

# INTERNATIONAL STANDARD

## NORME INTERNATIONALE

**Industrial communication networks – Fieldbus specifications –  
Part 5-19: Application layer service definition – Type 19 elements**

**Réseaux de communication industriels – Spécifications des bus de terrain –  
Partie 5-19: Définition des services de la couche application – Éléments de  
Type 19**



## THIS PUBLICATION IS COPYRIGHT PROTECTED

Copyright © 2007 IEC, Geneva, Switzerland

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

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

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

Tel.: +41 22 919 02 11  
Fax: +41 22 919 03 00  
[info@iec.ch](mailto:info@iec.ch)  
[www.iec.ch](http://www.iec.ch)

### About the IEC

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

### About IEC publications

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

#### IEC Catalogue - [webstore.iec.ch/catalogue](http://webstore.iec.ch/catalogue)

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

#### IEC publications search - [www.iec.ch/searchpub](http://www.iec.ch/searchpub)

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

#### IEC Just Published - [webstore.iec.ch/justpublished](http://webstore.iec.ch/justpublished)

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

#### Electropedia - [www.electropedia.org](http://www.electropedia.org)

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

#### IEC Glossary - [std.iec.ch/glossary](http://std.iec.ch/glossary)

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

#### IEC Customer Service Centre - [webstore.iec.ch/csc](http://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](mailto:csc@iec.ch).

### A propos de l'IEC

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

### A propos des publications IEC

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

#### Catalogue IEC - [webstore.iec.ch/catalogue](http://webstore.iec.ch/catalogue)

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

#### Recherche de publications IEC - [www.iec.ch/searchpub](http://www.iec.ch/searchpub)

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

#### IEC Just Published - [webstore.iec.ch/justpublished](http://webstore.iec.ch/justpublished)

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

#### Electropedia - [www.electropedia.org](http://www.electropedia.org)

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

#### Glossaire IEC - [std.iec.ch/glossary](http://std.iec.ch/glossary)

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

#### Service Clients - [webstore.iec.ch/csc](http://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](mailto:csc@iec.ch).

# INTERNATIONAL STANDARD

## NORME INTERNATIONALE

**Industrial communication networks – Fieldbus specifications –  
Part 5-19: Application layer service definition – Type 19 elements**

**Réseaux de communication industriels – Spécifications des bus de terrain –  
Partie 5-19: Définition des services de la couche application – Éléments de  
Type 19**

INTERNATIONAL  
ELECTROTECHNICAL  
COMMISSION

COMMISSION  
ELECTROTECHNIQUE  
INTERNATIONALE

PRICE CODE  
CODE PRIX



ICS 25.040.40; 35.100.70

ISBN 978-2-8322-1607-1

**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 terms .....                                       | 9  |
| 3.5 Fieldbus application-layer specific definitions .....          | 9  |
| 3.6 Abbreviations and symbols.....                                 | 11 |
| 3.7 Conventions .....                                              | 11 |
| 4 Concepts.....                                                    | 14 |
| 5 Data type ASE.....                                               | 14 |
| 5.1 Bitstring types .....                                          | 14 |
| 5.2 Unsigned types .....                                           | 15 |
| 5.3 Integer types .....                                            | 16 |
| 5.4 Floating Point types.....                                      | 16 |
| 5.5 Structure types.....                                           | 17 |
| 6 Communication model specification.....                           | 17 |
| 6.1 Concepts.....                                                  | 17 |
| 6.2 ASEs.....                                                      | 17 |
| 6.3 ARs.....                                                       | 30 |
| 6.4 Summary of AR classes .....                                    | 31 |
| 6.5 Permitted FAL services by AREP role.....                       | 32 |
| Bibliography.....                                                  | 33 |
| Table 1 – Read service parameters.....                             | 19 |
| Table 2 – Write service parameters.....                            | 20 |
| Table 3 – Read service parameters.....                             | 21 |
| Table 4 – Write service parameters.....                            | 22 |
| Table 5 – Notify service parameters.....                           | 23 |
| Table 6 – Get network status service parameters.....               | 24 |
| Table 7 – Get device status service parameters.....                | 24 |
| Table 8 – Network status change report service parameters .....    | 25 |
| Table 9 – Station status change report service parameters .....    | 26 |
| Table 10 – Set device status service parameters .....              | 26 |
| Table 11 – Enable RTC service parameters .....                     | 27 |
| Table 12 – Enable hotplug service parameters.....                  | 28 |
| Table 13 – Notify RTC service parameters.....                      | 29 |
| Table 14 – Disable RTC service parameters .....                    | 29 |

|                                              |    |
|----------------------------------------------|----|
| Table 15 – AREP (SVC) class summary .....    | 31 |
| Table 16 – AREP (RTC-MS) class summary ..... | 32 |
| Table 17 – AREP (RTC-CC) class summary ..... | 32 |
| Table 18 – FAL services by AR type .....     | 32 |

IECNORM.COM : Click to view the full PDF of IEC 61158-5-19:2007

Withdrawing

## INTERNATIONAL ELECTROTECHNICAL COMMISSION

**INDUSTRIAL COMMUNICATION NETWORKS –  
FIELDBUS SPECIFICATIONS –****Part 5-19: Application layer service definition – Type 19 elements**

## FOREWORD

- 1) The International Electrotechnical Commission (IEC) is a worldwide organization for standardization comprising all national electrotechnical committees (IEC National Committees). The object of IEC is to promote international co-operation on all questions concerning standardization in the electrical and electronic fields. To this end and in addition to other activities, IEC publishes International Standards, Technical Specifications, Technical Reports, Publicly Available Specifications (PAS) and Guides (hereafter referred to as "IEC Publication(s)"). Their preparation is entrusted to technical committees; any IEC National Committee interested in the subject dealt with may participate in this preparatory work. International, governmental and non-governmental organizations liaising with the IEC also participate in this preparation. IEC collaborates closely with the International Organization for Standardization (ISO) in accordance with conditions determined by agreement between the two organizations.
- 2) The formal decisions or agreements of IEC on technical matters express, as nearly as possible, an international consensus of opinion on the relevant subjects since each technical committee has representation from all interested IEC National Committees.
- 3) IEC Publications have the form of recommendations for international use and are accepted by IEC National Committees in that sense. While all reasonable efforts are made to ensure that the technical content of IEC Publications is accurate, IEC cannot be held responsible for the way in which they are used or for any misinterpretation by any end user.
- 4) In order to promote international uniformity, IEC National Committees undertake to apply IEC Publications transparently to the maximum extent possible in their national and regional publications. Any divergence between any IEC Publication and the corresponding national or regional publication shall be clearly indicated in the latter.
- 5) IEC provides no marking procedure to indicate its approval and cannot be rendered responsible for any equipment declared to be in conformity with an IEC Publication.
- 6) All users should ensure that they have the latest edition of this publication.
- 7) No liability shall attach to IEC or its directors, employees, servants or agents including individual experts and members of its technical committees and IEC National Committees for any personal injury, property damage or other damage of any nature whatsoever, whether direct or indirect, or for costs (including legal fees) and expenses arising out of the publication, use of, or reliance upon, this IEC Publication or any other IEC Publications.
- 8) Attention is drawn to the Normative references cited in this publication. Use of the referenced publications is indispensable for the correct application of this publication.
- 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** Use of some of the associated protocol types is restricted by their intellectual-property-right holders. In all cases, the commitment to limited release of intellectual-property-rights made by the holders of those rights permits a particular data-link layer protocol type to be used with physical layer and application layer protocols in type combinations as specified explicitly in the IEC 61784 series. Use of the various protocol types in other combinations may require permission of their respective intellectual-property-right holders.

International Standard IEC 61158-5-19 has been prepared by subcommittee 65C: Industrial networks, of IEC technical committee 65: Industrial-process measurement, control and automation.

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

This first edition and its companion parts of the IEC 61158-5 subseries cancel and replace IEC 61158-5:2003. This edition of this part constitutes a technical addition. This part and its Type 19 companion parts also cancel and replace IEC/PAS 62410, published in 2005.

This edition of IEC 61158-5 includes the following significant changes from the prior edition:

- a) deletion of the former Type 6 fieldbus for lack of market relevance;
- b) addition of new types of fieldbuses;
- c) partition of part 5 of the third edition into multiple parts numbered -5-2, -5-3, ...

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

| FDIS         | Report on voting |
|--------------|------------------|
| 65C/475/FDIS | 65C/486/RVD      |

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

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

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

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

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

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

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

## 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 61158-5-19:2007

## INDUSTRIAL COMMUNICATION NETWORKS – FIELDBUS SPECIFICATIONS –

### Part 5-19: Application layer service definition – Type 19 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 19 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

- 1) the FAL user at the boundary between the user and the application layer of the fieldbus reference model, and
- 2) 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 (Ed.2.0), *Industrial communication networks – Fieldbus specifications – Part 1: Overview and guidance for the IEC 61158 and IEC 61784 series*

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

ISO/IEC 8822, *Information technology – Open Systems Interconnection – Presentation service definition*

ISO/IEC 8824, *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 10646-1, *Information technology – Universal Multiple-Octet Coded Character Set (UCS) – Architecture and Basic Multilingual Plane*

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 terms

- a) object identifier
- b) type

#### 3.5 Fieldbus application-layer specific definitions

##### 3.6

##### **coded character set; code**

set of unambiguous rules that establish a character set and one-to-one relationship between the characters of the set and their representation by one or more bit combinations

##### 3.7

##### **cross communication**

direct data transfer between slave devices (without active involvement of master)

##### 3.8

##### **cycle time**

duration of a communication cycle

### 3.9

#### **cyclic data**

part of a telegram, which does not change its meaning during cyclic operation of the network

### 3.10

#### **device**

a slave in the communication network, (e.g., a power drive system as defined in the IEC 61800 standard family, I/O stations as defined in the IEC 61131 standard family).

### 3.11

#### **device status**

four adjacent octets inside the acknowledge telegram containing status information for each device

### 3.12

#### **element**

part of IDNs – each IDN has 7 elements, whereas each one has a specific meaning (e.g., number, name, data)

### 3.13

#### **hot plug**

possibility to open the communication network and insert or remove slaves while the network is still in real-time operation

### 3.14

#### **identification number (IDN)**

designation of operating data under which a data block is preserved with its attribute, name, unit, minimum and maximum input values, and the data

### 3.15

#### **loopback**

mode by which a device passes on a received telegram to the same port and to the other port, either changed or unchanged

### 3.16

#### **master**

node, which assigns the other nodes (i.e., slaves) the right to transmit

### 3.17

#### **physical layer**

first layer of the ISO-OSI reference model

### 3.18

#### **protocol**

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

### 3.19

#### **service channel (SVC)**

non real-time transmission of information upon master request during RT channel

### 3.20

#### **slave**

node, which is assigned the right to transmit by the master

### 3.21

#### **station**

node

**3.22****topology**

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

**3.23 Abbreviations and symbols**

|                |                                                            |
|----------------|------------------------------------------------------------|
| <b>AHS</b>     | Service transport handshake of the device (acknowledge HS) |
| <b>AP</b>      | Application Process                                        |
| <b>APO</b>     | Application Object                                         |
| <b>AR</b>      | Application Relationship                                   |
| <b>AREP</b>    | Application Relationship End Point                         |
| <b>ASE</b>     | Application Service Element                                |
| <b>CC-data</b> | Cross Communication                                        |
| <b>Cnf</b>     | Confirmation                                               |
| <b>DA</b>      | Destination address                                        |
| <b>DAT</b>     | Duration of acknowledge telegram                           |
| <b>FAL</b>     | Fieldbus Application Layer                                 |
| <b>ID</b>      | Identification Number                                      |
| <b>IDN</b>     | Identification Number                                      |
| <b>Ind</b>     | Indication                                                 |
| <b>MS</b>      | Master Slave                                               |
| <b>NRC</b>     | Non Real Time Channel                                      |
| <b>Req</b>     | Request                                                    |
| <b>Rsp</b>     | Response                                                   |
| <b>RTC</b>     | Real Time Channel                                          |
| <b>RTE</b>     | Real Time Ethernet                                         |

**3.24 Conventions****3.24.1 Overview**

The FAL is defined as a set of object-oriented ASEs. Each ASE is specified in a separate subclause. 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.

**3.24.2 General conventions**

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

**3.24.3 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:

|                      |                                   |
|----------------------|-----------------------------------|
| <b>FAL ASE:</b>      | <b>ASE Name</b>                   |
| <b>CLASS:</b>        | <b>Class name</b>                 |
| <b>CLASS ID:</b>     | <b>#</b>                          |
| <b>PARENT CLASS:</b> | Parent class name                 |
| <b>ATTRIBUTES:</b>   |                                   |
| 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.
- 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.
  - The label "OpsService" designates an operational service (1).
  - The label "MgtService" designates an management service (2).
  - 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.

### 3.24.4 Conventions for service definitions

#### 3.24.4.1 General

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

#### 3.24.4.2 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 uses 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

- 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.
- 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.

### 3.24.4.3 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

The common concepts and templates used to describe the application layer service in this standard are detailed in IEC/TR 61158-1, Clause 9.

## 5 Data type ASE

Data types as specified in IEC/TR 61158-1, Clause 9 is applied with the following restrictions:

Only nesting level of 1 is supported.

Only the following basic data types are supported:

BitString8

BitString16

BitString32

BitString64

Unsigned16

Unsigned32

Unsigned64

Integer16

Integer32

Integer64

VisibleString1

Float32

Float64

### 5.1 Bitstring types

#### 5.1.1 BitString8

**CLASS:**

**Data type**

**ATTRIBUTES:**

|     |                              |   |              |
|-----|------------------------------|---|--------------|
| 1   | Data type Numeric Identifier | = | 22           |
| 2   | Data type Name               | = | Bitstring8   |
| 3   | Format                       | = | FIXED LENGTH |
| 5.1 | Octet Length                 | = | 1            |

This type contains 1 element of type BitString.

### 5.1.2 BitString16

**CLASS:** Data type

**ATTRIBUTES:**

|     |                              |   |              |
|-----|------------------------------|---|--------------|
| 1   | Data type Numeric Identifier | = | 23           |
| 2   | Data type Name               | = | Bitstring16  |
| 3   | Format                       | = | FIXED LENGTH |
| 5.1 | Octet Length                 | = | 2            |

This type is a BitString16 and has a length of two octets.

### 5.1.3 BitString32

**CLASS:** Data type

**ATTRIBUTES:**

|     |                              |   |              |
|-----|------------------------------|---|--------------|
| 1   | Data type Numeric Identifier | = | 24           |
| 2   | Data type Name               | = | Bitstring32  |
| 3   | Format                       | = | FIXED LENGTH |
| 5.1 | Octet Length                 | = | 4            |

This type is a BitString16 and has a length of four octets.

### 5.1.4 BitString64

**CLASS:** Data type

**ATTRIBUTES:**

|     |                              |   |              |
|-----|------------------------------|---|--------------|
| 1   | Data type Numeric Identifier | = | 57           |
| 2   | Data type Name               | = | Bitstring64  |
| 3   | Format                       | = | FIXED LENGTH |
| 5.1 | Octet Length                 | = | 8            |

This type is a BitString16 and has a length of eight octets.

## 5.2 Unsigned types

### 5.2.1 Unsigned16

**CLASS:** Data type

**ATTRIBUTES:**

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

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

### 5.2.2 Unsigned32

**CLASS:** Data type

**ATTRIBUTES:**

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

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

### 5.2.3 Unsigned64

|                                |                  |
|--------------------------------|------------------|
| <b>CLASS:</b>                  | <b>Data type</b> |
| <b>ATTRIBUTES:</b>             |                  |
| 1 Data type Numeric Identifier | = 56             |
| 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 unsigned type has a length of eight octets.

## 5.3 Integer types

### 5.3.1 Integer16

|                                |                  |
|--------------------------------|------------------|
| <b>CLASS:</b>                  | <b>Data type</b> |
| <b>ATTRIBUTES:</b>             |                  |
| 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.

### 5.3.2 Integer32

|                                |                  |
|--------------------------------|------------------|
| <b>CLASS:</b>                  | <b>Data type</b> |
| <b>ATTRIBUTES:</b>             |                  |
| 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.

### 5.3.3 Integer64

|                                |                  |
|--------------------------------|------------------|
| <b>CLASS:</b>                  | <b>Data type</b> |
| <b>ATTRIBUTES:</b>             |                  |
| 1 Data type Numeric Identifier | = 55             |
| 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.

## 5.4 Floating Point types

### 5.4.1 Float32

|                                |                  |
|--------------------------------|------------------|
| <b>CLASS:</b>                  | <b>Data type</b> |
| <b>ATTRIBUTES:</b>             |                  |
| 1 Data type Numeric Identifier | = 8              |
| 2 Data type Name               | = Float32        |
| 4 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.

### 5.4.2 Float64

| <b>CLASS:</b>      |                              | <b>Data type</b> |
|--------------------|------------------------------|------------------|
| <b>ATTRIBUTES:</b> |                              |                  |
| 1                  | Data type Numeric Identifier | = 15             |
| 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.

## 5.5 Structure types

### 5.5.1 STRING2

| <b>CLASS:</b>      |                              | <b>Data type</b>          |
|--------------------|------------------------------|---------------------------|
| <b>ATTRIBUTES:</b> |                              |                           |
| 1                  | Data type Numeric Identifier | = not used                |
| 2                  | Data type Name               | = STRING2                 |
| 3                  | Format                       | = STRUCTURE               |
| 5.1                | Number of Fields             | = 2                       |
| 5.2.1              | Field Name                   | = Charcount_Element       |
| 5.2.2              | Field Data type              | = UINT                    |
| 5.3.1              | Field Name                   | = String2contents_Element |
| 5.3.2              | Field Data type              | = OctetString             |

This IEC 61131-3 data type extension is composed of two elements. Charcount\_Element gives the current number of characters in the String2contents\_Element (one UINT per character). Characters are as specified in ISO 10646.

## 6 Communication model specification

### 6.1 Concepts

#### 6.1.1 Communication mechanisms

Two communication mechanisms are supported by devices within the network:

- cyclic transmission of data in a high efficient manner using a publisher subscriber model,
- non cyclic transmission of data using a client server communication model.

AREPs, which act as push publisher or push subscriber, are used for cyclic transmission.

AREPs, which act as a client or server, are used for non cyclic data transmission.

#### 6.1.2 IDN concept

The application data which is transmitted cyclically and non cyclically between FAL users is mapped on so-called identification numbers (IDNs). These IDNs correspond to the APOs as defined and are described in IEC 61158-3-16, Annex A.

### 6.2 ASEs

#### 6.2.1 Identification number (IDN) ASE

##### 6.2.1.1 Overview

The IDN ASE provides read and write access to the attributes of IDNs provided by a device.

## 6.2.1.2 IDN class specification

### 6.2.1.2.1 Formal model

|               |                                          |
|---------------|------------------------------------------|
| FAL ASE:      | IDN ASE                                  |
| CLASS:        | IDN                                      |
| CLASS ID:     | not used                                 |
| PARENT CLASS: | TOP                                      |
| ATTRIBUTES:   |                                          |
| 1             | (m) Key Attribute: Identification        |
| 1.1           | (m) Key Attribute: Identification Number |
| 1.2           | (o) Key Attribute: Resource Element      |
| 1.3           | (o) Key Attribute: Subindex              |
| 2             | (o) Attribute: Name                      |
| 3             | (m) Attribute: Data Attribute            |
| 4             | (o) Attribute: Unit                      |
| 5             | (o) Attribute: Minimum value             |
| 6             | (o) Attribute: Maximum value             |
| 7             | (m) Attribute: Operation Data            |
| SERVICES:     |                                          |
| 1             | (m) OpsService: Read                     |
| 2             | (m) OpsService: Write                    |

### 6.2.1.2.2 Attributes

#### Identification

This key attribute identifies an instance of this object class. The addressing can be simple or extended.

#### Identification number

This mandatory attribute is a numerical identifier to address an instance of this object class. In case of a simple addressing scheme, this is the only attribute required for identification.

#### Resource element

This optional attribute is used in case of an extended addressing.

#### Subindex

This optional attribute is used in case of an extended addressing.

#### Name

This optional attribute specifies a symbolic name of this object class.

#### Data attribute

This mandatory attribute specifies all information which is needed to display or convert the data intelligibly. This includes data type, data length, conversion factor, read/write permission depending on the communication phase and whether the data is associated to a command.

#### Unit

This optional attribute specifies the unit of the operation data.

#### Minimum value

This optional attribute specifies the minimum input value for the operation data.

#### Maximum value

This optional attribute specifies the maximum input value for the operation data.

#### Operation data

This mandatory attribute specifies the operation data of this instance of the object class.

### 6.2.1.3 IDN ASE service specification

#### 6.2.1.3.1 Supported services

This subclause specifies the definition of the services that are unique to this ASE. The services defined for this ASE are:

- Read
- Write

#### 6.2.1.3.2 Read service

##### 6.2.1.3.2.1 Service overview

This confirmed service is used to read an element of an IDN on demand.

##### 6.2.1.3.2.2 Service primitives

The service parameters for each primitive are shown in Table 1.

**Table 1 – Read service parameters**

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

#### **AREP**

This parameter is the local identifier for the desired AR.

#### **Device address**

This parameter identifies the address of the device.

#### **Identification**

This parameter specifies an IDN object to be read by the key attribute.

#### **Attribute**

This parameter specifies the element of an IDN object to be read by the key attribute.

#### **Value**

This parameter specifies the value read.

#### **Error info**

This parameter provides error information for service errors.

### 6.2.1.3.3 Write service

#### 6.2.1.3.3.1 Service overview

This confirmed service is used to write an element of an IDN.

#### 6.2.1.3.3.2 Service primitives

The service parameters for each primitive are shown in Table 2.

**Table 2 – Write service parameters**

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

#### **AREP**

This parameter is the local identifier for the desired AR.

#### **Device address**

This parameter identifies the address of the device.

#### **Identification**

This parameter specifies an IDN object to be written by the key attribute.

#### **Attribute**

This parameter specifies the element of an IDN object to be written by the key attribute.

#### **Value**

This parameter specifies the value to be written.

#### **Error info**

This parameter provides error information for service errors.

## 6.2.2 CYCIDN ASE

### 6.2.2.1 Overview

The CYCIDN ASE provides cyclic read and write access to the operation data of IDNs provided by a device.

### 6.2.2.2 Cyclic Identification Number (CYCIDN) class specification

#### 6.2.2.2.1 Formal model

|           |            |
|-----------|------------|
| FAL ASE:  | CYCIDN ASE |
| CLASS:    | CYCIDN     |
| CLASS ID: | not used   |

|               |     |                |                       |
|---------------|-----|----------------|-----------------------|
| PARENT CLASS: |     |                | TOP                   |
| ATTRIBUTES:   |     |                |                       |
| 1             | (m) | Key Attribute: | Identification        |
| 1.1           | (m) | Key Attribute: | Identification Number |
| 1.2           | (o) | Key Attribute: | Resource Element      |
| 1.3           | (o) | Key Attribute: | Subindex              |
| 2             | (m) | Attribute:     | Data                  |
| SERVICES:     |     |                |                       |
| 1             | (m) | OpsService:    | Read                  |
| 2             | (o) | OpsService:    | Write                 |
| 3             | (m) | OpsService:    | Notify                |

#### 6.2.2.2.2 Attributes

##### AREP

This parameter is the local identifier for the desired AR.

##### Identification

This parameter specifies an IDN object to be read by the key attribute.

##### Data

This attribute specifies the data which is read or written.

#### 6.2.2.3 CYCIDN ASE service specification

##### 6.2.2.3.1 Supported services

This subclause specifies the definition of services that are unique to this ASE. The services defined for this ASE are:

- Read
- Write
- Notify

##### 6.2.2.3.2 Read service

##### 6.2.2.3.2.1 Service overview

This unconfirmed service is used to cyclically read operation data of an IDN.

##### 6.2.2.3.2.2 Service primitives

The service parameters for each primitive are shown in Table 3.

**Table 3 – Read service parameters**

| Parameter name | Req | Ind | Rsp | Cnf |
|----------------|-----|-----|-----|-----|
| Argument       |     |     |     |     |
| AREP           | M   |     |     |     |
| Device Address | M   |     |     |     |
| Identification | M   |     |     |     |
| Operation Data | M   |     |     |     |
| Error Info     | M   |     |     |     |

##### AREP

This parameter is the local identifier for the desired AR.

##### Device address

This parameter identifies the address of the device.

### Identification

This parameter specifies an IDN object to be read by the key attribute.

### Operation data

This parameter specifies the operation data which is being read.

### Error info

This parameter provides error information for service errors.

## 6.2.2.3.3 Write service

### 6.2.2.3.3.1 Service overview

This confirmed service is used to write cyclically an element of an IDN.

### 6.2.2.3.3.2 Service primitives

The service parameters for each primitive are shown in Table 4.

**Table 4 – Write service parameters**

| Parameter name                                                                                                                            | Req | Ind | Cnf |
|-------------------------------------------------------------------------------------------------------------------------------------------|-----|-----|-----|
| Argument                                                                                                                                  |     |     |     |
| AREP                                                                                                                                      | M   |     |     |
| Device Address                                                                                                                            | M   |     |     |
| Identification                                                                                                                            | M   |     |     |
| Operation Data                                                                                                                            | M   |     |     |
| NOTE The method by which a confirm primitive is correlated with its corresponding preceding request primitive is a local matter. See 1.2. |     |     |     |

### AREP

This parameter is the local identifier for the desired AR.

### Device address

This parameter identifies the address of the device.

### Identification

This parameter specifies an IDN object to be written by the key attribute.

### Operation data

This parameter specifies the operation data which is being written.

### Error info

This parameter provides error information for service errors.

## 6.2.2.3.4 Notify service

### 6.2.2.3.4.1 Service overview

This service is used to notify that new cyclic data has been received. This service may be used by an application to trigger the cyclic reading and writing of data depending on the device implementation.

### 6.2.2.3.4.2 Service primitives

The service parameters for each primitive are shown in Table 5.

**Table 5 – Notify service parameters**

| Parameter name | Ind |
|----------------|-----|
| Argument       |     |
| AREP           | M   |

**AREP**

This parameter is the local identifier for the desired AR.

**6.2.3 Management (MGT) ASE****6.2.3.1 Overview**

The MGT ASE provides management services for a Type 19 network and Type 19 devices.

**6.2.3.2 MGT class specification****6.2.3.2.1 Formal model**

|                   |                              |
|-------------------|------------------------------|
| FAL ASE:          | MGT ASE                      |
| CLASS:            | MGT ASE                      |
| CLASS ID:         | not used                     |
| PARENT CLASS:     | TOP                          |
| ATTRIBUTES:       |                              |
| 1 (m) Attribute:  | Topology                     |
| SERVICES:         |                              |
| 1 (m) OpsService: | Get Network Status           |
| 2 (m) OpsService: | Get Device Status            |
| 3 (m) OpsService: | Network Status Change Report |
| 4 (m) OpsService: | Device Status Change Report  |
| 5 (m) OpsService: | Set Device Status            |
| 6 (m) OpsService: | Enable RTC                   |
| 7 (m) OpsService: | Notify RTC                   |
| 8 (m) OpsService: | Disable RTC                  |

**6.2.3.2.2 Attributes****6.2.3.2.2.1 Topology**

This attribute specifies the current topology of the network.

**6.2.3.3 MGT ASE service specification****6.2.3.3.1 Supported services**

This subclause specifies the definition of the services that are unique to this ASE.

The services defined for this ASE are:

- Get Network Status
- Get Device Status
- Network Status Change Report
- Device Status Change Report
- Set Device Status
- Enable RTC
- Enable Hotplug
- Notify RTC

- Disable RTC

### 6.2.3.3.2 Get network status service

#### 6.2.3.3.2.1 Service overview

This local service is used to obtain the status of the network.

#### 6.2.3.3.2.2 Service primitives

The service parameters for each primitive are shown in Table 6.

**Table 6 – Get network status service parameters**

| Parameter name                                                                                                                            | Req | Cnf    |
|-------------------------------------------------------------------------------------------------------------------------------------------|-----|--------|
| Argument<br>AREP                                                                                                                          | M   |        |
| Result<br>Network Status                                                                                                                  |     | M<br>M |
| NOTE The method by which a confirm primitive is correlated with its corresponding preceding request primitive is a local matter. See 1.2. |     |        |

#### AREP

This parameter is the local identifier for the desired AR.

#### Network status

This parameter indicates consistency of the primary and secondary networks. Possible values are:

- RING
- LINE
- ERROR

### 6.2.3.3.3 Get device status service

#### 6.2.3.3.3.1 Service overview

This local service is used to obtain the status of the specified device.

#### 6.2.3.3.3.2 Service primitives

The service parameters for each primitive are shown in Table 7.

**Table 7 – Get device status service parameters**

| Parameter name                                                                                                                            | Req    | Ind | Rsp | Cnf    |
|-------------------------------------------------------------------------------------------------------------------------------------------|--------|-----|-----|--------|
| Argument<br>AREP<br>Device Identifier                                                                                                     | M<br>M |     |     |        |
| Result<br>Device Status                                                                                                                   |        |     |     | M<br>M |
| NOTE The method by which a confirm primitive is correlated with its corresponding preceding request primitive is a local matter. See 1.2. |        |     |     |        |

**AREP**

This parameter is the local identifier for the desired AR.

**Device identifier**

This parameter defines the device for which the status is requested.

**Device status**

This parameter indicates the status of the device specified by the device identifier, and includes the following information.

## a) existence

|       |                           |
|-------|---------------------------|
| TRUE  | the device exists         |
| FALSE | the device does not exist |

## b) mode

|    |                                    |
|----|------------------------------------|
| LB | the device is in loopback mode     |
| FF | the device is in fast forward mode |

**6.2.3.3.4 Network status change report service****6.2.3.3.4.1 Service overview**

This local service is used to inform of changes in network status.

**6.2.3.3.4.2 Service primitives**

The service parameters for each primitive are shown in Table 8.

**Table 8 – Network status change report service parameters**

| Parameter name | Ind |
|----------------|-----|
| Argument       |     |
| AREP           | M   |
| Network Status | M   |

**AREP**

This parameter is the local identifier for the desired AR.

**Network status**

This parameter indicates the current status of the network, and has the following possible values:

- RING
- LINE
- ERROR

**6.2.3.3.5 Device status change report service****6.2.3.3.5.1 Service overview**

This local service is used to inform of changes in the device status.

**6.2.3.3.5.2 Service primitives**

The service parameters for each primitive are shown in Table 9.

**Table 9 – Station status change report service parameters**

| Parameter name    | Ind |
|-------------------|-----|
| Argument          |     |
| AREP              | M   |
| Device Identifier | M   |
| Device Status     | M   |

### **AREP**

This parameter is the local identifier for the desired AR.

### **Device identifier**

This parameter indicates the status of the device, of which the status has been changed.

### **Device status**

This parameter indicates the status of the device specified by the request primitive and includes the following information.

#### **a) existence**

|       |                           |
|-------|---------------------------|
| TRUE  | the device exists         |
| FALSE | the device does not exist |

#### **b) mode**

|    |                                    |
|----|------------------------------------|
| LB | the device is in loopback mode     |
| FF | the device is in fast forward mode |

### **6.2.3.3.6 Set device status service**

#### **6.2.3.3.6.1 Service overview**

This local service is used by the master to set the status of the specified device.

#### **6.2.3.3.6.2 Service primitives**

The service parameters for each primitive are shown in Table 10.

**Table 10 – Set device status service parameters**

| Parameter name   | Req | Ind   | Rsp | Cnf |
|------------------|-----|-------|-----|-----|
| Argument         |     |       |     |     |
| AREP             | M   | M     |     |     |
| DeviceIdentifier | M   | M (=) |     |     |
| DeviceStatus     | M   | M (=) |     |     |

### **AREP**

This parameter is the local identifier for the desired AR.

### **Device identifier**

This parameter indicates the remote device for which the status is to be set.

### **Device status**

This parameter specifies the mode of the device specified by the device identifier, and includes the following information:

|    |                                    |
|----|------------------------------------|
| LB | the device is in loopback mode     |
| FF | the device is in fast forward mode |

### 6.2.3.3.7 Enable RTC

#### 6.2.3.3.7.1 Service overview

This service is used by the master to enable the cyclic communication (RTC) in the network.

#### 6.2.3.3.7.2 Service primitives

The service parameters for each primitive are shown in Table 11.

**Table 11 – Enable RTC service parameters**

| Parameter name                                                                                                                            | Req | Cnf |
|-------------------------------------------------------------------------------------------------------------------------------------------|-----|-----|
| Argument                                                                                                                                  |     |     |
| AREP                                                                                                                                      | M   |     |
| Cycle Time                                                                                                                                | M   |     |
| List of Device Addresses                                                                                                                  | M   |     |
| List of command values for each device                                                                                                    | M   |     |
| List of feedback values for each device                                                                                                   | M   |     |
| List of CC producer data for each device                                                                                                  | O   |     |
| List of CC consumer data for each device                                                                                                  | O   |     |
| Topology                                                                                                                                  | O   |     |
| Result(+)                                                                                                                                 |     | S   |
| List of Present Device Addresses                                                                                                          |     | M   |
| Result(-)                                                                                                                                 |     | S   |
| Error Info                                                                                                                                |     | M   |
| NOTE The method by which a confirm primitive is correlated with its corresponding preceding request primitive is a local matter. See 1.2. |     |     |

#### **AREP**

This parameter is the local identifier for the desired AR.

#### **Cycle time**

This parameter defines the cycle time for a network.

#### **List of device addresses**

This parameter specifies a list of devices that are part of a network.

#### **List of command values for each device**

This parameter specifies a list of command values for each device that is part of a network.

#### **List of feedback values for each device**

This parameter specifies a list of feedback values for each device that is part of a network.

#### **List of CC producer data for each device**

This parameter specifies a list of CC producer data for each device that is part of a network.

#### **List of CC consumer data for each device**

This parameter specifies a list of CC consumer data for each device that is part of the network.

#### **Topology**

This parameter defines the expected topology of a network.

### List of present device addresses

This parameter specifies a list of present devices in a network.

### Error info

This parameter provides error information for service errors.

## 6.2.3.3.8 Enable hotplug

### 6.2.3.3.8.1 Service overview

This service is used by the master to enable the hotplug capability in a network.

### 6.2.3.3.8.2 Service primitives

The service parameters for each primitive are shown in Table 12.

**Table 12 – Enable hotplug service parameters**

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

### AREP

This parameter is the local identifier for the desired AR.

### List of device addresses

This parameter specifies a list of devices that are to be hot plugged to a network.

### Error info

This parameter provides error information for service errors.

## 6.2.3.3.9 Notify RTC

### 6.2.3.3.9.1 Service overview

This service is used to notify a slave device that the network has been initialized.

### 6.2.3.3.9.2 Service primitives

The service parameters for each primitive are shown in Table 13.

**Table 13 – Notify RTC service parameters**

| Parameter name           | Ind |
|--------------------------|-----|
| Argument                 |     |
| AREP                     | M   |
| Cycle Time               | M   |
| List of command values   | M   |
| List of feedback values  | M   |
| List of CC producer data | O   |
| List of CC consumer data | O   |
| Topology                 | O   |

**AREP**

This parameter is the local identifier for the desired AR.

**Cycle time**

This parameter defines the cycle time for a network which is set by the master device.

**List of command values**

This parameter specifies a list of command values which are configured by the master device.

**List of feedback values**

This parameter specifies a list of feedback values which are configured by the master device.

**List of CC producer data**

This optional parameter specifies a list of CC producer data which are configured by the master device.

**List of CC consumer data**

This optional parameter specifies a list of CC consumer data which are configured by the master device.

**Topology**

This optional parameter defines the expected topology as set by the master device.

**6.2.3.3.10 Disable RTC****6.2.3.3.10.1 Service overview**

This service is used by the master to disable the cyclic communication (RTC) in a network.

**6.2.3.3.10.2 Service primitives**

The service parameters for each primitive are shown in Table 14.

**Table 14 – Disable RTC service parameters**

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

## **AREP**

This parameter is the local identifier for the desired AR.

## **Error Info**

This parameter provides error information for service errors.

## **6.3 ARs**

### **6.3.1 General**

Application Relationships provided are

- point-to-point user-Triggered Confirmed client/server AREP (SVC-AR);
- point-to-point network-Scheduled Unconfirmed publisher/subscriber AREP (RTC-MS-AR);
- multipoint network-Scheduled Unconfirmed publisher/subscriber AREP (RTC-CC-AR).

### **6.3.2 Point-to-point user-triggered confirmed client/server AREP (SVC)**

This class is defined to support the on-demand exchange of confirmed services between a master and a slave application process. Unconfirmed services are not supported by this type of AR. The behavior of this class is described as follows. An IDN ASE user wishing to convey a request APDU submits it as an IDN 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 IDN ASE user in the order in which it was received.

The AREP receiving the request APDU accepts the corresponding response APDU from its IDN 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 IDN ASE user in the order in which it was received. It also stops its associated service response timer.

### **6.3.3 Point-to-point network-scheduled unconfirmed publisher/subscriber AREP (RTC-MS)**

This class is defined to support the “push” model for scheduled unbuffered distribution of unconfirmed services between a master and a slave (MS) 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. Sending AREP writes the APDU into the internal buffer, completely replacing the existing contents of the buffer. The AR ASE transfers the buffer contents at the next scheduled transfer opportunity.

If the AREP receives another APDU before the buffer contents are transmitted, the buffer contents will be replaced with the new APDU, and the previous APDU will be lost. When the buffer contents are transmitted, the AR ASE notifies the user of transmission.

At the receiving endpoint, the APDU is received from the network and is written immediately into the buffer, completely overwriting the existing contents of the buffer. 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, it will be overwritten by the next APDU and lost.

An FAL user receiving the buffered transmission may request to receive the currently buffered APDU later.

#### 6.3.4 Point-to-multipoint network-scheduled unconfirmed publisher/subscriber AREP (RTC-CC)

This class is defined to support the “push” model for scheduled and buffered distribution of unconfirmed services to one or more application processes. The services provided by this AREP may be used for a direct communication between devices, named cross communication (CC).

The behaviour 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 buffer, completely replacing the existing contents of the buffer.

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

If the AREP receives another APDU before the buffer contents are transmitted, the buffer contents will be replaced with the new APDU, and the previous APDU will be lost. When the buffer contents are transmitted, the AR ASE notifies the user of transmission.

At the receiving endpoint, the APDU is received from the network and is written immediately into the buffer, completely overwriting the existing contents of the buffer. 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, it will be overwritten by the next APDU and lost.

An FAL user receiving the buffered transmission may request to receive the currently buffered APDU later.

### 6.4 Summary of AR classes

Table 15 defines the characteristics of point-to-point user-triggered confirmed client/server AREP (SVC) class. The Class ID values have been assigned to be compatible with existing standards.

**Table 15 – AREP (SVC) class summary**

| FAL ASE           | Class          | Class ID |
|-------------------|----------------|----------|
| Roles             | Client         | TBD      |
|                   | Server         | TBD      |
| Cardinality       | one-to-one     | TBD      |
| Conveyance paths  | Bi-directional | TBD      |
| Trigger policy    | User-triggered | TBD      |
| Conveyance policy | Queued         | TBD      |

Table 16 defines the characteristics of point-to-point network-scheduled unconfirmed publisher/subscriber AREP (RTC-MS) class. The Class ID values have been assigned to be compatible with existing standards.

**Table 16 – AREP (RTC-MS) class summary**

| FAL ASE           | Class             | Class ID |
|-------------------|-------------------|----------|
| Roles             | Publisher         | TBD      |
|                   | Subscriber        | TBD      |
| Cardinality       | One-to-one        | TBD      |
| Conveyance paths  | Unidirectional    | TBD      |
| Trigger policy    | Network-scheduled | TBD      |
| Conveyance policy | Buffered          | TBD      |

Table 17 defines the characteristics of point-to-point network-scheduled unconfirmed publisher/subscriber AREP (RTC-CC) class. The Class ID values have been assigned to be compatible with existing standards.

**Table 17 – AREP (RTC-CC) class summary**

| FAL ASE           | Class             | Class ID |
|-------------------|-------------------|----------|
| Roles             | Publisher         | TBD      |
|                   | Subscriber        | TBD      |
| Cardinality       | One-to-many       | TBD      |
| Conveyance paths  | Unidirectional    | TBD      |
| Trigger policy    | Network-scheduled | TBD      |
| Conveyance policy | Non buffered      | TBD      |

## 6.5 Permitted FAL services by AREP role

Table 18 below defines the valid combinations of services and AR types (which service APDUs) and can be sent or received by AR with the specified type. “Unc” and “Cnf” columns indicate whether the service listed in the left-hand column is unconfirmed or confirmed respectively.

**Table 18 – FAL services by AR type**

| FAL Services       | Used AREPs |        | Client |     | Server |     | Publisher | Subscriber |
|--------------------|------------|--------|--------|-----|--------|-----|-----------|------------|
|                    | SVC        | RTC-MS | req    | cnf | ind    | res | req       | req        |
| <b>IDN ASE</b>     |            |        |        |     |        |     |           |            |
| Read               | X          |        | X      | X   | X      | X   |           |            |
| Write              | X          |        | X      | X   | X      | X   |           |            |
| <b>CYC IDN ASE</b> |            |        |        |     |        |     |           |            |
| Read               |            | X      |        |     |        |     |           | X          |
| Write              |            | X      |        |     |        |     | X         |            |

## Bibliography

IEC 61131-1, *Programmable controllers — Part 1: General information*

IEC 61158-3-19, *Industrial communication networks – Fieldbus specifications – Part 3-19: Data-link layer service definition – Type 19 elements*

IEC 61158-4-19, *Industrial communication networks – Fieldbus specifications – Part 4-19: Data-link layer protocol specification – Type 19 elements*

IEC 61158-6-19, *Industrial communication networks – Fieldbus specifications – Part 6-19: Application layer protocol specification – Type 19 elements*

IEC 61784-1 (Ed.2.0), *Industrial communication networks – Profiles – Part 1: Fieldbus profiles*

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

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

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

ISO/IEC 8859-1, *Information technology – 8-bit single-octet coded graphic character sets – Part 1: Latin alphabet No. 1*

---

## SOMMAIRE

|                                                                              |    |
|------------------------------------------------------------------------------|----|
| AVANT-PROPOS.....                                                            | 36 |
| INTRODUCTION.....                                                            | 38 |
| 1 Domaine d'application .....                                                | 39 |
| 1.1 Vue d'ensemble.....                                                      | 39 |
| 1.2 Spécifications.....                                                      | 40 |
| 1.3 Conformité .....                                                         | 40 |
| 2 Références normatives.....                                                 | 40 |
| 3 Termes, définitions, abréviations, symboles et conventions .....           | 41 |
| 3.1 Termes de l'ISO/CEI 7498-1 .....                                         | 41 |
| 3.2 Termes de l'ISO/CEI 8822 .....                                           | 41 |
| 3.3 Termes de l'ISO/CEI 9545 .....                                           | 41 |
| 3.4 Termes de l'ISO/CEI 8824 .....                                           | 42 |
| 3.5 Définitions spécifiques à la couche application des bus de terrain ..... | 42 |
| 3.6 Abréviations et symboles .....                                           | 42 |
| 3.7 Conventions .....                                                        | 42 |
| 4 Concepts.....                                                              | 47 |
| 5 Data type ASE.....                                                         | 47 |
| 5.1 Types Bitstring (chaîne de bits).....                                    | 47 |
| 5.2 Types Unsigned (non signés).....                                         | 48 |
| 5.3 Types Integer (entiers).....                                             | 49 |
| 5.4 Types Floating Point (virgule flottante).....                            | 49 |
| 5.5 Types Structure.....                                                     | 50 |
| 6 Spécification de modèle de communication .....                             | 50 |
| 6.1 Concepts.....                                                            | 50 |
| 6.2 Les ASE.....                                                             | 51 |
| 6.3 Les AR.....                                                              | 63 |
| 6.4 Résumé des classes AR.....                                               | 65 |
| 6.5 Services de FAL autorisés par rôle d'AREP.....                           | 66 |
| Bibliographie.....                                                           | 67 |
| Tableau 1 – Paramètres du service "Read" .....                               | 52 |
| Tableau 2 – Paramètres du service "Write" .....                              | 53 |
| Tableau 3 – Paramètres du service "Read" .....                               | 55 |
| Tableau 4 – Paramètres du service "Write" .....                              | 55 |
| Tableau 5 – Paramètres du service "Notify" .....                             | 56 |
| Tableau 6 – Primitives du service "Get network status" .....                 | 57 |
| Tableau 7 – Primitives du service "Get device status" .....                  | 58 |
| Tableau 8 – Paramètres du service "Network status change report" .....       | 58 |
| Tableau 9 – Paramètres du service "Station status change report" .....       | 59 |
| Tableau 10 – Primitives du service "Set device status" .....                 | 60 |
| Tableau 11 – Paramètres du service "Enable RTC" .....                        | 60 |
| Tableau 12 – Paramètres du service "Enable hotplug" .....                    | 62 |
| Tableau 13 – Paramètres du service "Notifiy RTC" .....                       | 62 |

|                                                        |    |
|--------------------------------------------------------|----|
| Tableau 14 – Paramètres du service "Disable RTC" ..... | 63 |
| Tableau 15 – Résumé des classes AREP (SVC) .....       | 65 |
| Tableau 16 – Résumé des classes AREP (RTC-MS) .....    | 66 |
| Tableau 17 – Résumé des classes AREP (RTC-CC) .....    | 66 |
| Tableau 18 – Services de FAL par type d'AR .....       | 66 |

IECNORM.COM : Click to view the full PDF of IEC 61158-5-19:2007

Withdrawing

## COMMISSION ÉLECTROTECHNIQUE INTERNATIONALE

### RÉSEAUX DE COMMUNICATION INDUSTRIELS – SPÉCIFICATIONS DES BUS DE TERRAIN –

#### Partie 5-19: Définition des services de la couche application – Éléments de Type 19

#### AVANT-PROPOS

- 1) La Commission Electrotechnique Internationale (CEI) est une organisation mondiale de normalisation composée de l'ensemble des comités électrotechniques nationaux (Comités nationaux de la CEI). La CEI a pour objet de favoriser la coopération internationale pour toutes les questions de normalisation dans les domaines de l'électricité et de l'électronique. A cet effet, la CEI – entre autres activités – publie des Normes internationales, des Spécifications techniques, des Rapports techniques, des Spécifications accessibles au public (PAS) et des Guides (ci-après dénommés "Publication(s) de la CEI"). Leur élaboration est confiée à des comités d'études, aux travaux desquels tout Comité national intéressé par le sujet traité peut participer. Les organisations internationales, gouvernementales et non gouvernementales, en liaison avec la CEI, participent également aux travaux. La CEI collabore étroitement avec l'Organisation Internationale de Normalisation (ISO), selon des conditions fixées par accord entre les deux organisations.
- 2) Les décisions ou accords officiels de la CEI concernant les questions techniques représentent, dans la mesure du possible, un accord international sur les sujets étudiés, étant donné que les Comités nationaux de la CEI intéressés sont représentés dans chaque comité d'études.
- 3) Les Publications de la CEI se présentent sous la forme de recommandations internationales et sont agréées comme telles par les Comités nationaux de la CEI. Tous les efforts raisonnables sont entrepris afin que la CEI s'assure de l'exactitude du contenu technique de ses publications; la CEI ne peut pas être tenue responsable de l'éventuelle mauvaise utilisation ou interprétation qui en est faite par un quelconque utilisateur final.
- 4) Dans le but d'encourager l'uniformité internationale, les Comités nationaux de la CEI s'engagent, dans toute la mesure possible, à appliquer de façon transparente les Publications de la CEI dans leurs publications nationales et régionales. Toutes divergences entre toutes Publications de la CEI et toutes publications nationales ou régionales correspondantes doivent être indiquées en termes clairs dans ces dernières.
- 5) La CEI n'a prévu aucune procédure de marquage valant indication d'approbation et n'engage pas sa responsabilité pour les équipements déclarés conformes à une de ses Publications.
- 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 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 la série CEI 61784. 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 61158-5-19 a été établie par le sous-comité 65C: Réseaux industriels, du comité d'études 65 de la CEI: Mesure, commande et automation dans les processus industriels.

Cette première édition et les autres normes de la sous-partie CEI 61158-5 annulent et remplacent la CEI 61158-5:2003. Cette édition de la présente partie constitue un ajout

technique. La présente partie ainsi que les normes connexes traitant du Type 19 annulent et remplacent également la CEI/PAS 62410, parue en 2005.

La présente édition de la CEI 61158-5 inclut les modifications majeures suivantes par rapport à l'édition précédente:

- a) suppression de l'ancien bus de terrain de Type 6, qui n'est plus adapté au marché;
- b) ajout de nouveaux types de bus de terrain;
- c) éclatement de la partie 5 de la troisième édition en plusieurs parties numérotées -5-2, -5-3, ...

La présente version bilingue (2014-06) correspond à la version anglaise monolingue publiée en 2007-12.

Le texte anglais de cette norme est issu des documents 65C/475/FDIS et 65C/486/RVD.

Le rapport de vote 65C/486/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.

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

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

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

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

## 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 la CEI/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 5-19: Définition des services de la couche application – Eléments de Type 19

#### 1 Domaine d'application

##### 1.1 Vue d'ensemble

La couche application de bus de terrain (FAL<sup>1</sup>) 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 19. 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) d'interrelation entre ces actions et événements, et leurs séquences valides.

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

- 1) l'utilisateur de FAL à la frontière entre l'utilisateur et la couche application du modèle de référence de bus de terrain, et
- 2) 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 (AE<sup>2</sup>) de la FAL contenues dans les processus d'application. L'AE de la FAL se compose d'un jeu d'éléments de service application (ASE<sup>3</sup>) orientés objet et d'une entité de gestion de couche (LME<sup>4</sup>) 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 (APO<sup>5</sup>), 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).

<sup>5</sup> 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 en anglais seulement)

IEC/TR 61158-1 (Ed.2.0), *Industrial communication networks – Fieldbus specifications – Part 1: Overview and guidance for the IEC 61158 and IEC 61784 series* (disponible en anglais seulement)<sup>6</sup>

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

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

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

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

ISO/CEI 10646-1, *Technologies de l'information – Jeu universel de caractères codés à plusieurs octets (JUC) – Partie 1: Architecture et table multilingue*

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
- c) unité de données de protocole application
- d) élément de service application
- e) invocation d'entité d'application
- f) invocation de processus d'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*)

<sup>6</sup> Les publications monolingues des séries CEI 61158 et CEI 61784 sont actuellement en cours de traduction.

- 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

- a) identificateur d'objet
- b) type

### 3.5 Définitions spécifiques à la couche application des bus de terrain

#### 3.6

##### **jeu de caractères codés; code**

ensemble de règles non ambiguës qui établissent un jeu de caractères et une correspondance biunivoque entre les caractères du jeu et leur représentation par une ou plusieurs combinaisons de bits

#### 3.7

##### **communication transversale**

transfert direct de données entre des dispositifs esclaves (sans implication active du maître)

#### 3.8

##### **temps de cycle**

durée d'un cycle de communication

#### 3.9

##### **donnée cyclique**

partie intégrante d'un télégramme, qui ne change pas de signification au cours de l'exploitation cyclique du réseau

#### 3.10

##### **dispositif**

esclave dans le réseau de communication (par exemple: un système d'entraînement électrique tel que défini dans la famille de normes CEI 61800, les blocs d'E/S telles que définies dans la famille de normes CEI 61131)

#### 3.11

##### **état du dispositif**

ensemble de quatre octets adjacents à l'intérieur d'un télégramme d'acquittement contenant des informations d'état pour chaque dispositif

#### 3.12

##### **élément**

partie intégrante des IDN (numéros d'identification) – chaque IDN comportant 7 éléments, chacun ayant une signification spécifique (par exemple: numéro, nom, donnée)

#### 3.13

##### **connexion à chaud ("hot plug")**

possibilité d'ouvrir le réseau de communication et d'insérer ou retirer des esclaves alors que le réseau est toujours en exploitation temps réel

**3.14****numéro d'identification (IDN, *identification number*)**

dénomination de données opérationnelles sous laquelle un bloc de données est conservé avec ses valeurs d'attribut, de nom, d'unité, d'entrée minimales et maximales et les données

**3.15****bouclage**

mode par lequel un dispositif transmet un télégramme reçu au même port et à l'autre port, soit changé, soit inchangé

**3.16****maître**

nœud qui accorde aux autres nœuds (c'est-à-dire: les esclaves) le droit d'émettre

**3.17****couche physique**

première couche du Modèle de référence de l'interconnexion des systèmes ouverts (OSI) de l'ISO

**3.18****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

**3.19****voie de services (SVC, *service channel*)**

émission hors temps réel d'informations à la demande du maître sur une voie RT

**3.20****esclave**

nœud auquel le maître a accordé le droit d'émettre

**3.21****poste**

nœud

**3.22****topologie**

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

**3.23 Abréviations et symboles**

|                   |                                                                                                                                      |
|-------------------|--------------------------------------------------------------------------------------------------------------------------------------|
| <b>AHS</b>        | handshake (prise de contact) de transport de services du dispositif ("acknowledge HS" c'est-à-dire prise de contact d'acquiescement) |
| <b>AP</b>         | Application Process (Processus application)                                                                                          |
| <b>APO</b>        | Application Object (Objet d'application)                                                                                             |
| <b>AR</b>         | Application Relationship (Relation d'applications)                                                                                   |
| <b>AREP</b>       | Application Relationship End Point (Point d'extrémité de relation d'applications)                                                    |
| <b>ASE</b>        | Application Service Element (Élément de service application)                                                                         |
| <b>Données CC</b> | Cross Communication (Communication transversale)                                                                                     |
| <b>Cnf</b>        | Confirmation                                                                                                                         |
| <b>DA</b>         | Destination address (Adresse de destination)                                                                                         |
| <b>DAT</b>        | Duration of acknowledge telegram (Durée de télégramme d'acquiescement)                                                               |
| <b>FAL</b>        | Fieldbus Application Layer (Couche application de bus de terrain)                                                                    |

|            |                                                 |
|------------|-------------------------------------------------|
| <b>ID</b>  | Identification number (Numéro d'identification) |
| <b>IDN</b> | Identification number (Numéro d'identification) |
| <b>Ind</b> | Indication                                      |
| <b>MS</b>  | Master Slave (Maître Esclave)                   |
| <b>NRC</b> | Non Real Time Channel (Voie hors temps réel)    |
| <b>Req</b> | Request (Demande)                               |
| <b>Rsp</b> | Response (Réponse)                              |
| <b>RTC</b> | Real Time Channel (Voie temps réel)             |
| <b>RTE</b> | Real-Time Ethernet (Ethernet temps réel)        |

### 3.24 Conventions

#### 3.24.1 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.

#### 3.24.2 Conventions générales

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

#### 3.24.3 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:

|                      |                             |
|----------------------|-----------------------------|
| <b>FAL ASE:</b>      | <b>Nom de l'ASE</b>         |
| <b>CLASS:</b>        | <b>Nom de la classe</b>     |
| <b>CLASS ID:</b>     | <b>#</b>                    |
| <b>PARENT CLASS:</b> | Nom de la classe parent     |
| <b>ATTRIBUTS:</b>    |                             |
| 1 (o) Attribut clé:  | Identificateur numérique    |
| 2 (o) Attribut clé:  | name                        |
| 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)     |
| <b>SERVICES:</b>     |                             |
| 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 conserver 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.

### 3.24.4 Conventions pour les définitions des services

#### 3.24.4.1 Généralités

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.

#### 3.24.4.2 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",
- 3) primitive "indication",
- 4) primitive "response", et
- 5) primitive "confirm".

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

- a) 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.
- b) 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.

#### 3.24.4.3 Procédures de service

Les procédures sont définies en termes

- d'interactions entre entités d'application via l'échange d'unités de données de protocole d'application de bus de terrain, et

- d'interactions entre un fournisseur de services de couche application et un utilisateur de service de couche application dans le même système via 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

Les concepts et modèles communs utilisés pour décrire le service de couche application dans la présente norme sont détaillés dans la CEI/TR 61158-1, Article 9.

## 5 Data type ASE

Les types de données tels que spécifiés dans la CEI/TR 61158-1, Article 9 s'appliquent avec les restrictions suivantes:

Seul le niveau 1 d'imbrication est pris en charge.

Seuls sont pris en charge les types de données de base suivants:

BitString8  
 BitString16  
 BitString32  
 BitString64  
 Unsigned16  
 Unsigned32  
 Unsigned64  
 Integer16  
 Integer32  
 Integer64  
 VisibleString1  
 Float32  
 Float64

### 5.1 Types Bitstring (chaîne de bits)

#### 5.1.1 BitString8

**CLASS:** Data type (Type de données)

**ATTRIBUTES:**

|     |                              |   |              |
|-----|------------------------------|---|--------------|
| 1   | Data type Numeric Identifier | = | 22           |
| 2   | Data type Name               | = | Bitstring8   |
| 3   | Format                       | = | FIXED LENGTH |
| 5.1 | Octet Length                 | = | 1            |

Ce type contient un élément de type BitString.

#### 5.1.2 BitString16

**CLASS:** Type de données

**ATTRIBUTES:**

|   |                              |   |             |
|---|------------------------------|---|-------------|
| 1 | Data type Numeric Identifier | = | 23          |
| 2 | Data type Name               | = | Bitstring16 |

|     |              |   |              |
|-----|--------------|---|--------------|
| 3   | Format       | = | FIXED LENGTH |
| 5.1 | Octet Length | = | 2            |

Ce type est un BitString16 et a une longueur de deux octets.

### 5.1.3 BitString32

**CLASS:** Type de données

**ATTRIBUTES:**

|     |                              |   |              |
|-----|------------------------------|---|--------------|
| 1   | Data type Numeric Identifier | = | 24           |
| 2   | Data type Name               | = | Bitstring32  |
| 3   | Format                       | = | FIXED LENGTH |
| 5.1 | Octet Length                 | = | 4            |

Ce type est un BitString16 et a une longueur de quatre octets.

### 5.1.4 BitString64

**CLASS:** Type de données

**ATTRIBUTES:**

|     |                              |   |              |
|-----|------------------------------|---|--------------|
| 1   | Data type Numeric Identifier | = | 57           |
| 2   | Data type Name               | = | Bitstring64  |
| 3   | Format                       | = | FIXED LENGTH |
| 5.1 | Octet Length                 | = | 8            |

Ce type est un BitString16 et a une longueur de huit octets.

## 5.2 Types Unsigned (non signés)

### 5.2.1 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 unsigned (non signé) a une longueur de deux octets.

### 5.2.2 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 unsigned (non signé) a une longueur de quatre octets.

### 5.2.3 Unsigned64

**CLASS:** Type de données

**ATTRIBUTES:**

|   |                              |   |            |
|---|------------------------------|---|------------|
| 1 | Data type Numeric Identifier | = | 56         |
| 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 unsigned (non signé) a une longueur de huit octets.

### 5.3 Types Integer (entiers)

#### 5.3.1 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.

#### 5.3.2 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.

#### 5.3.3 Integer64

**CLASS:** **Type de données**

**ATTRIBUTS:**

|     |                              |   |              |
|-----|------------------------------|---|--------------|
| 1   | Data type Numeric Identifier | = | 55           |
| 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.

### 5.4 Types Floating Point (virgule flottante)

#### 5.4.1 Float32

**CLASS:** **Type de données**

**ATTRIBUTS:**

|     |                              |   |              |
|-----|------------------------------|---|--------------|
| 1   | Data type Numeric Identifier | = | 8            |
| 2   | Data type Name               | = | Float32      |
| 4   | 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").

#### 5.4.2 Float64

**CLASS:** **Type de données**

#### ATTRIBUTES:

|     |                              |   |              |
|-----|------------------------------|---|--------------|
| 1   | Data type Numeric Identifier | = | 15           |
| 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").

### 5.5 Types Structure

#### 5.5.1 STRING2

**CLASS:** Type de données

#### ATTRIBUTES:

|       |                              |   |                         |
|-------|------------------------------|---|-------------------------|
| 1     | Data type Numeric Identifier | = | non utilisé             |
| 2     | Data type Name               | = | STRING2                 |
| 3     | Format                       | = | STRUCTURE               |
| 5.1   | Number of Fields             | = | 2                       |
| 5.2.1 | Field Name                   | = | Charcount_Element       |
| 5.2.2 | Field Data type              | = | UINT                    |
| 5.3.1 | Field Name                   | = | String2contents_Element |
| 5.3.2 | Field Data type              | = | OctetString             |

Cette extension de type de la CEI 61131-3 est constituée de deux éléments. Charcount\_Element donne le nombre courant de caractères dans String2contents\_Element (un UINT par caractère). Les caractères sont tels que spécifiés dans l'ISO 10646.

## 6 Spécification de modèle de communication

### 6.1 Concepts

#### 6.1.1 Mécanismes de communication

Deux mécanismes de communication sont pris en charge par des dispositifs au sein du réseau:

- émission cyclique de données d'une façon hautement efficace en utilisant le modèle éditeur-abonné,
- émission non cyclique de données en utilisant un modèle de communication client-serveur.

Les AREP, qui agissent comme éditeur "push" ou abonné "push", sont utilisés pour l'émission cyclique.

Les AREP, qui agissent comme un client ou un serveur, sont utilisés pour l'émission non cyclique de données.

#### 6.1.2 Concept d'IDN

Les données d'application qui sont émises de façon cyclique et non cyclique entre des utilisateurs de la FAL sont mises en correspondance sur lesdits numéros d'identification (les IDN). Ces IDN correspondent aux APO tels que définis et sont décrits dans la CEI 61158-3-16, Annexe A.

## 6.2 Les ASE

### 6.2.1 ASE numéro d'identification (IDN)

#### 6.2.1.1 Vue d'ensemble

L'ASE IDN donne l'accès en lecture et en écriture aux attributs des IDN fournis par un dispositif.

#### 6.2.1.2 Spécification de classe d'IDN

##### 6.2.1.2.1 Modèle formel

|               |     |                                                               |
|---------------|-----|---------------------------------------------------------------|
| FAL ASE:      |     | ASE IDN                                                       |
| CLASS:        |     | IDN                                                           |
| CLASS ID:     |     | non utilisé                                                   |
| PARENT CLASS: |     | TOP                                                           |
| ATTRIBUTS:    |     |                                                               |
| 1             | (m) | Attribut-clé: Identification                                  |
| 1.1           | (m) | Attribut-clé: Identification Number (Numéro d'identification) |
| 1.2           | (o) | Attribut-clé: Resource Element (Élément de ressource)         |
| 1.3           | (o) | Attribut-clé: Subindex (Sous-indexe)                          |
| 2             | (o) | Attribut: Name (Nom)                                          |
| 3             | (m) | Attribut: Data Attribute (Attribut de données)                |
| 4             | (o) | Attribut: Unit (Unité)                                        |
| 5             | (o) | Attribut: Minimum value (Valeur minimale)                     |
| 6             | (o) | Attribut: Maximum value (Valeur maximale)                     |
| 7             | (m) | Attribut: Operation Data (Données d'exploitation)             |
| SERVICES:     |     |                                                               |
| 1             | (m) | OpsService: Read (Lecture)                                    |
| 2             | (m) | OpsService: Write (Écriture)                                  |

##### 6.2.1.2.2 Attributs

###### Identification

Cet attribut-clé identifie une instance de cette classe d'objets. L'adressage peut être simple ou étendu.

###### Identification number

Cet attribut obligatoire est un identificateur numérique pour adresser une instance de cette classe d'objets. En cas d'adressage simple, il est le seul attribut requis pour l'identification.

###### Resource element

Cet attribut facultatif est utilisé dans le cas d'un adressage étendu.

###### Subindex

Cet attribut facultatif est utilisé dans le cas d'un adressage étendu.

###### Name

Cet attribut facultatif spécifie un nom symbolique pour cette classe d'objets.

###### Data attribute

Cet attribut obligatoire spécifie toutes les informations qui sont nécessaires pour afficher ou convertir intelligiblement des données. Il inclut le type de données, la longueur des données, le facteur de conversion, la permission de lecture/écriture en fonction de la phase de communication et selon que les données sont, oui ou non, associées à une commande.

###### Unit

Cet attribut facultatif spécifie l'unité des données d'exploitation.

### Minimum value

Cet attribut facultatif spécifie la valeur minimale d'entrée pour les données d'exploitation.

### Maximum value

Cet attribut facultatif spécifie la valeur maximale d'entrée pour les données d'exploitation.

### Operation data

Cet attribut obligatoire spécifie les données d'exploitation de cette instance de la classe d'objets.

## 6.2.1.3 Spécification des services d'ASE IDN

### 6.2.1.3.1 Services pris en charge

Le présent paragraphe spécifie la définition des services qui sont propres à cet ASE. Les services définis pour cet ASE sont:

- Read (Lecture)
- Write (Écriture)

### 6.2.1.3.2 Service "Read"

#### 6.2.1.3.2.1 Vue d'ensemble du service

Ce service confirmé sert à lire un élément d'un IDN à la demande.

#### 6.2.1.3.2.2 Primitives du service

Les paramètres du service pour chaque primitive sont montrés dans le Tableau 1.

**Tableau 1 – Paramètres du service "Read"**

| Nom de paramètre                                                                                                                                               | Req | Ind   | Rsp | Cnf   |
|----------------------------------------------------------------------------------------------------------------------------------------------------------------|-----|-------|-----|-------|
| Argument                                                                                                                                                       |     |       |     |       |
| AREP                                                                                                                                                           | M   | M     |     |       |
| Device Address                                                                                                                                                 | M   | M (=) |     |       |
| Identification                                                                                                                                                 | M   | M (=) |     |       |
| Attribute                                                                                                                                                      | M   | M (=) |     |       |
| Result(+)                                                                                                                                                      |     |       | S   | S (=) |
| Value                                                                                                                                                          |     |       | M   | M (=) |
| Result (-)                                                                                                                                                     |     |       | S   | S (=) |
| Error Info                                                                                                                                                     |     |       | M   | M (=) |
| NOTE La méthode par laquelle une primitive "confirm" est corrélée à sa primitive "request" précédente correspondante relève d'une initiative locale. Voir 1.2. |     |       |     |       |

### AREP

Ce paramètre est l'identificateur local de l'AR souhaitée.

### Device address

Ce paramètre identifie l'adresse du dispositif.

### Identification

Ce paramètre spécifie un objet IDN devant être lu par l'attribut-clé.

### Attribute

Ce paramètre spécifie l'élément d'un objet IDN devant être lu par l'attribut-clé.

**Value**

Ce paramètre spécifie la valeur lue.

**Error info**

Ce paramètre donne des informations d'erreur pour les erreurs de service.

**6.2.1.3.3 Service "Write"****6.2.1.3.3.1 Vue d'ensemble du service**

Ce service confirmé sert à écrire un élément d'un IDN.

**6.2.1.3.3.2 Primitives du service**

Les paramètres du service pour chaque primitive sont montrés dans le Tableau 2.

**Tableau 2 – Paramètres du service "Write"**

| Nom de paramètre                                                                                                                                               | Req | Ind   | Rsp | Cnf   |
|----------------------------------------------------------------------------------------------------------------------------------------------------------------|-----|-------|-----|-------|
| Argument                                                                                                                                                       |     |       |     |       |
| AREP                                                                                                                                                           | M   | M     |     |       |
| Device Address                                                                                                                                                 | M   | M (=) |     |       |
| Identification                                                                                                                                                 | M   | M (=) |     |       |
| Attribut                                                                                                                                                       | M   | M (=) |     |       |
| Value                                                                                                                                                          | M   | M (=) |     |       |
| Result(+)                                                                                                                                                      |     |       | S   | S (=) |
| Result (-)                                                                                                                                                     |     |       | S   | S (=) |
| Error Info                                                                                                                                                     |     |       | M   | M (=) |
| NOTE La méthode par laquelle une primitive "confirm" est corrélée à sa primitive "request" précédente correspondante relève d'une initiative locale. Voir 1.2. |     |       |     |       |

**AREP**

Ce paramètre est l'identificateur local de l'AR souhaitée.

**Device address**

Ce paramètre identifie l'adresse du dispositif.

**Identification**

Ce paramètre spécifie un objet IDN devant être écrit par l'attribut-clé.

**Attribute**

Ce paramètre spécifie l'élément d'un objet IDN devant être écrit par l'attribut-clé.

**Value**

Ce paramètre spécifie la valeur devant être écrite.

**Error info**

Ce paramètre donne des informations d'erreur pour les erreurs de service.

**6.2.2 ASE CYCIDN****6.2.2.1 Vue d'ensemble**

L'ASE CYCIDN donne l'accès en lecture et en écriture cyclique aux données d'exploitation des IDN fournis par un dispositif.

### 6.2.2.2 Spécification de la classe de numéros d'identification cyclique (CYCIDN «Cyclic Identification Number»)

#### 6.2.2.2.1 Modèle formel

|                       |                                         |
|-----------------------|-----------------------------------------|
| FAL ASE:              | CYCIDN ASE                              |
| CLASS:                | CYCIDN                                  |
| CLASS ID:             | non utilisé                             |
| PARENT CLASS:         | TOP                                     |
| ATTRIBUTES:           |                                         |
| 1 (m) Attribut-clé:   | Identification                          |
| 1.1 (m) Attribut-clé: | Identification Number                   |
| 1.2 (o) Attribut-clé: | Resource Element (Élément de ressource) |
| 1.3 (o) Attribut-clé: | Subindex (Sous-indice)                  |
| 2 (m) Attribut:       | Data (Données)                          |
| SERVICES:             |                                         |
| 1 (m) OpsService:     | Read (Lecture)                          |
| 2 (o) OpsService:     | Write (Écriture)                        |
| 3 (m) OpsService:     | Notify (c'est-à-dire: Notifier)         |

#### 6.2.2.2.2 Attributs

##### AREP

Ce paramètre est l'identificateur local de l'AR souhaitée.

##### Identification

Ce paramètre spécifie un objet IDN devant être lu par l'attribut-clé.

##### Data

Cet attribut spécifie les données qui sont lues ou écrites.

### 6.2.2.3 Spécification des services d'ASE CYCIDN

#### 6.2.2.3.1 Services pris en charge

Le présent paragraphe spécifie la définition des services qui sont propres à cet ASE. Les services définis pour cet ASE sont:

- Read (Lecture)
- Write (Écriture)
- Notify (c'est-à-dire: Notifier)

#### 6.2.2.3.2 Service "Read"

##### 6.2.2.3.2.1 Vue d'ensemble du service

Ce service non confirmé sert à lire de façon cyclique les données d'exploitation d'un IDN.

##### 6.2.2.3.2.2 Primitives du service

Les paramètres du service pour chaque primitive sont montrés dans le Tableau 3.

**Tableau 3 – Paramètres du service "Read"**

| Nom de paramètre | Req | Ind | Rsp | Cnf |
|------------------|-----|-----|-----|-----|
| Argument         |     |     |     |     |
| AREP             | M   |     |     |     |
| Device Address   | M   |     |     |     |
| Identification   | M   |     |     |     |
| Operation Data   | M   |     |     |     |
| Error Info       | M   |     |     |     |

**AREP**

Ce paramètre est l'identificateur local de l'AR souhaitée.

**Device address**

Ce paramètre identifie l'adresse du dispositif.

**Identification**

Ce paramètre spécifie un objet IDN devant être lu par l'attribut-clé.

**Operation data**

Ce paramètre spécifie les données d'exploitation qui sont en cours de lecture.

**Error info**

Ce paramètre donne des informations d'erreur pour les erreurs de service.

**6.2.2.3.3 Service "Write"****6.2.2.3.3.1 Vue d'ensemble du service**

Ce service confirmé sert à écrire de manière cyclique un élément d'un IDN.

**6.2.2.3.3.2 Primitives du service**

Les paramètres du service pour chaque primitive sont montrés dans le Tableau 4.

**Tableau 4 – Paramètres du service "Write"**

| Nom de paramètre                                                                                                                                               | Req | Ind | Cnf |
|----------------------------------------------------------------------------------------------------------------------------------------------------------------|-----|-----|-----|
| Argument                                                                                                                                                       |     |     |     |
| AREP                                                                                                                                                           | M   |     |     |
| Device Address                                                                                                                                                 | M   |     |     |
| Identification                                                                                                                                                 | M   |     |     |
| Operation Data                                                                                                                                                 | M   |     |     |
| NOTE La méthode par laquelle une primitive "confirm" est corrélée à sa primitive "request" précédente correspondante relève d'une initiative locale. Voir 1.2. |     |     |     |

**AREP**

Ce paramètre est l'identificateur local de l'AR souhaitée.

**Device address**

Ce paramètre identifie l'adresse du dispositif.

**Identification**

Ce paramètre spécifie un objet IDN devant être écrit par l'attribut-clé.

**Operation data**

Ce paramètre spécifie les données d'exploitation qui sont en cours d'écriture.