

INTERNATIONAL STANDARD

NORME INTERNATIONALE



**Industrial communication networks – Fieldbus specifications –
Part 4-12: Data-link layer protocol specification – Type 12 elements**

**Réseaux de communication industriels – Spécification des bus de terrain –
Partie 4-12: Spécification du protocole de la couche liaison de données –
Éléments de Type 12**

INTERNET



THIS PUBLICATION IS COPYRIGHT PROTECTED

Copyright © 2007 IEC, Geneva, Switzerland

All rights reserved. Unless otherwise specified, no part of this publication may be reproduced or utilized in any form or by any means, electronic or mechanical, including photocopying and microfilm, without permission in writing from either IEC or IEC's member National Committee in the country of the requester. If you have any questions about IEC copyright or have an enquiry about obtaining additional rights to this publication, please contact the address below or your local IEC member National Committee for further information.

Droits de reproduction réservés. Sauf indication contraire, aucune partie de cette publication ne peut être reproduite ni utilisée sous quelque forme que ce soit et par aucun procédé, électronique ou mécanique, y compris la photocopie et les microfilms, sans l'accord écrit de l'IEC ou du Comité national de l'IEC du pays du demandeur. Si vous avez des questions sur le copyright de l'IEC ou si vous désirez obtenir des droits supplémentaires sur cette publication, utilisez les coordonnées ci-après ou contactez le Comité national de l'IEC de votre pays de résidence.

IEC Central Office
3, rue de Varembe
CH-1211 Geneva 20
Switzerland

Tel.: +41 22 919 02 11
Fax: +41 22 919 03 00
info@iec.ch
www.iec.ch

About the IEC

The International Electrotechnical Commission (IEC) is the leading global organization that prepares and publishes International Standards for all electrical, electronic and related technologies.

About IEC publications

The technical content of IEC publications is kept under constant review by the IEC. Please make sure that you have the latest edition, a corrigenda or an amendment might have been published.

IEC Catalogue - webstore.iec.ch/catalogue

The stand-alone application for consulting the entire bibliographical information on IEC International Standards, Technical Specifications, Technical Reports and other documents. Available for PC, Mac OS, Android Tablets and iPad.

IEC publications search - www.iec.ch/searchpub

The advanced search enables to find IEC publications by a variety of criteria (reference number, text, technical committee,...). It also gives information on projects, replaced and withdrawn publications.

IEC Just Published - webstore.iec.ch/justpublished

Stay up to date on all new IEC publications. Just Published details all new publications released. Available online and also once a month by email.

Electropedia - www.electropedia.org

The world's leading online dictionary of electronic and electrical terms containing more than 30 000 terms and definitions in English and French, with equivalent terms in 14 additional languages. Also known as the International Electrotechnical Vocabulary (IEV) online.

IEC Glossary - std.iec.ch/glossary

More than 55 000 electrotechnical terminology entries in English and French extracted from the Terms and Definitions clause of IEC publications issued since 2002. Some entries have been collected from earlier publications of IEC TC 37, 77, 86 and CISPR.

IEC Customer Service Centre - webstore.iec.ch/csc

If you wish to give us your feedback on this publication or need further assistance, please contact the Customer Service Centre: csc@iec.ch.

A propos de l'IEC

La Commission Electrotechnique Internationale (IEC) est la première organisation mondiale qui élabore et publie des Normes internationales pour tout ce qui a trait à l'électricité, à l'électronique et aux technologies apparentées.

A propos des publications IEC

Le contenu technique des publications IEC est constamment revu. Veuillez vous assurer que vous possédez l'édition la plus récente, un corrigendum ou amendement peut avoir été publié.

Catalogue IEC - webstore.iec.ch/catalogue

Application autonome pour consulter tous les renseignements bibliographiques sur les Normes internationales, Spécifications techniques, Rapports techniques et autres documents de l'IEC. Disponible pour PC, Mac OS, tablettes Android et iPad.

Recherche de publications IEC - www.iec.ch/searchpub

La recherche avancée permet de trouver des publications IEC en utilisant différents critères (numéro de référence, texte, comité d'études,...). Elle donne aussi des informations sur les projets et les publications remplacées ou retirées.

IEC Just Published - webstore.iec.ch/justpublished

Restez informé sur les nouvelles publications IEC. Just Published détaille les nouvelles publications parues. Disponible en ligne et aussi une fois par mois par email.

Electropedia - www.electropedia.org

Le premier dictionnaire en ligne de termes électroniques et électriques. Il contient plus de 30 000 termes et définitions en anglais et en français, ainsi que les termes équivalents dans 14 langues additionnelles. Egalement appelé Vocabulaire Electrotechnique International (IEV) en ligne.

Glossaire IEC - std.iec.ch/glossary

Plus de 55 000 entrées terminologiques électrotechniques, en anglais et en français, extraites des articles Termes et Définitions des publications IEC parues depuis 2002. Plus certaines entrées antérieures extraites des publications des CE 37, 77, 86 et CISPR de l'IEC.

Service Clients - webstore.iec.ch/csc

Si vous désirez nous donner des commentaires sur cette publication ou si vous avez des questions contactez-nous: csc@iec.ch.

INTERNATIONAL STANDARD

NORME INTERNATIONALE



**Industrial communication networks – Fieldbus specifications –
Part 4-12: Data-link layer protocol specification – Type 12 elements**

**Réseaux de communication industriels – Spécification des bus de terrain –
Partie 4-12: Spécification du protocole de la couche liaison de données –
Éléments de Type 12**

INTERNATIONAL
ELECTROTECHNICAL
COMMISSION

COMMISSION
ELECTROTECHNIQUE
INTERNATIONALE

PRICE CODE
CODE PRIX

XG

ICS 25.040.40; 35.100.20

ISBN 978-2-8322-1490-9

**Warning! Make sure that you obtained this publication from an authorized distributor.
Attention! Veuillez vous assurer que vous avez obtenu cette publication via un distributeur agréé.**

CONTENTS

FOREWORD.....	6
INTRODUCTION.....	8
1 Scope.....	9
2 Normative references	10
3 Terms, definitions, symbols and abbreviations.....	10
4 Overview of the DL-protocol	23
5 Frame structure	27
6 Attributes.....	48
7 DL-user memory.....	92
8 Type 12: FDL protocol state machines.....	98
Annex A (informative) Type 12: Additional specifications on DL-Protocol state machines	106
Bibliography.....	142
Figure 1 – Type description example	19
Figure 2 – Common structure of specific fields.....	20
Figure 3 – Frame structure.....	24
Figure 4 – Mapping of data in a frame.....	25
Figure 5 – Slave node reference model.....	26
Figure 6 – Type 12 PDUs embedded in Ethernet frame.....	27
Figure 7 – Type 12 PDUs embedded in UDP/IP.....	27
Figure 8 – DL information type description.....	50
Figure 9 – Address type description.....	52
Figure 10 – DL control type description.....	54
Figure 11 – DL status type description.....	56
Figure 12 – Successful write sequence to DL-user control register	58
Figure 13 – Successful read sequence to the DL-user status register	58
Figure 14 – RX error counter type description.....	65
Figure 15 – Lost link counter type description	66
Figure 16 – Additional counter type description.....	67
Figure 17 – Sync configuration type description	68
Figure 18 – Watchdog divider type description.....	68
Figure 19 – Sync manager watchdog type description.....	69
Figure 20 – Sync manager watchdog status type description	69
Figure 21 – Watchdog counter type description.....	70
Figure 22 – Slave information interface access type description	70
Figure 23 – Slave information interface control/status type description	72
Figure 24 – Slave information interface address type description	74
Figure 25 – Slave information interface data type description	75
Figure 26 – MII control/status type description	76
Figure 27 – MII address type description	77

Figure 28 – MII data type description	78
Figure 29 – FMMU mapping example	78
Figure 30 – FMMU entity type description	79
Figure 31 – SyncM mailbox interaction	82
Figure 32 – SyncM buffer allocation	82
Figure 33 – SyncM buffer interaction	83
Figure 34 – Handling of write/read toggle with read mailbox	84
Figure 35 – Sync manager channel type description	86
Figure 36 – Distributed clock local time parameter type description	90
Figure 37 – Successful write sequence to mailbox	93
Figure 38 – Bad write sequence to mailbox	94
Figure 39 – Successful read sequence to mailbox	94
Figure 40 – Bad read sequence to mailbox	95
Figure 41 – Successful write sequence to buffer	96
Figure 42 – Successful read sequence to buffer	97
Figure 43 – Structuring of the protocol machines of an slave	98
Figure 44 – Slave information interface read operation	100
Figure 45 – Slave information interface write operation	101
Figure 46 – Slave information interface reload operation	102
Figure 47 – Distributed clock	104
Figure 48 – Delay measurement sequence	105
Table 1 – PDU element description example	19
Table 2 – Example attribute description	20
Table 3 – State machine description elements	21
Table 4 – Description of state machine elements	22
Table 5 – Conventions used in state machines	22
Table 6 – Transfer Syntax for bit sequences	28
Table 7 – Transfer syntax for data type Unsignedn	29
Table 8 – Transfer syntax for data type Integern	29
Table 9 – Type 12 frame inside an Ethernet frame	30
Table 10 – Type 12 frame inside an UDP PDU	31
Table 11 – Type 12 frame structure containing Type 12 PDUs	32
Table 12 – Type 12 frame structure containing network variables	32
Table 13 – Type 12 frame structure containing mailbox	32
Table 14 – Auto increment physical read (APRD)	33
Table 15 – Configured address physical read (FPRD)	34
Table 16 – Broadcast read (BRD)	35
Table 17 – Logical read (LRD)	36
Table 18 – Auto Increment physical write (APWR)	37
Table 19 – Configured address physical write (FPWR)	38
Table 20 – Broadcast write (BWR)	39
Table 21 – Logical write (LWR)	40

Table 22 – Auto increment physical read write (APRW)	41
Table 23 – Configured address physical read write (FPRW).....	42
Table 24 – Broadcast read write (BRW)	43
Table 25 – Logical read write (LRW)	44
Table 26 – Auto increment physical read multiple write (ARMW).....	45
Table 27 – Configured address physical read multiple write (FRMW)	46
Table 28 – Network variable	47
Table 29 – Mailbox	47
Table 30 – Error Reply Service Data	48
Table 31 – DL information	51
Table 32 – Configured station address	53
Table 33 – DL control	54
Table 34 – DL status	57
Table 35 – DLS-user specific registers	59
Table 36 – DLS-user event	61
Table 37 – DLS-user event mask	62
Table 38 – External event	63
Table 39 – External event mask	64
Table 40 – RX error counter	65
Table 41 – Lost link counter	66
Table 42 – Additional counter	67
Table 43 – Watchdog divider	68
Table 44 – DLS-user watchdog	68
Table 45 – Sync manager channel watchdog	69
Table 46 – Sync manager watchdog Status	69
Table 47 – Watchdog counter	70
Table 48 – Slave information interface size	71
Table 49 – Slave information interface control/status	73
Table 50 – Actual slave information interface address	74
Table 51 – Actual slave information interface data	75
Table 52 – MII control/status	76
Table 53 – Actual MII address	77
Table 54 – Actual MII data	78
Table 55 – Fieldbus memory management unit (FMMU) entity	80
Table 56 – Fieldbus memory management unit (FMMU).....	81
Table 57 – Sync manager channel	86
Table 58 – Sync manager Structure	88
Table 59 – Distributed clock local time parameter	91
Table 60 – Distributed clock DLS-user parameter	92
Table A.1 – Primitives issued by DHSM to PSM	106
Table A.2 – Primitives issued by PSM to DHSM	106
Table A.3 – Parameters used with primitives exchanged between DHSM and PSM.....	106
Table A.4 – Identifier for the octets of a Ethernet frame	107

Table A.5 – DHSM state table	109
Table A.6 – DHSM function table	125
Table A.7 – Primitives issued by SYSM to DHSM.....	125
Table A.8 – Primitives issued by DHSM to SYSM.....	126
Table A.9 – Primitives issued by DL-User to SYSM.....	126
Table A.10 – Primitives issued by SYSM to DL-User	126
Table A.11 – Parameters used with primitives exchanged between SYSM and DHSM	126
Table A.12 – SYSM state table	128
Table A.13 – SYSM function table.....	138
Table A.14 – Primitives issued by RMSM to SYSM	138
Table A.15 – Primitives issued by SYSM to RMSM	138
Table A.16 – Parameters used with primitives exchanged between RMSM and SYSM	138
Table A.17 – RMSM state table.....	140
Table A.18 – RMSM function table	141

Withdrawn

INTERNATIONAL ELECTROTECHNICAL COMMISSION

**INDUSTRIAL COMMUNICATION NETWORKS –
FIELDBUS SPECIFICATIONS –****Part 4-12: Data-link layer protocol specification – Type 12 elements**

FOREWORD

- 1) The International Electrotechnical Commission (IEC) is a worldwide organization for standardization comprising all national electrotechnical committees (IEC National Committees). The object of IEC is to promote international co-operation on all questions concerning standardization in the electrical and electronic fields. To this end and in addition to other activities, IEC publishes International Standards, Technical Specifications, Technical Reports, Publicly Available Specifications (PAS) and Guides (hereafter referred to as "IEC Publication(s)"). Their preparation is entrusted to technical committees; any IEC National Committee interested in the subject dealt with may participate in this preparatory work. International, governmental and non-governmental organizations liaising with the IEC also participate in this preparation. IEC collaborates closely with the International Organization for Standardization (ISO) in accordance with conditions determined by agreement between the two organizations.
- 2) The formal decisions or agreements of IEC on technical matters express, as nearly as possible, an international consensus of opinion on the relevant subjects since each technical committee has representation from all interested IEC National Committees.
- 3) IEC Publications have the form of recommendations for international use and are accepted by IEC National Committees in that sense. While all reasonable efforts are made to ensure that the technical content of IEC Publications is accurate, IEC cannot be held responsible for the way in which they are used or for any misinterpretation by any end user.
- 4) In order to promote international uniformity, IEC National Committees undertake to apply IEC Publications transparently to the maximum extent possible in their national and regional publications. Any divergence between any IEC Publication and the corresponding national or regional publication shall be clearly indicated in the latter.
- 5) IEC provides no marking procedure to indicate its approval and cannot be rendered responsible for any equipment declared to be in conformity with an IEC Publication.
- 6) All users should ensure that they have the latest edition of this publication.
- 7) No liability shall attach to IEC or its directors, employees, servants or agents including individual experts and members of its technical committees and IEC National Committees for any personal injury, property damage or other damage of any nature whatsoever, whether direct or indirect, or for costs (including legal fees) and expenses arising out of the publication, use of, or reliance upon, this IEC Publication or any other IEC Publications.
- 8) Attention is drawn to the Normative references cited in this publication. Use of the referenced publications is indispensable for the correct application of this publication.

NOTE Use of some of the associated protocol types is restricted by their intellectual-property-right holders. In all cases, the commitment to limited release of intellectual-property-rights made by the holders of those rights permits a particular data-link layer protocol type to be used with physical layer and application layer protocols in Type combinations as specified explicitly in the IEC 61784 series. Use of the various protocol types in other combinations may require permission from their respective intellectual-property-right holders.

International Standard IEC 61158-4-12 has been prepared by subcommittee 65C: Industrial networks, of IEC technical committee 65: Industrial-process measurement, control and automation.

This bilingual version (2014-06) corresponds to the English version, published in 2007-12.

This first edition and its companion parts of the IEC 61158-4 subseries cancel and replace IEC 61158-4:2003. This edition of this part constitutes a technical addition, which also replaces IEC/PAS 62407, published in 2005.

This edition of IEC 61158-4 includes the following significant changes from the previous edition:

- a) deletion of the former Type 6 fieldbus, and the placeholder for a Type 5 fieldbus data link layer, for lack of market relevance;
- b) addition of new types of fieldbuses;
- c) division of this part into multiple parts numbered -4-1, -4-2, ..., -4-19.

The text of this standard is based on the following documents:

FDIS	Report on voting
65C/474/FDIS	65C/485/RVD

Full information on the voting for the approval of this standard can be found in the report on voting indicated in the above table.

The French version of this standard has not been voted upon.

This publication has been drafted in accordance with ISO/IEC Directives, Part 2.

The committee has decided that the contents of this publication will remain unchanged until the maintenance result date indicated on the IEC web site under <http://webstore.iec.ch> in the data related to the specific publication. At this date, the publication will be:

- reconfirmed;
- withdrawn;
- replaced by a revised edition, or
- amended.

NOTE The revision of this standard will be synchronized with the other parts of the IEC 61158 series.

The list of all the parts of the IEC 61158 series, under the general title *Industrial communication networks – Fieldbus specifications*, can be found on the IEC web site.

IMPORTANT – The 'colour inside' logo on the cover page of this publication indicates that it contains colours which are considered to be useful for the correct understanding of its contents. Users should therefore print this document using a colour printer.

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 data-link protocol provides the data-link service by making use of the services available from the physical layer. The primary aim of this standard is to provide a set of rules for communication expressed in terms of the procedures to be carried out by peer data-link entities (DLEs) at the time of communication. These rules for communication are intended to provide a sound basis for development in order to serve a variety of purposes:

- a) as a guide for implementors and designers;
- b) for use in the testing and procurement of equipment;
- c) as part of an agreement for the admittance of systems into the open systems environment;
- d) as a refinement to the understanding of time-critical communications within OSI.

This standard is concerned, in particular, with the communication and interworking of sensors, effectors and other automation devices. By using this standard together with other standards positioned within the OSI or fieldbus reference models, otherwise incompatible systems may work together in any combination.

Withdrawn

INDUSTRIAL COMMUNICATION NETWORKS – FIELDBUS SPECIFICATIONS –

Part 4-12: Data-link layer protocol specification – Type 12 elements

1 Scope

1.1 General

The data-link layer provides basic time-critical messaging communications between devices in an automation environment.

This protocol provides communication opportunities to all participating data-link entities

- a) in a synchronously-starting cyclic manner, and
- b) in a cyclic or acyclic asynchronous manner, as requested each cycle by each of those data-link entities.

Thus this protocol can be characterized as one which provides cyclic and acyclic access asynchronously but with a synchronous restart of each cycle.

1.2 Specifications

This standard specifies

- a) procedures for the transfer of data and control information from one data-link user entity to one or more user entity;
- b) the structure of the DLPDUs used for the transfer of data and control information by the protocol of this standard, and their representation as physical interface data units.

1.3 Procedures

The procedures are defined in terms of

- a) the interactions between DL-entities (DLEs) through the exchange of DLPDUs;
- b) the interactions between a DL-service (DLS) provider and a DLS-user in the same system through the exchange of DLS primitives;
- c) the interactions between a DLS-provider and the MAC services of ISO/IEC 8802-3.

1.4 Applicability

These procedures are applicable to instances of communication between systems which support time-critical communications services within the data-link layer of the OSI reference model, and which require the ability to interconnect in an open systems interconnection environment.

Profiles provide a simple multi-attribute means of summarizing an implementation's capabilities, and thus its applicability to various time-critical communications needs.

1.5 Conformance

This standard also specifies conformance requirements for systems implementing these procedures. This part of this standard does not contain tests to demonstrate compliance with such requirements.

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 61158-2 (Ed.4.0), *Industrial communication networks – Fieldbus specifications – Part 2: Physical layer specification and service definition*

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

IEC 61588, *Precision clock synchronization protocol for networked measurement and control system*

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

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

ISO/IEC 8802-1(E), ANSI/IEEE Std 802.1 *Local and metropolitan area networks – Virtual bridged local area networks.*

ISO/IEC 8802-3:2000, *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 9899, *Programming Languages – C.*

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

IEEE 802.1Q, *IEEE Standard for Local and metropolitan area networks – Virtual Bridged Local Area Networks*; available at <<http://www.ieee.org>>

IETF RFC 768, *User PDU Protocol*; available at <<http://www.ietf.org>>

IETF RFC 791, *Internet Protocol*; available at <<http://www.ietf.org>>

3 Terms, definitions, symbols and abbreviations

For the purposes of this document, the following terms, definitions, symbols and abbreviations 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 DL-duplex-transmission [7498-1]

3.1.2 DL-protocol [7498-1]

3.1.3 DL-protocol-data-unit

3.1.4 (N)-entity DL-entity Ph-entity	[7498-1]
3.1.5 (N)-interface-data-unit DL-service-data-unit (N=2) Ph-interface-data-unit (N=1)	[7498-1]
3.1.6 (N)-layer DL-layer (N=2) Ph-layer (N=1)	[7498-1]
3.1.7 (N)-service DL-service (N=2) Ph-service (N=1)	[7498-1]
3.1.8 (N)-service-access-point DL-service-access-point (N=2) Ph-service-access-point (N=1)	[7498-1]
3.1.9 (N)-service-access-point-address DL-service-access-point-address (N=2) Ph-service-access-point-address (N=1)	[7498-1]
3.1.10 peer-entities	[7498-1]
3.1.11 Ph-interface-data	[7498-1]
3.1.12 primitive name	[7498-3]
3.1.13 reassembling	[7498-1]
3.1.14 recombining	[7498-1]
3.1.15 reset	[7498-1]
3.1.16 routing	[7498-1]
3.1.17 segmenting	[7498-1]
3.1.18 sequencing	[7498-1]
3.1.19 splitting	[7498-1]
3.1.20 systems-management	[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 asymmetrical service**
- 3.2.2 confirm (primitive);**
requestor.deliver (primitive)
- 3.2.3 deliver (primitive)**
- 3.2.4 DL-service-primitive;**
primitive
- 3.2.5 DL-service-provider**
- 3.2.6 DL-service-user**

3.2.7 indication (primitive)
acceptor.deliver (primitive)

3.2.8 request (primitive);
requestor.submit (primitive)

3.2.9 requestor

3.2.10 response (primitive);
acceptor.submit (primitive)

3.2.11 submit (primitive)

3.2.12 symmetrical service

3.3 Common terms and definitions

NOTE Many definitions are common to more than one protocol Type; they are not necessarily used by all protocol Types.

For the purpose of this part of IEC 61158, the following definitions also apply:

3.3.1
frame
denigrated synonym for DLPDU

3.3.2
group DL-address
DL-address that potentially designates more than one DLSAP within the extended link. A single DL-entity may have multiple group DL-addresses associated with a single DLSAP. A single DL-entity also may have a single group DL-address associated with more than one DLSAP

3.3.3
node
single DL-entity as it appears on one local link

3.3.4
receiving DLS-user
DL-service user that acts as a recipient of DLS-user-data

NOTE A DL-service user can be concurrently both a sending and receiving DLS-user.

3.3.5
sending DLS-user
DL-service user that acts as a source of DLS-user-data

3.4 Additional Type 12 definitions

3.4.1
application
function or data structure for which data is consumed or produced [IEC 61158-5:2003]

3.4.2
application objects
multiple object classes that manage and provide a run time exchange of messages across the network and within the network device

3.4.3
basic slave
slave device that supports only physical addressing of data

3.4.4**bit**

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

3.4.5**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

3.4.6**connection**

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

3.4.7**cyclic**

events which repeat in a regular and repetitive manner

3.4.8**Cyclic Redundancy Check (CRC)**

residual value computed from an array of data and used as a representative signature for the array

3.4.9**data**

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

3.4.10**data consistency**

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

3.4.11**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.) [IEC 61158-2:2003]

3.4.12**distributed clocks**

method to synchronize slaves and maintain a global time base

3.4.13**error**

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

3.4.14**event**

instance of a change of conditions

3.4.15

Fieldbus Memory Management Unit

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

3.4.16

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

3.4.17

full slave

slave device that supports both physical and logical addressing of data

3.4.18

interface

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

3.4.19

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

3.4.20

mapping

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

3.4.21

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.

3.4.22

message

ordered series of octets intended to convey information

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

3.4.23

network

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

3.4.24

node

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

3.4.25**object**

abstract representation of a particular component within a device

NOTE An object can be

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

- a) data (information which changes with time);
- b) configuration (parameters for behavior);
- c) methods (things that can be done using data and configuration);

2) 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.

3.4.26**process data**

data object containing application objects designated to be transferred cyclically or acyclically for the purpose of processing

3.4.27**server**

object which provides services to another (client) object

3.4.28**service**

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

3.4.29**slave**

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

3.4.30**Sync manager**

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

3.4.31**Sync manager channel**

single control elements to coordinate access to concurrently used objects.

3.4.32**switch**

MAC bridge as defined in IEEE 802.1D

3.5 Common symbols and abbreviations

NOTE Many symbols and abbreviations are common to more than one protocol Type; they are not necessarily used by all protocol Types.

3.5.1 DL- Data-link layer (as a prefix)

3.5.2 DLC DL-connection

3.5.3 DLCEP DL-connection-end-point

3.5.4 DLE DL-entity (the local active instance of the data-link layer)

3.5.5 DLL DL-layer

3.5.6 DLPCI	DL-protocol-control-information
3.5.7 DLPDU	DL-protocol-data-unit
3.5.8 DLM	DL-management
3.5.9 DLME	DL-management Entity (the local active instance of DL-management)
3.5.10 DLMS	DL-management Service
3.5.11 DLS	DL-service
3.5.12 DLSAP	DL-service-access-point
3.5.13 DLSDU	DL-service-data-unit
3.5.14 FIFO	First-in first-out (queuing method)
3.5.15 OSI	Open systems interconnection
3.5.16 Ph-	Physical layer (as a prefix)
3.5.17 PhE	Ph-entity (the local active instance of the physical layer)
3.5.18 PhL	Ph-layer
3.5.19 QoS	Quality of service

3.6 Additional Type 12 symbols and abbreviations

3.6.1 AL	Application layer
3.6.2 DLSDU	Data-link protocol data unit
3.6.3 APRD	Auto increment physical read
3.6.4 APRW	Auto increment physical read write
3.6.5 APWR	Auto increment physical write
3.6.6 ARMW	Auto increment physical read multiple write
3.6.7 BRD	Broadcast read
3.6.8 BRW	Broadcast read write
3.6.9 BWR	Broadcast write
3.6.10 CAN	Controller area network
3.6.11 CoE	CANopen over Type 12 services
3.6.12 CSMA/CD	Carrier sense multiple access with collision detection
3.6.13 DC	Distributed clocks

3.6.14 Type 12	Prefix for DL services and protocols
3.6.15 E²PROM	Electrically erasable programmable read only memory
3.6.16 EoE	Ethernet tunneled over Type 12 services
3.6.17 ESC	Type 12 slave controller
3.6.18 FCS	frame check sequence
3.6.19 FMMU	Fieldbus memory management unit
3.6.20 FoE	File access with Type 12 services
3.6.21 FPRD	Configured address physical read
3.6.22 FPRW	Configured address physical read write
3.6.23 FPWR	Configured address physical write
3.6.24 FRMW	Configured address physical read multiple write
3.6.25 HDR	Header
3.6.26 ID	Identifier
3.6.27 IP	Internet protocol
3.6.28 LAN	Local area network
3.6.29 LRD	Logical memory read
3.6.30 LRW	Logical memory read write
3.6.31 LWR	Logical memory write
3.6.32 MAC	Media access control
3.6.33 MDI	Media dependent interface (specified in ISO/IEC 8802-3)
3.6.34 MDX	Mailbox data exchange
3.6.35 MII	Media independent interface (specified in ISO/IEC 8802-3)
3.6.36 PDI	Physical device interface (a set of elements that allows access to DL services from the DLS-user)
3.6.37 PDO	Process data object
3.6.38 PHY	Physical layer device (specified in ISO/IEC 8802-3)
3.6.39 PNV	Publish network variable

3.6.40 RAM	Random access memory
3.6.41 Rx	Receive
3.6.42 SDO	Service data object
3.6.43 SII	slave information interface
3.6.44 SyncM	Synchronization manager
3.6.45 TCP	Transmission control protocol
3.6.46 Tx	Transmit
3.6.47 UDP	User datagram protocol
3.6.48 WKC	Working counter

3.7 Conventions

3.7.1 General concept

The services are specified in IEC 61158-3-12. The service specification defines the services that are provided by the Type 12 DL. The mapping of these services to ISO/IEC 8802-3 is described in this international Standard.

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

3.7.1.1 Abstract syntax conventions

The DL syntax elements related to PDU structure are described as shown in the example of Table 1.

Frame part denotes the element that will be replaced by this reproduction.

Data field is the name of the elements.

Data Type denotes the type of the terminal symbol.

Value/Description contains the constant value or the meaning of the parameter.

Table 1 – PDU element description example

Frame part	Data Field	Data Type	Value/Description
Type 12 xxx	CMD	Unsigned8	0x01
	IDX	Unsigned8	Index
	ADP	Unsigned16	Auto Increment Address
	ADO	Unsigned16	Physical Memory Address
	LEN	Unsigned11	Length of data of YYY in octets
	Reserved	Unsigned4	0x00
	NEXT	Unsigned1	0x00: last Type 12 PDU 0x01: Type 12 PDU follows
	IRQ	Unsigned16	Reserved for future use
	YYY		next element see nn
	WKC	Unsigned16	Working Counter

The attribute types are described in C language notations (ISO/IEC 9899) as shown in Figure 1. BYTE and WORD are elements of type unsigned char and unsigned short.

```

typedef struct
{
    Unsigned8      Type;
    Unsigned8      Revision;
    Unsigned16     Build;
    Unsigned8      NoOfSuppFmmuChannels;
    Unsigned8      NoOfSuppSyncManChannels;
    Unsigned8      RamSize;
    Unsigned8      Reserved1;
    unsigned       FmmuBitOperationNotSupp: 1;
    unsigned       Reserved2: 7;
    unsigned       Reserved3: 8;
} TDLINFORMATION;

```

Figure 1 – Type description example

The attributes itself are described in a form as shown in Table 2.

Parameter describes a single element of the attribute.

Physical address denotes the location in physical address space.

Data Type denotes the type of this element.

Access type Type 12 DL/PDI shows the access right to this element. R means read access right, W means write access right. If neither Type 12 DL nor PDI has write access, this variable will be initialised and maintained by DL itself.

Value/Description contains the constant value and/or the meaning of the parameter.

Table 2 – Example attribute description

Parameter	Physical Address	Data Type	Access type	Access Type PDI	Value/Description
State	0x0120	Unsigned4	RW	R	0x01: Init Request 0x02: Pre-Operational Request 0x03: Bootstrap Mode Request 0x04: Safe Operational Request 0x08: Operational Request
Acknowledge	0x0120	Unsigned1	RW	R	0x00: no acknowledge 0x01: acknowledge (must be a positive edge)
Reserved	0x0120	Unsigned3	RW	R	0x00
Application Specific	0x0121	Unsigned8	RW	R	

3.7.1.2 Convention for the encoding of reserved bits and octets

The term "reserved" may be used to describe bits in octets or whole octets. All bits or octets that are reserved should be set to zero at the sending side and shall not be tested at the receiving side except it is explicitly stated or if the reserved bits or octets are checked by a state machine.

The term "reserved" may also be used to indicate that certain values within the range of a parameter are reserved for future extensions. In this case the reserved values should not be used at the sending side and shall not be tested at the receiving side except it is explicitly stated or if the reserved values are checked by a state machine.

3.7.1.3 Conventions for the common coding s of specific field octets

DLSDUs may contain specific fields that carry information in a primitive and condensed way. These fields shall be coded in the order according to Figure 2.

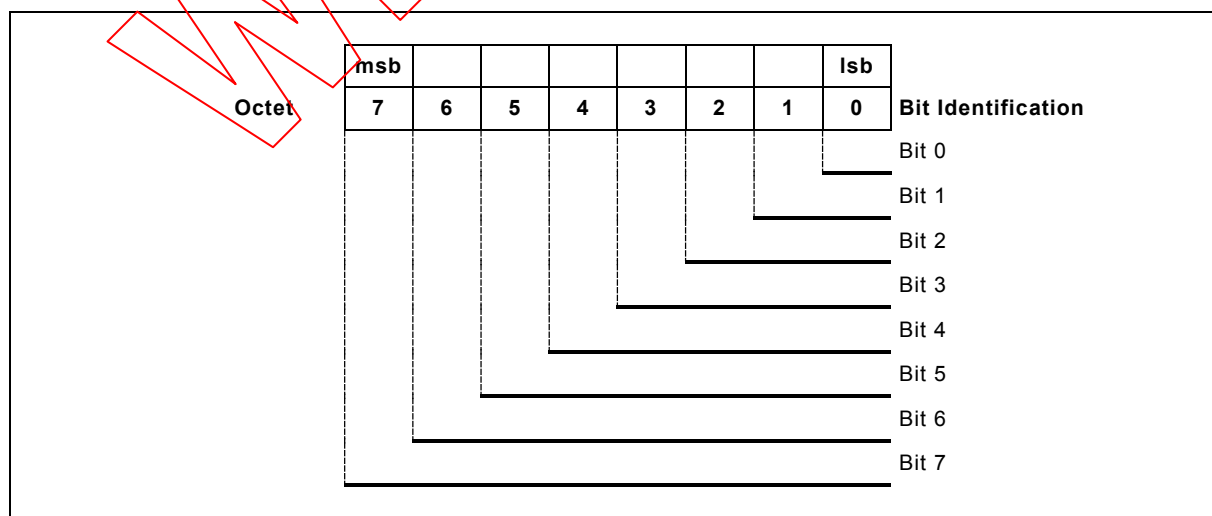


Figure 2 – Common structure of specific fields

Bits may be grouped as group of bits. Each bit or group of bits shall be addressed by its Bit Identification (e.g. Bit 0, Bit 1 to 4). The position within the octet shall be according to the

figure above. Alias names may be used for each bit or group of bits or they may be marked as reserved. The grouping of individual bits shall be in ascending order without gaps. The values for a group of bits may be represented as binary, decimal or hexadecimal values. This value shall only be valid for the grouped bits and can only represent the whole octet if all 8 bits are grouped. Decimal or hexadecimal values shall be transferred in binary values so that the bit with the highest number of the group represents the msb concerning the grouped bits.

EXAMPLE 1: Description and relation for the specific field octet

Bit 0: reserved.

Bit 1-3: Reason_Code The decimal value 2 for the Reason_Code means general error.

Bit 4-7: shall always set to one.

The octet that is constructed according to the description above looks as follows:

(msb) Bit 7 = 1,

Bit 6 = 1,

Bit 5 = 1,

Bit 4 = 1,

Bit 3 = 0,

Bit 2 = 1,

Bit 1 = 0,

(lsb) Bit 0 = 0.

This bit combination has an octet value representation of 0xf4.

3.7.2 State machine conventions

The protocol sequences are described by means of State Machines.

In state diagrams states are represented as boxes state transitions are shown as arrows. Names of states and transitions of the state diagram correspond to the names in the textual listing of the state transitions.

The textual listing of the state transitions is structured as follows, see also Table 3.

The first row contains the name of the transition.

The second row in define the current state.

The third row contains an optional event followed by Conditions starting with a "/" as first line character and finally followed by the actions starting with a "=>" as first line character.

The last row contains the next state.

If the event occurs and the conditions are fulfilled the transition fires, i.e. the actions are executed and the next state is entered.

The layout of a Machine description is shown in Table 3. The meaning of the elements of a State Machine Description are shown in Table 4.

Table 3 – State machine description elements

#	Current state	Event /Condition => Action	Next state

Table 4 – Description of state machine elements

Description element	Meaning
Current state	Name of the given states.
Next state	
#	Name or number of the state transition.
Event	Name or description of the event.
/Condition	Boolean expression. The preceding “\” is not part of the condition.
=> Action	List of assignments and service or function invocations. The preceding “=>” is not part of the action.

The conventions used in the state machines are shown in Table 5.

Table 5 – Conventions used in state machines

Convention	Meaning
=	Value of an item on the left is replaced by value of an item on the right. If an item on the right is a parameter, it comes from the primitive shown as an input event.
axx	A parameter name if a is a letter. Example: Identifier = reason means value of a 'reason' parameter is assigned to a parameter called 'Identifier.'
"xxx"	Indicates fixed visible string. Example: Identifier = "abc" means value "abc" is assigned to a parameter named 'Identifier.'
nnn	if all elements are digits, the item represents a numerical constant shown in decimal representation
0xnn	if all elements nn are digits, the item represents a numerical constant shown in hexadecimal representation
==	A logical condition to indicate an item on the left is equal to an item on the right.
<	A logical condition to indicate an item on the left is less than the item on the right.
>	A logical condition to indicate an item on the left is greater than the item on the right.
!=	A logical condition to indicate an item on the left is not equal to an item on the right.
&&	Logical "AND"
	Logical "OR"
!	Logical "NOT"
+ - * /	Arithmetic operators
;	Separator of expressions

Readers are strongly recommended to refer to the subclauses for the attribute definitions, the local functions, and the FDL-PDU definitions to understand protocol machines. It is assumed that readers have sufficient knowledge of these definitions and they are used without further explanations.

Further constructs as defined in C language notation (ISO/IEC 9899) can be used to describe conditions and actions.

4 Overview of the DL-protocol

4.1 Operating principle

Type 12 DL is 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, an 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 Type 12 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 mode of Ethernet: both communication directions are operated independently. Direct communication without switch between a master device and an Type 12 segment consisting of one or several slave devices may be established.

4.2 Topology

The topology of a communication system is one of the crucial factors for the successful application in automation. The topology has significant influence on the cabling effort, diagnostic features, redundancy options and hot-plug-and-play features.

The star topology commonly used for Ethernet leads to enhanced cabling effort and infrastructure costs. Especially for automation applications a line or tree topology is preferable.

The slave node arrangement represents an open ring bus. At the open end, the master device sends frames, either directly or via Ethernet switches, and receives them at the other end after they have been processed. All frames are relayed from the first node to the next ones. The last node returns the PDU back to the master. Utilizing the full duplex capabilities of Ethernet, the resulting topology is a physical line.

Branches, which in principle are possible anywhere, can be used to enhance the line structure into a tree structure from. A tree structure supports very simple wiring; individual branches, for example, can branch into control cabinets or machine modules, while the main line runs from one module to the next.

4.3 Frame processing principles

In order to achieve maximum performance, the Ethernet frames should be processed directly “on the fly”. If it is implemented this way, the slave node recognizes relevant commands and executes them accordingly while the frames are already passed on.

NOTE Type 12 DL can be implemented using standard Ethernet controllers without direct processing. The influence of the forwarding mechanism implementation on communication performance is detailed in IEC 61784-2.

The nodes have an addressable memory that can be accessed with read or write services, either each node consecutively or several nodes simultaneously. Several Type 12 PDUs can be embedded within an Ethernet frame, each PDU addressing a cohesive data section. As shown in Figure 3, the Type 12 PDUs are either transported:

- a) directly in the data area of the Ethernet frame,
- b) within the data section of a UDP datagram transported via IP.

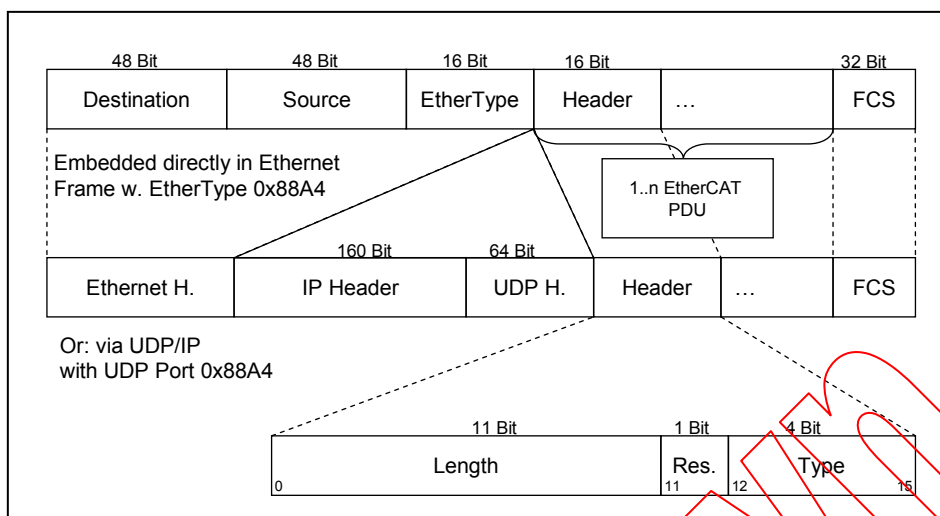


Figure 3 – Frame structure

Variant a) is limited to one Ethernet subnet, since associated frames are not relayed by routers. For machine control applications this usually does not represent a constraint. Multiple Type 12 segments can be connected to one or several switches. The Ethernet MAC address of the first node within the segment is used for addressing the Type 12 segment.

NOTE Further addressing details are given in the data-link layer service definition(IEC 61158-3-12).

Variant b) via UDP/IP generates a slightly larger overhead (IP and UDP header), but for less time-critical applications such as building automation it allows using IP routing. On the master side any standard UDP/IP implementation can be used.

4.4 Data-link layer overview

Several nodes can be addressed individually via a single Ethernet frame carrying several Type 12 PDUs. The Type 12 PDUs are packed without gaps. The frame is terminated with the last Type 12 PDU, unless the frame size is less than 64 octets, in which case the frame will be padded to 64 octets in length.

Compared with one frame per node this leads to a better utilization of the Ethernet bandwidth. However, for e.g. a 2 channel digital input node with just 2 bit of user data, the overhead of a single Type 12 PDU is still excessive.

Therefore the slave nodes may also support logical address mapping. The process data can be inserted anywhere within a logical address space. If an Type 12 PDU is sent that contains read or write services for a certain process image area located at the corresponding logical address, instead of addressing a particular node, the nodes insert the data at or extract the data from the right place within the process data, as noted in Figure 4.

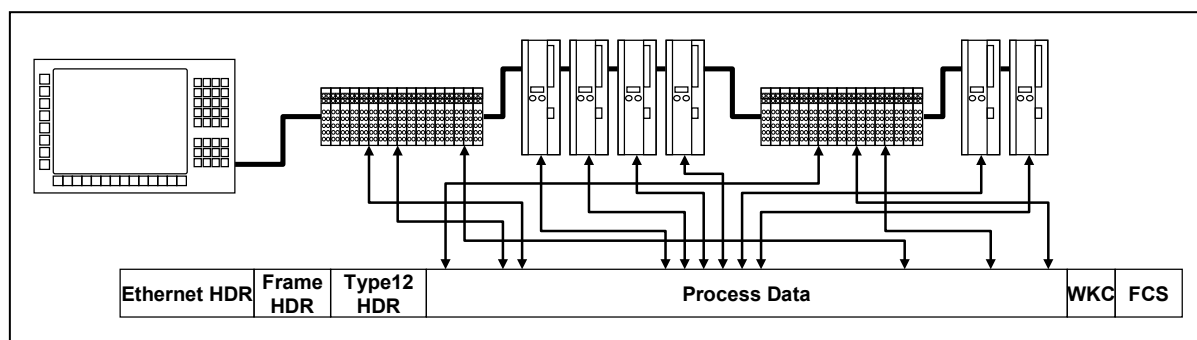


Figure 4 – Mapping of data in a frame

All other nodes that also detect an address match with the process image also insert their data, so that many nodes can be addressed simultaneously with a single Type 12 PDU. The master can assemble completely sorted logical process images via a single Type 12 PDU. Additional mapping is no longer required in the master, so that the process data can be assigned directly to the different control tasks. Each task can create its own process image and exchange it within its own timeframe. The physical order of the nodes is completely arbitrary and is only relevant during the first initialization phase.

The logical address space is 2^{32} Bytes = 4 GByte. Type 12 DL can be considered to be a serial backplane for automation systems that enables connection to distributed process data for both large and very small automation devices. Using a standard Ethernet controller and a standard Ethernet cable, a very large number of I/O channels without practical restrictions on the distribution can be connected to automation devices, which can be accessed with high bandwidth, minimum delay and near-optimum usable data rate. At the same time, devices such as fieldbus scanners can be connected as well, thus preserving existing technologies and standards.

4.5 Error detection overview

Type 12 DL checks by the Ethernet frame check sequence (FCS) whether a frame was transmitted correctly. Since one or several slaves modify the frame during the transfer, the FCS is recalculated by each slave. If a slave detects a checksum error, the slave does not repair the FCS but flags the master by incrementing the error counter, so that a fault can be located precisely.

When reading data from or writing data to an Type 12 PDU, the addressed slave increments a working counter (WKC) positioned at the end of each Type 12 PDU. Analyzing the working counter allows the master to check if the expected number of nodes has processed the corresponding Type 12 PDU.

4.6 Node reference model

4.6.1 Mapping onto OSI basic reference model

Type 12 DL 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 DL 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 DL data-link layer or the Type 12 DL Application layer. Likewise, features common to users of the Fieldbus Application layer may be provided by the Type 12 DL Application layer to simplify user operation, as noted in Figure 5.

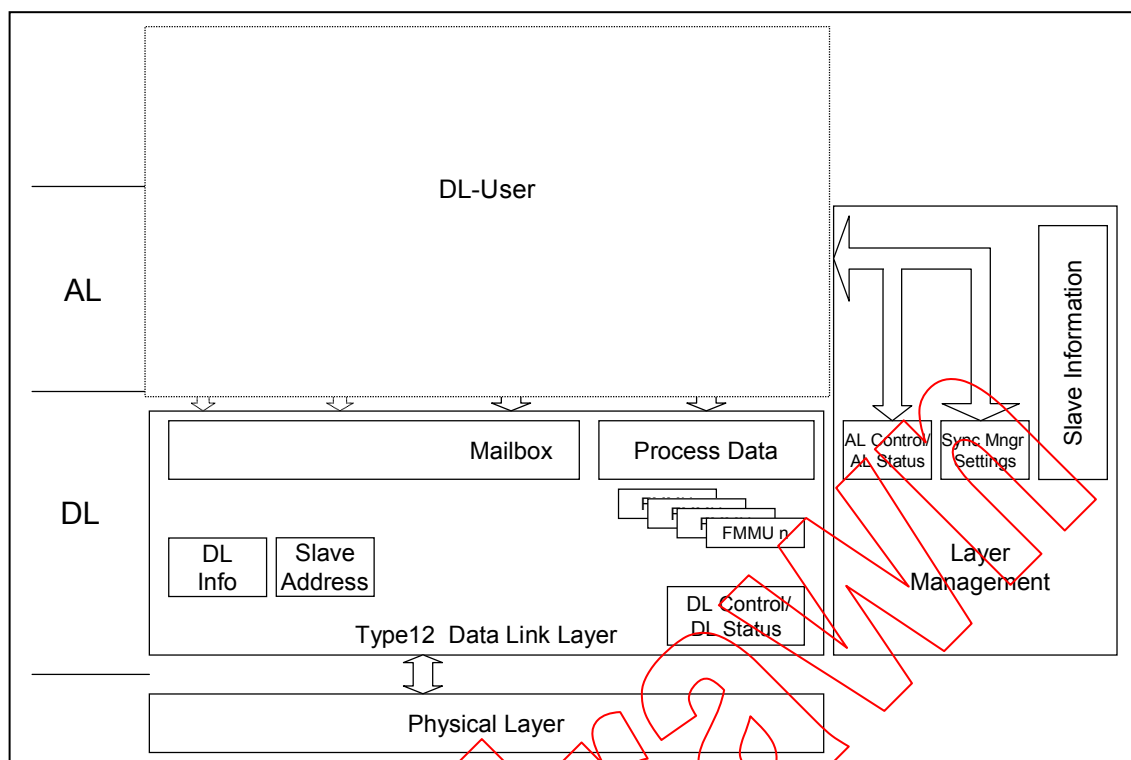


Figure 5 – Slave node reference model

4.6.2 Data-link Layer features

The data link layer provides basic time critical support for data communications among devices connected via Type 12 DL. 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.

4.7 Operation overview

4.7.1 Relation to ISO/IEC 8802-3

This part specifies data link layer services in addition to those specified in ISO/IEC 8802-3.

4.7.2 Frame structure

An Type 12 Ethernet frame contains one or several Type 12 PDUs (as shown in Figure 6), each addressing individual devices and/or memory areas. The Type 12 frame is recognized by the combination of the EtherType 0x88A4¹ and the corresponding Type 12 frame header or, when transported via UDP/IP according to IETF RFC 791/IETF RFC 768 (as shown in

¹ The EtherType 0x88A4 was assigned for Type 12 (EtherCAT) by the IEEE Registration Authority.

Figure 7) by the Destination UDP port 34980=0x88A4² and the Type 12 frame header. Fragmentation of IP packets will be ignored. The UDP checksum may be set to 0 by Slaves and could be ignored. No check on IP type of service, IP header checksum, IP packet length and UDP length is required.

Each Type 12 PDU consists of an Type 12 header, the data area and a subsequent counter area (working counter), which is incremented by all nodes that were addressed by the Type 12 PDU and have exchanged associated data.

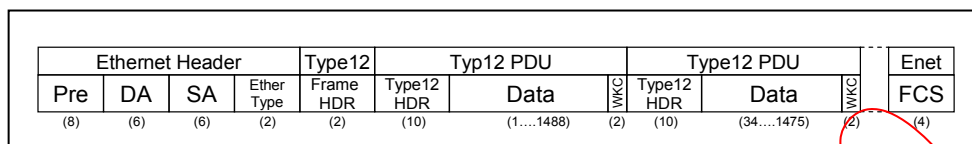


Figure 6 – Type 12 PDUs embedded in Ethernet frame

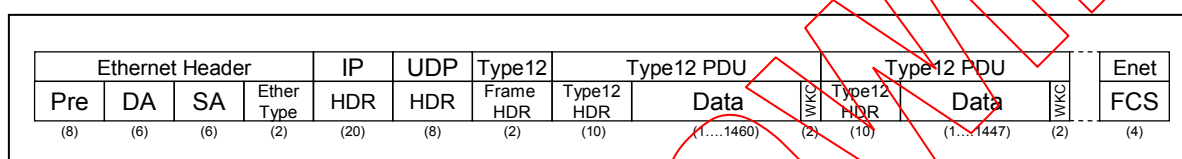


Figure 7 – Type 12 PDUs embedded in UDP/IP

5 Frame structure

5.1 Frame coding principles

Type 12 DL uses a standard ISO/IEC 8802-3 Ethernet frame structure for transporting Type 12 PDUs. The PDUs may alternatively be sent via UDP/IP. The Type 12 specific protocol parts are identical in both cases.

5.2 Data types and encoding rules

5.2.1 General description of data types and encoding rules

To be able to exchange meaningful data, the format of this data and it's meaning have to be known by the producer and consumer(s). This specification models this by the concept of data types.

The encoding rules define the representation of values of data types and the transfer syntax for the representations. Values are represented as bit sequences. Bit sequences are transferred in sequences of octets (bytes). For numerical data types the encoding is little endian style as shown in Table 6.

The data types and encoding rules shall be valid for the DL services and protocols as well as for the AL services and protocols specified. The encoding rules for the Ethernet frame are specified in ISO/IEC 8802-3. The DLSDU of Ethernet is an octet string. The transmission order within octets depends upon MAC and PhL encoding rules.

5.2.2 Transfer syntax for bit sequences

For transmission across Type 12 DL a bit sequence is reordered into a sequence of octets. Hexadecimal notation is used for octets as specified in ISO/IEC 9899. Let $b = b_0...b_{n-1}$ be a

² The UDP Port 34980 was assigned for Type 12 (EtherCAT) by the Internet Assigned Numbers Authority (IANA).

bit sequence. Denote k a non-negative integer such that $8(k - 1) < n \leq 8k$. Then b is transferred in k octets assembled as shown in Table 6. The bits b_i , $i \geq n$ of the highest numbered octet are do not care bits.

Octet 1 is transmitted first and octet k is transmitted last. Hence the bit sequence is transferred as follows across the network (transmission order within an octet is determined by ISO/IEC 8802-3):

$b_7, b_6, \dots, b_0, b_{15}, \dots, b_8, \dots$

Table 6 – Transfer Syntax for bit sequences

Octet number	1.	2.	k.
	$b_7 \dots b_0$	$b_{15} \dots b_8$	$b_{8k-1} \dots b_{8k-8}$

EXAMPLE

Bit 9	...	Bit 0
10b	0001b	1100b
0x2	0x1	0xC
		= 0x21C

The bit sequence $b = b_0 \dots b_9 = 0011\ 1000\ 01$, represents an Unsigned10 with the value 0x21C and is transferred in two octets: First 0x1C and then 0x02.

5.2.3 Unsigned Integer

Data of basic data type Unsigned n has values in the non-negative integers. The value range is $0, \dots, 2^n - 1$. The data is represented as bit sequences of length n . The bit sequence

$b = b_0 \dots b_{n-1}$

is assigned the value

$$\text{Unsigned}_n(b) = b_{n-1} \times 2^{n-1} + \dots + b_1 \times 2^1 + b_0 \times 2^0$$

The bit sequence starts on the left with the least significant byte.

EXAMPLE The value $266 = 0x10A$ with data type Unsigned16 is transferred in two octets, first 0x0A and then 0x01.

The Unsigned n data types are transferred as specified in Table 7. Unsigned data types as Unsigned1 to Unsigned7 and Unsigned 9 to Unsigned15 will be used too. In this case the next element will start at the first free bit position as denoted in 3.7.1.

Table 7 – Transfer syntax for data type Unsignedn

octet number	1.	2.	3.	4.	5.	6.	7.	8.
Unsigned8	b ₇ ..b ₀							
Unsigned16	b ₇ ..b ₀	b ₁₅ ..b ₈						
Unsigned32	b ₇ ..b ₀	b ₁₅ ..b ₈	b ₂₃ ..b ₁₆	b ₃₁ ..b ₂₄				
Unsigned64	b ₇ ..b ₀	b ₁₅ ..b ₈	b ₂₃ ..b ₁₆	b ₃₁ ..b ₂₄	b ₃₉ ..b ₃₂	b ₄₇ ..b ₄₀	b ₅₅ ..b ₄₈	b ₆₃ ..b ₅₆

5.2.4 Signed Integer

Data of basic data type Integer n has values in the integers. The value range is from -2^{n-1} to $2^{n-1}-1$. The data is represented as bit sequences of length n . The bit sequence

$$b = b_0 \dots b_{n-1}$$

is assigned the value

$$\text{Integer}_n(b) = b_{n-2} \times 2^{n-2} + \dots + b_1 \times 2^1 + b_0 \times 2^0 \text{ if } b_{n-1} = 0$$

and, performing two's complement arithmetic,

$$\text{Integer}_n(b) = -\text{Integer}_n(\text{^}b) - 1 \text{ if } b_{n-1} = 1$$

Note that the bit sequence starts on the left with the least significant bit.

Example: The value $-266 = 0x\text{FEP6}$ with data type Integer16 is transferred in two octets, first 0xF6 and then 0xFE.

The Integer n data types are transferred as specified in Table 8. Integer data types as Integer1 to Integer7 and Integer9 to Integer15 will be used too. In this case the next element will start at the first free bit position as denoted in 3.7.1.

Table 8 – Transfer syntax for data type Integer n

Octet number	1.	2.	3.	4.	5.	6.	7.	8.
Integer8	b ₇ ..b ₀							
Integer16	b ₇ ..b ₀	b ₁₅ ..b ₈						
Integer32	b ₇ ..b ₀	b ₁₅ ..b ₈	b ₂₃ ..b ₁₆	b ₃₁ ..b ₂₄				
Integer64	b ₇ ..b ₀	b ₁₅ ..b ₈	b ₂₃ ..b ₁₆	b ₃₁ ..b ₂₄	b ₃₉ ..b ₃₂	b ₄₇ ..b ₄₀	b ₅₅ ..b ₄₈	b ₆₃ ..b ₅₆

5.2.5 Octet String

The data type OctetString $length$ is defined below; $length$ is the length of the octet string.

ARRAY [length] OF Unsigned8 OctetString $length$

5.2.6 Visible String

The data type VisibleString $length$ is defined below. The admissible values of data of type `VISIBLE_CHAR` are 0_h and the range from 0x20 to 0x7E. The data are interpreted as 7-bit coded characters. $length$ is the length of the visible string.

Unsigned8 `VISIBLE_CHAR`

ARRAY [length] OF VISIBLE_CHAR VisibleString $length$

There is no 0x0 necessary to terminate the string.

5.3 DLPDU structure

5.3.1 Type 12 frame inside an Ethernet frame

The frame structure in consists of the following data entries as specified in Table 9.

Table 9 – Type 12 frame inside an Ethernet frame

Frame part	Data field	Data type	Value/description
Ethernet	Dest MAC	BYTE[6]	Destination MAC Address as specified in ISO/IEC 8802-3
	Src MAC	BYTE[6]	Source MAC Address as specified in ISO/IEC 8802-3
(optional)	VLAN Tag	BYTE[4]	0x81, 0x00 and two bytes Tag Control Information as specified in IEEE 802.1Q
	Ether Type	BYTE[2]	0x86, 0xA4 (Type 12)
	Type 12 frame		specified in 5.3.3
	Padding	BYTE[n]	shall be inserted if DL PDU is shorter than 64 octets as specified in ISO/IEC 8802-3
Ethernet FCS	FCS	Unsigned32	Standard Ethernet Checksum coding as specified in ISO/IEC 8802-3

Withhold

5.3.2 Type 12 frame inside a UDP datagram

The frame structure in consists of the following data entries as specified in Table 10.

Table 10 – Type 12 frame inside an UDP PDU

Frame part	Data field	Data type	Value/description
Ethernet	Dest MAC	BYTE[6]	See Table 9
	Src MAC	BYTE[6]	See Table 9
(optional)	VLAN Tag	BYTE[4]	See Table 9
	Ether Type	BYTE[2]	0x08, 0x00 (IP)
IP	VersionHL	BYTE	0x45 (IP Version(4) header length (5*4 octets))
	Service	BYTE	0x00 (IP Type of service)
	TotalLength	Unsigned16	(IP total length of service) - not checked within Type 12 segment
	Identification	Unsigned16	(IP identification packet for fragmented service) - not checked within Type 12 segment
	Flags	BYTE	(IP flags – they will not be considered but a fragmentation of Type 12 frame will result in an error) - not checked within Type 12 segment
	Fragments	BYTE	(IP fragment number - fragmentation of Type 12 frame will result in an error) - not checked within Type 12 segment
	Ttl	BYTE	(IP time to live – only checked at routers) - not checked within Type 12 segment
	Protocol	BYTE	0x11 (IP sub-protocol – this value is reserved for UDP)
	Header checksum	Unsigned16	(IP header checksum) - not checked within Type 12 segment
	Source IP address	BYTE[4]	(IP source address of the originator) - not checked within Type 12 segment
	Destination IP address	BYTE[4]	(IP destination address of the target of the frame – within a Type 12 segment usually a multicast address as a individual address requires the Address Resolution Protocol ARP) - not checked within Type 12 segment
UDP	Src port	WORD	(UDP Source Port) - not checked within Type 12 segment
	Dest port	WORD	0x88A4 (UDP Source Port)
	Length	WORD	(UDP length of frame)) - not checked within Type 12 segment
	Checksum	WORD	(UDP checksum of frame) – will be set to 0 for Type 12 frames but without checking
	Type 12 frame		specified in 5.3.3
	Padding	BYTE[n]	shall be inserted if DL PDU is shorter than 64 octets as specified in ISO/IEC 8802-3
Ethernet FCS	FCS	Unsigned32	Standard Ethernet Checksum coding as specified in ISO/IEC 8802-3

NOTE 1 IP packet structure and coding requirements are as specified in IETF RFC 791.

NOTE 2 The ordering of octets in multi-octet values is encoded differently in IETF protocols (see IETF RFC 768 and RFC 791) than it is within the Type 12 DL-protocol.

5.3.3 Type 12 frame structure

The Type 12 frame structure in shall consist one of the structures specified in Table 11, Table 12 and Table 13.

Table 11 – Type 12 frame structure containing Type 12 PDUs

Frame part	Data field	Data type	Value/description
Type 12 Frame	Length	unsigned11	Length of this frame (minus 2 octets)
	Reserved	unsigned 1	0
	Type	unsigned4	Protocol Type = Type 12 DLPDUs (0x01)
	Type 12 PDU 1		specified in 5.4
	...		specified in 5.4
	Type 12 PDU n		specified in 5.4

Table 12 – Type 12 frame structure containing network variables

Frame part	Data field	Data type	Value/description
Type 12 frame	Length	unsigned11	Length of this frame (minus 2 octets)
	reserved	unsigned 1	0
	Type	unsigned4	Protocol type = network variables (0x04)
Publisher header	PubID	BYTE[6]	Publisher ID
	CntNV	Unsigned16	Number of Network variables contained in this Type 12 frame
	CYC	Unsigned16	Cycle Number of the publisher side
	reserved	BYTE[2]	0x00, 0x00
	Network variable 1		Specified in 5.5
	...		Specified in 5.5
	Network variable n		Specified in 5.5

Table 13 – Type 12 frame structure containing mailbox

Frame part	Data field	Data type	Value/description
Type 12 frame	Length	unsigned11	Length of this frame (minus 2 octets)
	reserved	unsigned 1	0
	Type	unsigned4	Protocol type = mailbox (0x05)
	Mailbox		Specified in 5.6

5.4 Type 12 DLPDU structure

5.4.1 Read

5.4.1.1 Overview

With the read services a master reads data to memory of one or many slaves. The working counter shall be incremented by each slave if at least one of the addressed attribute is present.

5.4.1.2 Auto increment physical read (APRD)

The auto increment physical read (APRD) coding is specified in Table 14. Each slave increments the address ADP. The slave that receives an auto-increment address with value zero executes the requested read operation.

Table 14 – Auto increment physical read (APRD)

Frame part	Data field	Data type	Value/description
Type 12 PDU	CMD	Unsigned8	0x01 (command APRD)
	IDX	Unsigned8	Index
	ADP	WORD	Auto increment address
	ADO	WORD	Physical memory or register address
	LEN	Unsigned11	Length of the DATA data field
	reserved	Unsigned4	0x00
	NEXT	Unsigned1	0x00: last Type 12 PDU in Type 12 frame 0x01: Type 12 PDU in Type 12 frame follows
	IRQ	WORD	External event
	DATA	OctetString LEN	Data, structure as specified in 5.6, 6 or by DLS-user
	WKC	WORD	Working counter

CMD

The parameter Command shall contain the service command.

IDX

The parameter Index is the local identifier in the master of the service and shall not be changed by the slave.

ADP

Each slave shall increment this parameter and the slave that receives this parameter with a value of zero shall perform the read access.

NOTE That means, the parameter contains the negative position of the slave in the logical loop beginning with 0 at the master side (e.g. -7 means that seven slaves are between the master and the addressed slave). At the confirmation this parameter contains the value of the request incremented by the number of transited slave devices.

ADO

This parameter shall contain the start address in the physical memory of the slave where the data to be read is stored

LEN

This parameter shall contain the size in octets of the data to be read.

NEXT

This parameter shall specify if there is another Type 12 PDU in the frame.

IRQ

This parameter shall contain the External event (see Table 38) masked by the External event mask (see Table 39)

DATA

This parameter shall contain the read data if the access is valid at the addressed slaves site. Otherwise the value send out with the request remains unchanged.

WKC

This parameter shall be incremented by one if the data was successfully read

5.4.1.3 Configured address physical read (FPRD)

The configured address physical read (FPRD) coding is specified in Table 15.

Table 15 – Configured address physical read (FPRD)

Frame part	Data field	Data type	Value/description
Type 12 PDU	CMD	Unsigned8	0x04 (command FPRD)
	IDX	Unsigned8	Index
	ADP	WORD	Configured station address or configured station alias
	ADO	WORD	Physical memory or register address
	LEN	Unsigned11	Length of the DATA data field
	reserved	Unsigned4	0x00
	NEXT	Unsigned1	0x00: last Type 12 PDU in Type 12 frame 0x01: Type 12 PDU in Type 12 frame follows
	IRQ	WORD	External event
	DATA	OctetString LEN	Data, structure as specified in 5.6, 6 or by DLS-user
	WKC	WORD	Working counter

CMD

The parameter Command shall contain the service command.

IDX

The parameter Index is the local identifier in the master of the service; it shall not be changed by the slave.

ADP

The slave which has the value of D_address as station address or station address alias shall execute a read action.

ADO

This parameter shall contain the start address in the physical memory of the slave where the data to be read is stored.

LEN

This parameter shall contain the size in octets of the data to be read.

NEXT

This parameter shall specify if there is another Type 12 PDU in the frame.

IRQ

This parameter shall contain the External event (see Table 38) masked by the External event mask (see Table 39)

DATA

This parameter shall contain the read data if the access is valid at the addressed slaves site. Otherwise the value send out with the request remains unchanged.

WKC

This parameter shall be incremented by one if the data was successfully read.

5.4.1.4 Broadcast read (BRD)

The broadcast read (BRD) coding is specified in Table 16.

Table 16 – Broadcast read (BRD)

Frame part	Data field	Data type	Value/description
Type 12 PDU	CMD	Unsigned8	0x07 (command BRD)
	IDX	Unsigned8	Index
	ADP	WORD	Parameter incremented by 1 at each station forwarding BRD PDU
	ADO	WORD	Physical memory or register address
	LEN	Unsigned11	Length of the DATA data field
	reserved	Unsigned4	0x00
	NEXT	Unsigned1	0x00: last Type 12 PDU in Type 12 frame 0x01: Type 12 PDU in Type 12 frame follows
	IRQ	WORD	External event
	DATA	OctetString LEN	Data, structure as specified in 5.6, 6 or by DLS-user
	WKC	WORD	Working counter

CMD

The parameter Command shall contain the service command.

IDX

The parameter Index is the local identifier in the master of the service; it shall not be changed by the slave.

ADP

This parameter shall be incremented by one at each slave.

ADO

This parameter shall contain the start address in the physical memory where the data to be read is stored. Each slave who supports the requested physical memory area (physical memory address and length) shall respond to this service.

LEN

This parameter shall contain the size in octets of the data to be read.

NEXT

This parameter shall specify if there is another Type 12 PDU in the frame.

IRQ

This parameter shall contain the External event (see Table 38) masked by the External event mask (see Table 39)

DATA

This parameter shall contain the read data collected before entry or default values of the master. This parameter shall contain the result of the bitwise-OR operation between the parameter data of the request and the addressed data in the slave.

WKC

This parameter shall be incremented by one by all slaves which made the bitwise-OR of the requested data.

5.4.1.5 Logical read (LRD)

The logical read (LRD) coding is specified in Table 17. The slave copies only data to the parameter data that are mapped by an FMMU entity from the logical address space to a physical address.

Table 17 – Logical read (LRD)

Frame part	Data field	Data type	Value/description
Type 12 PDU	CMD	Unsigned8	0x0A (command LRD)
	IDX	Unsigned8	Index
	ADR	DWORD	Logical address
	LEN	Unsigned11	Length of the DATA data field
	reserved	Unsigned4	0x00
	NEXT	Unsigned1	0x00: last Type 12 PDU in Type 12 frame 0x01: Type 12 PDU in Type 12 frame follows
	IRQ	WORD	reserved for future use
	DATA	OctetString LEN	Data, structure as specified by DLS-user
	WKC	WORD	Working counter

CMD

The parameter Command shall contain the service command.

IDX

The parameter Index is the local identifier in the master of the service; it shall not be changed by the slave.

ADR

This parameter shall contain the start address in the logical memory where the data to be read is located. All slaves which have one or more address matches of the requested logical memory area (logical memory address and length) in their FMMU entities shall map the requested data to the data parameter as described by the FMMU entity settings and increment the working counter.

LEN

This parameter shall contain the size in octets of the data to be read.

NEXT

This parameter shall specify if there is another Type 12 PDU in the frame.

IRQ

This parameter shall contain the External event (see Table 38) masked by the External event mask (see Table 39)

DATA

On confirm this parameter specifies the data read from the device. Each slave which detects an address match of the requested logical memory area puts the data of the corresponding physical memory area in the correct part of this parameter.

WKC

This parameter shall be incremented by one by all slaves which detect an address match of the requested logical memory area.

5.4.2 Write

5.4.2.1 Overview

With the write services a master writes data to register or memory of one or many slaves. The working counter is incremented if the addressed attribute is present. The working counter can be incremented by one if at least one part of the data can be written.

5.4.2.2 Auto increment physical write (APWR)

The auto increment physical write (APWR) coding is specified in Table 18. Each slave increments the address. The slave that receives a zero value at auto-increment address parameter will execute the requested write operation.

Table 18 – Auto Increment physical write (APWR)

Frame part	Data field	Data type	Value/description
Type 12 PDU	CMD	Unsigned8	0x02 (command APWR)
	IDX	Unsigned8	Index
	ADP	WORD	Auto increment address
	ADO	WORD	Physical memory or register address
	LEN	Unsigned11	Length of the DATA data field
	reserved	Unsigned4	0x00
	NEXT	Unsigned1	0x00: last Type 12 PDU in Type 12 frame 0x01: Type 12 PDU in Type 12 frame follows
	IRQ	WORD	External event
	DATA	OctetString LEN	Data, structure as specified in 5.6, 6 or by DLS-user
	WKC	WORD	Working counter

CMD

The parameter Command shall contain the service command.

IDX

The parameter Index is the local identifier in the master of the service; it shall not be changed by the slave.

ADP

The slave will be addressed by its position in the segment. Each slave shall increment this parameter, the slave who receives the value zero of this parameter shall respond to this service.

NOTE That means, the parameter contains the negative position of the slave in the logical ring beginning with 0 at the master side (e.g. -7 means 7 slaves are between master and the addressed slave). At the confirmation this parameter contains the value of the request incremented by the number of transited slave devices.

ADO

This parameter shall contain the start address in the physical memory of the slave where the data to be written is stored.

LEN

This parameter shall contain the size in octets of the data to be written.

NEXT

This parameter shall specify if there is another Type 12 PDU in the frame.

IRQ

This parameter shall contain the External event (see Table 38) masked by the External event mask (see Table 39)

DATA

This parameter shall contain the data to be written.

WKC

This parameter shall be incremented by one if the data can be successfully written.

5.4.2.3 Configured address physical write (FPWR)

The configured address physical write (FPWR) coding is specified in Table 19.

Table 19 – Configured address physical write (FPWR)

Frame part	Data field	Data type	Value/description
Type 12 PDU	CMD	Unsigned8	0x05 (command NPWR)
	IDX	Unsigned8	Index
	ADP	WORD	Configured station address or configured station alias
	ADO	WORD	Physical memory or register address
	LEN	Unsigned11	Length of the DATA data field
	reserved	Unsigned4	0x00
	NEXT	Unsigned1	0x00: last Type 12 PDU in Type 12 frame 0x01: Type 12 PDU in Type 12 frame follows
	IRQ	WORD	External event
	DATA LEN	OctetString LEN	Data, structure as specified in 5.6, 6 or by DLS-user
	WKC	WORD	Working counter

CMD

The parameter Command shall contain the service command.

IDX

The parameter Index is the local identifier in the master of the service; it shall not be changed by the slave.

ADP

The slave which has the value of D_address as station address or station address alias shall execute a write action.

ADO

This parameter shall contain the start address in the physical memory of the slave where the data to be written is stored.

LEN

This parameter shall contain the size in octets of the data to be written.

NEXT

This parameter shall specify if there is another Type 12 PDU in the frame.

IRQ

This parameter shall contain the External event (see Table 38) masked by the External event mask (see Table 39)

DATA

This parameter shall contain the data to be written.

WKC

This parameter shall be incremented by one if the data was successfully written.

5.4.2.4 Broadcast write (BWR)

The broadcast write (BWR) coding is specified in Table 20.

Table 20 – Broadcast write (BWR)

Frame part	Data field	Data type	Value/description
Type 12 PDU	CMD	Unsigned8	0x08 (command BWR)
	IDX	Unsigned8	Index
	ADP	WORD	Parameter incremented by 1 at each station forwarding BWR PDU
	ADO	WORD	Physical memory or register address
	LEN	Unsigned11	Length of the DATA data field
	reserved	Unsigned4	0x00
	NEXT	Unsigned1	0x00: last Type 12 PDU in Type 12 frame 0x01: Type 12 PDU in Type 12 frame follows
	IRQ	WORD	External event
	DATA	OctetString LEN	Data, structure as specified in 5.6, 6 or by DLS-user
	WKC	WORD	Working counter

CMD

The parameter Command shall contain the service command.

IDX

The parameter Index is the local identifier in the master of the service; it shall not be changed by the slave.

ADP

This parameter shall be incremented by one at each slave.

ADO

This parameter shall contain the start address in the physical memory where the data to be written is stored. Each slave who supports the requested physical memory area (physical memory address and length) shall respond to this service.

LEN

This parameter shall contain the size in octets of the data to be written.

NEXT

This parameter shall specify if there is another Type 12 PDU in the frame.

IRQ

This parameter shall contain the External event (see Table 38) masked by the External event mask (see Table 39)

DATA

This parameter shall contain the data to be written.

WKC

This parameter shall be incremented by one by all slaves which write data in their physical memory.

5.4.2.5 Logical write (LWR)

The logical write (LWR) coding is specified in Table 21. The slave copies only data to the memory or register that are mapped by an FMMU entity from the logical address space to a physical address.

Table 21 – Logical write (LWR)

Frame part	Data field	Data type	Value/description
Type 12 PDU	CMD	Unsigned8	0x0B (command LWR)
	IDX	Unsigned8	Index
	ADR	DWORD	Logical address
	LEN	Unsigned11	Length of the DATA data field
	reserved	Unsigned4	0x00
	NEXT	Unsigned1	0x00: last Type 12 PDU in Type 12 frame 0x01: Type 12 PDU in Type 12 frame follows
	IRQ	WORD	reserved for future use
	DATA	OctetString LEN	Data, structure as specified by DLS-user
	WKC	WORD	Working counter

CMD

The parameter Command shall contain the service command.

IDX

The parameter Index is the local identifier in the master of the service; it shall not be changed by the slave.

ADR

This parameter shall contain the start address in the logical memory where the data to be written is located. All slaves which have one or more address matches of the requested logical memory area (logical memory address and length) in their FMMUs shall respond to this service.

LEN

This parameter shall contain the size in octets of the data to be written.

NEXT

This parameter shall specify if there is another Type 12 PDU in the frame.

IRQ

This parameter shall contain the External event (see Table 38) masked by the External event mask (see Table 39)

DATA

This parameter shall contain the data to be written. Each slave which detects an address match of the requested logical memory area will put the data of the correct part of this parameter in the corresponding physical memory area.

WKC

This parameter shall be incremented by one by all slaves who detect an address match of the requested logical memory area and if the data was successfully written

5.4.3 Read write

5.4.3.1 Auto increment physical read write (APRW)

The auto increment physical read write (APRW) coding is specified in Table 22. Each slave increments the address. The slave that receives a zero value at auto-increment address parameter will execute the requested operation.

Table 22 – Auto increment physical read write (APRW)

Frame part	Data field	Data type	Value/description
Type 12 PDU	CMD	Unsigned8	0x03 (command APRW)
	IDX	Unsigned8	Index
	ADP	WORD	Auto increment address
	ADO	WORD	Physical memory or register address
	LEN	Unsigned11	Length of the DATA data field
	reserved	Unsigned4	0x00
	NEXT	Unsigned1	0x00: last Type 12 PDU in Type 12 frame 0x01: Type 12 PDU in Type 12 frame follows
	IRQ	WORD	External event
	DATA	OctetString LEN	Data, structure as specified in 5.6, 6 or by DLS-user
	WKC	WORD	Working counter

CMD

The parameter Command shall contain the service command.

IDX

The parameter Index is the local identifier in the master of the service; it shall not be changed by the slave.

ADP

The slave will be addressed by its position in the segment. Each slave shall increment this parameter, the slave who receives the value zero of this parameter shall respond to this service.

NOTE That means, the parameter contains the negative position of the slave in the logical ring beginning with 0 at the master side (e.g. -7 means 7 slaves are between master and the addressed slave). At the confirmation this parameter contains the value of the request incremented by the number of transited slave devices.

ADO

This parameter shall contain the start address in the physical memory of the slave where data to be read and written is stored.

LEN

This parameter shall contain the size in octets of the data to be written and read.

NEXT

This parameter shall specify if there is another Type 12 PDU in the frame.

IRQ

This parameter shall contain the External event (see Table 38) masked by the External event mask (see Table 39)

DATA

This parameter shall contain the data to be written and the data read from the addressed slave if the service can be executed successfully.

WKC

This parameter shall be incremented by two if the data was successfully written and additionally by one if the data was successfully read.

5.4.3.2 Configured address physical read write (FPRW)

The configured address physical read write (FPRW) coding is specified in Table 23.

Table 23 – Configured address physical read write (FPRW)

Frame part	Data field	Data type	Value/description
Type 12 PDU	CMD	Unsigned8	0x06 (command NPRW)
	IDX	Unsigned8	Index
	ADP	WORD	Configured station address or configured station alias
	ADO	WORD	Physical memory or register address
	LEN	Unsigned11	Length of the DATA data field
	reserved	Unsigned4	0x00
	NEXT	Unsigned1	0x00: last Type 12 PDU in Type 12 frame 0x01: Type 12 PDU in Type 12 frame follows
	IRQ	WORD	External event
	DATA	OctetString LEN	Data, structure as specified in 5.6, 6 or by DLS-user
	WKC	WORD	Working counter

CMD

The parameter Command shall contain the service command.

IDX

The parameter Index is the local identifier in the master of the service; it shall not be changed by the slave.

ADP

The slave which has the value of D_address as station address or station address alias shall execute a read action followed by a write action.

ADO

This parameter shall contain the start address in the physical memory of the slave where the data to be read and written is stored.

LEN

This parameter shall contain the size in octets of the data to be written and read.

NEXT

This parameter shall specify if there is another Type 12 PDU in the frame.

IRQ

This parameter shall contain the External event (see Table 38) masked by the External event mask (see Table 39)

DATA

This parameter shall contain the data to be written and the data read from the addressed slave if the service can be executed successfully.

WKC

This parameter shall be incremented by two if the data was successfully written and additionally by one if the data was successfully read.

5.4.3.3 Broadcast read write (BRW)

The broadcast read write (BRW) coding is specified in Table 24.

Table 24 – Broadcast read write (BRW)

Frame part	Data field	Data type	Value/description
Type 12 PDU	CMD	Unsigned8	0x09 (command BRW)
	IDX	Unsigned8	Index
	ADP	WORD	Parameter incremented by 1 at each station forwarding BRW PDU
	ADO	WORD	Physical memory or register address
	LEN	Unsigned11	Length of the DATA data field
	reserved	Unsigned4	0x00
	NEXT	Unsigned1	0x00: last Type 12 PDU in Type 12 frame 0x01: Type 12 PDU in Type 12 frame follows
	IRQ	WORD	External event
	DATA	OctetString LEN	Data, structure as specified in 5.6, 6 or by DLS-user
	WKC	WORD	Working Counter

CMD

The parameter Command shall contain the service command.

IDX

The parameter Index is the local identifier in the master of the service; it shall not be changed by the slave.

ADP

This parameter shall be incremented by one at each slave.

ADO

This parameter shall contain the start address in the physical memory where the data to be read and written is stored. Each slave who supports the requested physical memory area (physical memory address and length) shall respond to this service.

LEN

This parameter shall contain the size in octets of the data to be written and read.

NEXT

This parameter shall specify if there is another Type 12 PDU in the frame.

IRQ

This parameter shall contain the External event (see Table 38) masked by the External event mask (see Table 39)

DATA

This parameter shall contain the data before entry and will be the written. A read operation is performed before write. This parameter shall contain the result of the bitwise-OR operation between the parameter data of the request and the addressed data in the slave.

WKC

This parameter shall be incremented by two by all slaves if the data was successfully written and additionally by one by all slaves which made the bitwise-OR of the requested data.

5.4.3.4 Logical read write (LRW)

The logical read write (LRW) coding is specified in Table 25. A slave device can retrieve data with this service (write operation) and put data with this service (read operation). The slave will copy in or out only data to or from the parameter data that are mapped by an FMMU entity from the logical address space to a physical address.

Table 25 – Logical read write (LRW)

Frame part	Data field	Data type	Value/description
Type 12 PDU	CMD	Unsigned8	0x0C (command LRW)
	IDX	Unsigned8	Index
	ADR	DWORD	Logical address
	LEN	Unsigned11	Length of the DATA data field
	reserved	Unsigned4	0x00
	NEXT	Unsigned1	0x00: last Type 12 PDU in Type 12 frame 0x01: Type 12 PDU in Type 12 frame follows
	IRQ	WORD	Reserved for future use
	DATA	OctetString LEN	Data, structure as specified by DLS-user
	WKC	WORD	Working counter

CMD

The parameter Command shall contain the service command.

IDX

The parameter Index is the local identifier in the master of the service; it shall not be changed by the slave.

ADR

This parameter shall contain the start address in the logical memory where the data to be read or written is located. All slaves which have one or more address matches of the requested logical memory area (logical memory address and length) in their FMMU shall respond to this service.

LEN

This parameter shall contain the size in octets of the data to be written and read.

NEXT

This parameter shall specify if there is another Type 12 PDU in the frame.

IRQ

This parameter shall contain the External event (see Table 38) masked by the External event mask (see Table 39)

DATA

This parameter shall contain the data to be written. Each slave who detects an address match of the requested logical memory area will put the data of the correct part of this parameter in the corresponding physical memory area. With the confirmation this parameter shall contain the read data. Each slave who detects an address match of

the addressed logical memory area will put the data of the corresponding physical memory area in the correct part of this parameter.

WKC

This parameter shall be incremented by each slave by two if a piece of data was successfully written and additional incremented by one if a piece of data was successfully read.

5.4.3.5 Auto increment physical read multiple write (ARMW)

The auto increment physical read multiple write (ARMW) coding is specified in Table 26.

Table 26 – Auto increment physical read multiple write (ARMW)

Frame part	Data field	Data type	Value/description
Type 12 PDU	CMD	Unsigned8	0x0D (command ARMW)
	IDX	Unsigned8	Index
	ADP	WORD	Auto increment or register address
	ADO	WORD	Physical memory or register address
	LEN	Unsigned11	Length of the DATA data field
	reserved	Unsigned4	0x00
	NEXT	Unsigned1	0x00: last Type 12 PDU in Type 12 frame 0x01: Type 12 PDU in Type 12 frame follows
	IRQ	WORD	External event
	DATA	OctetString LEN	Data, structure as specified in 5.6, 6 or by DLS-user
	WKC	WORD	Working counter

CMD

The parameter Command shall contain the service command.

IDX

The parameter Index is the local identifier in the master of the service; it shall not be changed by the slave.

ADP

The slave will be addressed by its position in the segment. Each slave shall increment this parameter, the slave who receives the value zero of this parameter shall execute a read action – the other slaves shall execute a write action.

NOTE That means, the parameter contains the negative position of the slave in the logical ring beginning with 0 at the master side (e.g. -7 means 7 slaves are between master and the addressed slave). At the confirmation this parameter contains the value of the request incremented by the number of transited slave devices.

ADO

This parameter shall contain the start address in the physical memory of the slave where data to be read and written is stored.

LEN

This parameter shall contain the size in octets of the data to be written and read.

NEXT

This parameter shall specify if there is another Type 12 PDU in the frame.

IRQ

This parameter shall contain the External event (see Table 38) masked by the External event mask (see Table 39)

DATA

This parameter shall contain the data to be written and the data read from the addressed slave if the service can be executed successfully.

WKC

This parameter shall be incremented by one by each slave if the data was successfully read or written.

5.4.3.6 Configured address physical read multiple write (FRMW)

The configured address physical read multiple write (FRMW) coding is specified in Table 27.

Table 27 – Configured address physical read multiple write (FRMW)

Frame part	Data field	Data type	Value/description
Type 12 PDU	CMD	Unsigned8	0x0E (command FRMW)
	IDX	Unsigned8	Index
	ADP	WORD	Configured station address or configured station alias
	ADO	WORD	Physical memory address
	LEN	Unsigned11	Length of the DATA data field
	reserved	Unsigned4	0x00
	NEXT	Unsigned1	0x00: last Type 12 PDU in Type 12 frame 0x01: Type 12 PDU in Type 12 frame follows
	IRQ	WORD	External event
	DATA	OctetString LEN	Data structure as specified in 5.6, 6 or by DLS-user
	WKC	WORD	Working counter

CMD

The parameter Command shall contain the service command.

IDX

The parameter Index is the local identifier in the master of the service; it shall not be changed by the slave.

ADP

The slave which has the value of D_address as station address or station address alias shall execute a read action - the other slaves shall execute a write action.

ADO

This parameter shall contain the start address in the physical memory of the slave where data to be read and written is stored.

LEN

This parameter shall contain the size in octets of the data to be written and read.

NEXT

This parameter shall specify if there is another Type 12 PDU in the frame.

IRQ

This parameter shall contain the External event (see Table 38) masked by the External event mask (see Table 39)

DATA

This parameter shall contain the data to be written and the data read from the addressed slave if the service can be executed successfully.

WKC

This parameter shall be incremented by one by all slaves if the data was successfully read or written.

5.5 Network variable structure

The network variable coding is specified in Table 28.

Table 28 – Network variable

Frame part	Data field	Data type	Value/description
Network variable	Index	Unsigned16	Index to a DLS-user object
	HASH	Unsigned16	Hash algorithm over the data structure of the data to detect changes
	LEN	Unsigned16	Length
	Q	Unsigned16	Quality
	DATA	OctetString [LEN]	Data, structure as specified by DLS-user

5.6 Type 12 mailbox structure

The mailbox coding is specified in Table 29. The mailbox encoding shall be used in conjunction with Type 12 mailbox memory elements or as coding for data structures conveying mailboxes via Ethernet DL or via IP.

Table 29 – Mailbox

Frame part	Data field	Data type	Value/description
Mailbox	Length	Unsigned16	Length of the Mailbox Service Data
	Address	WORD	Station Address of the source, if a master is client, Station Address of the destination, if a slave is client or data are transmitted outside the target Type 12 segment
	Channel	Unsigned6	0x00 (Reserved for future)
	Priority	Unsigned2	0x00: lowest priority ... 0x03: highest priority
	Type	Unsigned4	0x00: error(ERR) 0x01: vendor specific 0x02: Ethernet over Type 12 (EoE) 0x03: CANopen over Type 12 (CoE) 0x04: File Access over Type 12 (FoE) 0x05: Servo profile over Type 12 (SoE) 0x06 -0x0e: reserved 0x0f: vendor specific
	Cnt	Unsigned3	Counter of the mailbox services (0 ist start value, next value after 7 is 1.
	reserved	Unsigned1	0x00

	Service Data	OctetString [Length]	Mailbox Service Data
--	--------------	----------------------	----------------------

The encoding of Service Data in case of an error reply is specified in Table 30.

Table 30 – Error Reply Service Data

Frame part	Data field	Data type	Value/description
Service Data	Type	Unsigned16	0x01: Mailbox Command
	Detail	Unsigned16	0x01: MBXERR_SYNTAX 0x02: MBXERR_UNSUPPORTEDPROTOCOL 0x03: MBXERR_INVALIDCHANNEL 0x04: MBXERR_SERVICENOTSUPPORTED 0x05: MBXERR_INVALIDHEADER 0x06: MBXERR_SIZETOOSHORT 0x07: MBXERR_NOMOREMEMORY 0x08: MBXERR_INVALIDSIZE

6 Attributes

6.1 Management

6.1.1 DL Information

The DL information registers contain type, version and supported resources of the slave controller (ESC).

Parameter

Type

This parameter shall contain the type of the slave controller.

Revision (major revision)

This parameter shall contain the revision of the slave controller.

Build (minor revision)

This parameter shall contain the build number of the slave controller.

Number of supported FMMU entities

This parameter shall contain the number of supported FMMU entities of the slave controller.

Number of supported sync manager channels

This parameter shall contain the number of supported sync manager channels (or entities) of the slave controller.

RAM size

This parameter shall contain the RAM size in Kbyte supported by the slave controller (smaller size than a even number will be rounded up).

Port descriptor

Port 0 Physical Layer

This parameter should indicate the physical layer used for this port.