

INTERNATIONAL STANDARD

NORME INTERNATIONALE

**Industrial communication networks – Fieldbus specifications –
Part 5-22: Application layer service definition – Type 22 elements**

**Réseaux de communication industriels – Spécifications des bus de terrain –
Partie 5-22: Définition des services de la couche application – Éléments
de Type 22**



THIS PUBLICATION IS COPYRIGHT PROTECTED

Copyright © 2010 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 la CEI ou du Comité national de la CEI du pays du demandeur.

Si vous avez des questions sur le copyright de la CEI ou si vous désirez obtenir des droits supplémentaires sur cette publication, utilisez les coordonnées ci-après ou contactez le Comité national de la CEI de votre pays de résidence.

IEC Central Office
3, rue de Varembé
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.

Useful links:

IEC publications search - www.iec.ch/searchpub

The advanced search enables you 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It also gives information on projects, replaced and withdrawn publications.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 on-line 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 additional languages. Also known as the International Electrotechnical Vocabulary (IEV) on-line.

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 la CEI

La Commission Electrotechnique Internationale (CEI) 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 CEI

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

Liens utiles:

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

La recherche avancée vous permet de trouver des publications CEI 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.

Just Published CEI - webstore.iec.ch/justpublished

Restez informé sur les nouvelles publications de la CEI. 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 au monde 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 les langues additionnelles. Egalement appelé Vocabulaire Electrotechnique International (VEI) en ligne.

Service Clients - webstore.iec.ch/csc

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

INTERNATIONAL STANDARD

NORME INTERNATIONALE

**Industrial communication networks – Fieldbus specifications –
Part 5-22: Application layer service definition – Type 22 elements**

**Réseaux de communication industriels – Spécifications des bus de terrain –
Partie 5-22: Définition des services de la couche application – Éléments
de Type 22**

INTERNATIONAL
ELECTROTECHNICAL
COMMISSION

COMMISSION
ELECTROTECHNIQUE
INTERNATIONALE

PRICE CODE
CODE PRIX

XC

ICS 25.04.40; 35.100.70; 35.110

ISBN 978-2-88912-866-2

**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 Overview.....	7
1.2 Specifications.....	8
1.3 Conformance.....	8
2 Normative references.....	8
3 Terms, definitions, abbreviations, symbols and conventions.....	9
3.1 ISO/IEC 7498-1 terms.....	9
3.2 ISO/IEC 8822 terms.....	9
3.3 ISO/IEC 9545 terms.....	9
3.4 ISO/IEC 8824-1 terms.....	10
3.5 Type 22 fieldbus application-layer specific definitions.....	10
3.6 Abbreviations and symbols.....	13
3.7 Conventions.....	16
4 Concepts.....	19
4.1 Common concepts.....	19
4.2 Type specific concepts.....	19
5 Data type ASE.....	23
5.1 Overview.....	23
5.2 Formal definition of data type objects.....	23
5.3 FAL defined data types.....	23
6 Communication model specification.....	31
6.1 Application service elements (ASEs).....	31
6.2 Application relationships (ARs).....	74
Bibliography.....	79
Figure 1 – Producer-consumer interaction model.....	21
Figure 2 – RTFL device reference model.....	22
Figure 3 – RTFN device reference model.....	23
Figure 4 – Type 22 CeS device structure.....	32
Figure 5 – Successful SDO expedited download sequence.....	46
Figure 6 – Successful SDO normal download initialization sequence.....	46
Figure 7 – Successful SDO download sequence.....	46
Figure 8 – Successful SDO expedited upload sequence.....	47
Figure 9 – Successful SDO normal upload initialization sequence.....	47
Figure 10 – Successful SDO upload sequence.....	47
Figure 11 – Failed SDO expedited download initialization sequence.....	48
Figure 12 – Failed SDO download after initialization sequence.....	48
Figure 13 – Failed SDO download sequence.....	48
Figure 14 – Emergency sequence.....	49
Figure 15 – Heartbeat sequence.....	49
Figure 16 – Process data write sequence.....	50

Figure 17 – PDO mapping principle	50
Figure 18 – Process data object.....	51
Figure 19 – SEF service sequence.....	64

Table 1 – Object dictionary structure.....	32
Table 2 – Initiate SDO expedited download service.....	54
Table 3 – Initiate SDO normal download service	55
Table 4 – SDO download service	56
Table 5 – Initiate SDO expedited upload service	57
Table 6 – Initiate SDO normal upload service	58
Table 7 – SDO upload service	60
Table 8 – SDO abort service	61
Table 9 – Process data write service.....	62
Table 10 – Emergency service (EMCY).....	62
Table 11 – Heartbeat service	63
Table 12 – Send frame service	65
Table 13 – AL-Network verification service	67
Table 14 – AL-RTFL configuration service	68
Table 15 – AL-DelayMeasurement start service.....	69
Table 16 – AL-DelayMeasurement read service.....	70
Table 17 – PCS configuration service.....	70
Table 18 – MII read service	71
Table 19 – MII write service.....	71
Table 20 – AL-RTFN scan network read service	72
Table 21 – Application layer management service.....	72
Table 22 – Start synchronization service.....	73
Table 23 – Stop synchronization service	73
Table 24 – PTPNSU AREP class	75
Table 25 – PTMNSU AREP class.....	76
Table 26 – PTPNSC AREP class	76
Table 27 – PTPUTC AREP class.....	76
Table 28 – FAL services by AREP class	77
Table 29 – FAL services by AREP role	78

INTERNATIONAL ELECTROTECHNICAL COMMISSION

**INDUSTRIAL COMMUNICATION NETWORKS –
FIELDBUS SPECIFICATIONS –****Part 5-22: Application layer service definition –
Type 22 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.

NOTE 1 Use of some of the associated protocol Types is restricted by their intellectual-property-right holders. In all cases, the commitment to limited release of intellectual-property-rights made by the holders of those rights permits a particular data-link layer protocol Type to be used with physical layer and application layer protocols in Type combinations as specified explicitly in the profile parts. Use of the various protocol Types in other combinations may require permission of their respective intellectual-property-right holders.

International Standard IEC 61158-5-22 has been prepared by subcommittee 65C: Industrial networks, of IEC technical committee 65: Industrial-process measurement, control and automation.

This standard cancels and replaces IEC/PAS 61158-5-22 published in 2009. This first edition constitutes a technical revision.

This bilingual version published in 2012-01 corresponds to the English version published in 2010-08.

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

FDIS	Report on voting
65C/606/FDIS	65C/620/RVD

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

The French version has not been voted upon.

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.

NOTE 2 The revision of this standard will be synchronized with the other parts of the IEC 61158 series.

INTRODUCTION

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

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

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

Withdrawing
IECNORM.COM : Click to view the full PDF of IEC 61158-5-22:2010

INDUSTRIAL COMMUNICATION NETWORKS – FIELDBUS SPECIFICATIONS –

Part 5-22: Application layer service definition – Type 22 elements

1 Scope

1.1 Overview

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

This standard provides common elements for basic time-critical and non-time-critical messaging communications between application programs in an automation environment and material specific to Type 22 fieldbus. The term “time-critical” is used to represent the presence of a time-window, within which one or more specified actions are required to be completed with some defined level of certainty. Failure to complete specified actions within the time window risks failure of the applications requesting the actions, with attendant risk to equipment, plant and possibly human life.

This standard defines in an abstract way the externally visible service provided by the fieldbus application layer in terms of

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

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

- a) the FAL user at the boundary between the user and the application layer of the fieldbus reference model; and
- b) Systems Management at the boundary between the application layer and Systems Management of the fieldbus reference model.

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

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

Although these services specify, from the perspective of applications, how request and responses are issued and delivered, they do not include a specification of what the requesting and responding applications are to do with them. That is, the behavioral aspects of the applications are not specified; only a definition of what requests and responses they can

send/receive is specified. This permits greater flexibility to the FAL users in standardizing such object behavior. In addition to these services, some supporting services are also defined in this standard to provide access to the FAL to control certain aspects of its operation.

1.2 Specifications

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

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

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

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

1.3 Conformance

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

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

2 Normative references

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

IEC 60559, *Binary floating-point arithmetic for microprocessor systems*

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

IEC/TR 61158-1:2010¹, *Industrial communication networks – Fieldbus specifications – Part 1: Overview and guidance for the IEC 61158 and IEC 61784 series*

IEC 61158-4-22:2010¹, *Industrial communication networks – Fieldbus specifications – Part 4-22: Data-link layer protocol specification – Type 22 elements*

IEC 61158-6-22:2010¹, *Industrial communication networks – Fieldbus specifications – Part 6-22: Application layer protocol specification – Type 22 elements*

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

¹ To be published.

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

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 8822, *Information Technology – Open Systems Interconnection – Presentation service definition*

ISO/IEC 8824-1, *Information Technology – Abstract Syntax Notation One (ASN.1): Specification of basic notation*

ISO/IEC 9545, *Information Technology – Open Systems Interconnection – Application Layer structure*

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

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

3 Terms, definitions, abbreviations, symbols and conventions

For the purposes of this document, the following terms as defined in these publications apply:

3.1 ISO/IEC 7498-1 terms

- a) application entity
- b) application process
- c) application protocol data unit
- d) application service element
- e) application entity invocation
- f) application process invocation
- g) application transaction
- h) real open system
- i) transfer syntax

3.2 ISO/IEC 8822 terms

- a) abstract syntax
- b) presentation context

3.3 ISO/IEC 9545 terms

- a) application-association
- b) application-context
- c) application context name
- d) application-entity-invocation
- e) application-entity-type
- f) application-process-invocation
- g) application-process-type

- h) application-service-element
- i) application control service element

3.4 ISO/IEC 8824-1 terms

- a) object identifier
- b) type

3.5 Type 22 fieldbus application-layer specific definitions

3.5.1

application

function for which data is exchanged

3.5.2

application object

representation of a particular component within a device

3.5.3

acyclic data

data which is transferred from time to time for dedicated purposes

3.5.4

bit

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

3.5.5

cell

synonym for a single DL-segment which uses RTFL communication model

3.5.6

channel

path provided for conveying data

3.5.7

client

object which uses the services of a server by initiating a message to perform a task

3.5.8

communication cycle

fixed time period between which the root device issues empty DLPDUs for cyclic communication initiation in which data is transmitted utilizing CDC and MSC

3.5.9

connection

logical binding between two application objects

3.5.10

cycle time

duration of a communication cycle

3.5.11

cyclic

events which repeat in a regular and repetitive manner

3.5.12**cyclic communication**

periodic exchange of telegrams

3.5.13**cyclic data**

data which is transferred in a regular and repetitive manner for dedicated purposes

3.5.14**cyclic data channel****CDC**

part of one or more DLPDUs, which is reserved for cyclic data

3.5.15**data**

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

3.5.16**device**

physical entity connected to the fieldbus

3.5.17**error**

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

3.5.18**error code**

identification number of a specific type of error

3.5.19**gateway**

device acting as a linking element between different protocols

3.5.20**index**

position of an object within the object dictionary

3.5.21**inter-cell communication**

communication between a RTFL device and a RTFN device or communication between a RTFL device and another RTFL device in different cells linked by RTFN

3.5.22**interface**

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

3.5.23**intra-cell communication**

communication between a RTFL device and another RTFL device in the same cell

3.5.24**logical double line**

sequence of root device and all ordinary devices processing the communication DLPDU in forward and backward direction

3.5.25

mapping parameters

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

3.5.26

master clock

global time base for the PCS mechanism

3.5.27

message

ordered sequence of octets intended to convey data

3.5.28

message channel

MSC

part of one or more DLPDUs, which is reserved for acyclic data

3.5.29

network

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

3.5.30

ordinary device

OD

slave in the communication system, which utilizes RTFL for cyclic and acyclic data interchange with other ODs in the same logical double line

3.5.31

precise clock synchronization

PCS

mechanism to synchronize clocks of RTFL devices and maintain a global time base

3.5.32

process data

data designated to be transferred cyclically or acyclically for the purpose of processing

3.5.33

process data object

dedicated data object(s) designated to be transferred cyclically or acyclically for the purpose of processing

3.5.34

protocol

convention about the data formats, time sequences, and error correction in the data exchange of communication systems

3.5.35

root device

RD

master in the communication system, which organises, initiates and controls the RTFL cyclic and acyclic data interchange for one logical double line

3.5.36

real time frame line

RTFL

communication model communicating in a logical double line

3.5.37**real time frame network****RTFN**

communication model communicating in a switched network

3.5.38**round trip time**

transmission time needed by a DLPDU from the RD to the last OD in forward and backward direction

3.5.39**sub-index**

sub-position of an individual element of an object within the object dictionary

3.5.40**timing signal**

time-based indication of the occurrence of an event, commonly as an interrupt signal, used for DL-user synchronization

3.5.41**topology**

physical network architecture with respect to the connection between the stations of the communication system

3.6 Abbreviations and symbols

AE	Application entity
AL	Application layer
ALME	Application layer management entity
AP	Application process
APDU	Application layer protocol data unit
APO	Application process object
AR	Application relationship
AREP	Application relationship end point
ASE	Application service element
CAN	Controller area network
CDC	Cyclic data channel
CDCL	CDC line
CDCN	CDC network
CeS	CANopen expands Type 22
CL	Communication layer

Cnf	Confirmation
DA	Device address
DHCP	Dynamic Host Configuration Protocol
DL-	Data-link layer (as a prefix)
DLL	DL-layer
DLPDU	DL-protocol data unit
EDS	Electronic data sheet
EMCY	Emergency
FAL	Fieldbus application layer
GW	Gateway
ID	Identification
IETF	Internet Engineering Task Force
Ind	Indication
IP	Internet protocol
IPv4	IP version 4
IPv6	IP version 6
IRQ	Interrupt request
LME	Layer management entity
MAC	Medium access control
MC	Master clock
MII	Media independent interface
MSC	Message channel
MSCL	MSC line
MSCN	MSC network
OD	Ordinary device
OS	Operating system

OSI	Open systems interconnection
PCS	Precise clock synchronization
PDO	Process data object
PHY	Physical interface controller
PID	Packet ID
PTMNSU	Point-to-multipoint network-scheduled unconfirmed
PTPNSC	Point-to-point network-scheduled confirmed
PTPNSU	Point-to-point network-scheduled unconfirmed
PTPUTC	Point-to-point user-triggered confirmed
RD	Root device
Req	Request
RFC	Request for comments
Rsp	Response
RTF	Real time frame
RTFL	Real time frame line
RTFN	Real time frame network
RO	Read only
RW	Read and write access
Rx	Receive direction
RxPDO	Receive PDO
SDO	Service data object
SEF	Standard ISO/IEC 8802-3 DLPDU
StdErr	Standard error output
StdIn	Standard input
StdOut	Standard output
SYNC	Synchronization

TCP	Transmission control protocol
TT	Transmission type
Tx	Transmit direction
TxPDO	Transmit PDO
UDP	User datagram protocol
WO	Write only

3.7 Conventions

3.7.1 Overview

The FAL is defined as a set of object-oriented ASEs. Each ASE is specified in a separate sub-clause. Each ASE specification is composed of two parts, its class specification, and its service specification.

The class specification defines the attributes of the class. The attributes are accessible from instances of the class using the Object Management ASE services specified in Clause 5 of this standard. The service specification defines the services that are provided by the ASE.

3.7.2 General conventions

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

3.7.3 Conventions for class definitions

Class definitions are described using templates. Each template consists of a list of attributes for the class. The general form of the template is shown below:

FAL ASE:		ASE Name
CLASS:		Class name
CLASS ID:		#
PARENT CLASS:		Parent class name
ATTRIBUTES:		
1	(o)	Key Attribute: numeric identifier
2	(o)	Key Attribute: name
3	(m)	Attribute: attribute name(values)
4	(m)	Attribute: attribute name(values)
4.1	(s)	Attribute: attribute name(values)
4.2	(s)	Attribute: attribute name(values)
4.3	(s)	Attribute: attribute name(values)
5.	(c)	Constraint: constraint expression
5.1	(m)	Attribute: attribute name(values)
5.2	(o)	Attribute: attribute name(values)
6	(m)	Attribute: attribute name(values)
6.1	(s)	Attribute: attribute name(values)
6.2	(s)	Attribute: attribute name(values)
SERVICES:		

1	(o)	OpsService:	service name
2	(c)	Constraint:	constraint expression
2.1	(o)	OpsService:	service name
3	(m)	MgtService:	service name

- (1) The "FAL ASE:" entry is the name of the FAL ASE that provides the services for the class being specified.
- (2) The "CLASS:" entry is the name of the class being specified. All objects defined using this template will be an instance of this class. The class may be specified by this standard, or by a user of this standard.
- (3) The "CLASS ID:" entry is a number that identifies the class being specified. This number is unique within the FAL ASE that will provide the services for this class. When qualified by the identity of its FAL ASE, it unambiguously identifies the class within the scope of the FAL. The value "NULL" indicates that the class cannot be instantiated. Class IDs between 1 and 255 are reserved by this standard to identify standardized classes. They have been assigned to maintain compatibility with existing national standards. CLASS IDs between 256 and 2048 are allocated for identifying user defined classes.
- (4) The "PARENT CLASS:" entry is the name of the parent class for the class being specified. All attributes defined for the parent class and inherited by it are inherited for the class being defined, and therefore do not have to be redefined in the template for this class.

NOTE The parent-class "TOP" indicates that the class being defined is an initial class definition. The parent class TOP is used as a starting point from which all other classes are defined. The use of TOP is reserved for classes defined by this standard.

- (5) The "ATTRIBUTES" label indicate that the following entries are attributes defined for the class.
 - a) Each of the attribute entries contains a line number in column 1, a mandatory (m) / optional (o) / conditional (c) / selector (s) indicator in column 2, an attribute type label in column 3, a name or a conditional expression in column 4, and optionally a list of enumerated values in column 5. In the column following the list of values, the default value for the attribute may be specified.
 - b) Objects are normally identified by a numeric identifier or by an object name, or by both. In the class templates, these key attributes are defined under the key attribute.
 - c) The line number defines the sequence and the level of nesting of the line. Each nesting level is identified by period. Nesting is used to specify
 - i) fields of a structured attribute (4.1, 4.2, 4.3),
 - ii) attributes conditional on a constraint statement (5). Attributes may be mandatory (5.1) or optional (5.2) if the constraint is true. Not all optional attributes require constraint statements as does the attribute defined in (5.2).
 - iii) the selection fields of a choice type attribute (6.1 and 6.2).
- (6) The "SERVICES" label indicates that the following entries are services defined for the class.
 - a) An (m) in column 2 indicates that the service is mandatory for the class, while an (o) indicates that it is optional. A (c) in this column indicates that the service is conditional. When all services defined for a class are defined as optional, at least

one has to be selected when an instance of the class is defined.

- b) The label "OpsService" designates an operational service (1).
- c) The label "MgtService" designates an management service (2).
- d) The line number defines the sequence and the level of nesting of the line. Each nesting level is identified by period. Nesting within the list of services is used to specify services conditional on a constraint statement.

3.7.4 Conventions for service definitions

3.7.4.1 Overview

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

3.7.4.2 Service parameters

Service primitives are used to represent service user/service provider interactions (ISO/IEC 10731). They convey parameters which indicate information available in the user/provider interaction. In any particular interface, not all parameters need be explicitly stated.

The service specifications of this standard use a tabular format to describe the component parameters of the ASE service primitives. The parameters which apply to each group of service primitives are set out in tables. Each table consists of up to five columns for the:

- 1) parameter name;
- 2) request primitive;
- 3) indication primitive;
- 4) response primitive; and
- 5) confirm primitive.

One parameter (or component 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 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 dynamic usage of the service user. When not provided, a default value for the parameter is assumed.
- C parameter is conditional upon other parameters or upon the environment of the service user.
- (blank) parameter is never present.
- S parameter is a selected item.

Some entries are further qualified by items in brackets. These may be

- c) 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.
- d) an indication that some note applies to the entry:

“(n)” indicates that the following note “n” contains additional information pertaining to the parameter and its use.

3.7.4.3 Service procedures

The procedures are defined in terms of:

- the interactions between application entities through the exchange of fieldbus Application Protocol Data Units; and
- the interactions between an application layer service provider and an application layer service user in the same system through the invocation of application layer service primitives.

These procedures are applicable to instances of communication between systems which support time-constrained communications services within the fieldbus application layer.

4 Concepts

4.1 Common concepts

All of IEC/TR 61158-1:2010, Clause 9 is incorporated by reference, except as specifically overridden in 4.2.

4.2 Type specific concepts

4.2.1 Operating principle

Type 22 consists of two types of communication models: RTFL and RTFN. RTFL is used to ensure synchronized cyclic real-time communication. RTFN is used in to network several RTFL cells to an overall system providing data interchange between several RTFL cells and between RTFL cells and RTFN devices.

In this context, a RTFL cell describes a DL-segment which uses RTFL for communication. An RTFL cell consists of a root device (RD) and one or several ordinary devices (OD). The central RTFL cell element is the root device which organizes and controls RTFL cell sequences such as cyclic real-time frame sending. A RTFL RD has at least one connection to RTFL, and can include a gateway (GW) which additionally has connection to RTFN. As each OD in the RTFL cell can only have a RTFL connection, the RD incorporating a GW therefore operates as a link between RTFL and RTFN. RTFN communication is not coordinated like communication in RTFL, but utilized by a switched fully duplex ISO/IEC 8802-3 network. Thus, no determinism can be guaranteed for RTFN data transfer.

Communication of process and service data is accommodated by Type 22 networks using different mechanisms (channels) in RTFL and RTFN. Cyclic data can be transferred over the cyclic data channel (CDC). The message channel (MSC) allows additional acyclic data communication and is used for service data exchange.

Service data is typically transferred acyclic and is used for transfer of parameters, control commands, status and diagnostic data as well as for generally larger data segments. Service data are transferred either event driven or user driven (acyclic character). Parameter data used in particular in device configuration do not require strict time conditions whereas diagnostic data may have much greater time requirements.

In contrast, process data is typically transferred cyclically with different cycle times and higher real-time requirements.

Type 22 AL supports a variety of services and protocols to meet these differing requirements. Both communication models support the same fieldbus application layer. The services and protocols are mapped to the corresponding DL-services.

4.2.2 Communication model overview

4.2.2.1 Overview

Type 22 technology essentially specifies two communication models with corresponding protocols. RTFL communication is intended for fast machine communication while RTFN provides for the networking of individual machines or cells. The corresponding protocols aim to offer an equal set of services for cyclic process data exchange as well as for acyclic message data communication.

The application relationship can be modeled independent of communication relationship.

4.2.2.2 Communication model RTFL

For RTFL communication model, communication follows a line topology. RTFL communication is based on cyclic data transfer in an ISO/IEC 8802-3 DLPDU. This basic cyclic data transfer is provided by a special device, the root device (RD). Root devices act as communication master to cyclically initiate communication. The DLPDUs originated by the root device are passed to the Type 22 ordinary devices (OD). Each ordinary device receives the DLPDU, writes its data and passes the DLPDU on. A RTFL network requires exactly one root device. The last ordinary device of a RTFL network sends the processed DLPDU back. The DLPDU is transferred back in reverse device order to the root device so that it is returned by the first ordinary device to the root device as response DLPDU. In backward direction, the ordinary devices read their relevant data from the DLPDU.

4.2.2.3 Communication model RTFN

For RTFN communication model, communication is based on point to point connections between participating devices.

Networking of different RTFL parts or cells of an automation system into an overall automation system is supported by the usage of RTFN communication and corresponding gateways.

4.2.3 Application layer element description

4.2.3.1 CeS

The mandatory CeS ASE consists of several attributes and depicts the main application layer element to build up a distributed real-time application.

4.2.3.2 Communication of non Type 22 DLPDUs

The optional SEF communication ASE depicts a possibility to utilize tunneled non Type 22 communication within the RTFL communication system.

4.2.3.3 Management

The mandatory management ASE consists of a set of services to control the state of a network and participating devices. Constraints in available services are specified for the different communication models RTFL and RTFN.

4.2.4 Producer-consumer interaction

The producer-consumer interaction model involves one producer and zero or more consumer(s). The model is characterized by an unconfirmed service requested by the producer and a correlated service indication in all consumers. Figure 1 illustrates the interaction for one producer and two consumers.

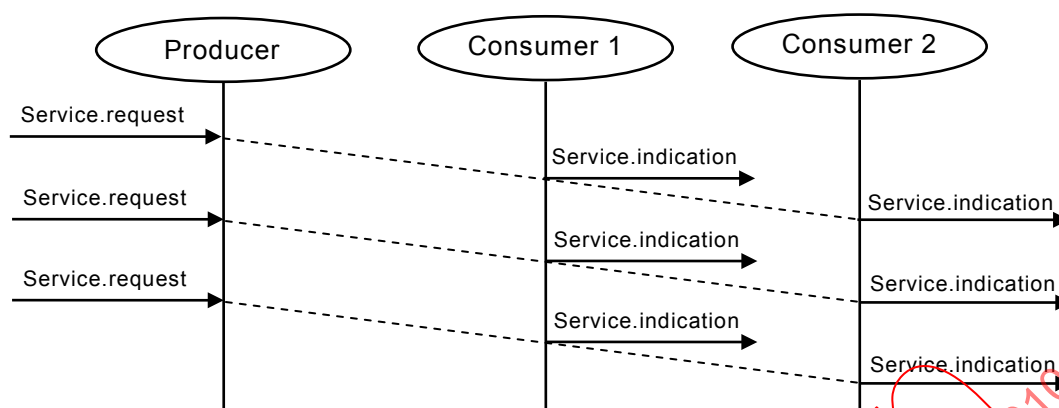


Figure 1 – Producer-consumer interaction model

The services supported by an interaction model are conveyed by application relationship endpoints (AREPs) associated with the communicating APs. The role that the AREP plays in the interaction (for example producer, consumer) is defined as an attribute of the AREP.

4.2.5 Device reference models

4.2.5.1 RTFL device reference model

Type 22 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 22 specification defines functionality from top to bottom of a full OSI model. Functions of the intermediate OSI layers, layers 3 to 6, are consolidated into either the Type 22 data-link layer or the Type 22 application layer. The device reference model for a Type 22 RTFL device is shown in Figure 2.

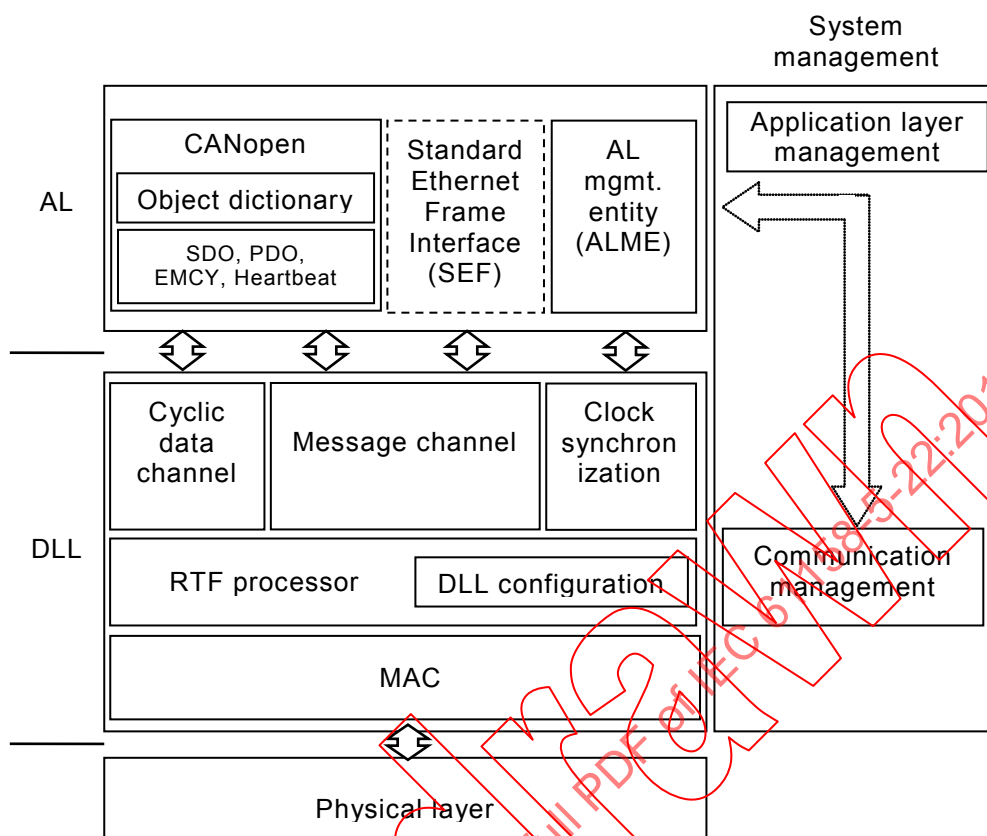


Figure 2 – RTFL device reference model

4.2.5.2 RTFN device reference model

Type 22 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 22 specification defines functionality from top to bottom of a full OSI model. Functions of the intermediate OSI layers, layers 3 to 6, are consolidated into either the Type 22 data-link layer or the Type 22 application layer. The device reference model for a Type 22 RTFN device is shown in Figure 3.

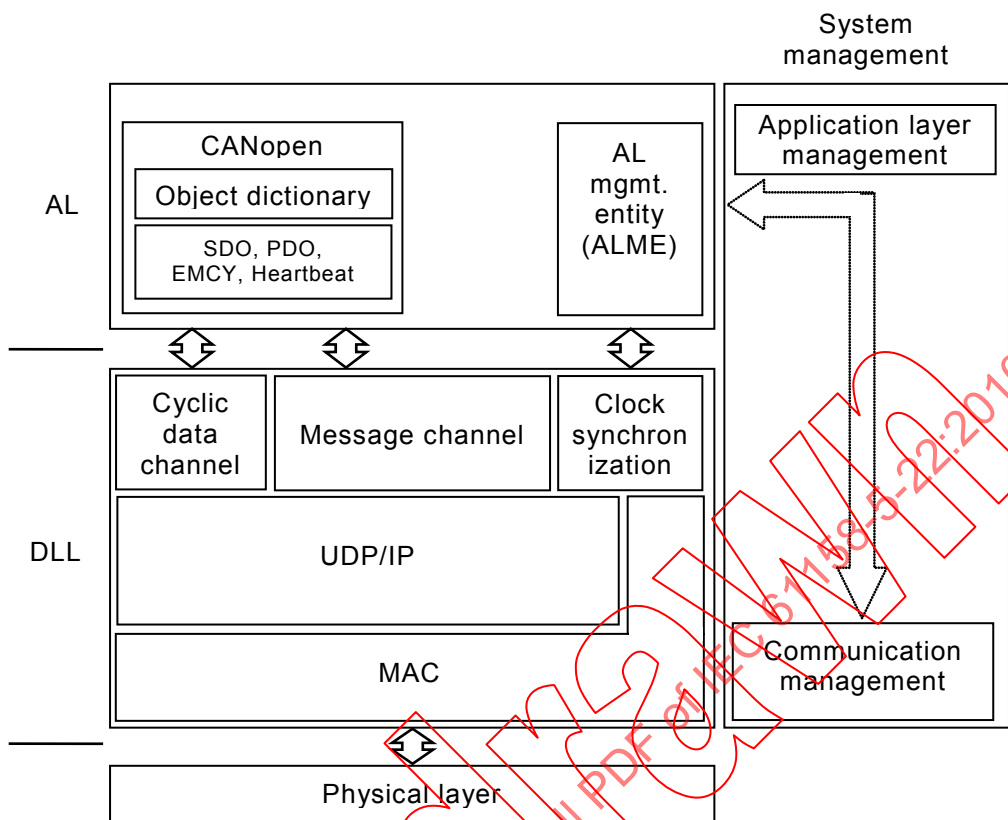


Figure 3 – RTFN device reference model

5 Data type ASE

5.1 Overview

All of IEC/TR 61158-1:2010, 10.1, is incorporated by reference.

5.2 Formal definition of data type objects

All of IEC/TR 61158-1:2010, 10.2, is incorporated by reference.

5.3 FAL defined data types

5.3.1 Fixed length types

5.3.1.1 Boolean types

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

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

5.3.1.2 Bitstring types

There are no Bitstring types defined for Type 22.

5.3.1.3 Currency types

There are no Currency types defined for Type 22.

5.3.1.4 Date/Time types

5.3.1.4.1 TimeOfDay

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

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

5.3.1.4.2 TimeDifference

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

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

5.3.1.5 Enumerated types

There are no Enumerated types defined for Type 22.

5.3.1.6 Handle types

There are no Handle types defined for Type 22.

5.3.1.7 Numeric types

5.3.1.7.1 float

This data type is the same as Float32.

5.3.1.7.2 Float32

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

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

5.3.1.7.3 double

This data type is the same as Float64.

5.3.1.7.4 Float64

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

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

5.3.1.8 Integer types

5.3.1.8.1 Integer8

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

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

5.3.1.8.2 SINT

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

5.3.1.8.3 char

This data type is the same as Integer8.

5.3.1.8.4 Integer16

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

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

5.3.1.8.5 INT

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

5.3.1.8.6 short

This data type is the same as Integer16.

5.3.1.8.7 Integer24

CLASS:		Data type
ATTRIBUTES:		
1	Data type Numeric Identifier	= 16
2	Data type Name	= Integer24
3	Format	= FIXED LENGTH
4.1	Octet Length	= 3

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

5.3.1.8.8 Integer32

CLASS:		Data type
ATTRIBUTES:		
1	Data type Numeric Identifier	= 4
2	Data type Name	= Integer32
3	Format	= FIXED LENGTH
4.1	Octet Length	= 4

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

5.3.1.8.9 DINT

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

5.3.1.8.10 long

This data type is the same as Integer32.

5.3.1.8.11 Integer40

CLASS:		Data type
ATTRIBUTES:		
1	Data type Numeric Identifier	= 18
2	Data type Name	= Integer40
3	Format	= FIXED LENGTH
4.1	Octet Length	= 5

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

5.3.1.8.12 Integer48

CLASS:		Data type
ATTRIBUTES:		
1	Data type Numeric Identifier	= 19
2	Data type Name	= Integer48
3	Format	= FIXED LENGTH
4.1	Octet Length	= 6

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

5.3.1.8.13 Integer56**CLASS:****Data type****ATTRIBUTES:**

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

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

5.3.1.8.14 Integer64**CLASS:****Data type****ATTRIBUTES:**

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

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

5.3.1.8.15 LINT

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

5.3.1.9 Unsigned types**5.3.1.9.1 Unsigned8****CLASS:****Data type****ATTRIBUTES:**

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

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

5.3.1.9.2 USINT

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

5.3.1.9.3 unsigned char

This data type is the same as Unsigned8.

5.3.1.9.4 Unsigned16**CLASS:****Data type****ATTRIBUTES:**

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

4.1 Octet Length = 2

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

5.3.1.9.5 UINT

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

5.3.1.9.6 WORD

This type is used in the same way as UINT.

5.3.1.9.7 Unsigned24

CLASS: Data type

ATTRIBUTES:

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

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

5.3.1.9.8 Unsigned32

CLASS: Data type

ATTRIBUTES:

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

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

5.3.1.9.9 UDINT

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

5.3.1.9.10 DWORD

This type is used in the same way as UDINT.

5.3.1.9.11 Unsigned40

CLASS: Data type

ATTRIBUTES:

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

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

5.3.1.9.12 Unsigned48

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

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

5.3.1.9.13 Unsigned56

CLASS:		Data type
ATTRIBUTES:		
1	Data type Numeric Identifier	= 26
2	Data type Name	= Unsigned56
3	Format	= FIXED LENGTH
4.1	Octet Length	= 7

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

5.3.1.9.14 Unsigned64

CLASS:		Data type
ATTRIBUTES:		
1	Data type Numeric Identifier	= 27
2	Data type Name	= Unsigned64
3	Format	= FIXED LENGTH
4.1	Octet Length	= 8

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

5.3.1.9.15 ULINT

This IEC 61131-3 type is the same as Unsigned64.

5.3.1.9.16 Unsigned128

CLASS:		Data type
ATTRIBUTES:		
1	Data type Numeric Identifier	= 28
2	Data type Name	= Unsigned128
3	Format	= FIXED LENGTH
4.1	Octet Length	= 16

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

5.3.1.9.17 Unsigned256

CLASS:		Data type
ATTRIBUTES:		
1	Data type Numeric Identifier	= 29
2	Data type Name	= Unsigned256
3	Format	= FIXED LENGTH
4.1	Octet Length	= 32

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

5.3.1.10 Pointer types

There are no Pointer types defined for Type 22.

5.3.1.11 OctetString types

There are no OctetString types of fixed length defined for Type 22.

5.3.1.12 VisibleString character types

There are no VisibleString types of fixed length defined for Type 22.

5.3.2 String types

5.3.2.1 OctetString

CLASS:		Data type
ATTRIBUTES:		
1	Data type Numeric Identifier	= 10
2	Data type Name	= OctetString
3	Format	= STRING
4.1	Octet Length	= 1 to n

An OctetString is an ordered sequence of octets, numbered from 1 to n.

NOTE IEC 61158-6-22:2010 defines the order of transmission.

5.3.2.2 VisibleString

CLASS:		Data type
ATTRIBUTES:		
1	Data type Numeric Identifier	= 9
2	Data type Name	= VisibleString
3	Format	= STRING
4.1	Octet Length	= 1 to n

This type is defined as the ISO/IEC 646 string type.

5.3.2.3 UnicodeString

CLASS:
Data type
ATTRIBUTES:

1	Data type Numeric Identifier	=	11
2	Data type Name	=	UnicodeString
3	Format	=	STRING
4.1	Octet Length	=	1 to n

This type is defined as the ISO/IEC 10646 string type.

5.3.3 Domain

CLASS:
Data type
ATTRIBUTES:

1	Data type Numeric Identifier	=	15
2	Data type Name	=	Domain
3	Format	=	STRING
4.1	Octet Length	=	1 to n

Large variable amount of data, for example executable program.

6 Communication model specification

6.1 Application service elements (ASEs)

6.1.1 CeS ASE

6.1.1.1 Overview

6.1.1.1.1 General information

CANopen is a non-proprietary open fieldbus standard specified and standardized by CiA (CAN in Automation) organization.

NOTE CANopen (CiA DS 301) is standardized as EN 50325-4.

In conjunction with the CAN Bus based protocol, a uniform and standardized application layer is provided for industrial applications. This includes standardization of communication, including technical and functional features allowing networking of distributed field automating devices and standardization of application objects using device profiles.

The device profiles are one of the core elements of CANopen, specifying uniform functions and standardized parameters/objects for different application areas or for automated device groups. Based on these standardized profiles, a great degree of vendor compatibility can be achieved due to interoperability and interchangeability of devices made by different manufacturers. All major device types used in automation engineering such as:

- digital and analogue I/O devices;
- drives;
- valves;
- programmable controls;
- encoders, etc.

are already standardized as device profiles and are reflected in the relevant CiA device profile specification.

The object dictionary contains parameters, application data and the mapping information between process data objects and application data (PDO mapping). Its entries can be accessed via service data objects (SDO), as shown in Figure 4.

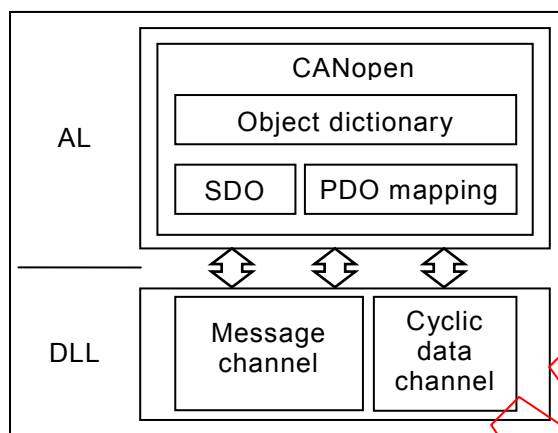


Figure 4 – Type 22 CeS device structure

6.1.1.1.2 Object dictionary structure

The object dictionary is the interface between the application and the communication sub-system. Essential as a central element, the object dictionary is a grouping of objects and specifies uniform communication and device parameters, data and functions which are stored and retrieved using objects. It is a collection of the device parameters data structures that can be accessed with the SDO Upload and SDO Download services.

The dictionary is organized in form of a table as indicated in Table 1 below. The structure corresponds to the CANopen specification 301 known from industrial automation.

NOTE CANopen (CIA DS 301) is standardized as EN 50325-4.

Table 1 – Object dictionary structure

Section	Sub-section	Content
Data type	Basic data types	Definition of basic data types
—	Complex data types	Definition of complex data types
—	Manufacturer specific data types	Definition of manufacturer specific data types
—	Device profile specific basic data types	Definition of device profile specific basic data types
—	Device profile specific complex data types	Definition of device profile specific complex data types
Communication profile	—	Definition of the parameters which are used for communication configuration and dedicated communication purposes
Manufacturer defined profile	—	Definition of manufacturer specific parameters
Standardized device profile	—	Definition of the parameters defined in a standardized device profile
Standardized interface profile	—	Definition of the parameters defined in standardized interface profile
Type 22 RTFN interface profile	—	Definition of the parameters defined in Type 22 RTFN interface profile

6.1.1.1.3 Data type areas

The Data type area consists of the following parts:

Basic data types

Definition of general simple data types

Complex data types

Definition of general structured data types

Manufacturer specific complex data types

Definition of manufacturer specific structured data types

Device profile specific basic data types

Definition of device profile specific simple data types

Device profile specific complex data types

Definition of device profile specific structured data types

6.1.1.1.4 Communication area

6.1.1.1.4.1 Device type

The device type object consists of the following parameter:

Parameter

Device profile number

This parameter specifies the device profile that is used of the device.

Additional information

This parameter specifies additional information defined by the device profile.

6.1.1.1.4.2 Error register

The error register object consists of the following parameter:

Parameter

Generic Error

This parameter indicates the presence of a generic error.

Current

This parameter indicates the presence of a current error.

Voltage

This parameter indicates the presence of a voltage error.

Temperature

This parameter indicates the presence of a temperature error.

Communication

This parameter indicates the presence of a communication error.

Device profile specific

This parameter indicates the presence of a device profile specific error.

Manufacturer specific

This parameter indicates the presence of a manufacturer specific error.

6.1.1.1.4.3 Manufacturer status register

The manufacturer status register object consists of the following parameter:

Parameter

Status register

This parameter specifies the status register of the device.

6.1.1.1.4.4 Event log

The event log object consists of the following parameter:

Parameter

Number of entries

This parameter specifies the number of entries for this object.

Emergency error code

This parameter specifies the emergency error code for the occurrence of an event.

Manufacturer specific error field

This parameter specifies the manufacturer specific error field for the occurrence of an event.

Time stamp

This parameter specifies the occurrence in time of an event.

Length

This parameter specifies length of the extended manufacturer information.

Extended manufacturer information

This parameter specifies the extended manufacturer specific information.

6.1.1.1.4.5 Manufacturer device name

The manufacturer device name object consists of the following parameter:

Parameter

Device name

This parameter specifies the device name of the device.

6.1.1.1.4.6 Manufacturer hardware version

The manufacturer hardware version object consists of the following parameter:

Parameter

Hardware version

This parameter specifies the manufacturer hardware version of the device.

6.1.1.1.4.7 Manufacturer software version

The manufacturer software version object consists of the following parameter:

Parameter

Software version

This parameter specifies the manufacturer software version of the device.

6.1.1.1.4.8 Communication layer configuration

The communication layer configuration object consists of the following parameter:

Parameter

Number of entries

This parameter specifies the number of entries for this object.

Symbolic device name

This parameter specifies the symbolic device name of the device.

Device role

This parameter specifies the role of the device within the communication system.

RTFN base cycle time

This parameter specifies the RTFN base cycle time of the device.

IP Address

This parameter specifies the IP address of the device.

Subnet mask

This parameter specifies the subnet mask of this device.

Default gateway

This parameter specifies the default gateway of the device.

DHCP enabled

This parameter specifies the usage of DHCP for the device.

Current IP configuration

This parameter specifies the activation for IP configuration.

6.1.1.1.4.9 Time sync IRQ configuration

The time sync IRQ configuration object consists of the following parameter:

Parameter

Sync ID number

This parameter specifies the device internal number of the time sync ID.

Number of entries

This parameter specifies the number of time sync IRQ configuration entries.

Time sync ID

This parameter specifies a network unique identifier for a group of synchronous devices.

Cycle time

This parameter specifies a value which triggers a specific timing signal (IRQ).

Time offset

This parameter specifies a value and describes the time offset of a device to the SYNC master.

Is master

This parameter specifies the role of the device for synchronization mechanism.

IPv4 address sync master

This parameter specifies the IP address for IPv4 of the sync master device.
IPv6 address sync master

This parameter specifies the IP address for IPv6 of the sync master device.

6.1.1.1.4.10 Time sync IRQ state

The time sync IRQ state object consists of the following parameter:

Parameter

Sync ID number

This parameter specifies the device internal number of the time sync ID.

Number of entries

This parameter specifies the number of time sync IRQ state entries.

Time sync IRQ state

This parameter specifies the state of synchronization for a particular time sync ID of the device.

6.1.1.1.4.11 Store parameters

The store parameters object consists of the following parameter:

Parameter

Number of entries

This parameter specifies the number of entries.

All parameters

This parameter specifies a command to store all device parameters.

Communication profile

This parameter specifies a command to store all communication profile parameters.

Application parameters

This parameter specifies a command to store all application parameters.

Manufacturer specific parameters

This parameter specifies a command to store manufacturer specific parameters or parameter groups.

6.1.1.1.4.12 Restore default parameters

The restore default parameters object consists of the following parameter:

Parameter

Number of entries

This parameter specifies the number of entries.

All parameters

This parameter specifies a command to restore all default device parameters.

Communication profile

This parameter specifies a command to restore all default communication profile parameters.

Application parameters

This parameter specifies a command to restore all default application parameters.

Manufacturer specific parameters

This parameter specifies a command to restore manufacturer specific default parameters or default parameters of parameter groups.

6.1.1.1.4.13 Diagnostic information

The diagnostic information object consists of the following parameter:

Parameter

Number of entries

This parameter specifies the number of entries.

Application layer state

This parameter specifies information about the application layer state.

Application state

This parameter specifies information about the application state.

CL state RTFL

This parameter specifies information about the communication layer state for RTFL.

CL state RTFN

This parameter specifies information about the communication layer state for RTFN.

Number of delayed RTFL DLPDUs

This parameter specifies information about the number of delayed RTFL DLPDUs.

Number of corrupt DLPDUs

This parameter specifies information about the number of corrupt DLPDUs.

Number of received DLPDUs since startup

This parameter specifies information about the number of received DLPDUs since startup.

Number of MSC buffer overflows

This parameter specifies information about the number of MSC buffer overflows.

Number of received MSC messages since startup

This parameter specifies information about the number of received MSC messages since startup.

Cable attenuation port 1

This parameter specifies information about the cable attenuation for port 1.

Cable attenuation port 2

This parameter specifies information about the cable attenuation for port 2.

Cable length port 1

This parameter specifies information about the cable length for port 1 cabling.

Cable length port 2

This parameter specifies information about the cable length for port 2 cabling.

Distance to fault port 1

This parameter specifies information about the distance to a cabling fault for port 1.

Distance to fault port 2

This parameter specifies information about the distance to a cabling fault for port 2.

6.1.1.1.4.14 Diagnostic thresholds

The diagnostic thresholds object consists of the following parameter:

Parameter

Number of entries

This parameter specifies the number of entries.

Expected RTFL round trip time

This parameter specifies information about the expected RTFL round trip time.

Delayed RTFL rate threshold

This parameter specifies information about the delayed RTFL rate threshold.

Corrupt DLPDU rate threshold

This parameter specifies information about the corrupt DLPDU rate threshold.

MSC buffer overflows rate threshold

This parameter specifies information about the MSC buffer overflows rate threshold.

Cable attenuation port 1 threshold

This parameter specifies information about the cable attenuation port 1 threshold.

Cable attenuation port 2 threshold

This parameter specifies information about the cable attenuation port 2 threshold.

6.1.1.1.4.15 IP address EMCY

The IP address EMCY object consists of the following parameter:

Parameter

IP address

This parameter specifies the IP address of the destination device for EMCY messages.

6.1.1.1.4.16 Inhibit time EMCY

The inhibit time EMCY object consists of the following parameter:

Parameter

Inhibit time

This parameter specifies a message inhibit time for emergency messages.

6.1.1.1.4.17 Consumer heartbeat list

The consumer heartbeat list object consists of the following parameter:

Parameter

Heartbeat producer number

This parameter specifies the device internal number of the heartbeat producer to be monitored.

Number of entries

This parameter specifies the number of consumer heartbeat list entries.

RTFL PID

This parameter specifies RTFL packet ID of the monitored heartbeat.

RTFN PID

This parameter specifies RTFN packet ID of the monitored heartbeat.

TT

This parameter specifies the transmission type of the monitored heartbeat.

Heartbeat time

This parameter specifies the heartbeat time as a multiple of the base cycle time.

Cycle multiplier

This parameter specifies the expected transmission cycle as a multiplier for the cycle time of the communication system in case of inter-cell communication.

Cycle offset

This parameter specifies an offset for the expected transmission cycle in relation to a communication system cycle in case of inter-cell communication.

Device address

This parameter specifies the device address of the heartbeat producer.

IPv4 address

This parameter specifies the IP address (IPv4) of the heartbeat producer.

IPv6 address

This parameter specifies the IP address (IPv6) of the heartbeat producer.

6.1.1.1.4.18 Producer heartbeat parameter

The producer heartbeat parameter object consists of the following parameter:

Parameter

Number of entries

This parameter specifies the number of valid entries.

RTFL PID

This parameter specifies the RTFL packet ID.

RTFN PID

This parameter specifies the RTFN packet ID.

Transmission type

This parameter specifies the transmission type of the heartbeat.

Time sync ID

This parameter specifies a time sync ID assigned to the heartbeat in case of synchronized heartbeat processing.

Cycle multiplier

This parameter specifies the transmission cycle as a multiplier of the cycle time of the communication system.

Cycle offset

This parameter specifies the transmission cycle of the heartbeat.

Device address

This parameter specifies the device address of the recipient of the heartbeat for message channel based communication.

IPv4 address

This parameter specifies the IPv4 address of the recipient of the heartbeat for message channel based inter-cell communication.

IPv6 address

This parameter specifies the IPv6 address of the recipient of the heartbeat for message channel based inter-cell communication.

6.1.1.1.4.19 Identity object

The identity object consists of the following parameter:

Parameter

Number of entries

This parameter specifies the number of entries.

Vendor ID

This parameter specifies the vendor ID of the device.

Product code

This parameter specifies the product code of the device.

Revision number

This parameter specifies the revision number of the device.

Serial number

This parameter specifies the serial number of the device.

Type 22 protocol version

This parameter specifies the Type 22 protocol version of the device.

6.1.1.1.4.20 SDO protocol timeout

The SDO protocol timeout object consists of the following parameter:

Parameter

Timeout time

This parameter specifies the SDO protocol timeout time for the device.

6.1.1.1.4.21 Enable client SDO parameter

The enable client SDO parameter object consists of the following parameter:

Parameter

Enable client SDO

This parameter specifies a command to enable client SDO for the device.

6.1.1.1.4.22 Enable EMCY

The enable EMCY object consists of the following parameter:

Parameter

Enable EMCY

This parameter specifies a command to enable EMCY for the device.

6.1.1.1.4.23 PDO timeout tolerance

The PDO timeout tolerance object consists of the following parameter:

Parameter

Timeout tolerance

This parameter specifies the PDO timeout tolerance for the device.

6.1.1.1.4.24 Store EDS

The store EDS object consists of the following parameter:

Parameter

Store EDS

This parameter specifies the EDS file of the device.

6.1.1.1.4.25 Storage format

The storage format object consists of the following parameter:

Parameter

Format

This parameter specifies the format of the file of the device.

6.1.1.1.4.26 OS command

The OS command object consists of the following parameter:

Parameter

Number of entries

This parameter specifies the number of entries.

Command

This parameter specifies an OS command.

Status

This parameter specifies a status of the command execution.

Reply

This parameter specifies a reply to the command.

6.1.1.1.4.27 OS command mode

The OS command mode object consists of the following parameter:

Parameter

Command mode

This parameter specifies the OS command execution.

6.1.1.1.4.28 OS debugger interface

The OS debugger interface object consists of the following parameter:

Parameter

Number of entries

This parameter specifies the number of entries.

Command

This parameter specifies an OS command.

Status

This parameter specifies a status for the command execution.

Reply

This parameter specifies a reply to the command executed.

6.1.1.1.4.29 OS prompt

The OS prompt object consists of the following parameter:

Parameter

Number of entries

This parameter specifies the number of entries.

StdIn

This parameter specifies OS standard input channel.

StdOut

This parameter specifies OS standard output channel.

StdErr

This parameter specifies OS standard error channel.

6.1.1.1.4.30 Module list

The module list object consists of the following parameter:

Parameter

Number of entries

This parameter specifies the number of connected modules.

Module

This parameter specifies the module type and characteristic of connected modules of a modular device.

6.1.1.1.4.31 Emergency subscriber

The module list object consists of the following parameter:

Parameter

Number of entries

This parameter specifies the number of subscribed emergency messages.

Device address

This parameter specifies the device address of the emergency message producer.

IP address

This parameter specifies the IP address of the emergency message producer.

Additional information

This parameter specifies additional information for a producer.

6.1.1.1.4.32 Client SDO parameter

The client SDO parameter object consists of the following parameter:

Parameter**Number of entries**

This parameter specifies the number of entries.

Server address

This parameter specifies the server address.

6.1.1.1.4.33 Receive PDO communication parameter

The receive PDO communication parameter object consists of the following parameter:

Parameter**Number of entries**

This parameter specifies the number of valid entries.

RTFL PID

This parameter specifies the RTFL packet ID.

RTFN PID

This parameter specifies the RTFN packet ID.

Transmission type

This parameter specifies the transmission type of the PDO.

Time sync ID

This parameter specifies a time sync ID assigned to the RxPDO in case of synchronized RxPDO processing.

Timeout

This parameter specifies a timeout time for the PDO.

Cycle multiplier

This parameter specifies the expected transmission cycle as a multiplier for the cycle time of the communication system in case of inter-cell communication.

Cycle offset

This parameter specifies an offset for the expected transmission cycle in relation to a communication system cycle in case of inter-cell communication.

Device address

This parameter specifies the device address of the producer of the PDO for message channel based communication.

IPv4 address

This parameter specifies the IPv4 address of the producer of the PDO for message channel based inter-cell communication.

IPv6 address

This parameter specifies the IPv6 address of the producer of the PDO for message channel based inter-cell communication.

6.1.1.1.4.34 Receive PDO mapping parameter

The receive PDO mapping parameter object consists of the following parameter:

Parameter

PDO number

This parameter specifies the number of the PDO.

Number of mapping entries

This parameter specifies the number of mapping entries.

List of mapping entries

This parameter specifies a list of mapping entries.

6.1.1.1.4.35 Transmit PDO communication parameter

The transmit PDO communication parameter object consists of the following parameter:

Parameter

Number of entries

This parameter specifies the number of valid entries.

RTFL PID

This parameter specifies the RTFL packet ID.

RTFN PID

This parameter specifies the RTFN packet ID.

Transmission type

This parameter specifies the transmission type of the PDO.

Time sync ID

This parameter specifies a time sync ID assigned to the TxPDO in case of synchronized TxPDO processing.

Cycle multiplier

This parameter specifies the transmission cycle as a multiplier of the cycle time of the communication system.

Cycle offset

This parameter specifies the transmission cycle of the PDO.

Device address

This parameter specifies the device address of the consumer of the PDO for message channel based communication.

IPv4 address

This parameter specifies the IPv4 address of the consumer of the PDO for message channel based inter-cell communication.

IPv6 address

This parameter specifies the IPv6 address of the consumer of the PDO for message channel based inter-cell communication.

6.1.1.1.4.36 Transmit PDO mapping parameter

The transmit PDO mapping parameter object consists of the following parameter:

Parameter

PDO number

This parameter specifies the number of the PDO.

Number of mapping entries

This parameter specifies the number of mapping entries.

List of mapping entries

This parameter specifies a list of mapping entries.

6.1.1.1.5 SDO interactions

6.1.1.1.5.1 SDO services

Object dictionary access is handled by service data object (SDO) services. Object entries in remote object dictionaries can be read or written (upload and download) using the appropriate SDO services. SDOs are data formats (structures) of any size which are addressable. The addressing follows the CANopen object dictionary addressing format and consists of a 16 bit index and 8 bit sub-index, used to address the appropriate object dictionary entries. Statement of index and sub-index is also referred to as multiplexor.

In order to provide concurrent SDO service processing in a device, Type 22 introduces the client-specified JobID parameter. This JobID is assigned to each single SDO communication. The server uses this JobID in responses. One JobID is valid for the entire protocol sequence.

SDO communication is based on the client-server model and is implemented using a point-to-point link between two devices. Type 22 CoS does not require a segmented SDO transport protocol for standard SDO services for the transfer of larger data volumes. The segmenting function is provided by the Type 22 DLL which transfers large data volumes of any length across the network.

SDO communication is confirmed, receiving an SDO message is acknowledged by an appropriate SDO message. An unconfirmed abort service records any SDO communication error. SDO communication starts with an initialization phase preparing client and server for the transmission. In case of little data volume, the data is transmitted directly in the initialization phase and the communication terminates with the subsequent acknowledgement of the server. This mechanism is called expedited transfer. If large SDOs are transferred, no data is transferred during the initialization phase. This mechanism is called normal transfer. Following the acknowledgement in the connection setup, which ensures that data can be received, the actual data transfer phase with the appropriate SDO services commences.

The primitives of the SDO services are mapped to the primitives of the message channel services.

6.1.1.1.5.2 SDO download sequence

The SDO download service is used to write data from a client to the server. The services are acknowledged by the server and return a request success status. In case of an error they return an additional error message containing the reason for error. The reason for the error is notified to the client using SDO abort service.

Figure 5 shows a successful SDO expedited download sequence between client and server and the primitives between client and server.

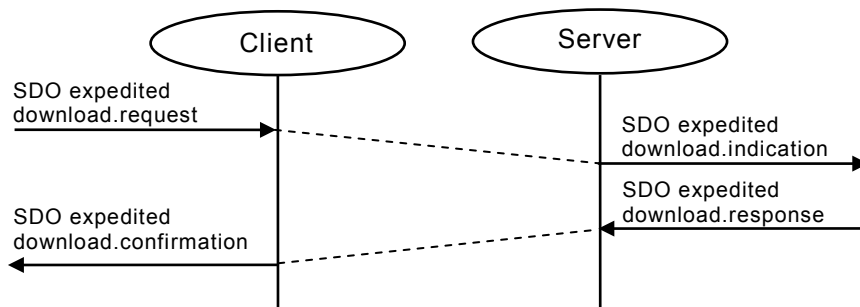


Figure 5 – Successful SDO expedited download sequence

Figure 6 shows a successful SDO normal download initialization sequence and the primitives between client and server. This transfer includes no data and has to be continued with SDO download service.

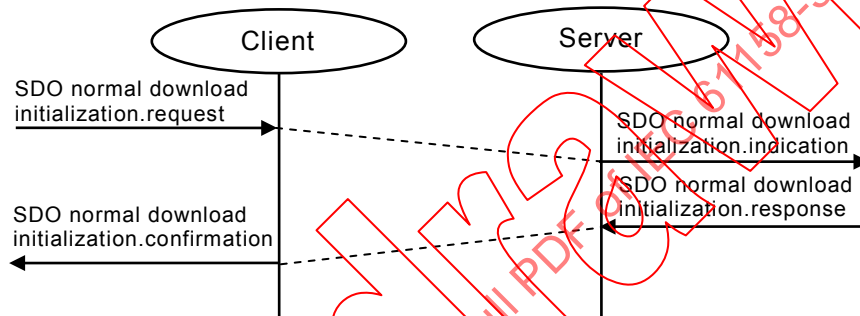


Figure 6 – Successful SDO normal download initialization sequence

Figure 7 shows a successful SDO download sequence and the primitives between client and server. This sequence follows after the successful SDO normal download initialization sequence.

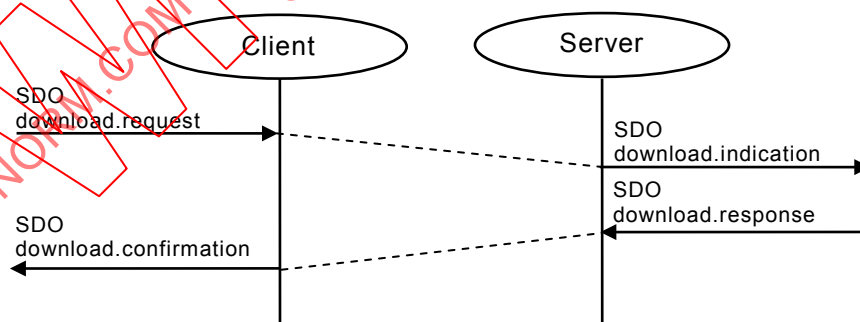


Figure 7 – Successful SDO download sequence

6.1.1.1.5.3 SDO upload sequence

The client uses the SDO upload service to read data from the server. The services are acknowledged and return a request success status as well as the requested data. In case of an error the reason is returned. This reason is notified to the client using SDO abort service.

Figure 8 shows a successful SDO expedited upload sequence and the primitives between client and server.

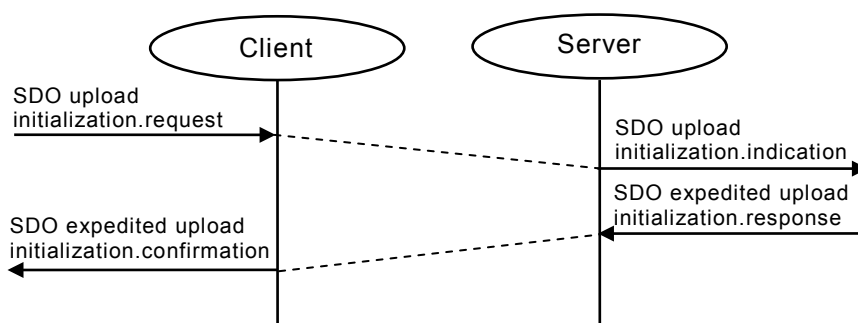


Figure 8 – Successful SDO expedited upload sequence

Figure 9 shows a successful SDO normal upload initialization sequence and the primitives between client and server. This transfer includes no data and has to be continued with SDO upload service.

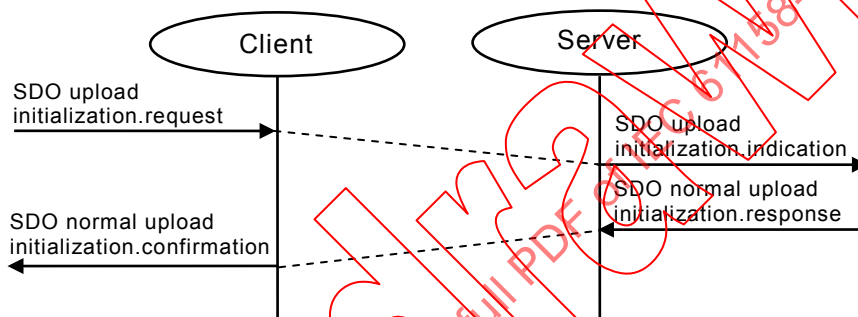


Figure 9 – Successful SDO normal upload initialization sequence

Figure 10 shows a successful SDO upload sequence and the primitives between client and server. This sequence follows a successful initialization sequence.

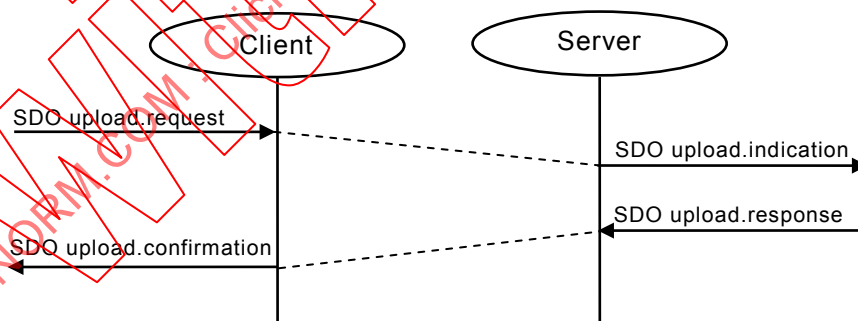


Figure 10 – Successful SDO upload sequence

6.1.1.1.5.4 SDO abort

This service aborts the SDO upload or download attempt whereby either client or server can request this service. The SDO abort service is unconfirmed and additionally contains a coded error description.

Figure 11 shows a failed SDO expedited download initialization sequence and the primitives between client and server. The server requests SDO abort service. The abort sequences for failed SDO normal download initialization or SDO upload initialization sequences are identical.

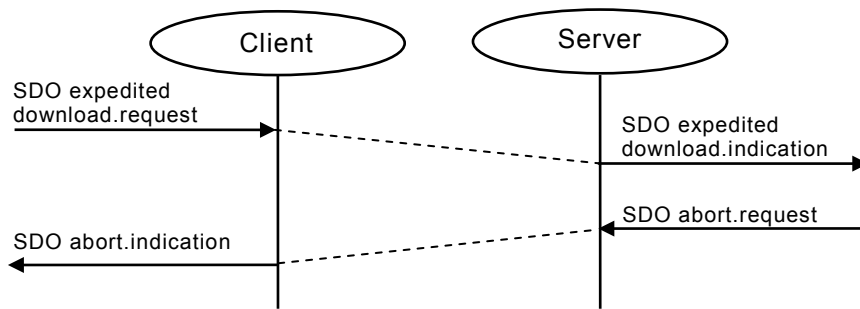


Figure 11 – Failed SDO expedited download initialization sequence

Figure 12 shows a download access error message initiated by the client to the server after a successful SDO normal download initialization sequence. The abort service sequence following an upload access is identical.

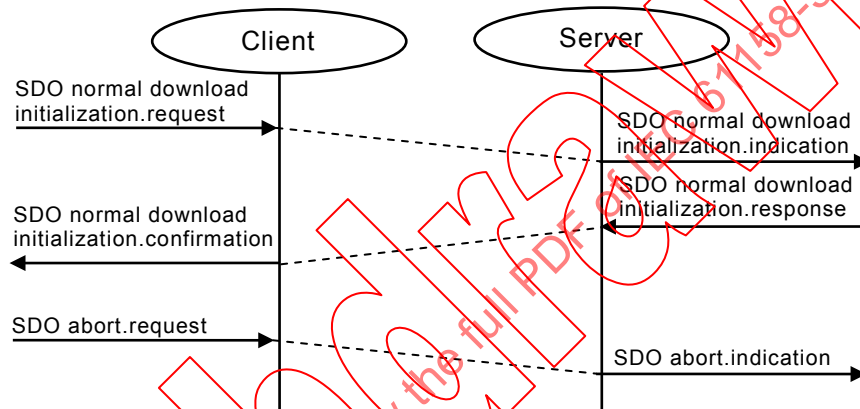


Figure 12 – Failed SDO download after initialization sequence

Figure 13 shows a failed SDO download sequence which can follow a successful SDO normal download initialization sequence. Due to an error, the server requests the SDO abort service after data reception.

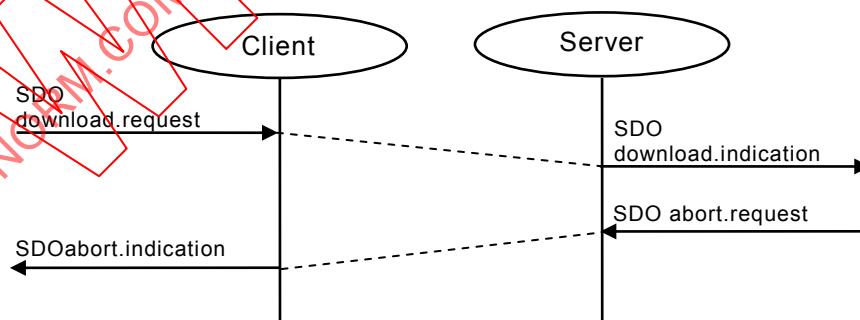


Figure 13 – Failed SDO download sequence

6.1.1.1.6 Emergency

Emergency (EMCY) messages are triggered by the occurrence of a device internal error situation. The primitives of the emergency service are mapped to the primitives of the message channel services.

Figure 14 shows the primitives between producer and consumer.

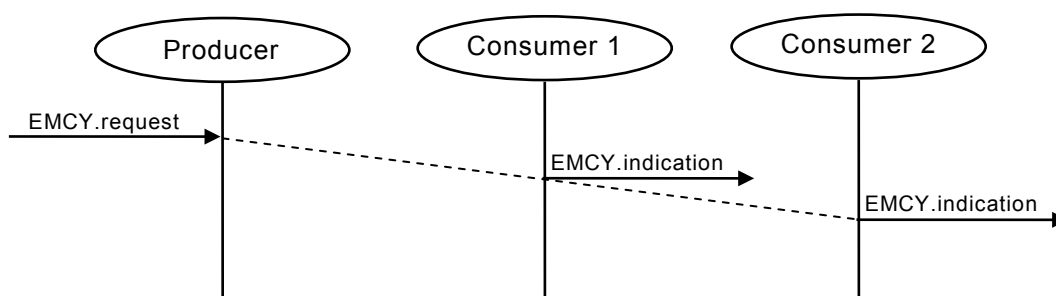


Figure 14 – Emergency sequence

6.1.1.1.7 Heartbeat

Devices can monitor each other by means of the heartbeat mechanism. Heartbeat producers generate life-signs which are received by heartbeat consumers. A heartbeat consumer can configure in the object dictionary which heartbeat producers it wants to monitor. If a heartbeat fails to arrive, a heartbeat event is generated at the heartbeat consumer.

The primitives of the heartbeat service can be mapped to CDC or to MSC primitives. This is configuration dependent. Figure 15 illustrates the functional principle of the heartbeat service.

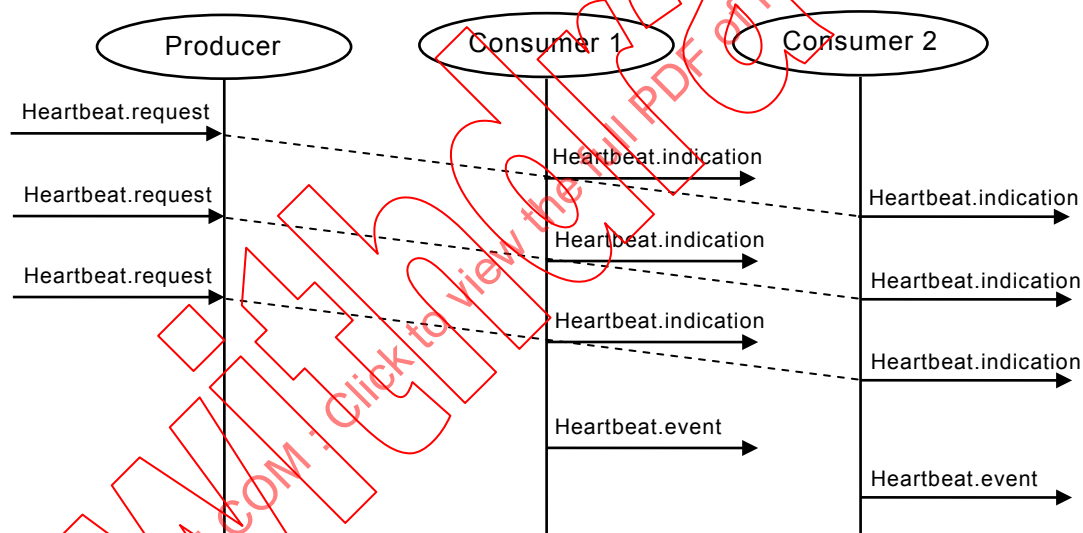


Figure 15 – Heartbeat sequence

6.1.1.1.8 Process data service

6.1.1.1.8.1 Process data interaction

The process data write service is used by a PDO producer to send the data of the mapped application objects to the PDO consumers. The process data write service indicates the PDO consumer that a valid PDO has been received.

The primitives of the process data service can be mapped to the to CDC or MSC service primitives depending on the communication parameters of a particular PDO. Figure 16 shows the process data write service sequence.

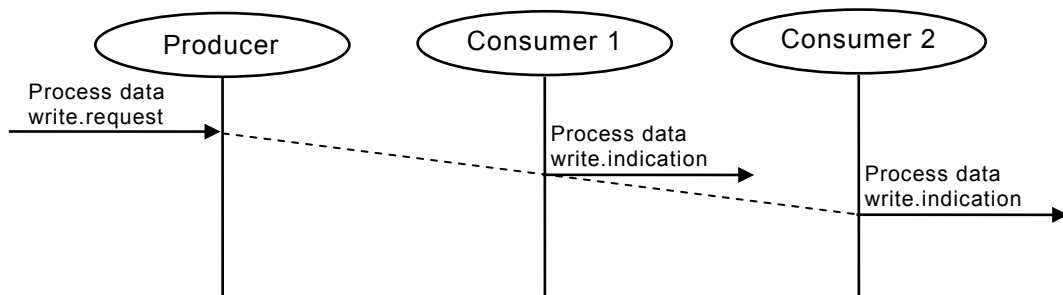


Figure 16 – Process data write sequence

6.1.1.1.8.2 Process data object communication parameters

The PDO communication parameters describe a PDO with regard to its communication characteristics. The transmission type parameter identifies the PDO transfer type. The communication channel (CDCL / CDCN / MSCL / MSCN) over which the PDO is to be transmitted or received and the communication mode, either cyclic, synchronized or event driven (acyclic) can be specified for a PDO. Figure 17 shows an example of a PDO configuration including communication and mapping parameters.

6.1.1.1.8.3 Mapping of objects to process data

The content of process data objects (PDOs) is described in the object dictionary and consists of application objects out of the object dictionary which can be mapped to PDOs. The PDO mapping objects depict lists of object references together with the length information. A valid PDO shall contain at least one application object and at most 254 application objects.

The current PDO mapping can be read by means of corresponding objects in the object dictionary. These are known as mapping objects. The first entry in a mapping object (sub-index 0) specifies the number of mapped application objects that are listed after it. The tables are located in the object dictionary at index 0x1600 to 0x17FF for the RxPDOs and at 0x1A00 to 0x1BFF for the TxPDOs. Figure 17 shows an example of a PDO configuration including communication and mapping parameters.

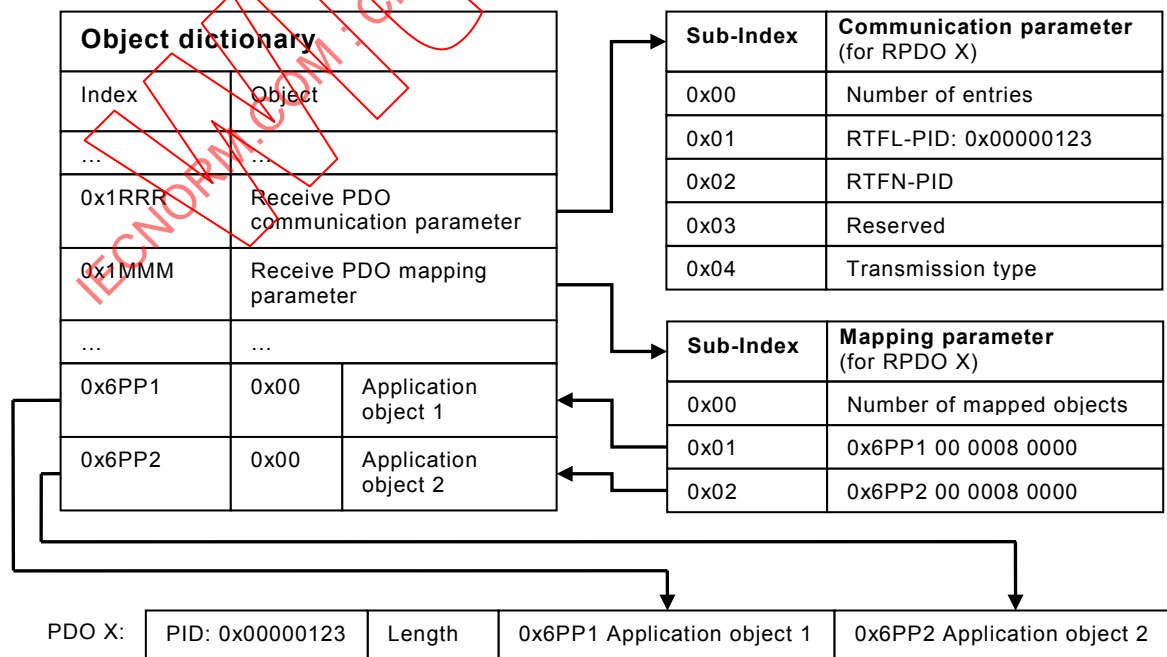


Figure 17 – PDO mapping principle

6.1.1.1.8.4 Process data objects

The general structure of a process data object is depicted in Figure 18 and corresponds to the CDC DLPPDU data as specified in IEC 61158-4-22:2010, 5.7.

PDO:	PID: 0x00000123	Length	0x6PP1 Application object 1	0x6PP2 Application object 2
------	-----------------	--------	-----------------------------	-----------------------------

Figure 18 – Process data object

6.1.1.2 CeS class specification

6.1.1.2.1 Formal model

FAL ASE:	CeS
CLASS:	CeS object dictionary
CLASS ID:	not used
PARENT CLASS:	TOP

ATTRIBUTES:

1	(m)	Key Attribute:	Implicit
1	(m)	Attribute:	Entry list
1.1	(m)	Attribute:	Index
1.2	(m)	Attribute:	Object type
1.3	(o)	Attribute:	Name and description
1.4	(m)	Attribute:	List of entry
1.4.1	(m)	Attribute:	Sub-index
1.4.2	(o)	Attribute:	Description
1.4.3	(m)	Attribute:	Data type
1.4.4	(o)	Attribute:	Access attribute
1.4.5	(o)	Attribute:	PDO mapping
1.4.6	(o)	Attribute:	Value range
1.4.7	(o)	Attribute:	Default value

SERVICES:

1	(o)	OpService:	Initiate SDO expedited download
2	(o)	OpService:	Initiate SDO normal download
3	(o)	OpService:	SDO download
4	(o)	OpService:	Initiate SDO expedited upload
5	(o)	OpService:	Initiate SDO normal upload
6	(o)	OpService:	SDO upload
7	(c)	Constraint:	Any SDO services supported
7.1	(m)	OpService:	SDO abort
8	(m)	OpService:	Process data write
9	(o)	OpService:	Emergency
10	(m)	OpService:	Heartbeat

6.1.1.2.2 Attributes

Implicit

The attribute indicates that the object is implicitly addressed by the service.

Entry list

One object is composed of the following list elements:

Index

This attribute specifies a numerical index of the object.

Object type

This attribute specifies a characterising object type of an object.

Name and description

This attribute specifies a name uniquely describing the object.

List of entry

This attribute consists of the following attributes.

Sub-index

This attribute specifies a numerical index to identify individual elements of an object.

Description

This attribute specifies a name of an element of the object.

Data type

This attribute specifies a data type of an element of the object.

Access attribute

This attribute specifies the access right of an element of the object. Its allowed values are:

- RW (Read and write access)
- WO (Write only)
- RO (Read only)

PDO mapping

This attribute specifies if an element of the object can be mapped into a PDO.

Value range

This attribute specifies value range or data type of the entire range of an element of the object.

Default value

This attribute specifies the default value on device initialization of an element of the object.

6.1.1.2.3 Services**Initiate SDO expedited download**

This service writes small data volumes (configuration dependent) to the server.

Initiate SDO normal download

This service is used for write access initialization and requests to prepare for SDO transfer.

SDO download

This service is used for data transfer to the server.

Initiate SDO expedited upload

This service reads small data volumes (configuration dependent) from the server.

Initiate SDO normal upload

This service is used for read access initialization and requests to prepare for SDO transfer.

SDO upload

This service is used for data transfer from the server.

SDO abort

Abort of service by client or server in case of an error.

Process data write

This service is used for transfer of process data.

Emergency

Service provides for diagnostic messages on occurrence of errors or unexpected conditions.

Heartbeat

This service provides for device monitoring.

6.1.1.3 CeS service specification**6.1.1.3.1 Supported services**

The CeS ASE defines the services

Initiate SDO expedited download

Initiate SDO normal download

SDO download

Initiate SDO expedited upload

Initiate SDO normal upload

SDO upload

SDO abort

Process data write

Emergency

Heartbeat

6.1.1.3.2 Initiate SDO expedited download service

This service depicts the expedited transfer initialization and is only used by the client if the data volume is small enough to be transferred directly. The maximum data volume is configuration dependent. The service requires the data write location in the server object dictionary with its multiplexor consisting of index and sub-index and the data. The server shall respond to the client's request or acknowledge the arrival of the SDO. SDO transfer is completed with receipt of the positive confirmation by the client. The service return parameter returns the communication success status to the client. In case of server error, the response must use the SDO abort service. Table 2 shows the service primitives and parameter of the service.

Table 2 – Initiate SDO expedited download service

Argument				
AREP	M	M		
Address	M	M (=)		
JobID	M	M (=)		
Index	M	M (=)		
Sub-index	M	M (=)		
Data	M	M (=)		
Result (+)			S	S (=)
Address			M	M
JobID			M	M (=)
Result (-)			S	S (=)
NOTE The method by which a confirm primitive is correlated with its corresponding preceding request primitive is a local matter. See 1.2.				

Argument

The argument shall convey the service specific parameters of the service request.

AREP

This parameter is the local identifier for the desired AR.

Address

This parameter specifies the address of the server for a request and the address of the client for indication.

JobID

This parameter specifies an ID selected by the client for a dedicated job to allow concurrent processing of several jobs.

Index

This parameter specifies the index in the server object dictionary which is to be downloaded.

Sub-index

This parameter specifies the sub-index in the server object dictionary which is to be downloaded.

Data

This parameter specifies the object value to be downloaded.

Result(+)

This selection type parameter indicates that the service request succeeded.

Address

This parameter specifies the address of the client for response and the address of the server for a confirmation.

JobID

This parameter specifies an ID selected by the client for a dedicated job to allow concurrent processing of several jobs.

Result(-)

This selection type parameter indicates that the service request failed.

6.1.1.3.3 Initiate SDO normal download

This service is used for the server object dictionary write access initialization phase and requests the server to prepare for SDO transfer. Normal indicates normal transfer initialization. This is used by the client if the volume of transfer data is large and the abort conditions are to be checked prior to the actual transfer. The service requires the data write location in the server object dictionary with its multiplexor consisting of index and sub-index, and the data. The server shall respond to the client's request. The service return parameter returns the communication success status to the client. In case of server error, the response must use the SDO abort service. Following successful initialization, this service is always succeeded by the SDO download service. Table 3 shows the service primitives and parameter of the service.

Table 3 – Initiate SDO normal download service

Parameter name	Req	Ind	Rsp	Cnf
Argument				
AREP	M	M		
Address	M	M		
JobID	M	M (=)		
Index	M	M (=)		
Sub-index	M	M (=)		
Size	M	M (=)		
Result (+)			S	S (=)
Address			M	M
JobID			M	M (=)
Result (-)			S	S (=)
NOTE The method by which a confirm primitive is correlated with its corresponding preceding request primitive is a local matter. See 4.2.				

Argument

The argument shall convey the service specific parameters of the service request.

AREP

This parameter is the local identifier for the desired AR.

Address

This parameter specifies the address of the server for a request and the address of the client for indication.

JobID

This parameter specifies an ID selected by the client for a dedicated job to allow concurrent processing of several jobs.

Index

This parameter specifies the index in the server object dictionary which is to be downloaded.

Sub-index

This parameter specifies the sub-index in the server object dictionary which is to be downloaded.

Size

This parameter specifies the data volume.

Result(+)

This selection type parameter indicates that the service request succeeded.

Address

This parameter specifies the address of the client for a response and the address of the server for confirmation.

JobID

This parameter specifies an ID selected by the client for a dedicated job to allow concurrent processing of several jobs.

Result(-)

This selection type parameter indicates that the service request failed.

6.1.1.3.4 SDO download

This service is the successor service of a normal-labeled initialization sequence and transfers the actual data to the server. The data write location via multiplexor is no longer required for this service as this has already occurred in the initialization phase. The server shall acknowledge the receipt of data to the client. The service return parameter returns the communication success status to the client. In case of server error, the response shall use the SDO abort service. SDO transfer is completed with the reception of the positive confirmation by the client. Table 4 shows the service primitives and parameter of the service.

Table 4 – SDO download service

Parameter name	Req	Ind	Rsp	Cnf
Argument				
AREP	M	M		
Address	M	M		
JobID	M	M (=)		
Data	M	M (=)		
Result (+)			S	S (=)
Address			M	M
JobID			M	M (=)
Result (-)			S	S (=)
NOTE The method by which a confirm primitive is correlated with its corresponding preceding request primitive is a local matter. See 1.2.				

Argument

The argument shall convey the service specific parameters of the service request.

AREP

This parameter is the local identifier for the desired AR.

Address

This parameter specifies the address of the server for a request and the address of the client for indication.

JobID

This parameter specifies an ID selected by the client for a dedicated job to allow concurrent processing of several jobs.

Data

This parameter specifies the data to be downloaded.

Result(+)

This selection type parameter indicates that the service request succeeded.

Address

This parameter specifies the address of the client for a response and the address of the server for confirmation.

JobID

This parameter specifies an ID selected by the client for a dedicated job to allow concurrent processing of several jobs.

Result(-)

This selection type parameter indicates that the service request failed.

6.1.1.3.5 Initiate SDO expedited upload

This service is used for the client read access initialization sequence to the server object dictionary and requests the server to prepare for SDO transfer. In this case, the server responds with expedited upload response in which the data is already transferred and which completes the upload protocol. The service requires the data read location in the server object dictionary with its multiplexor consisting of index and sub-index. The service return parameter returns the communication success status and the requested data to the client. In case of a server error, the response must use the SDO abort service. Table 5 shows the service primitives and parameter of the service.

Table 5 – Initiate SDO expedited upload service

Parameter name	Req	Ind	Rsp	Cnf
Argument				
AREP	M	M		
Address	M	M		
JobID	M	M (=)		
Index	M	M (=)		
Sub-index	M	M (=)		
Result (+)			S	S (=)
Address			M	M
JobID			M	M (=)
Data			M	M (=)
Result (-)			S	S (=)
NOTE The method by which a confirm primitive is correlated with its corresponding preceding request primitive is a local matter. See 1.2.				

Argument

The argument shall convey the service specific parameters of the service request.

AREP

This parameter is the local identifier for the desired AR.

Address

This parameter specifies the address of the server for a request and the address of the client for indication.

JobID

This parameter specifies an ID selected by the client for a dedicated job to allow concurrent processing of several jobs.

Index

This parameter specifies the index in the server object dictionary which is to be uploaded.

Sub-index

This parameter specifies the sub-index in the server object dictionary which is to be uploaded.

Result(+)

This selection type parameter indicates that the service request succeeded.

Address

This parameter specifies the address of the client for a response and the address of the server for confirmation.

JobID

This parameter specifies an ID selected by the client for a dedicated job to allow concurrent processing of several jobs.

Data

This parameter specifies the data to be transferred.

Result(-)

This selection type parameter indicates that the service request failed.

6.1.1.3.6 Initiate SDO normal upload

This service is used for the client read access initialization sequence to the server object dictionary and requests the server to prepare for SDO transfer. The server responses the transfer as normal as the data volume to be transferred does not allow expedited transfer. The service requires the data read location in the server object dictionary with its multiplexor consisting of index and sub-index. The service return parameter returns the communication success status and, optionally, the requested data volume. In case of server error, the response must use the SDO abort service. Following the successful initialization, this service is always succeeded by the SDO upload service. Table 6 shows the service primitives and parameter of the service.

Table 6 – Initiate SDO normal upload service

Parameter name	Req	Ind	Rsp	Cnf
Argument				
ARSP	M	M		
Address	M	M		
JobID	M	M (=)		
Index	M	M (=)		
Sub-index	M	M (=)		
Result (+)			S	S (=)
Address			M	M
JobID			M	M (=)
Size			O	O (=)
Result (-)			S	S (=)

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

Argument

The argument shall convey the service specific parameters of the service request.

AREP

This parameter is the local identifier for the desired AR.

Address

This parameter specifies the address of the server for a request and the address of the client for indication.

JobID

This parameter specifies an ID selected by the client for a dedicated job to allow concurrent processing of several jobs.

Index

This parameter specifies the index in the server object dictionary which is to be uploaded.

Sub-index

This parameter specifies the sub-index in the server object dictionary which is to be uploaded.

Result(+)

This selection type parameter indicates that the service request succeeded.

Address

This parameter specifies the address of the client for a response and the address of the server for confirmation.

JobID

This parameter specifies an ID selected by the client for a dedicated job to allow concurrent processing of several jobs.

Size

This parameter specifies the data volume.

Result(-)

This selection type parameter indicates that the service request failed.

6.1.1.3.7 SDO upload

This service is the successor service of a normal-labeled initialization sequence and requests the actual data from the server. The data read location via multiplexor is not required for this service as this has already occurred in the initialization phase. The service return parameter returns the communication success status and the requested data to the client. In case of server error, the response must use the SDO abort service. SDO transfer is completed with the reception of the positive confirmation by the client. Table 7 shows the service primitives and parameter of the service.

Table 7 – SDO upload service

Parameter name	Req	Ind	Rsp	Cnf
Argument				
AREP	M	M		
Address	M	M		
JobID	M	M (=)		
Result (+)			S	S (=)
Address			M	M
JobID			M	M (=)
Data			M	M (=)
Result (-)			S	S (=)
NOTE The method by which a confirm primitive is correlated with its corresponding preceding request primitive is a local matter. See 1.2.				

Argument

The argument shall convey the service specific parameters of the service request.

AREP

This parameter is the local identifier for the desired AR.

Address

This parameter specifies the address of the server for a request and the address of the client for indication.

JobID

This parameter specifies an ID selected by the client for a dedicated job to allow concurrent processing of several jobs.

Result(+)

This selection type parameter indicates that the service request succeeded.

Address

This parameter specifies the address of the client for a response and the address of the server for confirmation.

JobID

This parameter specifies an ID selected by the client for a dedicated job to allow concurrent processing of several jobs.

Data

This parameter specifies the data to be transferred.

Result(-)

This selection type parameter indicates that the service request failed.

6.1.1.3.8 SDO abort

This service aborts the SDO read or write attempt whereby either client or server can execute this service. The abort service is unconfirmed and additionally contains a coded error description. Table 8 shows the service primitives and parameter of the service.

Table 8 – SDO abort service

Parameter name	Req	Ind
Argument		
AREP	M	M
Address	M	M
JobID	M	M (=)
Abort code	M	M (=)

Argument

The argument shall convey the service specific parameters of the service request.

AREP

This parameter is the local identifier for the desired AR.

Address

This parameter specifies the address of the server for a request and the address of the client for indication in the case of a service request by a client or the address of the client for a request and the address of the server for indication in the case of a service request by a server.

JobID

This parameter specifies an ID selected by the client for a dedicated job to allow concurrent processing of several jobs.

Abort code

This parameter specifies an error code which characterizes the error.

6.1.1.3.9 Process data write

Process data communication in CeS is handled by process data objects (PDO). In CeS, PDOs are logic data containers transferred over the network or communication sub-system allowing process data communication between individual Type 22 devices. PDOs encapsulate application objects from the object dictionary. The DL-services which are used for transmission of a certain PDO shall be specified in the PDO transmission type parameters for each PDO.

Two types of PDOs are differentiated:

- Transmit PDOs (TxPDO); and
- Receive PDOs (RxPDO).

In the context of Type 22 networks, devices transmitting PDOs are PDO producers while devices receiving PDOs are PDO consumer. PDOs are described by PDO communication parameters and mapping parameters. The paired communication parameters and mapping parameters are mandatory for any supported PDO. The PDO communication parameters define the communication behavior with regard to the DLL. The PDO mapping parameters define the content of a PDO and must include at least one application object. The PDO length depends on the application and is defined in the appropriate mapping parameter. PDO configuration can be transmitted using SDO services.

Table 9 shows the service primitives and parameter of the service.

Table 9 – Process data write service

Parameter name	Req	Ind
Argument		
AREP	M	M
PDO number	M	M

Argument

The argument shall convey the service specific parameters of the service request.

AREP

This parameter is the local identifier for the desired AR.

PDO number

This parameter specifies a number which refers to a TxPDO for a request and to an RxPDO for an indication defined within the object dictionary.

6.1.1.3.10 Emergency service (EMCY)

The emergency service (EMCY) provides diagnostic messages for devices on occurrence of an internal error in a device. The diagnostic status is transferred using an emergency message. Only one emergency message is transmitted for each error event. If an error is no longer present, an error reset or no error coded emergency message is sent. The occurrence of errors is stored using the event log object described in 6.1.1.1.4.4. Table 10 shows the service primitives and parameter of the emergency service.

Table 10 – Emergency service (EMCY)

Parameter name	Req	Ind
Argument		
AREP	M	M
Destination device address	M	M (=)
Destination device IP address	O	O (=)
Emergency error code	M	M (=)
Error register	M	M (=)
Manufacturer specific error code	M	M (=)
Time stamp	O	O (=)
Length	O	O (=)
Extended manufacturer information	O	O (=)

Argument

The argument shall convey the service specific parameters of the service request.

AREP

This parameter is the local identifier for the desired AR.

Destination device address

This parameter specifies the device address of the destination device.

Destination device IP address

This parameter specifies the device IP address of the destination device.

Emergency error code

This parameter shall contain a standardized error code.

Error register

This parameter specifies the emergency error register.

Manufacturer specific error code

This parameter shall contain a manufacturer specific error code.

Time stamp

This optional parameter specifies the time passed since a configurable relative or absolute time.

Length

This parameter shall contain the length of the extended manufacturer information field in octets.

Extended manufacturer information

This parameter shall contain a more detailed description of the error or any additional information on the error.

6.1.1.3.11 Heartbeat service

The heartbeat service allows devices to monitor each other by means of heartbeat messages. A heartbeat producer generates a life-sign which is received by the heartbeat consumers. A heartbeat consumer can configure in the object dictionary which heartbeat producers it wants to monitor. If a heartbeat fails to arrive, a heartbeat event is generated at the heartbeat consumer. Table 11 shows the service primitives and parameter of the service.

Table 11 – Heartbeat service

Parameter name	Req	Ind
Argument		
AREP	M	M
Status	M	M (=)

Argument

The argument shall convey the service specific parameters of the service request.

AREP

This parameter is the local identifier for the desired AR.

Status

This parameter specifies the heartbeat producer status.

6.1.2 ISO/IEC 8802-3 DLPDU communication ASE**6.1.2.1 Overview****6.1.2.1.1 General information**

This ASE is an optional ASE. Communication of devices with engineering tools and the possibility of integrated web servers in some devices, require TCP/IP communication in addition to Type 22 RTFL communication on the same interface. For this purposes it is possible to transfer all types of ISO/IEC 8802-3 DLPDUs over Type 22.

Type 22 utilizes a variant of communication, in which the non Type 22 DLPDUs are tunnelled and re-assembled in the associated device, before being relayed as complete ISO/IEC 8802-3 DLPDUs. This procedure does not restrict the achievable communication cycle time, since the fragment size can be optimized according to the available bandwidth. Type 22 defines a

standard channel, which in principle enables any device to participate in the normal ISO/IEC 8802-3 based traffic.

The non Type 22 DLPDU segments are transferred to the destination OD using MSC. The receiving OD reassembles the individual segments back into the standard ISO/IEC 8802-3 Format.

6.1.2.1.2 Address Resolution

Sending of tunnelled SEFs over MSC requires appropriate addressing. An OD shall be capable of matching a MAC address in the SEF with the relevant device address. Each OD which is using the MSC for SEF communication needs to setup its own table containing matches of MSC addresses and corresponding MAC addresses.

6.1.2.1.3 SEF interaction

The SEF communication service is used to send data from a client to the server. The services are unconfirmed.

Figure 19 shows a successful SEF transfer sequence between client and server and the primitives between client and server.

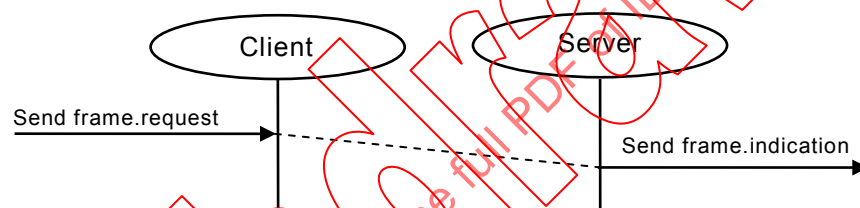


Figure 19 – SEF service sequence

6.1.2.2 SEF class specification

6.1.2.2.1 Formal model

FAL ASE:		SEF
CLASS:		SEF
CLASS ID:		not used
PARENT CLASS:		TOP
ATTRIBUTES:		
1	(m)	Key Attribute: Implicit
3	(m)	Attribute: MAC address
4	(o)	Attribute: Address mapping list
4.1	(o)	Attribute: MAC address
4.2	(o)	Attribute: Device address
SERVICES:		
1	(m)	OpsService: Send frame

6.1.2.2.2 Attributes

Implicit

The attribute indicates that the object is implicitly addressed by the service.

MAC address

This attribute specifies the MAC address of the device.

Address mapping list

The address mapping list is composed of the following elements:

MAC address

This attribute specifies the MAC address of a participating device.

Device address

This attribute specifies the device address to the corresponding MAC address of a participating device.

6.1.2.2.3 Services

Send frame

This service is used to send a ISO/IEC 8802-3 DLPDU.

6.1.2.3 SEF service specification

6.1.2.3.1 Supported services

The SEF ASE defines the services

Send frame

6.1.2.3.2 Send frame service

This service is used to send a SEF. Table 12 shows the service primitives and parameter of the service.

Table 12 – Send frame service

Parameter name	Req	Ind
Argument		
AREP	M	M
SEF frame	M	M (=)

Argument

The argument shall convey the service specific parameters of the service request.

AREP

This parameter is the local identifier for the desired AR.

SEF frame

This parameter specifies the SEF DLPDU.

6.1.3 Management ASE

6.1.3.1 Overview

The management ASE provides management services for a Type 22 network and Type 22 devices.

6.1.3.2 Management class specification

6.1.3.2.1 Formal model

FAL ASE: MGT

CLASS: MGT

CLASS ID:		not used	
PARENT CLASS:		TOP	
ATTRIBUTES:			
1	(m)	Key Attribute:	Implicit
2	(m)	Attribute:	Application layer state
SERVICES:			
1	(c)	Constraint:	Communication model RTFL device
1.1	(o)	OpsService:	AL-Network verification
1.2	(o)	OpsService:	AL-RTFL configuration
1.3	(c)	Constraint:	Device role root device and PCS in use
1.3.1	(m)	OpsService:	AL-DelayMeasurement start
1.3.2	(m)	OpsService:	AL-DelayMeasurement read
1.3.3	(m)	OpsService:	AL-PCS configuration
1.4	(o)	OpsService:	AL-MII read
1.5	(o)	OpsService:	AL-MII write
2	(c)	Constraint:	Communication model RTFN device
2.1	(o)	OpsService:	AL-RTFN scan network read
3	(m)	OpsService:	Application layer management
4	(o)	OpsService:	AL-sync start
5	(o)	OpsService:	AL-sync stop

6.1.3.2.2 Attributes

Implicit

The attribute indicates that the object is implicitly addressed by the service.

Application layer state

The attribute indicates the state of the application layer.

6.1.3.3 Management service specification

6.1.3.3.1 Supported services

The Management ASE defines the services

AL-Network verification

AL-RTFL configuration

AL-DelayMeasurement start

AL-DelayMeasurement read

AL-PCS configuration

AL-MII read

AL-MII write

AL-RTFN scan network read

Application layer management

AL-sync start

AL-sync stop

6.1.3.3.2 AL-Network verification

The network verification service is used by a RD to verify a Type 22 RTFLnetwork against a preset set of participating devices. Table 13 shows the service primitives and parameter of the service.

Table 13 – AL-Network verification service

Parameter name	Req	Cnf
Argument	M	
AREP		
Result (+)		S
Identification data list		M
Result (-)		S
NOTE The method by which a confirm primitive is correlated with its corresponding preceding request primitive is a local matter. See 1.2.		

Argument

The argument shall convey the service specific parameters of the service request.

AREP

This parameter is the local identifier for the desired AR.

Result(+)

This selection type parameter indicates that the service request succeeded.

Identification data list

This parameter specifies the result of the network verification. It shall reflect the participating devices by a list consisting of one identification data set for each device.

Result(-)

This selection type parameter indicates that the service request failed.

6.1.3.3.3 AL-RTFL configuration

The RTFL configuration service is used by a RD to configure a Type 22 RTFL device. Table 14 shows the service primitives and parameter of the service.

Table 14 – AL-RTFL configuration service

Parameter name	Req	Cnf
Argument		
AREP	M	
Predecessor MAC	M	
Successor MAC	M	
Successor MAC altern.	M	
Device address	M	
MSCShortMsgSize	M	
Number of frames	M	
Cycle time	M	
RTF timeout	M	
Master clock DA	M	
IP configuration	C	
Result (+)		S
Configuration summary		M
Result (-)		S
NOTE The method by which a confirm primitive is correlated with its corresponding preceding request primitive is a local matter. See 1.2.		

Argument

The argument shall convey the service specific parameters of the service request.

AREP

This parameter is the local identifier for the desired AR.

Predecessor MAC

This parameter indicates the MAC address of the preceding device within the logical double line.

Successor MAC

This parameter indicates the MAC address of the succeeding device within the logical double line.

Successor MAC alternative

This parameter indicates an alternative MAC address of the succeeding device within the logical double line.

Device address

This parameter indicates the device address which shall be used.

MSCShortMsgSize

This parameter indicates the maximum message size in octets for an un-segmented message transfer using MSC.

Number of frames

This parameter indicates the number of RTF DLPDUs used for both possible communication channels.

Cycle time

This parameter indicates the cycle time of the communication cycle.

RTF timeout

This parameter indicates a maximum delay time for the RTFL communication cycle time from the expected communication cycle time.

Master clock DA

This parameter indicates the device address of the device which integrates the master clock.

IP configuration

This parameter indicates IP configuration data.

Result(+)

This selection type parameter indicates that the service request succeeded.

Configuration summary

This parameter contains a summary of the performed device configuration in terms of state information for the configured parameters.

Result(-)

This selection type parameter indicates that the service request failed.

6.1.3.3.4 AL-DelayMeasurement start

With the DelayMeasurement start service as specified in Table 15 a RD in a Type 22 RTFL network starts the propagation delay measurement for PCS.

Table 15 – AL-DelayMeasurement start service

Parameter name	Req
Argument	
AREP	M
Repeat count	M

Argument

The argument shall convey the service specific parameters of the service request.

AREP

This parameter is the local identifier for the desired AR.

Repeat count

This parameter shall indicate the number of communication cycles used for propagation delay measurement.

NOTE Refer to IEC 61158-4-22:2010 for further information about DelayMeasurement sequence.

6.1.3.3.5 AL-DelayMeasurement read

With the DelayMeasurement read service as specified in Table 16 a RD in a Type 22 RTFL network shall read the propagation delay measurement results.

Table 16 – AL-DelayMeasurement read service

Parameter name	Req	Cnf
Argument	M	
AREP		
Result (+)		S
Delay		M
Result (-)		S
NOTE The method by which a confirm primitive is correlated with its corresponding preceding request primitive is a local matter. See 1.2.		

Argument

The argument shall convey the service specific parameters of the service request.

AREP

This parameter is the local identifier for the desired AR.

Result(+)

This selection type parameter indicates that the service request succeeded.

Delay

This parameter shall contain the delay measurement result.

NOTE Refer to IEC 61158-4-22:2010 for further information about DelayMeasurement sequence.

6.1.3.3.6 AL-PCS configuration

This service is used by a RD in a Type 22 RTFL network to configure the participating devices according to the delay measurement results. Table 17 shows the service primitives and parameter of the service.

Table 17 – PCS configuration service

Parameter name	Req
Argument	M
AREP	
Clock configuration	

Argument

The argument shall convey the service specific parameters of the service request.

AREP

This parameter is the local identifier for the desired AR.

Clock configuration

This parameter specifies the configuration data for clock adjustment.

6.1.3.3.7 AL-MII read

With the local service MII read as specified in Table 18 information from the MII is read.

Table 18 – MII read service

Parameter name	Req	Cnf
Argument		
Address of register	M	
Result (+)		S
Data		M
Result (-)		S

Argument

The argument shall convey the service specific parameters of the service request.

Address of register

This parameter indicates the address of the PHY register which is accessed by the read operation.

Result(+)

This selection type parameter indicates that the service request succeeded.

Data

This parameter shall contain the information of the read PHY register.

Result(-)

This selection type parameter indicates that the service request failed.

6.1.3.3.8 AL-MII write

With the local service MII write as specified in Table 19 information is written to the MII.

Table 19 – MII write service

Parameter name	Req
Argument	
Address of register	M
Data	M

Argument

The argument shall convey the service specific parameters of the service request.

Address of register

This parameter indicates the address of the PHY register which is accessed by the write operation.

Data

This parameter shall contain the information to be written to the PHY register.

6.1.3.3.9 AL-RTFN scan network read

The RTFN scan network read service as specified in Table 20 allows to explore a RTFN network. All participating devices are identified by descriptive identification data.

Table 20 – AL-RTFN scan network read service

Parameter name	Req	Cnf
Argument		
AREP	M	
Result (+)		S
Identification data list		M
Result (-)		S
NOTE The method by which a confirm primitive is correlated with its corresponding preceding request primitive is a local matter. See 1.2.		

Argument

The argument shall convey the service specific parameters of the service request.

AREP

This parameter is the local identifier for the desired AR.

Result(+)

This selection type parameter indicates that the service request succeeded.

Identification data list

This parameter shall contain the result of the RTFN topology exploration. It shall reflect the participating RTFN devices by a list consisting of one identification data set for each device.

Result(-)

This selection type parameter indicates that the service request failed.

6.1.3.3.10 Application layer management

With the application layer management service as specified in Table 21 the application layer state machine as specified in IEC 61158-6-22:2010 can be triggered to control the application behavior.

Table 21 – Application layer management service

Parameter name	Req	Ind
Argument		
AREP	M	M
Address	M	M (=)
Command	M	M (=)

Argument

The argument shall convey the service specific parameters of the service request.

AREP

This parameter is the local identifier for the desired AR.

Address

This parameter shall indicate the address of the destination device for the application layer management command.

Command

This parameter shall indicate the application layer management command.

6.1.3.3.11 AL-sync start

With the AL-sync start service as specified in Table 22 a Type 22 device shall request configuration information from the corresponding synchronization master and start the indication of sync interrupt signals to the AL-user.

Table 22 – Start synchronization service

Parameter name	Req	Cnf
Argument		
AREP	M	
Sync ID	M	
Result (+)		S
Sync ID		M
Start time		M
Cycle time		M
Result (-)		S
NOTE The method by which a confirm primitive is correlated with its corresponding preceding request primitive is a local matter. See 1.2.		

Argument

The argument shall convey the service specific parameters of the service request.

AREP

This parameter is the local identifier for the desired AR.

Sync ID

This parameter specifies the network wide unique ID for the requested sync interrupt.

Result(+)

This selection type parameter indicates that the service request succeeded.

Sync ID

This parameter specifies the network wide unique ID for the requested sync interrupt.

Start time

This parameter specifies the absolute start time of the master clock.

Cycle time

This parameter specifies the cycle time of sync interrupt indication.

Result(-)

This selection type parameter indicates that the service request failed.

6.1.3.3.12 AL-sync stop

The local AL-sync stop service is used to stop the indication of sync interrupt signals to the AL-user. Table 23 shows the service primitives and parameter of the service.

Table 23 – Stop synchronization service

Parameter name	Req
Argument	
Sync ID	M

Argument

The argument shall convey the service specific parameters of the service request.

Sync ID

This parameter specifies the network wide unique ID for the requested sync interrupt.

6.2 Application relationships (ARs)

6.2.1 Overview

Type 22 provides four pre-defined and pre-established application relationships, these are:

- point-to-point network-scheduled unconfirmed producer-consumer (PTPNSU) AREP;
- point-to-multipoint network-scheduled unconfirmed producer-consumer (PTMNSU) AREP;
- point-to-point network-scheduled confirmed client/server (PTPNSC) AREP;
- point-to-point user-triggered confirmed client/server (PTPUTC) AREP.

6.2.2 Point-to-point network-scheduled unconfirmed producer-consumer AREP

This class is defined to support the model for scheduled queued distribution of unconfirmed services between two devices acting as a producer and a consumer application process.

The behavior of this type of AR can be described as follows.

An AR ASE user wishing to convey a request APDU submits it as an AR ASE service data unit to its AREP for distribution. The sending AREP writes the APDU into the internal queue. The AR ASE transfers the queue contents at the next scheduled transfer opportunities.

If the AREP receives another APDU before the queue contents are transmitted, the queue will be extended by the new APDU.

At the receiving endpoint, the APDU is received from the network and is written immediately into the receive queue. The endpoint notifies the user that the APDU has arrived and delivers it to the user according to the local user interface. If the APDU has not been delivered before the next APDU arrives, the queue is extended by the recently received APDU and a further notification is issued.

6.2.3 Point-to-multipoint network-scheduled unconfirmed producer-consumer AREP

This class is defined to support the model for scheduled and queued distribution of unconfirmed services from one producer application process to one or more consumer application processes. The services provided by this AREP may be used for a direct communication between devices.

The behavior of this type of AR can be described as follows.

An AR ASE user wishing to convey a request APDU submits it as an AR ASE service data unit to its AREP for distribution. The sending AREP writes the APDU into the internal queue, extending the existing contents of the queue.

The AREP transfers the buffer contents at the next scheduled transfer opportunity.

If the AREP receives another APDU before the queue contents are transmitted, the queue will be extended by the new APDU.

At the receiving endpoint, the APDU is received from the network and is written immediately into the receive queue. The endpoint notifies the user that the APDU has arrived and delivers it to the user according to the local user interface. If the APDU has not been delivered before the next APDU arrives, the queue is extended by the recently received APDU and a further notification is issued.

6.2.4 Point-to-point network-scheduled confirmed client/server AREP

This class is defined to support the on-demand exchange of confirmed services between two devices acting as client and server application process. Unconfirmed services are not supported by this type of AR. The behavior of this class is described as follows.

A CeS ASE user wishing to convey a request APDU submits it as an ASE service data unit to its AREP and the AREP sending the request APDU queues it to its underlying layer for transfer at the next available opportunity. The AREP receiving the request APDU from its underlying layer queues it for delivery to its CeS ASE user in the order in which it was received.

The AREP receiving the request APDU accepts the corresponding response APDU from its CeS ASE user and queues it to the underlying layer for transfer.

The AREP that issued the request APDU receives the response APDU from its underlying layer and queues it for delivery to its CeS ASE user in the order in which it was received.

6.2.5 Point-to-point user-triggered confirmed client/server AREP

This class is defined to support the on-demand exchange of confirmed services between two devices acting as client and server application process. Unconfirmed services are not supported by this type of AR. The behavior of this class is described as follows.

A CeS ASE user wishing to convey a request APDU submits it as an ASE service data unit to its AREP and the AREP sending the request APDU directly sends it to its underlying layer for transfer. The AREP receiving the request APDU from its underlying layer queues it for delivery to its CeS ASE user in the order in which it was received.

The AREP receiving the request APDU accepts the corresponding response APDU from its CeS ASE user and sends it to the underlying layer for transfer.

The AREP that issued the request APDU receives the response APDU from its underlying layer and queues it for delivery to its CeS ASE user in the order in which it was received.

6.2.6 AR classes

Table 24 defines the characteristics of point-to-point network-scheduled unconfirmed producer-consumer (PTPNSU) AREP class.

Table 24 – PTPNSU AREP class

AREP characteristics	Class
Roles	Producer Consumer
Cardinality	One-to-one
Conveyance paths	Unidirectional
Trigger policy	Network-scheduled
Conveyance policy	Queued

Table 25 defines the characteristics of point-to-multipoint network-scheduled unconfirmed producer-consumer (PTMNSU) AREP class.

Table 25 – PTMNSU AREP class

AREP characteristics	Class
Roles	Producer Consumer
Cardinality	One-to-many
Conveyance paths	Unidirectional
Trigger policy	Network-scheduled
Conveyance policy	Queued

Table 26 defines the characteristics of point-to-point network-scheduled confirmed client/server (PTPNSC) AREP class.

Table 26 – PTPNSC AREP class

AREP characteristics	Class
Roles	Client Server
Cardinality	One-to-one
Conveyance paths	Bi-directional
Trigger policy	Network-scheduled
Conveyance policy	Queued

Table 27 defines the characteristics of point-to-point user-triggered confirmed client/server (PTPUTC) AREP class.

Table 27 – PTPUTC AREP class

AREP characteristics	Class
Roles	Client Server
Cardinality	One-to-one
Conveyance paths	Bi-directional
Trigger policy	User-triggered
Conveyance policy	Queued

6.2.7 FAL services by AREP class

Table 28 defines the list of FAL services which can be conveyed by each corresponding AREP class and its defined AREP role.

Table 28 – FAL services by AREP class

FAL service	Used AREP class			
	PTPNSU	PTMNSU	PTPNSC	PTPUTC
CeS ASE	—	—	—	—
Initiate SDO expedited download	—	—	X	X
Initiate SDO normal download	—	—	X	X
SDO download	—	—	X	X
Initiate SDO expedited upload	—	—	X	X
Initiate SDO normal upload	—	—	X	X
SDO upload	—	—	X	X
SDO abort	—	—	X	X
Process data write	X	X	—	—
Emergency	X	X	—	—
Heartbeat	X	X	—	—
SEF ASE	—	—	—	—
Send frame	X	X	—	—
Management ASE	—	—	—	—
AL-Network verification	—	—	X	—
AL-RTFL configuration	—	—	X	—
AL-DelayMeasurement start	—	X	—	—
AL-DelayMeasurement read	—	—	X	—
AL-PCS configuration	—	—	X	—
AL-RTFN scan network read	—	—	—	X
Application Layer Management	—	—	X	X
AL-sync start	—	—	X	X

6.2.8 Permitted FAL services by AREP role

Table 29 defines the list of permitted FAL services by AREP roles.

Table 29 – FAL services by AREP role

FAL service	AREP role							
	Client		Server		Producer		Consumer	
	req	cnf	ind	rsp	req	ind	req	ind
CeS ASE	—	—	—	—	—	—	—	—
Initiate SDO expedited download	X	X	X	X	—	—	—	—
Initiate SDO normal download	X	X	X	X	—	—	—	—
SDO download	X	X	X	X	—	—	—	—
Initiate SDO expedited upload	X	X	X	X	—	—	—	—
Initiate SDO normal upload	X	X	X	X	—	—	—	—
SDO upload	X	X	X	X	—	—	—	—
SDO abort	X	X	X	X	—	—	—	—
Process data write	—	—	—	—	X	—	—	X
Emergency	—	—	—	—	X	—	—	X
Heartbeat	—	—	—	—	X	—	—	X
SEF ASE	—	—	—	—	—	—	—	—
Send frame	—	—	—	—	X	—	—	X
Management ASE	—	—	—	—	—	—	—	—
AL-Network verification	X	X	X	X	—	—	—	—
AL-RTFL configuration	X	X	X	X	—	—	—	—
AL-DelayMeasurement start	—	—	—	—	X	—	—	X
AL-DelayMeasurement stop	X	X	X	X	—	—	—	—
AL-PCS configuration	X	X	X	X	—	—	—	—
AL-RTFN scan network read	X	X	X	X	—	—	—	—
Application Layer Management	X	—	X	—	—	—	—	—
AL-sync start	X	X	X	X	—	—	—	—

Bibliography

EN 50325-4, *Industrial communication subsystem based on ISO 11898 (CAN) for controller device interfaces – Part 4 CANopen*

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

Withdrawing
IECNORM.COM : Click to view the full PDF of IEC 61158-5-22:2010

SOMMAIRE

AVANT-PROPOS.....	82
INTRODUCTION.....	84
1 Domaine d'application	85
1.1 Vue d'ensemble.....	85
1.2 Spécifications.....	86
1.3 Conformité	86
2 Références normatives.....	86
3 Termes, définitions, abréviations, symboles et conventions	87
3.1 Termes de l'ISO/CEI 7498-1	87
3.2 Termes de l'ISO/CEI 8822	88
3.3 Termes de l'ISO/CEI 9545	88
3.4 Termes de l'ISO/CEI 8824-1	88
3.5 Définitions spécifiques à la couche application des bus de terrain de Type 22.....	88
3.6 Symboles et abréviations	92
3.7 Conventions	95
4 Concepts.....	98
4.1 Concepts communs.....	98
4.2 Concepts spécifiques de type.....	98
5 ASE Data type (type de données).....	103
5.1 Vue d'ensemble.....	103
5.2 Définition formelle des objets de types de données	103
5.3 Types de données définis pour la FAL.....	103
6 Spécification de modèle de communication	111
6.1 Éléments de service application (les ASE).....	111
6.2 Relations d'applications (les AR).....	160
Bibliographie.....	166
Figure 1 – Modèle d'interaction producteur-consommateur	100
Figure 2 – Modèle de référence des dispositifs RTFL.....	101
Figure 3 – Modèle de référence des dispositifs RTFN	102
Figure 4 – Structure de dispositif CeS de Type 22	112
Figure 5 – Séquence de téléchargement aval accéléré de SDO qui a réussi	127
Figure 6 – Séquence d'initialisation de téléchargement aval normal de SDO qui a réussi	128
Figure 7 – Séquence de téléchargement aval de SDO qui a réussi	128
Figure 8 – Séquence de téléchargement amont accéléré de SDO qui a réussi	129
Figure 9 – Séquence d'initialisation de téléchargement amont normal de SDO qui a réussi	130
Figure 10 – Séquence de téléchargement amont de SDO qui a réussi	130
Figure 11 – Séquence d'initialisation de téléchargement aval accéléré de SDO qui a échoué.....	131
Figure 12 – Téléchargement aval de SDO qui a échoué après la séquence d'initialisation.....	132
Figure 13 – Séquence de téléchargement aval de SDO qui a échoué	132

Figure 14 – Séquence d'urgence.....	133
Figure 15 – Séquence de heartbeat	134
Figure 16 – Séquence d'écriture de données de processus.....	135
Figure 17 – Principe de la mise en correspondance de PDO	136
Figure 18 – Objet de données de processus	137
Figure 19 – Séquence du service SEF	150

Tableau 1 – Structure de dictionnaire d'objets.....	113
Tableau 2 – Service "Initiate SDO expedited download"	140
Tableau 3 – Service "Initiate SDO normal download"	141
Tableau 4 – Service "SDO download"	142
Tableau 5 – Service "Initiate SDO expedited upload"	143
Tableau 6 – Service "Initiate SDO normal upload"	145
Tableau 7 – Service "SDO upload".....	146
Tableau 8 – Service "SDO abort".....	147
Tableau 9 – Service "Process data write".....	148
Tableau 10 – Service "Emergency" (EMCY).....	148
Tableau 11 – Service "Heartbeat"	149
Tableau 12 – Service "Send frame".....	152
Tableau 13 – Service "AL-Network verification"	153
Tableau 14 – Service "AL-RTFL configuration"	154
Tableau 15 – Service "AL-DelayMeasurement start"	155
Tableau 16 – Service "AL-DelayMeasurement read"	156
Tableau 17 – Service "PCS configuration"	156
Tableau 18 – Service "MII read".....	157
Tableau 19 – Service "MII write"	157
Tableau 20 – Service "AL-RTFN scan network read".....	158
Tableau 21 – Service "Application layer management"	159
Tableau 22 – Service "Start synchronization"	159
Tableau 23 – Service "Stop synchronization"	160
Tableau 24 – Classe d'AREP PTPNSU	162
Tableau 25 – Classe d'AREP PTMNSU	163
Tableau 26 – Classe d'AREP PTPNSC	163
Tableau 27 – Classe d'AREP PTPUTC	163
Tableau 28 – Services de FAL par classe d'AREP	164
Tableau 29 – Services de FAL par rôle d'AREP	165

COMMISSION ÉLECTROTECHNIQUE INTERNATIONALE

RÉSEAUX DE COMMUNICATION INDUSTRIELS – SPÉCIFICATIONS DES BUS DE TERRAIN –

Partie 5-22: Définition des services de la couche application – Éléments de Type 22

AVANT-PROPOS

- 1) La CEI (Commission Électrotechnique Internationale) 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 des questions techniques, représentent, dans la mesure du possible, un accord international sur les sujets étudiés, étant donné que les Comités nationaux 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. Toute divergence entre toute Publication de la CEI et toute publication nationale ou régionale correspondante doit être indiquée en termes clairs dans cette dernière.
- 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 propriété intellectuelle ou de droits analogues. La CEI ne saurait être tenue pour responsable de ne pas avoir identifié de tels droits de propriété et de ne pas avoir signalé leur existence.

NOTE 1 L'utilisation de certains des types de protocoles associés est limitée par les détenteurs de leurs droits de propriété intellectuelle. Dans tous les cas, l'engagement à un abandon limité des droits de propriété intellectuelle pris par les détenteurs de ces droits permet d'utiliser un type particulier de protocole de couche liaison de données avec des protocoles de couche physique et de couche application dans des combinaisons de types telles que spécifiées de façon explicite dans les parties profil. L'utilisation des divers types de protocoles dans d'autres combinaisons peut exiger la permission donnée par les détenteurs respectifs de leurs droits de propriété intellectuelle.

La Norme internationale CEI 61158-5-22 a été établie par le sous-comité 65C: Réseaux de communication industriels du comité d'études 65 de la CEI: Mesure, commande et automation dans les processus industriels.

La présente norme annule et remplace la IEC/PAS 61158-5-22 publiée en 2009. Cette première édition constitue une révision technique.

La présente version bilingue publiée en 2012-01 correspond à la version anglaise monolingue publiée en 2010-08.

Le texte anglais de cette norme est issu des documents 65C/606/FDIS et 65C/620/RVD.

Le rapport de vote 65C/620/RVD donne toute information sur le vote ayant abouti à l'approbation de cette norme.

La version française de cette norme n'a pas été soumise au vote.

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

Une liste de toutes les parties de la série CEI 61158, présentées sous le titre général *Réseaux de communication industriels – Spécifications de 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. A cette date, la publication sera

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

NOTE 2 La révision de la présente norme sera synchronisée avec les autres parties de la série CEI 61158.

INTRODUCTION

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

Le service application est fourni par le protocole d'application utilisant les services disponibles de la liaison de données ou autre couche immédiatement inférieure. La présente norme définit les caractéristiques de services d'application pouvant être exploitées par les applications de bus de terrain et/ou la gestion de système.

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 application 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 5-22: Définition des services de la couche application – Eléments de Type 22

1 Domaine d'application

1.1 Vue d'ensemble

La Couche application de bus de terrain (FAL¹) fournit aux programmes d'utilisateur un moyen d'accéder à l'environnement de communication du bus de terrain. À cet égard, la FAL peut être vue comme une «fenêtre entre des programmes d'application correspondants».

La présente norme fournit les éléments communs pour les communications de messagerie de base à temps critique et à temps non critique entre des programmes d'application dans un environnement d'automation et le matériel spécifique au bus de terrain de Type 22. 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 application de bus de terrain en termes

- a) d'un modèle abstrait pour définir des ressources (objets) d'application capables d'être manipulées par les utilisateurs par l'intermédiaire de l'utilisation du service FAL;
- b) des actions et événements primitifs du service;
- c) des paramètres associés à chaque action primitive et événement primitif, et la forme qu'ils prennent; et
- d) 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

- a) l'utilisateur de FAL à la frontière entre l'utilisateur et la couche application du modèle de référence de bus de terrain; et
- b) la gestion des systèmes au niveau de la frontière entre la couche application et la gestion des systèmes selon le Modèle de référence de bus de terrain.

La présente norme spécifie la structure et les services de la couche application des bus de terrain, en conformité avec le Modèle de référence de base de l'OSI (ISO/CEI 7498) et la structure de la couche application de l'OSI (ISO/CEI 9545).

Les services et protocoles de la FAL sont fournis par des entités d'application (AE²) de la FAL contenues dans les processus d'application. L'AE de la FAL se compose d'un jeu d'éléments de service application (ASE³) orientés objet et d'une entité de gestion de couche (LME⁴) qui

1 FAL = *Fieldbus Application Layer*

2 AE = *Application Entity*

3 ASE = *Application Service Element*

4 LME = *Layer Management Entity*

gère l'AE. Les ASE fournissent des services de communication qui fonctionnent sur un jeu de classes d'objets de processus application (APO⁵) connexes. L'un des ASE de la FAL est un ASE de gestion qui fournit un jeu commun de services pour la gestion des instances de classes de la FAL.

Bien que ces services spécifient, du point de vue des applications, la manière dont la demande et les réponses sont émises et délivrées, ils n'incluent pas une spécification de ce que les applications qui demandent et qui répondent doivent en faire. À savoir, les aspects comportementaux des applications ne sont pas spécifiés; seule une définition des demandes et réponses qu'elles peuvent envoyer/recevoir est spécifiée. Cela permet une plus grande flexibilité aux utilisateurs de la FAL pour normaliser un tel comportement d'objet. En plus de ces services, certains services d'appui sont également définis dans la présente norme pour fournir l'accès à la FAL afin de maîtriser certains aspects de son fonctionnement.

1.2 Spécifications

L'objectif principal de la présente norme est de spécifier les caractéristiques des services conceptuels d'une couche application qui sont adaptées à des communications à temps critique et, donc, complètent le Modèle de référence de base de l'OSI en guidant le développement des protocoles de couche application pour les communications à temps critique.

Un objectif secondaire est de fournir des trajets de migration à partir de protocoles industriels de communication préexistants. C'est ce dernier objectif qui donne naissance à la diversité des services normalisés comme les divers Types de la CEI 61158, et les protocoles correspondants normalisés dans les sous-parties de la CEI 61158-6.

La présente spécification peut être utilisée comme la base pour les interfaces de programmation d'applications (Application Programming-Interfaces) formelles. 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" (demande et confirmation) ou "indication-response" (indication et réponse).

1.3 Conformité

La présente norme ne spécifie aucune mise en œuvre ou aucun produit individuels, de même qu'elle ne restreint nullement les mises en œuvre des entités de couche application dans les systèmes d'automatisme industriels.

Il n'y a pas de conformité d'équipement à la présente norme de définition des services de couche application. Au contraire, la conformité est obtenue par une mise en œuvre de protocoles conformes de couche application qui satisfont aux services de couche application définis dans la présente norme.

2 Références normatives

Les documents de référence suivants sont indispensables pour l'application du présent document. 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).

⁵ APO = Application Process Object

CEI 60559, *Arithmétique binaire en virgule flottante pour systèmes à microprocesseurs*

IEC 61131-3, *Programmable controllers – Part 3: Programming languages* (disponible uniquement en anglais)

IEC/TR 61158-1:2010⁷, *Industrial communication networks – Fieldbus specifications – Part 1: Overview and guidance for the IEC 61158 and IEC 61784 series* (disponible uniquement en anglais)⁶

IEC 61158-4-22:2010⁷, *Industrial communication networks – Fieldbus specifications – Part 4-22: Data-link layer protocol specification – Type 22 elements* (disponible uniquement en anglais)

IEC 61158-6-22:2010⁷, *Industrial communication networks – Fieldbus specifications – Part 6-22: Application layer protocol specification – Type 22 elements* (disponible uniquement en anglais)

ISO/IEC 646, *Technologies de l'information – Jeu ISO de caractères codés à 7 éléments pour l'échange d'informations*

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

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 uniquement en anglais)

ISO/CEI 8822, *Technologies de l'information – Interconnexion de systèmes ouverts (OSI) – Définition du service de présentation*

ISO/CEI 8824-1, *Technologies de l'information – Notation de syntaxe abstraite numéro un (ASN.1): Spécification des contraintes*

ISO/CEI 9545, *Technologies de l'information – Interconnexion de systèmes ouverts (OSI) – Structure de la couche application*

ISO/CEI 10646, *Technologie de l'information – Jeu universel de caractères codés sur plusieurs octets (Unicode)*

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

3 Termes, définitions, abréviations, symboles et conventions

Pour les besoins du présent document, les termes suivants tels que définis dans ces publications s'appliquent:

3.1 Termes de l'ISO/CEI 7498-1

- a) entité d'application
- b) processus d'application

⁶ Les publications monolingues des séries IEC 61158 et IEC 61784 sont actuellement en cours de traduction.

⁷ À paraître.

- c) unité de données de protocole application
- d) élément de service application
- e) invocation d'entité d'application
- f) invocation de processus application
- g) transaction d'application
- h) système ouvert réel
- i) syntaxe de transfert

3.2 Termes de l'ISO/CEI 8822

- a) syntaxe abstraite
- b) contexte de présentation

3.3 Termes de l'ISO/CEI 9545

- a) association d'applications (*application-association*)
- b) contexte d'application (*application-context*)
- c) nom de contexte d'application (*application context name*)
- d) invocation d'entité d'application (*application-entity-invocation*)
- e) type d'entité d'application (*application-entity-type*)
- f) invocation de processus d'application (*application-process-invocation*)
- g) type de processus d'application (*application-process-type*)
- h) élément de service application (*application-service-element*)
- i) élément de service de contrôle d'application (*application control service element*)

3.4 Termes de l'ISO/CEI 8824-1

- a) Identificateur d'objet (*object identifier*)
- b) type

3.5 Définitions spécifiques à la couche application des bus de terrain de Type 22

3.5.1

application

fonction pour laquelle des données sont échangées

3.5.2

objet d'application

représentation d'un composant particulier dans un dispositif

3.5.3

donnée acyclique

donnée qui est transférée de temps à autre à des fins spécialisées

3.5.4

bit

unité d'informations constituée d'un 1 ou d'un 0. Il s'agit de la plus petite unité de donnée pouvant être émise

3.5.5

cellule

synonyme d'un segment de DL unique qui utilise le modèle de communication RTFL

3.5.6**voie**

chemin fourni pour acheminer des données

3.5.7**client**

objet qui utilise les services d'un serveur en déclenchant un message pour accomplir une tâche

3.5.8**cycle de communication**

période de temps fixe pendant laquelle le dispositif-racine émet des DLPDU vides pour le déclenchement d'une communication cyclique dans laquelle des données sont émises en utilisant la CDC et la MSC

3.5.9**connexion**

liaison logique entre deux objets d'application

3.5.10**temps de cycle**

durée d'un cycle de communication

3.5.11**cyclique**

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

3.5.12**communication cyclique**

échange périodique de télégrammes

3.5.13**donnée cyclique**

donnée qui est transférée de manière régulière et répétitive à des fins spécialisées

3.5.14**voie de données cycliques****CDC**

partie intégrante d'une ou plusieurs DLPDU, qui est réservée aux données cycliques

3.5.15**donnée**

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

3.5.16**dispositif**

entité physique connectée au bus de terrain

3.5.17**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)

3.5.18**code d'erreur**

numéro d'identification d'un type spécifique d'erreur

3.5.19

passerelle

dispositif agissant comme élément de liaison entre différents protocoles

3.5.20

indice

position d'un objet au sein du dictionnaire d'objets

3.5.21

communication intercellulaire

communication entre un dispositif RTFL et un dispositif RTFN ou bien communication entre un dispositif RTFL et un autre dispositif RTFL dans des cellules différentes reliées par un RTFN

3.5.22

interface

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

3.5.23

communication intracellulaire

communication entre un dispositif RTFL et un autre dispositif RTFL dans la même cellule

3.5.24

double ligne logique

séquence de dispositif-racine et de tous les dispositifs ordinaires traitant la DLPDU de communication dans le sens direct et inverse

3.5.25

paramètres de mise en correspondance

ensemble de valeurs définissant la correspondance entre des objets d'application et des objets de données de processus

3.5.26

horloge mère

base de temps globale pour le mécanisme PCS

3.5.27

message

séquence ordonnée d'octets censés acheminer des données

3.5.28

voie de messages

MSC

partie intégrante d'une ou plusieurs DLPDU, qui est réservée aux données acycliques

3.5.29

réseau

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

3.5.30

dispositif ordinaire

OD

esclave dans le système de communication qui utilise la RTFL pour l'échange de données cycliques et acycliques avec d'autres OD dans la même double ligne logique

3.5.31**synchronisation d'horloge précise****PCS**

mécanisme pour synchroniser des horloges de dispositifs RTFL et maintenir une base de temps globale

3.5.32**donnée de processus**

donnée désignée pour être transférée de façon cyclique ou acyclique à des fins de traitement

3.5.33**objet de données de processus**

objet de données spécialisé destiné à être transféré de façon cyclique ou acyclique à des fins de traitement

3.5.34**protocole**

convention relative aux formats de données, aux séquences temporelles et à la correction d'erreur dans l'échange de données des systèmes de communications

3.5.35**dispositif-racine****RD**

maître dans le système de communication, qui organise, lance et régule l'échange de données cycliques et acycliques RTFL pour une même double ligne logique

3.5.36**ligne de trame temps réel****RTFL**

modèle de communication communiquant en une double ligne logique

3.5.37**réseau de trame temps réel****RTFN**

modèle de communication communiquant en un réseau commuté

3.5.38**temps aller-retour**

temps de transmission dont une DLPDU a besoin pour aller du RD jusqu'au dernier OD dans les sens direct et inverse

3.5.39**sous-indice**

sous-position d'un élément individuel d'un objet au sein du dictionnaire d'objets

3.5.40**signal de synchronisation**

indication temporelle de la survenue d'un événement, habituellement sous la forme d'un signal d'interruption, utilisée pour la synchronisation d'utilisateur de DL

3.5.41**topologie**

architecture de réseau physique en rapport avec la connexion entre les stations du système de communication

3.6 Symboles et abréviations

AE	Application entity (Entité d'application)
AL	Application Layer (Couche Application)
ALME	Application Layer Management Entity (Entité de gestion de couche application)
AP	Application Process (Processus application)
APDU	Application Layer Protocol Data Unit (Unité de données de protocole d'application)
APO	Application process object (Objet de processus d'application)
AR	Application Relationship (Relation d'applications)
AREP	Application Relationship End Point (Point d'extrémité de relation d'applications)
ASE	Application Service Element (Élément de service application)
CAN	Controller Area Network (Gestionnaire de réseau de communication)
CDC	Cyclic Data Channel (Voie de données cycliques)
CDCL	CDC line (Ligne CDC)
CDCN	CDC network (Réseau CDC)
CeS	CANopen expands Type 22 (CANopen étend le Type 22)
CL	Communication Layer (Couche communication)
Cnf	Confirmation
DA	Device Address (Adresse de dispositif)
DHCP	Dynamic Host Configuration Protocol (Protocole de configuration dynamique d'hôte)
DL-	Data-link layer (couche liaison de données) (comme préfixe)
DLL	DL-layer (Couche DL)
DLPPDU	DL-Protocol-Data-Unit (Unité de données de protocole de DL)
EDS	Electronic data sheet (Feuille de données électronique)
EMCY	Emergency (Urgence)

FAL	Fieldbus Application Layer (Couche application de bus de terrain)
GW	Gateway (Passerelle)
ID	Identification
IETF	Internet Engineering Task Force (Groupe de travail d'ingénierie Internet)
Ind	Indication
IP	Internet Protocol (Protocole Internet)
IPv4	IP version 4
IPv6	IP version 6
IRQ	Interrupt Request (Demande d'interruption)
LME	Layer management entity (Entité de gestion de couche)
MAC	Medium Access Control (Commande d'accès au support)
MC	Master Clock (Horloge-mère)
MII	Media Independent Interface (Interface indépendante du support)
MSC	Message Channel (Voie de messages)
MSCL	MSC line (Ligne MSC)
MSCN	MSC network (Réseau MSC)
OD	Ordinary device (Dispositif ordinaire)
OS	Operating system (Système d'exploitation)
OSI	Open Systems Interconnection (Interconnexion des systèmes ouverts)
PCS	Precise clock synchronization (Synchronisation d'horloge précise)
PDO	Process Data Object (Objet de données de processus)
PHY	Physical interface controller (Contrôleur d'interface physique)
PID	Packet ID (Identificateur de paquet)
PTMNSU	Point-to-multipoint network-scheduled unconfirmed (Service non confirmé programmé par le réseau en mode point à multipoint)
PTPNSC	Point-to-point network-scheduled confirmed (Service confirmé programmé par le réseau en mode point à point)

PTPNSU	Point-to-point network-scheduled unconfirmed (Service non confirmé programmé par le réseau en mode point à point)
PTPUTC	Point-to-point user-triggered confirmed (Service confirmé déclenché par l'utilisateur en mode point à point)
RD	Root device (Dispositif-racine)
Req	Request (Demande)
RFC	Request for Comments (Demande de commentaires)
Rsp	Response (Réponse)
RTF	Real time frame (Trame temps réel)
RTFL	Real time frame line (Ligne de trame temps réel)
RTFN	Real time frame network (Réseau de trame temps réel)
RO	Read only (Lecture seule)
RW	Read and write access (Accès en lecture et en écriture)
Rx	Sens de réception
RxPDO	Receive PDO (PDO de réception)
SDO	Service Data Object (Objet de données de service)
SEF	DLPDO de la norme ISO/CEI 8802-3
StdErr	Standard error output (Sortie d'erreur normalisée)
StdIn	Standard input (Entrée normalisée)
StdOut	Standard output (Sortie normalisée)
SYNC	Synchronisation
TCP	Transmission Control Protocol (Protocole de commande de transport)
TT	Transmission Type (Type d'émission)
Tx	Transmit direction (Sens d'émission)
TxPDO	Transmit PDO (PDO d'émission)
UDP	User Datagram Protocol (Protocole datagramme d'utilisateur)
WO	Write only (Écriture seule)

3.7 Conventions

3.7.1 Vue d'ensemble

La couche FAL est définie comme un jeu d'éléments ASE orientés objet. Chaque ASE est spécifié dans un paragraphe distinct. Chaque spécification d'ASE est constituée de deux parties, à savoir sa spécification de classe et sa spécification de service.

La spécification de classe définit les attributs de la classe. Les attributs sont accessibles à partir d'instances de la classe en utilisant les services d'ASE de gestion d'objets spécifiés à l'Article 5 de la présente norme. La spécification de services définit les services qui sont fournis par l'ASE.

3.7.2 Conventions générales

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

3.7.3 Conventions pour les définitions de classes

Les définitions de classes sont décrites à l'aide de modèles. Chaque modèle est constitué d'une liste d'attributs de la classe. La forme générale du modèle est montrée ci-dessous:

FAL ASE:			Nom de l'ASE
CLASS:			Nom de la classe
CLASS ID:			#
PARENT CLASS:			Nom de la classe parent
ATTRIBUTS:			
1	(o)	Attribut-clé:	Identificateur numérique
2	(o)	Attribut-clé:	nom
3	(m)	Attribut:	nom d'attribut(valeurs)
4	(m)	Attribut:	nom d'attribut(valeurs)
4.1	(s)	Attribut:	nom d'attribut(valeurs)
4.2	(s)	Attribut:	nom d'attribut(valeurs)
4.3	(s)	Attribut:	nom d'attribut(valeurs)
5.	(c)	Contrainte:	expression de la contrainte
5.1	(m)	Attribut:	nom d'attribut(valeurs)
5.2	(o)	Attribut:	nom d'attribut(valeurs)
6	(m)	Attribut:	nom d'attribut(valeurs)
6.1	(s)	Attribut:	nom d'attribut(valeurs)
6.2	(s)	Attribut:	nom d'attribut(valeurs)
SERVICES:			
1	(o)	OpsService:	nom de service
2	(c)	Contrainte:	expression de la contrainte
2.1	(o)	OpsService:	nom de service
3	(m)	MgtService:	nom de service

- (1) La rubrique "FAL ASE:" est le nom de l'élément ASE de la couche FAL (FAL ASE) qui fournit les services pour la classe spécifiée.
- (2) La rubrique "CLASS:" est le nom de la classe spécifiée. Tous les objets définis à l'aide de ce modèle seront une instance de cette classe. La classe peut être spécifiée par la

présente norme ou par un utilisateur de la présente norme.

- (3) La rubrique "CLASS ID:" est un numéro qui identifie la classe spécifiée. Ce numéro est unique au sein du FAL ASE qui fournira les services pour cette classe. Lorsqu'il est qualifié par l'identité de son FAL ASE, il identifie sans ambiguïté la classe relevant du domaine d'application de la FAL. La valeur "NULL" indique que la classe ne peut pas être instanciée. Les Class ID (identificateurs de classe) entre 1 et 255 sont réservés par la présente norme pour identifier des classes normalisées. Ils ont été attribués pour maintenir la compatibilité avec des normes nationales existantes. Les CLASS ID entre 256 et 2 048 sont alloués pour identifier les classes définies par l'utilisateur.
- (4) La rubrique "PARENT CLASS:" est le nom de la classe parent pour la classe spécifiée. Tous les attributs définis pour la classe parent et hérités par celle-ci sont hérités pour la classe définie, et ils n'ont donc pas à être redéfinis dans le modèle pour cette classe.

NOTE La classe parent "TOP" indique que la classe définie est une définition de classe initiale. La classe parent "TOP" est utilisée comme point de départ à partir duquel toutes les autres classes sont définies. L'usage de "TOP" est réservé pour les classes définies par la présente norme.

- (5) L'étiquette "ATTRIBUTES" (ATTRIBUTS) indique que les entrées suivantes sont des attributs définis pour la classe.
 - a) Chacune des entrées d'attribut contient un numéro de ligne dans la colonne 1, un indicateur obligatoire (m) / facultatif (o) / conditionnel (c) / sélecteur (s) dans la colonne 2, une étiquette de type d'attribut dans la colonne 3, un nom ou une expression conditionnelle dans la colonne 4, et, facultativement, une liste de valeurs énumérées dans la colonne 5. Dans la colonne suivant la liste de valeurs, la valeur par défaut pour l'attribut peut être spécifiée.
 - b) Les objets sont normalement identifiés par un identificateur numérique et/ou par un nom d'objet. Dans les modèles de classe, ces attributs clés sont définis sous l'attribut clé.
 - c) Le numéro de ligne définit la séquence et le niveau d'imbrication de la ligne. Chaque niveau d'imbrication est identifié par un point. L'imbrication est utilisée pour spécifier
 - i) des champs d'un attribut structuré (4.1, 4.2, 4.3),
 - ii) des attributs conditionnés à un énoncé de contrainte (5). Les attributs peuvent être obligatoires (5.1) ou facultatifs (5.2) si la contrainte est vraie. Tous les attributs facultatifs n'exigent pas des énoncés de contraintes comme le fait l'attribut défini en (5.2).
 - iii) les champs sélection d'un attribut de type choix (6.1 et 6.2).
- (6) L'étiquette "SERVICES" indique que les entrées suivantes sont des services définis pour la classe.
 - a) Un (m) dans la colonne 2 indique que le service est obligatoire pour la classe, alors qu'un (o) indique qu'il est facultatif. Un (c) dans cette colonne indique que le service est conditionnel. Lorsque tous les services définis pour une classe le sont comme étant facultatifs, l'un au moins doit être sélectionné quand une instance de la classe est définie.
 - b) L'étiquette "OpsService" désigne un service opérationnel (1).
 - c) L'étiquette "MgtService" désigne un service de gestion (2).
 - d) Le numéro de ligne définit la séquence et le niveau d'imbrication de la ligne. Chaque niveau d'imbrication est identifié par un point. L'imbrication dans la liste de services sert à spécifier des services conditionnés à un énoncé de contrainte.

3.7.4 Conventions pour les définitions des services

3.7.4.1 Vue d'ensemble

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.

3.7.4.2 Paramètres du service

Les primitives de service sont utilisées pour représenter les interactions entre utilisateur de service et fournisseur de service (ISO/CEI 10731). Elles acheminent des paramètres qui indiquent des informations disponibles dans l'interaction entre utilisateur et fournisseur. Dans n'importe quelle interface particulière, il n'est pas indispensable d'énoncer tous les paramètres de façon explicite.

Les spécifications de service selon la présente norme utilisent un format de tableau pour décrire les paramètres de composants des primitives du service d'ASE. Les paramètres qui s'appliquent à chaque groupe de primitives de service sont consignés en tableaux. Chaque tableau comporte jusqu'à cinq colonnes pour le/la:

- 1) nom de paramètre;
- 2) primitive "request" (demande);
- 3) primitive "indication";
- 4) primitive "response" (réponse); et
- 5) primitive "confirm" (confirmation).

Un paramètre (ou un composant 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 spécifiée 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 du service. 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 du service;
(Blanc/Vide)	Le paramètre n'est jamais présent;
S	Le paramètre est un élément sélectionné.

Certaines entrées sont en plus qualifiées par des éléments entre parenthèses. Ceux-ci peuvent être

- c) une contrainte spécifique au 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;

- d) une indication qu'une certaine note s'applique à l'entrée:

"(n)" indique que la note "n" suivante contient des informations complémentaires relatives au paramètre et à son utilisation.

3.7.4.3 Procédures de service

Les procédures sont définies en termes des:

- interactions entre entités d'application par l'échange d'unités de données de protocole d'application de bus de terrain, et
- interactions entre un fournisseur de services de couche application et un utilisateur de service de couche application dans le même système par l'invocation de primitives de service de couche application.

Ces procédures sont applicables à des instances de communication entre systèmes qui prennent en charge des services de communications à contrainte temporelle au sein de la couche application de bus de terrain.

4 Concepts

4.1 Concepts communs

L'ensemble du CEI/TR 61158-1:2010, Article 9 est incorporé par référence, sauf neutralisation spécifique en 4.2.

4.2 Concepts spécifiques de type

4.2.1 Principe de fonctionnement

Le Type 22 consiste en deux types de modèles de communication: RTFL et RTFN. RTFL sert à assurer une communication temps réel cyclique synchronisée. RTFN sert à mettre en réseau plusieurs cellules RTFL en un système global assurant l'échange de données entre plusieurs cellules RTFL et entre des cellules RTFL et des dispositifs RTFN.

Dans ce contexte, une cellule RTFL décrit un segment de DL qui utilise le modèle RTFL pour la communication. Une cellule RTFL consiste en un dispositif-racine (RD) et un ou plusieurs dispositifs ordinaires (OD). L'élément central des cellules RTFL est le dispositif-racine qui organise et régule les séquences de cellules RTFL telles que l'envoi de trames temps réel cyclique. Un RD de RTFL a au moins une connexion à la RTFL et peut inclure une passerelle (GW) qui a en plus une connexion au RTFN. Étant donné que chaque OD dans la cellule RTFL peut seulement avoir une connexion RTFL, le RD comportant une GW fonctionne donc comme une liaison entre la RTFL et le RTFN. La communication RTFN n'est pas coordonnée comme la communication dans la RTFL, mais elle est utilisée par un réseau commuté duplex intégral conforme à la ISO/CEI 8802-3. Ainsi, aucun déterminisme ne peut être garanti pour le transfert de données RTFN.

Les données de processus et de services communication sont prises en charge par les réseaux de Type 22 en utilisant différents mécanismes (voies) dans la RTFL et le RTFN. Les données cycliques peuvent être transférées sur la voie de données cycliques (CDC). La voie de messages (MSC) permet une communication supplémentaire de données acycliques et sert à l'échange de données de services.

Les données de services sont typiquement transférées en étant acycliques et sont utilisées pour transférer des données de paramètres, de commandes de contrôle, de statut et de diagnostic ainsi que pour des segments de données généralement plus volumineux. Les données de services sont transférées soit guidées par des événements, soit guidées par l'utilisateur (caractère acyclique). Les données de paramètres utilisées en particulier dans la configuration des dispositifs n'exigent pas des conditions temporelles strictes alors que les données de diagnostic peuvent avoir de beaucoup plus grandes exigences de temps.

Par contraste, les données de processus sont typiquement transférées de manière cyclique avec des temps de cycle différents et de plus fortes exigences de temps réel.

La couche application (AL) de type 22 prend en charge une diversité de services et de protocoles pour satisfaire à ces exigences différentes. Les deux modèles de communication prennent en charge la même couche application de bus de terrain. Les services et les protocoles sont mis en correspondances aux services de DL correspondants.

4.2.2 Vue d'ensemble des modèles de communication

4.2.2.1 Vue d'ensemble

La technologie de Type 22 spécifie essentiellement deux modèles de communication avec les protocoles correspondants. La communication RTFL est destinée à la communication rapide dans une machine alors que le modèle RTFN prévoit la mise en réseau de machines ou cellules individuelles. Les protocoles correspondants visent à offrir un ensemble égal de services pour l'échange de données de processus cycliques ainsi que la communication de données de message acycliques.

La relation d'applications peut être modélisée indépendamment de la relation de communication.

4.2.2.2 Modèle de communication RTFL

Pour le modèle de communication RTFL, la communication suit une topologie en ligne. La communication RTFL est basée sur un transfert de données cycliques dans une DLPDU selon l'ISO/CEI 8802-3. Ce transfert élémentaire de données cycliques est assuré par un dispositif spécial, à savoir le dispositif-racine (RD). Les dispositifs-racines agissent comme maître de communication pour lancer de manière cyclique la communication. Les DLPDU produites par le dispositif-racine sont transmises aux dispositifs ordinaires (OD) de Type 22. Chaque dispositif ordinaire reçoit la DLPDU, écrit ses données et transmet la DLPDU. Un réseau RTFL exige exactement un et un seul dispositif-racine. Le dernier dispositif ordinaire d'un réseau RTFL renvoie la DLPDU traitée. La DLPDU est renvoyée en étant transférée dans l'ordre inverse des dispositifs jusqu'au dispositif-racine et, donc, elle est retournée par le premier dispositif ordinaire vers le dispositif-racine comme DLPDU de réponse. Dans le sens inverse, les dispositifs ordinaires lisent leurs données pertinentes dans la DLPDU.

4.2.2.3 Modèle de communication RTFN

Pour le modèle de communication RTFN, la communication est basée sur des connexions point à point entre les dispositifs participants.

La mise en réseau de différentes parties ou cellules RTFL d'un système d'automatisation en un système d'automatisation global est prise en charge par l'utilisation de la communication RTFN et des passerelles correspondantes.

4.2.3 Description de l'élément de couche application

4.2.3.1 CeS

L'ASE CeS obligatoire consiste en plusieurs attributs et décrit le principal élément de couche application pour bâtir une application temps réel distribuée.

4.2.3.2 Communication des DLPDU autres que de Type 22

L'ASE facultatif de communication des SEF décrit une possibilité d'utiliser une communication en tunnel autre que de Type 22 au sein d'un système de communication RTFL.

4.2.3.3 Gestion

L'ASE de gestion obligatoire consiste en un ensemble de services pour maîtriser l'état d'un réseau et des dispositifs participants. Les contraintes en services disponibles sont spécifiées pour les différents modèles de communication RTFL et RTFN.

4.2.4 Interaction producteur-consommateur

Le modèle d'interaction producteur-consommateur implique un producteur et zéro, un ou plusieurs consommateurs. Le modèle est caractérisé par un service non confirmé demandé par un producteur et une indication de service corrélée dans tous les consommateurs. La Figure 1 illustre l'interaction pour un producteur et deux consommateurs.

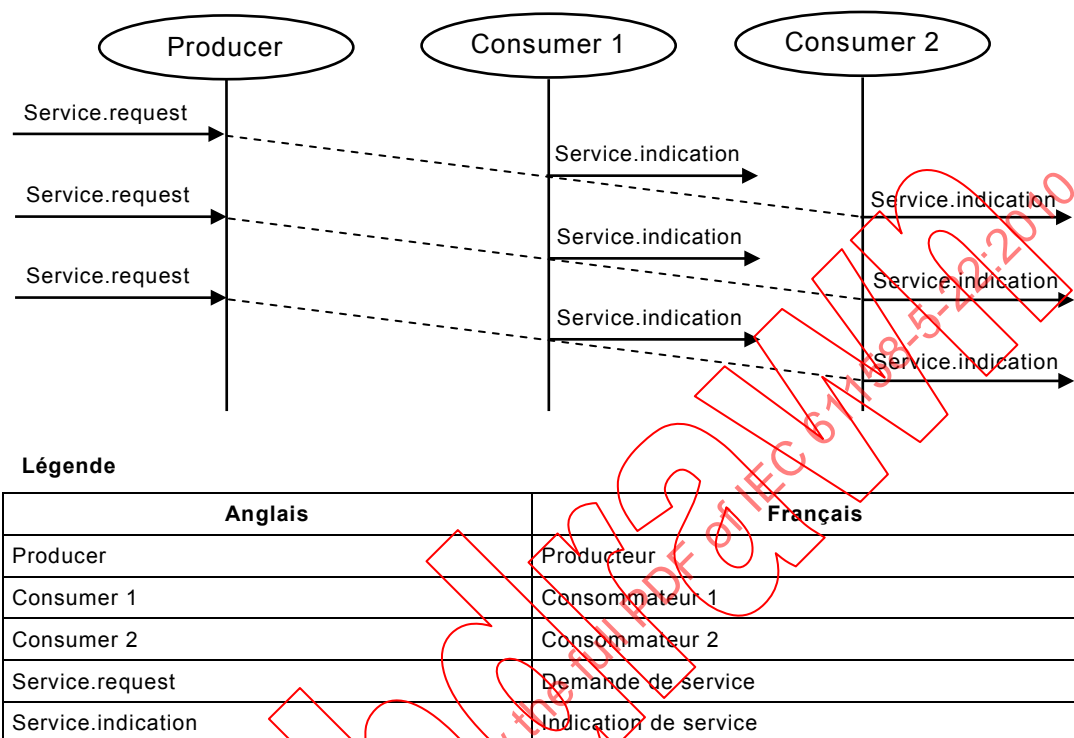


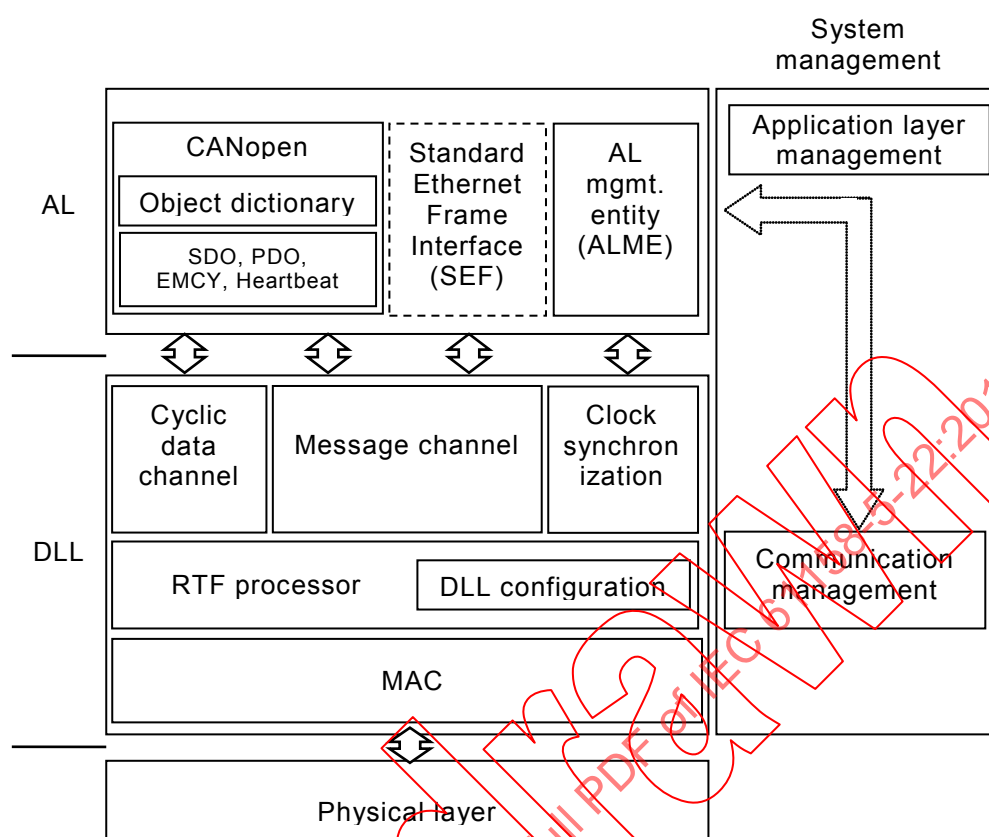
Figure 1 – Modèle d'interaction producteur-consommateur

Les services pris en charge par un modèle d'interaction sont acheminés par des points d'extrémité de relation d'applications (AREP) associés aux AP engagés dans la communication. Le rôle joué par l'AREP dans l'interaction (par exemple: producteur, consommateur) est défini comme un attribut de l'AREP.

4.2.5 Modèles de référence de dispositif

4.2.5.1 Modèle de référence des dispositifs RTFL

Les services de Type 22 sont décrits à l'aide des principes, de la méthodologie et du 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 22 définit la fonctionnalité de haut en bas d'un modèle de communications OSI complet. Les fonctions des couches OSI intermédiaires, les couches 3 à 6, sont consolidées soit en la couche liaison de données de Type 22, soit en la couche application de Type 22. Le modèle de référence des dispositifs pour les dispositifs RTFL de Type 22 est montré à la Figure 2.



Légende

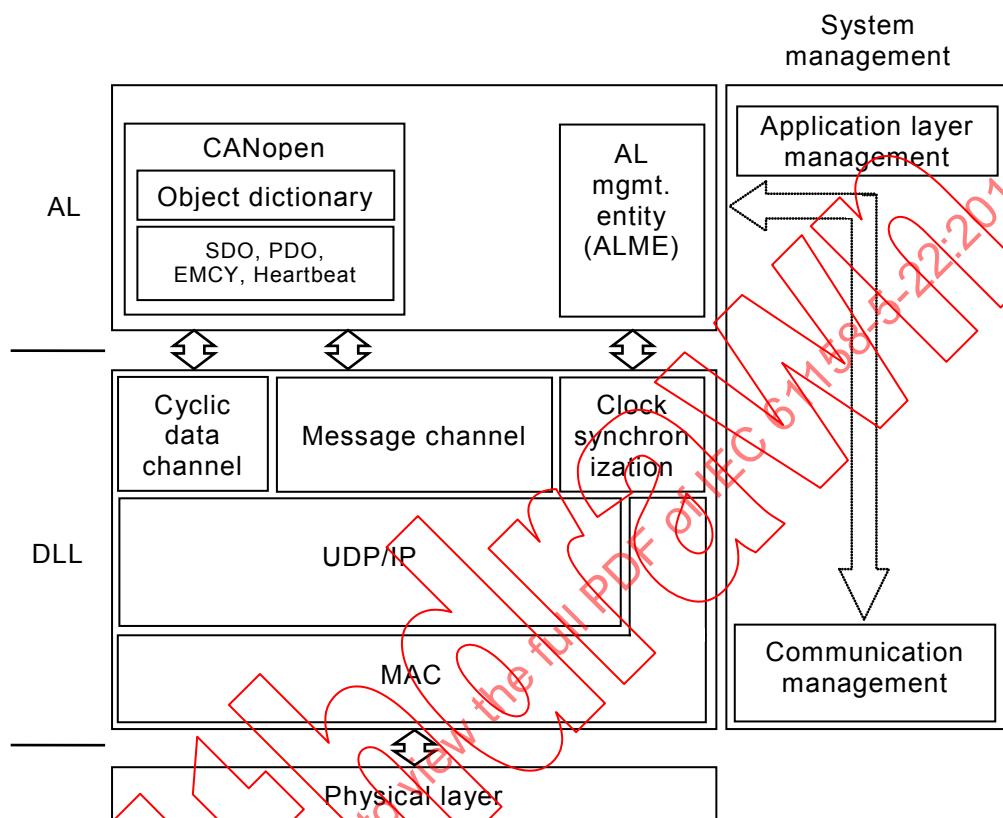
Anglais	Français
System management	Gestion de système
AL	Couche application
CANopen	CANopen
Object dictionary	Dictionnaire d'objets
SDO, PDO, EMCY, Heartbeat	SDO, PDO, EMCY, Heartbeat
Standard Ethernet Frame Interface (SEF)	Interface SEF (trame Ethernet normalisée)
AL mgmt. entity (ALME)	Entité de gestion de couche application (ALME)
Application layer management	Gestion de couche application
DLL	Couche liaison de données
Cyclic data channel	Voie de données cycliques
Message channel	Voie de messages
Clock synchronization	Synchronisation d'horloge
RTF processor	Processeur de trames temps réel
DLL configuration	Configuration de couche liaison de données
Communication management	Gestion de communication
MAC	Commande d'accès au support
Physical layer	Couche physique

Figure 2 – Modèle de référence des dispositifs RTFL

4.2.5.2 Modèle de référence des dispositifs RTFN

Les services de Type 22 sont décrits à l'aide des principes, de la méthodologie et du 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 22 définit la fonctionnalité de haut en bas d'un modèle de communications OSI complet. Les fonctions des couches OSI intermédiaires, les couches 3 à 6, sont consolidées soit en la couche liaison de données de Type 22, soit en la couche application de Type 22. Le modèle de référence des dispositifs pour les dispositifs RTFN de Type 22 est montré à la Figure 3.



Légende

Anglais	Français
System management	Gestion de système
AL	Couche application
CANopen	CANopen
Object dictionary	Dictionnaire d'objets
SDO, PDO, EMCY, Heartbeat	SDO, PDO, EMCY, Heartbeat
AL mgmt. entity (ALME)	Entité de gestion de couche application (ALME)
Application layer management	Gestion de couche application
DLL	Couche liaison de données
Cyclic data channel	Voie de données cycliques
Message channel	Voie de messages
Clock synchronization	Synchronisation d'horloge
Communication management	Gestion de communication
UDP/IP	Protocole UDP/IP
MAC	Commande d'accès au support
Physical layer	Couche physique

Figure 3 – Modèle de référence des dispositifs RTFN

5 ASE Data type (type de données)

5.1 Vue d'ensemble

L'ensemble du CEI/TR 61158-1:2010, 10.1 est incorporé par référence.

5.2 Définition formelle des objets de types de données

L'ensemble du CEI/TR 61158-1:2010, 10.2 est incorporé par référence.

5.3 Types de données définis pour la FAL

5.3.1 Types Fixed length (longueur fixe)

5.3.1.1 Types Boolean (booléens)

CLASS: Data type (Type de données)

ATTRIBUTES:

1	Data type numeric identifier (Identificateur numérique de type de données)	= 1
2	Data Type Name (Nom de type de données)	= Boolean
3	Format	= FIXED LENGTH
4.1	Octet Length (Longueur en octets)	= 1

Ce type de données exprime le type de données Boolean avec les valeurs TRUE et FALSE.

5.3.1.2 Types Bitstring (chaîne de bits)

Il n'y a pas de types Bitstring définis pour le Type 22.

5.3.1.3 Types Currency (monnaies)

Il n'y a pas de types Currency définis pour le Type 22.

5.3.1.4 Types Date/Time (date et heure)

5.3.1.4.1 TimeOfDay

CLASS: Type de données

ATTRIBUTES:

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

Ce type de données est constitué de deux éléments de valeurs unsigned (non signées) et exprime l'heure du jour et la date. Le premier élément est un type de données Unsigned32 et donne l'heure après minuit en millisecondes. Le second élément est un type de données Unsigned16 et donne la date en comptant les jours à partir du 1^{er} janvier 1984.

5.3.1.4.2 TimeDifference

CLASS: **Type de données**

ATTRIBUTS:

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

Ce type de données est constitué de deux éléments de valeurs unsigned (non signées) qui expriment la différence de temps. Le premier élément est un type de données Unsigned32 qui donne une partie fractionnaire d'un jour en millisecondes. Le second élément facultatif est un type de données Unsigned16 qui donne la différence en jours.

5.3.1.5 Types Enumerated (énumérés)

Il n'y a pas de types Enumerated définis pour le Type 22.

5.3.1.6 Types Handle (identification)

Il n'y a pas de types Handle définis pour le Type 22.

5.3.1.7 Types numériques

5.3.1.7.1 Float

Ce type de données est le même que le type Float32.

5.3.1.7.2 Float32

CLASS: **Type de données**

ATTRIBUTS:

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

Ce type a une longueur de quatre octets. Le format de float32 est celui défini par la CEI 60559 comme étant en simple précision ("single precision").

5.3.1.7.3 Double

Ce type de données est le même que le type Float64.

5.3.1.7.4 Float64

CLASS: **Type de données**

ATTRIBUTS:

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

Ce type a une longueur de huit octets. Le format de float64 est celui défini par la CEI 60559 comme étant en double précision ("double precision").

5.3.1.8 Types Integer (entiers)

5.3.1.8.1 Integer8

CLASS: Type de données

ATTRIBUTS:

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

Ce type entier (Integer) est un nombre binaire en complément à deux avec une longueur égale à un octet.

5.3.1.8.2 SINT

Ce type de la CEI 61131-3 est le même que le type Integer8.

5.3.1.8.3 Char

Ce type de données est le même que le type Integer8.

5.3.1.8.4 Integer16

CLASS: Type de données

ATTRIBUTS:

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

Ce type entier (Integer) est un nombre binaire en complément à deux avec une longueur égale à deux octets.

5.3.1.8.5 INT

Ce type de la CEI 61131-3 est le même que le type Integer16.

5.3.1.8.6 Short

Ce type de données est le même que le type Integer16.

5.3.1.8.7 Integer24

CLASS: Type de données

ATTRIBUTS:

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

Ce type integer est un nombre binaire en complément à deux avec une longueur égale à trois octets.

5.3.1.8.8 Integer32

CLASS: **Type de données**

ATTRIBUTS:

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

Ce type entier (Integer) est un nombre binaire en complément à deux avec une longueur égale à quatre octets.

5.3.1.8.9 DINT

Ce type de la CEI 61131-3 est le même que le type Integer32.

5.3.1.8.10 Long

Ce type de données est le même que le type Integer32.

5.3.1.8.11 Integer40

CLASS: **Type de données**

ATTRIBUTS:

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

Ce type integer est un nombre binaire en complément à deux avec une longueur égale à cinq octets.

5.3.1.8.12 Integer48

CLASS: **Type de données**

ATTRIBUTS:

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

Ce type integer est un nombre binaire en complément à deux avec une longueur égale à six octets.

5.3.1.8.13 Integer56

CLASS: **Type de données**

ATTRIBUTS:

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

Ce type integer est un nombre binaire en complément à deux avec une longueur égale à sept octets.

5.3.1.8.14 Integer64**CLASS:** Type de données**ATTRIBUTES:**

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

Ce type entier (Integer) est un nombre binaire en complément à deux avec une longueur égale à huit octets.

5.3.1.8.15 LINT

Ce type de la CEI 61131-3 est le même que le type Integer64.

5.3.1.9 Types non signés (Unsigned)**5.3.1.9.1 Unsigned8****CLASS:** Type de données**ATTRIBUTES:**

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

Ce type est un nombre binaire. Le bit de poids fort de l'octet de poids fort est toujours utilisé comme le bit de poids fort du nombre binaire; aucun bit de signe n'est inclus. Ce type a une longueur égale à un octet.

5.3.1.9.2 USINT

Ce type de la CEI 61131-3 est le même que le type Unsigned8.

5.3.1.9.3 Unsigned char

Ce type de données est le même que le type Unsigned8.

5.3.1.9.4 Unsigned16**CLASS:** Type de données**ATTRIBUTES:**

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

Ce type est un nombre binaire. Le bit de poids fort de l'octet de poids fort est toujours utilisé comme le bit de poids fort du nombre binaire; aucun bit de signe n'est inclus. Ce type a une longueur de deux octets.

5.3.1.9.5 UINT

Ce type de la CEI 61131-3 est le même que le type Unsigned16.

5.3.1.9.6 WORD

Ce type est utilisé de la même façon que le type UINT.

5.3.1.9.7 Unsigned24

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

Ce type est un nombre binaire. Le bit de poids fort de l'octet de poids fort est toujours utilisé comme le bit de poids fort du nombre binaire; aucun bit de signe n'est inclus. Ce type a une longueur de trois octets.

5.3.1.9.8 Unsigned32

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

Ce type est un nombre binaire. Le bit de poids fort de l'octet de poids fort est toujours utilisé comme le bit de poids fort du nombre binaire; aucun bit de signe n'est inclus. Ce type a une longueur de quatre octets.

5.3.1.9.9 UDINT

Ce type de la CEI 61131-3 est le même que le type Unsigned32.

5.3.1.9.10 DWORD

Ce type est utilisé de la même façon que le type UDINT.

5.3.1.9.11 Unsigned40

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

Ce type est un nombre binaire. Le bit de poids fort de l'octet de poids fort est toujours utilisé comme le bit de poids fort du nombre binaire; aucun bit de signe n'est inclus. Ce type a une longueur de cinq octets.

5.3.1.9.12 Unsigned48

CLASS:		Type de données
ATTRIBUTES:		
1	Data type Numeric Identifier	= 25
2	Data type Name	= Unsigned48

3	Format	=	FIXED LENGTH
4.1	Octet Length	=	6

Ce type est un nombre binaire. Le bit de poids fort de l'octet de poids fort est toujours utilisé comme le bit de poids fort du nombre binaire; aucun bit de signe n'est inclus. Ce type a une longueur de six octets.

5.3.1.9.13 Unsigned56

CLASS: Type de données

ATTRIBUTES:

1	Data type Numeric Identifier	=	26
2	Data type Name	=	Unsigned56
3	Format	=	FIXED LENGTH
4.1	Octet Length	=	7

Ce type est un nombre binaire. Le bit de poids fort de l'octet de poids fort est toujours utilisé comme le bit de poids fort du nombre binaire; aucun bit de signe n'est inclus. Ce type a une longueur de sept octets.

5.3.1.9.14 Unsigned64

CLASS: Type de données

ATTRIBUTES:

1	Data type Numeric Identifier	=	27
2	Data type Name	=	Unsigned64
3	Format	=	FIXED LENGTH
4.1	Octet Length	=	8

Ce type est un nombre binaire. Le bit de poids fort de l'octet de poids fort est toujours utilisé comme le bit de poids fort du nombre binaire; aucun bit de signe n'est inclus. Ce type a une longueur de huit octets.

5.3.1.9.15 ULINT

Ce type de la CEI 61131-3 est le même que le type Unsigned64.

5.3.1.9.16 Unsigned128

CLASS: Type de données

ATTRIBUTES:

1	Data type Numeric Identifier	=	28
2	Data type Name	=	Unsigned128
3	Format	=	FIXED LENGTH
4.1	Octet Length	=	16

Ce type est un nombre binaire. Le bit de poids fort de l'octet de poids fort est toujours utilisé comme le bit de poids fort du nombre binaire; aucun bit de signe n'est inclus. Ce type a une longueur de seize octets.

5.3.1.9.17 Unsigned256

CLASS: Type de données

ATTRIBUTES:

1	Data type Numeric Identifier	=	29
2	Data type Name	=	Unsigned256

3	Format	=	FIXED LENGTH
4.1	Octet Length	=	32

Ce type est un nombre binaire. Le bit de poids fort de l'octet de poids fort est toujours utilisé comme le bit de poids fort du nombre binaire; aucun bit de signe n'est inclus. Ce type a une longueur de trente-deux octets.

5.3.1.10 Types Pointer (pointeurs)

Il n'y a pas de types Pointer définis pour le Type 22.

5.3.1.11 Types OctetString (chaîne d'octets)

Il n'y a pas de types OctetString de longueur fixe définis pour le Type 22.

5.3.1.12 Types de caractères VisibleString (chaîne visible)

Il n'y a pas de types VisibleString de longueur fixe définis pour le Type 22.

5.3.2 Types String (chaîne)

5.3.2.1 OctetString

CLASS:		Type de données
ATTRIBUTES:		
1	Data type Numeric Identifier	= 10
2	Data type Name	= OctetString
3	Format	= STRING
4.1	Octet Length	= 1 à n

Un OctetString est une séquence ordonnée d'octets, numérotés de 1 à n .

NOTE La CEI 61158-6-22:2010 définit l'ordre d'émission.

5.3.2.2 VisibleString

CLASS:		Type de données
ATTRIBUTES:		
1	Data type Numeric Identifier	= 9
2	Data type Name	= VisibleString
3	Format	= STRING
4.1	Octet Length	= 1 à n

Ce type est défini comme étant le type "string" (chaîne) de l'ISO/CEI 646.

5.3.2.3 UnicodeString

CLASS:		Type de données
ATTRIBUTES:		
1	Data type Numeric Identifier	= 11
2	Data type Name	= UnicodeString
3	Format	= STRING
4.1	Octet Length	= 1 à n

Ce type est défini comme étant le type "string" (chaîne) de l'ISO/CEI 10646.

5.3.3 Domain (c'est-à-dire Domaine)

CLASS: Type de données

ATTRIBUTES:

1	Data type Numeric Identifier	=	15
2	Data type Name	=	Domain (c'est-à-dire Domaine)
3	Format	=	STRING
4.1	Octet Length	=	1 à n

Grande quantité variable de données. Par exemple: programme exécutable.

6 Spécification de modèle de communication

6.1 Éléments de service application (les ASE)

6.1.1 ASE CeS

6.1.1.1 Vue d'ensemble

6.1.1.1.1 Informations générales

CANopen est une norme ouverte de bus de terrain non propriétaire qui est spécifiée et normalisée par l'organisation CiA (CAN in Automation).

NOTE CANopen (la spécification particulière CiA DS 301) est normalisé comme EN 50325-4.

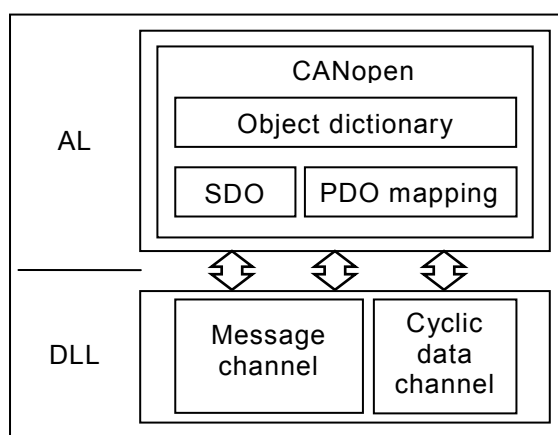
Conjointement au protocole basé sur le CAN Bus, une couche application uniforme et normalisée est prévue pour les applications industrielles. Elle inclut la normalisation de communication, y compris les caractéristiques techniques et fonctionnelles des dispositifs distribués d'automation de terrain et la normalisation d'objets d'application en utilisant des profils de dispositifs.

Les profils de dispositifs constituent l'un des éléments centraux de la CANopen, spécifiant fonctions uniformes et des paramètres/objets normalisés pour différents domaines d'application ou pour des groupes de dispositifs automatisés. Sur la base de ces profils normalisés, un haut niveau de compatibilité des vendeurs peut être atteint en raison de l'interopérabilité et de l'interchangeabilité des dispositifs construits par des fabricants différents. Tous les types majeurs de dispositifs utilisés en ingénierie d'automation tels que:

- les dispositifs E/S numériques et analogiques;
- les entraînements;
- les soupapes;
- les commandes programmables;
- les codeurs, etc.

sont déjà normalisés comme profils de dispositifs et sont reflétés dans la spécification pertinente de profils de dispositifs CiA.

Le dictionnaire d'objets contient des paramètres, des données d'application et les informations de mise en correspondance entre objets de données de processus et données d'application (mise en correspondance de PDO). Ses rubriques sont accessibles par l'intermédiaire d'objets de données de service (SDO), comme montré à la Figure 4.



Légende

Anglais	Français
AL	Couche application
CANopen	CANopen
Object dictionary	Dictionnaire d'objets
SDO	SDO (Objet de données de service)
PDO mapping	Mise en correspondance de PDO (objets de données de processus)
DLL	Couche liaison de données
Message channel	Voie de messages
Cyclic data channel	Voie de données cycliques

Figure 4 – Structure de dispositif CeS de Type 22

6.1.1.1.2 Structure de dictionnaire d'objets

Le dictionnaire d'objets est l'interface entre l'application et les sous-systèmes de communication. Essentiel comme élément central, le dictionnaire d'objets est un regroupement d'objets et spécifie des paramètres, données et fonctions de dispositifs et de communication qui sont stockés et récupérés avec l'aide d'objets. Il est une collection des structures de données de paramètres de dispositifs qui sont accessibles par l'intermédiaire des services SDO Upload (téléchargement vers l'amont d'objets de données de services) et SDO Download (téléchargement vers l'aval d'objets de données de services).

Le dictionnaire est organisé sous forme de tableau tel qu'indiqué dans le Tableau 1 ci-dessous. La structure correspond à la Spécification CANopen 301 connue de l'automatisation industrielle.

NOTE CANopen (la spécification particulière CiA DS 301) est normalisé comme EN 50325-4.

Tableau 1 – Structure de dictionnaire d'objets

Section	Sous-section	Contenu
Type de Données	Types de données de base	Définition de types de données de base
—	Types de données complexes	Définition de types de données complexes
—	Types de données spécifiques à un fabricant	Définition de types de données spécifiques à un fabricant
—	Types de données de base spécifiques à un profil de dispositifs	Définition de types de données de base spécifiques à un profil de dispositifs
—	Types de données complexes spécifiques à un profil de dispositifs	Définition de types de données complexes spécifiques à un profil de dispositifs
Profil de communication	—	Définition des paramètres qui sont utilisés pour les besoins de configuration de communication et de communications spécialisées
Profil défini par le fabricant	—	Définition des paramètres spécifiques à un fabricant
Profil normalisé de dispositifs	—	Définition des paramètres définis dans un profil normalisé de dispositifs
Profil normalisé d'interfaces	—	Définition des paramètres définis dans un profil normalisé d'interfaces
Profil d'interfaces RTFN de Type 22	—	Définition des paramètres définis dans un profil d'interfaces RTFN de Type 22

6.1.1.1.3 Zones de types de données

La zone de type de données est constituée des parties suivantes:

Types de données de base

Définition de types de données simples généraux

Types de données complexes

Définition de types de données structurés généraux

Types de données complexes spécifiques à un fabricant

Définition de types de données structurés spécifiques à un fabricant

Types de données de base spécifiques à un profil de dispositifs

Définition de types de données simples spécifiques à un profil de dispositifs

Types de données complexes spécifiques à un profil de dispositifs

Définition de types de données structurés spécifiques à un profil de dispositifs

6.1.1.1.4 Zone de communication**6.1.1.1.4.1 Device type**

L'objet "device type" (type de dispositifs) est constitué du paramètre suivant:

Paramètre

Device profile number

Ce paramètre spécifie le profil de dispositifs qui est utilisé pour le dispositif.

Additional information

Ce paramètre spécifie des informations complémentaires définies par le profil de dispositifs.

6.1.1.1.4.2 Error register

L'objet "error register" (registre d'erreurs) est constitué du paramètre suivant:

Paramètre

Generic Error

Ce paramètre indique la présence d'une erreur générique.

Current

Ce paramètre indique la présence d'une erreur de courant.

Voltage

Ce paramètre indique la présence d'une erreur de tension.

Temperature

Ce paramètre indique la présence d'une erreur de température.

Communication

Ce paramètre indique la présence d'une erreur de communication.

Device profile specific

Ce paramètre indique la présence d'une erreur spécifique à un profil de dispositifs

Manufacturer specific

Ce paramètre indique la présence d'une erreur spécifique à un fabricant.

6.1.1.1.4.3 Manufacturer status register

L'objet "manufacturer status register" (registre de statuts de fabricant) est constitué du paramètre suivant:

Paramètre

Status register

Ce paramètre spécifie le registre de statuts du dispositif.

6.1.1.1.4.4 Event log

L'objet "event log" (journal d'événements) est constitué du paramètre suivant:

Paramètre

Number of entries

Ce paramètre spécifie le nombre d'entrées pour cet objet.

Emergency error code

Ce paramètre spécifie le code d'erreur d'urgence pour la survenue d'un événement.

Manufacturer specific error field

Ce paramètre spécifie le champ "erreur spécifique à un fabricant" pour la survenue d'un événement.

Time stamp

Ce paramètre spécifie la survenue dans le temps d'un événement.

Length

Ce paramètre spécifie la longueur des informations étendues du fabricant.

Extended manufacturer information

Ce paramètre spécifie les informations étendues spécifiques à un fabricant.

6.1.1.1.4.5 Manufacturer device name

L'objet "manufacturer device name" (nom de dispositif de fabricant) est constitué du paramètre suivant:

Paramètre

Device name

Ce paramètre spécifie le nom de dispositif relatif au dispositif.

6.1.1.1.4.6 Manufacturer hardware version

L'objet "manufacturer hardware version" (version de matériel du fabricant) est constitué du paramètre suivant:

Paramètre

Hardware version

Ce paramètre spécifie la version de matériel de fabricant pour le dispositif.

6.1.1.1.4.7 Manufacturer software version

L'objet "manufacturer software version" (version de logiciel du fabricant) est constitué du paramètre suivant:

Paramètre

Software version

Ce paramètre spécifie la version de logiciel de fabricant pour le dispositif.

6.1.1.1.4.8 Communication layer configuration

L'objet "communication layer configuration " (configuration de couche communication) est constitué du paramètre suivant:

Paramètre

Number of entries

Ce paramètre spécifie le nombre d'entrées pour cet objet.

Symbolic device name

Ce paramètre spécifie le nom de dispositif relatif au dispositif.

Device role

Ce paramètre spécifie le rôle du dispositif au sein du système de communication.

RTFN base cycle time

Ce paramètre spécifie le temps de cycle de base RTFN du dispositif.

IP Address

Ce paramètre spécifie l'adresse IP du dispositif.

Subnet mask

Ce paramètre spécifie le masque de sous-réseau de ce dispositif.

Default gateway

Ce paramètre spécifie la passerelle par défaut du dispositif.

DHCP enabled

Ce paramètre spécifie l'utilisation de DHCP pour le dispositif.

Current IP configuration

Ce paramètre spécifie l'activation pour la configuration IP.

6.1.1.1.4.9 Time sync IRQ configuration

L'objet "time sync IRQ configuration" (configuration de demande d'interruption de synchronisation du temps) est constitué du paramètre suivant:

Paramètre

Sync ID number

Ce paramètre spécifie le numéro interne de dispositif pour le "time sync ID" (identificateur de synchronisation du temps).

Number of entries

Ce paramètre spécifie le nombre d'entrées de "time sync IRQ configuration".

Time sync ID

Ce paramètre spécifie un identificateur propre au réseau pour un groupe de dispositifs synchrones.

Cycle time

Ce paramètre spécifie une valeur qui déclenche un signal de synchronisation spécifique (IRQ).

Time offset

Ce paramètre spécifie une valeur et décrit le décalage temporel d'un dispositif par rapport au maître de synchronisation (SYNC master).

Is master

Ce paramètre spécifie le rôle du dispositif pour le mécanisme de synchronisation.

IPv4 address sync master

Ce paramètre spécifie l'adresse IP pour IPv4 du dispositif maître de synchronisation.

IPv6 address sync master

Ce paramètre spécifie l'adresse IP pour IPv6 du dispositif maître de synchronisation.

6.1.1.1.4.10 Time sync IRQ state

L'objet "time sync IRQ state" (état d'IRQ de synchronisation du temps) est constitué du paramètre suivant:

Paramètre

Sync ID number

Ce paramètre spécifie le numéro interne de dispositif de l'ID de synchronisation de temps (time sync ID).

Number of entries

Ce paramètre spécifie le nombre d'entrées de "time sync IRQ state".

Time sync IRQ state

Ce paramètre spécifie l'état de synchronisation pour un ID spécifique de synchronisation de temps du dispositif.

6.1.1.1.4.11 Store parameters

L'objet "store parameters" (stocker des paramètres) est constitué du paramètre suivant:

Paramètre

Number of entries

Ce paramètre spécifie le nombre d'entrées.

All parameters

Ce paramètre spécifie une commande de stocker tous les paramètres de dispositifs.

Communication profile

Ce paramètre spécifie une commande de stocker tous les paramètres de profils de communication.

Application parameters

Ce paramètre spécifie une commande de stocker tous les paramètres d'application.

Manufacturer specific parameters

Ce paramètre spécifie une commande de stocker des paramètres ou groupes de paramètres spécifiques à un fabricant.

6.1.1.1.4.12 Restore default parameters

L'objet "restore default parameters" (restaurer les paramètres par défaut) est constitué du paramètre suivant:

Paramètre

Number of entries

Ce paramètre spécifie le nombre d'entrées.

All parameters

Ce paramètre spécifie une commande de restaurer tous les paramètres par défaut des dispositifs.

Communication profile

Ce paramètre spécifie une commande de restaurer tous les paramètres par défaut des profils de communication.

Application parameters

Ce paramètre spécifie une commande de restaurer tous les paramètres d'application par défaut.

Manufacturer specific parameters

Ce paramètre spécifie une commande de restaurer les paramètres par défaut spécifiques à un fabricant ou les paramètres par défauts de groupes de paramètres.

6.1.1.1.4.13 Diagnostic information

L'objet "diagnostic information" (informations de diagnostic) est constitué du paramètre suivant:

Paramètre

Number of entries

Ce paramètre spécifie le nombre d'entrées.

Application layer state

Ce paramètre spécifie les informations relatives à l'état de la couche application.

Application state

Ce paramètre spécifie les informations relatives à l'état de l'application.

CL state RTFL

Ce paramètre spécifie les informations relatives à l'état de la couche communication pour la RTFL.

CL state RTFN

Ce paramètre spécifie les informations relatives à l'état de la couche communication pour le RTFN.

Number of delayed RTFL DLPDUs

Ce paramètre spécifie les informations relatives au nombre des DLPDU différées de la RTFL.

Number of corrupt DLPDUs

Ce paramètre spécifie les informations relatives au nombre des DLPDU corrompues de la RTFL.

Number of received DLPDUs since startup

Ce paramètre spécifie les informations relatives au nombre des DLPDU reçues depuis le démarrage

Number of MSC buffer overflows

Ce paramètre spécifie les informations relatives au nombre des dépassements de capacité de tampon de la MSC.

Number of received MSC messages since startup

Ce paramètre spécifie les informations relatives au nombre de messages MSC reçus depuis le démarrage

Cable attenuation port 1

Ce paramètre spécifie les informations relatives à l'affaiblissement de câbles pour le port 1.

Cable attenuation port 2

Ce paramètre spécifie les informations relatives à l'affaiblissement de câbles pour le port 2.

Cable length port 1

Ce paramètre spécifie les informations relatives à la longueur de câbles pour le câblage de port 1.

Cable length port 2

Ce paramètre spécifie les informations relatives à la longueur de câbles pour le câblage de port 2.

Distance to fault port 1

Ce paramètre spécifie les informations relatives à la distance à un défaut de câblage pour le port 1.

Distance to fault port 2

Ce paramètre spécifie les informations relatives à la distance à un défaut de câblage pour le port 2.

6.1.1.1.4.14 Diagnostic thresholds

L'objet "diagnostic thresholds" (seuils de diagnostic) est constitué du paramètre suivant:

Paramètre

Number of entries

Ce paramètre spécifie le nombre d'entrées.

Expected RTFL round trip time

Ce paramètre spécifie les informations relatives au temps prévu d'aller-retour de RTFL.

Delayed RTFL rate threshold

Ce paramètre spécifie les informations relatives au seuil de débit différé de la RTFL.

Corrupt DLPDU rate threshold

Ce paramètre spécifie les informations relatives au seuil de débit de DLPDU corrompue.

MSC buffer overflows rate threshold

Ce paramètre spécifie les informations relatives au seuil de fréquence des dépassements de capacité de tampon de la MSC.

Cable attenuation port 1 threshold

Ce paramètre spécifie les informations relatives au seuil de l'affaiblissement de câbles pour le port 1.

Cable attenuation port 2 threshold

Ce paramètre spécifie les informations relatives au seuil de l'affaiblissement de câbles pour le port 2.

6.1.1.1.4.15 IP address EMCY

L'objet "IP address EMCY" (adresse IP d'urgence) est constitué du paramètre suivant:

Paramètre

IP address

Ce paramètre spécifie l'adresse IP pour le dispositif de destination pour des messages EMCY.

6.1.1.1.4.16 Inhibit time EMCY

L'objet "inhibit time EMCY" (temps de neutralisation d'urgence) est constitué du paramètre suivant:

Paramètre

Inhibit time

Ce paramètre spécifie un temps de neutralisation de message pour les messages d'urgence.

6.1.1.1.4.17 Consumer heartbeat list

L'objet "consumer heartbeat list" (liste de «heartbeat» de consommateurs) est constitué du paramètre suivant:

Paramètre

Heartbeat producer number

Ce paramètre spécifie le numéro interne de dispositif du producteur de «heartbeat» («battements de cœur») à surveiller.

Number of entries

Ce paramètre spécifie le nombre d'entrées de la liste de «heartbeat» de consommateurs.

RTFL PID

Ce paramètre spécifie l'ID de paquet RTFL du «heartbeat» surveillé.

RTFN PID

Ce paramètre spécifie l'ID de paquet RTFL du «heartbeat» surveillé.

TT

Ce paramètre spécifie le type d'émission du «heartbeat» surveillé.

Heartbeat time

Ce paramètre spécifie le temps «heartbeat» comme un multiple du temps de cycle de base.

Cycle multiplier

Ce paramètre spécifie le cycle d'émission prévu comme un multiplicateur du temps de cycle du système de communication dans le cas d'une communication intercellulaire.

Cycle offset

Ce paramètre spécifie un décalage pour le cycle d'émission en relation avec le cycle du système de communication dans le cas d'une communication intercellulaire.

Device address

Ce paramètre spécifie l'adresse de dispositif du producteur de «heartbeat».

IPv4 address

Ce paramètre spécifie l'adresse IP (IPv4) du producteur de «heartbeat».

IPv6 address

Ce paramètre spécifie l'adresse IP (IPv6) du producteur de «heartbeat».

6.1.1.1.4.18 Producer heartbeat parameter

L'objet "producer heartbeat parameter" (paramètre de «heartbeat» de producteur) est constitué du paramètre suivant:

Paramètre

Number of entries

Ce paramètre spécifie le nombre d'entrées valides.

RTFL PID

Ce paramètre spécifie l'ID de paquet RTFL.

RTFN PID

Ce paramètre spécifie l'ID de paquet RTFN.

Transmission type

Ce paramètre spécifie le type d'émission du heartbeat.

Time sync ID

Ce paramètre spécifie un ID de synchronisation de temps attribué au heartbeat dans le cas d'un traitement de «heartbeat» synchronisé.

Cycle multiplier

Ce paramètre spécifie le cycle d'émission comme un multiple du temps de cycle du système de communication.

Cycle offset

Ce paramètre spécifie le cycle d'émission du heartbeat.

Device address

Ce paramètre spécifie l'adresse de dispositif du destinataire du heartbeat pour la communication basée sur la voie de messages.

IPv4 address

Ce paramètre spécifie l'adresse IPv4 du destinataire du heartbeat pour la communication intercellulaire basée sur la voie de messages.

IPv6 address

Ce paramètre spécifie l'adresse IPv6 du destinataire du heartbeat pour la communication intercellulaire basée sur la voie de messages.

6.1.1.1.4.19 Objet "Identity"

L'objet "identity" (identité) est constitué du paramètre suivant:

Paramètre

Number of entries

Ce paramètre spécifie le nombre d'entrées.

Vendor ID

Ce paramètre spécifie l'ID de vendeur du dispositif.

Product code

Ce paramètre spécifie le code produit du dispositif.

Revision number

Ce paramètre spécifie le numéro de révision du dispositif.

Serial number

Ce paramètre spécifie le numéro de série du dispositif.

Type 22 protocol version

Ce paramètre spécifie la version de protocole de Type 22 du dispositif.

6.1.1.1.4.20 SDO protocol timeout

L'objet "SDO protocol timeout" (temporisation de protocole SDO) est constitué du paramètre suivant:

Paramètre

Timeout time

Ce paramètre spécifie le temps de temporisation de protocole SDO pour le dispositif.

6.1.1.1.4.21 Enable client SDO parameter

L'objet "enable client SDO parameter " (activer le paramètre SDO de client) est constitué du paramètre suivant:

Paramètre

Enable client SDO

Ce paramètre spécifie une commande pour activer le SDO client pour le dispositif.

6.1.1.1.4.22 Enable EMCY

L'objet "enable EMCY" (activer EMCY) est constitué du paramètre suivant:

Paramètre

Enable EMCY

Ce paramètre spécifie une commande d'activer l'EMCY pour le dispositif.

6.1.1.1.4.23 PDO timeout tolerance

L'objet "PDO timeout tolerance" (tolérance de temporisation de PDO) est constitué du paramètre suivant:

Paramètre

Timeout tolerance

Ce paramètre spécifie la tolérance de temporisation de PDO pour le dispositif.

6.1.1.1.4.24 Store EDS

L'objet "store EDS" (stocker des feuilles électroniques de données) est constitué du paramètre suivant:

Paramètre

Store EDS

Ce paramètre spécifie le fichier EDS du dispositif.

6.1.1.1.4.25 Storage format

L'objet "storage format" (format de stockage) est constitué du paramètre suivant:

Paramètre

Format

Ce paramètre spécifie le format du fichier du dispositif.

6.1.1.1.4.26 OS command

L'objet "OS command" (commande de système d'exploitation) est constitué du paramètre suivant:

Paramètre

Number of entries

Ce paramètre spécifie le nombre d'entrées.

Command

Ce paramètre spécifie une commande de système d'exploitation.

Status

Ce paramètre spécifie un statut de l'exécution d'une commande.

Reply

Ce paramètre spécifie une réponse à une commande.

6.1.1.1.4.27 OS command mode

L'objet "OS command mode" (mode de commande de système d'exploitation) est constitué du paramètre suivant:

Paramètre

Command mode

Ce paramètre spécifie l'exécution de commande de système d'exploitation.

6.1.1.1.4.28 OS debugger interface

L'objet "OS debugger interface" (interface de débogueur de système d'exploitation) est constitué du paramètre suivant:

Paramètre

Number of entries

Ce paramètre spécifie le nombre d'entrées.

Command

Ce paramètre spécifie une commande de système d'exploitation.

Status

Ce paramètre spécifie un statut pour l'exécution d'une commande.

Reply

Ce paramètre spécifie une réponse à la commande exécutée.

6.1.1.1.4.29 OS prompt

L'objet "OS prompt" (invite de système d'exploitation) est constitué du paramètre suivant:

Paramètre

Number of entries

Ce paramètre spécifie le nombre d'entrées.

StdIn

Ce paramètre spécifie la voie d'entrées normalisée de système d'exploitation.

StdOut

Ce paramètre spécifie la voie de sorties normalisée de système d'exploitation.

StdErr

Ce paramètre spécifie la voie d'erreurs normalisée de système d'exploitation.

6.1.1.1.4.30 Module list

L'objet "module list" (liste de modules) est constitué du paramètre suivant:

Paramètre

Number of entries

Ce paramètre spécifie le nombre de modules connectés.

Module

Ce paramètre spécifie le type et la caractéristique de module relatifs aux modules connectés d'un dispositif modulaire.

6.1.1.1.4.31 Emergency subscriber

L'objet "module list" (liste de modules) est constitué du paramètre suivant:

Paramètre

Number of entries

Ce paramètre spécifie le nombre de messages d'urgence pris en abonnement.

Device address

Ce paramètre spécifie l'adresse de dispositif du producteur de messages d'urgence.

IP address

Ce paramètre spécifie l'adresse IP du producteur de messages d'urgence.

Additional information

Ce paramètre spécifie des informations complémentaires pour un producteur.

6.1.1.1.4.32 Client SDO parameter

L'objet "client SDO parameter" (paramètre SDO de client) est constitué du paramètre suivant:

Paramètre

Number of entries

Ce paramètre spécifie le nombre d'entrées.

Server address

Ce paramètre spécifie l'adresse du serveur.

6.1.1.1.4.33 Receive PDO communication parameter

L'objet "receive PDO communication parameter" (paramètre de communication d'objets de données de processus de réception) est constitué du paramètre suivant:

Paramètre

Number of entries

Ce paramètre spécifie le nombre d'entrées valides.

RTFL PID

Ce paramètre spécifie l'ID de paquet RTFL.

RTFN PID

Ce paramètre spécifie l'ID de paquet RTFN.

Transmission type

Ce paramètre spécifie le type d'émission du PDO.

Time sync ID

Ce paramètre spécifie un ID de synchronisation de temps attribué au RxPDO dans le cas d'un traitement de RxPDO synchronisé.

Timeout

Ce paramètre spécifie un temps de temporisation pour le PDO.

Cycle multiplier

Ce paramètre spécifie le cycle d'émission prévu comme un multiplicateur du temps de cycle du système de communication dans le cas d'une communication intercellulaire.

Cycle offset

Ce paramètre spécifie un décalage pour le cycle d'émission en relation avec le cycle du système de communication dans le cas d'une communication intercellulaire.

Adresse de l'appareil

Ce paramètre spécifie l'adresse de dispositif du producteur du PDO pour la communication basée sur la voie de messages.

IPv4 address

Ce paramètre spécifie l'adresse IPv4 du producteur du PDO pour la communication intercellulaire basée sur la voie de messages.

IPv6 address

Ce paramètre spécifie l'adresse IPv6 du producteur du PDO pour la communication intercellulaire basée sur la voie de messages.

6.1.1.1.4.34 Receive PDO mapping parameter

L'objet "receive PDO mapping parameter" (paramètre de mise en correspondance d'objets de données de processus de réception) est constitué du paramètre suivant:

Paramètre

PDO number

Ce paramètre spécifie le numéro du PDO.

Number of mapping entries

Ce paramètre spécifie le nombre d'entrées de mise en correspondance.

List of mapping entries

Ce paramètre spécifie une liste d'entrées de mise en correspondance.

6.1.1.1.4.35 Transmit PDO communication parameter

L'objet "transmit PDO communication parameter" (paramètre de communication d'objets de données de processus d'émission) est constitué du paramètre suivant:

Paramètre

Number of entries

Ce paramètre spécifie le nombre d'entrées valides.

RTFL PID

Ce paramètre spécifie l'ID de paquet RTFL.

RTFN PID

Ce paramètre spécifie l'ID de paquet RTFN.

Transmission type

Ce paramètre spécifie le type d'émission du PDO.

Time sync ID

Ce paramètre spécifie un ID de synchronisation de temps attribué au TxPDO dans le cas d'un traitement de TxPDO synchronisé.

Cycle multiplier

Ce paramètre spécifie le cycle d'émission comme un multiple du temps de cycle du système de communication.

Cycle offset

Ce paramètre spécifie le cycle d'émission du PDO.

Device address

Ce paramètre spécifie l'adresse de dispositif du consommateur du PDO pour la communication basée sur la voie de messages.

IPv4 address

Ce paramètre spécifie l'adresse IPv4 du consommateur du PDO pour la communication intercellulaire basée sur la voie de messages.

IPv6 address

Ce paramètre spécifie l'adresse IPv6 du consommateur du PDO pour la communication intercellulaire basée sur la voie de messages.

6.1.1.1.4.36 Transmit PDO mapping parameter

L'objet "transmit PDO mapping parameter " (paramètre de mise en correspondance d'objets de données de processus d'émission) est constitué du paramètre suivant:

Paramètre

PDO number

Ce paramètre spécifie le numéro du PDO.

Number of mapping entries

Ce paramètre spécifie le nombre d'entrées de mise en correspondance.

List of mapping entries

Ce paramètre spécifie une liste d'entrées de mise en correspondance.

6.1.1.1.5 Interaction de SDO

6.1.1.1.5.1 Services de SDO

L'accès au dictionnaire d'objets est géré par les services d'objets de données de service (SDO). Les entrées d'objet dans les dictionnaires d'objets distants peuvent être lues ou écrites (téléchargement vers l'amont et téléchargement vers l'aval) en utilisant les services SDO appropriés. Les SDO sont des formats de données (structures) de toute taille qui sont adressables. L'adressage suit le format d'adressage de dictionnaire d'objets CANopen et consiste en un indice de 16 bits et un sous-indice de 8 bits, utilisé pour adresser les entrées appropriées du dictionnaire d'objets. L'énoncé d'indice et de sous-indice est également appelé "multiplexeur".

Afin de fournir un traitement simultané de services SDO dans un dispositif, le Type 22 introduit le paramètre JobID spécifié par le client. Ce JobID est attribué à chaque communication SDO individuelle. Le serveur utilise ce JobID dans les réponses. Un même JobID est valide pour toute la séquence de protocoles.

La communication SDO est basée sur le modèle client-serveur et est mise en œuvre à l'aide d'une liaison point à point entre deux dispositifs. Pour le transfert de plus gros volumes de données, le CeS de Type 22 n'exige pas un protocole de transport de SDO segmenté pour services SDO normalisés. La fonction de segmentation est fournie par la DLL de Type 22 qui transfère de gros volumes de données de n'importe quelle longueur à travers le réseau.

La communication de SDO est confirmée, la réception d'un message SDO est acquittée par un message SDO approprié. Un service d'abandon non confirmé enregistre toute erreur de communication de SDO. La communication de SDO débute par une phase d'initialisation préparant le client et le serveur pour l'émission. Dans le cas d'un petit volume de données,

les données sont émises directement dans la phase d'initialisation et la communication se termine par l'acquiescement consécutif du serveur. Ce mécanisme est appelé "transfert accéléré". Si de gros SDO sont transférés, aucune donnée n'est transférée pendant la phase d'initialisation. Ce mécanisme est appelé "transfert normal". Suivant l'acquiescement dans l'établissement de connexion, qui assure que les données peuvent être reçues, la phase de transfert réel de données avec les services de SDO appropriés commence.

Les primitives des services de SDO sont mises en correspondance avec les primitives des services de la voie de messages.

6.1.1.1.5.2 Séquence de téléchargement de SDO

Le service de téléchargement de SDO est utilisé pour écrire des données d'un client dans un serveur. Les services sont acquiescés par le serveur et retournent un statut de succès de demande. En cas d'erreur, ils retournent un message d'erreurs complémentaire contenant la cause de l'erreur. La cause de l'erreur est notifiée au client à l'aide du service d'abandon de SDO.

La Figure 5 montre une séquence de téléchargement aval accéléré de SDO qui a réussi entre le client et le serveur ainsi que les primitives entre client et serveur.

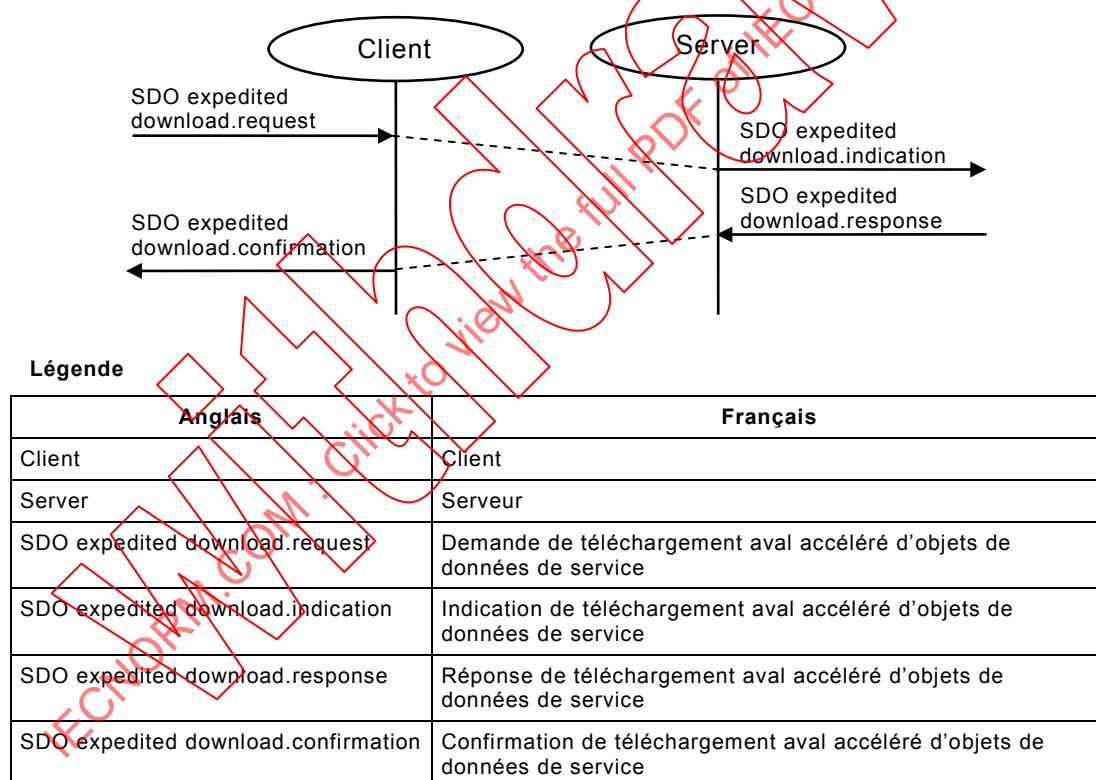


Figure 5 – Séquence de téléchargement aval accéléré de SDO qui a réussi

La Figure 6 montre une séquence d'initialisation de téléchargement aval normal de SDO qui a réussi ainsi que les primitives entre client et serveur. Ce transfert n'inclut pas de données et doit être poursuivi avec le service de téléchargement aval de SDO.

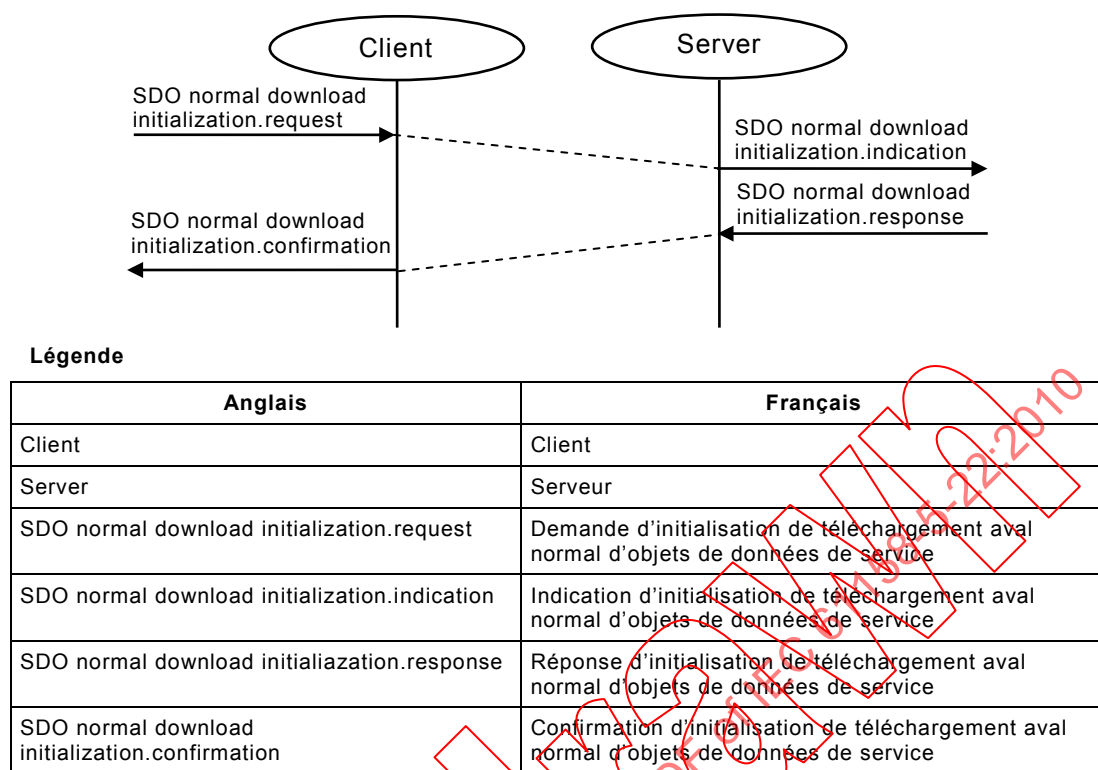


Figure 6 – Séquence d'initialisation de téléchargement aval normal de SDO qui a réussi

La Figure 7 montre une séquence de téléchargement aval de SDO qui a réussi ainsi que les primitives entre client et serveur. Cette séquence suit la séquence d'initialisation de téléchargement aval normal de SDO qui a réussi.

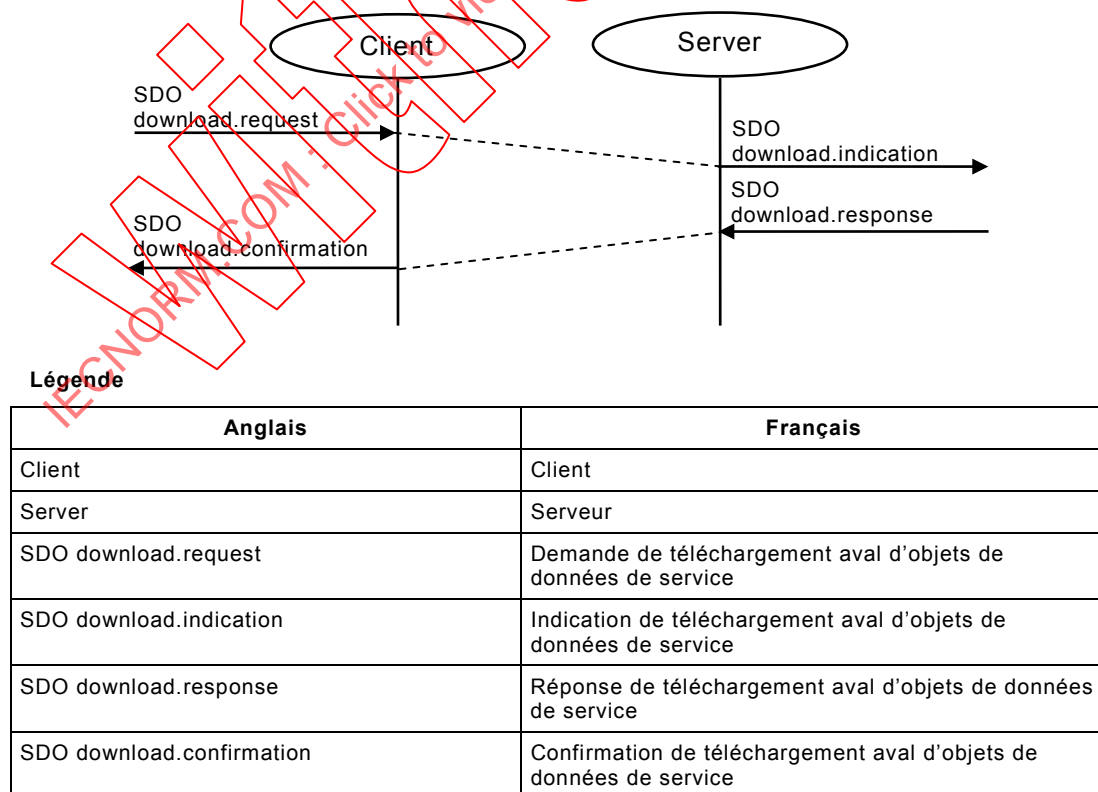


Figure 7 – Séquence de téléchargement aval de SDO qui a réussi

6.1.1.1.5.3 Séquence de téléchargement vers l'amont de SDO

Le client utilise le service de téléchargement amont de SDO pour lire des données dans le serveur. Les services sont acquittés et retournent un statut de succès de demande ainsi que les données demandées. En cas d'erreur, la cause en est retournée. Cette cause est notifiée au client à l'aide du service d'abandon de SDO.

La Figure 8 montre une séquence de téléchargement amont accéléré de SDO qui a réussi ainsi que les primitives entre client et serveur.

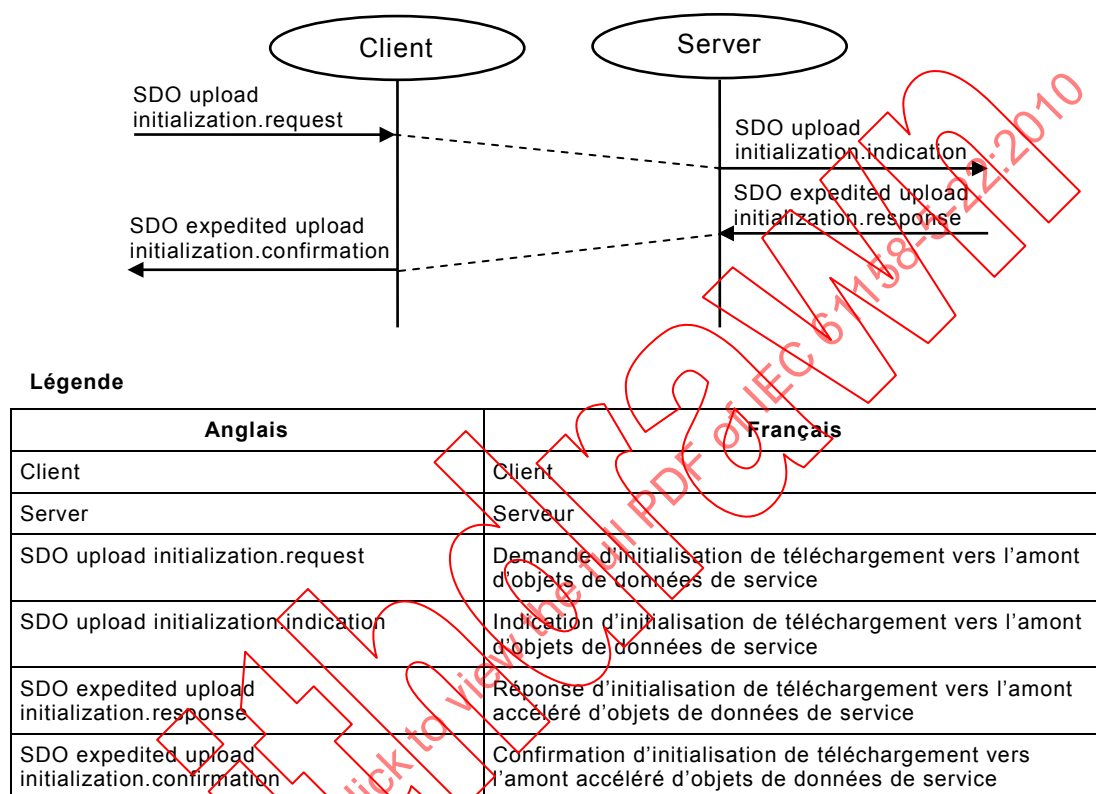
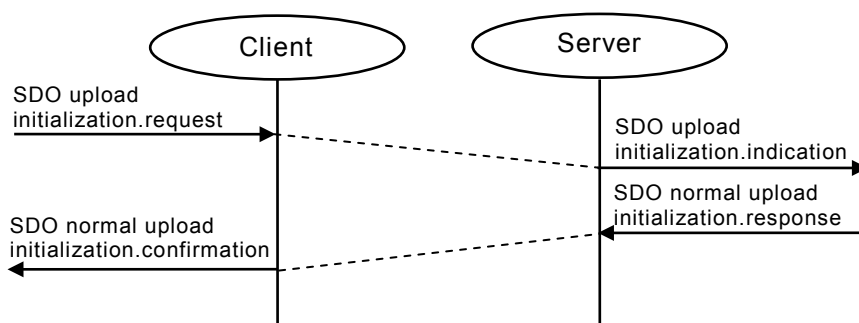


Figure 8 – Séquence de téléchargement amont accéléré de SDO qui a réussi

La Figure 9 montre une séquence d'initialisation de téléchargement amont normal de SDO qui a réussi ainsi que les primitives entre client et serveur. Ce transfert n'inclut pas de données et doit être poursuivi avec le service de téléchargement amont de SDO.

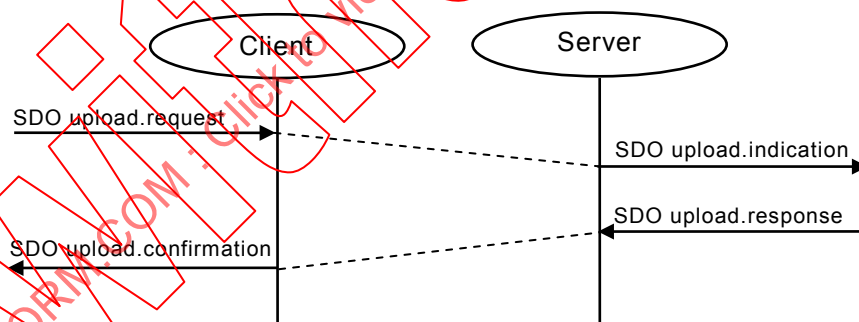


Légende

Anglais	Français
Client	Client
Server	Serveur
SDO upload initialization.request	Demande d'initialisation de téléchargement vers l'amont d'objets de données de service
SDO upload initialization.indication	Indication d'initialisation de téléchargement vers l'amont d'objets de données de service
SDO normal upload initialization.response	Réponse d'initialisation de téléchargement vers l'amont normal d'objets de données de service
SDO normal upload initialization.confirmation	Confirmation d'initialisation de téléchargement vers l'amont normal d'objets de données de service

Figure 9 – Séquence d'initialisation de téléchargement amont normal de SDO qui a réussi

La Figure 10 montre une séquence de téléchargement amont de SDO qui a réussi ainsi que les primitives entre client et serveur. Cette séquence suit une séquence d'initialisation qui a réussi.



Légende

Anglais	Français
Client	Client
Server	Serveur
SDO upload.request	Demande de téléchargement vers l'amont d'objets de données de service
SDO upload.indication	Indication de téléchargement vers l'amont d'objets de données de service
SDO upload.response	Réponse de téléchargement vers l'amont d'objets de données de service
SDO upload.confirmation	Confirmation de téléchargement vers l'amont d'objets de données de service

Figure 10 – Séquence de téléchargement amont de SDO qui a réussi

6.1.1.1.5.4 Abandon de SDO

Ce service abandonne la tentative de téléchargement aval ou amont de SDO par laquelle le client ou le serveur peut demander ce service. Le service d'abandon de SDO est non confirmé et contient en plus une description d'erreur codée.

La Figure 11 montre une séquence d'initialisation de téléchargement aval accéléré de SDO qui a échoué ainsi que les primitives entre client et serveur. Le serveur demande le service d'abandon de SDO. Les séquences d'abandon pour les séquences d'initialisation de téléchargement aval normal de SDO ou de téléchargement amont de SDO qui a échoué sont identiques.

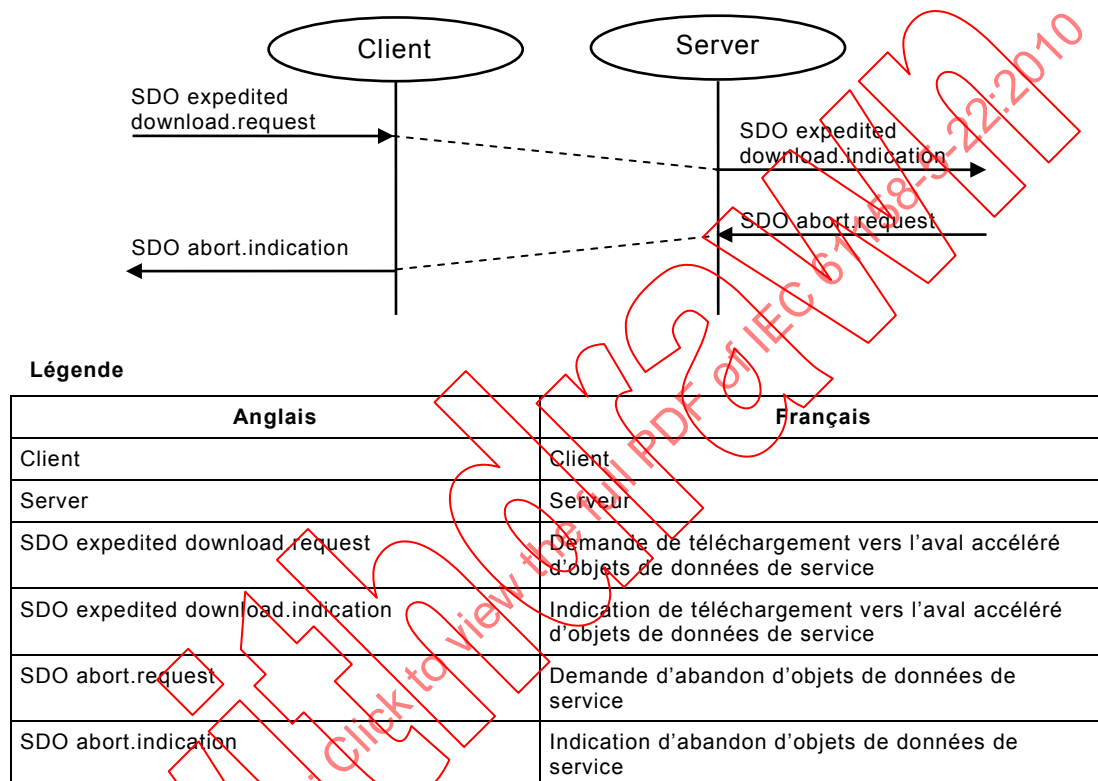


Figure 11 – Séquence d'initialisation de téléchargement aval accéléré de SDO qui a échoué

La Figure 12 montre un message d'erreur d'accès de téléchargement aval déclenché par le client vers le serveur après une séquence d'initialisation de téléchargement aval normal de SDO qui a réussi. La séquence de service d'abandon suivant un accès de téléchargement vers l'amont est identique.

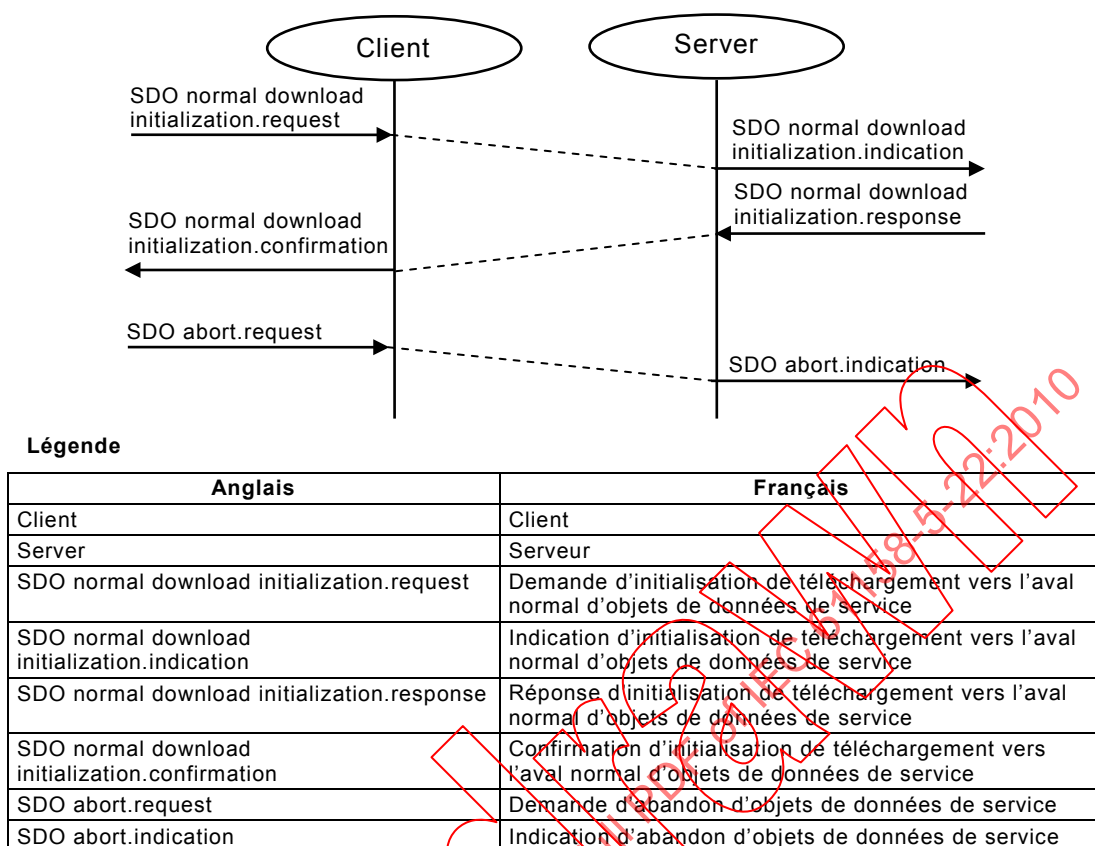


Figure 12 – Téléchargement aval de SDO qui a échoué après la séquence d'initialisation

La Figure 13 montre une séquence de téléchargement aval de SDO ayant échoué qui peut suivre une séquence d'initialisation de téléchargement aval normal de SDO qui a réussi. En raison d'une erreur, le serveur demande le service d'abandon de SDO après réception des données.

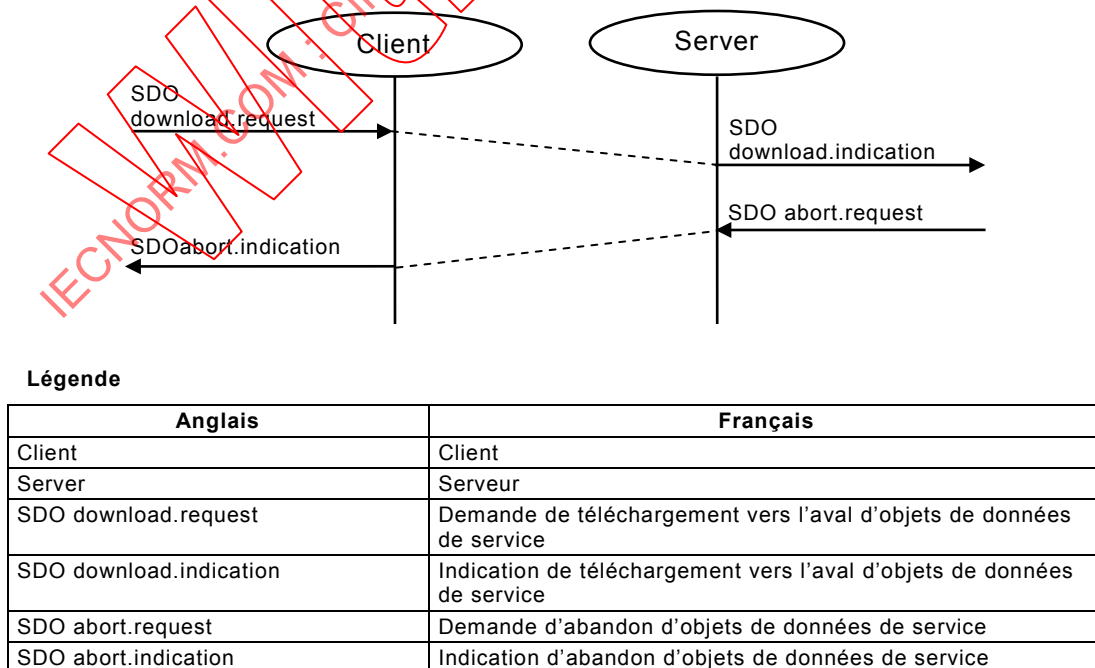


Figure 13 – Séquence de téléchargement aval de SDO qui a échoué

6.1.1.1.6 Urgence

Les messages d'urgence (EMCY) sont déclenchés par la survenue d'une situation d'erreur interne au dispositif. Les primitives des services d'urgence sont mises en correspondance avec les primitives des services de la voie de messages.

La Figure 14 montre les primitives entre producteur et consommateur.

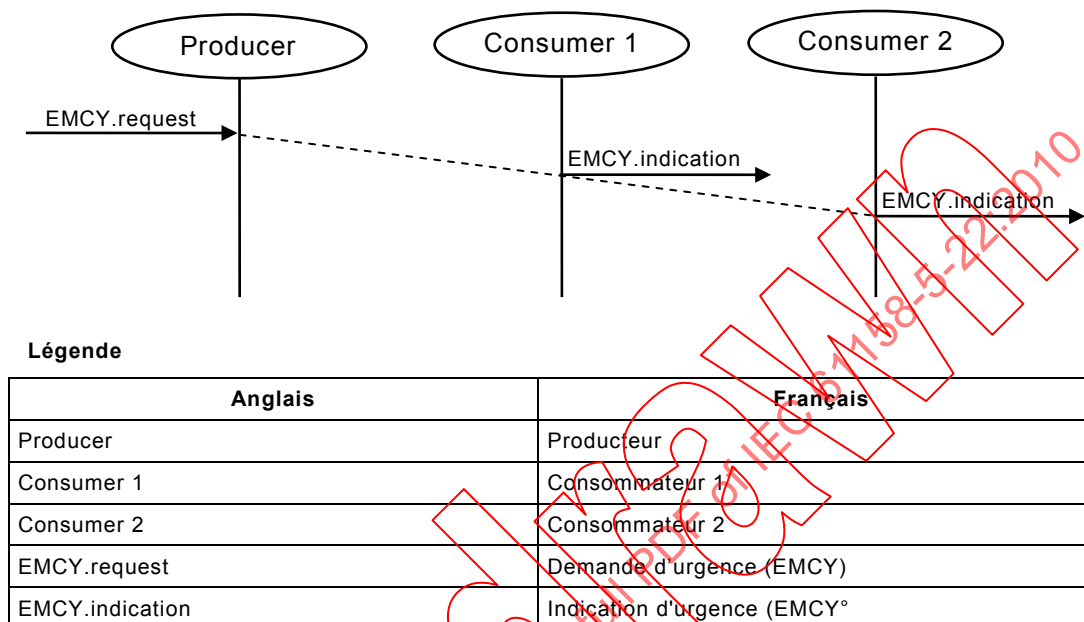
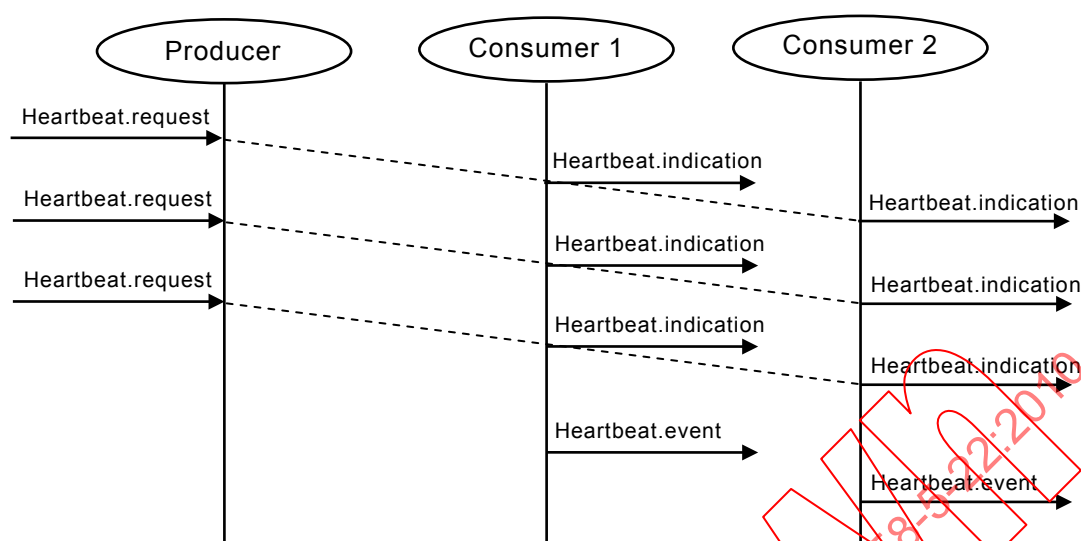


Figure 14 – Séquence d'urgence

6.1.1.1.7 Heartbeat

Les dispositifs peuvent se surveiller les uns les autres au moyen du mécanisme "heartbeat". Les producteurs de heartbeat génèrent des signes de vie qui sont reçus par les consommateurs de heartbeat. Un consommateur de heartbeat peut configurer dans le dictionnaire d'objets quels producteurs de heartbeat il veut surveiller. Si un heartbeat manque à arriver, un événement heartbeat est généré au niveau du consommateur de heartbeat.

Les primitives du service heartbeat peuvent être mises en correspondance avec les primitives de la CDC ou de la MSC. Cela est dépendant de la configuration. La Figure 15 illustre le principe fonctionnel du service heartbeat.



Légende

Anglais	Français
Producer	Producteur
Consumer 1	Consommateur 1
Consumer 2	Consommateur 2
Heartbeat.request	Demande de heartbeat (demande de battements de cœur)
Heartbeat.indication	Indication de heartbeat (indication de battements de cœur)
Heartbeat.event	Événement heartbeat (événement "battements de cœur")

Figure 15 – Séquence de heartbeat

6.1.1.1.8 Service de données de processus

6.1.1.1.8.1 Interaction de données de processus

Le service "process data write" (écriture de données de processus) est utilisé par un producteur de PDO pour envoyer aux consommateurs de PDO les données des objets d'application mis en correspondance. Le service d'écriture de données de processus indique au consommateur de PDO qu'un PDO valide a été reçu.

Les primitives du service de données de processus peuvent être mises en correspondance avec les primitives de services CDC ou MSC en fonction des paramètres de communication d'un PDO particulier. La Figure 16 montre la séquence du service d'écriture de données de processus.