

DIN EN 61158-5-12

ICS 35.100.70; 35.160

Ersatz für
DIN EN 61158-5-12:2008-09
Siehe Anwendungsbeginn

**Industrielle Kommunikationsnetze –
Feldbusse –
Teil 5-12: Dienstfestlegungen des Application Layer
(Anwendungsschicht) –
Typ 12-Elemente
(IEC 61158-5-12:2010);
Englische Fassung EN 61158-5-12:2012, nur auf CD-ROM**

Industrial communication networks –
Fieldbus specifications –
Part 5-12: Application layer service definition –
Type 12 elements
(IEC 61158-5-12:2010);
English version EN 61158-5-12:2012, only on CD-ROM

Réseaux de communication industriels –
Spécifications des bus de terrain –
Partie 5-12: Définition des services des couches d'application –
Éléments de type 12
(CEI 61158-5-12:2010);
Version anglaise EN 61158-5-12:2012, seulement en CD-ROM

Gesamtumfang 122 Seiten

DKE Deutsche Kommission Elektrotechnik Elektronik Informationstechnik im DIN und VDE

Anwendungsbeginn

Anwendungsbeginn für die von CENELEC am 2012-03-28 angenommene Europäische Norm als DIN-Norm ist 2013-01-01.

Für DIN EN 61158-5-12:2008-09 gilt eine Übergangsfrist bis zum 2015-03-28.

Nationales Vorwort

Vorausgegangener Norm-Entwurf: E DIN IEC 61158-500:2009-06.

Für dieses Dokument ist das nationale Arbeitsgremium K 956 „Feldbus“ der DKE Deutsche Kommission Elektrotechnik Elektronik Informationstechnik im DIN und VDE (www.dke.de) zuständig.

Die enthaltene IEC-Publikation wurde vom SC 65C „Industrial networks“ erarbeitet.

Das IEC-Komitee hat entschieden, dass der Inhalt dieser Publikation bis zu dem Datum (stability date) unverändert bleiben soll, das auf der IEC-Website unter „<http://webstore.iec.ch>“ zu dieser Publikation angegeben ist. Zu diesem Zeitpunkt wird entsprechend der Entscheidung des Komitees die Publikation

- bestätigt,
- zurückgezogen,
- durch eine Folgeausgabe ersetzt oder
- geändert.

Für den Fall einer undatierten Verweisung im normativen Text (Verweisung auf eine Norm ohne Angabe des Ausgabedatums und ohne Hinweis auf eine Abschnittsnummer, eine Tabelle, ein Bild usw.) bezieht sich die Verweisung auf die jeweils neueste gültige Ausgabe der in Bezug genommenen Norm.

Für den Fall einer datierten Verweisung im normativen Text bezieht sich die Verweisung immer auf die in Bezug genommene Ausgabe der Norm.

Der Zusammenhang der zitierten Normen mit den entsprechenden Deutschen Normen ergibt sich, soweit ein Zusammenhang besteht, grundsätzlich über die Nummer der entsprechenden IEC-Publikation. Beispiel: IEC 60068 ist als EN 60068 als Europäische Norm durch CENELEC übernommen und als DIN EN 60068 ins Deutsche Normenwerk aufgenommen.

Das Präsidium des DIN hat mit Präsidialbeschluss 1/2004 festgelegt, dass DIN-Normen, deren Inhalt sich auf internationale Arbeitsergebnisse der Informationsverarbeitung gründet, unter bestimmten Bedingungen allein in englischer Sprache veröffentlicht werden dürfen. Diese Bedingungen sind für die vorliegende Norm erfüllt.

Da sich die Benutzer der vorliegenden Norm der englischen Sprache als Fachsprache bedienen, wird die Englische Fassung der EN 61158-5-12 veröffentlicht. Zu deren Abschnitt 3, der die Begriffe festlegt, wurde eine Übersetzung angefertigt und als informativer Nationaler Anhang NA der vorliegenden Norm hinzugefügt. Für viele der verwendeten Begriffe existieren keine gebräuchlichen deutschen Benennungen, da sich die deutschen Anwender in der Regel ebenfalls der englischen Benennungen bedienen. Diese Norm steht nicht in unmittelbarem Zusammenhang mit Rechtsvorschriften und ist nicht als Sicherheitsnorm anzusehen.

Einführung

Ein Feldbus ist ein digitaler serieller Datenbus für die Kommunikation zwischen Geräten der industriellen Automatisierungstechnik wie beispielsweise Messeinrichtungen, Stellgliedern, Reglern und speicherprogrammierbaren Steuerungen. Die Normen der Reihe DIN EN 61158 legen eine Anzahl von Feldbus-Protokolltypen fest. Die vorliegende Norm legt die Dienste der Anwendungsschicht für Typ 12 fest.

Änderungen

Gegenüber DIN EN 61158-5-12:2008-09 wurden folgende Änderungen vorgenommen:

- a) Fehlerberichtigungen;
- b) redaktionelle Verbesserungen.

Frühere Ausgaben

DIN V 19245-1: 1988-01
 DIN 19245-1: 1991-04
 DIN 19245-2: 1991-04
 DIN EN 50170/1: 1997-07
 DIN EN 50170/2: 1997-07
 DIN EN 50170/3: 1997-07
 DIN EN 50254: 1999-07
 DIN EN 50170/A2: 2000-08
 DIN EN 50170/A1: 2002-11
 DIN EN 50170/A3: 2002-11
 DIN EN 61158-5: 2005-05
 DIN EN 61158-5-12: 2008-09

Nationaler Anhang NA (informativ)

3.3 Begriffe zu den Diensten der Anwendungsschicht und der Sicherungsschicht

3.3.1

Anwendung

(en: application)

Funktion oder Datenstruktur, für die Daten verarbeitet oder erzeugt werden

3.3.2

Anwendungsobjekte

(en: application objects)

mehrere Objektklassen, die einen Nachrichtenaustausch über das Netzwerk und innerhalb des Netzwerkgerätes zur Laufzeit verwalten und bereitstellen

3.3.3

Basis-Slave

(en: basic slave)

Slave, der nur die physikalische Adressierung von Daten unterstützt

3.3.4

Bit

(en: bit)

Informationseinheit, die aus einer 0 oder 1 besteht. Ein Bit ist die kleinste Dateneinheit, die übertragen werden kann

3.3.5

Client

(en: client)

- 1) Objekt, das die Dienste eines anderen Objekts (Servers) nutzt, um eine Aufgabe auszuführen
- 2) Initiator einer Nachricht, auf die ein Server reagiert

3.3.6

Kommunikationsobjekt

(en: **communication object**)

Komponente, die einen Austausch von Nachrichten über das Netzwerk zur Laufzeit bereitstellt und verwaltet

3.3.7

Verbindung

(en: **connection**)

logische Verbindung zwischen zwei Anwendungsobjekten innerhalb eines Gerätes oder in unterschiedlichen Geräten

3.3.8

zyklisch

(en: **cyclic**)

Ereignisse, die sich regelmäßig wiederholen

3.3.9

Daten

(en: **data**)

allgemeine Bezeichnung für jegliche über einen Feldbus übertragene Informationen

3.3.10

Datenkonsistenz

(en: **data consistency**)

Mittel für die kohärente Übertragung des Eingabe- oder Ausgabedatenobjekts zwischen Client und Server und für den kohärenten Zugriff auf das Eingabe- oder Ausgabedatenobjekt innerhalb von Client und Server

3.3.11

Datentyp

(en: **data type**)

Beziehung zwischen den Werten und der Codierung für Daten dieses Typs. Für die vorliegende Spezifikation gelten die Definitionen für Datentypen aus IEC 61131-3

3.3.12

Datentypobjekt

(en: **data type object**)

Eintrag im Objektverzeichnis, der einen Datentyp angibt

3.3.13

Standard-Gateway

(en: **default gateway**)

Gerät mit mindestens zwei Schnittstellen in zwei verschiedenen IP-Subnetzen, das als Router für ein Subnetz agiert

3.3.14

Gerät

(en: **device**)

mit dem Feldbus verbundene physikalische Einheit, die aus mindestens einem Kommunikationselement (dem Netzwerkelement) besteht und über ein Bedienelement und/oder ein Endelement (Wandler, Aktor usw.) verfügen kann

3.3.15

Geräteprofil

(en: **device profile**)

Zusammenstellung geräteabhängiger Informationen und Funktionalitäten, um Konsistenz zwischen ähnlichen Geräten desselben Gerätetyps herzustellen

3.3.16**Diagnose-Informationen****(en: diagnosis information)**

alle Daten, die serverseitig für Wartungszwecke verfügbar sind

3.3.17**verteilte Uhren****(en: distributed clocks)**

Methode zur Synchronisierung von Slaves und zur Verwaltung einer globalen Zeitbasis

3.3.18**Fehler****(en: error)**

Nichtübereinstimmung zwischen einem berechneten, beobachteten oder gemessenen Wert oder Zustand und dem festgelegten oder theoretisch richtigen Wert oder Zustand

3.3.19**Fehlerklasse****(en: error class)**

allgemeine Gruppierung für zugehörige Fehlerdefinitionen und entsprechende Fehlercodes

3.3.20**Fehlercode****(en: error code)**

Kennzeichnung eines bestimmten Fehlertyps innerhalb einer Fehlerklasse

3.3.21**Ereignis****(en: event)**

Instanz einer Änderung von Zuständen

3.3.22**Speicherverwaltungseinheit des Feldbusses****(en: fieldbus memory management unit)**

Funktion, die eine oder mehrere Korrespondenzen zwischen logischen Adressen und dem physikalischen Speicher aufbaut

3.3.23**Element der Speicherverwaltungseinheit des Feldbusses****(en: fieldbus memory management unit entity)**

einzelnes Element der Speicherverwaltungseinheit des Feldbusses: eine Korrespondenz zwischen einem kohärenten logischen Adressraum und einem kohärenten physikalischen Speicherort

3.3.24**Rahmen****(en: frame)**

umgangssprachliches Synonym für DLPDU

3.3.25**Full-Slave****(en: full slave)**

Slave-Gerät, das sowohl die physikalische als auch die logische Adressierung von Daten unterstützt

3.3.26**Index****(en: index)**

Adresse eines Objektes innerhalb eines Anwendungsprozesses

3.3.27

Schnittstelle

(en: interface)

gemeinsam genutzte Grenze zwischen zwei funktionalen Einheiten, die durch funktionale Merkmale, Signalmerkmale oder entsprechend andere Merkmale festgelegt ist

3.3.28

Little-Endian

(en: little endian)

Methode für die Datendarstellung von Zahlen größer als 8 Bit, wobei das am wenigsten signifikante Oktett zuerst übertragen wird

3.3.29

Master

(en: master)

Gerät, das den Datentransfer im Netzwerk steuert und den Medienzugriff der Slaves durch Senden von Nachrichten initiiert und das die Schnittstelle zum Steuerungssystem bildet

3.3.30

Zuordnung

(en: mapping)

Verbindung zwischen zwei Objekten in einer Weise, dass ein Objekt ein Teil des anderen Objekts ist

3.3.31

Zuordnungsparameter

(en: mapping parameters)

Satz von Werten, der die Verbindung zwischen Anwendungsobjekten und Prozessdatenobjekten festlegt

3.3.32

Medium

(en: medium)

Kabel, Lichtwellenleiter oder anderes Mittel, über das Kommunikationssignale zwischen zwei oder mehreren Punkten übertragen werden

ANMERKUNG „Medien“ wird als Plural von Medium verwendet.

3.3.33

Nachricht

(en: message)

geordnete Folge von Oktetten für die Übermittlung von Informationen

ANMERKUNG Nachrichten werden normalerweise für die Übermittlung von Informationen zwischen Peers auf der Anwendungsschicht verwendet.

3.3.34

Netzwerk

(en: network)

Menge von Knoten, die über ein bestimmtes Kommunikationsmedium miteinander verbunden sind, einschließlich zwischengeschalteter Repeater, Bridges, Router und Gateways unterer Schichten

3.3.35

Knoten

(en: node)

- a) einzelne DL-Instanz, wie sie auf einem lokalen Verbindungsweig existiert
- b) Endpunkt eines Verbindungsweiges in einem Netzwerk oder ein Punkt, an dem zwei oder mehrere Verbindungsweige aufeinandertreffen [abgeleitet von IEC 61158-2]

3.3.36**Objekt****(en: object)**

abstrakte Darstellung einer bestimmten Komponente innerhalb eines Gerätes

ANMERKUNG Ein Objekt kann sein:

- a) eine abstrakte Darstellung der Fähigkeiten eines Gerätes. Objekte können aus einigen der folgenden Komponenten oder aus allen folgenden Komponenten bestehen:
 - 1) Daten (Informationen, die sich im Laufe der Zeit ändern);
 - 2) Konfiguration (Parameter für das Verhalten);
 - 3) Methoden (Dinge, die mithilfe der Daten und der Konfiguration durchgeführt werden können);
- b) eine Zusammenstellung zugehöriger Daten (in Form von Variablen) und Methoden (Prozeduren) für Operationen in Bezug auf diese Daten mit klar definierter Schnittstelle und klar definiertem Verhalten.

3.3.37**Objektverzeichnis****(en: object dictionary)**

Datenstruktur, die von Index und Subindex adressiert wird und Beschreibungen von Datentypobjekten, Kommunikationsobjekten und Anwendungsobjekten enthält

3.3.38**Prozessdaten****(en: process data)**

Zusammenstellung von Anwendungsobjekten, die zum Zweck der Messung und Steuerung zyklisch oder azyklisch übertragen werden

3.3.39**Prozessdatenobjekt****(en: process data object)**

Struktur, die durch Zuordnungsparameter beschrieben wird und eine oder mehrere Prozessdateneinheiten enthält

3.3.40**Segment****(en: segment)**

Zusammenstellung von einem realen Master und einem oder mehreren Slaves

3.3.41**Server****(en: server)**

Objekt, das einem anderen Objekt (Client) Dienste bereitstellt

3.3.42**Dienst****(en: service)**

Operation oder Funktion, die ein Objekt und/oder eine Objektklasse auf Aufforderung eines anderen Objekts und/oder einer anderen Objektklasse ausführt

3.3.43**Slave****(en: slave)**

DL-Instanz, die nur nach Initiierung durch den vorausgehenden Slave oder den Master Zugang zum Medium hat

3.3.44**Subindex****(en: sub-index)**

untergeordnete Adresse eines Objektes innerhalb des Objektverzeichnisses

3.3.45

Sync-Manager

(en: **Sync Manager**)

Zusammenstellung von Steuerelementen zur Koordinierung des Zugriffs auf gleichzeitig verwendete Objekte

3.3.46

Sync-Manager-Kanal

(en: **Sync Manager channel**)

einzelne Steuerelemente zur Koordinierung des Zugriffs auf gleichzeitig verwendete Objekte

3.3.47

Switch

(en: **switch**)

MAC-Bridge, wie in IEEE 802.1D definiert

English version

**Industrial communication networks -
Fieldbus specifications -
Part 5-12: Application layer service definition -
Type 12 elements
(IEC 61158-5-12:2010)**

Réseaux de communication industriels -
Spécifications des bus de terrain -
Partie 5-12: Définition des services des
couches d'application -
Eléments de type 12
(CEI 61158-5-12:2010)

Industrielle Kommunikationsnetze -
Feldbusse -
Teil 5-12: Dienstfestlegungen des
Application Layer (Anwendungsschicht) -
Typ 12-Elemente
(IEC 61158-5-12:2010)

This European Standard was approved by CENELEC on 2012-03-28. CENELEC members are bound to comply with the CEN/CENELEC Internal Regulations which stipulate the conditions for giving this European Standard the status of a national standard without any alteration.

Up-to-date lists and bibliographical references concerning such national standards may be obtained on application to the CEN-CENELEC Management Centre or to any CENELEC member.

This European Standard exists in three official versions (English, French, German). A version in any other language made by translation under the responsibility of a CENELEC member into its own language and notified to the CEN-CENELEC Management Centre has the same status as the official versions.

CENELEC members are the national electrotechnical committees of Austria, Belgium, Bulgaria, Croatia, Cyprus, the Czech Republic, Denmark, Estonia, Finland, France, Germany, Greece, Hungary, Iceland, Ireland, Italy, Latvia, Lithuania, Luxembourg, Malta, the Netherlands, Norway, Poland, Portugal, Romania, Slovakia, Slovenia, Spain, Sweden, Switzerland, Turkey and the United Kingdom.

CENELEC

European Committee for Electrotechnical Standardization
Comité Européen de Normalisation Electrotechnique
Europäisches Komitee für Elektrotechnische Normung

Management Centre: Avenue Marnix 17, B - 1000 Brussels

Foreword

The text of document 65C/606/FDIS, future edition 2 of IEC 61158-5-12, prepared by SC 65C, "Industrial networks", of IEC/TC 65, "Industrial-process measurement, control and automation" was submitted to the IEC-CENELEC parallel vote and approved by CENELEC as EN 61158-5-12:2012.

The following dates are fixed:

- latest date by which the document has to be implemented at national level by publication of an identical national standard or by endorsement (dop) 2012-12-28
- latest date by which the national standards conflicting with the document have to be withdrawn (dow) 2015-03-28

This document supersedes EN 61158-5-12:2008.

EN 61158-5-12:2012 includes the following significant technical change with respect to EN 61158-5-12:2008:

- bug fixes and;
- editorial improvements.

Attention is drawn to the possibility that some of the elements of this document may be the subject of patent rights. CENELEC [and/or CEN] shall not be held responsible for identifying any or all such patent rights.

Endorsement notice

The text of the International Standard IEC 61158-5-12:2010 was approved by CENELEC as a European Standard without any modification.

In the official version, for Bibliography, the following notes have to be added for the standards indicated:

IEC/TR 61158-1:2010	NOTE Harmonized as CLC/TR 61158-1:2010 (not modified).
IEC 61158-2	NOTE Harmonized as EN 61158-2.
IEC 61158-4-12	NOTE Harmonized as EN 61158-4-12.
IEC 61158-6-12	NOTE Harmonized as EN 61158-6-12.
IEC 61784-2	NOTE Harmonized as EN 61784-2.

CONTENTS

INTRODUCTION	6
1 Scope	7
1.1 Overview	7
1.2 Specifications	8
1.3 Conformance	8
2 Normative references	8
3 Terms, definitions, symbols, abbreviations and conventions	9
3.1 Reference model terms and definitions	9
3.2 Service convention terms and definitions	9
3.3 Application layer and data-link service terms and definitions	10
3.4 Common symbols and abbreviations	14
3.5 Conventions	15
4 Concepts	16
4.1 Common concepts	16
4.2 Type specific concepts	16
5 Data type ASE	23
5.1 General	23
5.2 Formal definition of data type objects	23
5.3 FAL defined data types	23
5.4 Data type ASE service specification	31
6 Communication model specification	31
6.1 ASEs	31
6.2 AR	102
Bibliography	113
Annex ZA (normative) Normative references to international publications with their corresponding European publications	114
Figure 1 – Producer consumer model	17
Figure 2 – Client server model	18
Figure 3 – Server triggered invocation	18
Figure 4 – Slave reference model	19
Figure 5 – Simple slave device	20
Figure 6 – Complex slave device	21
Figure 7 – Master functional overview	22
Figure 8 – Process output data sequence	32
Figure 9 – Process input data sequence	33
Figure 10 – CoE server model	49
Figure 11 – Successful single SDO-Download sequence	54
Figure 12 – Unsuccessful single SDO-Download sequence	54
Figure 13 – Successful segmented SDO-Download sequence	55
Figure 14 – Successful single SDO-Upload sequence	56
Figure 15 – Unsuccessful single SDO-Upload sequence	56
Figure 16 – Successful segmented SDO-Upload sequence	57

Figure 17 – SDO information sequence	58
Figure 18 – Emergency service	59
Figure 19 – Command sequence	60
Figure 20 – PDO mapping	61
Figure 21 – Sync manager PDO assignment	62
Figure 22 – RxPDO service	63
Figure 23 – TxPDO service	64
Figure 24 – RxPDO remote transmission sequence	65
Figure 25 – TxPDO remote transmission sequence	65
Figure 26 – EoE sequence	84
Figure 27 – FoE read sequence with success	91
Figure 28 – FoE read sequence with error	92
Figure 29 – FoE write sequence with success	92
Figure 30 – FoE write sequence with error	93
Figure 31 – FoE write sequence with busy	93
Figure 32 – Successful AL control sequence	103
Figure 33 – Unsuccessful AL control sequence	104
Figure 34 – AL state changed sequence	105
Table 1 – Process output data	35
Table 2 – Process input data	36
Table 3 – Update process input data	37
Table 4 – SII read	44
Table 5 – SII write	45
Table 6 – SII reload	46
Table 7 – Allocation of SDO areas	50
Table 8 – SDO download expedited	69
Table 9 – SDO download normal	70
Table 10 – Download SDO segment	71
Table 11 – SDO upload expedited	72
Table 12 – SDO upload normal	73
Table 13 – Upload SDO segment	74
Table 14 – Abort SDO transfer	74
Table 15 – Get OD list	75
Table 16 – OD list segment	76
Table 17 – Get object description	77
Table 18 – Get entry description	78
Table 19 – Object entry segment	80
Table 20 – Emergency	81
Table 21 – RxPDO	81
Table 22 – TxPDO	82
Table 23 – RxPDO remote transmission	82
Table 24 – TxPDO remote transmission	83

Table 25 – Initiate EoE	87
Table 26 – EoE fragment	88
Table 27 – Set IP parameter	89
Table 28 – Set address filter	90
Table 29 – FoE read	95
Table 30 – FoE write	95
Table 31 – FoE data	96
Table 32 – FoE ack	96
Table 33 – FoE busy	97
Table 34 – FoE error	97
Table 35 – MBX read.....	99
Table 36 – MBX write	100
Table 37 – MBX read upd	101
Table 38 – AL management and ESM service primitives	102
Table 39 – AL control	111
Table 40 – AL state change	111

INTRODUCTION

This part of IEC 61158 is one of a series produced to facilitate the interconnection of automation system components. It is related to other standards in the set as defined by the “three-layer” fieldbus reference model described in IEC/TR 61158-1.

The application service is provided by the application protocol making use of the services available from the data-link or other immediately lower layer. This standard defines the application service characteristics that fieldbus applications and/or system management may exploit.

Throughout the set of fieldbus standards, the term “service” refers to the abstract capability provided by one layer of the OSI Basic Reference Model to the layer immediately above. Thus, the application layer service defined in this standard is a conceptual architectural service, independent of administrative and implementation divisions.

INDUSTRIAL COMMUNICATION NETWORKS – FIELDBUS SPECIFICATIONS –

Part 5-12: Application layer service definition – Type 12 elements

1 Scope

1.1 Overview

The fieldbus Application Layer (FAL) provides user programs with a means to access the fieldbus communication environment. In this respect, the FAL can be viewed as a “window between corresponding application programs.”

This standard provides common elements for basic time-critical and non-time-critical messaging communications between application programs in an automation environment and material specific to Type 12 fieldbus. The term “time-critical” is used to represent the presence of a time-window, within which one or more specified actions are required to be completed with some defined level of certainty. Failure to complete specified actions within the time window risks failure of the applications requesting the actions, with attendant risk to equipment, plant and possibly human life.

This standard defines in an abstract way the externally visible service provided by the different Types of the fieldbus Application Layer in terms of

- a) an abstract model for defining application resources (objects) capable of being manipulated by users via the use of the FAL service,
- b) the primitive actions and events of the service;
- c) the parameters associated with each primitive action and event, and the form which they take; and
- d) the interrelationship between these actions and events, and their valid sequences.

The purpose of this standard is to define the services provided to

- a) the FAL user at the boundary between the user and the Application Layer of the Fieldbus Reference Model, and
- b) Systems Management at the boundary between the Application Layer and Systems Management of the Fieldbus Reference Model.

This standard specifies the structure and services of the IEC fieldbus Application Layer, in conformance with the OSI Basic Reference Model (ISO/IEC 7498) and the OSI Application Layer Structure (ISO/IEC 9545).

FAL services and protocols are provided by FAL application-entities (AE) contained within the application processes. The FAL AE is composed of a set of object-oriented Application Service Elements (ASEs) and a Layer Management Entity (LME) that manages the AE. The ASEs provide communication services that operate on a set of related application process object (APO) classes. One of the FAL ASEs is a management ASE that provides a common set of services for the management of the instances of FAL classes.

Although these services specify, from the perspective of applications, how request and responses are issued and delivered, they do not include a specification of what the requesting and responding applications are to do with them. That is, the behavioral aspects of the applications are not specified; only a definition of what requests and responses they can send/receive is specified. This permits greater flexibility to the FAL users in standardizing such object behavior. In addition to these services, some supporting services are also defined in this standard to provide access to the FAL to control certain aspects of its operation.

1.2 Specifications

The principal objective of this standard is to specify the characteristics of conceptual application layer services suitable for time-critical communications, and thus supplement the OSI Basic Reference Model in guiding the development of application layer protocols for time-critical communications.

A secondary objective is to provide migration paths from previously-existing industrial communications protocols. It is this latter objective which gives rise to the diversity of services standardized as the various Types of IEC 61158, and the corresponding protocols standardized in subparts of IEC 61158-6.

This specification may be used as the basis for formal Application Programming-Interfaces. Nevertheless, it is not a formal programming interface, and any such interface will need to address implementation issues not covered by this specification, including

- a) the sizes and octet ordering of various multi-octet service parameters, and
- b) the correlation of paired request and confirm, or indication and response, primitives.

1.3 Conformance

This standard does not specify individual implementations or products, nor does it constrain the implementations of application layer entities within industrial automation systems.

There is no conformance of equipment to this application layer service definition standard. Instead, conformance is achieved through implementation of conforming application layer protocols that fulfill any given Type of application layer services as defined in this standard.

2 Normative references

The following referenced documents are indispensable for the application of this document. For dated references, only the edition cited applies. For undated references, the latest edition of the referenced document (including any amendments) applies.

IEC 61131-3, *Programmable controllers – Part 3: Programming languages*

IEC 61158-3-12, *Industrial communication networks – Fieldbus specifications – Part 3-12: Data-link layer service definition – Type 12 elements*

ISO/IEC 646:1991, *Information technology – ISO 7-bit coded character set for information interchange*

ISO/IEC 7498-1, *Information technology – Open Systems Interconnection – Basic Reference Model: The Basic Model*

ISO/IEC 7498-3, *Information technology – Open Systems Interconnection – Basic Reference Model: Naming and addressing*

ISO/IEC 8802-3, *Information technology – Telecommunications and information exchange between systems – Local and metropolitan area networks – Specific requirements – Part 3: Carrier sense multiple access with collision detection (CSMA/CD) access method and physical layer specifications*

ISO/IEC 10646, *Information technology – Universal Multiple-Octet Coded Character Set (UCS)*

ISO/IEC 10731, *Information technology – Open Systems Interconnection – Basic Reference Model – Conventions for the definition of OSI services*

IEEE 802.1D, *IEEE standard for local and metropolitan area networks – Media access control (MAC) Bridges*; available at <<http://www.ieee.org>>

IETF RFC 791, *Internet Protocol darpa internet program protocol specification*; available at <http://www.ietf.org>

3 Terms, definitions, symbols, abbreviations and conventions

For the purposes of this document, the following terms, definitions, symbols, abbreviations and conventions apply.

3.1 Reference model terms and definitions

This standard is based in part on the concepts developed in ISO/IEC 7498-1 and ISO/IEC 7498-3, and makes use of the following terms defined therein:

3.1.1 correspondent (N)-entities	[ISO/IEC 7498-1]
correspondent AL-entities (N=7)	
3.1.2 (N)-entity	[ISO/IEC 7498-1]
AL-entity (N=7)	
3.1.3 (N)-layer	[ISO/IEC 7498-1]
AL-layer (N=7)	
3.1.4 layer-management	[ISO/IEC 7498-1]
3.1.5 peer-entities	[ISO/IEC 7498-1]
3.1.6 primitive name	[ISO/IEC 7498-3]
3.1.7 AL-protocol	[ISO/IEC 7498-1]
3.1.8 AL-protocol-data-unit	[ISO/IEC 7498-1]
3.1.9 reset	[ISO/IEC 7498-1]
3.1.10 routing	[ISO/IEC 7498-1]
3.1.11 segmenting	[ISO/IEC 7498-1]
3.1.12 (N)-service	[ISO/IEC 7498-1]
AL-service (N=7)	
3.1.13 AL-service-data-unit	[ISO/IEC 7498-1]
3.1.14 AL-simplex-transmission	[ISO/IEC 7498-1]
3.1.15 AL-subsystem	[ISO/IEC 7498-1]
3.1.16 systems-management	[ISO/IEC 7498-1]
3.1.17 AL-user-data	[ISO/IEC 7498-1]

3.2 Service convention terms and definitions

This standard also makes use of the following terms defined in ISO/IEC 10731 as they apply to the data-link layer:

3.2.1 acceptor

3.2.2 asymmetrical service

**3.2.3 confirm (primitive);
requestor.deliver (primitive)**

3.2.4 deliver (primitive)

**3.2.5 AL-service-primitive;
primitive**

3.2.6 AL-service-provider

3.2.7 AL-service-user

**3.2.8 indication (primitive);
acceptor.deliver (primitive)**

**3.2.9 request (primitive);
requestor.submit (primitive)**

3.2.10 requestor

**3.2.11 response (primitive);
acceptor.submit (primitive)**

3.2.12 submit (primitive)

3.2.13 symmetrical service

3.3 Application layer and data-link service terms and definitions

3.3.1

application

function or data structure for which data is consumed or produced

3.3.2

application objects

multiple object classes that manage and provide a run time exchange of messages across the network and within the network device]

3.3.3

basic slave

slave device that supports only physical addressing of data

3.3.4

bit

unit of information consisting of a 1 or a 0. This is the smallest data unit that can be transmitted

3.3.5

client

1) object which uses the services of another (server) object to perform a task

2) initiator of a message to which a server reacts

3.3.6

communication object

component that manage and provide a run time exchange of messages across the network

3.3.7

connection

logical binding between two application objects within the same or different devices

3.3.8

cyclic

events which repeat in a regular and repetitive manner

3.3.9

data

generic term used to refer to any information carried over a fieldbus

3.3.10

data consistency

means for coherent transmission and access of the input- or output-data object between and within client and server

3.3.11

data type

relation between values and encoding for data of that type. For this specification the data type definitions of IEC 61131-3 apply.

3.3.12

data type object

Entry in the object dictionary indicating a data type

3.3.13

default gateway

device with at least two interfaces in two different IP subnets acting as router for a subnet.

3.3.14

device

physical entity connected to the fieldbus composed of at least one communication element (the network element) and which may have a control element and/or a final element (transducer, actuator, etc.)

3.3.15

device profile

collection of device dependent information and functionality providing consistency between similar devices of the same device

3.3.16

diagnosis information

all data available at the server for maintenance purposes

3.3.17

distributed clocks

method to synchronize slaves and maintain a global time base

3.3.18

error

discrepancy between a computed, observed or measured value or condition and the specified or theoretically correct value or condition

3.3.19

error class

general grouping for related error definitions and corresponding error codes

3.3.20

error code

identification of a specific type of error within an error class

3.3.21

event

instance of a change of conditions

3.3.22

fieldbus memory management unit

function that establishes one or several correspondences between logical addresses and physical memory

3.3.23

fieldbus memory management unit entity

single element of the fieldbus memory management unit: one correspondence between a coherent logical address space and a coherent physical memory location

3.3.24

frame

denigrated synonym for DLPDU

3.3.25

full slave

slave device that supports both physical and logical addressing of data

3.3.26

index

address of an object within an application process

3.3.27

interface

shared boundary between two functional units, defined by functional characteristics, signal characteristics, or other characteristics as appropriate

3.3.28

little endian

Method for data representation of numbers greater 8 bit where the least significant octet is transmitted first.

3.3.29

master

device that controls the data transfer on the network and initiates the media access of the slaves by sending messages and that constitutes the interface to the control system

3.3.30

mapping

correspondence between two objects in that way that one object is part of the other object

3.3.31

mapping parameters

set of values defining the correspondence between application objects and process data objects

3.3.32

medium

cable, optical fibre, or other means by which communication signals are transmitted between two or more points

NOTE "media" is used as the plural of medium.

3.3.33

message

ordered series of octets intended to convey information

NOTE Normally used to convey information between peers at the application layer.

3.3.34

network

set of nodes connected by some type of communication medium, including any intervening repeaters, bridges, routers and lower-layer gateways

3.3.35

node

a) single DL-entity as it appears on one local link

b) end-point of a link in a network or a point at which two or more links meet [derived from IEC 61158-2]

3.3.36

object

abstract representation of a particular component within a device

NOTE An object can be

a) an abstract representation of the capabilities of a device. Objects can be composed of any or all of the following components:

- 1) data (information which changes with time);
- 2) configuration (parameters for behavior);
- 3) methods (things that can be done using data and configuration).

b) a collection of related data (in the form of variables) and methods (procedures) for operating on that data that have clearly defined interface and behavior.

3.3.37

object dictionary

data structure addressed by Index and Sub-index that contains descriptions of data type objects, communication objects and application objects

3.3.38

process data

collection of application objects designated to be transferred cyclically or acyclically for the purpose of measurement and control

3.3.39

process data object

structure described by mapping parameters containing one or several process data entities

3.3.40

segment

collection of one real master with one or more slaves

3.3.41

server

object which provides services to another (client) object

3.3.42

service

operation or function that an object and/or object class performs upon request from another object and/or object class

3.3.43

slave

DL-entity accessing the medium only after being initiated by the preceding slave or

3.3.44

sub-index

subaddress of an object within the object dictionary

3.3.45

Sync Manager

collection of control elements to coordinate access to concurrently used objects.

3.3.46

Sync Manager channel

single control elements to coordinate access to concurrently used objects.

3.3.47

switch

MAC bridge as defined in IEEE 802.1D

3.4 Common symbols and abbreviations

AL-	Application layer (as a prefix)
ALE	AL-entity (the local active instance of the application layer)
AL	AL-layer
APDU	AL-protocol-data-unit
ALM	AL-management
ALME	AL-management Entity (the local active instance of AL-management)
ALMS	AL-management service
ALS	AL-service
AR	Application relationship
ASE	Application service element
CAN	Controller Area Network
CiA	CAN in Automation
CoE	CAN application protocol over Type 12 services
CSMA/CD	Carrier sense multiple access with collision detection
DC	Distributed clocks
DL	Data-link-layer
DNS	Domain name system (server for name resolution in IP networks)
E²PROM	Electrically erasable programmable read only memory
EoE	Ethernet tunneled over Type 12 services
ESC	Type 12 slave controller
FCS	Frame check sequence
FIFO	First-in first-out (queuing method)
FMU	Fieldbus memory management unit
FoE	File access with Type 12 services
HDR	Header
ID	Identifier
IETF	Internet engineering task force
IP	Internet protocol
LAN	Local area network
MAC	Medium access control
OD	Object dictionary
OSI	Open systems interconnection
PDI	Physical device internal interface (a set of elements that allows access to DL-services from the AL)
PDO	Process data object
PhL	Ph-layer

QoS	Quality of service
RAM	Random access memory
Rx	Receive
SDO	Service data object
SII	slave information interface
SM	Synchronization manager
SyncM	Synchronization manager
TCP	Transmission control protocol
Tx	Transmit
UDP	User datagram protocol
WKC	Working counter

3.5 Conventions

This standard uses the descriptive conventions given in ISO/IEC 10731.

The service model, service primitives, and time-sequence diagrams used are entirely abstract descriptions; they do not represent a specification for implementation.

Service primitives, used to represent service user/service provider interactions (see ISO/IEC 10731), convey parameters that indicate information available in the user/provider interaction.

This standard uses a tabular format to describe the component parameters of the service primitives. The parameters that apply to each group of service primitives are set out in tables throughout the remainder of this standard. Each table consists of up to five columns, containing the name of the service parameter, and a column each for those primitives and parameter-transfer directions used by the service:

- the request primitive's input parameters;
- the indication primitive's output parameters;
- the response primitive's input parameters; and
- the confirm primitive's output parameters.

NOTE The request, indication, response and confirm primitives are also known as requestor.submit, acceptor.deliver, acceptor.submit, and requestor.deliver primitives, respectively (see ISO/IEC 10731).

One parameter (or part of it) is listed in each row of each table. Under the appropriate service primitive columns, a code is used to specify the type of usage of the parameter on the primitive and parameter direction specified in the column:

M	parameter is mandatory for the primitive.
U	parameter is a User option, and may or may not be provided depending on the dynamic usage of the service-user. When not provided, a default value for the parameter is assumed.
C	parameter is conditional upon other parameters or upon the environment of the service-user.
(blank)	parameter is never present.

Some entries are further qualified by items in brackets. These may be a parameter-specific constraint:

- (=) indicates that the parameter is semantically equivalent to the parameter in the service primitive to its immediate left in the table.

In any particular interface, not all parameters need be explicitly stated. Some may be implicitly associated with the primitive.

In the diagrams which illustrate these interfaces, dashed lines indicate cause-and-effect or time-sequence relationships, and wavy lines indicate that events are roughly contemporaneous.

4 Concepts

4.1 Common concepts

All of IEC/TR 61158-1, Clause 9 is incorporated by reference, except as specifically overridden in 4.2.

4.2 Type specific concepts

4.2.1 Operating principle

This standard and its companion Type 12 standards describe a real-time Ethernet technology that aims to maximize the utilization of the full duplex Ethernet bandwidth. Medium access control employs the master/slave principle, where the master node (typically the control system) sends the Ethernet frames to the slave nodes, which extract data from and insert data into these frames.

From an Ethernet point of view, a Type 12 segment is a single Ethernet device, which receives and sends standard ISO/IEC 8802-3 Ethernet frames. However, this Ethernet device is not limited to a single Ethernet controller with downstream microprocessor, but may consist of a large number of Type 12 slave devices. These process the incoming frames directly and extract the relevant user data, or insert data and transfer the frame to the next slave device. The last slave device within the segment sends the fully processed frame back, so that it is returned by the first slave device to the master as response frame.

This procedure utilizes the full duplex capability of Ethernet: both communication directions are operated independently. Communication without switch between a master device and a Type 12 segment consisting of one or several slave devices may be established.

Industrial communication systems have to meet different requirements in terms of the data transmission characteristics. Parameter data is transferred acyclically and in large quantities, whereby the timing requirements are relatively non-critical, and the transmission is usually triggered by the control system. Diagnostic data is also transferred acyclically and event-driven, but the timing requirements are more demanding, and the transmission is usually triggered by a peripheral device.

Process data, on the other hand, is typically transferred cyclically with different cycle times. The timing requirements are most stringent for process data communication. Type 12 AL supports a variety of services and protocols to meet these differing requirements.

4.2.2 Communication model overview

The Type 12 application layer distinguishes between master and slave. The communication relationship is always initiated by the master.

A Type 12 segment consists of at least one master device and one or many slave devices. All slave devices support the Type 12 State Machine (ESM) and support the transmission of Type 12 process data.

The application relationship can be modeled independent of communication relationship. The master-slave relationship is the standard application relationship.

4.2.3 Application layer element description

4.2.3.1 Management

The mandatory management consists of a set of object to control the state of a slave. An interface to DL provides read access to all DL registers.

4.2.3.2 Information interface

The mandatory slave information interface (SII) consists of all objects that can be stored persistently.

4.2.3.3 Synchronization support

The optional support of isochronous operation consists of several attributes for synchronization and timestamping of binary signals.

4.2.3.4 Access to slave

The real time entity consists of an interface for network triggered exchange of data and an interface for user triggered access to slave objects. Objects mainly used for network triggered access are called PDO. SDO are the objects that are used for user triggered access.

The access methods for PDO are read and write to a buffer. The communication structure is a producer-consumer relationship as shown in Figure 1 with no direct acknowledgement of the delivery of data. The delivery of data will be monitored by additional elements in the slave and in the master (i.e. watchdog and working counter). The producer can be a master as well as a slave.

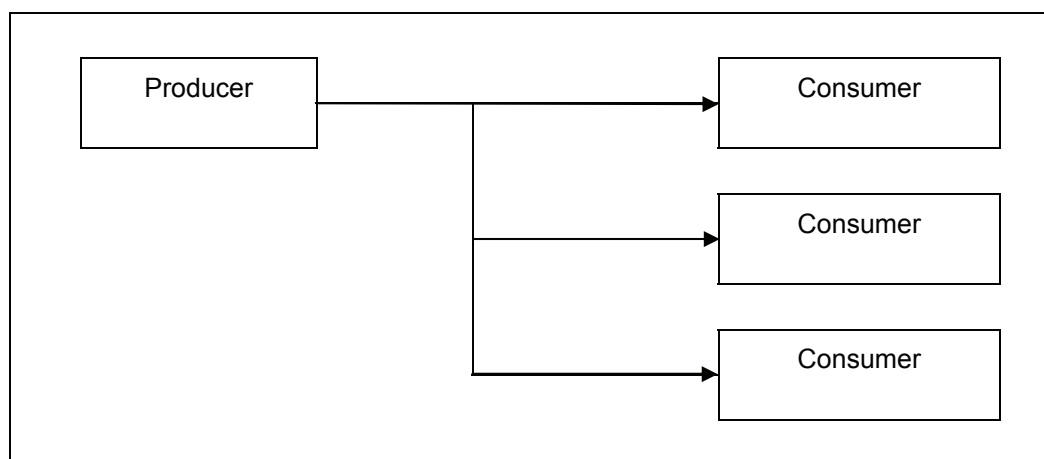


Figure 1 – Producer consumer model

The access to SDO follows the client server principle. The client issues a service invocation to the server. The slave starts the service execution and replies the result afterwards. There is always a response needed to conclude this type of service.

Figure 2 shows the workflow of this communication interaction.

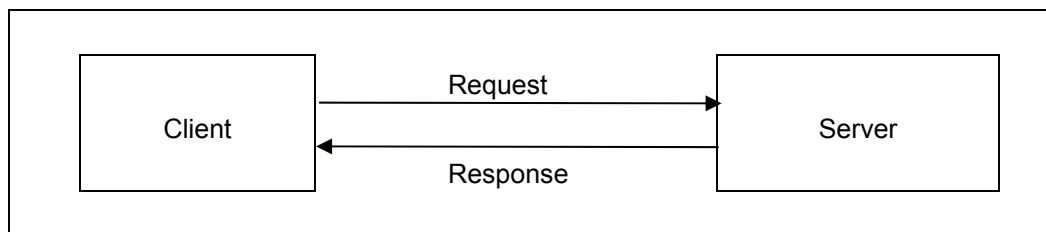


Figure 2 – Client server model

There may be an unsolicited type of server to client interaction as well. This model is used to convey data server triggered. This type of service called notification is shown in Figure 3.

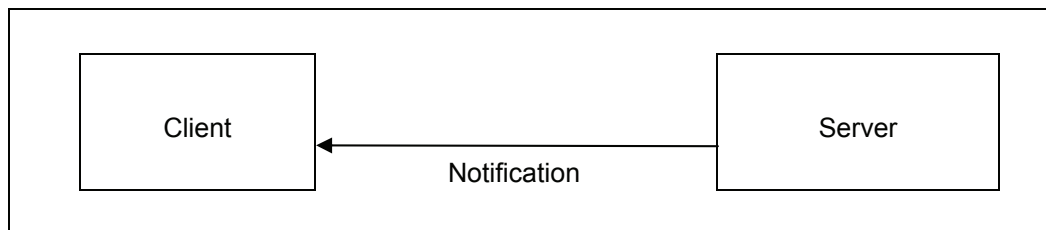


Figure 3 – Server triggered invocation

4.2.3.5 TCP/UDP/IP suite

The optional protocol is dedicated for slaves who will use the standard internet protocol suite. The protocols itself are defined in IETF. EoE describes the mapping of the IP-Protocol (and similar kind of communication) to Type 12 data-link layer. IP is a connectionless communication type with bidirectional data flow.

4.2.3.6 File access

The primary use of the file transfer is the download and upload of program files and configuration data. The file access is done with client server protocol architecture.

4.2.4 Slave reference model

4.2.4.1 Mapping onto OSI basic reference model

Type 12 is described using the principles, methodology and model of ISO/IEC 7498 Information processing systems — Open Systems Interconnection — Basic Reference Model (OSI). The OSI model provides a layered approach to communications standards, whereby the layers can be developed and modified independently. The Type 12 specification defines functionality from top to bottom of a full OSI stack, and some functions for the users of the stack. Functions of the intermediate OSI layers, layers 3 – 6, are consolidated into either the Type 12 data-link layer or the Type 12 application layer. Likewise, features common to users of the Fieldbus application layer may be provided by the Type 12 application layer to simplify user operation, as noted in Figure 4.

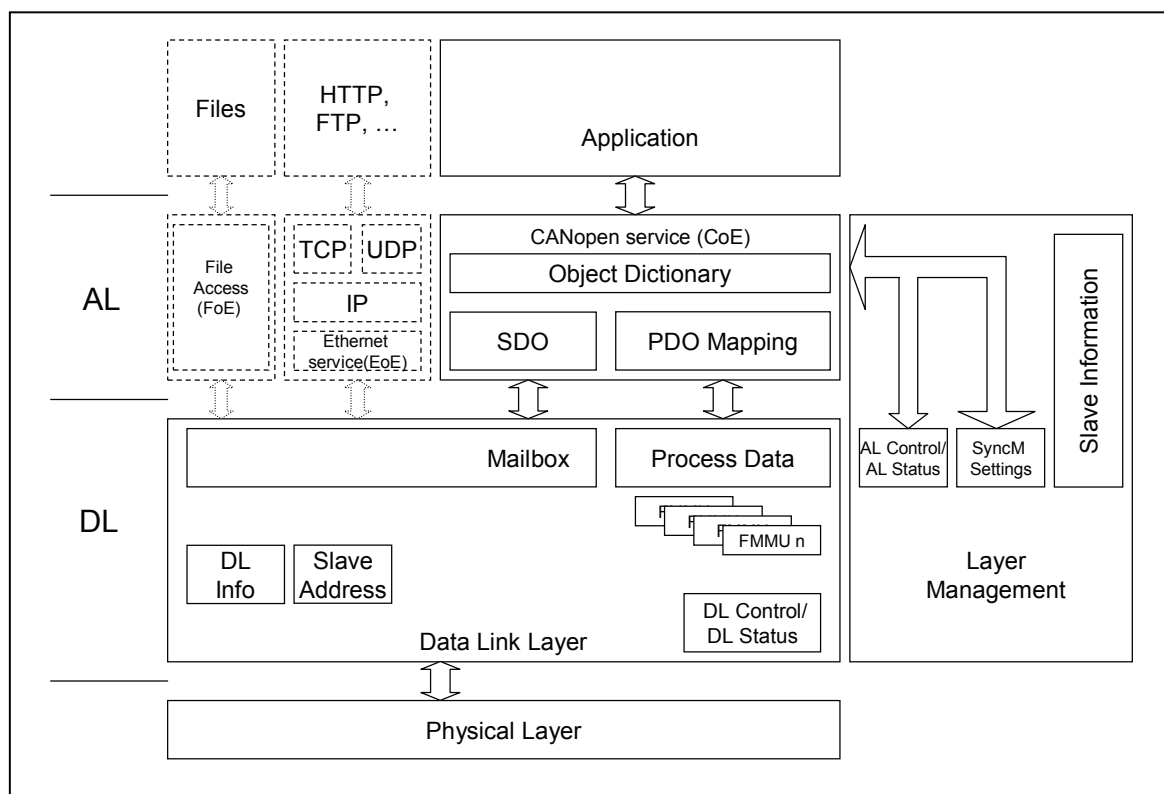


Figure 4 – Slave reference model

4.2.4.2 Data-link layer features

The data-link layer provides basic time critical support for data communications among devices connected via a Type 12 segment. The term “time-critical” is used to describe applications having a time-window, within which one or more specified actions are required to be completed with some defined level of certainty. Failure to complete specified actions within the time window risks failure of the applications requesting the actions, with attendant risk to equipment, plant and possibly human life.

The data-link layer has the task to compute, compare and generate the frame check sequence and provide communications by extracting data from and/or including data into the Ethernet frame. This is done depending on the data-link layer parameters which are stored at pre-defined memory locations. The application data is made available to the application layer in physical memory, either in a mailbox configuration or within the process data section.

Additionally some data structures in the data-link layer will be used to allow a coordination of the interaction between Master and slave such as AL Control, Status and Event and Sync Manager settings.

4.2.4.3 Slave AL classification

4.2.4.3.1 Simple slave device

From the application layer point of view, slave devices are classified in simple devices without an application controller and complex devices with an application controller.

NOTE The DL slave classification in basic slaves and full slaves is independent of the AL view, since DL addressing mechanisms are invisible at the AL interface.

Simple devices have a fixed process data layout, which is described in the device description file. Simple devices may confirm the AL Management services without a reaction within the local application, as noted in Figure 5. There is no special reaction needed for safe state operation (e.g. the value 0 will be processed in the same way as no valid value will be send).

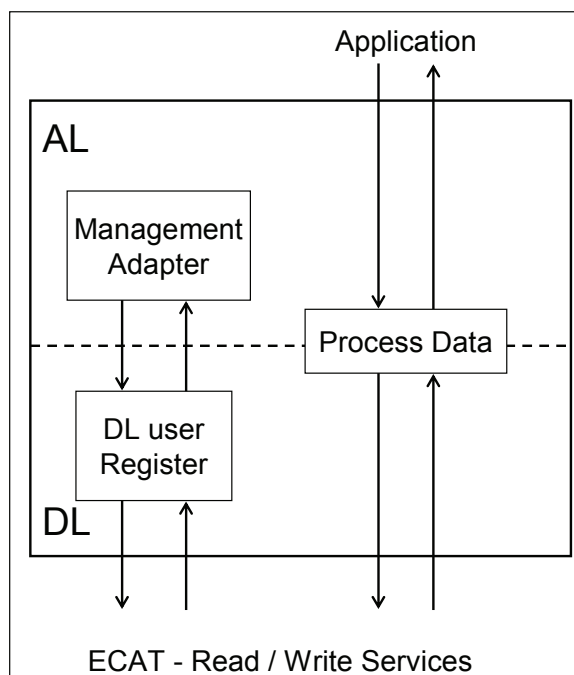


Figure 5 – Simple slave device

4.2.4.3.2 Complex slave device

As noted in Figure 6, complex devices support

- the ESM,
- the Mailbox (optional),
- the CoE object dictionary (recommended if mailbox is supported),
- the SDO services to read and/or write the object dictionary data entries (recommended if mailbox is supported), and
- the SDO information service to read the defined objects in the object dictionary and each entry description in compact format (recommended if mailbox is supported).

For the process data transmission the PDO mapping objects and the Sync Manager PDO assign objects, which describe the process data layout, shall be supported for reading. If an complex device supports configurable process data, the configuration is done by writing the PDO mapping and/or the Sync Manager PDO assign objects.

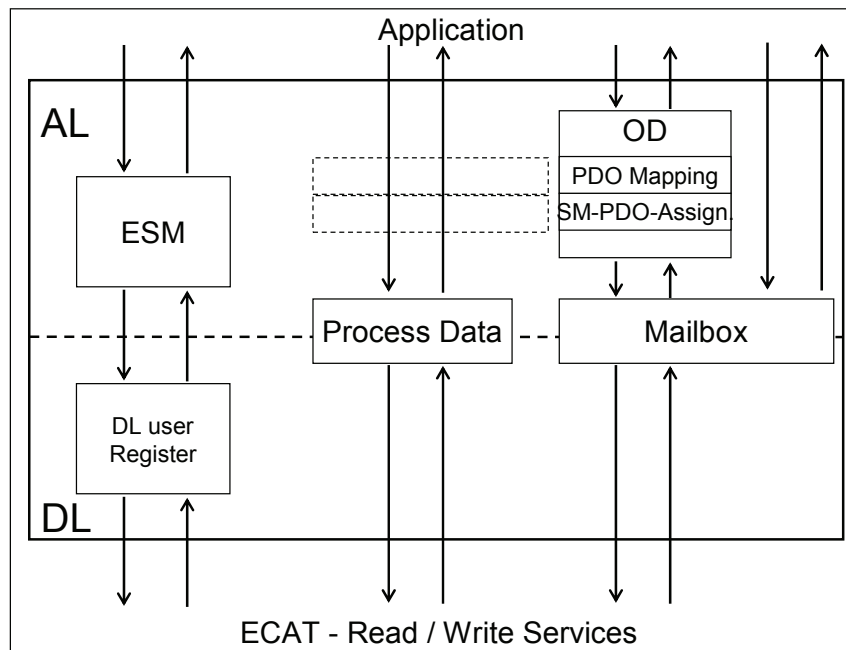


Figure 6 – Complex slave device

There are different interaction types defined in this standard:

- CAN application protocol over Type 12 services (CoE)
- Ethernet over Type 12 services (EoE)
- File Access over Type 12 services (FoE)

The different types are used to address different classes of objects. These types can be mixed at a single application relationship.

4.2.5 Master reference model

4.2.5.1 Overview

The master communicates with the slaves by using the services described in the slave chapter. Additionally there is a slave Handler for each slave defined in the master to control the ESM of the slave and a Router which enables slave-to-slave communication via the mailbox, as noted in Figure 7.

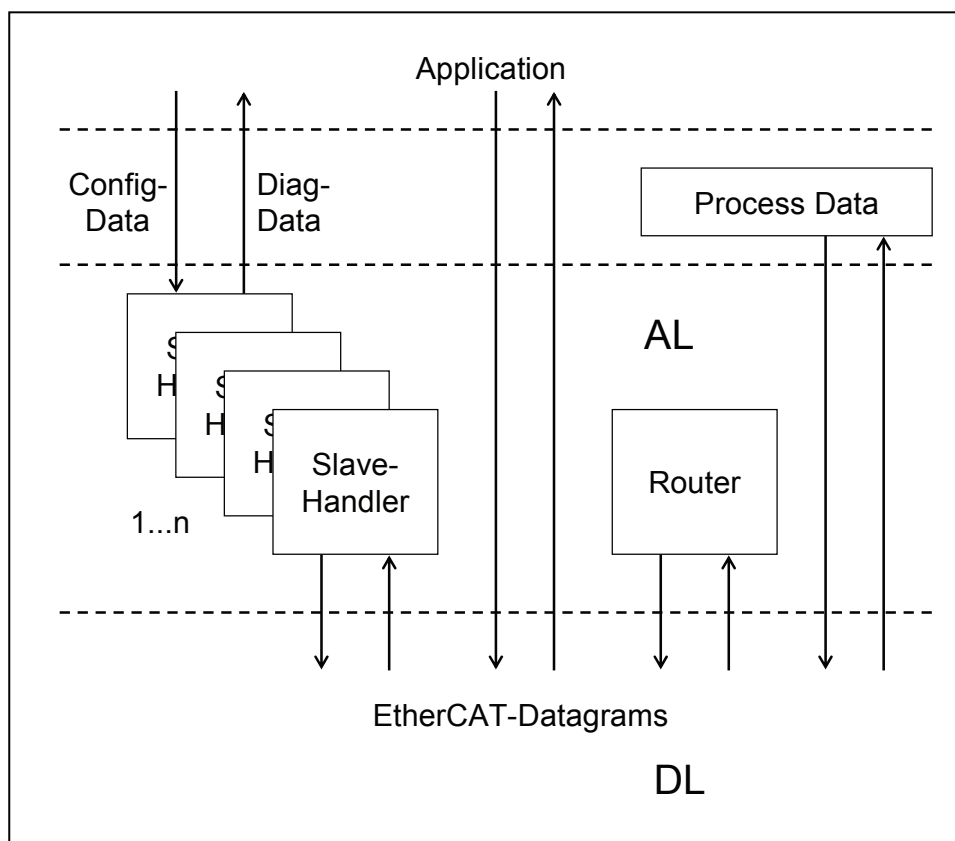


Figure 7 – Master functional overview

4.2.5.2 Slave handler

The master should support a slave handler for each slave using the state services to control the ESM of the slave. The slave Handler is the image of the slave's ESM in the master. Additionally the slave Handler may send SDO services before changing the state of the slave's ESM

Parameter

Position

This parameter specifies the position in the logical ring which is used to address the slave when reading the Identification and writing the Station Address. It is mandatory for all slaves.

Expected Identification

This parameter specifies the expected identification of the slave, which should be read and compared by the master in the Init state. This parameter is mandatory for all slaves.

Station Address

This parameter specifies the station address which is assigned in the Init state to the slave. All further services will use this station address to address the slave. This parameter is mandatory for all slaves.

Mailbox Configuration

This parameter specifies the configuration of the Sync Manager channels 0 and 1 for the mailbox which is written in the Init state to the slave. This parameter is mandatory for complex slaves.

FMMU Configuration

This parameter specifies the configuration of the FMMU channels which is written in the Pre-Operational state to the slave.

Process Data Configuration

This parameter specifies the configuration of the Sync Manager channels which is used for process data and is written in the Pre-Operational state to the slave.

PDO mapping

This parameter specifies the PDO mapping objects which may be written in the Pre-Operational state to the slave.

Sync Manager PDO assign

This parameter specifies the Sync Manager PDO assign objects which may be written in the Pre-Operational state to the slave.

Start Up Objects

This parameter specifies the objects from the object dictionary of the slave which may be written to the slave from the slave Handler during start up.

4.2.5.3 Router

The Router can be used for several applications:

- routing mailbox services from the client slave to the server slave
- routing mailbox service responses from the server slave to the client slave
- forwarding mailbox services from third party devices
- forwarding mailbox service responses to third party devices

The task of the router is to overwrite the address field of the mailbox service with the station address of the client or with a virtual address before routing the mailbox service to the server addressed by the original address field. The address field of the mailbox service response is overwritten by the router with the station address of the server before routing the mailbox service response to the client slave addressed by the original address field or to the corresponding IP Address/ MAC Address in case of a virtual address.

5 Data type ASE

5.1 General

All of IEC/TR 61158-1, 5.1 is incorporated by reference.

5.2 Formal definition of data type objects

All of IEC/TR 61158-1, 5.2 is incorporated by reference.

5.3 FAL defined data types

5.3.1 Fixed length types

5.3.1.1 Boolean types

CLASS:	Data type
ATTRIBUTES:	
1	Data type Numeric Identifier = 1
2	Data type Name = Boolean
3	Format = FIXED LENGTH
4.1	Octet Length = 1

This data type expresses a Boolean data type with the values TRUE and FALSE.

5.3.1.2 Bitstring types

5.3.1.2.1 BIT2

CLASS:	Data type
ATTRIBUTES:	
1	Data type Numeric Identifier = 31
2	Data type Name = BIT2

3	Format	=	FIXED LENGTH
4.1	Octet Length	=	1

A BIT2 is an ordered sequence of Boolean data types, numbered from 1 to 2.

5.3.1.2.2 BIT3

CLASS: Data type

ATTRIBUTES:

1	Data type Numeric Identifier	=	32
2	Data type Name	=	BIT3
3	Format	=	FIXED LENGTH
4.1	Octet Length	=	1

A BIT3 is an ordered sequence of Boolean data types, numbered from 1 to 3.

5.3.1.2.3 BIT4

CLASS: Data type

ATTRIBUTES:

1	Data type Numeric Identifier	=	33
2	Data type Name	=	BIT3
3	Format	=	FIXED LENGTH
4.1	Octet Length	=	1

A BIT4 is an ordered sequence of Boolean data types, numbered from 1 to 4.

5.3.1.2.4 BIT5

CLASS: Data type

ATTRIBUTES:

1	Data type Numeric Identifier	=	34
2	Data type Name	=	BIT5
3	Format	=	FIXED LENGTH
4.1	Octet Length	=	1

A BIT5 is an ordered sequence of Boolean data types, numbered from 1 to 5.

5.3.1.2.5 BIT6

CLASS: Data type

ATTRIBUTES:

1	Data type Numeric Identifier	=	35
2	Data type Name	=	BIT6
3	Format	=	FIXED LENGTH
4.1	Octet Length	=	1

A BIT6 is an ordered sequence of Boolean data types, numbered from 1 to 6.

5.3.1.2.6 BIT7

CLASS: Data type

ATTRIBUTES:

1	Data type Numeric Identifier	=	36
2	Data type Name	=	BIT7
3	Format	=	FIXED LENGTH
4.1	Octet Length	=	1

A BIT7 is an ordered sequence of Boolean data types, numbered from 1 to 7.

5.3.1.2.7 BIT8

CLASS: Data type

ATTRIBUTES:

1	Data type Numeric Identifier	=	37
2	Data type Name	=	BIT8
3	Format	=	FIXED LENGTH
4.1	Octet Length	=	1

A BIT8 is an ordered sequence of Boolean data types, numbered from 1 to 8.

5.3.1.3 Currency types

There are no Currency types defined for Type 12.

5.3.1.4 Date/Time types

5.3.1.4.1 TimeOfDay

CLASS: Data type

ATTRIBUTES:

1	Data type Numeric Identifier	=	12
2	Data type Name	=	TimeOfDay
3	Format	=	FIXED LENGTH
4.1	Octet Length	=	6

This data type is composed of two elements of unsigned values and expresses the time of day and the date. The first element is an Unsigned32 data type and gives the time after the midnight in milliseconds. The second element is an Unsigned16 data type and gives the date counting the days from January 1, 1984.

5.3.1.4.2 TimeDifference

CLASS: Data type

ATTRIBUTES:

1	Data type Numeric Identifier	=	13
2	Data type Name	=	TimeDifference
3	Format	=	FIXED LENGTH
4.1	Octet Length	=	4 or 6

This data type is composed of two elements of unsigned values that express the difference in time. The first element is an Unsigned32 data type that provides the fractional portion of one day in milliseconds. The optional second element is an Unsigned16 data type that provides the difference in days.

5.3.1.5 Enumerated types

There are no Enumerated types defined for Type 12.

5.3.1.6 Handle types

There are no Handle types defined for Type 12.

5.3.1.7 Numeric types

5.3.1.7.1.1 float

This data type is the same as Float32.

5.3.1.7.1.2 Float32

CLASS: Data type

ATTRIBUTES:

1	Data type Numeric Identifier	=	8
2	Data type Name	=	Float32
3	Format	=	FIXED LENGTH
4.1	Octet Length	=	4

This type has a length of four octets. The format for float32 is that defined by IEC 60559 as single precision.

5.3.1.7.1.3 double

This data type is the same as Float64.

5.3.1.7.1.4 Float64

CLASS: Data type

ATTRIBUTES:

1	Data type Numeric Identifier	=	17
2	Data type Name	=	Float64
3	Format	=	FIXED LENGTH
4.1	Octet Length	=	8

This type has a length of eight octets. The format for float64 is that defined by IEC 60559 as double precision.

5.3.1.7.2 Integer types

5.3.1.7.2.1 Integer8

CLASS: Data type

ATTRIBUTES:

1	Data type Numeric Identifier	=	2
2	Data type Name	=	Integer8
3	Format	=	FIXED LENGTH
4.1	Octet Length	=	1

This integer type is a two's complement binary number with a length of one octet.

5.3.1.7.2.2 SINT

This IEC 61131-3 type is the same as Integer8.

5.3.1.7.2.3 char

This data type is the same as Integer8.

5.3.1.7.2.4 Integer16

CLASS: Data type

ATTRIBUTES:

1	Data type Numeric Identifier	=	3
2	Data type Name	=	Integer16
3	Format	=	FIXED LENGTH
4.1	Octet Length	=	2

This integer type is a two's complement binary number with a length of two octets.

5.3.1.7.2.5 INT

This IEC 61131-3 type is the same as Integer16.

5.3.1.7.2.6 short

This data type is the same as Integer16.

5.3.1.7.2.7 Integer24

CLASS: Data type

ATTRIBUTES:

- | | | | |
|-----|------------------------------|---|--------------|
| 1 | Data type Numeric Identifier | = | 16 |
| 2 | Data type Name | = | Integer24 |
| 3 | Format | = | FIXED LENGTH |
| 4.1 | Octet Length | = | 3 |

This integer type is a two's complement binary number with a length of three octets.

5.3.1.7.2.8 Integer32

CLASS: Data type

ATTRIBUTES:

- | | | | |
|-----|------------------------------|---|--------------|
| 1 | Data type Numeric Identifier | = | 4 |
| 2 | Data type Name | = | Integer32 |
| 3 | Format | = | FIXED LENGTH |
| 4.1 | Octet Length | = | 4 |

This integer type is a two's complement binary number with a length of four octets.

5.3.1.7.2.9 DINT

This IEC 61131-3 type is the same as Integer32.

5.3.1.7.2.10 long

This data type is the same as Integer32.

5.3.1.7.2.11 Integer40

CLASS: Data type

ATTRIBUTES:

- | | | | |
|-----|------------------------------|---|--------------|
| 1 | Data type Numeric Identifier | = | 18 |
| 2 | Data type Name | = | Integer40 |
| 3 | Format | = | FIXED LENGTH |
| 4.1 | Octet Length | = | 5 |

This integer type is a two's complement binary number with a length of five octets.

5.3.1.7.2.12 Integer48

CLASS: Data type

ATTRIBUTES:

- | | | | |
|-----|------------------------------|---|--------------|
| 1 | Data type Numeric Identifier | = | 19 |
| 2 | Data type Name | = | Integer48 |
| 3 | Format | = | FIXED LENGTH |
| 4.1 | Octet Length | = | 6 |

This integer type is a two's complement binary number with a length of six octets.

5.3.1.7.2.13 Integer56

CLASS: Data type

ATTRIBUTES:

1	Data type Numeric Identifier	=	20
2	Data type Name	=	Integer56
3	Format	=	FIXED LENGTH
4.1	Octet Length	=	7

This integer type is a two's complement binary number with a length of seven octets.

5.3.1.7.2.14 Integer64

CLASS: Data type

ATTRIBUTES:

1	Data type Numeric Identifier	=	21
2	Data type Name	=	Integer64
3	Format	=	FIXED LENGTH
4.1	Octet Length	=	8

This integer type is a two's complement binary number with a length of eight octets.

5.3.1.7.2.15 LINT

This IEC 61131-3 type is the same as Integer64.

5.3.1.7.3 Unsigned types

5.3.1.7.3.1 Unsigned8

CLASS: Data type

ATTRIBUTES:

1	Data type Numeric Identifier	=	5
2	Data type Name	=	Unsigned8
3	Format	=	FIXED LENGTH
4.1	Octet Length	=	1

This type is a binary number. The most significant bit of the most significant octet is always used as the most significant bit of the binary number; no sign bit is included. This type has a length of one octet.

5.3.1.7.3.2 USINT

This IEC 61131-3 type is the same as Unsigned8.

5.3.1.7.3.3 unsigned char

This data type is the same as Unsigned8.

5.3.1.7.3.4 Unsigned16

CLASS: Data type

ATTRIBUTES:

1	Data type Numeric Identifier	=	6
2	Data type Name	=	Unsigned16
3	Format	=	FIXED LENGTH
4.1	Octet Length	=	2

This type is a binary number. The most significant bit of the most significant octet is always used as the most significant bit of the binary number; no sign bit is included. This unsigned type has a length of two octets.

5.3.1.7.3.5 **UINT**

This IEC 61131-3 type is the same as Unsigned16.

5.3.1.7.3.6 **WORD**

This type is used in the same way as UINT.

5.3.1.7.3.7 **Unsigned24**

CLASS:		Data type
ATTRIBUTES:		
1	Data type Numeric Identifier	= 22
2	Data type Name	= Unsigned24
3	Format	= FIXED LENGTH
4.1	Octet Length	= 3

This type is a binary number. The most significant bit of the most significant octet is always used as the most significant bit of the binary number; no sign bit is included. This unsigned type has a length of three octets.

5.3.1.7.3.8 **Unsigned32**

CLASS:		Data type
ATTRIBUTES:		
1	Data type Numeric Identifier	= 7
2	Data type Name	= Unsigned32
3	Format	= FIXED LENGTH
4.1	Octet Length	= 4

This type is a binary number. The most significant bit of the most significant octet is always used as the most significant bit of the binary number; no sign bit is included. This unsigned type has a length of four octets.

5.3.1.7.3.9 **UDINT**

This IEC 61131-3 type is the same as Unsigned32.

5.3.1.7.3.10 **Unsigned40**

CLASS:		Data type
ATTRIBUTES:		
1	Data type Numeric Identifier	= 24
2	Data type Name	= Unsigned40
3	Format	= FIXED LENGTH
4.1	Octet Length	= 5

This type is a binary number. The most significant bit of the most significant octet is always used as the most significant bit of the binary number; no sign bit is included. This unsigned type has a length of five octets.

5.3.1.7.3.11 **Unsigned48**

CLASS:		Data type
ATTRIBUTES:		
1	Data type Numeric Identifier	= 25
2	Data type Name	= Unsigned48
3	Format	= FIXED LENGTH
4.1	Octet Length	= 6

This type is a binary number. The most significant bit of the most significant octet is always used as the most significant bit of the binary number; no sign bit is included. This unsigned type has a length of six octets.