Info: Muss implementiert werden, wenn für die Objekte 0x0020 „PDTimeoutCode“ oder 0x0024 „ResetCode“ der Wert 0x0003 zulässig ist. Siehe dort</p>	D
0031	PDIN_Subst	R/W	PD Länge	dependent	<p>Process IN Data Substitute Ersatzwert Eingangsprozessdaten Ersatzwert für die Eingangsprozessdaten (vom Teilnehmer zum Master) im Falle eines Fehlers der angeschlossenen Peripherie. Die Zählung der PD-Bytes wie auch der Bytes im Octet String erfolgt mit 0 beginnend von links (oben) aufsteigend.</p> <p>Dependency Info: Muss implementiert werden, wenn für das Objekt 0x0030 „PF_Code“ der Wert 0x0003 zulässig ist. Siehe dort</p>	D
0027	GetExRight	R/W	1 Byte	UINT8 (Hex)	<p>Get Exclusive Process Data Write Rights Exklusive Schreibrechte anfordern Durch diesen Parameter können die exklusiven Schreibrechte auf die Prozessausgänge über den Parameterkanal angefordert werden. Die Daten werden bei positiver Confirmation ab dann nicht mehr über den Prozessdatenkanal aktualisiert. Ein Ändern der Prozessausgänge erfolgt über das dann lese- und <u>schreibbare</u> Objekt „PDOOUT“. Mit jedem Verbindungsabbau oder Bus-Reset werden die Exklusivrechte wieder zurückgesetzt.</p> <p>Achtung: diese Aktion kann schwerwiegende Folgen für den angeschlossenen Prozess haben. Deshalb sollte dieses Objekt mit einem Passwort geschützt werden.</p> <p>= „0x00“ Ausgangsdaten über den PD-Kanal = „0x01“ Ausgangsdaten über den Parameterkanal sonst: reserviert</p> <p>ACHTUNG: Bei der Möglichkeit des Zugriffs, z. B. von Tools, auf Ausgangsprozessdaten sollte darauf hingewiesen werden, dass hier eventuelle Prozessdaten-Watchdog-Funktionalitäten außer Kraft gesetzt werden/sind!</p>	O

0028	ChangePDSet	R/W	2 Byte	UINT16 (Dez)	<p>Change Process Data Settings Prozessdatenzuordnungen einstellen Wählt eine der möglichen, vom Hersteller vorgegebenen, Prozessdatenzuordnungen aus und aktiviert diese.</p> <p>0x0000 Default 0x0001 bis 0x7FFF reserviert 0x8000 bis 0xFFFF herstellerspezifisch</p> <p>Die Änderung der Prozessdatenzuordnung ändert im Regelfall auch die inhaltliche Bedeutung und Struktur der Prozessdaten und erfordert ggf. eine neue Zuordnung der Prozessdaten.</p> <p>Wenn sich dadurch Datenlänge, Teilnehmertyp- und/oder Längencode ändern, so kann dies erst mit dem nächsten Bus-Reset oder auf explizite Veranlassung des Bus-Masters gültig werden.</p>	O
003B	PDIN_Descr	R	N x 12 Byte	Array of Records (N x 3 Elemente)	<p>Process Data Description Beschreibung der Eingangsprozessdaten Beschreibung des Aufbaus der Prozessdaten. Dadurch ist es Tools und Systemen möglich, automatisch den Prozessdaten die richtigen Variablen zuzuordnen, die Endianess zu berücksichtigen oder diese einer passenden Verbindung zuzuordnen. Alle Typen können beliebig oft vorkommen. Die Beschreibung muss lückenlos aufsteigend, von links beginnend erfolgen.</p> <p>Die Anzahl der Elemente des Arrays ist durch die Anzahl der Elemente des Objekts PDIN (0x0025) definiert. N = Anzahl der Elemente PDIN (0x0025)</p>	M

.01	• Type	R	8 Byte	Octet String[8] (Text)	<p>Type des I/O-Datums. Definiert sind derzeit:</p> <p>Management-Datentypen:</p> <ul style="list-style-type: none"> • „ACK“ - Antwort auf ein Kommando • „DIAG“ - Diagnoseinformationen • „NC“ - Not Connected, nicht benutzt • „STATUS“ - Statusinformationen <p>Nutzdatentypen:</p> <ul style="list-style-type: none"> • „AI“ - Analog IN • „AO_F“ - Analog OUT - Feedback • „CmpP“ - Complex-Protokoll • „CNT“ - Counter • „DI“ - Digital IN • „DO_F“ - Digital OUT - Feedback • „DRV“ - Drive (Antrieb) • „MotSt“ - Motorstarter • „NO“ - Number • „PLC“ - PLC (Daten eines intelligenten Moduls) • „POS“ - Positionierdaten • „PWM“ - PWM-Daten • „SFCH“ - Channel for safe Data • „SubMa“ - Sub-Bus-Master • „SubSl“ - Sub-Bus-Slaves-Container <p>Der Vorsatz „S_“ (z. B. „S_DO_F“) kennzeichnet einen Safety-Datentyp.</p> <p>Der Type ist genau 8 Zeichen lang. Nicht benutzte Zeichen sind mit 0x00 aufzufüllen.</p> <p>Bsp: „AI“ - 0x41 49 00 00 00 00 00 00</p>	M
.02	• ChNo	R	2 Byte	UINT16 (Dez)	<p>Channel Number Anzahl der Kanäle eines Typs</p> <p><u>Anmerkung:</u> Ein Kanal ist immer eine Anzahl (auch 1) von Bits, die zusammen eine Bedeutung haben. Beispiel: Vier analoge Eingänge sind 4 Kanäle oder 16 digital Eingänge sind 16 Kanäle</p>	M
.03	• ChLength	R	2 Byte	UINT16 (Dez)	<p>Channel Length Länge eines Kanals in Bit</p>	M
003C	PDOOUT_Descr	R	N x 12 Byte	Array of Records (N x 3 Elemente)	<p>Process Data Description Beschreibung der Ausgangsprozessdaten Beschreibung des Aufbaus der Prozessdaten Dadurch ist es Tools und Systemen möglich, automatisch den Prozessdaten die richtigen Variablen zuzuordnen, die Endianess zu berücksichtigen oder diese einer passenden Verbindung zuzuordnen. Alle Typen können beliebig oft vorkommen. Die Beschreibung muss lückenlos aufsteigend, von links beginnend erfolgen.</p> <p>Die Anzahl der Elemente des Arrays ist durch die Anzahl der Elemente des Objekts PDOOUT (0x0026) definiert. N = Anzahl der Elemente PDOOUT (0x0026)</p>	M

.01	• Type	R	8 Byte	Octet String[8] (Text)	Type des I/O-Datums. Definiert sind derzeit: Management-Datentypen: <ul style="list-style-type: none"> • „CMD“ - Kommandos • „CTRL“ - Steuerung/Kontrolle • „NC“ - not connected, nicht benutzt Nutzdatentypen: <ul style="list-style-type: none"> • „AO“ - Analog Out • „CmpP“ - Complex-Protokoll • „CNT“ - Counter • „DO“ - Digital Out • „DRV“ - Drive (Antrieb) • „MotSt“ - Motorstarter • „NO“ - Number • „PLC“ - PLC (Daten eines intelligenten Moduls) • „POS“ - Positionierdaten • „PWM“ - PWM-Daten • „SFCH“ - Channel for safe Data • „SubMa“ - Sub-Bus-Master • „SubSl“ - Sub-Bus-Slaves-Container Der Type ist genau 8 Zeichen lang. Nicht benutzte Zeichen sind mit 0x00 aufzufüllen. Bsp.: „CNT“- 0x43 4E 54 00 00 00 00 00	M
.02	• ChNo	R	2 Byte	UINT16 (Dez)	Channel Number Anzahl der Kanäle eines Typs <u>Anmerkung:</u> Ein Kanal ist immer eine Anzahl (auch 1) von Bits, die zusammen eine Bedeutung haben. Beispiel: Vier analoge Eingänge sind 4 Kanäle oder 16 digital Eingänge sind 16 Kanäle	M
.03	• ChLength	R	2 Byte	UINT16 (Dez)	Channel Length Länge eines Kanals in Bit	M

Beispiel für ein vierkanaliges analoges Ausgabemodul:

Inhalt von 0x003B „PDIN_Descr“, 2 Einträge (hex):

```
53 54 41 54 55 53 00 00 00 10 00 01
4E 43 00 00 00 00 00 00 00 08 00 08
```

Inhalt von 0x003C „PDOOUT_Descr“, 2 Einträge (hex):

```
4E 43 00 00 00 00 00 00 00 02 00 08
41 4F 00 00 00 00 00 00 00 04 00 10
```

7.3.2 Ersatzwertverhalten/Einschaltverhalten

Es wird unterschieden zwischen:

7.3.2.1 Einschaltverhalten (PowerOn behaviour):

Ein-/Ausgänge müssen **nach Einschalten der Spannungsversorgung** einen, im Sinne der Maschinenrichtlinie, sicheren (definierten) Zustand einnehmen (inaktiv, typ. „0“).

Dieser Zustand ist nach dem PowerOn so lange beizubehalten, bis

- das Gerät über eine gültige Parametrierung verfügt, bei der die Ein-/Ausgänge verwendet werden (nicht deaktiviert sind)
- und
 - gültige Ausgangsdaten empfangen worden sind. bzw.
 - gültige Eingangsdaten zur Verfügung gestellt werden können.

Es handelt sich hierbei, wenn möglich, um eine Verlängerung des Zustands „PowerOff“. Nicht verwendete (deaktivierte) Ein-/Ausgänge verlassen das Einschaltverhalten nicht. Wenn sie schon einmal aktiv waren, wechseln die Ein-/Ausgänge in das Einschaltverhalten.

Hinweis: Es ist nicht zwingend erforderlich, nach dem Start einen Parameter zu schreiben, um in diesen Zustand zu gelangen, wenn in der lokal gespeicherten Parametrierung bereits ein Kanal aktiv genutzt wird.

7.3.2.2 Ersatzwertverhalten (FailSafe behaviour):

Das Gerät muss bei **Ausbleiben gültiger Ausgangsdaten***) bzgl. seiner Ausgänge in den Zustand „Ersatzwertverhalten“ wechseln, wenn

- eine Parametrierung existiert, bei der die Ausgänge verwendet werden (aktiv sind).
- und
- zuvor mindestens einmalig gültige Prozessdaten ausgetauscht wurden.

Für die Ausgangsprozessdaten wird das Ersatzwertverhalten bis zum Erhalt gültiger Ausgangsprozessdaten aufrechterhalten.

Das Gerät muss, wenn **keine gültigen Eingangsdaten zur Verfügung gestellt werden** können, bzgl. seiner Eingänge in den Zustand „Ersatzwertverhalten“ wechseln, wenn

- eine Parametrierung existiert, bei der die Eingänge verwendet werden (aktiv sind).

Nur verwendete Ein-/Ausgänge nehmen das Ersatzwertverhalten an. Nicht verwendete Ein-/Ausgänge verlassen das Einschaltverhalten nicht.

Hinweis: Es ist nicht zwingend erforderlich, nach dem Start einen Parameter zu schreiben, um in diesen Zustand zu gelangen, wenn in der lokal gespeicherten Parametrierung bereits ein Kanal aktiv genutzt wird.

*) Wie „das Ausbleiben gültiger Ausgangsdaten“ definiert ist, legt das System fest. Gewöhnlich ist es das Ablaufende eines Timeouts (Reset), der Empfang eines entsprechenden Kommandos oder das Ungültigwerden eines Prozessdatenbegleiters.

7.3.4 Funktionale Sicherheit – Safety

Eintrag in Objekt 0x0005 „Capabilities“: „Safety0“

Das Basisprofil beschreibt keine Verfahren oder Protokolle, die eine, im Sinne der funktionalen Sicherheit, sichere Übertragung von Daten gewährleisten.

Die Unterstützung für Safety-Geräte beschränkt sich darauf, dass diese im System identifizierbar sind. Dazu wurden an den entsprechenden Stellen Objekte und Einträge definiert, die es dem Master ermöglichen, zu erkennen,

- Dass es sich um ein Gerät aus der Gerätefamilie Safety handelt (DeviceFamily Index 0x0004)
- Wo sich die sicherheitsgerichteten Daten im Prozessdatenkanal befinden (PDIN_Descr Index 0x003B und PDOOUT_Descr Index 0x003C)
- Dass sichere I/O-Daten über den Standardmechanismus mitgelesen werden können (PDIN Index 0x0025 und PDOOUT Index 0x0026)
- Welche Störungen sicherheitsrelevant sind (Störungscode 0x8FXX)
- Welches Safety-Protokoll unterstützt wird (SafetyProtType Index 0x0047.1)
- usw.

7.3.5 Parameterkanalmanagement

Dieser Abschnitt definiert das Verhalten des Parameterkanals bei Timeouts und Kommunikationsabbruch.

Index (hex)	Objektname	R/W	Länge	Datentyp (Darstellung)	Bedeutung	M/O
0021	PChTimeout	R/W	2 Byte	UINT16 (Dez)	<p>Parameterkanal-Überwachungszeit Maximal zulässige Zeit in ms (z. B. 500 ms), bis zu der neue Daten über den Parameterkanal übertragen worden sein müssen/der Service abgeschlossen sein muss. Er gilt für das Lesen und Schreiben gleichermaßen.</p> <p>Die dann auf dem Slave ausgelöste Aktion wird im Objekt 0x0022 „PChTimeoutCode“ festgelegt.</p> <p>Der Master kann, wenn er innerhalb dieser Zeit keine Confirmation auf seinen Request erhalten hat, die Anfrage abbrechen und meldet an das überlagerte System einen PCh–Timeout. Ausnahme: „Busy“</p> <p>Default: 500 ms 0xFFFF = ausgeschaltet</p> <p>Der Slave muss Zeiten, die er nicht realisieren kann, ablehnen.</p>	M
0022	PChTimeoutCode	R/W	2 Byte	UINT16 (Hex)	<p>Parameterkanal-Überwachungscode Bei Ablauf der Kommunikationsüberwachungszeit wird die dem Parameterkanalüberwachungscode zugeordnete Funktion ausgeführt.</p> <p>Eine solche Funktion kann z. B. das definierte Herunterfahren eines Antriebs sein oder das kontinuierliche Senden von Fehlertelegrammen über eine serielle Schnittstelle.</p> <p>0x0000 keine Aktion (Default) 0x0001 Wiederholung des Dienstes (erwarten) 0x0002 Abbruch (Abort) des Dienstes erwarten 0x0003 Abbruch des Dienstes selbst aktiv durchführen 0x0004 bis 0x7FFF reserviert 0x8000 bis 0xFFFF herstellerspezifisch</p>	O
0023	AbortCode	R/W	2 Byte	UINT16 (Hex)	<p>Verbindungsabbruchcode Bei Auftreten eines Verbindungsabbruchsauf dem Parameterkanal wird die dem Parameterkanalabbruchcode zugeordnete Funktion ausgeführt.</p> <p>Eine solche Funktion kann z. B. das definierte Herunterfahren eines Antriebs sein oder das kontinuierliche Senden von Fehlertelegrammen über eine serielle Schnittstelle.</p> <p>0x0000 keine Aktion (Default) 0x0001 bis 0x7FFF reserviert 0x8000 bis 0xFFFF herstellerspezifisch</p>	O

7.4 Gerätemanagement

Feldgeräte erlangen immer komplexere Funktionalitäten, die durch die Anwendung oder Software-Tools geeignet parametrierbar sein müssen. Im Folgenden wird ein Grundgerüst festgelegt, das es erlaubt, die dabei üblichen grundlegenden Handlungsabläufe zu vereinheitlichen.

Dependency Info:

Wenn das Objekt 0x0029 „ParamSetWriteControl“ implementiert ist, muss auch das Objekt 0x002A „ConflictDictionary“ implementiert sein.

Index (hex)	Objektname	R/W	Länge	Datentyp (Darstellung)	Bedeutung	M/O
0029	ParamSetWriteControl	R/W	1 Byte	UINT8 (Hex)	Steuerung der Blockparametrierung siehe Kapitel „voneinander abhängige Parameter“ unten	O
002A	ConflictDictionary	R	N x 8 Byte	Array of Records N x 6 Elemente	Konfliktverzeichnis Ergebnis der Blockparametrierung siehe Kapitel „voneinander abhängige Parameter“ unten	D
1. voneinander abhängiger Parameter, der zu mindestens einem weiteren im Konflikt steht.						
.01	• ConfGrNo_1	R	1 Byte	UINT8 (Dez)	Konfliktgruppennummer des 1. abhängigen Parameters Durch diese Nummer kann der folgende Parameter einer von ggf. mehreren Gruppen von Parametern zugeordnet werden, die zueinander im Konflikt stehen.	D
.02	• Subslot_1	R	1 Byte	UINT8 (Dez)	Subslot des 1. abhängigen Parameters	D
.03	• Index_1	R	2 Byte	UINT16 (Hex)	Index des 1. abhängigen Parameters	D
.04	• Subindex_1	R	1 Byte	UINT8 (Hex)	Subindex des 1. abhängigen Parameters	D
.05	• Element_1	R	1 Byte	UINT8 (Hex)	Element im Record des 1. abhängigen Parameters	D
.06	• Add. Code_1	R	2 Byte	UINT16 (Hex)	Add. Info zum 1. abhängigen Parameter	D
	• ...					
N. voneinander abhängiger Parameter, der zu mindestens einem weiteren im Konflikt steht.						
.6xN-5	• ConfGrNo_N	R	1 Byte	UINT8 (Dez)	Konfliktgruppennummer des N. abhängigen Parameters	D
.6xN-4	• Subslot_N	R	1 Byte	UINT8 (Dez)	Subslot des N. abhängigen Parameters	D
.6xN-3	• Index_N	R	2 Byte	UINT16 (Hex)	Index des N. abhängigen Parameters	D
.6xN-2	• Subindex_N	R	1 Byte	UINT8 (Hex)	Subindex des N. abhängigen Parameters	D
.6xN-1	• Element_N	R	1 Byte	UINT8 (Hex)	Element im Record des N. abhängigen Parameters	D
.6xN	• Add. Code_N	R	2 Byte	UINT16 (Hex)	Add. Info zum N. abhängigen Parameter	D

002B	ParamSet	R/W	2 Byte	UINT16 (Hex)	Parametersatzkennung (Index) Wenn das Gerät über Parametersätze verfügt, wird hier der jeweils aktuelle Parametersatz (Nummer/Index) eingestellt bzw. angezeigt. Wenn eine Beschreibung des Parametersatzes existiert, so entspricht die Parametersatzkennung dem Index der Beschreibung. Siehe Kapitel „Parametersatzkennung“	O
					<ul style="list-style-type: none"> • 0x0000 Der Parametersatz des Geräts wurde noch nicht über den Bus initialisiert. Es ist der Default-Parametersatz gültig. • 0x0001 – 0xFEFF Der Parametersatz des Geräts, der über Mechanismen des Basisprofils definiert wurde oder den der Hersteller frei definiert hat, wurde über den Bus initialisiert. Zur Kennung hat er die frei wählbare Nummer/den Index. 0x0001 - 0xFEFF erhalten. • 0xFF00 – 0xFFFD Reserviert • 0xFFFE Es ist nicht sicher gestellt, dass der zuvor geladenen Parametersatz unverändert ist. Dieser Wert kann nur gelesen werden. • 0xFFFF Das Gerät wurde auf „Local Mode“ oder Handbetrieb geschaltet und es ist nicht sicher gestellt, dass der zuvor geladenen Parametersatz unverändert ist. 	
002C	ParameterMoment	R/W	20 Byte	Record (2 Elemente)	Zeitpunkt der letzten Änderung der Parametrierung	O
.01	<ul style="list-style-type: none"> • Date 	R/W	10+1 Byte	Visible String (Text)	Datum YYYY-MM-DD	D
.02	<ul style="list-style-type: none"> • Time 	R/W	8+1 Byte	Visible String (Text)	Uhrzeit hh:mm:ss	D
002D	ResetParam	R/W	1 Byte	UINT 8 (Hex)	Parametrierung zurücksetzen Wird als Kommando zum Rücksetzen in den Ursprungszustand benutzt.	M

					<p>0x00 keine Aktion</p> <p>0x01 Parametrierung zurücksetzen Mit diesem Kommando werden alle Einstellungen rückgängig gemacht und durch die Factory-Default-Werte, bezogen auf den in 0x47.3 „LegacyInfo“ definierten Zustand, ersetzt. Das gilt auch für die Passworte und sonstige, durch den Anwender vorgenommene, Einstellungen. Dieser Wert ist mandatory.</p> <p>0x02 Applikationsparametrierung zurücksetzen Mit diesem Kommando werden nur die Applikationsparameter rückgängig gemacht und durch die Default-Werte, bezogen auf den in 0x47.3 „LegacyInfo“ definierten Zustand, ersetzt. Die im Abschnitt „Identifizierung“ aufgeführten Parameter sind keine Applikationsparameter.</p> <p>0x03 Legacy Reset Wie 0x01, nur bezogen auf den Auslieferungszustand. Es wird der sogenannte „Out-of-the-box“-Zustand hergestellt.</p> <p>0x04 Legacy-Applikationsparametrierung zurücksetzen Wie 0x02, nur bezogen auf den Auslieferungszustand.</p> <p>Nach erfolgter Aktion wird der Inhalt selbsttätig auf 0x00 zurückgesetzt.</p> <p>Wenn das Objekt 0x47.3 „LegacyInfo“ nicht vorhanden ist, haben die Werte 0x01 und 0x03 sowie 0x02 und 0x04 die gleiche Bedeutung.</p>	
002E	ParamHash	R	4 Byte	UINT32 (Hex)	<p>Hash-Wert Gerätspezifisch erzeugter, eindeutiger Wert (z. B. CRC16), der die Integrität der Parameterdaten sicherstellt. Der dafür verwendete Sicherungsmechanismus bezieht alle Parameterdaten des Geräts ein.</p> <p>Der Hash-Wert wird bei jeder Änderung eines Parameters neu generiert und ermöglicht es dem Anwender, zu kontrollieren, ob noch alle Parametereinstellungen unverändert sind. Dazu muss der aktuell ausgelesene Hash-Wert mit dem bei der Parametrierung ausgelesenen Hash-Wert verglichen werden.</p> <p>Wie der Hash-Wert generiert wird, ist dem Hersteller überlassen. Auch ein „Änderungszähler“ ist möglich. Wichtig ist nur, dass bei identischem Parametersatz derselbe Hash-Wert geliefert wird.</p>	O

0040	ListOfObjToRestore	R	N x 4 Byte	Array of Records (3 Elemente)	List of Objects to be Restore Liste der wiederherzustellenden Objekte Liste aller Applikationsobjekte [Sub-Modul-Nr. (1 Byte) + Index (2 Byte) + Subindex (1 Byte)], deren Inhalt bei einem Gerätetausch wiederhergestellt werden muss. (siehe Kapitel „Anwendungsfall Gerätetausch“) Über das Objekt 0x002D „ResetParam“ wird für alle Objekte der „Factory Default“ wiederhergestellt. Es müssen nur die Objekte berücksichtigt werden, deren aktueller Inhalt sich von dem des “Factory Defaults” unterscheidet. Die Liste kann dynamisch sein, muss es aber nicht. Bei einfachen Geräten können das alle Applikationsobjekte sein. Es kann aber auch nur das Objekt 0xE809 „BackUpDataComp“ enthalten sein.	M
.01	<ul style="list-style-type: none"> • SubModNo1 • Index1 • Subindex1 	R	4 Byte	Record (3 Elemente)	Sub-Modul-Nummer, Index und Subindex des 1. Objekts (Subindex = „0“ ist zulässig, wenn dadurch das gesamte Objekt gelesen/geschrieben werden kann)	M
.02	<ul style="list-style-type: none"> • SubModNo2 • Index2 • Subindex2 	R	4 Byte	Record (3 Elemente)	Sub-Modul-Nummer, Index und Subindex des 2. Objekts (Subindex = „0“ ist zulässig, wenn dadurch das gesamte Objekt gelesen/geschrieben werden kann)	M
	<ul style="list-style-type: none"> • 					
.N	<ul style="list-style-type: none"> • SubModNoN • IndexN • SubindexN 	R	4 Byte	Record (3 Elemente)	Sub-Modul-Nummer, Index und Subindex des N. Objekts (Subindex = „0“ ist zulässig, wenn dadurch das gesamte Objekt gelesen/geschrieben werden kann)	M

Anmerkung zum Objekt 0x0040 “ListOfObjToRestore”:

Das System (überlagertes Netzwerk/Steuerung) entscheidet darüber, ob Parameter generell gespeichert werden und wie Änderungen (insbesondere, wenn sie systemunabhängig, von außen direkt in das Modul eingespeist werden) erkannt werden.

7.4.1 Anwendungsfall Gerätetausch

Ein Gerätetausch kann mit Hilfe dieser Parameterliste durch folgende Vorgehensweise optimal durchgeführt werden:

Nach/bei Inbetriebnahme der „Anlage“ sind die Inhalte der Parameter dieser Liste (dieser Parametersatz) „zu sichern“.

Wenn ein Gerätetausch erforderlich wird, so ist bei dem kompatiblen Ersatzgerät immer zuerst ein Factory-Reset durchzuführen und der zuvor gesicherte Parametersatz zu schreiben.

7.4.2 Voneinander abhängige Parameter

Es kann vorkommen, dass verschiedene Parameter voneinander abhängig sind. D. h. die Gültigkeit des Werts eines Parameters kann von anderen Parametern abhängen und kann dementsprechend erst beurteilt werden, wenn alle voneinander abhängigen Parameter ihre endgültigen Werte angenommen haben.

Um das zu gewährleisten, wird der Mechanismus „Blockparametrierung“ eingeführt.

7.4.2.1 Blockparametrierung

Die Blockparametrierung dient dazu, voneinander abhängige Parameter gemeinsam übergeben zu können. Kommunikationstechnisch wird so das Schreiben mehrerer voneinander abhängiger Parameter mit Hilfe der Blockparametrierung wie das Schreiben eines Parameters betrachtet und auch so behandelt.

Über den Parameter „*ParamSetWriteControl = 01*“ wird die Blockparametrierung eingeleitet und mit „*ParamSetWriteControl = 00*“ beendet.

Die Prüfung der Abhängigkeiten erfolgt erst mit Abschluss der Blockparametrierung. Dementsprechend werden sich daraus ergebende Konflikte erst dann und nicht beim Schreiben des entsprechenden Parameters gemeldet.

Aufgetretene Konflikte werden im „Konfliktverzeichnis“ (Objekt ConflictDictionary Index 0x002A) abgelegt.

7.4.2.1.1 Steuerung der Blockparametrierung (*ParamSetWriteControl Index 0x0029*)

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

ParamSetWriteControl = „0x00“: Beenden der Blockparametrierung, ab jetzt Einzelparametrierung

ParamSetWriteControl = „0x01“: Einleiten der Blockparametrierung

Folgende Aktionen werden bei einer Änderung des Parameterinhalts durchgeführt:

Einleiten der Blockparametrierung (ParamSetWriteControl = „0x00“ -> „0x01“)

- Einleiten der Blockparametrierung
- Keine Einzelparametrierung
- Zurücksetzen des Listenparameters „Konfliktverzeichnis“

Abschließen der Blockparametrierung (ParamSetWriteControl = „0x01“ -> „0x00“)

- Beenden der Blockparametrierung, ab jetzt wieder Einzelparametrierung
- Überprüfung auf Verträglichkeit der Parameter untereinander (Parameterkonflikte.)
Dabei müssen nicht zwingend alle für die Überprüfung herangezogenen Parameter innerhalb der Blockparametrierung geschrieben worden sein. Die für den Plausibilitätscheck herangezogenen Objekte bestimmt das Modul selbst.
 - Wenn die Verträglichkeit gegeben ist,
 - werden die Parameterinhalte übernommen
 - wird der Schreibzugriff auf den Parameter „ParamSetWriteControl“ positiv quittiert.
 - Bei Unverträglichkeit
 - bleiben die alten Inhalte aller an der Blockparametrierung beteiligten Parameter wirksam, die neu geschriebenen Werte werden nicht wirksam.
 - Eintrag des Parameters in das „Konfliktverzeichnis“.
 - der Schreibzugriff auf den Parameter „ParamSetWriteControl“ wird wie folgt negativ quittiert.

Error Class “8” Other, Error Code “1” Profile specific

Additional Code	Bedeutung
0x0040 DependencyIgnored	Kollision mit anderen Werten, Abhängigkeit nicht beachtet In der Regel wurden abhängige Werte nicht berücksichtigt.

7.4.2.1.2 Konfliktverzeichnis (*ConflictDictionary Index 0x002A*)

Der Parameter enthält die Indizes und die Error Types (Additional Code) der am Konflikt beteiligten Parameter.

- Wenn die Werte zweier oder mehrerer voneinander abhängiger Parameter einen Konflikt verursachen würden, werden diese Parameter in das Konfliktverzeichnis eingetragen.
- Wenn die Werte zweier oder mehrerer voneinander abhängiger Parameter keinen Konflikt mehr verursachen, werden diese Parameter aus dem Konfliktverzeichnis gelöscht.
Dies ist unabhängig davon, ob die Blockparametrierung aktiv ist oder nicht.

Das Konfliktverzeichnis muss aktualisiert werden bei:

- negativer Quittierung beim Abschließen der Blockparametrierung (neg. Confirmation auf das Schreiben des ParamSetWriteControl = „0x01“ -> „0x00“)
- der Einzelparametrierung von voneinander abhängigen Parametern.

Das Konfliktverzeichnis kann aktualisiert werden:

- nach jedem Schreibzugriff auf abhängige Parameter, innerhalb des Zustands „Blockparametrierung“.

Das Konfliktverzeichnis wird gelöscht:

- beim Einleiten der Blockparametrierung
- beim Schreiben eines beliebigen anderen voneinander abhängigen Parameters außerhalb der Blockparametrierung (Einzelparametrierung)

Das Konfliktverzeichnis ist leer, wenn bei der Parametrierung keine Konsistenzverletzung festgestellt wurde.

Das Konfliktverzeichnis kann nur bis max. PDU-Size gefüllt werden. Falls weitere Einträge vorhanden sein sollten, so entfallen diese.

7.4.3 Parametersatzkennung

Dieser Parameter dient zur Kennzeichnung des aktuell wirksamen Geräte-Parametersatzes. Dies kann sich z. B. auf ein Schweißprogramm oder eine Rezeptur beziehen. Parametersätze können vom Hersteller voreingestellt sein oder vom Anwender je nach Gerät mit Hilfe von Einzel- oder Blockparametrierung oder auch „Download-Write“-Diensten geladen werden.

Wenn das Gerät die empfangenen Geräteparameterwerte nicht netzausfallsicher speichern kann, so wird nach Einschalten der Netzspannung diese Parametersatzkennung vom Gerät selbstständig auf 0x0000 gesetzt. Diese Information kann der Gerätenutzer auswerten und entsprechend neu initialisieren.

Ein Parametersatz ist immer eine Zusammenstellung von Parametern. In der Handhabung gibt es unterschiedliche Möglichkeiten:

1. Die Zusammenstellung kann vom Hersteller festgelegt werden, ohne dass diese über Objekte auslesbar ist. Sie bekommt einfach eine Nummer. Die Objekte, aus denen sich der Parametersatz zusammensetzt, muss dann auch nicht zwingend im Objekt 0x002B.02 - 0x002B.N „ParamSet. ObjectN“ beschrieben sein. Die Bedeutung kann im Anwenderhandbuch beschrieben sein.
2. Die Zusammenstellung kann vom Hersteller festgelegt werden und der Inhalt und die Struktur werden über ein Objekt XYZ offengelegt. Sie bekommt eine Nummer. Das Objekt XYZ ist dann wie unten beschrieben zu definieren. Der Parametersatz kann in diesem Fall nur Objekte enthalten, die auch einzeln, über ihren individuellen Index/Subindex, angesprochen werden können.
3. Wie 2., nur erlaubt der Hersteller dem Anwender selbst Parametersätze zu definieren. In diesem Fall ist das Objekt XYZ schreibbar.

Definition des Objekts, das einen Parametersatz enthält:

Index (hex)	Objektname	R/W	Länge	Datentyp (Darstellung)	Bedeutung	M/O
XYZ	ParamSet1	R/W _D	N Byte	Record (3 Elemente)	Parametersatz für ein Beispiel	O
.01	• NoOfObj	R/W _D	2 Byte	UINT16 (Dez)	Number of Objects Anzahl der in diesem Parametersatz zusammengestellten Objekte	O
.02	• Structure	R/W _D	N x 5 Byte	Array of Records (3 Elemente)	Die zusammengefassten Objekte werden hier in der Form, <ul style="list-style-type: none"> • Index (2 Byte), • Subindex (1 Byte), • Länge der Nutzdaten in Bytes (2 Byte) aufgelistet.	O
.03	• UserData	R/W _D	N Byte	Record (N Elemente)	Hier werden die Nutzdaten der oben aufgeführten Objekte hintereinander weg, ohne Trennzeichen, angegeben.	O

7.4.4 Flüchtige und nicht flüchtige Parameter

Viele Objekte enthalten nicht flüchtige Parameter. D. h. sie werden selten, ggf. während des Anlaufs, geschrieben und bleiben auch nach einem PowerDown erhalten.

Es gibt jedoch auch viele Objekte, deren Inhalt sich ständig ändert und deren Daten mit einem PowerDown verloren gehen.

Für beide Fälle ist das Verhalten durch den Namen definiert.

Es gibt daneben, weitaus seltener, Mischformen.

(1) Objekte, deren Parameter bei einem PowerOn definiert sein sollen, jedoch zur Laufzeit ggf. ständig geändert werden.

Solche Parameter müssen in die Grundformen aufgespreizt werden. D. h. es gibt ein Objekt, das die flüchtigen Daten enthält und ein zugehöriges, das den nicht flüchtigen Startwert des flüchtigen Parameters nach PowerOn enthält. Somit ist der Handhabung der üblicherweise verwendeten Flash-Speicher Rechnung getragen.

(2) Objekte, deren Parameter bei einem PowerOn den letzten zur Laufzeit ggf. ständig geänderten Wert behalten sollen, bedürfen einer entsprechenden Speichertechnik, keiner Definition der Handhabung.

7.4.5 Datensicherung

Index (hex)	Objektname	R/W	Länge	Datentyp (Darstellung)	Bedeutung	M/O
E809	BackUpDataCompr	UR/DW	N Byte	Domain-Variable Octet String (Hex)	<p>Komprimierte Daten für Geräte-Backup Das Objekt enthält Daten, die für ein BackUp/Restore des Geräts notwendig sind. Das können alle Daten des Geräts sein, müssen es aber nicht notwendigerweise.</p> <p>Die Struktur dieses Objekts spielt keine Rolle. Sie ist geräte-/herstellerspezifisch.</p> <p>Es muss lediglich sicher gestellt werden, dass, wenn aus einem Gerät der Inhalt dieses Objekts ausgelesen wird, und dieser Inhalt erfolgreich in das Objekt 0xE809 „BackUpDataCompr“ eines Geräts mit demselben DeviceType-Code eingespielt wird, dieses sich dann identisch verhält.</p> <p>Es ist zulässig, dass weniger Daten zum Slave (z. B. neuere FW) übertragen werden, als dieser evtl. erwartet. (z. B. der Inhalt von BackUpDataCompr stammt von einem Gerät mit älterer FW) Aus Kompatibilitätsgründen sollten zusätzlich Daten nur hinten angehängt werden. Die nicht übertragenen Parameter sind dann mit Default-Werten zu belegen.</p>	O

7.4.6 Firmware-Update

Eintrag in Objekt 0x0005 „Capabilities“: „FwUpdt0“

Im Folgenden wird ein Verfahren zum FW-Update eines I/O-Moduls beschrieben. Aus Sicht des I/O-Geräts ist das Verfahren unabhängig von dem der Station überlagerten Bussystem (z. B. Sercos, PROFINET, PROFIBUS, EtherNet/IP™ etc.), da die Kommunikation über den Parameterkanal erfolgt.

Randbedingungen

- Das I/O-Modul muss den Download-Write-Dienst unterstützen (siehe Basisprofil Kap. 6.1).
- Im CommunicationProfile (Objekt 0x000E) ist die Eigenschaft „Up-/Download-Protokoll“ (634, 635, 636 bzw. 637) eingetragen.
- In Capabilities (Objekt 0x0005) ist die Eigenschaft „FwUpdt0“ eingetragen.
- Objekt "InitFwDownload" (0x0045) und Download-Write-Objekt "DeviceFw" (0xE807) sind implementiert.
- Eine Instanz (Bootloader) muss die entsprechenden Prüfungen vornehmen und das FW-Update-File entgegen nehmen.
- Das FW-Update ist im Zustand „RUN“ nicht erlaubt. Der Master stellt sicher, dass vor und während des FW-Updates die Prozessdaten ungültig/auf Ersatzwerte gesetzt werden.
- Die Randbedingungen gelten für ein Gerät mit Mikrocontroller.

Dependency Info:

Wenn das Objekt 0x0045 „InitFwDownload“ implementiert ist, muss auch das Objekt 0xE807 „DeviceFW“ implementiert sein.

Index (hex)	Objektname	R/W	Länge	Datentyp (Darstellung)	Bedeutung	M/O
0045	InitFWDownload	R/W	Max. 58 Byte	Record (4 Elemente)	Firmware-Download-Initialisierung Enthält Informationen für den folgenden FW-Download und versetzt das Modul in den FW-Update-Modus	O
.01	Status	R	1 Byte	UINT8 (Hex)	Statusinformationen bzgl. des FW-Downloads	D
					<ul style="list-style-type: none"> • Bit 0: Ready for Download Dieses Bit wird vom Gerät gesetzt, sobald es bereit ist, eine neue FW über Objekt 0xE807 zu erhalten. Typischerweise wird dieses Bit gesetzt, wenn das Gerät nach der Initialisierung des Updates durch das Kommando 0x01 in die FW-Update-Task gewechselt ist. Das Bit wird mit Erscheinen des Bit 3 „FW-Update successful“ (oder Power-Up) zurückgesetzt. • Bit 1: Device FW invalid Dieses Bit zeigt an, dass das Gerät eine defekte/keine FW beinhaltet. Es wird gesetzt, wenn der Bootloader hochläuft und per Prüfsumme feststellt, dass FW nicht vorhanden oder defekt ist. Das Bit wird mit Erscheinen des Bit 15 „FW-Update successful“ (oder Power-Up) zurück gesetzt • Bit 2: reserviert • Bit 3: FW-Update successful Dieses Bit wird vom Slave nach erfolgreichem FW-Update gesetzt und erst durch Auslesen und einen Neustart zurückgesetzt. So wird sichergestellt, daß der Erfolg des Firmware-Updates auch von verschiedenen Instanzen kontrolliert werden kann. Das Bit wird unabhängig davon immer mit dem Kommando 0x01 „Init FW-Update“ zurückgesetzt. 	
.02	Control	R/W	1 Byte	UINT8 (Hex)	Kommandos bzgl. des FW-Downloads	D
					<ul style="list-style-type: none"> • 0x00 No Action • 0x01 CMD: Init FW-Update I/O-Gerät prüft anhand nachfolgender Felder, ob die FW passt und springt danach ggfs. in die FW-Update-Task Andernfalls wird ein entsprechender Error Type zurückgegeben. • 0x02 CMD: Prepare Restart Der Master schließt mit diesem Kommando das Update ab, es folgen keine weiteren Blöcke mehr. Das I/O-Gerät wartet auf das Signal/Kommando Reset/PowerDown bzw. erfolgt hierdurch automatisch ein Reset des Mikrocontrollers (entsprechend der Verdrahtung) <p>Nach erfolgter Aktion wird der Inhalt selbsttätig auf 0x00 zurückgesetzt.</p>	
.03	HeaderVersion	R/W	2 Byte	UINT16 (Hex)	Version des Headers des FW-Update-Files Dieser wird bei Änderungen am Header hochgezählt High Byte: inkompatible Änderung Low Byte: kompatible Erweiterung	D
.04	UpdateVersion	R/W	2 Byte	UINT16 (Hex)	Version des Update-Verfahrens Wird bei Änderungen am Update-Verfahren hochgezählt High Byte: inkompatible Änderung Low Byte: kompatible Erweiterung	D

.05	BaseData	R/W	N Byte	Octet String (Hex)	Basisdaten, um die Voraussetzungen für ein FW-Update prüfen zu können. Die Informationen sind herstellerspezifisch, sollten aber enthalten: <ul style="list-style-type: none"> • Firmware-Version bzw. -Versionen • HW-Version bzw. -Versionen • Gerätetyp • Vendor-Kennung • Prüfsumme • Reserve für zukünftige Erweiterungen 	D
E807	DeviceFW	UR/DW	N Byte	Domain-Variable Octet String (Hex)	Firmware-File Enthält das Firmware-File	D

Error Types

Das Gerät kann Zugriffe auf die FW-Update-Objekte (0x0045 und 0xE807) ablehnen. Hierzu sind folgende Error Types definiert.

Gerät lehnt die FW-Update-Anforderung per Objekt 0x0045 ab.

Error Class	Error Code	AddInfo	Kommentar
0x08	0x01	0x0022	FW-Update im aktuellen Gerätestatus nicht möglich
0x08	0x01	0x0050	Allgemein
0x08	0x01	0x0051	Major Header-/Update-Version passt nicht
0x08	0x01	0x0052	Firmware passt nicht zur Hardware
0x08	0x0B	0x0000	Gerät ist noch nicht bereit für Neustart (Busy), bei Control Restart Der Master kann ggf. auch Timeout (0x0F12) melden.

Gerät lehnt das FW-Update über Objekt 0xE807 ab

Error Class	Error Code	AddInfo	Kommentar
0x08	0x01	0x0022	Gerät lehnt Schreibzugriff auf Objekt 0xE807 ab, da das FW-Update noch nicht über Objekt 0x0045 gestartet wurde.
0x08	0x01	0x00A4	Gerät kann Segment nicht schreiben, z. B. wegen Problemen beim Flash-Zugriff.
0x06	0x02	0x0080	Gerät kann Segment nicht schreiben, z. B. wegen Problemen beim Flash-Zugriff: Hardware defekt.
0x08	0x01	0x0050	Ein über Objekt 0xE807 geschriebener Block-Type kann vom Gerät nicht verarbeitet werden, z. B die Länge passt nicht
0x08	0x01	0x0053	FW-Block identisch -> überspringe Block

Zusätzlich zu den hier definierten Error Types können die passenden übrigen Error Types (z. B. für das Download-Write-Protokoll) verwendet werden.

Ablauf

Das FW-Update wird über Objekt 0x0045 angestoßen. Das Gerät prüft anhand des Inhalts aus Objekt 0x0045, ob die neue FW geeignet ist und quittiert dies positiv, andernfalls negativ (Error Type s. o.)

Das Gerät wechselt dann in den FW-Update-Modus und sollte diesen Zustandswechsel „Bereit für Update“ als Information per Report über das Objekt 0x0018 „DiagState“ an den Master melden, damit dies z. B. in sein Diagnoseloggbuch eingetragen werden kann.

Danach erhält das Gerät den Firmware-Block per Download-Write auf das Objekt 0xE807. Nach vollständigem Download wird die während des Downloads mitberechnete Prüfsumme (BaseData) verglichen.

Der Master initiiert dann den Neustart des Geräts mit Objekt 0x0045. Das Gerät quittiert dies positiv, wenn alles OK, andernfalls negativ. Wenn alles OK ist, wartet es bis zum Neustart auf ein explizites Signal zum Neustart bzw. es erfolgt hierdurch automatisch ein Neustart des Mikrocontrollers (entsprechend der Verdrahtung). Alternativ kann ein PowerOFF – PowerON durchgeführt werden.

7.4.7 Passwortschutz

Index (hex)	Objektname	R/W	Länge	Datentyp (Darstellung)	Bedeutung	M/O
001D	Password	R/W	Max. 40 Byte	Octet String[40] (Text)	<p>Passwort</p> <p>Mit diesem Kommando wird ein Passwort (z. B. „Superuser“) übertragen, das für die folgenden Aktionen gültig ist. Der Zugriff auf Passwort-gesicherte Objekte ist beim Zugriff mit falschem oder fehlendem Passwort mit Verweis auf die fehlenden Rechte abzulehnen.</p> <p>Das Passwort wird deaktiviert bei:</p> <ul style="list-style-type: none"> • Kommunikationsabbruch • Bus-Stopp (Bus-Reset). • Schreiben keines Passworts (String mit der Länge „0“) • Schreiben eines falschen Passworts <p>Wenn das Objekt gelesen wird, wird 8x“““und 0x00 „0x2A2A2A2A 2A2A2A2A0x00“ zurück gegeben.</p> <p>Für dieses Objekt kann es keinen Zugriffsschutz geben.</p>	O
001E	SetPassword	W	Max. 45 Byte	Record (5 Elemente)	<p>Passwort setzen</p> <p>Mit diesem Objekt werden ein oder mehrere Passwort(e) für einen bestimmten Index gesetzt. Das Passwort wird nur beim Kommando „ResetParam“ wieder gelöscht. Dann gilt wieder das Default-Passwort. Der Hersteller kann Default-mäßig für jedes beliebige Objekt ein Passwort definiert haben. Um Änderungen der Passworte zu verhindern, kann dieses Objekt selbst geschützt werden. Derjenige, der das Objekt „SetPassword“ schützt, ist quasi der „Superuser“.</p> <p>Durch den Hersteller können Passworte voreingestellt worden sein. Diese müssen nicht löschar sein.</p>	O
.01	<ul style="list-style-type: none"> • SubModNo 	W	1 Byte	UINT8 (Hex)	<p>Sub-Modul-Nummer</p> <p>„FF“ als Sub-Modul setzt das Passwort für alle Sub-Module.</p> <p>Wenn das Objekt gelesen wird, wird 0x00 zurückgegeben.</p>	D
.02	<ul style="list-style-type: none"> • Index 	W	2 Byte	UINT16 (Hex)	<p>Index</p> <p>„FFFF“ als Index setzt das Passwort für alle Objekte.</p> <p>Wenn das Objekt gelesen wird, wird 0x0000 zurückgegeben.</p>	D
.03	<ul style="list-style-type: none"> • Subindex 	W	1 Byte	UINT8 (Hex)	<p>Subindex</p> <p>„FF“ als Subindex setzt das Passwort für alle Sub-Objekte.</p> <p>Wenn das Sub-Objekt gelesen wird, wird 0x00 zurückgegeben.</p>	D

.04	• Add/Replace	W	1 Byte	Bit-String 8 (Bin)	Ergänzen/Ersetzen 0x0X = alle vorhandenen Passworte ersetzen 0x1X = zu vorhandenen Passwort hinzufügen 0xX1 = gilt für den lesenden Zugriff 0xX2 = gilt für den schreibenden Zugriff 0xX3 = gilt für den lesenden und schreibenden Zugriff Wenn das Objekt gelesen wird, wird 0x00 zurückgegeben.	D
.05	• Password	W	Max. 40 Byte	Octet String[40] (Text)	Passwort für das Objekt. „Index“ Wird „Ersetzen“ 0x0X gewählt aber kein Passwort angegeben, hebt dies den Passwortschutz auf. Wenn das Objekt gelesen wird, wird 8x****und 0x00 „0x2A2A2A2A 2A2A2A2A0x00“ zurück gegeben.	D

Wenn beim Schreiben/Lesen der Objekte Fehler auftreten, z. B. es werden unzulässige Werte übertragen, so sind die Zugriffe mit den entsprechenden Error Types abzulehnen (siehe: Kapitel „Liste der zulässigen Error Types“).

Error Class: 0x08, Error Code: 0x01

Additional Code (hex)	Bedeutung
00B1	Das Passwort kann nicht ersetzt (gelöscht) werden.
00B2	Das Passwort kann nicht hinzugefügt werden (zu viele Passworte).
00B3	Das Passwort kann für die gewünschte Art des Zugriffs nicht vergeben werden.

7.4.8 Energiemanagement

Eintrag in Objekt 0x0005 „Capabilities“: „Energy0“

Im Objekt 0x0005 Capabilities wird mit dem Eintrag „Energy0“ angezeigt, dass das Gerät ein Energiemanagement unterstützt.

Index (hex)	Objektname	R/W	Länge	Datentyp (Darstellung)	Bedeutung	M/O
003D	WakeUpTime	R	2 Byte	UINT16 (Dez)	<p>Hochlaufzeit Zeit bis zur Betriebsbereitschaft in ms</p> <p>Zeit vom Einschalten der Versorgungsspannung (ggf. auch bei mehreren der letzten relevanten) zur Herstellung der Betriebsbereitschaft des Geräts angegeben in ms.</p> <p>Unter „Betriebsbereitschaft“ wird hier verstanden, dass das Gerät auf <u>alle</u> Kommunikationsanfragen (z.B: das Schreiben von Parametern) seitens des Masters in der definierten Art und Weise reagiert. Grundeinstellungen, das Booten des Kommunikationsprozessors etc. sind abgeschlossen.</p> <p>Die WakeUpTime beinhaltet gewöhnlich <u>keine</u> applikativen Zeiten, wie z. B. Selbsttests, Kalibrieren des Front-ends oder sich aus Abhängigkeiten vom Steuerungsprogramm ergebenden Zeiten.</p> <p>Legacy Information: Wenn dieser Parameter nicht vorhanden ist, wird von einem Default-Wert von < 500ms ausgegangen.</p>	M
003E	EnergyMgmt	R/W	5 Byte	Record (2 Elemente)	<p>Energiemanagement</p> <p>Die für das Energiemanagement entscheidenden Energiesparmodi sind durch den Hersteller zu definieren und können im Bereich 0x01 ... 0x7F frei definiert werden.</p> <p>Hinweis: Das Energiemanagement bezieht sich immer auf die unmittelbare und mittelbare Peripherie des Geräts. Lässt sich diese nicht vom Kommunikationsteil trennen, kann das z. B. bedeuten, dass das ganze Gerät abgeschaltet wird. Dann kann dieses Objekt nicht mehr zurück gelesen werden.</p>	O

.01	<ul style="list-style-type: none"> ActualMode 	R/W	1 Byte	UINT8 (Hex)	<p>Aktueller Energiesparmodus</p> <p>0x00 - PowerOFF 0x01 Mode01 (Standby) 0x02 – 0x0F Reserviert 0x10 – 0x7F Herstellerspezifisch 0x80 – 0x87 Reserviert 0x88 - undefiniert</p> <p>Es wird dem Gerät überlassen, den günstigsten Energiesparmodus (möglicherweise den einzigen) einzunehmen. Das Gerät muss dabei die evtl. vorgegebene Pausenzeit beachten</p> <p>0x89 – 0xFE Reserviert 0xFF - Full Operational</p> <p>Wenn dieses Objekt nicht angelegt ist, wird davon ausgegangen, dass das Modul nur die Modi „0x00“ und „0xFF“ unterstützt.</p> <p>Geschrieben wird der gewünschte Energiesparmodus. Gelesen wird der Energiesparmodus, in dem sich das Gerät aktuell befindet.</p>	D
.02	<ul style="list-style-type: none"> Pause 	R/W	4 Byte	UINT32 (Dez)	<p>Pausenzeit</p> <p>Zeit in Sekunden (sec), die das Gerät im Energiesparmode verweilen soll. Danach wechselt das Gerät selbsttätig wieder in den Mode „0xFF - Full Operational“</p> <p>„0x0000.0000“ - Reserviert „0xFFFF.FFFF“ - Unendlich (Gerät wechselt nicht selbsttätig in einen anderen Energiesparmode)</p> <p>Geschrieben wird die gewünschte Pausenzeit. Gelesen wird die Zeit, die das Gerät noch im Energiesparmode verweilen wird.</p>	D

7.5 Mehrsprachigkeit

Über das Objekt „Language“ kann die aktuell ausgewählte Sprache ausgelesen werden und auch die Sprachauswahl vorgenommen werden.

Die Domain-Variable „LanguageAvailable“ zeigt an, welche Sprachen verfügbar sind. Der erste Eintrag enthält die Default-Sprache.

Index (hex)	Objektname	R/W	Länge	Datentyp (Darstellung)	Bedeutung	M/O	Beispiel
0017	Language	R/W	Max. 65 Byte	Record (2 Elemente)	Sprache Objekt für die Sprachauswahl des Geräts Hier kann die aktuell gültige Sprache ausgelesen und geändert werden.	M	
.01	<ul style="list-style-type: none"> LanguageCode 	R/W	5+1 Byte	Visible String (Text)	Sprachencode Sprachencode nach ISO 639-1 und, optional durch „_“ getrennt, Ländercode nach ISO 3166-1-Alpha-2 code z. B. „en“ bzw. „en-us“ Wenn der Ländercode nicht benutzt wird, wird mit 0x00 aufgefüllt. Der Sprachencode der auszuwählenden Sprache wird hier eingeschrieben. Nach erfolgter pos. Response sind alle Textausgaben in der gewählten Sprache durchzuführen.	M	en
.02	<ul style="list-style-type: none"> NameLanguage 	R	Max. 50 Byte	Visible String (Text)	Sprachename Hier kann der Textstring für die aktuell gültige Sprache ausgelesen werden. Der Name ändert sich, sobald ein neuer „LanguageCode“ eingetragen wurde.	M	English
E804	LanguageAvailable	R		Domain-Variable Record	Verfügbare Sprachen Objekt für die Sprachauswahl des Geräts – Anzeige aller verfügbaren Sprachen Sprachename, schon in der entsprechenden Sprache	O	
Seg.1 -a	<ul style="list-style-type: none"> LanguageCode1 (Default) 	R	5+1	Visible String (Text)	Sprachencode nach ISO 639-1 und, optional durch „_“ getrennt, Ländercode nach ISO 3166-1-Alpha-2 code (Default)	D	en-gb
-b	<ul style="list-style-type: none"> NameLanguage1 (Default) 	R	Max. 50	Visible String (Text)	Textstring für die 1. verfügbare Sprache. (Default)	D	English
Seg.2 -a	<ul style="list-style-type: none"> LanguageCode2 	R	5+1	Visible String (Text)	Sprachencode nach ISO 639-1 und, optional durch „_“ getrennt, Ländercode nach ISO 3166-1-Alpha-2 code der 2. verfügbaren Sprache	O	de
-b	<ul style="list-style-type: none"> NameLanguage2 	R	Max. 50	Visible String (Text)	Textstring für die 2. verfügbare Sprache	O	Deutsch

Seg.3 -a	• LanguageCode3	R	5+1	Visible String (Text)	Sprachencode nach ISO 639-1 und, optional durch “-“ getrennt, Ländercode nach ISO 3166-1-Alpha-2 code der 3. verfügbaren Sprache	O	fr
-b	• NameLanguage3	R	Max. 50	Visible String (Text)	Textstring für die 3. verfügbare Sprache	O	Français
Seg.4 -a	• LanguageCode4	R	5+1	Visible String (Text)	Sprachencode nach ISO 639-1 und, optional durch “-“ getrennt, Ländercode nach ISO 3166-1-Alpha-2 code der 4. verfügbaren Sprache	O	es
-b	• NameLanguage4	R	Max. 50	Visible String (Text)	Textstring für die 4. verfügbare Sprache	O	Español
	...						
	...						
Seg.N -a	• LanguageCodeN	R	5+1	Visible String (Text)	Sprachencode nach ISO 639-1 und, optional durch “-“ getrennt, Ländercode nach ISO 3166-1-Alpha-2 code der N. verfügbaren Sprache	O	it
-b	• NameLanguageN	R	Max. 50	Visible String (Text)	Textstring für die N. verfügbare Sprache.	O	Italiano

Das Sub-Objekt „LanguageCode“ (Subindex 1) enthält die Sprachencodes entsprechend dem Sprachencode nach ISO 639-1 und, optional durch “-“ getrennt, den Ländercode nach ISO 3166-1-Alpha-2 code. Wenn der Ländercode nicht benutzt wird, dann wird er mit 0x00 aufgefüllt.

Durch Schreiben dieses Sub-Objekts wird die Sprache ausgewählt. Diese Sprache taucht dann im Sub-Objekt „NameLanguage“ auf.

Beim schreibenden Zugriff auf das gesamte Objekt „Language“ wird der Eintrag für das Sub-Objekt „NameLanguage“ ignoriert, da das Sub-Objekt „NameLanguage“ den Status „Nur lesen“ hat.

Jedes Segment der Domain-Variable „LanguageAvailable“ enthält genau die beiden Sub-Objekte „LanguageCode und „NameLanguage“ der verfügbaren Sprache. Die einzelnen Read-Dienste des Makrodienstes „Upload-Read“ haben also (fast) immer andere Längen, je ein Spracheintrag wird aber immer komplett in einem Segment übertragen. Dadurch wird eine einfache Handhabung erreicht.

7.6 Modulare Geräte - Subsysteme

Eintrag in Objekt 0x0005 „Capabilities“: „SubMa_0“

7.6.1 Grundlagen

Neben kompakten Geräten gibt es vermehrt auch modulare Geräte. Diese sind dem Feldbus unterlagert und werden deshalb aus Sicht eines übergeordneten Netzwerks auch als Subsystem mit Sub-Modulen bezeichnet. Deren Handhabung soll ebenfalls in diesem Profil berücksichtigt werden. Ein modulares Gerät definiert sich wie folgt:

- Es besitzt genau ein Kommunikationszugangsmodule (Kopf).
- Es besitzt N Module, wobei $0 \leq N < 253$.
Die Modulnummern $N = 253$, $N = 254$ und $N = 255$ sind reserviert.
- Ein Modul muss nicht notwendigerweise aktiv/vorhanden sein.
- Jedes Modul hat seinen eigenen unabhängigen Satz von Objekten.
Dieser Objektraum muss nicht der in diesem Basisprofil beschriebene Objektraum sein, sondern richtet sich nach dem Subsystem.

In diesem Profil ist hier eine Ebene von Geräten unterhalb des Feldbussystems beschrieben. Natürlich kann der Feldbus selbst als modulare Station oder auch als Gateway unter einem anderen Feldbus/Netzwerk ausgeprägt sein, welches ggf. seinen eigenen Regeln und Profilen unterliegt. In diesem Fall spricht man aus der Sicht des überlagerten Feldbus/Netzwerks von einem Subsystem.

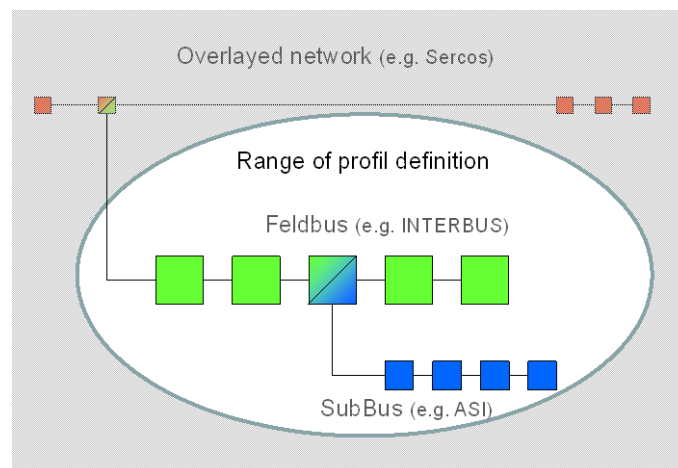


Bild: Definitionsraum im Basisprofil

Wenn ein Gerät ein Sub-Bus-Master ist, dann sind die in diesem Kapitel beschriebenen Mechanismen und Objekte zu implementieren.

7.6.2 Parameter

Auch Sub-Bus-Module verfügen in der Regel über Parameter. Diese können direkt angesprochen werden.

Zur Adressierung von Objekten in den Sub-Modulen eines modularen Geräts wird der Parameter „Modulnummer“ genutzt.

Das ist letztlich das Modul, für das der Zugriff ausgeführt wurde. Somit ist es möglich, die Objekte in den Sub-Modulen über

- Modulnummer
- Index im Sub-Modul
- Subindex im Sub-Modul

direkt, ohne bspw. ein Tunnelprotokoll nutzen zu müssen, anzusprechen.

Das Verhalten für kompakte Geräte ändert sich ebenfalls nicht. Hier könnte weiterhin eine beliebige Modulnummer genutzt werden. Es wird jedoch aus Gründen der Durchgängigkeit festgelegt, als Modulnummer 0x00 zu wählen.

So kann der Kopf eines modularen Geräts immer über die Modulnummer „0“ angesprochen werden; die einzelnen Module über die entsprechende Modulnummer.

Die Adressierung der Module beginnt mit „1“.

Ein kompaktes Gerät soll, wie oben beschrieben, immer mit Modulnummer „0“ angesprochen werden, der Kopf eines modularen Geräts immer mit Modulnummer „0“, die einzelnen Sub-Module N mit $0 < N < 253$.

0xFF ist bei einem modularen Gerät als Modulnummer nicht zulässig, kann aber in einer Diagnosemeldung als Kennzeichnung für das gesamte Subsystem auftauchen.

7.6.3 Diagnose

Um systemische Störungen im modularen System (z. B. Modul fehlt) oder in der Kommunikation melden zu können, sind die Störungscodes 0xA0XX definiert.

Um applikative Störungen von Modulen im modularen System (z. B. Peripheriespannung fehlt) zu melden, sind die Standard-Störungscodes (siehe Kapitel „Störungscodes“) zu benutzen. In diesem Fall kann, wenn vorhanden, eine detailliertere Information aus dem Objekt 0x0018 „DiagState“ des Moduls, welches über die Modulnummer adressiert wird, ausgelesen werden. Ein System- oder Sub-Modul eigener Störungscodes kann im SubObjekt 0x0018.09 „DiagState.AddValue“ abgelegt werden.

Dependency Info:

Wenn das Objekt 0x0035 „SubBusInfo“ implementiert ist, müssen auch alle anderen Objekte dieses Kapitels implementiert werden.

Index (hex)	Objektname	R/W	Länge	Datentyp (Darstellung)	Bedeutung	M/O
0035	SubBusInfo	R	16 Byte	Record (5 Elemente)	Sub-Bus-Information Grundlegende Informationen über den angeschlossenen Sub-Bus.	O
.01	<ul style="list-style-type: none"> SubBusType 	R	8 Byte	Octet String[8]	Bezeichnung des Sub-Busses Zzt. sind definiert: <ul style="list-style-type: none"> 0x4153495F30330000 ASI_03 0x43414E5F30310000 CAN_01 0x44414C495F303100 DALI_01 0x4942535F30310000 IBS_01 0x494F4C5F30310000 IOL_01 0x50425F3031000000 PB_01 Der Type ist genau 8 Zeichen lang. Nicht benutzte Zeichen sind mit 0x00 aufgefüllt.	D
.02	<ul style="list-style-type: none"> MTStructFormat MT-Length PD-IN-Length PD-OUT-Length 	R	3 Byte	Record 1 Eintrag	Feldlängen Feldlängen in den Objekten 0x0036 „ActSubBusStructure“ und 0x0041 „RefSubBusStructure“ Feldlänge von "ModuleType" in Byte (typ.: 4) Feldlänge von "PD-IN-Length" in Byte (typ.: 2) Feldlänge von "PD-OUT-Length" in Byte (typ.: 2)	D

.03	<ul style="list-style-type: none"> ActNoOfModules 	R	1 Byte	UINT8 (Dez)	Anzahl angeschlossener Sub-Bus-Module Anzahl derzeit angeschlossener Sub-Bus-Module = Anzahl der Einträge in „ActSubBusStructure“	D
.04	<ul style="list-style-type: none"> LastMappedModule 	R	1 Byte	UINT8 (Dez)	Letztes direkt abgebildetes Sub-Bus-Modul Letzter über den zyklischen Prozessdatenkanal erreichbares Sub-Bus-Modul (bezogen auf „RefSubBusStructure“)	D
.05	<ul style="list-style-type: none"> RemainingSystem 	R	1 Byte	UINT8 (Dez)	Restsystem Statusinformation darüber, ob aktuell ein Sub-Bus-Restsystem betrieben wird oder nicht. <ul style="list-style-type: none"> 0x00: Es wird derzeit kein Restsystem betrieben. Die Istkonfiguration („ActSubBusStructure“) stimmt mit der Referenzkonfiguration („RefSubBusStructure“) überein. 0x01: Es wird derzeit ein Restsystem betrieben. Voraussetzung: die Option im Objekt 0x0043.3 „Sub-BusBehaviour.Remaining“ = ON ist gewählt und ein oder mehrere Module der Sollkonfiguration („RefSubBusStructure“) sind in der Istkonfiguration („ActSubBusStructure“) nicht vorhanden. 	D
0036	ActSubBusStructure	R	N x typ. 8 Bytes	Array of Records (N x 3 Elemente)	Sub-Sub-Istkonfiguration Hier werden die aktuelle Struktur des modularen Geräts und die Abbildung der Prozessdaten der Module auf die Prozessdaten des Geräts festgelegt. Dabei können der Gerätetyp und Länge der Abbildung der Prozessdaten für jedes Modul unter dem Subindex mit seiner Modulnummer aufgerufen werden. Dieses Objekt eignet sich auch, um kompakte Geräte zu beschreiben. Pflichtobjekt für modulare Geräte.	D
.Na	<ul style="list-style-type: none"> ModuleType 	R	Siehe Objekt 0x0035.2 typ. 4 Byte	UINT32 (Hex)	Gerätetypenidentifikation des N. Moduls, die im Sub-Bus festgelegt ist. Dies kann im einfachsten Fall eine Adressnummer sein. „0“: an dieser Stelle existiert kein Modul „0xFF ... FF“: reserviert.	D
.Nb	<ul style="list-style-type: none"> PDINLength 	R	2 Byte	UINT16 (Dez)	Länge in Bit im Eingangs-Prozessdatenkanal des N. Moduls	D
.Nc	<ul style="list-style-type: none"> PDOOUTLength 	R	2 Byte	UINT16 (Dez)	Länge in Bit im Ausgangs-Prozessdatenkanal des N. Moduls	D

0041	RefSubBusStructure	R/ W	N x typ. 8 Bytes	Array of Records (N x 3 Elemente)	Sub-Bus-Sollkonfiguration Hier werden die Vorgabestruktur des modularen Geräts und die Abbildung der Prozessdaten der Module auf die Prozessdaten des Geräts festgelegt. Dabei können der Gerätetyp und die Länge der Abbildung der Prozessdaten für jedes Modul unter dem Subindex mit seiner Modulnummer aufgerufen werden. Dieses Objekt eignet sich auch, um kompakte Geräte zu beschreiben. Pflichtobjekt für modulare Geräte.	D
.Na	<ul style="list-style-type: none"> ModuleType 	R/ W	Siehe Objekt 0x0035. 2 typ. 4 Byte	UINT32 (Hex)	Gerätetypenidentifikation des N. Moduls, die im Sub-Bus festgelegt ist. Dies kann im einfachsten Fall eine Adressnummer sein. „0“: an dieser Stelle existiert kein Modul „0xFF..FF“: reserviert.	D
.Nb	<ul style="list-style-type: none"> PDINLength 	R/ W	2 Byte	UINT16 (Dez)	Länge in Bit im Eingangs- Prozessdatenkanal des N. Moduls	D
.Nc	<ul style="list-style-type: none"> PDOOUTLength 	R/ W	2 Byte	UINT16 (Dez)	Länge in Bit im Ausgangs- Prozessdatenkanal des N. Moduls	D
0042	ModuleStatus	R	N x 1 Byte	Array of Bit-Strings[8]	Modulstatus im Sub-Bus Hier wird bezogen auf die Sollkonfiguration „RefSubBusStructure“ festgehalten, welchen Status ein Modul momentan hat. Bit0 = 1: Modul meldet einen Fehler Bit1 = 1: Modul meldet eine Warnung Bit2 = 1: Modul meldet eine Information Bit7 = 0: Modul ist nicht erreichbar = 1: Modul ist erreichbar	D
.01	<ul style="list-style-type: none"> Module1 	R	1 Byte	UINT8 (Hex)		
.02	<ul style="list-style-type: none"> Module2 	R	1 Byte	UINT8 (Hex)		
	<ul style="list-style-type: none"> 	R	1 Byte	UINT8 (Hex)		
.N	<ul style="list-style-type: none"> ModuleN 	R	1 Byte	UINT8 (Hex)		
0043	SubBusBehaviour	R	3 Byte	Record (3 Elemente)	Sub-Bus-Verhalten Definition des Verhaltens des Sub-Bus- Systems	D

.01	<ul style="list-style-type: none"> AutoSetup 	R/ W	1 Byte	UINT8 (Hex)	<p>Automatischer Anlauf Nach der Inbetriebnahme des Sub-Bus-Masters kann der angeschlossene Sub-Bus auf unterschiedliche Art und Weise starten:</p> <ul style="list-style-type: none"> 0x00 - OFF Kein Konfigurationsvergleich, keine Parametrierung des Sub-Busses. Der angeschlossene Sub-Bus wird so weit wie möglich in Betrieb genommen. 0x01 – ON (Default) Nach erfolgreichem Konfigurationsvergleich und ggf. Parametrierung des Sub-Busses, wird dieser automatisch in Betrieb genommen. <p>Die Ausgänge bleiben im Einschalt-/Ersatzwertverhalten. Die Eingangsdaten sollten gültig sein. Der Sub-Bus-Master wartet auf die entsprechenden Aktivitäten im überlagerten System, um die Ausgänge zu aktivieren.</p>	
.02	<ul style="list-style-type: none"> AutoReStart 	R/ W	1 Byte	UINT8 (Hex)	<p>Automatischer Wiederanlauf</p> <ul style="list-style-type: none"> 0x00 - OFF Nach einem Busfehler wird zur Wiederinbetriebnahme auf das Kommando 0x03 „Restart“ des Objekts SubBusControlCMD gewartet. 0x01 - ON (Default) Sobald eine zulässige Konfiguration erkannt wird (siehe auch „Remaining“), wird der Sub-Bus wieder aktiviert und ggf. auch parametriert. 	
.03	<ul style="list-style-type: none"> Remaining 	R/ W	1 Byte	UINT8 (Hex)	<p>Restsystem</p> <ul style="list-style-type: none"> 0x00 - OFF Alle Sub-Module müssen erreichbar sein, um beim Anlauf oder nach einem Fehler in Betrieb zu gehen. 0x01 - ON (Default) Der Sub-Bus wird beim Anlauf oder nach einem Fehler mit den erreichbaren Sub-Modulen (Restsystem) in Betrieb genommen. 	
0044	SubBusControl	R/ W	1 Byte	Record 1 Elemente	Steuerung des Sub-Busses	D

.01	<ul style="list-style-type: none"> SubBusControlCMD 	R/ W	1 Byte	UINT8 (Hex)	Sub-Bus Kontrolle <ul style="list-style-type: none"> 0x00 – „NoAction“: keine Aktivität 0x01 – „Reconfigure“: Gespeicherte Sollkonfiguration (RefSubBusStructure) löschen, Istkonfiguration (ActSubBusStructure) neu ermitteln und ggf. fixieren 0x02 – „SaveConfig“: Istkonfiguration als Sollkonfiguration übernehmen. 0x03 – „Restart“: Sub-Bus-Restart, Erneutes Starten des Sub-Busses nach einem Busfehler, wenn kein „AutoReStart“ in Objekt 0x0043 konfiguriert ist. 0x04 – „Stop“: Stopp bzw. Reset des Sub-Busses Wiederinbetriebnahme per Kommando „Restart“ oder „Reconfigure“ 	D
C000 - C07F	ProjBasProf	Siehe Objekt im Basisprofil		Projektion des Basisprofils in die Sub-Bus-Module	D	
		<p>Auch für Sub-Module soll eine generische Abbildung der durch die Objekte des Basisprofil beschriebenen Eigenschaften möglich sein. Da die notwendigen Daten in den integrierten Subsystemen an beliebigen Stellen liegen können, werden (soweit dies möglich und sinnvoll ist) die Inhalte der Objekte des Basisprofils in diesen Index-Bereich in den Adressraum des Sub-Bus-Masters projiziert.</p> <p>Da die Sub-Bus-Nummer für diese Objekte nicht genutzt werden kann, wird diese auf den Subindex abgebildet.</p> <p><i>Beispiel:</i> <i>Die Seriennummer (0x0008 im Basisprofil) des 3 Sub-Moduls würde man, so diese existiert, generisch unter</i> <i>Index: 0xC008.03,</i> <i>erreichen, auch wenn diese an einer Sub-Modul-spezifischen Stelle ein weiteres Mal gefunden werden kann.</i></p> <p>Die Einschränkung, daß die Subobjekte des Basisprofils nicht mehr adressierbar sind, wird an dieser Stelle akzeptiert, da sie aufgrund der Verwendung des Subindex = 0x00 trotzdem vollständig abgebildet werden können.</p> <p>Existiert das zu projizierende Objekt im Sub-Bus-Modul nicht, so wird der</p> <ul style="list-style-type: none"> Error type 0x0607 (Unter diesem Index/Subindex existiert kein Objekt) Additional Code 0x00A1 (Ressource nicht verfügbar) erzeugt.				
.01 .. .FF	Proj<Objekname>	Siehe Objekt im Basisprofil (Subindex = 0x00)		Projizierte Objekte des Basisprofils	D	

In den o. g. Strukturen kann es durch nicht angelegte Module bei steckplatzorientierten Systemen „Lücken“ geben. Da der Zugriff auf die Objekte sowohl als Ganzes als auch über den Subindex möglich ist, muss der dann leere Datensatz trotzdem ausgeprägt werden. Die Inhalte werden dann mit „0“ gefüllt bzw. auf „nicht aktiv“ gesetzt.

7.7 Objektbeschreibung

Im Inbetriebnahme- und Servicefall ist es nicht nur wichtig, die Soll-Parametrierung zu kennen, sondern es ist notwendig, die Ist-Parametrierung des Geräts zu kennen.

Voraussetzung dafür ist die Kenntnis der vorhandenen Anwenderobjekte.

Diese Objekte und deren Bedeutungen können mit den folgenden beiden Profilobjekten ausgelesen werden, die sich gegenseitig bedingen.

Dependency Info:

Alle nicht über das Basisprofil definierten Mandatory-Objekte müssen über die Objekte 0x0038 „ObjDescrReq“ und 0x0039 „ObjDescr“ beschrieben werden.

Index (hex)	Objektname	R/W	Länge	Datentyp	Bedeutung	M/O
0038	ObjDescrReq	R/W	3 Byte	Record (2 Elemente)	Anfrage Objektbeschreibung Objekt, dessen Beschreibung angefragt wird.	D
.01	• Index	R/W	2 Byte	UINT16 (Hex)	Index	D
.02	• Subindex	R/W	1 Byte	UINT8 (Hex)	Subindex	D
0039	ObjDescr	R/W _D	Max 58 Byte	Record (16. Elemente)	Objektbeschreibung Beschreibung des Objekts, dessen Index angefragt wurde. Selbstinkrementierend zum nächsten gültigen Index/Subindex beim nächsten Zugriff. Details s. u.	D
.01	• Index	R	2 Byte	UINT16 (Hex)	Index Objektnummer Z. B. 0x6051	D
.02	• Subindex	R	1 Byte	UINT8 (Hex)	Subindex Substrukturnummer eines Objekts Z. B. 0x00	D
.03	• ObjectCode	R	1 Byte	UINT8 (Hex)	Objektcode Wie unter „Datenobjekte“ definiert Z. B. 0x07 - Simple-Variable Auswertung bei komplexen Datenobjekten s. u.	D
.04	• IndexOfType	R	1 Byte	UINT8 (Hex)	Datentypcode Wie unter „Datentypen“ definiert Z. B. 0x0A – Octet String	D
.05	• Length	R	1 Byte	UINT8 (Dez)	Länge des (Sub-)Objekts Angabe in Byte Z. B. 0x04	D
.06	• UnitText	R	5+1 Byte	Visible String (Text)	Einheit des Werts - 0x00 terminiert Z. B. U/min	D
.07	• UnitCode	R	1 Byte	UINT8 (Hex)	Einheiten-Größencode des Werts lt. Tabelle „Indizes der physikalischen Größen“ Z. B. 0x11	D
.08	• UnitCodeExp	R	1 Byte	INT8 (Hex)	Einheiten-Exponentenindex lt. Tabelle „Indizes der physikalischen Größen“ In Abhängigkeit des „UnitCode“ kann der „UnitCodeExp“ entweder einen Exponentencode oder eine Sonderbedeutung enthalten.	D

		<p>Die Sonderbedeutung muss aus der Tabelle „Indizes der physikalischen Größen“ ermittelt werden. Alle Codes mit Sonderbedeutung sind größer als +64. Bei diesen Einheiten handelt es sich z. B. um die Einheiten Tag, Stunde, Minute oder um nicht SI-kompatible Einheiten, wie Grad Fahrenheit. z. B. 95</p> <p>Für SI-Einheiten gibt der Einheitenindex den Einheiten-Exponentencode, also die Zehnerpotenz des Einheitenwerts an. Der Einheiten-Exponentencode hat den Wertebereich -127 bis 128.</p> <p>Bsp.</p> <ul style="list-style-type: none"> • Einheitenindex 0 für 10^0 • Einheitenindex 3 für 10^3 • Einheitenindex -3 für 10^{-3} <p>usw.</p> <p>Die aufgelisteten Einheiten-Exponentencodes für SI-kompatible Einheiten (Einheiten-Exponentencode < 64) sollen nur als Beispiel dienen. Die Einheitenindizes für weitere SI-kompatible Vorsätze (Pico usw.) ergeben sich entsprechend.</p>				
.09	• Offset	R	2 Byte	INT16 (Dez)	<p>Offset</p> <p>Offset des entsprechenden Werts. Dieser muss zu den Prozessdaten hinzu addiert werden, um so, unter Berücksichtigung der Auflösung, den Messwert zu errechnen.</p> <p>Bsp:</p> <p>Der Prozessdatenwert ist 12345, der Offset ist -15000, daraus ergibt sich bei der unten benutzten Auflösung ein Messwert von -8,84 V</p>	D
.0A	• RDR	R	2 Byte	UINT16 (Dez)	<p>ResolutionDimensionRange</p> <p>Dimensionsbereich der Auflösung</p> $\text{Resolution} = \frac{RDR}{RNR}$ <p>Dimensionsbereich der Auflösung, bezogen auf die Einheit, in der die Größe dargestellt wird, in dem sich das Prozessdatum bewegt. Wird zur Berechnung der Auflösung herangezogen.</p> <p>Bsp: 10 (Volt)</p> <p>Auflösung = $10 \text{ V} / 3000 = 3,333 \text{ mV}$</p> <p>Default ist hier „1“.</p>	D
.0B	• RNR	R	2 Byte	UINT16 (Dez)	<p>ResolutionNumberRange</p> <p>Zahlenbereich der Auflösung</p> $\text{Resolution} = \frac{RDR}{RNR}$ <p>Zahlenbereich der Auflösung, bezogen auf den Dimensionsbereich, in dem sich das Prozessdatum bewegt. Wird zur Berechnung der Auflösung herangezogen.</p> <p>Bsp: 3000</p> <p>Auflösung = $10 \text{ V} / 3000 = 3,333 \text{ mV}$</p> <p>Default ist hier „1“.</p>	D

.0C	• Access Rights	R	1 Byte	Bit-String 8 (Bin)	<p>Zugriffsrechte</p> <p>Bit 0 = ReadAccess (Leserechte) 1 = Lesen erlaubt 0 = Lesen nicht erlaubt</p> <p>Bit 1 = WriteAccess (Schreibrechte) 1 = Schreiben erlaubt, 0 = Schreiben nicht erlaubt</p> <p>Bit 2 = Read_WithPassword (Bit 0 = „0“) 1 = nach Setzen des Passworts Lesen erlaubt, 0 = Leserechte sind unabhängig vom Passwort</p> <p>Bit 3 = Write_WithPassword (Bit 0 = „0“) 1 = nach Setzen des Passworts Schreiben erlaubt, 0 = Schreibrechte sind unabhängig vom Passwort</p> <p>Mögliche Kombinationen Bit [3 ... 0]: Bit [3210] [0001] = read only [0010] = write only, (aber nur selten sinnvoll) [0011] = (read & write) allowed [0110] = write allowed & read allowed after setting password [1001] = read allowed & write allowed after setting password [1100] = (read & write) allowed after setting password</p> <p>Bit 4 = Avoid Presentation Der Inhalt des Objekts soll in Tools/im Engineering 1 = nicht angezeigt werden 0 = angezeigt werden</p> <p>Bit 5-7 Reserved</p>	D
.0D	• DisplayFormat	R	1 Byte	UINT8 (Hex)	<p>Anzeigeformat</p> <p>0x00 - Undefined 0x01 - Binary 0x02 - Unsigned Decimal 0x03 - Signed Decimal 0x04 - Hexadecimal 0x05 - Text 0x06 - Float 0x07 - Time 0x08 - Date 0x09 - 0xFF Reserved</p>	D
.0E	• Min	R	“Length” Byte	Nach Datentyp	<p>Untere Grenze</p> <p>Untere zulässige Grenze eines Werts. Die Länge richtet sich nach 0x0039.5 “ObjDescr.Length”. Z. B. -500</p> <p>Für alle String-Variablen gibt es keinen Min-Wert. Die Länge ist in diesem Fall 0.</p>	D
.0F	• Max	R	“Length” Byte	Nach Datentyp	<p>Obere Grenze</p> <p>Obere zulässige Grenze eines Werts. Die Länge richtet sich nach 0x0039.5 “ObjDescr.Length”. Z. B. 500</p> <p>Für alle String-Variablen gibt es keinen Min-Wert. Die Länge ist in diesem Fall 0.</p>	D

.10	• Symbol	R	M+1 Byte	Visible String (Text)	Name/Bezeichnung des Objekts Die Länge des Namens kann bis max. PDU-Size aufgefüllt werden. Z. B. Solldrehzahl Das Objekt ist 0x00 terminiert. <u>Anmerkung:</u> Um die einzelnen Bits eines Bit-Strings benamen zu können, wird für diesen Fall folgender Aufbau des Symbols festgelegt: <Bezeichnung des Bit-Strings>; <Name Bit0>; <Name Bit1>; .. <Name BitN> Als Trennzeichen ist „;“ (Semikolon) festgelegt. Nach dem Namen des letzten Bits folgt 0x00. Die Bezeichnung eines nicht genutzten Bits kann mit „;;“ (2xSemikolon) übersprungen werden. Weitere entsprechend.	D
E805	ObjDescrLong	R	N Segmente	Domain-Variable Record	Objektbeschreibung (Langform) Beschreibung aller vorhandener Objekte (Index/Subindex) Pro Segment wird eine Beschreibung übertragen. Strings werden mit 0x00 aufgefüllt.	O
Seg.1	ObjDescr1	R	57+1 Byte	Record	Siehe oben	D
...	
Seg.N	ObjDescr1	R	57+1 Byte	Record	Siehe oben	D

Beispiel für den Inhalt des Objekts 0x0039:

Index:0x0039; Subindex: 0; Data: [1]6051 [2]00 [3]07 [4]0A [5]04 [6]552F6D696E00 [7]11 [8]00 [9]C568 [10]000A [11]0BB8 [12]0E [13]05 [14]FFFFFF0C [15]000001F4 [16]536F6C6C647265687A61686C00

Das Objekt 0x0039 ist autoinkrementierend. D. h. nach einem Lesezugriff wird beim folgenden Lesezugriff immer die Beschreibung des nächsten vorhandenen Index.Subindex zur Verfügung gestellt. Wenn der letzte Subindex eines Objekts ausgelesen ist, so wird beim nächsten Lesen der erste Subindex des nächsten Objekts zurückgegeben, d. h. der Index wird inkrementiert und der Subindex beginnt wieder bei 0.

Wenn der letzte Subindex des letzten Index erreicht ist, so beginnt es wieder von vorn.

Beispiel:

0x0200.2, 0x0200.3, 0x0221.0, 0x0240.0, 0x0240.1, 0x0240.2, 0x0240.3, 0x0240.4, ...
0x0200.0, 0x0200.1, 0x0200.2, 0x0200.3, ...

Durch die Autoinkrementfunktion des Objekts 0x0039 ist es einer Applikation leicht möglich, alle vorhandenen Indizes/Subindizes auszulesen und somit ein vollständiges Abbild aller im Gerät vorhandenen Objekte zu erhalten. Die Applikation liest dafür solange das Objekt 0x0039, bis sich der Index.Subindex wiederholt.

Leerobjekte bzw. nicht vorhandene Indizes und Subindizes werden übersprungen.

Wenn über das Objekt 0x0038 „ObjDescrReq“ ein nicht vorhandener Index/Subindex angefordert wird, dann erfolgt eine negative Confirmation mit dem Error Type Error Class

0x08 (Application), Error Code 0x01 (Das Datum hat einen unzulässigen Wert angenommen), Additional Code 0x0030 (Wertebereich eines Parameters ist verlassen). Im Objekt 0x0039 „ObjDescr“ wird dann die Beschreibung des nächsten vorhandenen Index/Subindex zur Verfügung gestellt.

Für die Objekte des Basisprofils muss keine Beschreibung hinterlegt werden, da diese (inklusive der Subindizes) durch das Basisprofil systemweit bekannt sind. Das Objekt 0x0039 enthält dann lediglich den Index. Die restlichen Einträge fehlen (Länge des Nutzdatums: 2 Byte).

Index (hex)	Objektname	R/W	Länge	Datentyp	Bedeutung	M/O
0039	ObjDescr	R/W _D	2 Byte	Record (1 Elemente)	Objektbeschreibung Beschreibung des Basisprofil-Objekts	O
.01	• Index	R	2 Byte	UINT16 (Hex)	Objektindex Z. B. 0x0007	D

Damit wird das Vorhandensein von optionalen Objekten (inklusive aller Ihrer Subindizes) angezeigt. Ein Basisprofil-Objekt darf deshalb nicht übersprungen werden.

Beispiel:

Ein Gerät besitzt die Objekte: 0x6051, 0x6055, 0x6056, 0x6059. Bis auf das Objekt 0x6056 sind es alles einfache Objekte ohne Sub-Objekte. Für das Objekt 0x6056 existieren die Subindizes 01, 02, 03, 07 und 08.

Wenn über das Objekt 0x0039 die Beschreibung gelesen wird, so würde das durch die Autoinkrementfunktion wie folgt aussehen:

```
Write.req: Index:0x0038; Subindex: 0; Data: 0x6000 00 (Objekt 0x6000.0 wird angefragt)
Read.cnf Nr.1: Index:0x0039; Subindex: 0; Data: [1]6051 [2]00 [3]07 [4]0A [5]04 ... [16]536F6C6C00
Read.cnf Nr.2: Index:0x0039; Subindex: 0; Data: [1]6055 [2]00 [3]07 [4]0A [5]04 ... [16]536F6C00
Read.cnf Nr.3: Index:0x0039; Subindex: 0; Data: [1]6056 [2]00 [3]09
Read.cnf Nr.4: Index:0x0039; Subindex: 0; Data: [1]6056 [2]01 [3]07 [4]03 [5]02 ... [16]506F7765723100
Read.cnf Nr.5: Index:0x0039; Subindex: 0; Data: [1]6056 [2]02 [3]07 [4]03 [5]02 ... [16]506F7765723200
Read.cnf Nr.6: Index:0x0039; Subindex: 0; Data: [1]6056 [2]03 [3]07 [4]03 [5]02 ... [16]506F7765723300
Read.cnf Nr.7: Index:0x0039; Subindex: 0; Data: [1]6056 [2]07 [3]07 [4]07 [5]04 ... [16]4170706C657300
Read.cnf Nr.8: Index:0x0039; Subindex: 0; Data: [1]6056 [2]08 [3]07 [4]06 [5]02 ... [16]506561727300
Read.cnf Nr.9: Index:0x0039; Subindex: 0; Data: [1]6059 [2]00 [3]07 [4]0F [5]01 ... [16]4d794269747300
Read.cnf Nr.10: Index:0x0039; Subindex: 0; Data: [1]0001
Read.cnf Nr.11: Index:0x0039; Subindex: 0; Data: [1]0004
...
Read.cnf Nr.N: Index:0x0039; Subindex: 0; Data: [1]6051 [2]00 [3]07 [4]0A [5]04 ... [16]536F6C6C00
```

Ausnahme

Eine Ausnahme bilden die Prozessdatenobjekte 0x0025 und 0x0026, da deren Vorhandensein und Funktion zwar profilspezifisch, deren Struktur aber herstellerspezifisch ist und nur so bestimmt werden kann. Sie müssen deshalb vollständig beschrieben werden. Dabei müssen auch Prozessdatenbereiche, die in den Objekten 0x003B „PDIN_Descr“ oder 0x003C „PDOUT_Descr“ mit „NC“ gekennzeichnet sind, unter dem entsprechenden Subindex beschrieben werden. Als 0x0039.10 ObjDescr.Symbol ist „unused“ zu verwenden. Wenn eine Prozessdatenrichtung gar nicht benutzt wird, das Objekt 0x0025 bzw. 0x0026 selbst ist aus z. B. technischen Gründen vorhanden, so ist dieses nicht weiter zu berücksichtigen. Der Index wird übersprungen.

Wenn das Objekt 0x0039 selbst nicht vorhanden ist, sind die Prozessdatenobjekte 0x0025 und 0x0026 Default-mäßig als Bit-String der entsprechenden Länge definiert.

Alle anderen Objekte sind zu beschreiben.

R/W_D

Unter bestimmten Bedingungen kann es der Hersteller erlauben, über diesen Mechanismus auch die Beschreibung eines Objekts vorzugeben bzw. die Beschreibung zu ändern. Unterstützt der Hersteller diese Funktionalität, kann das Objekt 0x0039 auch zum Schreiben freigegeben werden. In diesem Falle muss es aber auf jeden Fall durch ein Passwort geschützt sein.

Auswertung bei komplexen Datenobjekten.

Für die komplexen Datenobjekte endet die Auswertung des Objekts 0x0039 „ObjDescr“ nach dem Subindex 0x0039.3 „ObjDescr.ObjectCode“. Alle weiteren Indizes 0x0039.4 und folgende haben für die Datenobjekte in der folgenden Tabelle keine Bedeutung und können deshalb auch nicht ausgewertet werden.

Objectcode	Object
0x02	Domain-Variable
0x08	Array-Variable
0x09	Record-Variable
0x0A	Variablenliste Statisch und dynamisch

Die Beschreibung des gesamten Record/Arrays (Subindex 0) ist nicht vorgesehen. Bei Record- und Array-Objekten werden nur die einzelnen Elemente (über Subindex) beschrieben.

Tabelle: Indizes der physikalischen Größen

Physikalische Größe	Größenindex (UnitCode)	Einheit	Einheiten-Exponentenindex (UnitCodeExp) (nicht vollständig)
	0	Dimensionslos	0
Länge (l)	1	Kilometer (km)	3
Länge (l)	1	Meter (m)	0
Länge (l)	1	Millimeter (mm)	-3
Fläche (A)	2	Quadratmeter (m ²)	0
Volumen (V)	3	Kubikmeter (m ³)	0
		Liter (l)	-3
Zeit (t)	4	Sekunde (s)	0
		Minute (min)	70
		Stunde (h)	74
		Tag (d)	77
		Halbwellen	78
		Millisekunde (ms)	-3
		Mikrosekunde (µs)	-6
Kraft (F)	6	Newton (N)	0
Blindleistung (Q)	8	Var (var)	0
(Wirk-) Leistung (P)	9	Watt (W)	0
Scheinleistung (S)	10	Voltampere (VA)	0
Drehzahl (n)	11	Umdrehungen/Sekunde (U/s)	0
		Umdrehungen/Minute (U/min)	73
		Umdrehungen/Stunde (U/h)	74
Winkel ($\alpha, \beta, \gamma, \dots$)	12	Radian (rad)	0
		Sekunde (")	75
		Minute (')	76
		(Alt-)Grad (°)	77
		Neugrad (gon)	78
Geschwindigkeit (v)	13	Meter/Sekunde (m/s)	0
		Millimeter/Sek. (mm/s)	-3
		Millimeter/Min. (mm/min)	79
		Meter/Min. (m/min)	80
		Kilometer/Min. (km/min)	81
		Millimeter/Std. (mm/h)	82
		Meter/Std. (m/h)	83
		Kilometer/Std. (km/h)	84
Drehmoment (M)	16	Newtonmeter (Nm)	0
Temperatur (T)	17	Kelvin (K)	0
		Grad_Celsius (°C)	94
		Grad_Fahrenheit (°F)	95
Elektrische Spannung (U)	21	Volt (V)	0
Elektrischer Strom (I)	22	Ampere (A)	0
Elektrischer Widerstand (R)	23	Ohm (Ω)	0
Verhältnis	24	Prozent (%)	0
Frequenz (f)	28	Hertz (Hz)	0
Schritte	32	Schritte	0
Encoderauflösung	33	Schritte/Umdrehung	0
Durchfluß (Q)	34	Kubikmeter/Sekunde (m ³ /s)	0
		Liter/Sekunde (l/s)	-3
		Milliliter/Sekunde (ml/s)	-6
		Kubikmeter/Minute (m ³ /min)	85
		Liter/Minute (l/min)	86
		Milliliter/Minute (ml/min)	87
Beschleunigung (a)	35	Meter/Sekunde ² (m/s ²)	0
Druck (p)	40	Pascal (Pa)	0
		Bar (bar)	5
		Technische Atmosphäre (at)	85
Wirkungsgrad (η)	41	Dimensionslos	0
(Wirk-) Leistungsfaktor ($\cos \varphi$)	42	Dimensionslos	0
Arbeit/Energie (W/E)	44	Joule (J)	0

Der Einheiten-Exponentenindex entspricht im Standardfall dem Exponenten (zur Basis 10):

z. B.

UnitCodeExp 3 für $10^3 \Rightarrow$ k (Kilo-)

UnitCodeExp 0 für $10^0 \Rightarrow$ ohne

UnitCodeExp -3 für $10^{-3} \Rightarrow$ m (Milli-)

UnitCodeExp -6 für $10^{-6} \Rightarrow$ μ (Miko-)

UnitCodeExp -9 für $10^{-9} \Rightarrow$ n (Nano-)

7.8 OnBoard Gerätebeschreibung

Jedes Gerät wird durch seine digitale Entsprechung, die Gerätebeschreibungsdatei, im Engineering des Zielsystems repräsentiert. Es gibt jedoch nicht nur eine Gerätebeschreibungsdatei, da jedes System sein eigenes Format definiert hat.

Um auch eine Online-Parametrierung ohne systemspezifisches Engineering-System mit Hersteller-Tools zu ermöglichen, wird auf dem Gerät, entsprechend seinen Ressourcen, eine reduzierte und komprimierte OnBoard Gerätebeschreibungsdatei abgelegt. Mit deren Hilfe kann z. B. ein Webbrowser eine adäquate Darstellung der Parameter vornehmen.

Index (hex)	Objektname	R/W	Länge	Datentyp (Darstellung)	Bedeutung	M/O
E808	OnBoardDeviceDescFile	UR	N Byte	Domain-Variable Octet String (Hex)	onBoard Gerätebeschreibungsdatei (Komprimiert) Sie enthält alle notwendigen Informationen, damit ein Tool, z. B. Webbrowser, eine Parametrierung des Geräts vornehmen kann.	O

8 Handhabung einzelner Parameter durch den Master

8.1 0x002D - ResetParam

Das Objekt 0x002D „ResetParam“ darf vom Master nicht regelmäßig bei einem Start beschrieben werden. Genauso wenig darf sich dieses Objekt in der Gerätebeschreibungdatei befinden.

Das Auslösen der Aktion „Reset Parameter“ beschreibt das Flash des Moduls mit den Default-Werten. Ein zu häufiges Beschreiben des Flash zerstört dieses.

Der Master kann die Aktion „Reset Parameter“ nur dann systemisch auslösen, wenn er ein neues Modul erkennt, welches sich potenziell in einem anderen Aufbau befunden haben könnte.

Die Aktion kann aber durch die Applikation unterdrückt werden, wenn eine bestimmte Parametrierung in ein Gerät außerhalb des Systems eingeschrieben wurde, die erhalten bleiben soll.

Einfache Altgeräte unterstützen dieses Objekt ggf. nicht. Es muss dann aber davon ausgegangen werden, dass alle Parameter für dieses Gerät geschrieben werden.

8.2 Behandlung der Error Types

Der Master muss die Error Types im Sinne der Verfügbarkeit generell möglichst tolerant behandeln.

So können z. B. die folgenden Error Types unter bestimmten Umständen ignoriert werden.

- 0x06, 0x05 „Attribute passen nicht“
- 0x06, 0x07 „Objekt nicht vorhanden“
- 0x08, 0x01 „Wert wird nicht unterstützt“

Das ist erforderlich, um die Verfügbarkeit des Systems zu erhöhen und wird in den allermeisten Fällen umgesetzt, bei denen pauschal eine Aktion ausgelöst werden soll.

Wenn es für die gewünschte Aktion sinnvoll ist, wird als Wert in diesem Fall der Default-Wert angenommen bzw. es wird von einem Default-Verhalten des Geräts ausgegangen.

Bei den folgenden Error Types ist eine endliche Zahl von Wiederholungen (min. 3) sinnvoll.

- 0x05, 0x01 „Object-State-Conflict“
- 0x05, 0x03 „Object-Constrain-Conflict“
- 0x06, 0x0A „Data-not-yet-available“

Damit soll vermieden werden, dass Applikationen aus „formalen Gründen“ nicht verfügbar sind. Eine entsprechende Information bzw. einen Eintrag z. B. in das Diagnosenlogbuch sollte dennoch erfolgen.

8.3 Erweiterungen – Zugriff über Subindex

Der Master muss davon ausgehen, dass auch in der Zukunft einzelne Objekte erweitert werden. Diese Erweiterung kann/wird aber nur durch „Anhängen“ geschehen.

Die Adressierung über weitere Subindizes ist ohne Probleme möglich, weil sie eigenständige (Sub-)Objekte darstellen.

Beim **Lesen** von Objekten über den Subindex „0“ (das ganze Objekt wird gelesen) können mehr Daten empfangen werden als erwartet. Alle bis dato bekannten Elemente haben weiterhin die gleiche Bedeutung. Neue, zusätzliche Elemente können aus Unkenntnis heraus nicht ausgewertet werden. Das sollte aber die Funktion der bekannten Elemente und auch des Geräts generell nicht beeinträchtigen.

Beim **Schreiben** von Objekten über den Subindex „0“ (das ganze Objekt wird geschrieben) wird ein Slave zu viele Daten jedoch abweisen. Der Master muss die Parameter dann für die ältere Version aufbereiten. Zu wenige Daten kann der Slave, wenn es ihm möglich ist, akzeptieren. Damit kann ein neueres Gerät in einer alten Anlage kompatibel betrieben werden.

Eine entsprechende Information bzw. einen Eintrag z. B. in das Diagnosedlogbuch sollte dennoch erfolgen.

9 Anhang A

9.1 Begriffsdefinitionen

Ersatzwerte

Ersatzwerte gibt es sowohl für Eingänge als auch für Ausgänge.

Für Ausgänge sind es die Werte, die anstelle der Prozessdaten an der Peripherie ausgegeben werden, wenn die Prozessdaten nicht mehr gültig sind. Das ist z. B. nach einem Bus-Reset oder einem Prozessdaten-Timeout der Fall.

Für Eingänge sind es die Werte, die anstelle der Applikationsdaten an die Prozessdaten übergeben werden, wenn die Applikation keine gültigen Daten mehr liefern kann. Das ist z. B. nach einem Peripheriespannungsausfall oder einem Mikro-Controller-Ausfall der Fall.

Error Type

Der Error Type ist die Fehlermeldung, die zurückgegeben wird, wenn ein Service nicht ausgeführt werden konnte (negative Confirmation).

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

Error Class	1 Byte, UINT8, hexadezimale Darstellung
Error Code	1 Byte, UINT8, hexadezimale Darstellung
Additional Code	2 Byte, UINT16, hexadezimale Darstellung

Firmware

Firmware ist der Oberbegriff für alle (ohne das Gerät zu öffnen) von außen änderbaren Bestandteile eines Geräts. Eine solche Änderung nennt man meist Update.

Übliche änderbare Bestandteile sind:

- Firmware-Kommunikationsprozessor
- Firmware-Applikationsprozessor
- Bootloader
- FPGA Image
- Usw.

Alle Bestandteile werden als zusammengehörig betrachtet. Ihnen wird gemeinsam eine Version zugeordnet.

Funktionsgruppe

Eine Funktionsgruppe beschreibt die Gerätefunktion. Die Funktion wird durch eine Einheit definiert, die durch Eingänge gesteuert und parametrisiert wird. Außerdem können interne Signale und Parameter auf die Funktion einwirken. Der Ausgang einer Funktionsgruppe kann auf die Eingänge von anderen Funktionen geschaltet oder über den Bus zugänglich gemacht werden.

Eine Funktionsgruppe kann umfassen:

- Eingänge und Ausgänge
- Interne Signale und Variablen
- Aktionen
- Externe Signale
- Parameter (Services, Prozessdaten)
- Kommunikationsobjekte

Geräteparameter

Geräteparameter sind Parameter, die die Identifikation, Diagnose und Funktion eines Geräts nach außen hin über die Kommunikation repräsentieren. Für alle Basisgeräteparameter, also die, die bei allen Geräten gleich sein sollen, sind Default-Werte in diesem Profil festgelegt.

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 die 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.

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 Kommunikationsobjekts).

Kommunikations-Interface

Das Kommunikations-Interface setzt sich aus einem Prozessdatenkanal und einem Parameterkanal zusammen.

Kommunikationsprofil

Im Kommunikationsprofil werden die in der Spezifikation des Übertragungsmediums enthaltenen Freiheitsgrade anwendungs- oder gerätegruppenspezifisch eingeschränkt bzw. klassifiziert. Im Kommunikationsprofil werden Kommunikationsdienste 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. Im Profil werden weiterhin Wertebereiche von Attributen und Parametern eingegrenzt bzw. festgelegt.

Kommunikationsreferenz

Die Kommunikationsreferenz dient zum Adressieren des Kommunikationspartners. 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.

Parameterkanal

Der Parameterkanal ist ein Kommunikationskanal, über den auf alle Kommunikationsobjekte zugegriffen werden kann. Die Parameterkanaldienste erlauben einen quitierten Zugriff auf die Geräteparameter, d. h. der Zugriff auf einen Geräteparameter wird vom Gerät bestätigt.

PDU

Protocol Data Unit. Dateneinheit, in der alle Nutz- und Verwaltungsinformationen eines Protokolls übertragen werden.

PD-PDU

Process Data PDU. Dateneinheit, in der die Prozessdaten eines Protokolls übertragen werden.

Prozessdatenbeschreibung

In der Prozessdatenbeschreibung beschreibt der Hersteller die Struktur und die Bedeutung der Prozessdaten.

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.,

- Ausgangsprozessdaten sind Daten, die vom Steuerungssystem zum Gerät übertragen werden. Dieses liest diese Daten aus dem Prozessdatenkanal und gibt sie je nach Funktion an den Prozess aus.
- Eingangsprozessdaten sind Daten, die vom Gerät zum Steuerungssystem übertragen werden. Das Gerät schreibt diese Daten in den Prozessdatenkanal und überträgt sie damit zum Steuerungssystem.

Terminal

Ein Terminal ist der physikalische Endpunkt eines Signals an einem Gerät. Gewöhnlich ist das eine elektrische Klemmverbindung, an der ein Signaldraht angeschlossen wird.

Zustandsmaschine

Eine Zustandsmaschine ist ein festgelegter Ablauf mit Verzweigungen und Wiederholungen, der schematisch dargestellt ist.

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.

9.2 Symbole und Abkürzungen

Abkürzungen

ALI	Application Layer Interface
con	Confirmation-Primitive
D	Dependent (abhängig von anderen Objekten/bestimmten Eigenschaften. Muss, wenn die Abhängigkeit gegeben ist, implementiert werden)
ind	Indication-Primitive
KBL	Kommunikationsbeziehungsliste
KBL-Header	Kopf der Kommunikationsbeziehungsliste
KR	Kommunikationsreferenz
M	Mandatory (Pflicht, muss in jedem Fall implementiert werden)
MAP	Manufacturing Automation Protocol
O	Optional (Kann, muss aber nicht implementiert werden)
OV	Objektverzeichnis
PCP	Peripherals Communication Protocol
PCh	Parameter Channel
PD	Process Data
PMS	Peripherals Message Specification
R	Read only, dieses Objekt kann nur gelesen werden
R/W	Read/Write, dieses Objekt kann gelesen und geschrieben werden
req	Request-Primitive
res	Response-Primitive
S	Selection, Auswahl des einen oder anderen Parameters
W	Write only, dieses Objekt kann nur geschrieben werden
X	Dieses Objekt soll in Zukunft nicht mehr verwendet werden.

10 Anhang B

10.1 Übersetzungstabelle für die Objektnamen

Index	Subindex	Objektname (Basisprofil)	DE	EN-US	ES	FR	IT	ZH-CN
0001		VendorName	Herstellername	Manufacturer name	Nombre del fabricante	Nom fabricant	Nome produttore	制造商名称
0002		VendorID	Herstellerkennung	Manufacturer ID	Identificación del fabricante	Identification fabricant	Identificazione produttore	制造商ID
0003		VendorText	Herstellertext	Manufacturer text	Texto del fabricante	Texte fabricant	Testo produttore	制造商文本
0004		DeviceFamily	Gerätefamilie	Device range	Familia de dispositivos	Gamme d'appareils	Gruppo dispositivi	设备范围
0005		Capabilities	Geräteeigenschaften	Device properties	Propiedades de los dispositivos	Propriétés de l'appareil	Caratteristiche del dispositivo	设备属性
0006		ProductFamily	Produktfamilie	Product range	Línea de productos	Gamme de produits	Famiglia di prodotti	产品类别
0007		ProductName	Produktname	Product name	Nombre del producto	Nom de produit	Nome del prodotto	产品名称
0008		SerialNo	Seriennummer	Serial number	Número de serie	Numéro de série	Numero serie	序列号
0009		ProductText	Produkttext	Product text	Texto del producto	Texte produit	Testo del prodotto	产品文本
000A		OrderNumber	Artikel-Nr.	Order no.	Código de artículo	Référence	Cod. art.	订货号
000B		HardwareVersion	Hardware-Version	Hardware version	Versión de hardware	Version matériel	Versione hardware	硬件版本
	.01	BuildDate	Herstelldatum	Manufacturing date	Fecha de fabricación	Date de fabrication	Data di produzione	制造日期
	.02	Version	Versionskennung	Version ID	Identificación de la versión	Identifiant de version	Riconoscimento versione	版本ID
000C		FirmwareVersion	Firmware-Version	Firmware version	Versión de firmware	Version firmware	Versione firmware	固件版本
	.01	BuildDate	Herstelldatum	Manufacturing date	Fecha de fabricación	Date de fabrication	Data di produzione	制造日期
	.02	Version	Versionskennung	Version ID	Identificación de la versión	Identifiant de version	Riconoscimento versione	版本ID
000D		PChVersion	Parameterkanalversion	Parameter channel version	Versión de canal de parámetros	Version canal de paramètres	Versione canale parametro	参数通道版本
	.01	BuildDate	Herstelldatum	Manufacturing date	Fecha de fabricación	Date de fabrication	Data di produzione	制造日期
	.02	Version	Versionskennung	Version ID	Identificación de la versión	Identifiant de version	Riconoscimento versione	版本ID
000E		CommProfile	Kommunikationsprofil	Communication profile	Perfil de comunicación	Profil de communication	Protocollo di comunicazione	通讯配置文件

Index	Subindex	Objektname (Basisprofil)	DE	EN-US	ES	FR	IT	ZH-CN
000F		DeviceProfile	Geräteprofil	Device profile	Perfil del dispositivo	Profil d'appareil	Profilo del dispositivo	设备配置文件
0010		Reserved	Reserviert	Reserved	Reservado	Réservé	Riservato	保留
0011		ProfileVersion	Profilversion	Profile version	Versión de perfil	Version profil	Versione profilo	配置文件版本
	.01	BuildDate	Herstelldatum	Manufacturing date	Fecha de fabricación	Date de fabrication	Data di produzione	制造日期
	.02	Version	Versionskennung	Version ID	Identificación de la versión	Identifiant de version	Riconoscimento versione	版本ID
0012		VendorURL	Hersteller-URL	Manufacturer URL	URL del fabricante	URL fabricant	URL produttore	制造商URL
0013		OnBoardDeviceDescFileName	Name der OnBoard Gerätebeschreibung sdatei	Name of the onboard device description file	Nombre del archivo de descripción del dispositivo Onboard	Nom du fichier de description de l'appareil embarqué	Nome del file di descrizione del dispositivo OnBoard	机载设备描述文件名称
0014		Location	Einbauort	Installation location	Lugar de montaje	Emplacement de montage	Luogo di installazione	安装位置
0015		EquipmentIdent	Betriebsmittelkennzeichen	Equipment ID	Identificador de equipo	Identification équipement	Caratteristiche apparecchiatura	设备ID
0016		ApplDeviceAddr	Applikationsspezifische Geräteadresse	Application-specific device address	Dirección de dispositivo específica de la aplicación	Adresse appareil spécifique à l'application	Indirizzo dispositivo specifico per applicazione	应用特定的设备地址
0017		Language	Sprache	Language	Idioma	Langue	Lingua	语言
	.01	LanguageCode	Sprachencode	Language code	Código de idioma	Code de langue	Codice lingua	语言代码
	.02	NameLanguage	Sprachename	Language name	Nombre de idioma	Nom de la langue	Nome lingua	语言名称
0018		DiagState	Diagnosezustand	Diagnostic state	Estado de diagnóstico	Etat de diagnostic	Stato diagnosi	诊断状态
	.01	Lfd.Nr.	Laufende Nummer	Consecutive number	Número de orden	Numéro d'ordre	Numero progressivo	连续号
	.02	Priority	Priorität	Priority	Prioridad	Priorité	Priorità	优先级
	.03	Channel	Kanal	Channel	Canal	Canal	Canale	通道
	.04	Code	Störungscode	Error code	Código de fallo	Code de défaut	Codice di guasto	错误代码
	.05	MoreFollows	Zusatzinformationen	Additional information	Información adicional	Informations complémentaires	Informazioni supplementari	更多信息
	.06	Reserved	Reserviert	Reserved	Reservado	Réservé	Riservato	保留
	.07	SubModNo	Sub-Modul-Nummer	Submodule number	Número de submódulo	Numéro sous-module	Numero di sottomodulo	子模块号
	.08	FunctionGroup	Funktionsgruppe	Function group	Grupos funcionales	Groupe fonctionnel	Gruppo funzionale	功能组
	.09	AddValue	Zusatzinformation	Additional information	Información adicional	Information complémentaire	Informazione supplementare	更多信息
	.0A	TextLength	Textlänge	Text length	Longitud del texto	Longueur de texte	Lunghezza del testo	文本长度
	.0B	Text	Diagnosetext	Diagnostic text	Texto de diagnóstico	Texte de diagnostic	Testo di diagnosi	诊断文本

Index	Subindex	Objektname (Basisprofil)	DE	EN-US	ES	FR	IT	ZH-CN
0019		ResetDiag	Diagnosemeldungen quittieren	Acknowledge diagnostic message	Acusar recibo de mensajes de diagnóstico	Acquitter messages de diagnostic	Conferma messaggi di diagnosi	确认诊断信息
001A		GetError ClassRepMethod	Meldemethode für Störung	Error reporting method	Método de mensaje para el fallo	Méthode de signalisation des défauts	Metodo di segnalazione in caso di guasto	错误报告方法
001B		TestMode	Testbetrieb	Test mode	Modo de prueba	Mode d'essai	Funzionamento di prova	测试模式
001C		ControlTrace	Steuerung der Protokollierung	Logging control	Sistema de control del registro	Commande de l'établissement de protocoles	Controllo della registrazione	记录控制
001D		Password	Passwort	Password	Contraseña	Mot de passe	Password	密码
001E		SetPassword	Passwort setzen	Set password	Establecer contraseña	Définir mot de passe	Imposta password	设置密码
	.01	SubModNo	Sub-Modul-Nummer	Submodule number	Número de submódulo	Numéro sous- module	Numero di sottomodulo	子模块号
	.02	Index	Index	Index	Índice	Index	Indice	索引
	.03	Subindex	Subindex	Subindex	Subíndice	Sous-index	Indice subordinato	子索引
	.04	Add/Replace	Ergänzen/Ersetzen	Add/replace	Completar/reemplazar	Compléter/remplacer	Completa/Sostituisci	添加/替换
	.05	Passwort	Passwort	Password	Contraseña	Mot de passe	Password	密码
001F		PDTimeout	Prozessdaten- Überwachungszeit	Process data monitoring time	Tiempo de monitorización de datos de proceso	Temps de surveillance données de process	Tempo di monitoraggio dei dati di processo	过程数据监控时间
0020		PDTimeoutCode	Prozessdaten- Überwachungscode	Process data monitoring code	Código de monitorización de datos de proceso	Temps de surveillance données de process	Codice di monitoraggio dei dati di processo	过程数据监控代码
0021		PChTimeout	Parameterkanal- Überwachungszeit	Parameter channel monitoring time	Tiempo de monitorización de canal de parámetros	Temps de surveillance canal de paramètres	Tempo di monitoraggio del canale parametri	参数通道监控时间
0022		PChTimeoutCode	Parameterkanal- Überwachungscode	Parameter channel monitoring code	Código de monitorización de canal de parámetros	Code de surveillance canal de paramètres	Codice di monitoraggio del canale parametri	参数通道监控代码
0023		AbortCode	Verbindungsabbruchcode	Connection abort code	Código de interrupción de conexión	Code de rupture de connexion	Codice di interruzione collegamento	连接中断代码
0024		ResetCode	Ersatzwertverhalten bei Bus-Reset (PDOOUT)	Substitute value behavior during bus reset (PDOOUT)	Comportamiento de valor sustitutivo en reset de bus (PDOOUT)	Comportement des valeurs de substitution avec RAZ de bus (PDOOUT)	Andamento del valore sostitutivo in caso di reset del bus (PDOOUT)	总线复位期间的替换值特性 (PDOOUT)
0025		PDIN	Eingangsprozessdaten	IN process data	Datos de proceso de entrada	Données d'entrée de process	Dati di processo in ingresso	IN过程数据

Index	Subindex	Objektname (Basisprofil)	DE	EN-US	ES	FR	IT	ZH-CN
	.010N	Part N	N-ter Teil des Prozessdatums	Nth part of process data item	Parte nº n de datos de proceso	Nième partie de la donnée de process	Parte Nesima della data di processo	过程数据项的第N部分
0026		PDOOUT	Ausgangsprozessdaten	OUT process data	Datos de proceso de salida	Données de sortie de process	Dati di processo in uscita	OUT过程数据
	.010N	Part N	N-ter Teil des Prozessdatums	Nth part of process data item	Parte nº n de datos de proceso	Nième partie de la donnée de process	Parte Nesima della data di processo	过程数据项的第N部分
0027		GetExRight	Exklusive Schreibrechte anfordern	Request exclusive write access	Solicitar permisos de escritura exclusivos	Demander droits exclusifs en écriture	Richiedi diritti di scrittura esclusivi	请求排他写入权限
0028		ChangePDSet	Prozessdatenzuordnungen einstellen	Set process data assignment	Configurar asignaciones de datos de proceso	Régler affectations de données de process	Imposta assegnazioni dei dati di processo	设置过程数据分配
0029		ParamSetWriteControl	Steuerung der Blockparametrierung	Block parameterization control	Sistema de control de parametrización de bloque	Commande du paramétrage par blocs	Controllo della parametrizzazione di blocco	块参数设置控制
002A		ConflictDictionary	Konfliktverzeichnis	Conflict dictionary	Registro de conflictos	Répertoire de conflits	Cartella conflitti	冲突对象
	.01	ConfGrNo	Konfliktgruppennummer	Number of conflict group	Número de grupo de conflictos	Numéro de groupe de conflits	Numero di gruppo conflitti	冲突组数目
	.02	Subslot_1	Subslot des ersten abhängigen Parameters	Subslot of the first dependent parameter	Subslot del primer parámetro dependiente	Subslot du premier paramètre dépendant	Slot subordinato del primo parametro dipendente	第一个相关参数的子槽位
	.03	Index_1	Index des ersten abhängigen Parameters	Index of the first dependent parameter	Índice del primer parámetro dependiente	Index du premier paramètre dépendant	Indice del primo parametro dipendente	第一个相关参数的索引
	.04	Subindex_1	Subindex des ersten abhängigen Parameters	Subindex of the first dependent parameter	Subíndice del primer parámetro dependiente	Sous-index du premier paramètre dépendant	Sottoindice del primo parametro dipendente	第一个相关参数的子索引
	.05	Element_1	Element im Record des ersten abhängigen Parameters	Element in the record of the first dependent parameter	Elemento en registro del primer parámetro dependiente	Élément de l'enregistrement du premier paramètre dépendant	Elemento nel record del primo parametro dipendente	第一个相关参数记录中的元素
	.06	Add.Code_1	Zusatzinformation zum ersten abhängigen Parameters	Additional information about the first dependent parameter	Información adicional sobre el primer parámetro dependiente	Informations complémentaires relatives au premier paramètre dépendant	Informazione supplementare sul primo parametro dipendente	有关第一个相关参数的附加信息
	...							
002B		ParamSet	Parametersatzkennung	Parameter record identification	Identificador de juego de parámetros	Identifiant du jeu de paramètres	Codice record parametri	参数记录标识
002C		ParameterMoment	Zeitpunkt der letzten Änderung der Parametrierung	Time of last parameterization modification	Momento de la última modificación de la parametrización	Moment de la dernière modification du paramétrage	Ora dell'ultima modifica della parametrizzazione	上次参数化修改的时间

Index	Subindex	Objektname (Basisprofil)	DE	EN-US	ES	FR	IT	ZH-CN
	.01	Date	Datum	Date	Fecha	Date	Data	日期
	.02	Time	Uhrzeit	Time	Hora	Heure	Ora	时间
002D		ResetParam	Parametrierung zurücksetzen	Reset parameterization	Restablecer la parametrización	Réinitialiser le paramétrage	Resetta parametrizzazione	复位参数设置
002E		ParamHash	Hash-Wert	Hash value	Valor hash	Valeur Hash	Valore Hash	哈希值
002F		PDOOUT_Subst	Ersatzwert Ausgangsprozessdaten	Substitute value for OUT process data	Valor sustitutivo datos de proceso de salida	Valeur de substitution données de sortie de process	Valore sostitutivo dei dati di processo in uscita	OUT过程数据的替代值
0030		PF_Code	Ersatzwertverhalten bei Peripheriefehler (PDIN)	Substitute value behavior during peripheral fault (PDIN)	Comportamiento de valor sustitutivo en caso de error en periferia (PDIN)	Comportement de valeur de substitution en cas d'erreur périphérique (PDIN)	Andamento sostitutivo in caso di errori periferici (PDIN)	外围设备故障期间的替换值特性 (PDIN)
0031		PDIN_Subst	Ersatzwert Eingangsprozessdaten	Substitute value for IN process data	Valor sustitutivo datos de proceso de entrada	Valeur de substitution données d'entrée de process	Valore sostitutivo dati di processo in ingresso	IN过程数据的替代值
0032		FieldBus_ID	Feldbusidentifikation / reserviert	Fieldbus identification/reserved	Identificación de bus de campo / reservada	Identification bus de terrain / réservé	Identificazione bus di campo / riservato	现场总线标识 / 保留
	.01	ID-Code	ID-Code	ID code	Código de ID	Code ID	Codice ID	识别码
	.02	PDLength	Prozessdatenlänge	Process data length	Longitud de datos de proceso	Longueur données de process	Lunghezza dei dati di processo	过程数据长度
0035		SubBusInfo	Sub-Bus-Information	Sub-bus information	Información de subbus	Informations sous-bus	Informazione bus subordinato	子总线信息
	.01	SubBusType	Bezeichnung des Sub-Busses	Sub-bus designation	Denominación del subbus	Désignation du sous-bus	Denominazione del bus subordinato	子总线名称
	.02	MTStructFormat	Feldlängen	Field lengths	Longitudes de campo	Longueurs de champs	Lunghezze di campo	字段长度
	.03	ActNoOfModules	Anzahl angeschlossener Sub-Bus-Module	Number of connected sub-bus modules	Número de módulos subbus conectados	Nombre de modules sous-bus raccordés	Numero di moduli bus subordinato collegati	连接的子总线模块的数量
	.04	LastMappedModule	Letztes direkt abgebildetes Sub-Bus-Modul	Last sub-bus module directly mapped	Último módulo subbus representado directamente	Dernier module sous-bus directement représenté	Ultimo modulo bus subordinato raffigurato direttamente	最后一个子总线模块已直接映射
	.05	RemainingSystem	Restsystem	Residual system	Sistema residual	Système résiduel	Sistema residuo	剩余系统
0036		ActSubBusStructure	Sub-Bus-Istkonfiguration	Actual sub-bus configuration	Configuración actual del subbus	Configuration réelle sous-bus	Configurazione effettiva bus subordinato	实际的子总线组态
0037		DeviceType	Gerätetyp	Device type	Tipo de dispositivo	Type d'appareil	Tipo di dispositivo	设备类型
0038		ObjDescrReq	Anfrage Objektbeschreibung	Request object description	Solicitud de descripción de	Demande description d'objet	Richiesta descrizione oggetto	请求对象描述

Index	Subindex	Objektname (Basisprofil)	DE	EN-US	ES	FR	IT	ZH-CN
					objeto			
	.01	Index	Index	Index	Índice	Index	Indice	索引
	.02	Subindex	Subindex	Subindex	Subíndice	Sous-index	Indice subordinato	子索引
0039		ObjDescr	Objektbeschreibung	Object description	Descripción de objeto	Description d'objet	Descrizione oggetto	对象描述
	.01	Index	Index	Index	Índice	Index	Indice	索引
	.02	Subindex	Subindex	Subindex	Subíndice	Sous-index	Indice subordinato	子索引
	.03	ObjectCode	Objektcode	Object code	Código de objeto	Code d'objet	Codice oggetto	对象代码
	.04	IndexOfType	Datentypindex	Data type index	Índice de tipo de datos	Index type de données	Indice tipo dati	数据类型索引
	.05	Length	Länge des (Sub-)Objekts	Length of (sub) object	Longitud del (sub)objeto	Longueur du (sous) objet	Lunghezza del (sotto)oggetto	(子) 对象的长度
	.06	UnitText	Einheit des Werts	Unit of value	Unidad del valor	Unité de la valeur	Unità del valore	值的单位
	.07	UnitCode	Einheiten-Größencode	Unit size code	Código de tamaño de unidades	Unités code de taille	Codice di grandezza unità	单位尺寸代码
	.08	UnitCodeExp	Einheitenindex	Unit index	Índice de unidades	Index d'unités	Indice unità	单位索引
	.09	Offset	Offset	Offset	Offset	Décalage	Offset	偏移量
	.0A	RDR	Dimensionsbereich der Auflösung	Dimension range of the resolution	Gama de dimensiones de la resolución	Plage dimensionnelle de la résolution	Campo dimensionale della risoluzione	分辨率的尺寸范围
	.0B	RNR	Zahlenbereich der Auflösung	Number range of the resolution	Rango numérico de la resolución	Plage numérique de la résolution	Campo numerico della risoluzione	分辨率的数字范围
	.0C	AccessRights	Zugriffsrechte	Access rights	Derechos de acceso	Droits d'accès	Diritti di accesso	访问权限
	.0D	DisplayFormat	Anzeigeformat	Display format	Formato de visualización	Format d'affichage	Formato di visualizzazione	显示格式
	.0E	Min	Untere Grenze	Lower limit	Límite inferior	Limite inférieure	Limite inferiore	下限
	.0F	Max	Obere Grenze	Upper limit	Límite superior	Limite supérieure	Limite superiore	上限
	.10	Symbol	Name/Bezeichnung des Objekts	Name/designation of the object	Nombre / denominación del objeto	Nom/désignation de l'objet	Nome/Denominazione dell'oggetto	对象的名称/称号
003A		VersionCount	Versionszähler	Version counter	Contador de versiones	Compteur de versions	Contatore versione	版本计数器
	.01	ProfileVersion	Profilversion	Profile version	Versión de perfil	Version profil	Versione profilo	配置文件版本
	.02	PChVersion	Parameterkanalversion	Parameter channel version	Versión de canal de parámetros	Version du canal de paramètres	Versione canale parametro	参数通道版本
	.03	HardwareVersion	Hardware-Version	Hardware version	Versión de hardware	Version matériel	Versione hardware	硬件版本
	.04	FirmwareVersion	Firmware-Version	Firmware version	Versión de firmware	Version firmware	Versione firmware	固件版本
003B		PDIN_Descr	Beschreibung der Eingangsprozessdat	IN process data description	Descripción de datos de proceso de	Description des données d'entrée de	Descrizione dei dati di processo in	IN过程数据描述

Index	Subindex	Objektname (Basisprofil)	DE	EN-US	ES	FR	IT	ZH-CN
			en		entrada	process	ingresso	
	.01	Type	Typ des I/O-Datums	Type of I/O data item	Referencia de la fecha de E/S	Type de date E/S	Tipo di data I/O	I/O数据项的类型
	.02	ChNo	Anzahl der Kanäle	Number of channels	Número de canales	Nombre de canaux	Numero di canali	通道数
	.03	ChLength	Länge eines Kanals	Length of a channel	Longitud de un canal	Longueur d'un canal	Lunghezza di un canale	通道长度
003C		PDOOUT_Descr	Beschreibung der Ausgangsprozessdaten	OUT process data description	Descripción de datos de proceso de salida	Description des données de sortie de process	Descrizione dei dati di processo in uscita	OUT过程数据描述
	.01	Type	Typ des I/O-Datums	Type of I/O data item	Referencia de la fecha de E/S	Type de date E/S	Tipo di data I/O	I/O数据项的类型
	.02	ChNo	Anzahl der Kanäle	Number of channels	Número de canales	Nombre de canaux	Numero di canali	通道数
	.03	ChLength	Länge eines Kanals	Length of a channel	Longitud de un canal	Longueur d'un canal	Lunghezza di un canale	通道长度
003D		WakeUpTime	Hochlaufzeit	Startup time	Tiempo de arranque	Temps d'accélération	Tempo di avviamento	启动时间
003E		EnergyMgmt	Energiemanagement	Energy management	Gestión de energía	Gestion de l'énergie	Gestione dell'energia	能源管理
	.01	ActualMode	Aktueller Energiesparmodus	Current energy-saving mode	Modo de ahorro de energía actual	Mode actuel d'économie d'énergie	Modalità di risparmio energetico attuale	当前的节能模式
	.02	Pause	Pausenzeit	Pause time	Tiempo de pausa	Durée de pause	Tempo di pausa	暂停时间
0040		ListOfObjToRestore	Liste der wiederherzustellenden Objekte	List of objects to be restored	Lista de objetos a restaurar	Liste des objets à rétablir	Elenco degli oggetti da ripristinare	待恢复的对象列表
	.01N	SubModNoN IndexN SubindexN	Sub-Modul-Nummer, Index und Subindex des N-ten Objekts	Submodule number, index and subindex of the Nth object	Número de submódulo, índice y subíndice del objeto nº n	Numéro du sous-module, index et sous-index du Nième objet	Numero modulo subordinato, indice e indice subordinato dell'oggetto Nesimo	子模块编号, 第N个对象的索引和子索引
0041		RefSubBusStructure	Sub-Bus-Sollkonfiguration	Desired sub-bus configuration	Configuración nominal del subbus	Configuration consigne de sous-bus	Configurazione nominale del bus subordinato	所需的子总线组态
0042		ModuleStatus	Modulstatus im Sub-Bus	Module status on the sub-bus	Estado del módulo en el subbus	Statut de module dans le sous-bus	Stato modulo nel bus subordinato	子总线上的模块状态
	.01N	ModuleN	Status des N-ten Moduls	Status of the Nth module	Estado del módulo nº n	Statut du Nième module	Stato del Nesimo modulo	第N个模块的状态
0043		SubBusBehaviour	Sub-Bus-Verhalten	Sub-bus behavior	Comportamiento del subbus	Comportement sous-bus	Comportamento del bus subordinato	子总线特性
	.01	AutoSetup	Automatischer Anlauf	Automatic startup	Secuencia de activación automática	Démarrage automatique	Avvio automatico	自动重启
	.02	AutoRestart	Automatischer Wiederanlauf	Automatic restart	Reinicio automático	Redémarrage automatique	Riavvio automatico	自动重启

Index	Subindex	Objektname (Basisprofil)	DE	EN-US	ES	FR	IT	ZH-CN
	.03	Remaining	Restsystem	Residual system	Sistema residual	Système résiduel	Sistema residuo	剩余系统
0044		SubBusControl	Steuerung des Sub-Busses	Sub-bus control	Sistema de control del subbus	Commande du sous-bus	Controllo del bus subordinato	子总线控制
	.01	SubBusControlCMD	Sub-Bus Kontrolle	Sub-bus control	Control del subbus	Contrôle du sous-bus	Controllo del bus subordinato	子总线控制
0045		InitFWDownload	Firmware-Download-Initialisierung	Initializing firmware download	Inicialización de descarga de firmware	Initialisation du téléchargement du firmware	Inizializzazione download firmware	初始化固件下载
	.01	Status	Status	Status	Estado	Etat	Stato	状态
	.02	Control	Steuerung	Controller	Sistema de control	Automate	Comando	控制器
	.03	HeaderVersion	Header-Information	Header information	Información de cabecera	Information en-tête	Informazione header	标题版本
	.04	UpdateVersion	Update-Version	Update version	Versión de actualización	Version d'actualisation	Versione update	更新版本
	.05	BaseData	Basisdaten	Basic data	Datos básicos	Données de base	Dati essenziali	基本数据
0047		AddInfo	Zusätzliche Informationen	Additional information	Informaciones adicionales	Informations complémentaires	Informazioni aggiuntive	更多信息
	.01	AddInfo.SafetyProtType	Safety-Protokolltyp	Safety protocol type	Tipo de protocolo de seguridad	Type de protocole Safety	Tipo di protocollo safety	安全协议类型
	.02	AddInfo.SafetyProtVers	Safety-Protokollversion	Safety protocol version	Versión de protocolo de seguridad	Version de protocole Safety	Versione di protocollo safety	安全协议版本
	.03	AddInfo.LegacyInfo	Information zu Vorgängerversionen	Information about previous version	Información sobre versiones anteriores	Informations relatives aux versions précédentes	Informazione sulle versioni precedenti	有关之前版本的信息
C000 ... C07F		ProjBasProf	Projektion des Basisprofils für Sub-Busse	Projection of basic profile for sub-buses	Proyección del perfil básico para subbus	Projection du profil de base pour sous-bus	Proiezione del profilo di base per bus subordinati	子总线基本配置文件的映射
E800		DiagStateLong	Diagnosezustand (Langform)	Diagnostic state (long form)	Estado de diagnóstico (formato largo)	Etat de diagnostic (forme longue)	Stato di diagnosi (forma estesa)	诊断状态（长格式）
E801		DiagHistory	Diagnosearchiv	Diagnostics archive	Archivo de diagnóstico	Archive de diagnostic	Archivio diagnosi	诊断归档
	.01	DiagState1	Ältester Diagnosezustand in der Kurzform	Oldest diagnostic state in short form	Estado de diagnóstico más antiguo en el formato corto	Etat de diagnostic le plus ancien, forme courte	Stato di diagnosi più vecchio nella forma breve	短格式中最旧的诊断状态
	...							
	N	DiagStateN	Neuester Diagnosezustand in der Kurzform	Newest diagnostic state in short form	Estado de diagnóstico más nuevo en el formato corto	Etat de diagnostic le plus récent, forme courte	Stato di diagnosi più recente nella forma breve	短格式中最新的诊断状态

Index	Subindex	Objektname (Basisprofil)	DE	EN-US	ES	FR	IT	ZH-CN
E802		DiagHistoryLong	Diagnosearchiv (Langform)	Diagnostics archive (long form)	Archivo de diagnóstico (formato largo)	Archive de diagnostic (forme longue)	Archivio diagnosi (forma estesa)	诊断归档 (长格式)
	.01	DiagState1	Ältester Diagnosezustand in der Langform	Oldest diagnostic state in long form	Estado de diagnóstico más antiguo en el formato largo	Etat de diagnostic le plus ancien, forme longue	Stato di diagnosi più vecchio nella forma estesa	长格式中最旧的诊断状态
	...							
	N	DiagStateN	Neuester Diagnosezustand in der Langform	Newest diagnostic state in long form	Estado de diagnóstico más nuevo en el formato largo	Etat de diagnostic le plus récent, forme longue	Stato di diagnosi più recente nella forma estesa	长格式中最新的诊断状态
E803		TraceBuffer	Puffer für Protokollierung	Logging buffer	Búfer para registro	Tampon d'établissement de protocole	Buffer per registrazione	记录缓冲区
E804		LanguageAvailable	Verfügbare Sprachen	Available languages	Idiomas disponibles	Langues disponibles	Lingue disponibili	可用的语言
E805		ObjDescrLong	Objektbeschreibung (Langform)	Object description (long form)	Descripción de objeto (formato largo)	Description d'objet (forme longue)	Descrizione oggetto (forma estesa)	对象描述 (长格式)
E806		ComplDiagState	Vollständiger aktueller Diagnosezustand	Complete current diagnostic state	Estado de diagnóstico actual completo	Etat de diagnostic actuel, complet	Stato di diagnosi attuale completo	完整的当前诊断状态
	.01	DiagState1	Ältester Diagnosezustand in der Kurzform	Oldest diagnostic state in short form	Estado de diagnóstico más antiguo en el formato corto	Etat de diagnostic le plus ancien, forme courte	Stato di diagnosi più vecchio nella forma breve	短格式中最旧的诊断状态
	...							
	N	DiagStateN	Neuester Diagnosezustand in der Kurzform	Newest diagnostic state in short form	Estado de diagnóstico más nuevo en el formato corto	Etat de diagnostic le plus récent, forme courte	Stato di diagnosi più recente nella forma breve	短格式中最新的诊断状态
E807		Device FW	Firmware-File	Firmware file	Archivo de firmware	Fichier firmware	File firmware	固件文件
E808		OnBoardDeviceDescFile	OnBoard Gerätebeschreibung sdatei (komprimiert)	Onboard device description file (compressed)	Archivo de descripción de dispositivos Onboard (comprimido)	Fichier de description de l'appareil embarqué (comprimé)	Nome del file di descrizione del dispositivo OnBoard (compresso)	机载设备描述文件名称 (已压缩)
E809		BackUpDataCompr	Komprimierte Daten für Geräte-Backup	Compressed data for device backup	Datos comprimidos para respaldo de dispositivos	Données comprimées pour sauvegarde d'appareils	Dati compressi per backup dispositivo	设备备份的压缩数据

▪
▪
▪
▪
- Ende -