

INTERNATIONAL STANDARD

NORME INTERNATIONALE



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

**Réseaux de communication industriels – Spécifications des bus de terrain –
Partie 3-12: Définition des services de la couche liaison de données – Éléments
de type 12**



THIS PUBLICATION IS COPYRIGHT PROTECTED
Copyright © 2014 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.



IEC 61158-3-12

Edition 3.0 2014-08

INTERNATIONAL STANDARD

NORME INTERNATIONALE



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

**Réseaux de communication industriels – Spécifications des bus de terrain –
Partie 3-12: Définition des services de la couche liaison de données – Éléments
de type 12**

INTERNATIONAL
ELECTROTECHNICAL
COMMISSION

COMMISSION
ELECTROTECHNIQUE
INTERNATIONALE

PRICE CODE
CODE PRIX



ICS 25.040.40; 35.100.20; 35.110

ISBN 978-2-8322-1713-9

**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.....	4
INTRODUCTION.....	6
1 Scope.....	7
1.1 General.....	7
1.2 Specifications.....	7
1.3 Conformance.....	7
2 Normative references.....	8
3 Terms, definitions, symbols, abbreviations and conventions.....	8
3.1 Reference model terms and definitions.....	8
3.2 Service convention terms and definitions.....	9
3.3 Data-link service terms and definitions.....	10
3.4 Symbols and abbreviations.....	13
3.5 Common conventions.....	14
4 Data-link layer services and concepts.....	15
4.1 Operating principle.....	15
4.2 Topology.....	16
4.3 Data-link layer overview.....	16
4.4 Error detection overview.....	17
4.5 Parameter and process data handling introduction.....	17
4.6 Node reference model.....	18
4.7 Operation overview.....	19
4.8 Addressing.....	20
4.9 Slave classification.....	22
4.10 Structure of the communication layer in the slave.....	23
5 Communication services.....	24
5.1 Overview.....	24
5.2 Read services.....	24
5.3 Write services.....	27
5.4 Combined read/write services.....	29
5.5 Network services.....	33
5.6 Mailbox.....	34
6 Local interactions.....	38
6.1 Read local.....	38
6.2 Write local.....	39
6.3 Event local.....	40

Figure 1 – Mapping of logical data in an Ethernet frame consisting of a single Type 12 DLPDU	17
Figure 2 – Type 12 data-link reference model	18
Figure 3 – Type 12 segments in open mode.....	19
Figure 4 – Type 12 segment in direct mode	19
Figure 5 – Addressing mode overview	20
Figure 6 – Fieldbus memory management unit overview	22
Figure 7 – Layering of communication.....	23
Figure 8 – Flow of Type 12 service primitives	24
Figure 9 – Successful mailbox write sequence.....	35
Figure 10 – Successful mailbox read sequence.....	35
Table 1 – Auto-increment physical read (APRD)	25
Table 2 – Configured-address physical read (FPRD).....	25
Table 3 – Broadcast read (BRD)	26
Table 4 – Logical read (LRD)	27
Table 5 – Auto-increment physical write (APWR)	27
Table 6 – Configured-address physical write (FPWR)	28
Table 7 – Broadcast write (BWR).....	28
Table 8 – Logical write (LWR).....	29
Table 9 – Auto-increment physical read/write (APRW)	30
Table 10 – Configured-address physical read/write (FPRW).....	30
Table 11 – Broadcast read/write (BRW)	31
Table 12 – Logical read/write (LRW)	31
Table 13 – Auto-increment physical read / multiple write (ARMW).....	32
Table 14 – Configured-address physical read / multiple write (FRMW).....	32
Table 15 – Provide network variable (PNV).....	33
Table 16 – Mailbox write	36
Table 17 – Mailbox read update	37
Table 18 – Mailbox read	38
Table 19 – Read local	39
Table 20 – Write local	39
Table 21 – Event local	40

INTERNATIONAL ELECTROTECHNICAL COMMISSION

INDUSTRIAL COMMUNICATION NETWORKS – FIELDBUS SPECIFICATIONS –

Part 3-12: Data-link layer service definition – 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 itself does not provide any attestation of conformity. Independent certification bodies provide conformity assessment services and, in some areas, access to IEC marks of conformity. IEC is not responsible for any services carried out by independent certification bodies.
- 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.
- 9) Attention is drawn to the possibility that some of the elements of this IEC Publication may be the subject of patent rights. IEC shall not be held responsible for identifying any or all such patent rights.

Attention is drawn to the fact that the use of the associated protocol type is restricted by its 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 layer protocol type to be used with other layer protocols of the same type, or in other type combinations explicitly authorized by its intellectual-property-right holders.

NOTE Combinations of protocol types are specified in IEC 61784-1 and IEC 61784-2.

International Standard IEC 61158-3-12 has been prepared by subcommittee 65C: Industrial networks of IEC technical committee 65: Industrial-process measurement, control and automation.

This third edition cancels and replaces the second edition published in 2010. This edition constitutes a technical revision. The main changes with respect to the previous edition are listed below:

- Editorial improvements for clarification.

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

FDIS	Report on voting
65C/759/FDIS	65C/769/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.

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

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

The committee has decided that the contents of this publication will remain unchanged until the stability 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.

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 61158-1.

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

INDUSTRIAL COMMUNICATION NETWORKS – FIELDBUS SPECIFICATIONS –

Part 3-12: Data-link layer service definition – Type 12 elements

1 Scope

1.1 General

This part of IEC 61158 provides common elements for basic time-critical messaging communications between devices in an automation environment. The term “time-critical” is used to represent the presence of a time-window, within which one or more specified actions are required to be completed with some defined level of certainty. Failure to complete specified actions within the time window risks failure of the applications requesting the actions, with attendant risk to equipment, plant and possibly human life.

This standard defines in an abstract way the externally visible service provided by the Type 12 fieldbus data-link layer in terms of

- a) the primitive actions and events of the service;
- b) the parameters associated with each primitive action and event, and the form which they take;
- c) the interrelationship between these actions and events, and their valid sequences.

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

- the Type 12 fieldbus application layer at the boundary between the application and data-link layers of the fieldbus reference model;
- systems management at the boundary between the data-link layer and systems management of the fieldbus reference model.

1.2 Specifications

The principal objective of this standard is to specify the characteristics of conceptual data-link layer services suitable for time-critical communications, and thus supplement the OSI Basic Reference Model in guiding the development of data-link protocols for time-critical communications. A secondary objective is to provide migration paths from previously-existing industrial communications protocols.

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

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

1.3 Conformance

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

There is no conformance of equipment to this data-link layer service definition standard. Instead, conformance is achieved through implementation of the corresponding data-link protocol that fulfils the Type 12 data-link layer services defined in this standard.

2 Normative references

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

NOTE All parts of the IEC 61158 series, as well as IEC 61784-1 and IEC 61784-2 are maintained simultaneously. Cross-references to these documents within the text therefore refer to the editions as dated in this list of normative references.

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

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

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

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

IEEE 802.1D, *IEEE Standard for Local and metropolitan area networks – Media Access Control (MAC) Bridges*; available at <<http://www.ieee.org>>

3 Terms, definitions, symbols, abbreviations and conventions

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

3.1 Reference model terms and definitions

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

3.1.1 DL-address	[7498-3]
3.1.2 DL-connectionless-mode transmission	[7498-1]
3.1.3 correspondent (N)-entities	[7498-1]
correspondent DL-entities (N=2)	
correspondent Ph-entities (N=1)	
3.1.4 DL-duplex-transmission	[7498-1]
3.1.5 (N)-entity	[7498-1]
DL-entity (N=2)	
Ph-entity (N=1)	
3.1.6 (N)-layer	[7498-1]
DL-layer (N=2)	
Ph-layer (N=1)	
3.1.7 layer-management	[7498-1]
3.1.8 peer-entities	[7498-1]

3.1.9 primitive name	[7498-3]
3.1.10 DL-protocol	[7498-1]
3.1.11 DL-protocol-data-unit	[7498-1]
3.1.12 DL-relay	[7498-1]
3.1.13 reset	[7498-1]
3.1.14 responding-DL-address	[7498-3]
3.1.15 routing	[7498-1]
3.1.16 segmenting	[7498-1]
3.1.17 (N)-service	[7498-1]
DL-service (N=2)	
Ph-service (N=1)	
3.1.18 (N)-service-access-point	[7498-1]
DL-service-access-point (N=2)	
Ph-service-access-point (N=1)	
3.1.19 DL-service-data-unit	[7498-1]
3.1.20 DL-simplex-transmission	[7498-1]
3.1.21 DL-subsystem	[7498-1]
3.1.22 systems-management	[7498-1]
3.1.23 DLS-user	[7498-1]
3.1.24 DLS-user-data	[7498-1]

3.2 Service convention terms and definitions

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

3.2.1 acceptor
3.2.2 asymmetrical service
3.2.3 confirm (primitive); requestor.deliver (primitive)
3.2.4 deliver (primitive)
3.2.5 DL-service-primitive; primitive
3.2.6 DL-service-provider
3.2.7 DL-service-user
3.2.8 DL-user-optional-facility
3.2.9 indication (primitive); acceptor.deliver (primitive)
3.2.10 request (primitive); requestor.submit (primitive)
3.2.11 requestor
3.2.12 response (primitive); acceptor.submit (primitive)

3.2.13 submit (primitive)

3.2.14 symmetrical service

3.3 Data-link service terms and definitions

3.3.1

application

function or data structure for which data is consumed or produced

3.3.2

application objects

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

3.3.3

basic slave

slave device that supports only physical addressing of data

3.3.4

bit

unit of information consisting of a 1 or a 0

Note 1 to entry: This is the smallest data unit that can be transmitted.

3.3.5

client

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

2) initiator of a message to which a server reacts

3.3.6

connection

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

3.3.7

cyclic

events which repeat in a regular and repetitive manner

3.3.8

cyclic redundancy check

CRC

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

3.3.9

data

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

3.3.10

data consistency

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

3.3.11

device

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

3.3.12**distributed clocks**

method to synchronize slaves and maintain a global time base

3.3.13**DL-segment****link****local link**

single DL-subnetwork in which any of the connected DLEs may communicate directly, without any intervening DL-relaying, whenever all of those DLEs that are participating in an instance of communication are simultaneously attentive to the DL-subnetwork during the period(s) of attempted communication

3.3.14**error**

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

3.3.15**event**

instance of a change of conditions

3.3.16**fieldbus memory management unit**

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

3.3.17**fieldbus memory management unit entity**

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

3.3.18**frame**

denigrated synonym for DLPDU

3.3.19**full slave**

slave device that supports both physical and logical addressing of data

3.3.20**interface**

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

3.3.21**master**

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

3.3.22**mapping**

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

3.3.23

medium

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

Note 1 to entry: "media" is used as the plural of medium.

3.3.24

message

ordered series of octets intended to convey information

Note 1 to entry: Normally used to convey information between peers at the application layer.

3.3.25

network

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

3.3.26

node

- a) single DL-entity as it appears on one local link
- b) end-point of a link in a network or a point at which two or more links meet

[SOURCE: IEC 61158-2, 3.1.31 for option b), with some wording adjustment]

3.3.27

object

abstract representation of a particular component within a device

Note 1 to entry: An object can be

- a) an abstract representation of the capabilities of a device, composed of any or all of the following components:
 - 1) data (information which changes with time);
 - 2) configuration (parameters for behavior);
 - 3) methods (things that can be done using data and configuration); or
- b) a collection of related data (in the form of variables) and methods (procedures) for operating on that data that have a clearly defined interface and behavior.

3.3.28

process data

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

3.3.29

receiving DLS-user

DL-service user that acts as a recipient of DL-user-data

Note 1 to entry: A DL-service user can be concurrently both a sending and receiving DLS-user.

3.3.30

sending DLS-user

DL-service user that acts as a source of DL-user-data

3.3.31

server

object which provides services to another (client) object

3.3.32 service

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

3.3.33 slave

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

3.3.34 Sync manager

collection of control elements to coordinate access to concurrently used objects

3.3.35 Sync manager channel

single control elements to coordinate access to concurrently used objects

3.3.36 switch

MAC bridge as defined in IEEE 802.1D

3.4 Symbols and abbreviations

APRD	Auto-increment physical read
APRW	Auto-increment physical read/write
APWR	Auto-increment physical write
ARMW	Auto-increment physical read / multiple write
BRD	Broadcast read
BRW	Broadcast read/write
BWR	Broadcast write
CAN	Controller area network
CoE	CAN application protocol over Type 12 services
CSMA/CD	Carrier sense multiple access with collision detection
DC	Distributed clocks
DL-	Data-link layer (as a prefix)
DLC	DL-connection
DLCEP	DL-connection-end-point
DLE	DL-entity (the local active instance of the data-link layer)
DLL	DL-layer
DLPCI	DL-protocol-control-information
DLPDU	DL-protocol-data-unit
DLM	DL-management
DLME	DL-management entity (the local active instance of DL-management)
DLMS	DL-management service
DLS	DL-service
DLSAP	DL-service-access-point
DLSDU	DL-service-data-unit
E ² PROM	Electrically erasable programmable read only memory
EoE	Ethernet tunneled over Type 12 services
ESC	Type 12 slave controller
FCS	Frame check sequence

FIFO	First-in first-out (queuing method)
FMMU	Fieldbus memory management unit
FoE	File access with Type 12 services
FPRD	Configured address physical read
FPRW	Configured address physical read/write
FPWR	Configured address physical write
FRMW	Configured address physical read/multiple write
HDR	Header
ID	Identifier
IP	Internet protocol
LAN	Local area network
LRD	Logical memory read
LRW	Logical memory read/write
LWR	Logical memory write
MAC	Medium access control
MDI	Media-dependent interface (specified in ISO/IEC 8802-3)
MDX	Mailbox data exchange
MII	Media-independent interface (specified in ISO/IEC 8802-3)
PDI	Physical device interface (a set of elements that allows access to DL-services from the
PDO	Process data object
Ph-	Physical layer (as a prefix)
PhE	Ph-entity (the local active instance of the physical layer)
PhL	Ph-layer
PHY	Physical layer device (specified in ISO/IEC 8802-3)
PNV	Publish network variable
OSI	Open systems interconnection
QoS	Quality of service
RAM	Random access memory
Rx	Receive
SDO	Service data object
SII	Slave information interface
SyncM	Synchronization manager
TCP	Transmission control protocol
Tx	Transmit
UDP	User datagram protocol
WKC	Working counter

3.5 Common conventions

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

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

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

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

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

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

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

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

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

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

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

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

4 Data-link layer services and concepts

4.1 Operating principle

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

From an Ethernet point of view, a Type 12 segment is a single Ethernet device which receives and sends standard ISO/IEC 8802-3 Ethernet frames. However, this Ethernet device is not limited to a single Ethernet controller with downstream microprocessor, but may consist of a large number of Type 12 slave devices. These process the incoming Ethernet frames while they are in transit within the device, reading data from the Ethernet frame and/or inserting their own data into the frame before transferring the frame to the next slave device. The last slave device within the segment sends the fully processed Ethernet frame back in the reverse

direction through the chain of devices, returning the collected information through the first slave device to the master, which receives it as an Ethernet response frame.

This procedure utilizes the full duplex capability of Ethernet: both communication directions are operated independently with reading and writing by the slaves on the outbound path and only transmission-to-reception timing measurements on the inbound path as the Ethernet frame retraverses each intermediate slave device.

Full-duplex communication between a master device and a Type 12 segment consisting of one or several slave devices may be established without using a switch.

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 can lead to increased cabling effort and infrastructure cost. Particularly for automation applications, a line or tree topology often is preferable.

The slave node arrangement represents an open-loop bus. At the open end, the master device sends frames, either directly or via Ethernet switches; it receives them at the other end after they have been processed by each intervening device. Each Ethernet frame is relayed from the first node to the next one, and thence to each other node in series. The last node returns the Ethernet frame back to the master using 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 form. 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. Branches are possible if a device has more than two ports. This standard allows up to two branching links in addition to the basic set of two series interfaces.

An Ethernet frame received on port n (n not zero) is forwarded to port $n+1$. If there is no port $n+1$ the Ethernet frame is forwarded to port 0. If no device is connected or the port is closed by the master, a request to send to that port will be processed as if the same data are received by this port (i.e. loop is closed).

4.3 Data-link layer overview

A single Ethernet frame can carry several Type 12 DLPDUs, which are blocked into the Ethernet frame without gaps. Several nodes can be addressed individually by these DLPDUs. The Ethernet frame is terminated with the last Type 12 DLPDU, except when the frame size is less than 64 octets, in which case the Ethernet frame is padded to 64 octets.

This blocking leads to better utilization of the Ethernet bandwidth than would separate Ethernet frames to and from each slave node. However, for e.g. a 2-channel digital input node with just two bits of user data, the overhead of a single Type 12 DLPDU can still be excessive.

Therefore slave nodes may also support logical address mapping. The process data can be inserted anywhere within a logical address space. If a Type 12 DLPDU 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 their appropriate place(s) within the process data, as noted in Figure 1.

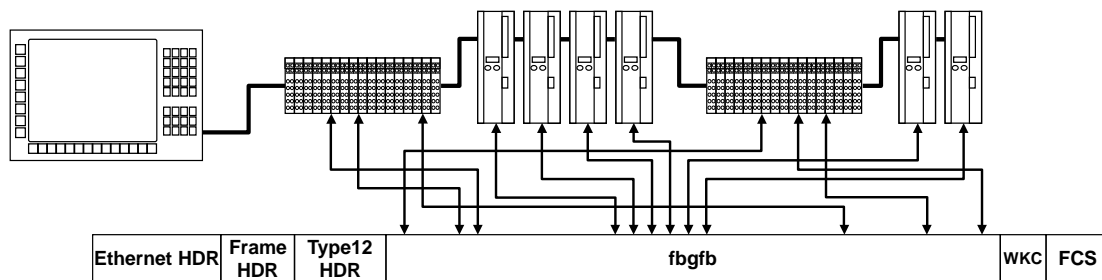


Figure 1 – Mapping of logical data in an Ethernet frame consisting of a single Type 12 DLPDU

Each node that detects an address match with the process image inserts its data, so that many nodes can be addressed simultaneously with a single Type 12 DLPDU. The master can assemble a completely sorted logical process image via a single Type 12 DLPDU, independent of the physical wiring order of the slave devices.

Additional mapping is no longer required in the master, so that the process data can be transferred directly to one or more 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} octets (= 4 GB). Thus a Type 12 fieldbus 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 standard Ethernet cables, a very large number of I/O channels can be connected to automation devices so that they can be accessed with high bandwidth, minimum delay and a near-optimum effective usable data rate. At the same time, devices such as fieldbus scanners can be connected as well, thus preserving existing technologies and standards.

4.4 Error detection overview

Type 12 master and slave nodes (DLEs) check the Ethernet frame check sequence (FCS) to determine whether a frame is received correctly. Since one or several slaves may modify the frame during the transfer, the FCS is checked by each node on reception and recalculated during retransmission. If a slave detects a checksum error, the slave does not repair the FCS but flags the master by incrementing an error counter, so that the source of a single fault can be located precisely within the open-loop topology.

When reading data from or writing data to a Type 12 DLPDU, the addressed slave increments a working counter (WKC) positioned at the end of the DLPDU. Slaves which are merely forwarding the DLPDU, but not extracting information from it or inserting information within it, do not modify the counter. By comparing the working counter with the expected number of accessing slave nodes, a master can check whether the expected number of nodes have processed the corresponding DLPDU.

4.5 Parameter and process data handling introduction

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

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

4.6 Node reference model

4.6.1 Mapping onto OSI Basic Reference Model

Type 12 services are described using the principles, methodology and model of ISO/IEC 7498-1 (OSI). The OSI model provides a layered approach to communications standards, whereby the layers can be developed and modified independently. The Type 12 specification defines functionality from top to bottom of a full OSI communications stack. Functions of the intermediate OSI layers, layers 3–6, are consolidated into either the Type 12 data-link layer or the DL-user of the Type 12 data-link layer. The Type 12 data-link reference model is shown in Figure 2.

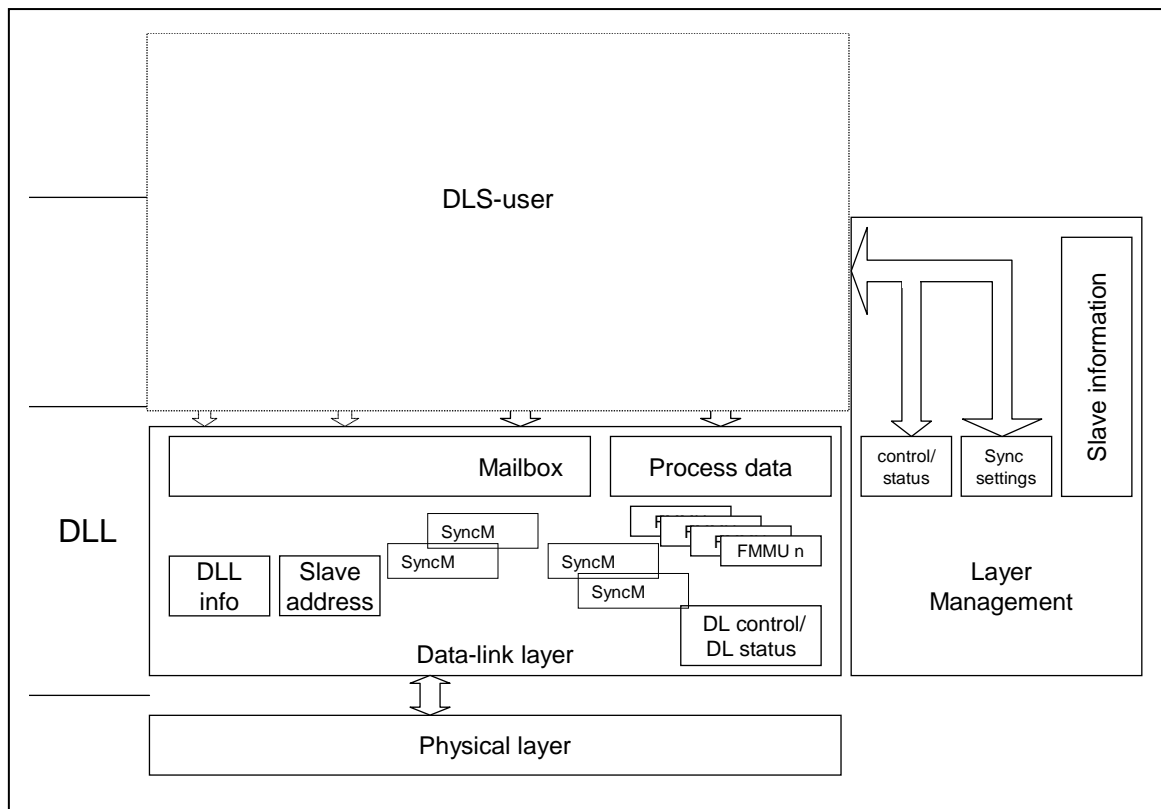


Figure 2 – Type 12 data-link reference model

4.6.2 Data-link layer features

The data-link layer provides basic time-critical support for data communications among devices connected. 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 data is made available to the DL-user 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 Type 12 modes

4.7.2.1 Open mode

In the open mode, one or several Type 12 segments may be connected to a standard switching device as shown in Figure 3. The first slave device within a Type 12 segment then has an ISO/IEC 8802-3 MAC address representing the entire segment. This segment address slave device replaces destination address field with the source address field and source address field with its own MAC address within the Ethernet frame if the frame follows the coding rules of Type 12. If this type of frame is transported via UDP, this device will handle the source and destination IP addresses in the same way as the MAC addresses and the UDP source and destination port numbers in order to ensure that the response frame fully satisfies UDP/IP protocol standards. Additionally this device protects the slaves within the segment against unauthorized access by master devices or generic Ethernet devices.

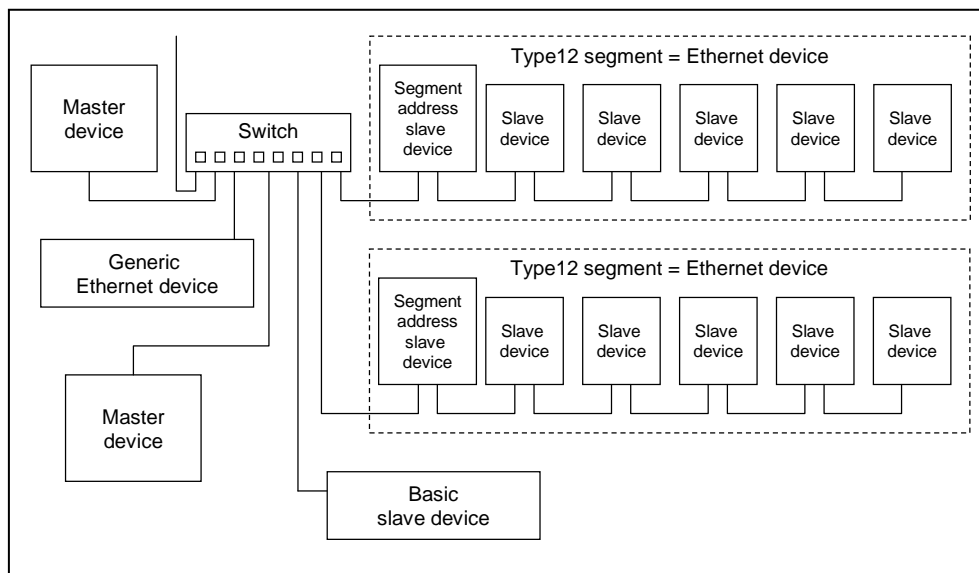


Figure 3 – Type 12 segments in open mode

4.7.2.2 Direct mode

In the direct mode, one Type 12 segment is connected to the standard Ethernet port of the controlling or master device as shown in Figure 4. The MAC address fields of the Ethernet frames are not checked.

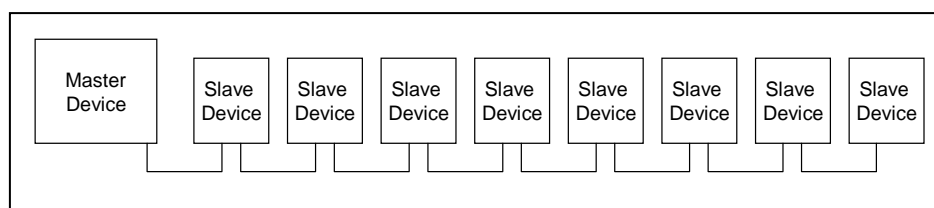


Figure 4 – Type 12 segment in direct mode

4.7.3 Logical topology

In logical terms, the slave arrangement within a Type 12 segment represents a bus connected as an open full-duplex loop. At the input end of the open loop, the master device inserts Ethernet frames, either directly or via standard Ethernet switches, and receives them at the output end of the open loop after they have been processed by all slave devices. All frames are relayed from the first slave device to the next one. The last slave device returns the frame back through all the other slave devices to the master. The result is an open logical loop realized by consecutive segments of full-duplex physical line.

Received Ethernet frames are processed octet by octet "on the fly" by the slave devices according to their physical sequence within the open loop structure. In this case, each slave device recognizes relevant commands and executes them accordingly while the frames (delayed by a constant time, typically below 1 μ s) are forwarded to the next device in the open loop. Data extraction and insertion are performed by the data-link layer as the Ethernet frame transits the slave device, in a manner that is independent of the response times of any microprocessors within (or connected to) the slave device.

Full-duplex physical branches are possible in the Type 12 segment at any location, since a branch does not break the logical loop. Branches can be used to build a flexible tree structure, thus permitting very simple wiring.

4.8 Addressing

4.8.1 Addressing overview

Different addressing modes are supported for slaves, as noted in Figure 5. The header within the Type 12 DLPDU contains a 32-bit address, which is used for physical node addressing or logical addressing.

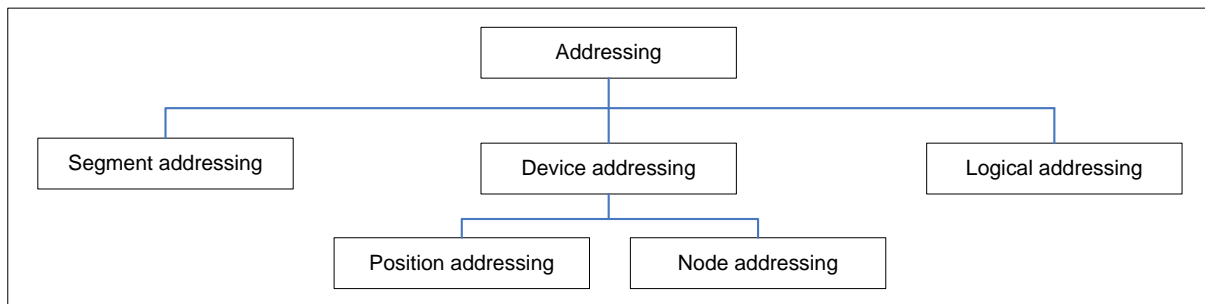


Figure 5 – Addressing mode overview

4.8.2 Segment addressing

MAC addresses according to ISO/IEC 8802-3 are used for segment addressing.

4.8.3 Device addressing

4.8.3.1 Structure of device addresses

With this address mode, a 32-bit address within each Type 12 DLPDU is split into a 16-bit slave device address and a 16-bit physical address within the slave device, thus leading to 2^{16} slave device addresses, each with an associated 16-bit local address space. With device addressing, each Type 12 DLPDU uniquely addresses one single slave device.

This mode is most suitable for transferring parameter data. There are two different device addressing mechanisms as follows:

- position addressing;
- node addressing.

4.8.3.2 Position addressing

Position addressing is used to address each slave device via its physical position within the segment. Each slave device increments the 16-bit address field as the DLPDU transits the slave device; that device which receives a DLPDU with an address field of value 0 is the one being addressed. Due to the mechanism employed to update this address while transiting the node, the slave device address in position addressing is referred to as an *auto-increment address*.

EXAMPLE If the 10th slave device in the segment is to be addressed, the master device sends a DLPDU with position addressing with a start device address value of -9, which is incremented by one by each device which the DLPDU transits.

In practice, position addressing is used during a start-up phase, during which the master assigns configured node addresses to the slaves, after which they can be addressed irrespective of their physical position in the segment via use of those node addresses.

This topology-based addressing mechanism has the advantage that no slave node addresses need to be set manually at the slaves.

4.8.3.3 Node addressing

With node addressing, the slaves are addressed via configured node addresses assigned by the master during the data-link start-up phase. This ensures that, even if the segment topology is changed or devices are added or removed, the slave devices can still be addressed via the same configured address.

The slave device address in node addressing is referred to as *configured station address*.

4.8.4 Logical addressing

For logical addressing within a segment the entire 32-bit address field of each Type 12 DLPDU is used as a single unstructured address. With logical addressing, slaves are not addressed individually, but instead a section of the segment-wide 4 GB logical address space is addressed. Any number of slaves may use the same or overlapping sections.

The data region address used in this mode is referred to as a *logical address*.

The logical addressing mode is particularly suitable for transferring and/or exchanging cyclic process data.

4.8.5 FMMU introduction

Fieldbus memory management units (FMMU) handle the local assignment of physical slave memory addresses to logical segment-wide addresses, as shown in Figure 6.

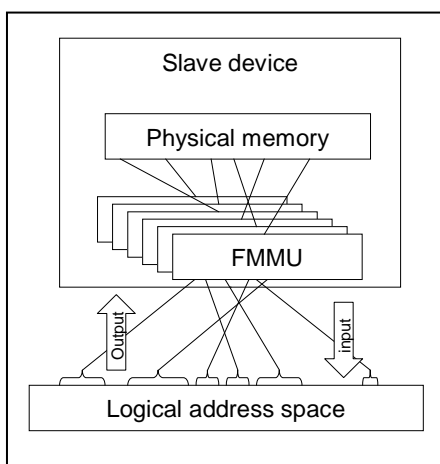


Figure 6 – Fieldbus memory management unit overview

Configuration of the FMMU entities is performed by the master device and transferred to the slave devices during the data-link start-up phase. For each FMMU entity, the following items are configured: a logical, bit-oriented start address, a physical memory start address, a bit length, and a type that specifies the direction of the mapping (input or output). Any data within the memory of a slave device can thus be mapped bit-wise to any logical address.

When a Type 12 DLPDU with logical addressing is received, the slave device checks whether one of its FMMU entities shows an address match. If appropriate, it inserts data at the associated position of the data field into the DLPDU (input type) or extracts data from the associated position of the DLPDU (output type). DLPDUs can therefore be assembled flexibly and optimized to the requirements of the control application.

4.8.6 Sync Manager introduction

The Sync Manager (SyncM) controls the access to the DLS-user memory. Each SyncM channel defines a consistent area of the DLS-user memory.

4.9 Slave classification

4.9.1 Full slave

There is a differentiation between full slaves, which support all addressing modes, and basic slaves, which support only a subset of the addressing modes. Master devices may support the basic slave functionality to allow for direct communication with another master device. Slave devices should support the full slave functionality.

A full slave supports

- logical addressing;
- position addressing; and
- node addressing.

Thus full slave devices need both an FMMU and address auto-increment functionality.

Full slaves may support segment addressing. Full slaves that support segment addressing are called segment address slave device.

Only full slaves can be connected within a Type 12 segment.

4.9.2 Basic slave

Basic slave devices support node addressing and segment addressing.

4.10 Structure of the communication layer in the slave

The attributes are related to the physical memory of a slave, which can be read or written from the master. The physical memory consists of registers and DL-user memory. The register area contains information for configuration, management and device identification in the DLL. The use of the DL-user memory is defined by the DL-user. Figure 7 shows the outline of the interactions between DL-user and DLL and between DLL and communication.

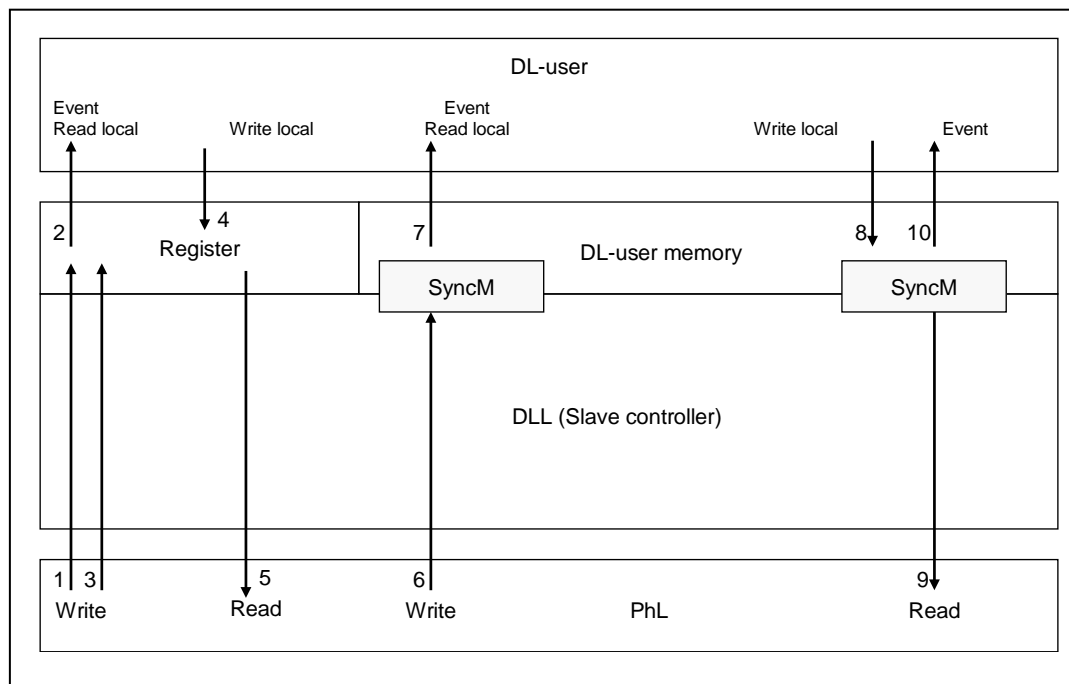


Figure 7 – Layering of communication

A DL write service to the register area (1) may (depending on the written register) result in an event indication primitive to the DL-User, followed by a read local request primitive from the DL-user to get the written value (2). Otherwise, the DL write service will only access the register area without informing the DL-user (3). The DL-user can read the register area with read local primitive at any time.

The DL-user will set up the register with write register local and update it if needed and possible (4). A DL read service to the register area will only access the register area without informing the DL-user (5).

The DL-user memory area access is coordinated by the sync manager. Access without the sync manager can be done in a similar way as the register access but consistency constraints and the lack of events indicating changes caused by the master may limit this method of use. The description of DL-user memory access assumes use of the sync manager.

A DL write service to the DL-user memory area (6) will result in an event indication primitive to the DL-user, followed by a read local request primitive from the DL-user to get the written values (7).

The DL-user will write the DL-user memory area with a write local request primitive (8). The DL-user memory area will be read by the master with a DL read service (9) which will issue an event indication primitive to the DL-user (10) to indicate that the DL-user memory area has been read and can be written by the DL-user again.

A slave responds to all read and write requests, and may respond to combined read/write requests.

5 Communication services

5.1 Overview

Services are described from the point of view of the master. The execution of the service within the slave is described in Clause 6.

The data-link layer specifies services for reading, writing and exchanging (reading followed immediately by overwriting) data from physical memory within the slaves.

NOTE For simplification the expression "reads memory" is used instead of "reads data from physical memory". Equivalently the expression "writes memory" is used instead of "writes data to physical memory".

With the read service a master reads registers or DL-user memory from one or many slaves.

The basic service procedure of all variants of read service is the same except for broadcast read. The addressed unit will copy the data in the data parameter. With broadcast service, the slave will execute a bitwise-OR operation of the parameter data with the memory or register data.

If there is only one slave connected to a master, the service procedure is executed according to the client server model. If several slaves are connected (always in series), the invocation of the service procedure is handled in such a way that the output of one slave serves as the input to the next slave.

The service procedures are similar to the procedures defined in IEEE 802.1D but forwarding and processing are combined. As Type 12 uses confirmed services instead of unconfirmed services as specified in IEEE 802.1D, the flow of information between service primitives is adapted to this situation. The master initiates a request service and receives a corresponding confirmation. Each slave receives an indication of the data it receives, while forwarding that data after possible update to the next slave. Figure 8 shows this control flow.

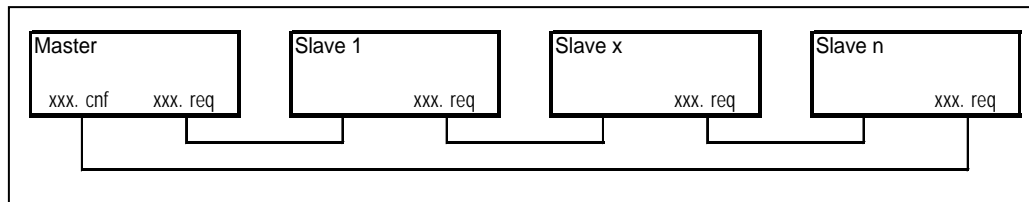


Figure 8 – Flow of Type 12 service primitives

5.2 Read services

5.2.1 Overview

With the read services, a master reads data from the memory of one or many slaves.

5.2.2 Positional physical read (APRD)

With the APRD service, a master reads data from memory or register of one slave selected by the physical ordering of the slave in the segment. Table 1 shows the service primitives and parameter of the APRD service.

Table 1 – Auto-increment physical read (APRD)

DL-AUTOINCREMENT-PHYSICALREAD Parameter name	Request	Confirm
	input	output
Ordinal device number	M	M
Device data area	M	
DLS-user data	U	U
Working counter	M	M
NOTE The method by which a confirm primitive is correlated with its corresponding preceding request primitive is a local matter. See 1.2.		

Parameter description

Ordinal device number

This parameter specifies the ordinal index of the addressed device in the wired communication chain. In the confirmation to the master the number of transited slave devices is given.

Device data area

This parameter specifies the location in the physical memory of the slave where the data to be read is stored.

DLS-user data

On confirm this parameter specifies the data read from the device if the access was valid at the addressed slave. Otherwise the value specified by the request is returned unchanged.

Working counter

This parameter is incremented if the data was successfully read.

5.2.3 Configured-address physical read (FPRD)

With the FPRD service, a master reads data from memory or register of one slave selected by the slave's configured station address. Table 2 shows the service primitives and parameter of the FPRD service.

Table 2 – Configured-address physical read (FPRD)

DL-CONFIGURED-PHYSICALREAD Parameter name	Request	Confirm
	input	output
Configured device number	M	
Device data area	M	
DLS-user data	U	U
Working counter	M	M
NOTE The method by which a confirm primitive is correlated with its corresponding preceding request primitive is a local matter. See 1.2.		

Parameter description

Configured device number

This parameter specifies the configured station address of the addressed device.

Device data area

This parameter specifies the location in the physical memory of the slave where the data to be read is stored.

DLS-user data

On confirm this parameter specifies the data read from the device if the access was valid at the addressed slave. Otherwise the value specified by the request is returned unchanged.

Working counter

This parameter is incremented if the data was successfully read.

5.2.4 Broadcast read (BRD)

With the BRD service, a master reads data from a physical memory area or register, which will be a bitwise-OR between the incoming data and the selected object at all slaves. Table 3 shows the service primitives and parameter of the BRD service.

Table 3 – Broadcast read (BRD)

DL-BROADCAST-READ Parameter name	Request	Confirm
	input	output
Broadcast address	M	M
Device data area	M	
DLS-user data	U	U
Working counter	M	M
NOTE The method by which a confirm primitive is correlated with its corresponding preceding request primitive is a local matter. See 1.2.		

Parameter description

Broadcast address

This parameter is incremented at each slave.

Device data area

This parameter specifies the location in the physical memory where the data to be read is stored.

DLS-user data

On confirm this parameter specifies the result of the bitwise-OR operation between the parameter data of the request and the selected object at the response.

Working counter

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

5.2.5 Logical read (LRD)

With the LRD service, a master reads data from the memory or a register of one or many slaves selected by a logical address. Table 4 shows the service primitives and parameter of the LRD service.

Table 4 – Logical read (LRD)

DL-LOGICAL-READ Parameter name	Request	Confirm
	input	output
Logical memory address	M	
DLS-user data		U
Working counter	M	M
NOTE The method by which a confirm primitive is correlated with its corresponding preceding request primitive is a local matter. See 1.2.		

Parameter description**Logical memory address**

This parameter specifies the start address in the logical memory where the data to be read is located.

DLS-user data

This parameter specifies the data that was read.

Working Counter

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

5.3 Write services**5.3.1 Overview**

With the write services a master writes data to a register or the memory of one or many slaves.

5.3.2 Positional physical write (APWR)

With the APWR service, a master writes data in memory or register of one slave selected by the physical ordering of the slave in the segment. Table 5 shows the service primitives and parameter of the APWR service.

Table 5 – Auto-increment physical write (APWR)

DL-AUTOINCREMENT-PHYSICALWRITE Parameter name	Request	Confirm
	input	output
Ordinal device number	M	M
Device data area	M	
DLS-user data	U	
Working counter	M	M
NOTE The method by which a confirm primitive is correlated with its corresponding preceding request primitive is a local matter. See 1.2.		

Parameter description**Ordinal device number**

This parameter specifies the ordinal index of the address device in the wired communications chain. In the confirmation to the master the number of transited slave devices is given.

Device data area

This parameter specifies the location in the physical memory of the slave where the data to be written is stored.

DLS-user data

This parameter specifies the data to be written.

Working counter

This parameter is incremented if the data was successfully written.

5.3.3 Configured-address physical write (FPWR)

With the FPWR service, a master writes to memory or register of one slave selected by the slave's configured station address. Table 6 shows the service primitives and parameter of the FPWR service.

Table 6 – Configured-address physical write (FPWR)

DL-CONFIGURED-PHYSICALWRITE Parameter name	Request	Confirm
	input	output
Configured device number	M	
Device data area	M	
DLS-user data	U	
Working counter	M	M (=)

NOTE The method by which a confirm primitive is correlated with its corresponding preceding request primitive is a local matter. See 1.2.

Parameter description

Configured device number

This parameter specifies the configured station address of the addressed device.

Device data area

This parameter specifies the location in the physical memory of the slave where the data to be written is stored.

DLS-user data

This parameter specifies the data to be written.

Working counter

This parameter is incremented if the data was successfully written.

5.3.4 Broadcast write (BWR)

With the BWR service, a master writes a physical memory area to all slaves. Table 7 shows the service primitives and parameter of the BWR service.

Table 7 – Broadcast write (BWR)

DL-BROADCAST-WRITE Parameter name	Request	Confirm
	input	output
Broadcast address	M	M
Device data area	M	
DLS-user data	U	
Working counter	M	M (=)

NOTE The method by which a confirm primitive is correlated with its corresponding preceding request primitive is a local matter. See 1.2.

Parameter description

Broadcast address

This parameter is incremented at each slave.

Device data area

This parameter specifies the location in the physical memory where the data to be written is stored.

DLS-user data

This parameter specifies the data to be written.

Working counter

This parameter is incremented by each slave that writes data in its physical memory.

5.3.5 Logical write (LWR)

With the LWR service, a master writes to memory or register of one or many slaves selected by a logical address. Table 8 shows the service primitives and parameter of the LWR service.

Table 8 – Logical write (LWR)

DL-LOGICAL-WRITE Parameter name	Request	Confirm
	input	output
Logical memory address	M	
DLS-user data	U	
Working counter	M	M
NOTE The method by which a confirm primitive is correlated with its corresponding preceding request primitive is a local matter. See 1.2.		

Parameter description

Logical memory address

This parameter specifies the start address in the logical memory where the data to be written is located.

DLS-user data

This parameter specifies the data to be written.

Working counter

This parameter is incremented by all slaves that detect an address match of the requested logical memory area.

5.4 Combined read/write services

5.4.1 Overview

For the combined read/write services the rules for read and/or write of the addressed slave apply.

5.4.2 Positional physical read/write (APRW)

With the APRW service, a master reads out memory or register of one slave selected by the physical ordering of the slave in the segment and writes data in memory or register of the same slave. Table 9 shows the service primitives and parameter of the APRW service.

Table 9 – Auto-increment physical read/write (APRW)

DL-AUTOINCREMENT-PHYSICALREADWRITE Parameter name	Request	Confirm
	input	output
Ordinal device number	M	M
Device data area	M	
DLS-user data	U	U
Working counter	M	M
NOTE The method by which a confirm primitive is correlated with its corresponding preceding request primitive is a local matter. See 1.2.		

Parameter description

Ordinal device number

This parameter specifies the ordinal index of the addressed device in the wired communication chain. In the confirmation to the master the number of transited slave devices is given.

Device data area

This parameter specifies the location in the physical memory of the slave where data to be read and written is stored.

DLS-user data

This parameter specifies the data to be written, or the data that was read.

Working counter

This parameter is incremented if the data was successfully written and read.

5.4.3 Configured-address physical read/write (FPRW)

With the FPRW service, a master reads from memory or register of one slave selected by the slave’s configured station address and writes data to the same object. Table 10 shows the service primitives and parameter of the FPRW service.

Table 10 – Configured-address physical read/write (FPRW)

DL-CONFIGURED-PHYSICALREADWRITE Parameter name	Request	Confirm
	input	output
D_address	M	
Device data area	M	
DLS-user data	U	U
Working counter	M	M
NOTE The method by which a confirm primitive is correlated with its corresponding preceding request primitive is a local matter. See 1.2.		

Parameter description

Configured device number

This parameter specifies the configured station address of the addressed device.

Device data area

This parameter specifies the location in the physical memory of the slave where the data to be read and written is stored.

DLS-user data

This parameter specifies the data to be written, or the data that was read.

Working counter

This parameter is incremented if the data was successfully written and read.

5.4.4 Broadcast read/write (BRW)

With the BRW service, a master reads a physical memory area or register, which will be bitwise-OR by all slaves and writes in data collected at all previous slaves. Table 11 shows the service primitives and parameter of the BRW service.

Table 11 – Broadcast read/write (BRW)

DL-BROADCAST-READWRITE Parameter name	Request	Confirm
	input	output
Broadcast address	M	M
Device data area	M	
DLS-user data	U	U
Working counter	M	M
NOTE The method by which a confirm primitive is correlated with its corresponding preceding request primitive is a local matter. See 1.2.		

Parameter description

Broadcast address

This parameter is incremented at each slave.

Device data area

This parameter specifies the location in the physical memory where the data to be read and written is stored.

DLS-user data

This parameter specifies the data to be written to the device, or the result of the bitwise-OR operation of the data that was read from each device.

Working counter

This parameter is incremented by all slaves which made the bitwise-OR of the requested data and wrote data into their physical memory.

5.4.5 Logical read/write (LRW)

With the LRW service, a master writes and reads memory to one or many slaves selected by a logical address. Table 12 shows the service primitives and parameter of the LRW service.

Table 12 – Logical read/write (LRW)

DL-LOGICAL-READWRITE Parameter name	Request	Confirm
	input	output
Logical memory address	M	
DLS-user data	U	U
Working counter	M	M
NOTE The method by which a confirm primitive is correlated with its corresponding preceding request primitive is a local matter. See 1.2.		

Parameter description

Logical memory address

This parameter specifies the start address in the logical memory where the data to be read or written is located.

DLS-user data

This parameter specifies the data to be written, or the data that was read.

Working counter

This parameter is incremented if data was successfully written and if data was successfully read.

5.4.6 Positional physical read / multiple write (ARMW)

With the ARMW service, a master reads data out of memory or register of one slave selected by the physical ordering of the slave in the segment and writes the value of the parameter data to the same memory or register of all other slaves following. Table 13 shows the service primitives and parameter of the ARMW service.

Table 13 – Auto-increment physical read / multiple write (ARMW)

DL-AUTOINCREMENT-READMULTIPLEWRITE Parameter name	Request	Confirm
	input	output
Ordinal device number	M	M
Device data area	M	
DLS-user data	U	U
Working counter	M	M
NOTE The method by which a confirm primitive is correlated with its corresponding preceding request primitive is a local matter. See 1.2.		

Parameter description

Ordinal device number

This parameter specifies the ordinal index of the device in the wired communications chain that executes the read action. In the confirmation to the master the number of transited slave devices is given.

Device data area

This parameter specifies the location in the physical memory of the slave where data to be read and written is stored.

DLS-user data

This parameter specifies the data to be written, or the data that was read.

Working counter

This parameter is incremented if the data was successfully read.

5.4.7 Configured-address physical read / multiple write (FRMW)

With the FRMW service, a master reads from memory or register of one slave selected by the slave's configured station address and writes data to the same object of all other slaves. Table 14 shows the service primitives and parameter of the FRMW service.

Table 14 – Configured-address physical read / multiple write (FRMW)

DL-CONFIGURED-READMULTIPLEWRITE Parameter name	Request	Confirm
	input	output
Configured device number	M	
Device data area	M	
DLS-user data	U	U
Working counter	M	M
NOTE The method by which a confirm primitive is correlated with its corresponding preceding request primitive is a local matter. See 1.2.		

Parameter description**Configured device number**

This parameter specifies the configured station address of the device that is to perform the read operation.

Device data area

This parameter specifies the location in the physical memory of the slave where data to be read and written is stored.

DLS-user data

This parameter specifies the data to be written, or the data that was read.

Working counter

This parameter is incremented if the data was successfully read.

5.5 Network services**5.5.1 Overview**

Network variable services are described from the point of publisher. The data-link layer specifies services for publishing. This service is dedicated for the communication between masters or between master and standard Ethernet devices.

5.5.2 Provide network variables (PNV)

With the PNV service, a master provides data to one or many other stations (master or slaves). The primary addressing is done by the destination MAC address (group address/ individual address). The stations receiving an indication will pass the data to the DL-user. Table 15 shows the service primitives and parameter of the PNV service.

Table 15 – Provide network variable (PNV)

DL-PROVIDE-NETWORKVARIABLE Parameter name	Request	Indication
	input	output
Publisher ID	M	M (=)
Cycle	M	M (=)
List of network variables	M	M (=)

Parameter description**Publisher ID**

This parameter specifies the values of the identification octet string.

Cycle

This parameter represents a numeric identifier of the slaves' cycle and may be used to detect new values.

List of network variables

This parameter specifies a list of network variables. Each element within the list specifies:

Index

This parameter specifies the unique identifier within the provider of the network variable.

Hash value

This parameter specifies a hash value of the variable structure description of the network variables. The hash algorithm is provider-specific.

Data

This parameter specifies the values of the publisher data.

5.6 Mailbox

5.6.1 Overview

The mailbox works in both directions – from the master to a slave and from a slave to the master. It supports full duplex, independent communication in both directions and multiple DL-user protocols. Slave to slave communication is managed by the master, operating as router. The mailbox header contains an address field that allows the master to redirect services.

The mailbox uses the two sync manager channels, one per each direction (e.g. sync manager channel 0 from the master to the slave and sync manager channel 1 from the slave to the master). The sync manager channels configured as mailbox prevent the other side from an overrun. Normally the mailbox communication is non cyclic and addresses a single slave. Therefore the physical addressing without the need of a FMMU is used instead of the logical addressing.

5.6.1.1 Communication from master to slave

The master has to check the working counter at reply of a mailbox command to a slave. If the working counter did not increment (normally because the slave has not completely read the last command) or there is no response within the time limit the master has to retransmit the mailbox command. Further error recovery is in the responsibility of higher protocols.

5.6.1.2 Communication from slave to master

The master has to determine that a slave has filled the sync manager with a mailbox command and to send an appropriate read command as quickly as possible.

There are different ways to determine that a slave has filled its sync manager. A clever solution is to configure the “written bit” of the configuration header of sync manager 1 to a logical address and to read this bit cyclically. Using a logical address enables the possibility to read the bits from several slaves together and to configure each slave on an individual bit address. The drawback of this solution is that one FMMU per slave is needed.

Another solution is to simply poll the sync manager data area. The working counter of that read command will only be incremented once if the slave has filled the area with a new command.

The master has to check the working counter at reply of the mailbox command to a slave. If the working counter did not increments (normally because of the slave has not completely read the last command) or there is no response within the time limit the master has to toggle the retry parameter in the sync manager area. With a toggled retry parameter, the slave has to put the last read data in the mailbox. Further error recovery is in the responsibility of higher protocols.

The primitives of the mailbox services are mapped at the slave to the DL-user memory primitives:

Mailbox write	event, read local
Mailbox read update	write local
Mailbox read	event

Figure 9 shows the primitives between master and slave in case of a successful mailbox write sequence.

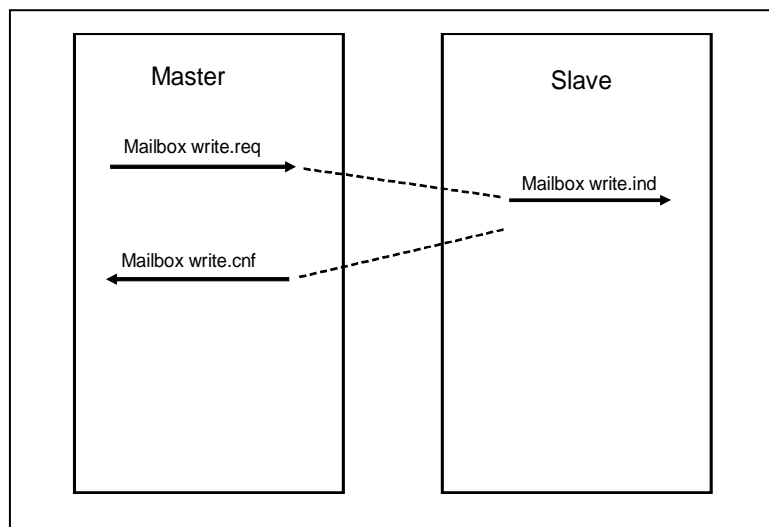


Figure 9 – Successful mailbox write sequence

Figure 10 shows the primitives between master and slave in case of a successful mailbox read sequence.

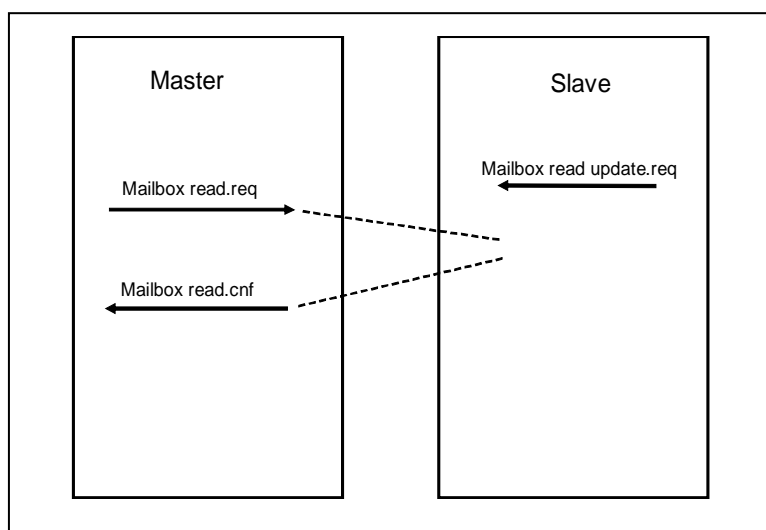


Figure 10 – Successful mailbox read sequence

5.6.2 Mailbox data transmission services

5.6.2.1 Mailbox write

The mailbox Write service as specified in Table 16 is based on writing (transmission from master to slave) memory to get an acknowledged transmission of data.

Table 16 – Mailbox write

DL-MAILBOX-WRITE Parameter name	Request	Indication	Confirm
	input	output	output
D_address	M		
MBX	M		
S_address	M	M (=)	
Channel	M	M (=)	
Priority	M	M (=)	
Type	M	M (=)	
Cnt	M	M (=)	
DLS-user data	U	U (=)	
DL-status			M

NOTE The method by which a confirm primitive is correlated with its corresponding preceding request primitive is a local matter. See 1.2.

Parameter description

D_address

This parameter specifies the node address of the destination node to allow slave to slave communication or communication beyond network boundaries using a virtual address.

MBX

This parameter specifies the mailbox.

S_address

This parameter specifies the station address of the source station to allow slave to slave communication or communication beyond network boundaries using a virtual address.

Channel

This parameter specifies the communication channel.

Priority

This parameter specifies a communication priority.

Type

This parameter specifies the protocol type of the used mailbox service.

Cnt

This parameter specifies a service counter

DLS-user data

This parameter specifies the data to be written, or that was written.

DL-status

This parameter specifies the result of the operation.

5.6.2.2 Mailbox read update

The mailbox read update service as specified in Table 17 is based on a local write to memory. The update buffer has to be retained as long as a repeated operation is possible.

Table 17 – Mailbox read update

DL-MAILBOX-READUPD Parameter name	Request	Confirm
	input	output
D_address	M	
Channel	M	
Priority	M	
Type	M	
Cnt	M	
DLS-user data	U	
DL-status		M
NOTE The method by which a confirm primitive is correlated with its corresponding preceding request primitive is a local matter. See 1.2.		

Parameter description**D_address**

This parameter specifies the station address of the destination station to allow slave to slave communication or communication beyond network boundaries using a virtual address.

Channel

This parameter specifies the communication channel.

Priority

This parameter specifies a communication priority.

Type

This parameter specifies the protocol type of the used mailbox service.

Cnt

This parameter specifies a service counter

DLS-user data

This parameter specifies the data that was read.

DL-status

This parameter specifies the result of the operation.

5.6.2.3 Mailbox read

The mailbox read service as specified in Table 18 is based on reading (transmission from slave to master) memory to get an acknowledged transmission of data.

Table 18 – Mailbox read

DL-MAILBOX-READ Parameter name	Request	Indication	Confirm
	input	output	output
S_address	M		
MBX	M		
D_address			C
Channel			C
Priority			C
Type			C
Cnt			C
DLS-user data			C
DL-status		M	M
NOTE The method by which a confirm primitive is correlated with its corresponding preceding request primitive is a local matter. See 1.2.			

Parameter description

S_address

This parameter specifies the station address of the source station.

MBX

This parameter specifies the mailbox.

D_address

This parameter specifies the station address of the destination station to allow slave to slave communication or communication beyond network boundaries using a virtual address.

Channel

This parameter specifies the communication channel.

Priority

This parameter specifies a communication priority.

Type

This parameter specifies the protocol type of the used mailbox service.

Cnt

This parameter specifies a service counter

DLS-user data

This parameter specifies the data that was read.

DL-status

This parameter specifies the result of the operation.

6 Local interactions

6.1 Read local

With this function the DL-user reads data from a memory area. Table 19 shows the primitives and parameter of this function.

Table 19 – Read local

DL-READ LOCAL Parameter name	Request	Confirm
	input	output
Memory area	M	
DLS-user data		U
DL-status		M

NOTE The method by which a confirm primitive is correlated with its corresponding preceding request primitive is a local matter. See 1.2.

Parameter description**Memory area**

This parameter specifies the memory area to be read.

DLS-user data

This parameter specifies the data that was read from the specified memory area.

DL-status

This parameter specifies the result of the operation.

6.2 Write local

With this function the DL-user writes data to a memory area. Table 20 shows the primitives and parameter of this function.

Table 20 – Write local

DL-WRITE LOCAL Parameter name	Request	Confirm
	input	output
Memory area	M	
DLS-user data	U	
DL-status		M

NOTE The method by which a confirm primitive is correlated with its corresponding preceding request primitive is a local matter. See 1.2.

Parameter description**Memory area**

This parameter specifies the memory area to be written.

DLS-user data

This parameter specifies the data to be written to the specified memory area.

DL-status

This parameter specifies the result of the operation.

6.3 Event local

With this function the DL-user gets an indication of an event. Table 21 shows the primitives and parameter of this function.

Table 21 – Event local

DL-EVENT LOCAL Parameter name	Indication
	output
Sync manager	C
Type	C

Parameter description

Sync manager

This parameter indicates the sync manager channel

Type

This parameter indicates the type of the event (read or write).

Bibliography

IEC 61158-1:2014, *Industrial communication networks – Fieldbus specifications – Part 1: Overview and guidance for the IEC 61158 and IEC 61784 series*

IEC 61158-2, *Industrial communication networks – Fieldbus specifications – Part 2: Physical layer specification and service definition*

IEC 61158-4-12, *Industrial communication networks – Fieldbus specifications – Part 4-12: Data-link layer protocol specification – Type 12 elements*

IEC 61158-5-12, *Industrial communication networks – Fieldbus specifications – Part 5-12: Application layer service definition – Type 12 elements*

IEC 61158-6-12, *Industrial communication networks – Fieldbus specifications – Part 6-12: Application layer protocol specification – Type 12 elements*

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

IEC 61784-1, *Industrial communication networks – Profiles – Part 1: Fieldbus profiles*

IEC 61784-2, *Industrial communication networks – Profiles – Part 2: Additional fieldbus profiles for real-time networks based on ISO/IEC 8802-3*

ISO/IEC/TR 8802-1, *Information technology – Telecommunications and information exchange between systems – Local and metropolitan area networks – Specific requirements – Part 1: Overview of Local Area Network Standards*

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 Datagram Protocol*; available at <<http://www.ietf.org>>

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

SOMMAIRE

AVANT-PROPOS.....	44
INTRODUCTION.....	46
1 Domaine d'application	47
1.1 Généralités.....	47
1.2 Spécifications.....	47
1.3 Conformité	48
2 Références normatives.....	48
3 Termes, définitions, symboles, abréviations et conventions	48
3.1 Termes et définitions du modèle de référence	48
3.2 Termes et définitions de convention pour les services	49
3.3 Termes et définitions pour les services de liaison de données.....	50
3.4 Symboles et abréviations	54
3.5 Conventions communes	55
4 Concepts et services de couche liaison de données	56
4.1 Principe de fonctionnement	56
4.2 Topologie	57
4.3 Vue générale de la couche liaison de données	57
4.4 Vue d'ensemble de la détection d'erreurs	58
4.5 Introduction à la gestion de données de processus et de paramètres	59
4.6 Modèle de référence des nœuds	59
4.7 Vue d'ensemble du fonctionnement	61
4.8 Adressage.....	63
4.9 Classification d'esclaves	65
4.10 Structure de la couche communication dans l'esclave	66
5 Services de communication	67
5.1 Vue d'ensemble.....	67
5.2 Services de lecture.....	68
5.3 Services Write (écriture).....	71
5.4 Services combinés de lecture/écriture	73
5.5 Services de réseau.....	78
5.6 Boîte à lettres.....	79
6 Interactions locales	84
6.1 Lecture locale.....	84
6.2 Écriture locale	85
6.3 Événement local.....	85

Figure 1 – Mapping des données logiques dans une trame Ethernet composée d'une seule DLPDU de Type 12.....	58
Figure 2 – Modèle de référence de la liaison de données de Type 12	60
Figure 3 – Segments en mode ouvert de Type 12	62
Figure 4 – Segments en mode direct de Type 12	62
Figure 5 – Vue d'ensemble du mode d'adressage	63
Figure 6 – Vue d'ensemble de l'unité de gestion de mémoire de bus de terrain	65
Figure 7 – Structuration en couches de communication	66
Figure 8 – Flot de primitives de services Type 12	68
Figure 9 – Séquence réussie d'écriture de boîte à lettres.....	80
Figure 10 – Séquence réussie de lecture de boîte à lettres	81
Tableau 1 – Lecture physique à incréments automatiques (APRD)	68
Tableau 2 – Lecture physique d'adresse configurée (FPRD)	69
Tableau 3 – Broadcast read (BRD)	70
Tableau 4 – Logical read (LRD)	70
Tableau 5 – Écriture physique d'auto-incrément (APWR)	71
Tableau 6 – Écriture physique d'adresse configurée (FPWR)	72
Tableau 7 – Broadcast write (BWR)	72
Tableau 8 – Logical write (LWR)	73
Tableau 9 – Lecture/écriture physique à incréments automatiques (APRW)	74
Tableau 10 – Lecture/écriture physique d'adresse configurée (FPRW).....	75
Tableau 11 – Broadcast read/write (BRW)	75
Tableau 12 – Logical read/write (LRW)	76
Tableau 13 – Lecture/multiples écritures physiques à incréments automatiques (ARMW).....	77
Tableau 14 – Lecture/multiples écritures physiques d'adresse configurée (FRMW).....	77
Tableau 15 – Fourniture de variables de réseau (PNV)	78
Tableau 16 – Mailbox write	81
Tableau 17 – Mise à jour de lecture de boîte à lettres	82
Tableau 18 – Lecture de boîte à lettres.....	83
Tableau 19 – Lecture locale	84
Tableau 20 – Écriture locale	85
Tableau 21 – Événement local.....	85

COMMISSION ÉLECTROTECHNIQUE INTERNATIONALE

RÉSEAUX DE COMMUNICATION INDUSTRIELS – SPÉCIFICATIONS DES BUS DE TERRAIN –

Partie 3-12: Définition des services de la couche liaison de données – Éléments de type 12

AVANT-PROPOS

- 1) La Commission Électrotechnique Internationale (CEI) est une organisation mondiale de normalisation composée de l'ensemble des comités électrotechniques nationaux (Comités nationaux de la CEI). La CEI a pour objet de favoriser la coopération internationale pour toutes les questions de normalisation dans les domaines de l'électricité et de l'électronique. A cet effet, la CEI – entre autres activités – publie des Normes internationales, des Spécifications techniques, des Rapports techniques, des Spécifications accessibles au public (PAS) et des Guides (ci-après dénommés "Publication(s) de la CEI"). Leur élaboration est confiée à des comités d'études, aux travaux desquels tout Comité national intéressé par le sujet traité peut participer. Les organisations internationales, gouvernementales et non gouvernementales, en liaison avec la CEI, participent également aux travaux. La CEI collabore étroitement avec l'Organisation Internationale de Normalisation (ISO), selon des conditions fixées par accord entre les deux organisations.
- 2) Les décisions ou accords officiels de la CEI concernant les questions techniques représentent, dans la mesure du possible, un accord international sur les sujets étudiés, étant donné que les Comités nationaux de la CEI intéressés sont représentés dans chaque comité d'études.
- 3) Les Publications de la CEI se présentent sous la forme de recommandations internationales et sont agréées comme telles par les Comités nationaux de la CEI. Tous les efforts raisonnables sont entrepris afin que la CEI s'assure de l'exactitude du contenu technique de ses publications; la CEI ne peut pas être tenue responsable de l'éventuelle mauvaise utilisation ou interprétation qui en est faite par un quelconque utilisateur final.
- 4) Dans le but d'encourager l'uniformité internationale, les Comités nationaux de la CEI s'engagent, dans toute la mesure possible, à appliquer de façon transparente les Publications de la CEI dans leurs publications nationales et régionales. Toutes divergences entre toutes Publications de la CEI et toutes publications nationales ou régionales correspondantes doivent être indiquées en termes clairs dans ces dernières.
- 5) La CEI elle-même ne fournit aucune attestation de conformité. Des organismes de certification indépendants fournissent des services d'évaluation de conformité et, dans certains secteurs, accèdent aux marques de conformité de la CEI. La CEI n'est responsable d'aucun des services effectués par les organismes de certification indépendants.
- 6) Tous les utilisateurs doivent s'assurer qu'ils sont en possession de la dernière édition de cette publication.
- 7) Aucune responsabilité ne doit être imputée à la CEI, à ses administrateurs, employés, auxiliaires ou mandataires, y compris ses experts particuliers et les membres de ses comités d'études et des Comités nationaux de la CEI, pour tout préjudice causé en cas de dommages corporels et matériels, ou de tout autre dommage de quelque nature que ce soit, directe ou indirecte, ou pour supporter les coûts (y compris les frais de justice) et les dépenses découlant de la publication ou de l'utilisation de cette Publication de la CEI ou de toute autre Publication de la CEI, ou au crédit qui lui est accordé.
- 8) L'attention est attirée sur les références normatives citées dans cette publication. L'utilisation de publications référencées est obligatoire pour une application correcte de la présente publication.
- 9) L'attention est attirée sur le fait que certains des éléments de la présente Publication de la CEI peuvent faire l'objet de droits de brevet. La CEI ne saurait être tenue pour responsable de ne pas avoir identifié de tels droits de brevets et de ne pas avoir signalé leur existence.

L'attention est attirée sur le fait que l'utilisation du type de protocole associé est restreinte par les détenteurs des droits de propriété intellectuelle. En tout état de cause, l'engagement de renonciation partielle aux droits de propriété intellectuelle pris par les détenteurs de ces droits autorise l'utilisation d'un type de protocole de couche avec les autres protocoles de couche du même type, ou dans des combinaisons avec d'autres types autorisées explicitement par les détenteurs des droits de propriété intellectuelle pour ce type.

NOTE Les combinaisons de types de protocoles sont spécifiées dans la CEI 61784-1 et la CEI 61784-2.

La Norme internationale CEI 61158-3-12 a été établie par le sous-comité 65C: Réseaux industriels, du comité d'études 65 de la CEI: Mesure, commande et automation dans les processus industriels.

Cette troisième édition annule et remplace la deuxième édition parue en 2010. Cette édition constitue une révision technique. Les principales modifications par rapport à l'édition précédente sont énumérées ci-dessous:

- améliorations rédactionnelles pour clarification.

Le texte de cette norme est issu des documents suivants:

FDIS	Rapport de vote
65C/759/FDIS	65C/769/RVD

Le rapport de vote indiqué dans le tableau ci-dessus donne toute information sur le vote ayant abouti à l'approbation de cette norme.

Cette publication a été rédigée selon les Directives ISO/CEI, Partie 2.

Une liste de toutes les parties de la série CEI 61158, publiées sous le titre général *Réseaux de communication industriels – Spécifications des bus de terrain*, peut être consultée sur le site web de la CEI.

Le comité a décidé que le contenu de cette publication ne sera pas modifié avant la date de stabilité indiquée sur le site web de la CEI sous "<http://webstore.iec.ch>" dans les données relatives à la publication recherchée. À cette date, la publication sera

- reconduite,
- supprimée,
- remplacée par une édition révisée, ou
- amendée.

IMPORTANT – Le logo "colour inside" qui se trouve sur la page de couverture de cette publication indique qu'elle contient des couleurs qui sont considérées comme utiles à une bonne compréhension de son contenu. Les utilisateurs devraient, par conséquent, imprimer cette publication en utilisant une imprimante couleur.

INTRODUCTION

La présente partie de la CEI 61158 est l'une d'une série produite pour faciliter l'interconnexion des composants d'un système d'automatisation. Elle est liée à d'autres normes de la série telle que définie par le modèle de référence des bus de terrain "à trois couches" décrit dans la CEI 61158-1.

Dans toute la série de normes relatives aux bus de terrain, le terme "service" se réfère à la capacité abstraite fournie par une couche du Modèle de référence de base de l'Interconnexion des systèmes ouverts (OSI) à la couche immédiatement supérieure. Ainsi, le service de la couche liaison de données défini dans la présente norme est un service architectural conceptuel, indépendant des divisions administratives et de mise en œuvre.

RÉSEAUX DE COMMUNICATION INDUSTRIELS – SPÉCIFICATIONS DES BUS DE TERRAIN –

Partie 3-12: Définition des services de la couche liaison de données – Éléments de type 12

1 Domaine d'application

1.1 Généralités

La présente partie de la CEI 61158 fournit les éléments communs pour les communications de messagerie de base à temps critique entre des appareils dans un environnement d'automatisation. Le terme "à temps critique" sert à représenter la présence d'une fenêtre temporelle, dans les limites de laquelle une ou plusieurs actions spécifiées sont tenues d'être parachevées avec un certain niveau défini de certitude. Le manquement à parachever les actions spécifiées dans les limites de la fenêtre temporelle risque d'entraîner la défaillance des applications qui demandent ces actions, avec le risque concomitant pour l'équipement, l'installation et éventuellement pour la vie humaine.

La présente norme définit de manière abstraite le service visible de l'extérieur fourni par la couche liaison de données de réseau de terrain Type 12 en termes

- a) des actions et événements primitifs du service;
- b) des paramètres associés à chaque action primitive et événement primitif, et la forme qu'ils prennent;
- c) de l'interrelation entre ces actions et événements, et leurs séquences valides.

Le but de la présente norme est de définir les services fournis à

- la couche application de réseau de terrain Type 12 au niveau de la frontière entre l'application et les couches liaison de données du modèle de référence de réseau de terrain;
- la gestion des systèmes au niveau de la frontière entre la couche liaison de données et la gestion des systèmes selon le modèle de référence de bus de terrain.

1.2 Spécifications

L'objectif principal de la présente norme est de spécifier les caractéristiques des services conceptuels d'une couche liaison de données qui sont adaptées à des communications en temps critique, et donc complètent le Modèle de référence de base en guidant le développement des protocoles de liaison de données pour les communications à temps critique. Un objectif secondaire est de fournir des trajets de migration à partir de protocoles de communications industrielles préexistants.

La présente spécification peut être utilisée comme la base pour les interfaces formelles de programmation à couche de liaison (DL-Programming-Interfaces). Néanmoins, elle n'est pas une interface de programmation formelle et il sera nécessaire pour toute interface de ce type de traiter de questions de mise en œuvre qui ne sont pas couvertes par la présente spécification, y compris

- a) les tailles et l'ordonnement des octets pour les divers paramètres de service à plusieurs octets, et
- b) la corrélation de primitives appariées "request-confirm" (c'est-à-dire: demande et confirmation) ou "indication-response" (c'est-à-dire: indication et réponse).

1.3 Conformité

La présente norme ne spécifie de mises en œuvre individuelles ou de produits individuels, ni n'impose de mise en œuvre d'entités de liaison de données au sein des systèmes d'automatisation industriels.

Il n'y a pas de conformité d'équipement à la présente norme de définition des services de couche liaison de données. Au contraire, la conformité est obtenue par une mise en œuvre du protocole de liaison de données correspondant qui satisfait aux services de couche liaison de données de Type 12 définis dans la présente norme.

2 Références normatives

Les documents suivants sont cités en référence de manière normative, en intégralité ou en partie, dans le présent document et sont indispensables pour son application. Pour les références datées, seule l'édition citée s'applique. Pour les références non datées, la dernière édition du document de référence s'applique (y compris les éventuels amendements).

NOTE Toutes les parties de la série CEI 61158, ainsi que la CEI 61784-1 et la CEI 61784-2 font l'objet d'une maintenance simultanée. Les références croisées à ces documents dans le texte se rapportent par conséquent aux éditions datées dans la présente liste de références normatives.

ISO/CEI 7498-1, *Technologies de l'information – Interconnexion de systèmes ouverts (OSI) – Modèle de référence de base: Le modèle de base*

ISO/CEI 7498-3, *Technologies de l'information – Interconnexion de systèmes ouverts (OSI) – Modèle de référence de base: Dénomination et adressage*

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

ISO/CEI 10731, *Technologies de l'information – Interconnexion de systèmes ouverts – Modèle de référence de base – Conventions pour la définition des services OSI*

IEEE 802.1D, *IEEE Standard for Local and metropolitan area networks – Media Access Control (MAC) Bridges*; disponible à l'adresse <<http://www.ieee.org>>

3 Termes, définitions, symboles, abréviations et conventions

Pour les besoins du présent document, les termes, définitions, symboles, abréviations et conventions suivants s'appliquent.

3.1 Termes et définitions du modèle de référence

La présente norme est basée en partie sur les concepts développés dans l'ISO/CEI 7498-1 et l'ISO/CEI 7498-3 et utilise les termes suivants définis à cet égard.

3.1.1	adresse de DL	[7498-3]
3.1.2	transmission en mode sans connexion de DL	[7498-1]
3.1.3	entités correspondantes (N)	[7498-1]
	entités correspondantes de DL (N=2)	
	entités correspondantes de Ph (N=1)	
3.1.4	transmission duplex DL	[7498-1]
3.1.5	entité (N)	[7498-1]
	entité de DL (N=2)	
	entité de Ph (N=1)	
3.1.6	couche (N)	[7498-1]
	couche DL (N=2)	
	couche Ph (N=1)	
3.1.7	gestion de couche	[7498-1]
3.1.8	entités homologues	[7498-1]
3.1.9	nom primitif	[7498-3]
3.1.10	protocole DL	[7498-1]
3.1.11	unité de données de protocole DL	[7498-1]
3.1.12	relais de DL	[7498-1]
3.1.13	réinitialisation	[7498-1]
3.1.14	adresse de DL en réponse	[7498-3]
3.1.15	acheminement	[7498-1]
3.1.16	segmentation	[7498-1]
3.1.17	service-(N)	[7498-1]
	service DL (N=2)	
	service Ph (N=1)	
3.1.18	point d'accès au service (N)	[7498-1]
	point d'accès au service DL (N=2)	
	point d'accès au service Ph (N=1)	
3.1.19	unité de données de service de DL	[7498-1]
3.1.20	transmission simplex de DL	[7498-1]
3.1.21	sous-système de DL	[7498-1]
3.1.22	gestion-systèmes	[7498-1]
3.1.23	utilisateur de DLS	[7498-1]
3.1.24	données d'utilisateur de DLS	[7498-1]
3.2	Termes et définitions de convention pour les services	

La présente norme utilise également les termes suivants définis dans l'ISO/CEI 10731 tels qu'ils s'appliquent à la couche liaison de données:

- 3.2.1 **acceptant**
- 3.2.2 **service asymétrique**
- 3.2.3 (primitive) "confirm";
(primitive) "requestor.deliver"
- 3.2.4 (primitive) "deliver"
- 3.2.5 **primitive de service DL;**
primitive
- 3.2.6 **fournisseur de service DL**
- 3.2.7 **utilisateur de service DL**
- 3.2.8 **fonctionnalité facultative d'utilisateur de DL**
- 3.2.9 (primitive) "indication";
(primitive) "acceptor.deliver"
- 3.2.10 (primitive) "request";
(primitive) "requestor.submit"
- 3.2.11 **demandeur**
- 3.2.12 **réponse (primitive);**
(primitive) "acceptor.submit"
- 3.2.13 (primitive) "submit"
- 3.2.14 **service symétrique**
- 3.3 **Termes et définitions pour les services de liaison de données**

3.3.1 **application**

fonction ou structure de données pour laquelle des données sont consommées ou produites

3.3.2 **objets d'applications**

classes d'objets multiples qui gèrent et assurent un échange de messages pendant le mode exécution à travers le réseau et à l'intérieur de l'appareil de réseau

3.3.3 **esclave de base**

appareil esclave qui prend en charge uniquement l'adressage physique des données

3.3.4 **bit**

unité d'information consistant en un 1 ou un 0

Note 1 à l'article: Il s'agit de la plus petite unité de données qui puisse être transmise

3.3.5 **client**

- 1) objet qui utilise les services d'un autre objet (serveur) pour accomplir une tâche
- 2) initiateur d'un message auquel un serveur réagit

3.3.6 **connexion**

liaison logique entre deux objets d'application au sein du même appareil ou dans des appareils différents

3.3.7 **cyclique**

relatif à des événements qui se répètent d'une manière régulière et répétitive

3.3.8 **contrôle de redondance cyclique** **CRC**

valeur résiduelle calculée à partir d'une matrice de données utilisée comme une signature représentative pour la matrice

Note 1 à l'article : L'abréviation «CRC» est dérivée du terme anglais développé correspondant «Cyclic Redundancy Check».

3.3.9 **donnée**

terme générique servant à se référer à toute information transportée sur un bus de terrain

3.3.10 **cohérence de données**

moyen pour une émission et un accès cohérents de l'objet de donnée d'entrée ou de sortie entre client et serveur et au sein du client et du serveur

3.3.11 **appareil**

entité physique connectée au réseau de terrain composée d'au moins un élément de communication (l'élément de réseau) et qui peut avoir un élément de commande et/ou un élément final (transducteur, actionneur, etc.)

3.3.12 **horloges réparties**

méthode permettant de synchroniser les esclaves et maintenir une base de temps globale

3.3.13 **segment DL** **liaison** **liaison locale**

simple sous-réseau de DL dans lequel toutes les éventuelles DLE connectées peuvent communiquer directement, sans intervention de relaying de DL, chaque fois que toutes ces DLE qui participent à une instance de communication sont simultanément attentives au sous-réseau de DL pendant la/les période(s) de communication tentée(s)

3.3.14 **erreur**

discordance entre une valeur ou un état calculé(e), observé(e) ou mesuré(e) et la valeur ou l'état spécifié(e) ou théoriquement correct(e)

3.3.15 **événement**

instance d'un changement de conditions

3.3.16 **unité de gestion de mémoire de réseau de terrain**

fonction qui établit une ou plusieurs correspondances entre les adresses logiques et la mémoire physique

3.3.17 **entité d'unité de gestion de mémoire de réseau de terrain**

élément simple de l'unité de gestion de mémoire de réseau de terrain: une correspondance entre un espace d'adresses logiques cohérent et un emplacement cohérent de la mémoire physique

3.3.18

trame

synonyme déconseillé de DLPDU

3.3.19

esclave complet

appareil esclave qui prend en charge à la fois l'adressage physique et l'adressage logique des données

3.3.20

interface

frontière partagée entre deux unités fonctionnelles, définies par les caractéristiques fonctionnelles, caractéristiques de signal ou autres caractéristiques selon le cas approprié

3.3.21

maître

appareil qui commande le transfert de données sur le réseau et lance l'accès des esclaves aux supports en envoyant des messages et qui constitue l'interface au système de commande

3.3.22

mapping

correspondance entre deux objets de telle manière que l'un des objets soit partie intégrante de l'autre objet

3.3.23

support

câble, fibre optique ou autre moyen par lequel des signaux de communication sont émis entre deux ou plusieurs points

Note 1 à l'article: "media" est utilisé comme pluriel de "medium".

3.3.24

message

série ordonnée d'octets censés acheminer de l'information

Note 1 à l'article: Normalement utilisé pour acheminer de l'information entre homologues au niveau de la couche application.

3.3.25

réseau

ensemble de nœuds reliés par un certain type de support de communication, notamment des répéteurs intermédiaires, ponts, routeurs et passerelles de couche inférieure

3.3.26

nœud

- a) simple entité de DL telle qu'elle apparaît sur une liaison locale
- b) point d'extrémité d'une liaison dans un réseau ou un point auquel deux ou plusieurs liaisons se rencontrent

[SOURCE: CEI 61158-2, 3.1.31 pour l'option b), avec un ajustement rédactionnel]

3.3.27**objet**

représentation abstraite d'un composant particulier au sein d'un appareil

Note 1 à l'article: Un objet peut être

- a) une représentation abstraite des capacités d'un appareil, constituée de tout ou partie des composants suivants:
 - 1) données (informations qui varient au fil du temps);
 - 2) configuration (paramètres pour le comportement);
 - 3) méthodes (choses qui peuvent être faites en utilisant les données et la configuration); ou
- b) un ensemble de données connexes (sous la forme de variables) et de méthodes (procédures) pour opérer sur les données en question qui ont une interface et un comportement clairement définis.

3.3.28**donnée de processus**

objet de donnée contenant des objets d'application destinés à être transférés de façon cyclique ou acyclique à des fins de traitement

3.3.29**utilisateur DLS destinataire**

Utilisateur de service de DL qui agit comme un destinataire de données d'utilisateur de DL

Note 1 à l'article: Un utilisateur de service DL peut être à la fois un utilisateur DLS expéditeur et destinataire.

3.3.30**utilisateur DLS expéditeur**

utilisateur de service de DL qui agit comme une source de données d'utilisateur de DL

3.3.31**serveur**

objet qui fournit des services à un autre objet (client)

3.3.32**service**

opération ou fonction qu'un objet et/ou une classe d'objets réalise ou assure à la demande d'un autre objet et/ou d'une autre classe d'objets

3.3.33**esclave**

entité de DL accédant au support uniquement après avoir été initiée par l'esclave précédent ou le maître

3.3.34**gestionnaire de synchronisation**

ensemble d'éléments de commande servant à coordonner l'accès à des objets utilisés simultanément

3.3.35**canal du gestionnaire de synchronisation**

simples éléments de commande servant à coordonner l'accès à des objets utilisés simultanément

3.3.36**commutateur**

pont MAC tel que défini dans l'IEEE 802.1D

3.4 Symboles et abréviations

APRD	Auto-increment physical read (Lecture physique à incréments automatiques)
APRW	Auto-increment physical read/write (Lecture/écriture physique à incréments auto)
APWR	Auto-increment physical write (Écriture physique à incréments automatiques)
ARMW	Auto-increment physical read/multiple write (Lecture/multiple écriture physique à incréments automatiques)
BRD	Broadcast read (Lecture en diffusion)
BRW	Broadcast read/write (Lecture/écriture en diffusion)
BWR	Broadcast write (Écriture en diffusion)
CAN	Controller area network (Gestionnaire de réseau de communication)
CoE	CAN application protocol over Type 12 services (Protocole d'application CAN sur)
CSMA/CD	Carrier sense multiple access with collision detection (Accès multiple par surveillance du signal et détection de collision)
DC	Distributed Clocks (Horloges distribuées)
DL-	Préfixe désignant la couche liaison de donnée
DLC	DL-connection (Connexion DL)
DLCEP	DL-connection-end-point (Extrémité de connexion DL)
DLE	DL-entity (Entité DL) (instance active locale de la couche de liaison de données)
DLL	DL-Layer (Couche DL)
DLPCI	DL-protocol-control-information (Informations de commande de protocole DL)
DLPDU	DL-protocol-data-unit (Unité de données de protocole DL)
DLM	DL-management (Gestion DL)
DLME	DL-management entity (Entité de gestion DL, instance active locale de la gestion)
DLMS	DL-management service (Service de gestion DL)
DLS	DL-service (Service DL) (
DLSAP	DL-service-access-point (Point d'accès au service DL)
DLSDU	DL-service-data-unit (Unité de données de service DL)
E ² PROM	Electrically erasable programmable read only memory (Mémoire morte à reprogrammation électrique)
EoE	Ethernet tunneled over Type 12 services (Services Ethernet en tunnel sur Type 1)
ESC	Type 12 slave controller (Appareil de commande d'esclave de Type 12)
FCS	Frame Check Sequence (Séquence de contrôle de trame)
FIFO	First-in first-out («Premier entré, premier sorti», méthode de mise en file d'attente)
FMMU	Fieldbus Memory Management Unit (Unité de gestion de mémoire de bus de terrain)
FoE	File access with Type 12 services (Accès fichier avec les services de Type 12)
FPRD	Configured address physical read (Lecture physique d'adresse configurée)
FPRW	Configured address physical read/write (Lecture/écriture physique d'adresse configurée)
FPWR	Configured address physical write (Écriture physique d'adresse configurée)
FRMW	Configured address physical read/multiple write (Lecture/multiple écriture physique d'adresse configurée)
HDR	Header (En-tête)
ID	Identifiant (Identificateur)
IP	Internet Protocol (Protocole Internet)
LAN	Local Area Network (Réseau local)
LRD	Logical memory read (Lecture de mémoire logique)
LRW	Logical memory read/write (Lecture/écriture de mémoire logique)
LWR	Logical memory write (Écriture de mémoire logique)
MAC	Medium Access Control (Commande d'accès au support)

MDI	Media-dependent interface (Interface dépendant du support, spécifiée dans l'ISC)
MDX	Mailbox data exchange (Échange de données de boîte à lettres)
MII	Media-independent interface (Interface indépendante du support, spécifiée dans l'ISC)
PDI	Physical device interface (Interface d'appareil physique, un jeu d'éléments qui permet l'accès à des services de DL depuis l'utilisateur de DL)
PDO	Process Data Object (Objet de données de processus)
Ph-	Préfixe désignant la couche physique
PhE	Ph-entity (Entité Ph) (instance active locale de la couche physique)
PhL	Ph-layer (couche Physique)
PHY	Physical layer device (Appareil de couche physique, spécifié dans l'ISO/CEI 8802-3)
PNV	Publish network variable (Variable réseau d'édition)
OSI	Open Systems Interconnection (Interconnexion des systèmes ouverts)
QoS	Quality of service (qualité de service)
RAM	Random access memory (Mémoire vive)
Rx	Receive (recevoir)
SDO	Service Data Object (Objet de données de service)
SII	Slave information interface (Interface d'information esclave)
SyncM	Synchronization manager (Gestionnaire de synchronisation)
TCP	Transmission Control Protocol (Protocole de commande de transport)
Tx	Transmit (émettre)
UDP	User Datagram Protocol (Protocole datagramme d'utilisateur)
WKC	Working counter (Compteur en fonction)

3.5 Conventions communes

La présente norme utilise les conventions descriptives données dans l'ISO/CEI 10731.

Le modèle de service, les primitives de service et les diagrammes de temps-séquence utilisés sont des descriptions totalement abstraites; ils ne constituent pas une spécification pour une mise en œuvre.

Les primitives de service, utilisées pour représenter les interactions utilisateur de service/fournisseur de service (voir ISO/CEI 10731), acheminent des paramètres qui indiquent les informations disponibles dans l'interaction entre utilisateur et fournisseur.

La présente norme utilise un format de tableau pour décrire les paramètres de composants des primitives de DLS. Les paramètres qui s'appliquent à chaque groupe de primitives de DLS sont consignés en tableaux dans toute la suite de la présente norme. Chaque tableau comporte jusqu'à six colonnes, contenant le nom du paramètre de service, et une colonne chacun pour les primitives et les sens de transfert de paramètres utilisés par le DLS:

- les paramètres d'entrée de la primitive "request" (demande);
- les paramètres de sortie de la primitive "indication";
- les paramètres d'entrée de la primitive "response" (réponse); et
- les paramètres de sortie de la primitive "confirm" (confirmation).

NOTE Les primitives "request", "indication", "response" et "confirm" sont aussi appelées respectivement primitives "requestor.submit", "acceptor.deliver", "acceptor.submit" et "requestor.deliver" (voir ISO/CEI 10731).

Un paramètre (ou une partie de celui-ci) est énuméré dans chaque rangée de chaque tableau. Dans les colonnes appropriées de la primitive de service, un code est utilisé pour spécifier le type d'usage du paramètre sur la primitive et le sens de paramètres spécifiés dans la colonne:

- M** Le paramètre est obligatoire pour la primitive;
 - U** Le paramètre est une option de l'utilisateur et peut ou peut ne pas être fourni, cela dépendant de l'usage dynamique de l'utilisateur de DLS. Lorsqu'il n'est pas fourni, une valeur par défaut est supposée pour le paramètre;
 - C** Le paramètre est conditionné à d'autres paramètres ou à l'environnement de l'utilisateur de DLS.
- (Blanc/Vide) Le paramètre n'est jamais présent;

Certaines entrées sont en plus qualifiées par des éléments entre parenthèses. Ceux-ci peuvent être une constante spécifique à un paramètre:

- (=) indique que le paramètre équivaut du point de vue de la sémantique au paramètre dans la primitive de service située immédiatement à sa gauche dans le tableau;

Dans n'importe quelle interface particulière, il n'est pas indispensable d'énoncer tous les paramètres de façon explicite. Certains peuvent être associés de façon implicite à la primitive.

Dans les diagrammes qui illustrent ces interfaces, des lignes tiretées indiquent des relations de cause à effet ou de temps-séquence tandis que les traits ondulés indiquent que des événements sont grosso modo contemporains.

4 Concepts et services de couche liaison de données

4.1 Principe de fonctionnement

La présente norme décrit une technologie Ethernet temps réel qui vise à maximiser l'utilisation de la bande passante Ethernet en "full duplex" (duplex intégral). La commande d'accès au support utilise le principe de maître/esclave selon lequel le nœud maître (typiquement le système de commande) envoie les trames Ethernet aux nœuds esclaves, qui extraient des données de ces trames et insèrent des données dans ces trames.

Du point de vue de l'Ethernet, un segment de Type 12 est un appareil Ethernet pris séparément qui reçoit et envoie des trames Ethernet de la norme ISO/CEI 8802-3. Cependant, cet appareil Ethernet n'est pas limité à un seul appareil de commande Ethernet avec microprocesseur aval, mais peut consister en un grand nombre d'appareils esclaves de Type 12. Ceux-ci traitent les trames Ethernet entrantes pendant qu'elles sont émises au sein de l'appareil, en lisant des données dans la trame Ethernet et/ou en y insérant leurs propres données avant de transférer la trame au prochain appareil esclave. Le dernier appareil esclave dans le segment renvoie la trame Ethernet complètement traitée en sens inverse à travers la chaîne d'appareils, retournant l'information recueillie par le truchement du premier appareil esclave au maître, qui la reçoit comme une trame de réponse Ethernet.

Cette procédure utilise la capacité "full duplex" (duplex intégral) de l'Ethernet: les deux sens de communication sont exploités indépendamment avec écriture et lecture par les esclaves sur le trajet sortant et seulement les mesures de temps dans le sens émission vers réception sur le trajet entrant à mesure que la trame Ethernet retransverse chaque appareil esclave intermédiaire.

La communication full-duplex entre un appareil maître et un segment de Type 12 constitué d'un ou plusieurs appareils esclaves peut être établie sans utiliser de commutateur.

4.2 Topologie

La topologie d'un système de communication est l'un des facteurs cruciaux pour l'application réussie dans le domaine de l'automatisation. La topologie a une influence considérable sur l'effort de câblage, les caractéristiques de diagnostic, les options de redondance et les caractéristiques de branchement à chaud et lecture.

La topologie en étoile communément utilisée pour l'Ethernet peut accroître l'effort de câblage et augmenter le coût en infrastructure. En particulier pour les applications d'automatisation, une topologie linéaire ou arborescente est souvent préférable.

L'agencement des nœuds esclaves représente un bus à boucle ouverte. À l'extrémité ouverte, l'appareil maître envoie des trames, soit directement, soit par le biais de commutateurs Ethernet; il les reçoit à l'autre extrémité après qu'elles ont été traitées par chaque appareil intermédiaire. Chaque trame Ethernet est relayée du premier nœud au suivant et de celui-ci à chaque autre nœud en série. Le dernier nœud retourne la trame Ethernet vers le maître en utilisant les capacités "full duplex" de l'Ethernet. La topologie qui en résulte est une ligne physique.

Des branches, qui en principe sont possibles n'importe où, peuvent être utilisées pour renforcer la structure linéaire en une forme de structure arborescente. Une structure arborescente prend en charge un câblage très simple; des branches individuelles, par exemple, peuvent se diviser en des armoires de commande ou modules de machine, alors que la ligne principale passe d'un module au prochain. Les branches sont possibles si un appareil comporte plus de deux ports. La présente norme permet jusqu'à deux liaisons de branchement en plus du jeu élémentaire de deux interfaces en série.

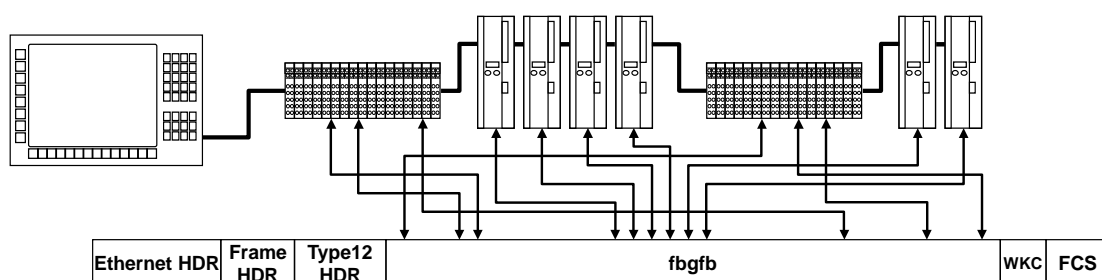
Une trame Ethernet reçue sur le port n (n différent de zéro) est transmise au port $n+1$. S'il n'y a aucun port $n+1$ la trame Ethernet est transmise au port 0. Si aucun périphérique n'est connecté ou si le port est fermé par le maître, une invitation à émettre (request to send) à ce port sera traitée comme si les mêmes données sont reçues par ce port (c'est-à-dire la boucle est fermée).

4.3 Vue générale de la couche liaison de données

Une trame Ethernet simple peut transporter plusieurs DLPDU de Type 12, qui sont mises en blocs dans la trame Ethernet sans trous entre elles. Plusieurs nœuds peuvent être adressés individuellement par ces DLPDU. La trame Ethernet est terminée par la dernière DLPDU de Type 12, excepté lorsque la taille de la trame est inférieure à 64 octets, auquel cas la trame Ethernet est bourrée pour atteindre 64 octets.

Cette mise en blocs conduit à une meilleure utilisation de la bande passante Ethernet que ne le feraient des trames Ethernet séparées à destination et en provenance de chaque nœud esclave. Cependant, par exemple pour un nœud d'entrée numérique à deux voies avec juste deux bits de données utilisateur, le surdébit d'une seule DLPDU de Type 12 peut encore être excessif.

Par conséquent, les nœuds esclaves peuvent aussi prendre en charge le mapping d'adresses logiques. Les données de processus peuvent être insérées n'importe où dans l'espace d'adresses logiques. Si une DLPDU de Type 12 est envoyée qui contient des services de lecture ou écriture pour une certaine zone de mémoire image située à l'adresse logique correspondante, au lieu d'adresser un nœud particulier, les nœuds insèrent les données à ou extraient les données de leur/leurs location(s) appropriée(s) dans les données de processus, comme noté à la Figure 1.



Légende

Anglais	Français
Ethernet HDR	En-tête Ethernet
Frame HDR	En-tête de trame
Type 12 HDR	En-tête de type 12
Process data	Données de processus
WKC	Working counter (compteur en fonction)
FCS	Frame Check Sequence (Séquence de contrôle de trame)

Figure 1 – Mapping des données logiques dans une trame Ethernet composée d'une seule DLPDU de Type 12

Chaque nœud qui détecte une concordance d'adresse avec l'image de processus insère sa donnée et, de ce fait, plusieurs nœuds peuvent être adressés simultanément par une seule DLPDU de Type 12. Le maître peut assembler une image de processus logique complètement triée via une seule DLPDU de Type 12, et ce, quel que soit l'ordre de câblage physique des appareils esclaves.

Un mapping supplémentaire n'est plus nécessaire dans le maître et, donc, les données de processus peuvent être transférées directement vers une ou plusieurs tâches de commande différentes. Chaque tâche peut créer sa propre image de processus et l'échanger dans les limites de son propre créneau de temps. L'ordre physique des nœuds est complètement arbitraire et n'est pertinent que pendant la première phase d'initialisation.

L'espace d'adresses logiques est de 2^{32} octets (= 4 Go). Ainsi, un bus de terrain de Type 12 peut être considéré comme étant un fond de panier sériel pour systèmes d'automatisation qui permet la connexion à des données de processus distribuées pour des appareils d'automatisation aussi bien de grandes dimensions que de très petites dimensions. Avec l'aide contrôleur Ethernet standard et des câbles Ethernet standard, un très grand nombre de voies E/S peuvent être raccordées à des appareils d'automatisation et il sera possible d'y avoir accès avec une bande passante élevée, un retard minimal et un débit de données utilisables quasi optimal. Dans le même temps, des appareils tels que scanners de bus de terrain peuvent également être raccordés, ce qui préserve les technologies et normes existantes.

4.4 Vue d'ensemble de la détection d'erreurs

Les nœuds maîtres et esclaves de Type 12 (les DLE) vérifient la séquence de contrôle de trames (FCS) Ethernet afin de déterminer si une trame est reçue correctement ou non. Sachant qu'un ou plusieurs esclaves peuvent modifier la trame au cours du transfert, la séquence FCS est vérifiée par chaque nœud à la réception et recalculée au cours de la retransmission. Si un esclave détecte une erreur de somme de contrôle, l'esclave ne répare pas la FCS, mais signale ce fait au maître en incrémentant un compteur d'erreurs. De ce fait, la source d'une seule faute peut être localisée de façon précise au sein de la topologie de boucle ouverte.

Au moment d'écrire des données dans une DLPDU de Type 12 ou d'en lire, l'esclave adressé incrémente un compteur en fonction (WKC) placé à la fin de la DLPDU. Les esclaves qui ne font que transmettre la DLPDU, sans en extraire d'information ou sans y insérer d'information, ne modifient pas le compteur. En comparant le compteur en fonction avec le nombre prévu de nœuds esclaves ayant accès, un maître peut vérifier si le nombre prévu de nœuds a traité la DLPDU correspondante.

4.5 Introduction à la gestion de données de processus et de paramètres

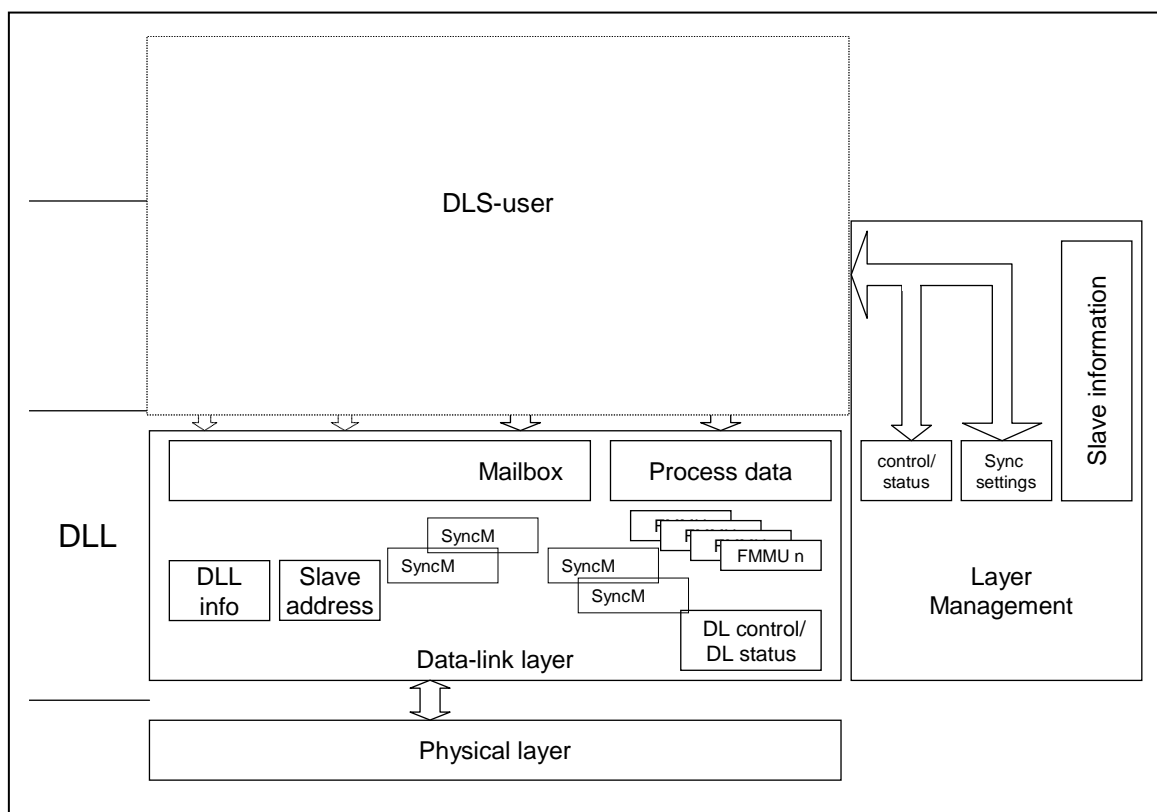
Les systèmes de communication industriels ont besoin de satisfaire à différentes exigences en termes de leurs caractéristiques de transmission de données. Les données de paramètres peuvent être transférées de manière acyclique et en grandes quantités, habituellement dans des situations où les exigences de temps sont relativement non critiques et la transmission est déclenchée par le système de commande. Les données de diagnostic sont également transférées de manière acyclique dans un mode événementiel, mais les exigences de temps sont plus drastiques et la transmission est habituellement déclenchée par un appareil périphérique.

D'autre part, les données de processus sont typiquement transférées de manière cyclique avec des temps de cycle différents. Les exigences de temps sont le plus drastiques pour la communication de données de processus. La présente Norme internationale prend en charge une diversité de services et de protocoles pour satisfaire à ces exigences différentes.

4.6 Modèle de référence des nœuds

4.6.1 Mapping avec le modèle de référence de base OSI

Les services de Type 12 sont décrits à l'aide des principes, méthodologie et modèle de l'ISO/CEI 7498-1 (OSI). Le modèle OSI fournit une approche stratifiée aux normes de communications et, par ce biais, des couches peuvent être mises au point et modifiées de manière indépendante. La spécification de Type 12 définit la fonctionnalité de haut en bas d'une pile de communications complète de l'OSI. Les fonctions des couches OSI intermédiaires, couches 3 à 6, sont consolidées soit en couche liaison de données de Type 12, soit en utilisateur de DL de la couche liaison de données de Type 12. Le modèle de référence de la liaison de données de Type 12 est montré à la Figure 2.



Légende

Anglais	Français
DLS-user	Utilisateur de DLS
Mailbox	Boîte à lettres (Boîte d'échange)
Process data	Données de processus
control/status	commande/statut
Sync settings	Réglages de synchronisation
Slave information	Informations esclave
DLL info	Informations de couche liaison de données
DLL	Data Link Layer (Couche liaison de données)
Slave address	Adresse d'esclave
SyncM	Gestionnaire de synchronisation
FMMU n	N ^{ième} Fieldbus Memory Management Unit (Unité de gestion de mémoire de réseau de terrain)
Data Link Layer	Couche liaison de données
DL control/ DL status	Commande de DL /statut de DL
Layer management	Gestion de couche
Physical layer	Couche physique

Figure 2 – Modèle de référence de la liaison de données de Type 12

4.6.2 Caractéristiques de la couche liaison de données

La couche liaison de données fournit un support de base à temps critique pour les communications de données parmi des appareils connectés. Le terme "à temps critique" sert à décrire des applications ayant une fenêtre temporelle, dans les limites de laquelle une ou

plusieurs actions spécifiées sont tenues d'être parachevées avec un certain niveau défini de certitude. Le manquement à parachever les actions spécifiées dans les limites de la fenêtre temporelle risque d'entraîner la défaillance des applications qui demandent ces actions, avec le risque concomitant pour l'équipement, l'installation et éventuellement pour la vie humaine.

La couche liaison de données a la tâche de calculer, comparer et générer la séquence de contrôle de trame et fournir des communications en extrayant des données dans la trame Ethernet et/ou en y incluant des données. Cela se fait en fonction des paramètres de la couche liaison de données qui sont stockés en des emplacements mémoire prédéfinis. Les données sont rendues disponibles à l'utilisateur DL en mémoire physique, soit dans une configuration de boîte à lettres, soit au sein d'une section de données de processus.

4.7 Vue d'ensemble du fonctionnement

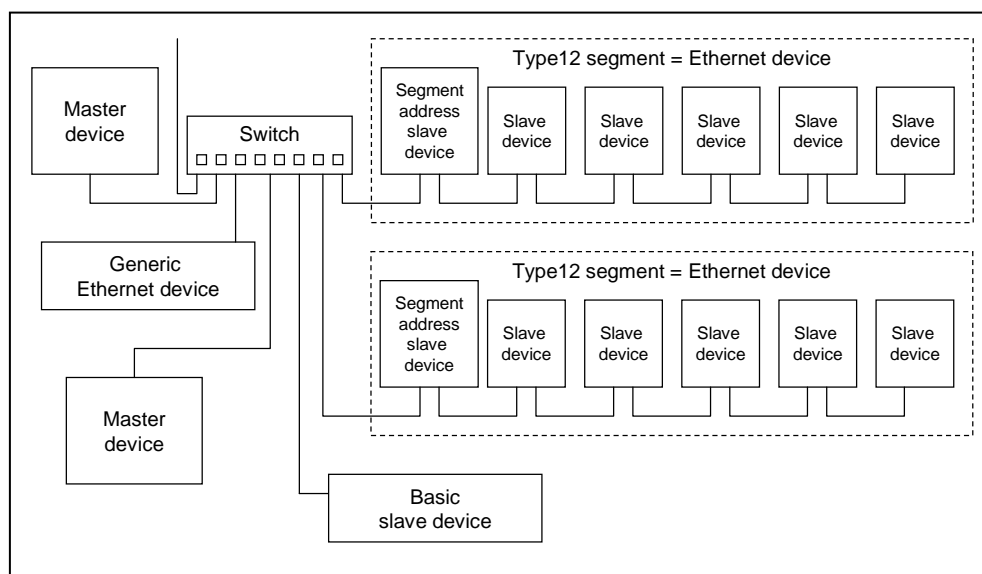
4.7.1 Relation à l'ISO/CEI 8802-3

La présente partie spécifie les services de couche liaison de données en plus de ceux spécifiés dans l'ISO/CEI 8802-3.

4.7.2 Modes de Type 12

4.7.2.1 Mode ouvert

Dans le mode ouvert, un ou plusieurs segments de Type 12 peuvent être connectés à un appareil de commutation standard comme montré à la Figure 3. Le premier appareil esclave au sein du segment de Type 12 a alors une adresse MAC selon l'ISO/CEI 8802-3 représentant le segment total. Cet appareil esclave à adresse de segment remplace le champ adresse de destination par le champ adresse de source et le champ adresse de source par sa propre adresse MAC au sein de la trame Ethernet si la trame suit les règles de codage du Type 12. Si ce type de trame est transporté via UDP, cet appareil gèrera les adresses IP de source et de destination de la même façon que les adresses MAC et les numéros de port de source et de destination de l'UDP afin d'apporter l'assurance que la trame de réponse satisfait pleinement aux normes des protocoles UDP/IP. De plus, cet appareil protège les esclaves au sein du segment contre l'accès non autorisé des appareils maîtres ou des appareils Ethernet génériques.



Légende

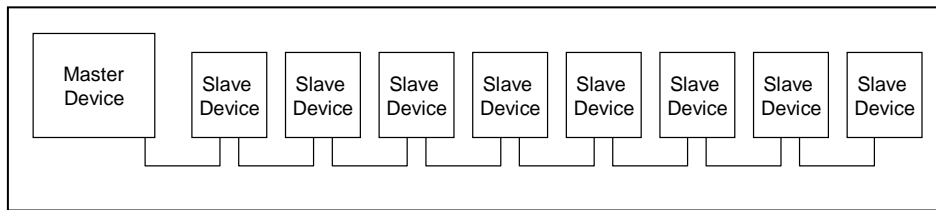
Anglais	Français
Master Device	Appareil maître

Anglais	Français
Switch	Commutateur
Generic Ethernet device	Appareil Ethernet générique
Segment address slave device	Adresse de segment appareil esclave
Slave device	Appareil esclave
Basic slave device	Appareil esclave de base
Type 12 segment = Ethernet device	Segment de Type 12 = Appareil Ethernet

Figure 3 – Segments en mode ouvert de Type 12

4.7.2.2 Mode direct

Dans le mode direct, un segment de Type 12 est connecté au port Ethernet standard de l'appareil de contrôle ou de l'appareil maître comme il est montré à la Figure 4. Les champs d'adresses MAC des trames Ethernet ne sont pas vérifiés.



Légende

Anglais	Français
Master Device	Appareil maître
Slave Device	Appareil esclave

Figure 4 – Segments en mode direct de Type 12

4.7.3 Topologie logique

En termes logiques, l'agencement des esclaves au sein d'un segment de Type 12 représente un bus connecté comme une boucle "full-duplex" ouverte. En l'extrémité d'entrée de la boucle ouverte, l'appareil maître insère des trames Ethernet, directement ou via des commutateurs Ethernet standard, et les reçoit en l'extrémité de sortie de la boucle ouverte après qu'elles ont été traitées par tous les appareils esclaves. Toutes les trames sont relayées du premier appareil esclave vers le suivant. Le dernier appareil esclave retourne la trame, en passant par tous les autres appareils esclaves, vers le maître. Le résultat est une boucle logique ouverte réalisée par des segments consécutifs de la ligne physique full-duplex.

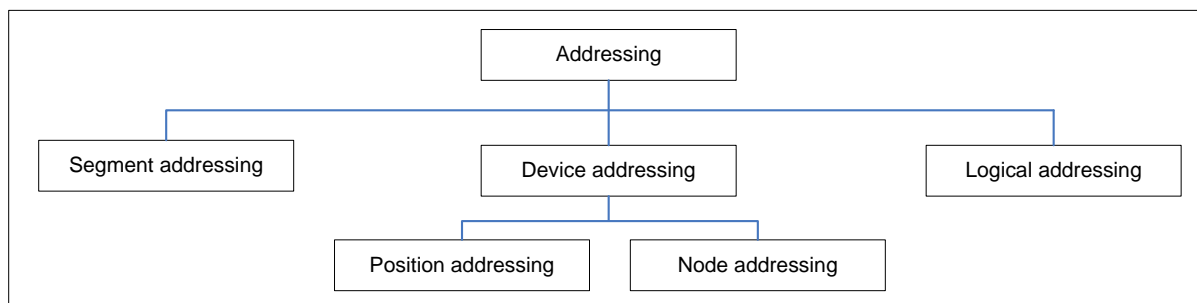
Les trames Ethernet reçues sont traitées octet par octet "à la volée" par les appareils esclaves selon leur séquence physique au sein de la structure de boucle ouverte. Dans ce cas, chaque appareil esclave reconnaît les commandes pertinentes et les exécute en conséquence alors que les trames (retardées par une constante de temps, typiquement en dessous de 1 µs) sont transmises au prochain appareil dans la boucle ouverte. L'extraction et l'insertion de données sont accomplies par la couche liaison de données à mesure que la trame Ethernet transite par l'appareil esclave, d'une manière qui est indépendante des temps de réponse des éventuels microprocesseurs présents (ou connectés) au sein de l'appareil esclave.

Les branches physiques "full-duplex" sont possibles en n'importe quel endroit du segment de Type 12, car une branche ne brise pas la boucle logique. Les branches peuvent être utilisées pour bâtir une structure arborescente flexible, ce qui permet un câblage très simple.

4.8 Adressage

4.8.1 Vue d'ensemble de l'adressage

Différents modes d'adressage sont pris en charge pour les esclaves, comme noté à la Figure 5. L'en-tête dans la DLPDU de Type 12 contient une adresse 32 bits qui est utilisée pour l'adressage de nœud physique ou l'adressage logique.



Légende

Anglais	Français
Addressing	Adressage
Segment addressing	Adressage de segment
Device addressing	Adressage d'appareil
Logical addressing	Adressage logique
Position addressing	Adressage de position
Node addressing	Adressage de nœud

Figure 5 – Vue d'ensemble du mode d'adressage

4.8.2 Adressage de segment

Les adresses MAC selon l'ISO/CEI 8802-3 sont utilisées pour l'adressage de segment.

4.8.3 Adressage d'appareil

4.8.3.1 Structure de l'adresse d'appareil

Avec ce mode d'adressage, une adresse 32 bits au sein de chaque DLPDU de Type 12 est divisée en une adresse d'appareil esclave 16 bits et une adresse physique 16 bits dans l'appareil esclave, ce qui donne 2^{16} adresses d'appareil esclave, chacune avec un espace associé d'adresses locales 16 bits. Avec l'adressage d'appareil, chaque DLPDU de Type 12 adresse de manière univoque un seul appareil esclave.

Ce mode est le plus approprié pour transférer des données de paramètre. Il existe deux mécanismes différents d'adressage d'appareil comme suit:

- adressage de position;
- adressage de nœud.

4.8.3.2 Adressage de position

L'adressage de position est utilisé pour adresser chaque appareil esclave via sa position physique au sein du segment. Chaque appareil esclave incrémente le champ adresse 16 bits à mesure que la DLPDU transite par l'appareil esclave; l'appareil qui reçoit une DLPDU avec un champ adresse de valeur 0 est celui qui est adressé. En raison du mécanisme employé pour mettre à jour cette adresse lors du passage par le nœud, l'adresse d'appareil esclave dans l'adressage de position est appelée *adresse à incréments automatiques*.

EXEMPLE Si le 10e appareil esclave dans le segment est à adresser, l'appareil maître envoie une DLPDU avec adressage de position avec une valeur de départ d'adresse d'appareil de -9, qui est incrémentée de 1 par chaque appareil par lequel la DLPDU transite.

Dans la pratique, l'adressage de position est utilisé au cours de la phase de démarrage, pendant laquelle le maître attribue des adresses de nœud configurées aux esclaves, après quoi ceux-ci peuvent être adressés quelle que soit leur position physique dans le segment par le biais de l'utilisation de ces adresses de nœud.

Ce mécanisme d'adressage basé sur la topologie a l'avantage en ce qu'il n'est pas nécessaire d'établir manuellement les adresses de nœud au niveau des esclaves.

4.8.3.3 Adressage de nœud

Avec l'adressage de nœud, les esclaves sont adressés via des adresses de nœud configurées attribuées par le maître pendant la phase de démarrage de la liaison de données. Cela assure que, même si la topologie de segment change ou des appareils sont ajoutés ou retirés, les appareils esclaves peuvent encore être adressés via la même adresse configurée.

L'adresse d'appareil esclave en adressage de nœud est appelée *adresse de poste configurée*.

4.8.4 Adressage logique

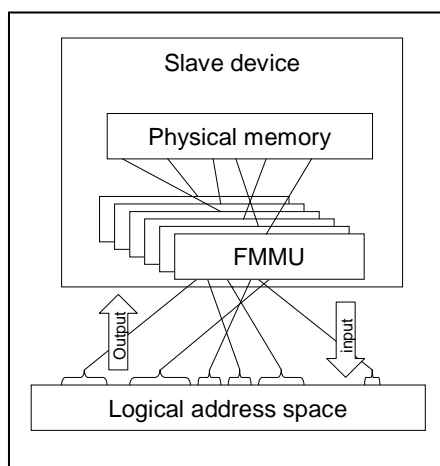
Pour l'adressage logique au sein d'un segment, le champ adresse 32 bits complet de chaque DLPDU de Type 12 est utilisé comme une seule adresse non structurée. Avec l'adressage local, les esclaves ne sont pas adressés individuellement, mais au contraire c'est une section de l'espace d'adresses logiques de 4 Go à l'échelle du segment qui est adressée. N'importe quel nombre d'esclaves peut utiliser les mêmes sections ou des sections qui se chevauchent.

L'adresse de la région de données utilisée dans ce mode est appelée *adresse logique*.

Le mode d'adressage logique est particulièrement adapté pour transférer et/ou échanger des données de processus cycliques.

4.8.5 Introduction des FMMU

Les unités de gestion de mémoire de bus de terrain (FMMU) gèrent l'attribution des adresses de la mémoire physique esclave aux adresses à l'échelle du segment logique, comme montré à la Figure 6.



Légende

Anglais	Français
Slave device	Appareil esclave

Anglais	Français
Physical memory	Mémoire physique
FMMU	FMMU
Logical address space	Espace d'adresses logiques
Type 12 segment = Ethernet device	Segment de Type 12 = Appareil Ethernet
Input	Entrée
Output	Sortie

Figure 6 – Vue d'ensemble de l'unité de gestion de mémoire de bus de terrain

La configuration des entités FMMU est accomplie par l'appareil maître et transférée aux appareils esclaves pendant la phase de démarrage de la liaison de données. Pour chaque entité FMMU, les éléments suivants sont configurés: une adresse logique de départ orientée bit, une adresse mémoire physique de départ, une longueur de bits et un type qui spécifie le sens du mapping (entrée ou sortie). Toute donnée au sein de la mémoire d'un appareil esclave peut ainsi être mise en correspondance («mappée») bit par bit avec n'importe quelle adresse logique.

Lorsqu'une DLPDU de Type 12 avec adressage logique est reçue, l'appareil esclave vérifie si l'une de ses entités FMMU montre une concordance d'adresse. Si cela s'avère approprié, il insère des données à la position associée du champ de données dans la DLPDU (de type entrée) ou extrait des données de la position associée de la DLPDU (de type sortie). Les DLPDU peuvent donc être assemblées en toute flexibilité et optimisées selon les exigences de l'application de commande.

4.8.6 Introduction du gestionnaire de synchronisation

Le gestionnaire de synchronisation (SyncM) commande l'accès à la mémoire d'utilisateur de DLS. Chaque voie du SyncM définit une zone cohérente de la mémoire d'utilisateur de DLS.

4.9 Classification d'esclaves

4.9.1 esclave complet

Une distinction est faite entre les esclaves complets (qui prennent en charge tous les modes d'adressage) et les esclaves de base (qui prennent en charge uniquement un sous-ensemble des modes d'adressage). Les appareils maîtres peuvent prendre en charge la fonctionnalité d'esclave de base afin de permettre la communication directe avec un autre appareil maître. Il convient que les appareils esclaves prennent en charge la fonctionnalité d'esclave complet.

Un esclave complet prend en charge

- l'adressage logique,
- adressage de position; et
- adressage de nœud.

Ainsi, les appareils esclaves complets ont besoin à la fois d'une FMMU et de la fonctionnalité d'adressage à incréments automatiques.

Les esclaves complets peuvent prendre en charge l'adressage de segment. Les esclaves complets qui prennent en charge l'adressage de segment sont appelés "appareils esclaves à adressage de segment".

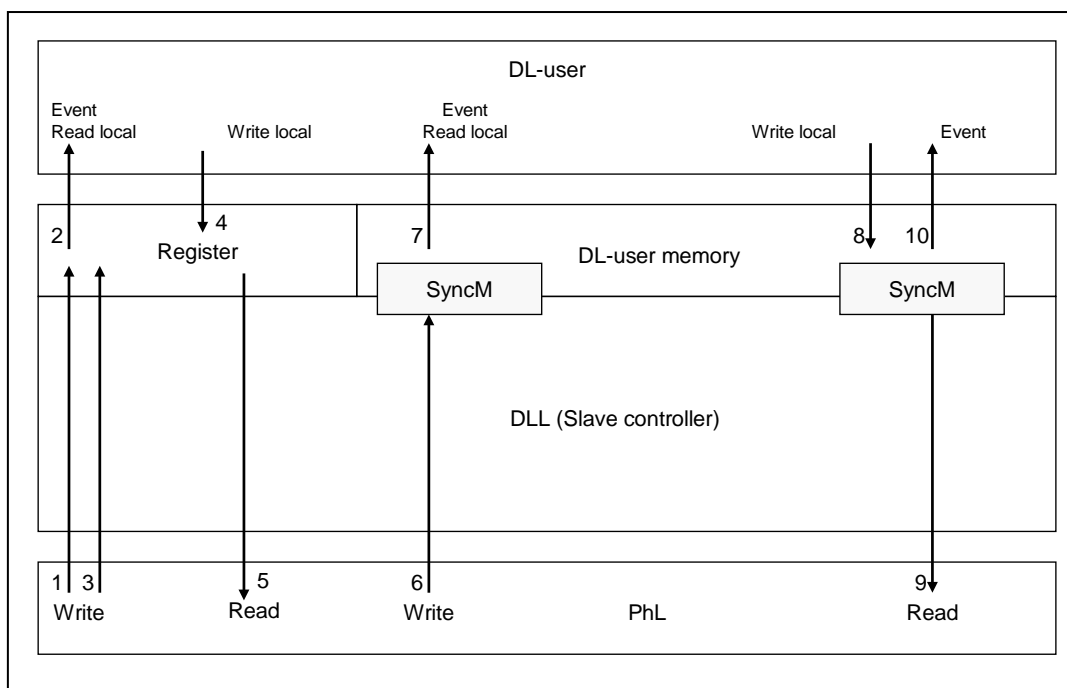
Seuls les esclaves complets peuvent être connectés au sein d'un segment de Type 12.

4.9.2 esclave de base

Les appareils esclaves de base prennent en charge l'adressage de nœud et l'adressage de segment.

4.10 Structure de la couche communication dans l'esclave

Les attributs sont relatifs à la mémoire physique d'un esclave, qui peut être lue ou écrite depuis le maître. La mémoire physique est constituée de registres et de mémoire d'utilisateur de DL. La zone registre contient des informations pour la configuration, la gestion et l'identification des appareils dans la DLL. L'utilisation de la mémoire d'utilisateur de DL est définie par l'utilisateur de DL. La Figure 7 montre le contour des interactions entre utilisateur DL-et DLL et entre DLL et communication.



Légende

Anglais	Français
DL-user	Utilisateur de DL
Event Read local	Lecture locale d'événement
Write local	Écriture locale
Event	Événement
Register	Registre
DL-user memory	Mémoire utilisateur de DL
SyncM	Gestionnaire de synchronisation
DLL (Slave controller)	DLL (contrôleur esclave)
Write	Écriture
Read	Lecture
PhL	Couche physique

Figure 7 – Structuration en couches de communication

Un service d'écriture de DL à la zone registre (1) peut (en fonction du registre écrit) se traduire par une primitive d'indication d'événement à destination de l'utilisateur de DL, suivie d'une primitive de demande locale de lecture en provenance de l'utilisateur de DL pour

recupérer la valeur écrite (2). Autrement, le service d'écriture de DL accédera uniquement à la zone registre sans en informer l'utilisateur de DL (3). L'utilisateur de DL peut à tout moment lire la zone registre avec la primitive locale de lecture.

L'utilisateur de DL établira le registre avec l'écriture locale de registre et le mettra à jour si cela est nécessaire et possible (4). Un service de lecture de DL à la zone registre accédera uniquement à la zone registre sans en informer l'utilisateur de DL (5).

L'accès à la zone mémoire d'utilisateur de DL est coordonné par le gestionnaire de synchronisation. L'accès sans le gestionnaire de synchronisation peut être réalisé d'une manière similaire à l'accès au registre, mais cette méthode d'utilisation peut être limitée par des contraintes de cohérence et l'absence d'événements indiquant des modifications apportées par le maître. La description de l'accès à la mémoire d'utilisateur de DL suppose l'utilisation du gestionnaire de synchronisation.

Un service d'écriture de DL à la zone mémoire d'utilisateur de DL (6) se traduira par une primitive d'indication d'événement à destination de l'utilisateur de DL, suivie d'une primitive de demande locale de lecture en provenance de l'utilisateur de DL pour récupérer les valeurs écrites (7).

L'utilisateur de DL écrira dans la zone mémoire d'utilisateur de DL avec une primitive de demande locale d'écriture (8). La zone mémoire d'utilisateur de DL sera lue par le maître avec le service lecture de DL (9) qui émettra une primitive d'indication d'événement (10) pour signaler que la zone mémoire d'utilisateur de DL a été lue et peut être de nouveau écrite par l'utilisateur de DL.

Un esclave répond à toutes les demandes de lecture et d'écriture et peut répondre à des demandes combinées de lecture/écriture.

5 Services de communication

5.1 Vue d'ensemble

Les services sont décrits du point de vue du maître. L'exécution du service au sein de l'esclave est décrite à l'Article 6.

La couche liaison de données spécifie les services pour la lecture, l'écriture et l'échange (lecture immédiatement suivie d'un écrasement) de données à partir de la mémoire physique au sein de l'esclave.

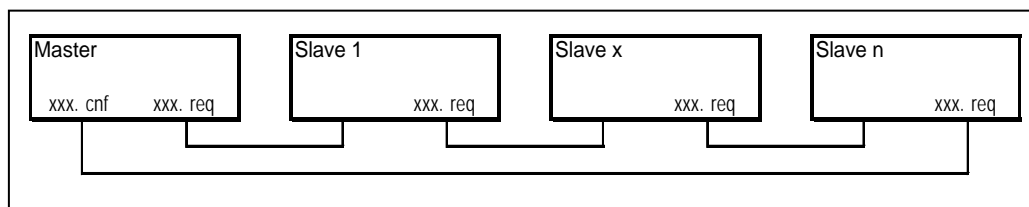
NOTE Par simplification, l'expression "lit dans la mémoire" est utilisée à la place de "lit les données dans la mémoire physique". De manière équivalente, l'expression "écrit dans la mémoire" est utilisée au lieu de "écrit les données dans la mémoire physique".

Avec le service de lecture, un maître lit dans les registres ou dans la mémoire d'utilisateur de DL d'un ou plusieurs esclaves.

La procédure de service de base de toutes les variantes du service lecture est la même excepté la lecture en diffusion. L'unité adressée copiera les données dans le paramètre de données. Avec le service diffusion, l'esclave exécutera une opération OU logique sur les bits des données de paramètre avec les données de la mémoire ou du registre.

Si un seul esclave est connecté au maître, la procédure de service est exécutée conformément au modèle client-serveur. Si plusieurs esclaves sont connectés (toujours en série), l'invocation de la procédure de service est gérée de telle manière que la sortie produite par un esclave serve d'entrée injectée dans l'esclave suivant.

Les procédures de service sont semblables aux procédures définies dans l'IEEE 802.1D mais le transfert et le traitement sont combinés. Sachant que le Type 12 utilise des services confirmés au lieu de services non confirmés tels que spécifiés dans l'IEEE 802.1D, le flot d'informations entre les primitives de service est adapté à cette situation. Le maître déclenche un service de demande et reçoit une confirmation correspondante. Chaque esclave reçoit une indication des données qu'il reçoit, tout en transmettant au prochain esclave les données en question, après une éventuelle mise à jour. La Figure 8 montre le flot de contrôle.



Légende

Anglais	Français
Master	Maître
Slave 1	Esclave 1
Slave x	Esclave x
Slave n	Esclave n

Figure 8 – Flot de primitives de services Type 12

5.2 Services de lecture

5.2.1 Vue d'ensemble

Avec les services de lecture, un maître lit des données dans la mémoire d'un ou plusieurs esclaves.

5.2.2 Lecture physique positionnelle (APRD)

Avec le service APRD, un maître lit des données dans la mémoire ou dans le registre d'un esclave sélectionné par l'ordre de classement physique de l'esclave dans le segment. Le Tableau 1 montre les primitives de service et le paramètre du service APRD.

Tableau 1 – Lecture physique à incréments automatiques (APRD)

DL-AUTOINCREMENT-PHYSICALREAD (LECTURE PHYSIQUE A INCREMENTS AUTOMATIQUES DE DL) Nom du paramètre	Request (demande)	Confirm (confirmation)
	Entrée	Sortie
Ordinal device number (numéro ordinal de l'appareil)	M	M
Device data area (zone Données de l'appareil)	M	
DLS-user data (données d'utilisateur de DLS)	U	U
Working counter (Compteur en fonction)	M	M
NOTE La méthode par laquelle une primitive "confirm" est corrélée à sa primitive "request" précédente correspondante relève d'une initiative locale. Voir 1.2.		

Description du paramètre

Ordinal device number (numéro ordinal de l'appareil)

Ce paramètre spécifie l'indice ordinal de l'appareil adressé dans la chaîne de communications câblées. La confirmation au maître contient le nombre d'appareils esclaves passés.

Device data area (zone Données de l'appareil)

Ce paramètre spécifie l'emplacement dans la mémoire physique de l'esclave où les données à lire sont stockées.

DLS-user data (données d'utilisateur de DLS)

À la confirmation, ce paramètre spécifie les données lues dans l'appareil si l'accès était valide au niveau de l'esclave adressé. Autrement, la valeur spécifiée par la demande est retournée inchangée.

Working counter (Compteur en fonction)

Ce paramètre est incrémenté si les données ont été lues avec succès.

5.2.3 Lecture physique d'adresse configurée (FPRD)

Avec le service FPRD, un maître lit des données dans la mémoire ou dans le registre d'un esclave sélectionné par l'adresse de poste configurée de l'esclave. Le Tableau 2 montre les primitives de service et le paramètre de service FPRD.

Tableau 2 – Lecture physique d'adresse configurée (FPRD)

DL-CONFIGURED-PHYSICALREAD (LECTURE PHYSIQUE D'ADRESSE CONFIGUREE DE DL) Nom du paramètre	Request (demande)	Confirm (confirmation)
	Entrée	Sortie
Configured device number (numéro d'appareil configuré)	M	
Device data area (zone Données de l'appareil)	M	
DLS-user data (données d'utilisateur de DLS)	U	U
Working counter (Compteur en fonction)	M	M
NOTE La méthode par laquelle une primitive "confirm" est corrélée à sa primitive "request" précédente correspondante relève d'une initiative locale. Voir 1.2.		

Description du paramètre

Configured device number (numéro d'appareil configuré)

Ce paramètre spécifie l'adresse de poste configurée de l'appareil adressé.

Device data area (zone Données de l'appareil)

Ce paramètre spécifie l'emplacement dans la mémoire physique de l'esclave où les données à lire sont stockées.

DLS-user data (données d'utilisateur de DLS)

À la confirmation, ce paramètre spécifie les données lues dans l'appareil si l'accès était valide au niveau de l'esclave adressé. Autrement, la valeur spécifiée par la demande est retournée inchangée.

Working counter (Compteur en fonction)

Ce paramètre est incrémenté si les données ont été lues avec succès.

5.2.4 Lecture de diffusion (BRD)

Avec le service BRD, un maître lit des données dans la zone mémoire physique ou dans le registre, ce qui sera une opération OU logique sur les bits entre les données entrantes et l'objet sélectionné au niveau de tous les esclaves. Le Tableau 3 montre les primitives de service et le paramètre de service BRD.

Tableau 3 – Broadcast read (BRD)

DL- BROADCAST-READ (LECTURE DE DIFFUSION DE DL) Nom du paramètre	Request (demande)	Confirm (confirmation)
	Entrée	Sortie
Broadcast address (adresse de diffusion)	M	M
Device data area (zone Données de l'appareil)	M	
DLS-user data (données d'utilisateur de DLS)	U	U
Working counter (Compteur en fonction)	M	M
NOTE La méthode par laquelle une primitive "confirm" est corrélée à sa primitive "request" précédente correspondante relève d'une initiative locale. Voir 1.2.		

Description du paramètre

Broadcast address (adresse de diffusion)

Ce paramètre est incrémenté au niveau de chaque esclave.

Device data area (zone Données de l'appareil)

Ce paramètre spécifie l'emplacement dans la mémoire physique où les données à lire sont stockées.

DLS-user data (données d'utilisateur de DLS)

À la confirmation, ce paramètre spécifie le résultat de logique sur les bits OU l'opération entre les données de paramètre de la demande et l'objet sélectionné au niveau de la réponse.

Working counter (Compteur en fonction)

Ce paramètre est incrémenté par tous les esclaves qui ont effectué l'opération OU logique au niveau du bit des données demandées.

5.2.5 Lecture logique (LRD)

Avec le service LRD, un maître lit des données dans la mémoire ou dans un registre d'un ou plusieurs esclaves sélectionnés par une adresse logique. Le Tableau 4 montre les primitives de service et le paramètre de service LRD.

Tableau 4 – Logical read (LRD)

DL-LOGICAL-READ (LECTURE LOGIQUE DE DL) Nom du paramètre	Request (demande)	Confirm (confirmation)
	Entrée	Sortie
Logical memory address (adresse mémoire logique)	M	
DLS-user data (données d'utilisateur de DLS)		U
Working counter (Compteur en fonction)	M	M
NOTE La méthode par laquelle une primitive "confirm" est corrélée à sa primitive "request" précédente correspondante relève d'une initiative locale. Voir 1.2.		

Description du paramètre

Logical memory address (adresse mémoire logique)

Ce paramètre spécifie l'adresse de début dans la mémoire logique où les données à lire sont situées.

DLS-user data (données d'utilisateur de DLS)

Ce paramètre spécifie les données qui ont été lues.

Working counter (Compteur en fonction)

Ce paramètre est incrémenté par tous les esclaves qui détectent une concordance d'adresses de la zone mémoire logique demandée.

5.3 Services Write (écriture)

5.3.1 Vue d'ensemble

Avec le service d'écriture, un maître écrit des données dans un registre ou dans la mémoire d'un ou plusieurs esclaves.

5.3.2 Écriture physique à une adresse définie par la position (APWR)

Avec le service APWR, un maître écrit des données dans la mémoire ou dans le registre d'un esclave sélectionné par l'ordre de classement physique de l'esclave dans le segment. Le Tableau 5 montre les primitives de service et le paramètre de service APWR.

Tableau 5 – Écriture physique d'auto-incrément (APWR)

DL-AUTOINCREMENT-PHYSICALWRITE (ÉCRITURE PHYSIQUE A INCREMENTS AUTOMATIQUES DE DL) Nom du paramètre	Request (demande)	Confirm (confirmation)
	Entrée	Sortie
Ordinal device number (numéro ordinal de l'appareil)	M	M
Device data area (zone Données de l'appareil)	M	
DLS-user data (données d'utilisateur de DLS)	U	
Working counter (Compteur en fonction)	M	M
NOTE La méthode par laquelle une primitive "confirm" est corrélée à sa primitive "request" précédente correspondante relève d'une initiative locale. Voir 1.2.		

Description du paramètre

Ordinal device number (numéro ordinal de l'appareil)

Ce paramètre spécifie l'indice ordinal de l'appareil adressé dans la chaîne de communications câblées. La confirmation au maître contient le nombre d'appareils esclaves passés.

Device data area (zone Données de l'appareil)

Ce paramètre spécifie l'emplacement dans la mémoire physique de l'esclave où les données à écrire sont stockées.

DLS-user data (données d'utilisateur de DLS)

Ce paramètre spécifie les données à écrire.

Working counter (Compteur en fonction)

Ce paramètre est incrémenté si les données ont été écrites avec succès.

5.3.3 Écriture physique à une adresse configurée (FPWR)

Avec le service FPWR, un maître écrit dans la mémoire ou dans le registre d'un esclave sélectionné par l'adresse de poste configurée de l'esclave. Le Tableau 6 montre les primitives de service et le paramètre de service FPWR.

Tableau 6 – Écriture physique d'adresse configurée (FPWR)

DL-CONFIGURED-PHYSICALWRITE (ÉCRITURE PHYSIQUE CONFIGURÉE DE DL) Nom du paramètre	Request (demande)	Confirm (confirmation)
	Entrée	Sortie
Configured device number (numéro d'appareil configuré)	M	
Device data area (zone Données de l'appareil)	M	
DLS-user data (données d'utilisateur de DLS)	U	
Working counter (Compteur en fonction)	M	M (=)
NOTE La méthode par laquelle une primitive "confirm" est corrélée à sa primitive "request" précédente correspondante relève d'une initiative locale. Voir 1.2.		

Description du paramètre

Configured device number (numéro d'appareil configuré)

Ce paramètre spécifie l'adresse de poste configurée de l'appareil adressé.

Device data area (zone Données de l'appareil)

Ce paramètre spécifie l'emplacement dans la mémoire physique de l'esclave où les données à écrire sont stockées.

DLS-user data (données d'utilisateur de DLS)

Ce paramètre spécifie les données à écrire.

Working counter (Compteur en fonction)

Ce paramètre est incrémenté si les données ont été écrites avec succès.

5.3.4 Écriture de diffusion (BWR)

Avec le service BWR, un maître écrit une zone mémoire physique dans tous les esclaves. Le Tableau 7 montre les primitives de service et le paramètre de service BWR.

Tableau 7 – Broadcast write (BWR)

DL-BROADCAST-WRITE (ÉCRITURE DE DIFFUSION DE DL) Nom du paramètre	Request (demande)	Confirm (confirmation)
	Entrée	Sortie
Broadcast address (adresse de diffusion)	M	M
Device data area (zone Données de l'appareil)	M	
DLS-user data (données d'utilisateur de DLS)	U	
Working counter (Compteur en fonction)	M	M (=)
NOTE La méthode par laquelle une primitive "confirm" est corrélée à sa primitive "request" précédente correspondante relève d'une initiative locale. Voir 1.2.		

Description du paramètre

Broadcast address (adresse de diffusion)

Ce paramètre est incrémenté au niveau de chaque esclave.

Device data area (zone Données de l'appareil)

Ce paramètre spécifie l'emplacement dans la mémoire physique où les données à écrire sont stockées.

DLS-user data (données d'utilisateur de DLS)

Ce paramètre spécifie les données à écrire.

Working counter (Compteur en fonction)

Ce paramètre est incrémenté par chaque esclave qui écrit des données dans sa mémoire physique.

5.3.5 Écriture logique (LWR)

Avec le service LWR, un maître écrit dans la mémoire ou dans un registre d'un ou plusieurs esclaves sélectionnés par une adresse logique. Le Tableau 8 montre les primitives de service et le paramètre de service LWR.

Tableau 8 – Logical write (LWR)

DL-LOGICAL-WRITE (ÉCRITURE LOGIQUE DE DL) Nom du paramètre	Request (demande)	Confirm (confirmation)
	Entrée	Sortie
Logical memory address (Adresse mémoire logique)	M	
DLS-user data (Données d'utilisateur de DLS)	U	
Working counter (Compteur en fonction)	M	M
NOTE La méthode par laquelle une primitive "confirm" est corrélée à sa primitive "request" précédente correspondante relève d'une initiative locale. Voir 1.2.		

Description du paramètre

Logical memory address (adresse mémoire logique)

Ce paramètre spécifie l'adresse de début dans la mémoire logique où les données à écrire sont situées.

DLS-user data (données d'utilisateur de DLS)

Ce paramètre spécifie les données à écrire.

Working counter (Compteur en fonction)

Ce paramètre est incrémenté par tous les esclaves qui détectent une concordance d'adresses de la zone mémoire logique demandée.

5.4 Services combinés de lecture/écriture

5.4.1 Vue d'ensemble

Pour les services combinés de lecture/écriture, les règles pour la lecture et/ou l'écriture de l'esclave adressé s'appliquent.

5.4.2 Lecture/écriture physique à une adresse définie par la position (APRW)

Avec le service APRW, un maître lit dans la mémoire ou dans le registre d'un esclave sélectionné par ordre de classement physique de l'esclave dans le segment et écrit des données dans la mémoire ou dans le registre du même esclave. Le Tableau 9 montre les primitives de service et le paramètre de service APRW.

Tableau 9 – Lecture/écriture physique à incréments automatiques (APRW)

DL-AUTOINCREMENT-PHYSICALREADWRITE (LECTURE/ECRIURE PHYSIQUE A INCREMENTS AUTOMATIQUES DE DL) Nom du paramètre	Request (demande)	Confirm (confirmation)
	Entrée	Sortie
Ordinal device number (Numéro ordinal de l'appareil)	M	M
Device data area (Zone Données de l'appareil)	M	
DLS-user data (Données d'utilisateur de DLS)	U	U
Working counter (Compteur en fonction)	M	M
NOTE La méthode par laquelle une primitive "confirm" est corrélée à sa primitive "request" précédente correspondante relève d'une initiative locale. Voir 1.2.		

Description du paramètre

Ordinal device number (numéro ordinal de l'appareil)

Ce paramètre spécifie l'indice ordinal de l'appareil adressé dans la chaîne de communications câblées. La confirmation au maître contient le nombre d'appareils esclaves passés.

Device data area (zone Données de l'appareil)

Ce paramètre spécifie l'emplacement dans la mémoire physique de l'esclave où les données à lire et à écrire sont stockées.

DLS-user data (données d'utilisateur de DLS)

Ce paramètre spécifie les données à écrire ou les données qui ont été lues.

Working counter (Compteur en fonction)

Ce paramètre est incrémenté si les données ont été écrites et lues avec succès.

5.4.3 Lecture/écriture physique à une adresse configurée (FPRW)

Avec le service FPRW, un maître lit dans la mémoire ou dans le registre d'un esclave sélectionné par l'adresse de poste configurée de l'esclave et écrit des données dans le même objet. Le Tableau 10 montre les primitives de service et le paramètre de service FPRW.

Tableau 10 – Lecture/écriture physique d'adresse configurée (FPRW)

DL-CONFIGURED-PHYSICALREADWRITE (LECTURE/ECRITURE PHYSIQUE CONFIGUREE DE DL) Nom du paramètre	Request (demande)	Confirm (confirmation)
	Entrée	Sortie
D_address (Adresse de destination)	M	
Device data area (Zone Données de l'appareil)	M	
DLS-user data (Données d'utilisateur de DLS)	U	U
Working counter (Compteur en fonction)	M	M
NOTE La méthode par laquelle une primitive "confirm" est corrélée à sa primitive "request" précédente correspondante relève d'une initiative locale. Voir 1.2.		

Description du paramètre

Configured device number (numéro d'appareil configuré)

Ce paramètre spécifie l'adresse de poste configurée de l'appareil adressé.

Device data area (zone Données de l'appareil)

Ce paramètre spécifie l'emplacement dans la mémoire physique de l'esclave où les données à lire et à écrire sont stockées.

DLS-user data (données d'utilisateur de DLS)

Ce paramètre spécifie les données à écrire ou les données qui ont été lues.

Working counter (Compteur en fonction)

Ce paramètre est incrémenté si les données ont été écrites et lues avec succès.

5.4.4 Lecture/écriture de diffusion (BRW)

Avec le service BRW, un maître lit dans une zone mémoire physique ou dans un registre, qui subira une opération OU logique effectuée sur les bits par tous les esclaves et écrit des données recueillies au niveau de tous les esclaves précédents. Le Tableau 11 montre les primitives de service et le paramètre de service BRW.

Tableau 11 – Broadcast read/write (BRW)

DL-BROADCAST-READWRITE (LECTURE/ECRITURE DE DIFFUSION DE DL) Nom du paramètre	Request (demande)	Confirm (confirmation)
	Entrée	Sortie
Broadcast address (Adresse de diffusion)	M	M
Device data area (Zone Données de l'appareil)	M	
DLS-user data (Données d'utilisateur de DLS)	U	U
Working counter (Compteur en fonction)	M	M
NOTE La méthode par laquelle une primitive "confirm" est corrélée à sa primitive "request" précédente correspondante relève d'une initiative locale. Voir 1.2.		

Description du paramètre

Broadcast address (adresse de diffusion)

Ce paramètre est incrémenté au niveau de chaque esclave.

Device data area (zone Données de l'appareil)

Ce paramètre spécifie l'emplacement dans la mémoire physique où les données à lire et à écrire sont stockées.

DLS-user data (données d'utilisateur de DLS)

Ce paramètre spécifie les données à écrire dans l'appareil ou le résultat de l'opération OU logique sur les bits des données qui ont été lues en provenance de chaque appareil.

Working counter (Compteur en fonction)

Ce paramètre est incrémenté par tous les esclaves qui ont effectué le OU logique sur les bits des données demandées et ont écrit des données dans leur mémoire physique.

5.4.5 Lecture/écriture logique (LRW)

Avec le service LRW, un maître écrit et lit dans la mémoire d'un ou plusieurs esclaves sélectionnés par une adresse logique. Le Tableau 12 montre les primitives de service et le paramètre de service LRW.

Tableau 12 – Logical read/write (LRW)

DL-LOGICAL-READWRITE (LECTURE/ECRITURE LOGIQUE DE DL) Nom du paramètre	Request (demande)	Confirm (confirmation)
	Entrée	Sortie
Logical memory address (Adresse mémoire logique)	M	
DLS-user data (Données d'utilisateur de DLS)	U	U
Working counter (Compteur en fonction)	M	M
NOTE La méthode par laquelle une primitive "confirm" est corrélée à sa primitive "request" précédente correspondante relève d'une initiative locale. Voir 1.2.		

Description du paramètre

Logical memory address (adresse mémoire logique)

Ce paramètre spécifie l'adresse de début dans la mémoire logique où les données à lire ou à écrire sont situées.

DLS-user data (données d'utilisateur de DLS)

Ce paramètre spécifie les données à écrire ou les données qui ont été lues.

Working counter (Compteur en fonction)

Ce paramètre est incrémenté si les données ont été écrites avec succès et si les données ont été lues avec succès.

5.4.6 Lecture/multiples écritures physiques à une adresse définie par la position (ARMW)

Avec le service ARMW, un maître lit des données dans la mémoire ou dans le registre d'un esclave sélectionné par l'ordre de classement physique de l'esclave dans le segment et écrit la valeur des données de paramètre dans la même mémoire ou dans le même registre de tous les autres esclaves suivants. Le Tableau 13 montre les primitives de service et le paramètre de service ARMW.

Tableau 13 – Lecture/multiples écritures physiques à incréments automatiques (ARMW)

DL-AUTOINCREMENT-READMULTIPLEWRITE (LECTURE/MULTIPLE ECRITURE A INCREMENTS AUTOMATIQUES DE DL) Nom du paramètre	Request (demande)	Confirm (confirmation)
	Entrée	Sortie
Ordinal device number (Numéro ordinal de l'appareil)	M	M
Device data area (Zone Données de l'appareil)	M	
DLS-user data (Données d'utilisateur de DLS)	U	U
Working counter (Compteur en fonction)	M	M
NOTE La méthode par laquelle une primitive "confirm" est corrélée à sa primitive "request" précédente correspondante relève d'une initiative locale. Voir 1.2.		

Description du paramètre

Ordinal device number (numéro ordinal de l'appareil)

Ce paramètre spécifie l'indice ordinal de l'appareil dans la chaîne de communications câblées qui exécute l'action de lecture. La confirmation au maître contient le nombre d'appareils esclaves passés.

Device data area (zone Données de l'appareil)

Ce paramètre spécifie l'emplacement dans la mémoire physique de l'esclave où les données à lire et à écrire sont stockées.

DLS-user data (données d'utilisateur de DLS)

Ce paramètre spécifie les données à écrire ou les données qui ont été lues.

Working counter (Compteur en fonction)

Ce paramètre est incrémenté si les données ont été lues avec succès.

5.4.7 Lecture/multiple écritures physiques à une adresse configurée (FRMW)

Avec le service FRMW, un maître lit dans la mémoire ou dans le registre d'un esclave sélectionné par l'adresse de poste configurée de l'esclave et écrit des données dans le même objet de tous les autres esclaves. Le Tableau 14 montre les primitives de service et le paramètre de service FRMW.

Tableau 14 – Lecture/multiples écritures physiques d'adresse configurée (FRMW)

DL-CONFIGURED-READMULTIPLEWRITE (LECTURE/MULTIPLES ECRITURES PHYSIQUES DE DL) Nom du paramètre	Request (demande)	Confirm (confirmation)
	Entrée	Sortie
Configured device number (Numéro d'appareil configuré)	M	
Device data area (Zone Données de l'appareil)	M	
DLS-user data (Données d'utilisateur de DLS)	U	U
Working counter (Compteur en fonction)	M	M
NOTE La méthode par laquelle une primitive "confirm" est corrélée à sa primitive "request" précédente correspondante relève d'une initiative locale. Voir 1.2.		

Description du paramètre

Configured device number (numéro d'appareil configuré)

Ce paramètre spécifie l'adresse de poste configurée de l'appareil qui a accompli l'opération de lecture.

Device data area (zone Données de l'appareil)

Ce paramètre spécifie l'emplacement dans la mémoire physique de l'esclave où les données à lire et à écrire sont stockées.

DLS-user data (données d'utilisateur de DLS)

Ce paramètre spécifie les données à écrire ou les données qui ont été lues.

Working counter (Compteur en fonction)

Ce paramètre est incrémenté si les données ont été lues avec succès.

5.5 Services de réseau

5.5.1 Vue d'ensemble

Les services variables de réseau sont décrits du point de vue de l'éditeur. La couche liaison de données spécifie les services pour édition. Ce service est dédié aux communications entre maîtres ou entre maître et appareils Ethernet standard.

5.5.2 Fourniture de variables de réseau (PNV)

Avec le service PNV (Provide Network Variables), un maître fournit des données à un ou plusieurs postes (maître ou esclaves). L'adressage primaire est effectué par l'adresse MAC de destination (adresse de groupe/adresse individuelle). Les postes recevant une indication transmettront les données à l'utilisateur de DL. Le Tableau 15 montre les primitives de service et le paramètre de service PNV.

Tableau 15 – Fourniture de variables de réseau (PNV)

DL-PROVIDE-NETWORKVARIABLE (FOURNITURE DE VARIABLES DE RESEAU DE DL) Nom du paramètre	Request (demande)	Indication
	Entrée	Sortie
Publisher ID (Identificateur d'éditeur)	M	M (=)
Cycle	M	M (=)
List of network variables (Liste des variables de réseau)	M	M (=)

Description du paramètre

Publisher ID (Identificateur d'éditeur)

Ce paramètre spécifie les valeurs de la chaîne d'octets d'identification.

Cycle

Ce paramètre représente un identificateur numérique du cycle d'esclaves et peut servir à détecter de nouvelles valeurs.

List of network variables (Liste des variables de réseau)

Ce paramètre spécifie une liste de variables de réseau. Chaque élément de la liste spécifie:

Index (Indice)

Ce paramètre spécifie l'identificateur unique au sein du fournisseur de la variable de réseau.

Hash value (Valeur de hachage)

Ce paramètre spécifie une valeur de hachage de la description de la structure de variable applicable aux variables de réseau. L'algorithme de hachage est spécifique au fournisseur.

Données

Ce paramètre spécifie les valeurs des données d'éditeur.

5.6 Boîte à lettres

5.6.1 Vue d'ensemble

La boîte à lettres fonctionne dans les deux sens – du maître vers un esclave et d'un esclave vers le maître. Elle prend en charge la communication indépendante "full duplex" dans les deux sens et plusieurs protocoles d'utilisateur de DL. La communication esclave-esclave est gérée par le maître, agissant comme routeur. L'en-tête de boîte à lettres contient un champ adresse qui permet au maître de rediriger les services.

La boîte à lettres utilise les deux voies de gestionnaire de synchronisation, une pour chaque sens (par exemple: voie de gestionnaire de synchronisation 0 allant du maître vers l'esclave et voie de gestionnaire de synchronisation 1 allant de l'esclave vers le maître). Les voies de gestionnaire de synchronisation configurées comme boîte à lettres empêchent l'engorgement de l'autre bout. Normalement, la communication par boîte à lettres est non cyclique et adresse un seul esclave. Par conséquent, c'est l'adressage physique sans recours à une FMMU qui est utilisé, en lieu et place de l'adressage logique.

5.6.1.1 Communication de maître à esclave

Le maître doit vérifier le compteur en fonction au moment de la réponse d'une commande de boîte à lettres envoyée à un esclave. Si le compteur en fonction ne s'incrémente pas (normalement parce que l'esclave n'a pas complètement lu la dernière commande) ou s'il n'y a pas de réponse dans les délais, le maître doit émettre de nouveau la commande de boîte à lettres. Une reprise sur erreur plus poussée est de la responsabilité des protocoles supérieurs.

5.6.1.2 Communication d'esclave à maître

Le maître doit déterminer qu'un esclave a rempli le gestionnaire de synchronisation avec une commande de boîte à lettres et envoyer une commande de lecture appropriée aussi rapidement que possible.

Il existe différentes façons de déterminer qu'un esclave a rempli son gestionnaire de synchronisation. Une solution astucieuse est de configurer le "bit écrit" de l'en-tête de configuration du gestionnaire de synchronisation 1 à une adresse logique et de lire ce bit de façon cyclique. L'utilisation d'une adresse logique permet de lire les bits depuis plusieurs esclaves ensemble et de configurer chaque esclave sur une adresse binaire individuelle. L'inconvénient de cette solution est qu'il faut une FMMU par esclave.

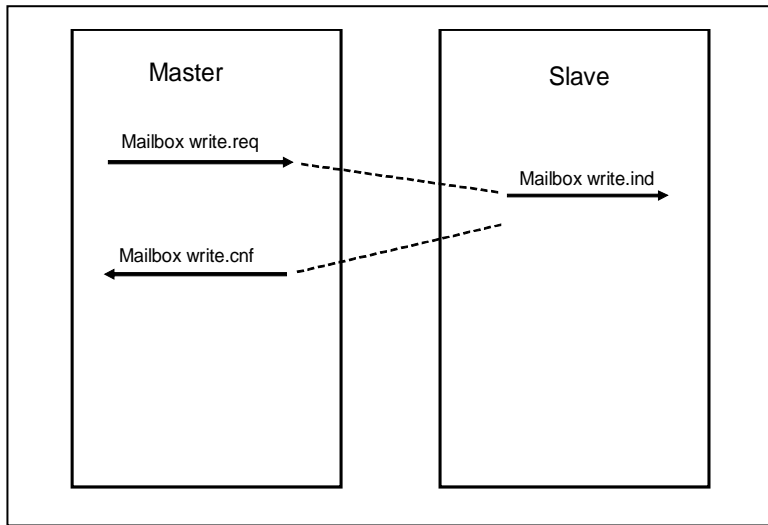
Une autre solution consiste simplement à sonder la zone Données du gestionnaire de synchronisation. Le compteur en fonction de la commande de lecture en question ne sera incrémenté qu'une seule fois si l'esclave a rempli la zone avec une nouvelle commande.

Le maître doit vérifier le compteur en fonction au moment de la réponse de la commande de boîte à lettres envoyée à un esclave. Si le compteur en fonction ne s'incrémente pas (normalement parce que l'esclave n'a pas complètement lu la dernière commande) ou s'il n'y a pas de réponse dans les délais, le maître doit basculer le paramètre de répétition de tentative dans la zone du gestionnaire de synchronisation. Avec un paramètre de répétition de tentative basculé, l'esclave doit placer les dernières données de lecture dans la boîte à lettres. Une reprise sur erreur plus poussée est de la responsabilité des protocoles supérieurs.

Les primitives des services de boîte à lettres sont mises en correspondance au niveau de l'esclave avec les primitives de mémoire d'utilisateur de DL.

Écriture de boîte à lettres	événement, lecture locale
Mise à jour de lecture de boîte à lettres	écriture locale
Lecture de boîte à lettres	événement

La Figure 9 montre les primitives entre maître et esclave dans le cas d'une séquence réussie d'écriture de boîte à lettres.

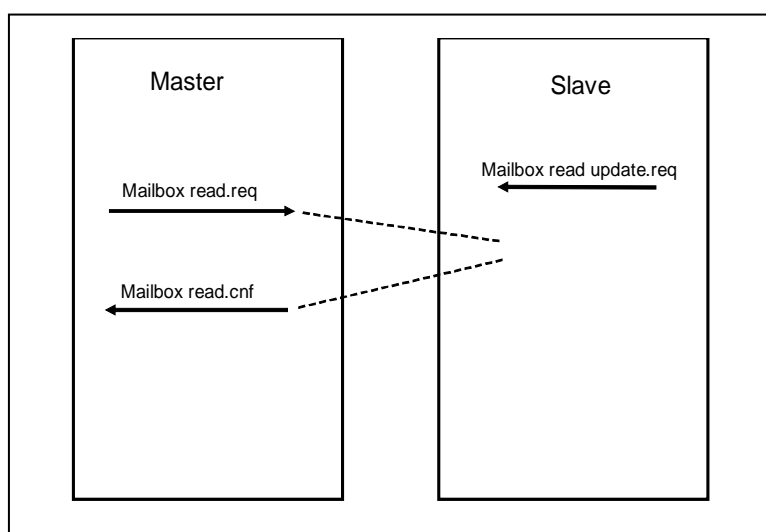


Légende

Anglais	Français
Master	Maître
Slave	Esclave

Figure 9 – Séquence réussie d'écriture de boîte à lettres

La Figure 10 montre les primitives entre maître et esclave dans le cas d'une séquence réussie de lecture de boîte à lettres.



Légende

Anglais	Français
Master	Maître
Slave	Esclave

Figure 10 – Séquence réussie de lecture de boîte à lettres

5.6.2 Services de transmission de données de boîte à lettres

5.6.2.1 Écriture de boîte à lettres

Le service écriture de boîte à lettres tel que spécifié dans le Tableau 16 est basé sur l'écriture (transmission de maître à esclave) de mémoire pour obtenir une transmission acquittée de données.

Tableau 16 – Mailbox write

DL-MAILBOX-WRITE (ÉCRITURE DE BOÎTE A LETTRES DE DL) Nom du paramètre	Request (demande)	Indication	Confirm (confirmation)
	Entrée	Sortie	Sortie
D_address (Adresse de destination)	M		
MBX	M		
S_address (Adresse source)	M	M (=)	
Channel (Voie)	M	M (=)	
Priority (Priorité)	M	M (=)	
Type	M	M (=)	
Cnt	M	M (=)	
DLS-user data (Données d'utilisateur de DLS)	U	U (=)	
DL-status (Statut de DL)			M

NOTE La méthode par laquelle une primitive "confirm" est corrélée à sa primitive "request" précédente correspondante relève d'une initiative locale. Voir 1.2.

Description du paramètre

D_address (adresse de destination)

Ce paramètre spécifie l'adresse de nœud pour le nœud de destination afin de permettre la communication d'esclave à esclave ou la communication au-delà des frontières du réseau en utilisant une adresse virtuelle.

MBX

Ce paramètre spécifie la boîte à lettres.

S_address (adresse source)

Ce paramètre spécifie l'adresse de poste pour le poste source afin de permettre la communication d'esclave à esclave ou la communication au-delà des frontières du réseau en utilisant une adresse virtuelle.

Channel (voie)

Ce paramètre spécifie la voie de communication.

Priority (priorité)

Ce paramètre spécifie une priorité de communication

Type

Ce paramètre spécifie le type de protocole du service de boîte à lettres utilisé.

Cnt

Ce paramètre spécifie un compteur de service.

DLS-user data (données d'utilisateur de DLS)

Ce paramètre spécifie les données à écrire ou les données qui ont été écrites.

DL-status (statut de DL)

Ce paramètre spécifie le résultat de l'opération.

5.6.2.2 Mise à jour de lecture de boîte à lettres

Le service lecture de boîte à lettres tel que spécifié dans le Tableau 17 est basé sur l'écriture locale de mémoire. Le tampon de mise à jour doit être conservé tant qu'un fonctionnement répété est possible.

Tableau 17 – Mise à jour de lecture de boîte à lettres

DL-MAILBOX-READUPD (MISE A JOUR DE LECTURE DE BOITE A LETTRES DE DL) Nom du paramètre	Request (demande)	Confirm (confirmation)
	Entrée	Sortie
D_address (Adresse de destination)	M	
Channel (Voie)	M	
Priority (Priorité)	M	
Type	M	
Cnt	M	
DLS-user data (Données d'utilisateur de DLS)	U	
DL-status (Statut de DL)		M
NOTE La méthode par laquelle une primitive "confirm" est corrélée à sa primitive "request" précédente correspondante relève d'une initiative locale. Voir 1.2.		

Description du paramètre

D_address (adresse de destination)

Ce paramètre spécifie l'adresse de poste pour le poste de destination afin de permettre la communication d'esclave à esclave ou la communication au-delà des frontières du réseau en utilisant une adresse virtuelle.

Channel (voie)

Ce paramètre spécifie la voie de communication.

Priority (priorité)

Ce paramètre spécifie une priorité de communication

Type

Ce paramètre spécifie le type de protocole du service de boîte à lettres utilisé.

Cnt

Ce paramètre spécifie un compteur de service.

DLS-user data (données d'utilisateur de DLS)

Ce paramètre spécifie les données qui ont été lues.

DL-status (statut de DL)

Ce paramètre spécifie le résultat de l'opération.

5.6.2.3 Lecture de boîte à lettres

Le service lecture de boîte à lettres tel que spécifié dans le Tableau 18 est basé sur la lecture (transmission de maître à esclave) de mémoire pour obtenir une transmission acquittée de données.

Tableau 18 – Lecture de boîte à lettres

DL-MAILBOX-READ (LECTURE DE BOITE A LETTRES DE DL) Nom du paramètre	Request (demande)	Indication	Confirm (confirmation)
	Entrée	Sortie	Sortie
S_address (Adresse source)	M		
MBX	M		
D_address (Adresse de destination)			C
Channel (Voie)			C
Priority (Priorité)			C
Type			C
Cnt			C
DLS-user data (Données d'utilisateur de DLS)			C
DL-status (Statut de DL)		M	M

NOTE La méthode par laquelle une primitive "confirm" est corrélée à sa primitive "request" précédente correspondante relève d'une initiative locale. Voir 1.2.

Description du paramètre

S_address (adresse source)

Ce paramètre spécifie l'adresse de poste pour le poste source.

MBX

Ce paramètre spécifie la boîte à lettres.

D_address (adresse de destination)

Ce paramètre spécifie l'adresse de poste pour le poste de destination afin de permettre la communication d'esclave à esclave ou la communication au-delà des frontières du réseau en utilisant une adresse virtuelle.

Channel (voie)

Ce paramètre spécifie la voie de communication.

Priority (priorité)

Ce paramètre spécifie une priorité de communication

Type

Ce paramètre spécifie le type de protocole du service de boîte à lettres utilisé.

Cnt

Ce paramètre spécifie un compteur de service.

DLS-user data (données d'utilisateur de DLS)

Ce paramètre spécifie les données qui ont été lues.

DL-status (statut de DL)

Ce paramètre spécifie le résultat de l'opération.

6 Interactions locales

6.1 Lecture locale

Avec cette fonction, l'utilisateur de DL lit des données dans une zone mémoire. Le Tableau 19 montre les primitives de service et le paramètre de cette fonction.

Tableau 19 – Lecture locale

DL-READ LOCAL (LECTURE LOCALE DE DL)	Request (demande)	Confirm (confirmation)
	Entrée	Sortie
Nom du paramètre		
Memory area (Zone mémoire)	M	
DLS-user data (Données d'utilisateur de DLS)		U
DL-status (Statut de DL)		M
NOTE La méthode par laquelle une primitive "confirm" est corrélée à sa primitive "request" précédente correspondante relève d'une initiative locale. Voir 1.2.		

Description du paramètre

Memory area (zone mémoire)

Ce paramètre spécifie la zone mémoire devant être lue.

DLS-user data (données d'utilisateur de DLS)

Ce paramètre spécifie les données qui ont été lues dans la zone mémoire spécifiée.

DL-status (statut de DL)

Ce paramètre spécifie le résultat de l'opération.

6.2 Écriture locale

Avec cette fonction, l'utilisateur de DL écrit des données dans une zone mémoire. Le Tableau 20 montre les primitives de service et le paramètre de cette fonction.

Tableau 20 – Écriture locale

DL-WRITE LOCAL (ECRITURE LOCALE DE DL)	Request (demande)	Confirm (confirmation)
Nom du paramètre	Entrée	Sortie
Memory area (Zone mémoire)	M	
DLS-user data (Données d'utilisateur de DLS)	U	
DL-status (Statut de DL)		M
NOTE La méthode par laquelle une primitive "confirm" est corrélée à sa primitive "request" précédente correspondante relève d'une initiative locale. Voir 1.2.		

Description du paramètre

Memory area (zone mémoire)

Ce paramètre spécifie la zone mémoire devant être écrite.

DLS-user data (données d'utilisateur de DLS)

Ce paramètre spécifie les données à écrire dans la zone mémoire spécifiée.

DL-status (statut de DL)

Ce paramètre spécifie le résultat de l'opération.

6.3 Événement local

Avec cette fonction, l'utilisateur de DL récupère une indication d'un événement. Le Tableau 21 montre les primitives de service et le paramètre de cette fonction.

Tableau 21 – Événement local

DL-EVENT LOCAL (EVENEMENT LOCAL DE DL)	Indication
Nom du paramètre	Sortie
Sync manager (Gestionnaire de synchronisation)	C
Type	C

Description du paramètre

Sync manager (Gestionnaire de synchronisation)

Ce paramètre indique la voie du gestionnaire de synchronisation.

Type

Ce paramètre indique le type d'événement (lecture ou écriture).

Bibliographie

CEI 61158-1:2014, *Réseaux de communication industriels – Spécifications des bus de terrain – Partie 1 : Présentation et lignes directrices des séries CEI 61158 et CEI 61784*

CEI 61158-2, *Réseaux de communication industriels – Spécifications des bus de terrain – Partie 2 : Spécification et définition des services de la couche physique*

CEI 61158-4-12, *Réseaux de communication industriels – Spécifications des bus de terrain – Partie 4-12 : Spécification du protocole de la couche liaison de données – Éléments de type 12*

CEI 61158-5-12, *Réseaux de communication industriels – Spécifications des bus de terrain – Partie 5-12: Définition des services de la couche application – Éléments de type 12*

CEI 61158-6-12, *Réseaux de communication industriels – Spécifications des bus de terrain – Partie 6-12 : Spécification du protocole de la couche application – Éléments de type 12*

IEC 61588, *Precision clock synchronization protocol for networked measurement and control systems* (disponible en anglais seulement)

CEI 61784-1, *Réseaux de communication industriels – Profils – Partie 1 : Profils de bus de terrain*

CEI 61784-2, *Réseaux de communication industriels – Profils – Partie 2 : Profils de bus de terrain supplémentaires pour les réseaux en temps réel basés sur l'ISO/CEI 8802-3*

ISO/IEC/TR 8802-1, *Information technology – Telecommunications and information exchange between systems – Local and metropolitan area networks – Specific requirements – Part 1: Overview of Local Area Network Standards* (disponible en anglais seulement)

IEEE 802.1Q, *IEEE Standard for Local and metropolitan area networks – Virtual Bridged Local Area Networks*; disponible à l'adresse <<http://www.ieee.org>>

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

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

INTERNATIONAL
ELECTROTECHNICAL
COMMISSION

3, rue de Varembé
PO Box 131
CH-1211 Geneva 20
Switzerland

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