

BS EN 61158-6-22:2014



BSI Standards Publication

Industrial communication networks — Fieldbus specifications

Part 6-22: Application layer protocol specification — Type 22 elements

bsi.

...making excellence a habit.
AVA Documents™
Document Sharing Hub

National foreword

This British Standard is the UK implementation of EN 61158-6-22:2014. It is identical to IEC 61158-6-22:2014. It supersedes BS EN 61158-6-22:2012 which is withdrawn.

The UK participation in its preparation was entrusted to Technical Committee AMT/7, Industrial communications: process measurement and control, including fieldbus.

A list of organizations represented on this committee can be obtained on request to its secretary.

This publication does not purport to include all the necessary provisions of a contract. Users are responsible for its correct application.

© The British Standards Institution 2014.
Published by BSI Standards Limited 2014

ISBN 978 0 580 79480 3
ICS 25.040.40; 35.100.70; 35.110

Compliance with a British Standard cannot confer immunity from legal obligations.

This British Standard was published under the authority of the Standards Policy and Strategy Committee on 30 November 2014.

Amendments issued since publication

Date	Text affected
------	---------------

EUROPEAN STANDARD
NORME EUROPÉENNE
EUROPÄISCHE NORM

EN 61158-6-22

October 2014

ICS 25.040.40; 35.100.70; 35.110

Supersedes EN 61158-6-22:2012

English Version

Industrial communication networks - Fieldbus specifications -
Part 6-22: Application layer protocol specification - Type 22
elements
(IEC 61158-6-22:2014)

Réseaux de communication industriels - Spécifications des
bus de terrain - Partie 6-22: Spécification du protocole de la
couche application - Éléments de type 22
(CEI 61158-6-22:2014)

Industrielle Kommunikationsnetze - Feldbusse - Teil 6-22:
Protokollspezifikation des Application Layer
(Anwendungsschicht) - Typ 22-Elemente
(IEC 61158-6-22:2014)

This European Standard was approved by CENELEC on 2014-09-23. CENELEC members are bound to comply with the CEN/CENELEC Internal Regulations which stipulate the conditions for giving this European Standard the status of a national standard without any alteration.

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

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

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



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

CEN-CENELEC Management Centre: Avenue Marnix 17, B-1000 Brussels

Foreword

The text of document 65C/764/FDIS, future edition 2 of IEC 61158-6-22, prepared by SC 65C "Industrial networks" of IEC/TC 65 "Industrial-process measurement, control and automation" was submitted to the IEC-CENELEC parallel vote and approved by CENELEC as EN 61158-6-22:2014.

The following dates are fixed:

- latest date by which the document has to be implemented at national level by publication of an identical national standard or by endorsement (dop) 2015-06-23
- latest date by which the national standards conflicting with the document have to be withdrawn (dow) 2017-09-23

This document supersedes EN 61158-6-22:2012.

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

This document has been prepared under a mandate given to CENELEC by the European Commission and the European Free Trade Association.

Endorsement notice

The text of the International Standard IEC 61158-6-22:2014 was approved by CENELEC as a European Standard without any modification.

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

IEC 61158-1	NOTE	Harmonized as EN 61158-1.
IEC 61784-1	NOTE	Harmonized as EN 61784-1.
IEC 61784-2	NOTE	Harmonized as EN 61784-2.

Annex ZA

(normative)

Normative references to international publications with their corresponding European publications

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

NOTE 1 When an International Publication has been modified by common modifications, indicated by (mod), the relevant EN/HD applies.

NOTE 2 Up-to-date information on the latest versions of the European Standards listed in this annex is available here: www.cenelec.eu.

<u>Publication</u>	<u>Year</u>	<u>Title</u>	<u>EN/HD</u>	<u>Year</u>
IEC 61158-3-22	-	Industrial communication networks - Fieldbus specifications - Part 3-22: Data-link layer service definition - Type 22 elements	EN 61158-3-22	-
IEC 61158-4-22	-	Industrial communication networks - Fieldbus specifications - Part 4-22: Data-link layer protocol specification - Type 22 elements	EN 61158-4-22	-
IEC 61158-5-22	-	Industrial communication networks - Fieldbus specifications - Part 5-22: Application layer service definition - Type 22 elements	EN 61158-5-22	-
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	-	-

<u>Publication</u>	<u>Year</u>	<u>Title</u>	<u>EN/HD</u>	<u>Year</u>
ISO/IEC 10731	-	Information technology - Open Systems Interconnection - Basic Reference Model - Conventions for the definition of OSI services	-	-
ISO/IEC/IEEE 60559	-	Information technology - Microprocessor Systems - Floating-Point arithmetic	-	-

CONTENTS

INTRODUCTION.....	8
1 Scope.....	9
1.1 General.....	9
1.2 Specifications.....	10
1.3 Conformance.....	10
2 Normative references	10
3 Terms, definitions, abbreviations, symbols and conventions	11
3.1 Terms and definitions from other ISO/IEC standards	11
3.2 Fieldbus application-layer specific definitions	11
3.3 Abbreviations and symbols.....	15
3.4 Conventions	17
4 Application layer protocol specification	18
4.1 Operating principle	18
4.2 Device reference models	19
4.3 Application layer structure	21
5 FAL syntax description	21
5.1 Introduction and coding principles	21
5.2 Data type encoding	21
5.3 CeS encoding.....	25
5.4 ISO/IEC 8802-3 DLPDU communication inside Type 22 RTFL	71
5.5 Management encoding	71
6 FAL protocol state machines	72
6.1 Overview	72
6.2 Fieldbus service protocol machine (FSPM).....	74
6.3 Application relationship protocol machine (ARPM).....	74
6.4 DLL mapping protocol machine	74
7 AP-context state machine.....	74
8 FAL service protocol machine (FSPM).....	74
9 Application layer state machine (ALSM)	75
9.1 Description	75
9.2 States	77
9.3 Primitive definitions	77
9.4 State table.....	78
9.5 AL-service forwarding depending on AL-state.....	79
10 DLL mapping protocol machine (DMPM).....	80
10.1 Overview	80
10.2 Primitives exchanged between ALSM and DMPM.....	80
10.3 Primitives exchanged between DLL and DMPM.....	84
10.4 ALSM to DLL mapping.....	86
Bibliography.....	87
Figure 1 – RTFL device reference model	20
Figure 2 – RTFN device reference model	21
Figure 3 – Encoding of TimeOfDay value	22

Figure 4 – Encoding of TimeDifference value	23
Figure 5 – Object dictionary addressing schema	26
Figure 6 – Relationships among protocol machines and adjacent layers	73
Figure 7 – ALSM protocol machine	74
Figure 8 – ALSM diagram	76
Table 1 – PDU element definition.....	18
Table 2 – Object definition	18
Table 3 – Transfer syntax for bit sequences.....	22
Table 4 – Transfer syntax for Integer data type	24
Table 5 – Transfer syntax for Unsigned data type	24
Table 6 – Object dictionary structure.....	25
Table 7 – Object dictionary object type definitions	26
Table 8 – Basic data type definitions.....	26
Table 9 – Complex data type definition	27
Table 10 – Communication section	28
Table 11 – Device type	30
Table 12 – Error register encoding.....	30
Table 13 – Error register	31
Table 14 – Object definition template.....	31
Table 15 – Encoding of event log entries	32
Table 16 – Event log	32
Table 17 – Manufacturer device name	33
Table 18 – Manufacturer HW version	33
Table 19 – Manufacturer SW version	33
Table 20 – CL configuration	34
Table 21 – Time sync IRQ configuration encoding	36
Table 22 – Time sync IRQ configuration	36
Table 23 – Time sync IRQ state	36
Table 24 – Store parameters read information	37
Table 25 – Store parameters.....	37
Table 26 – Restore parameters read information	39
Table 27 – Restore default parameters	39
Table 28 – Diagnostic information	40
Table 29 – Diagnostic threshold.....	43
Table 30 – IP address EMCY	45
Table 31 – Inhibit time EMCY.....	45
Table 32 – Encoding of consumer heartbeat entries.....	45
Table 33 – Consumer heartbeat list	46
Table 34 – Producer heartbeat parameter	47
Table 35 – Identity object.....	49
Table 36 – SDO protocol timeout	50
Table 37 – Enable client SDO parameter	50

Table 38 – Enable EMCY	51
Table 39 – PDO timeout tolerance	51
Table 40 – Store EDS	52
Table 41 – Storage format	52
Table 42 – OS command	52
Table 43 – OS command mode	53
Table 44 – OS debugger interface	54
Table 45 – OS prompt	55
Table 46 – Module list	56
Table 47 – Emergency subscriber encoding	57
Table 48 – Emergency subscriber	57
Table 49 – Client SDO parameter encoding	57
Table 50 – Client SDO parameter	58
Table 51 – Receive PDO communication parameter	58
Table 52 – Transmit PDO communication parameter	60
Table 53 – Mapping format	63
Table 54 – Receive PDO mapping parameter	63
Table 55 – Transmit PDO mapping parameter	64
Table 56 – Initiate SDO expedited download request	64
Table 57 – Initiate SDO expedited download response	65
Table 58 – Initiate SDO normal download request	65
Table 59 – Initiate SDO normal download response	65
Table 60 – SDO download request	65
Table 61 – SDO download response	66
Table 62 – Initiate SDO expedited upload request	66
Table 63 – Initiate SDO expedited upload response	66
Table 64 – Initiate SDO normal upload response	67
Table 65 – SDO upload request	67
Table 66 – SDO upload response	67
Table 67 – SDO abort request	67
Table 68 – SDO abort codes	68
Table 69 – Process data write request via MSC	69
Table 70 – Process data write request via CDC	69
Table 71 – Emergency request	69
Table 72 – Emergency error codes	69
Table 73 – Heartbeat request via MSC	70
Table 74 – Heartbeat request via CDC	71
Table 75 – Send frame request	71
Table 76 – Application layer management request	72
Table 77 – State transitions and management services	76
Table 78 – Primitives issued by ALSM to DLL	78
Table 79 – Primitives issued by DLL to ALSM	78
Table 80 – Primitives issued by FSPM to ALSM	78

Table 81 – Primitives issued by ALSM to FSPM	78
Table 82 – ALSM state table	79
Table 83 – Application layer states and communication services.....	79
Table 84 – Primitives issued by ALSM to DMPM	80
Table 85 – Primitives issued by DMPM to ALSM	82
Table 86 – Primitives issued by DMPM to DLL	84
Table 87 – Primitives issued by DLL to DMPM	85
Table 88 – ALSM to DLL mapping.....	86

INTRODUCTION

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

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

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

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

INDUSTRIAL COMMUNICATION NETWORKS – FIELDBUS SPECIFICATIONS –

Part 6-22: Application layer protocol specification – Type 22 elements

1 Scope

1.1 General

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 behavior provided by the different Types of the fieldbus Application Layer in terms of:

- a) the abstract syntax defining the application layer protocol data units conveyed between communicating application entities;
- b) the transfer syntax defining the application layer protocol data units conveyed between communicating application entities;
- c) the application context state machine defining the application service behavior visible between communicating application entities; and
- d) the application relationship state machines defining the communication behavior visible between communicating application entities.

The purpose of this standard is to define the protocol provided to:

- a) define the wire-representation of the service primitives defined in IEC 61158-5-22; and
- b) define the externally visible behavior associated with their transfer.

This standard specifies the protocol of the IEC fieldbus Application Layer, in conformance with the OSI Basic Reference Model (ISO/IEC 7498-1) 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 syntax and behavior of the application layer protocol that conveys the application layer services defined in IEC 61158-5-22.

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 protocols standardized in subparts of IEC 61158-6.

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 the application layer service definition standard. Instead, conformance is achieved through implementation of this application layer protocol specification.

2 Normative references

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

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

IEC 61158-3-22, *Industrial communication networks – Fieldbus specifications – Part 3-22: Data-link layer service definition – Type 22 elements*

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

IEC 61158-5-22, *Industrial communication networks – Fieldbus specifications – Part 5-22: Application layer service definition – Type 22 elements*

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

ISO/IEC/IEEE 60559, *Information technology – Microprocessor Systems – Floating-Point arithmetic*

3 Terms, definitions, abbreviations, symbols and conventions

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

3.1 Terms and definitions from other ISO/IEC standards

3.1.1 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

3.1.2 ISO/IEC 8822 terms

- a) abstract syntax
- b) presentation context

3.1.3 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

3.1.4 ISO/IEC 8824-1 terms

- a) object identifier
- b) type

3.2 Fieldbus application-layer specific definitions

3.2.1

acyclic data

data which is transferred from time to time for dedicated purposes

3.2.2**bit**

unit of information consisting of a 1 or a 0

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

3.2.3**cell**

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

3.2.4**channel**

path provided for conveying data

3.2.5**client**

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

3.2.6**communication cycle**

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

3.2.7**connection**

logical binding between two application objects

3.2.8**cycle time**

duration of a communication cycle

3.2.9**cyclic**

events which repeat in a regular and repetitive manner

3.2.10**cyclic communication**

periodic exchange of telegrams

3.2.11**cyclic data**

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

3.2.12**cyclic data channel****CDC**

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

3.2.13**data**

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

3.2.14**device**

physical entity connected to the fieldbus

3.2.15**error**

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

3.2.16**error code**

identification number of a specific type of error

3.2.17**gateway**

device acting as a linking element between different protocols

3.2.18**index**

position of an object within the object dictionary

3.2.19**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

3.2.20**interface**

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

3.2.21**intra-cell communication**

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

3.2.22**logical double line**

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

3.2.23**mapping parameters**

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

3.2.24**master clock**

global time base for the PCS mechanism

3.2.25**message**

ordered sequence of octets intended to convey data

3.2.26**message channel****MSC**

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

3.2.27**network**

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

3.2.28**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

3.2.29**precise clock synchronization****PCS**

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

3.2.30**process data**

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

3.2.31**process data object**

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

3.2.32**protocol**

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

3.2.33**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

3.2.34**real time frame line****RTFL**

communication model communicating in a logical double line

3.2.35**real time frame network****RTFN**

communication model communicating in a switched network

3.2.36**round trip time**

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

3.2.37**sub-index**

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

3.2.38**timing signal**

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

3.2.39**topology**

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

3.3 Abbreviations and symbols

AE	Application entity
AL	Application layer
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
CDC	Cyclic data channel
CL	Communication layer
Cnf	Confirmation
DA	Device address or destination 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
FCS	Frame check sequence
ID	Identification
Ind	Indication
IP	Internet protocol
IPv4	IP version 4

IPv6	IP version 6
IRQ	Interrupt request
LME	Layer management entity
MAC	Medium access control
MII	Media independent interface
MSC-MTP	Message channel message transfer protocol
MSC	Message channel
OD	Ordinary device
OS	Operating system
OSI	Open systems interconnection
PCS	Precise clock synchronization
PDO	Process data object
PID	Packet ID
PTPMSU	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
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
SA	Source address
SDO	Service data object
SEF	Standard ISO/IEC 8802-3 Ethernet 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

3.4 Conventions

3.4.1 General concept

The FAL is defined as a set of object-oriented ASEs. Each ASE is specified in a separate clause. Each ASE specification is composed of three parts: its class definitions, its services, and its protocol specification. The first two are contained in IEC 61158-5-22. The protocol specification for each of the ASEs is defined in this standard.

The class definitions define the attributes of the classes supported by each ASE. The attributes are accessible from instances of the class using the Management ASE services specified in IEC 61158-5-22. The service specification defines the services that are provided by the ASE.

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

3.4.2 Abstract syntax conventions

The AL syntax elements related to PDU structure are described as shown in Table 1.

- Frame part denotes the element that will be replaced by this reproduction.
- Data field is the name of the elements.
- Data type denotes the type of the terminal symbol.
- Value/description contains the constant value or the meaning of the parameter.

Table 1 – PDU element definition

Frame part	Data field	Data type	Value/description

The attributes of an object of the object dictionary are described in a form as shown in Table 2.

- Index describes the position within the object dictionary of an object.
- Sub-index describes a single element of the object.
- Name denotes a name string for this attribute.
- Object type denotes the characterizing type for each object as specified in Table 7.
- Data type denotes the data type of this element.
- Category indicates whether the element is mandatory (M), optional (O) or depends upon setting of other attributes (C).
- Access attribute shows the access right to this element. RO means read access right, RW means read and write access right, while WO means write access right.
- PDO mapping denotes the possibility to map this attribute to TxPDO or RxPDO or to indicate that this parameter is not mappable.
- Value range contains the value range of a dedicated element or 'No' for no pre-defined value range.
- Value contains the constant value(s) and/or the meaning of the parameter or 'No' for no pre-defined value.

Table 2 – Object definition

Attribute	Value
Index	
Sub-index	
Name	
Object type	
Data type	
Category	
Access attribute	
PDO mapping	
Value range	
Value	

4 Application layer protocol specification

4.1 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 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 cell 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 Device reference models

4.2.1 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 1.

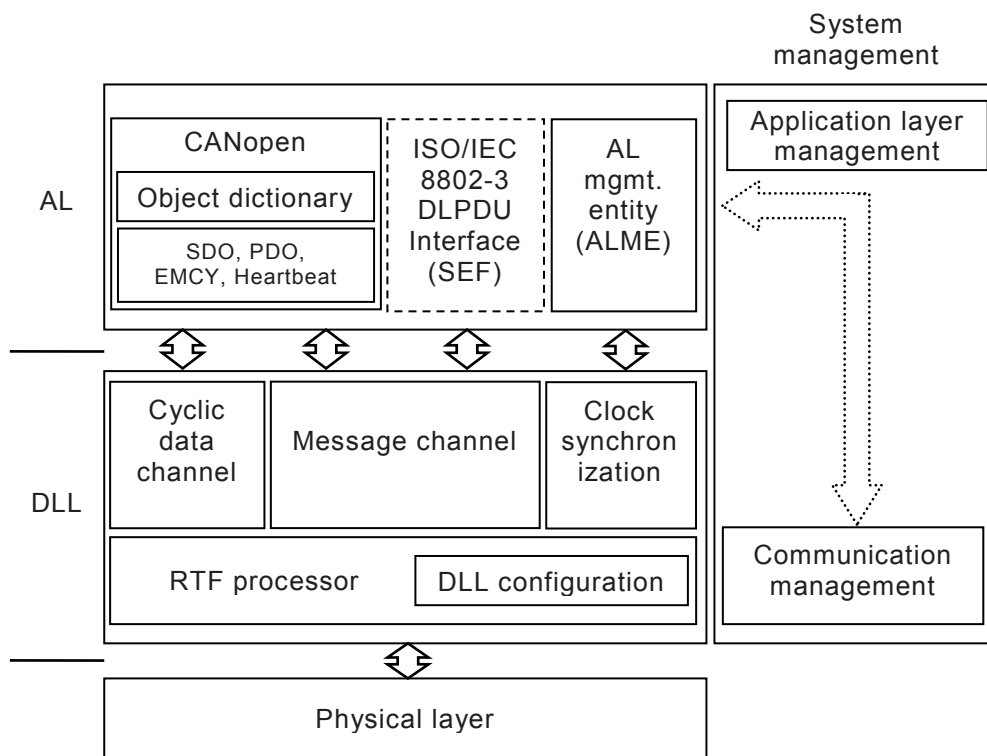


Figure 1 – RTFL device reference model

4.2.2 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 2.

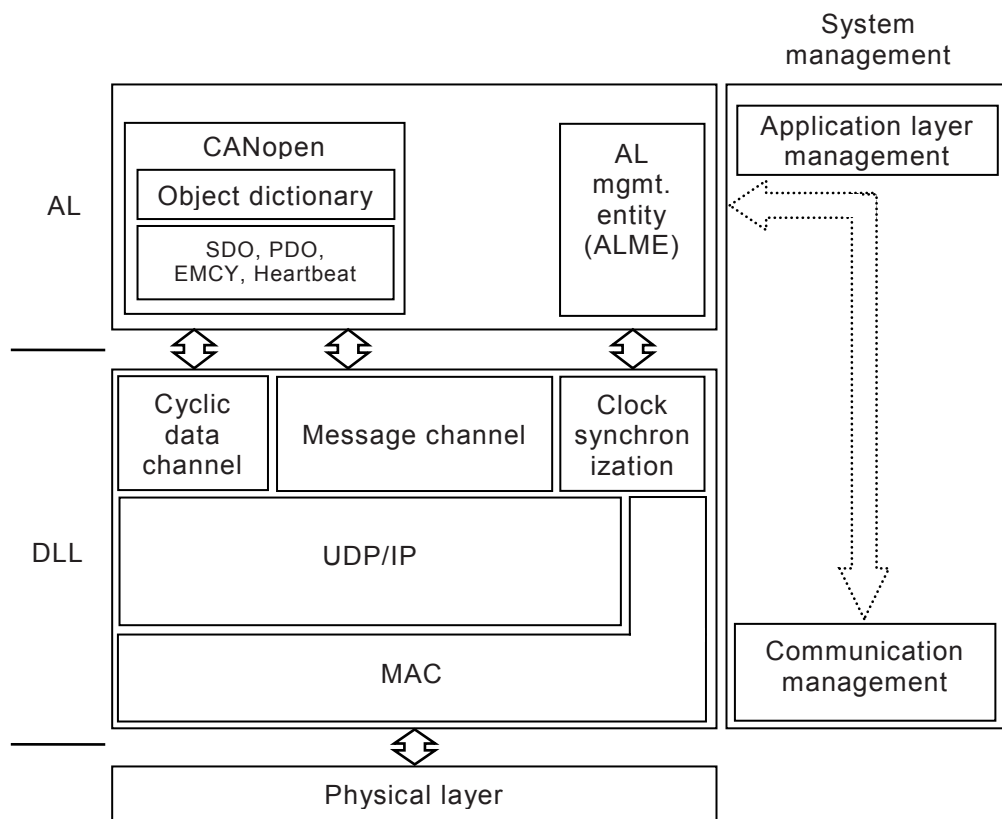


Figure 2 – RTFN device reference model

4.3 Application layer structure

The application layer consists of the following elements.

- A mandatory unit for real time data processing (CeS).
- An optional entity to enable Standard ISO/IEC 8802-3 Ethernet DLPDU communication.
- A mandatory management entity for system management.

The Application Layer uses the services provided by the Type 22 DLL to convey the AL-service data.

5 FAL syntax description

5.1 Introduction and coding principles

The abstract syntax and the transfer syntax are merged into a fixed format that is defined in the following clauses.

5.2 Data type encoding

5.2.1 Overview

To be able to exchange meaningful data across a Type 22 network, the format of this data and its meaning have to be known by communicating entities. This specification models this by the concept of data types.

The encoding rules define the representation of values of data types and the transfer syntax for the representation. Values are represented as bit sequences. Bit sequences are

transferred in sequences of octets. For numerical data types the encoding is little endian style.

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

5.2.2 Transfer syntax for bit sequences

For transmission across Type 22 AL a bit sequence is reordered into a sequence of octets. Let $b = b_{n-1} \dots b_0$ be a bit sequence. Denote k a non-negative integer such that $8(k-1) < n < 8k$. Then b is transferred in k octets assembled as shown in Table 3. The bits b_i , $i > n$ of the highest numbered octet are do not care bits.

Octet 1 is transmitted first and octet k is transmitted last. Hence the bit sequence is transferred as follows across the network:

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

Table 3 – Transfer syntax for bit sequences

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

5.2.3 Encoding of a Boolean value

- The encoding of a Boolean value shall be primitive. The Boolean value shall consist of a single octet.
- If the Boolean value is FALSE, the single octet shall be 0 (zero). If the Boolean value is TRUE, the single octet shall be 0xff.

5.2.4 TimeOfDay

- The encoding of a TimeOfDay with and without date indication value shall be primitive.
- The octets shall be equal in value to the octets in the data value, as shown in Figure 3.

Bits	7	6	5	4	3	2	1	0	Meaning
Octets									Number of milliseconds since midnight
1	0	0	0	0	2 ²⁷	2 ²⁶	2 ²⁵	2 ²⁴	
2	2 ²³	2 ²²	2 ²¹	2 ²⁰	2 ¹⁹	2 ¹⁸	2 ¹⁷	2 ¹⁶	
3	2 ¹⁵	2 ¹⁴	2 ¹³	2 ¹²	2 ¹¹	2 ¹⁰	2 ⁹	2 ⁸	
4	2 ⁷	2 ⁶	2 ⁵	2 ⁴	2 ³	2 ²	2 ¹	2 ⁰	
5	2 ¹⁵	2 ¹⁴	2 ¹³	2 ¹²	2 ¹¹	2 ¹⁰	2 ⁹	2 ⁸	Number of days since 1984-01-01 only with date indication
6	2 ⁷	2 ⁶	2 ⁵	2 ⁴	2 ³	2 ²	2 ¹	2 ⁰	
msb									

Figure 3 – Encoding of TimeOfDay value

5.2.5 TimeDifference

- The encoding of a Time Difference with and without date indication value shall be primitive.

- b) The content octets shall be equal in value to the octets in the data value, as shown in Figure 4.

Bits	7	6	5	4	3	2	1	0	Meaning
Octets									milliseconds
1									
2	2 ³¹	2 ³⁰	2 ²⁹	2 ²⁸	2 ²⁷	2 ²⁶	2 ²⁵	2 ²⁴	
3	2 ²³	2 ²²	2 ²¹	2 ²⁰	2 ¹⁹	2 ¹⁸	2 ¹⁷	2 ¹⁶	
4	2 ¹⁵	2 ¹⁴	2 ¹³	2 ¹²	2 ¹¹	2 ¹⁰	2 ⁹	2 ⁸	
5	2 ⁷	2 ⁶	2 ⁵	2 ⁴	2 ³	2 ²	2 ¹	2 ⁰	
6	2 ¹⁵	2 ¹⁴	2 ¹³	2 ¹²	2 ¹¹	2 ¹⁰	2 ⁹	2 ⁸	days only with date indication
7	2 ⁷	2 ⁶	2 ⁵	2 ⁴	2 ³	2 ²	2 ¹	2 ⁰	
msb									

Figure 4 – Encoding of TimeDifference value

5.2.6 Encoding of floating point values

These encoding rules are applied for data types float, Float32, double and Float64 as specified in IEC 61158-5-22.

- a) The encoding of a fixed-length Floating-Point value of Floating32 and Floating64 types shall be primitive, and the content octets shall consist of exactly four or eight octets, respectively.
- b) The content octets shall contain floating-point values defined in conformance with ISO/IEC/IEEE 60559. The sign is encoded in bit 7 of the first octet. It is followed by the exponent starting from bit 6 of the first octet, and then the mantissa starting from bit 6 of the second octet for Floating32 or Floating64.

5.2.7 Encoding of Integer values

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

$$b = b_0 \text{ to } b_{n-1}$$

is assigned the value

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

and, performing two's complement arithmetic,

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

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

EXAMPLE The value $-257 = 0xFEFF$ with data type Integer16 is transferred in two octets, first $0xFF$ and then $0xFE$.

The Integer data types are transferred as specified in Table 4.

Table 4 – Transfer syntax for Integer data type

Octet number	1.	2.	3.	4.	5.	6.	7.	8.
Integer8	$b_7 - b_0$	—	—	—	—	—	—	—
Integer16	$b_7 - b_0$	$b_{15} - b_8$	—	—	—	—	—	—
Integer32	$b_7 - b_0$	$b_{15} - b_8$	$b_{23} - b_{16}$	$b_{31} - b_{24}$	—	—	—	—
Integer64	$b_7 - b_0$	$b_{15} - b_8$	$b_{23} - b_{16}$	$b_{31} - b_{24}$	$b_{39} - b_{32}$	$b_{47} - b_{40}$	$b_{55} - b_{48}$	$b_{63} - b_{56}$

NOTE 1 Data types SINT and char as defined in IEC 61158-5-22 corresponds to Integer8.

NOTE 2 Data types INT and short as defined in IEC 61158-5-22 corresponds to Integer16.

NOTE 3 Data types DINT and long as defined in IEC 61158-5-22 corresponds to Integer32.

NOTE 4 Data type LINT as defined in IEC 61158-5-22 corresponds to Integer64.

5.2.8 Encoding of Unsigned Integer values

Data of basic data type Unsigned Integer has values in the non-negative integers. The value range is 0 to 2^{n-1} . The data is represented as bit sequences of length n. The bit sequence

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

is assigned the value

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

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

EXAMPLE The value 286 = 0x11E with data type Unsigned16 is transferred in two octets, first 0x1E and then 0x01.

The unsigned data types are transferred as specified in Table 5.

Table 5 – Transfer syntax for Unsigned data type

Octet number	1.	2.	...	4.	...	8.	...	16.	...	32.
Unsigned8	$b_7 - b_0$	—	—	—	—	—	—	—	—	—
Unsigned 16	$b_7 - b_0$	$b_{15} - b_8$	—	—	—	—	—	—	—	—
Unsigned32	$b_7 - b_0$	$b_{15} - b_8$	—	$b_{31} - b_{24}$	—	—	—	—	—	—
Unsigned64	$b_7 - b_0$	$b_{15} - b_8$	—	$b_{31} - b_{24}$	—	$b_{63} - b_{56}$	—	—	—	—
Unsigned128	$b_7 - b_0$	$b_{15} - b_8$	—	$b_{31} - b_{24}$	—	$b_{63} - b_{56}$	—	$b_{127} - b_{120}$	—	—
Unsigned256	$b_7 - b_0$	$b_{15} - b_8$	—	$b_{31} - b_{24}$	—	$b_{63} - b_{56}$	—	$b_{127} - b_{120}$	—	$b_{255} - b_{248}$

NOTE 1 Data types USINT and unsigned char as defined in IEC 61158-5-22 corresponds to Unsigned8.

NOTE 2 Data types UINT and WORD as defined in IEC 61158-5-22 corresponds to Unsigned16.

NOTE 3 Data types UDINT and DWORD as defined in IEC 61158-5-22 corresponds to Unsigned32.

NOTE 4 Data type ULINT as defined in IEC 61158-5-22 corresponds to Unsigned64.

5.2.9 Encoding of an OctetString value

a) The encoding of a variable length OctetString value shall be primitive.

- b) There is no length field; the length is encoded implicitly.
- c) The content octets shall be a sequence of octets. The leftmost string element is encoded in the first octet, followed by second octet, followed by each octet in turn up to and including the last octet as rightmost of the content octets.

5.2.10 Encoding of a VisibleString value

- a) The encoding of a variable length VisibleString value shall be primitive.
- b) There is no length field and no termination symbol; the length is encoded implicitly.
- c) The content octets shall be a sequence of octets. The leftmost string element is encoded in the first octet, followed by the second octet, followed by each octet in turn up to and including the last octet as rightmost of the content octets.

5.2.11 Encoding of an UnicodeString value

- a) The encoding of a variable length UnicodeString value shall be primitive.
- b) There is no length field; the length is encoded implicitly.
- c) The content octets shall be a sequence of unsigned integer. The leftmost string element is encoded in the first unsigned integer, followed by the second unsigned integer, followed by each unsigned integer in turn up to and including the last unsigned integer as rightmost of the content octets.

5.3 CeS encoding

5.3.1 Object dictionary

5.3.1.1 Object dictionary structure

The dictionary is structured in form of a table as indicated in Table 6.

Table 6 – Object dictionary structure

Index	Section	Sub-section	Content
0x0001 to 0x001F	Data type	Basic data types	Definition of basic data types
0x0020 to 0x003F	—	Complex data types	Definition of complex data types
0x0040 to 0x005F	—	Manufacturer specific data types	Definition of manufacturer specific data types
0x0060 to 0x007F	—	Device profile specific basic data types	Definition of device profile specific basic data types
0x0080 to 0x009F	—	Device profile specific complex data types	Definition of device profile specific complex data types
0x00A0 to 0x0FFF	Reserved	—	—
0x1000 to 0x1FFF	Communication profile	—	Definition of the parameters which are used for communication configuration and dedicated communication purposes
0x2000 to 0x5FFF	Manufacturer defined profile	—	Definition of manufacturer specific parameters
0x6000 to 0x9FFF	Standardized device profile	—	Definition of the parameters defined in a standardized device profile
0xA000 to 0xBFFF	Standardized interface profile	—	Definition of the parameters defined in standardized interface profile
0xC000 to 0xC8FF	Type 22 RTFN interface profile	—	Definition of the parameters defined in Type 22 RTFN interface profile
0xC900 to 0xFFFF	Reserved	—	—

A logical addressing scheme is used for object dictionary access. The index field describes the position within the object dictionary. Each index is further described by an 8 bit sub-index.