

INTERNATIONAL STANDARD

NORME INTERNATIONALE



**Industrial communication networks – Fieldbus specifications –
Part 6-20: Application layer protocol specification – Type 20 elements**

**Réseaux de communication industriels – Spécifications des bus de terrain –
Partie 6-20: Spécification du protocole de la couche application – Eléments
de type 20**



THIS PUBLICATION IS COPYRIGHT PROTECTED

Copyright © 2014 IEC, Geneva, Switzerland

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

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

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

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

About the IEC

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

About IEC publications

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

IEC Catalogue - webstore.iec.ch/catalogue

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

IEC publications search - www.iec.ch/searchpub

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

IEC Just Published - webstore.iec.ch/justpublished

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

Electropedia - www.electropedia.org

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

IEC Glossary - std.iec.ch/glossary

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

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

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

A propos de l'IEC

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

A propos des publications IEC

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

Catalogue IEC - webstore.iec.ch/catalogue

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

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

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

IEC Just Published - webstore.iec.ch/justpublished

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

Electropedia - www.electropedia.org

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

Glossaire IEC - std.iec.ch/glossary

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

Service Clients - webstore.iec.ch/csc

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

INTERNATIONAL STANDARD

NORME INTERNATIONALE



**Industrial communication networks – Fieldbus specifications –
Part 6-20: Application layer protocol specification – Type 20 elements**

**Réseaux de communication industriels – Spécifications des bus de terrain –
Partie 6-20: Spécification du protocole de la couche application – Eléments
de type 20**

INTERNATIONAL
ELECTROTECHNICAL
COMMISSION

COMMISSION
ELECTROTECHNIQUE
INTERNATIONALE

PRICE CODE
CODE PRIX

XC

ICS 25.040.40; 35.100.70; 35.110

ISBN 978-2-8322-1766-5

**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	7
INTRODUCTION	9
1 Scope	10
2 Normative references	10
3 Terms, definitions, symbols, abbreviations and conventions	11
3.1 Terms and definitions from other ISO/IEC standards	11
3.2 IEC 61158-1 terms	12
3.3 Type 20 fieldbus application-layer specific definitions	14
3.4 Abbreviations and symbols	17
3.5 Conventions	18
3.6 Conventions used in state machines	18
4 Abstract syntax	20
5 Transfer syntax	20
5.1 Common APDU fields	20
5.2 Common APDU structure	22
5.3 Device application service-specific APDU structures	24
5.4 Data coding rules	51
6 Common procedures	56
6.1 Delayed response	56
6.2 Publish mode procedure	56
7 FAL protocol state machines	56
7.1 General	56
7.2 AREP mapping to data link layer	57
7.3 Client ARPM	58
7.4 Server ARPM	60
7.5 Functions used by FAL state machines	61
Annex A (normative) Application process status	63
A.1 General	63
A.2 Device malfunction	63
A.3 Configuration changed	63
A.4 Cold start	64
A.5 More status available	64
A.6 Loop current fixed	64
A.7 Loop current saturated	64
A.8 Non-primary variable out of limits	65
A.9 Primary variable out of limits	65
Annex B (normative) Device and dynamic variable	66
B.1 Device variable	66
B.2 Dynamic variable	66
B.3 Primary variable	67
B.4 Device variable classification	68
B.5 Device families	68
B.6 Device variable status	68
Annex C (normative) Common tables	70
C.1 Overview	70

C.1.1	General	70
C.1.2	Enumeration	70
C.1.3	Bit Field	70
C.2	Table definitions	70
C.2.1	Publish mode control codes	70
C.2.2	Write device variable codes	70
C.2.3	Device variable family codes	70
C.2.4	Device variable classification codes	70
C.2.5	Analog channel saturated codes	70
C.2.6	Analog channel fixed codes	70
C.2.7	Standardized status 0 codes	71
C.2.8	Standardized status 1 codes	71
C.2.9	Standardized status 2 codes	71
C.2.10	Standardized status 3 codes	71
C.2.11	Publish trigger mode codes	71
C.2.12	Transfer function codes	71
C.2.13	Alarm Selection Codes	71
C.2.14	Write Protect Codes	72
C.2.15	Physical layer signalling codes	72
C.2.16	Flag Assignment codes	72
C.2.17	Loop current mode codes	73
C.2.18	Trim point codes	73
C.2.19	Analog channel flag codes	73
C.2.20	Device variable codes	73
C.2.21	Device profile codes	74
Annex D (normative)	Command requirements	75
D.1	General	75
D.2	Stateless request and response	75
D.3	Read command	75
D.4	Write command	75
D.5	Action command	75
D.6	Indexed command	76
D.7	Multi-transaction command	76
Bibliography		77
Figure 1	Request APDU	22
Figure 2	Normal response APDU	22
Figure 3	Command error response from slave to master	23
Figure 4	Aggregated command APDU	24
Figure 5	Coding without identification	51
Figure 6	Coding of Integer type data	51
Figure 7	Coding of Integer16 type data	52
Figure 8	Coding of Unsigned type data	52
Figure 9	Coding of Unsigned16 type data	52
Figure 10	Coding of single precision Floating Point type data	52
Figure 11	Coding of double precision Floating Point type data	53
Figure 12	Coding of Date type data	53

Figure 13 – Client state machine	58
Figure 14 – Server state machine	60
Figure A.1 – Loop current saturation and alarm levels	65
Figure B.1 – Device and Dynamic variables	66
Figure B.2 – Primary variable domains	67
Figure B.3 – Device variable status	69
Table 1 – Conventions used for state machines	19
Table 2 – Response code values	20
Table 3 – Application process status values	21
Table 4 – Extended status values	21
Table 5 – Identify request APDU	24
Table 6 – Identify response value field	25
Table 7 – Identify command specific response codes	25
Table 8 – Read primary variable response value field	26
Table 9 – Read primary variable command specific response codes	26
Table 10 – Read loop current and percent of range value field	26
Table 11 – Read loop current and percent of range command specific response codes	27
Table 12 – Read dynamic variables and loop current value field	27
Table 13 – Read dynamic variables and loop current command specific response codes	27
Table 14 – Write loop configuration value field	28
Table 15 – Write loop configuration command specific response codes	28
Table 16 – Read loop configuration value field	29
Table 17 – Read loop configuration command specific response codes	29
Table 18 – Read dynamic variable families classifications value field	29
Table 19 – Read dynamic variable families classifications command specific response codes	29
Table 20 – Read device variables with status request value field	30
Table 21 – Read device variables with status value field	30
Table 22 – Read device variables with status command specific response codes	31
Table 23 – Read message response value field	32
Table 24 – Read message command specific response codes	32
Table 25 – Read tag, descriptor, date response value field	33
Table 26 – Read tag, descriptor, date command specific response codes	33
Table 27 – Read primary variable transducer information response value field	33
Table 28 – Read primary variable transducer information command specific response codes	34
Table 29 – Read device information response value field	34
Table 30 – Read device information command specific response codes	35
Table 31 – Read final assembly number response value field	35
Table 32 – Read final assembly number command specific response codes	35
Table 33 – Write message value field	35
Table 34 – Write message command specific response codes	36

Table 35 – Write tag, descriptor, date value field	36
Table 36 – Write tag, descriptor, date command specific response codes	36
Table 37 – Write final assembly number value field	37
Table 38 – Write final assembly number command specific response codes	37
Table 39 – Read long tag response value field	37
Table 40 – Read long tag command-specific response codes	37
Table 41 – Write long tag value field	38
Table 42 – Write long tag command specific Response codes	38
Table 43 – Write primary variable range value field	39
Table 44 – Write primary variable range command specific response codes	39
Table 45 – Enter-exit fixed current mode request value field	40
Table 46 – Enter-exit fixed current mode response value field	40
Table 47 – Enter-exit fixed current mode command specific response codes	40
Table 48 – Write primary variable unit value field	41
Table 49 – Write primary variable unit command specific response codes	41
Table 50 – Trim loop current zero request value field	41
Table 51 – Trim loop current zero command specific response codes	41
Table 52 – Trim loop current gain request value field	42
Table 53 – Trim loop current gain command specific response codes	42
Table 54 – Read dynamic variable assignment response value field	43
Table 55 – Read dynamic variable assignment command specific response codes	43
Table 56 – Write dynamic variable assignment value field	44
Table 57 – Write dynamic variable assignment command specific response codes	44
Table 58 – Write number of response preambles value field	45
Table 59 – Write number of response preambles command specific response codes	45
Table 60 – Read device variable trim points request value field	45
Table 61 – Read device variable trim points response value field	46
Table 62 – Read device variable trim points command specific response codes	46
Table 63 – Read device variable trim guidelines request value field	46
Table 64 – Read device variable trim guidelines response value field	46
Table 65 – Read device variable trim points command specific response codes	47
Table 66 – Write device variable trim point value field	47
Table 67 – Write device variable trim point command specific response codes	48
Table 68 – Reset device variable trim value field	48
Table 69 – Reset device variable trim command specific response codes	49
Table 70 – Aggregated command specific response codes	50
Table 71 – Coding for Date type	53
Table 72 – Coding for one octet Enumerated Type	54
Table 73 – One octet bit field	54
Table 74 – Packed ASCII character set	55
Table 75 – Acceptable subset of ISO Latin-1 characters	56
Table 76 – Client machine state transitions	59
Table 77 – Server machine state transitions	61

Table 78 – Function FormReqApdu	61
Table 79 – Function Command	61
Table 80 – Function CommErr.....	61
Table 81 – Function RespCode	62
Table 82 – Function Commcode.....	62
Table 83 – Function ApStatus	62
Table 84 – Function Value	62
Table A.1 – Commands that cause configuration change	63
Table C.1 – Transfer function codes	71
Table C.2 – Alarm Selection codes	72
Table C.3 – Write Protect codes	72
Table C.4 – Physical layer signalling codes	72
Table C.5 – Flag Assignment codes	73
Table C.6 – Loop current mode codes.....	73
Table C.7 – Trim point codes	73
Table C.8 – Analog channel flag codes	73
Table C.9 – Device variable codes	74

IECNORM.COM : Click to view the full PDF of IEC 61158-6-20:2014

INTERNATIONAL ELECTROTECHNICAL COMMISSION

**INDUSTRIAL COMMUNICATION NETWORKS –
FIELDBUS SPECIFICATIONS –****Part 6-20: Application layer protocol specification –
Type 20 elements**

FOREWORD

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

Attention is drawn to the fact that the use of the associated protocol type is restricted by its intellectual-property-right holders. In all cases, the commitment to limited release of intellectual-property-rights made by the holders of those rights permits a layer protocol type to be used with other layer protocols of the same type, or in other type combinations explicitly authorized by its intellectual-property-right holders.

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

International Standard IEC 61158-6-20 has been prepared by subcommittee 65C: Industrial networks, of IEC technical committee 65: Industrial-process measurement, control and automation.

This third edition cancels and replaces the second edition published in 2010. This edition constitutes a technical revision.

The main changes with respect to the previous edition are listed below:

- a) added protocol for new services that are added to IEC 61158-5-20;
- b) added normative annexes;
- c) updated then references, terms, definitions, symbols, abbreviations;
- d) corrected the editorial errors and the text.

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

FDIS	Report on voting
65C/764/FDIS	65C/774/RVD

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

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

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

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

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

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

INTRODUCTION

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

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

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

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

IECNORM.COM : Click to view the full PDF of IEC 61158-6-20:2014

INDUSTRIAL COMMUNICATION NETWORKS – FIELDBUS SPECIFICATIONS –

Part 6-20: Application layer protocol specification – Type 20 elements

1 Scope

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 20 fieldbus. The term “time-critical” is used to represent the presence of a time-window, within which one or more specified actions are required to be completed with some defined level of certainty. Failure to complete specified actions within the time window risks failure of the applications requesting the actions, with attendant risk to equipment, plant and possibly human life.

This standard defines in an abstract way the externally visible behavior provided by the Type 20 of the fieldbus Application Layer in terms of

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

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

- a) the wire-representation of the service primitives defined in IEC 61158-5-20, and
- b) the externally visible behavior associated with their transfer.

This standard specifies the protocol of the Type 20 IEC fieldbus application layer, in conformance with the OSI Basic Reference Model (ISO/IEC 7498-1) and the OSI Application Layer Structure (ISO/IEC 9545).

2 Normative references

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

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

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

IEC 61158-5-20:2014, *Industrial communication networks – Fieldbus specifications – Part 5-20: Application layer service definition – Type 20 elements*

IEC 62591:2010, *Industrial communication networks – Wireless communication network and communication profiles – WirelessHART™*

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

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

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

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

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

IEEE 802.15.4: *IEEE Standard for Local and metropolitan area networks – Part 15.4: Low-Rate Wireless Personal Area Networks (LR-WPANs)*

3 Terms, definitions, symbols, abbreviations and conventions

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

3.1 Terms and definitions from other ISO/IEC standards

3.1.1 Terms and definitions from ISO/IEC 7498-1

- a) abstract syntax
- b) application entity
- c) application process
- d) application protocol data unit
- e) application service element
- f) application entity invocation
- g) application process invocation
- h) transfer syntax

3.1.2 Terms and definitions from ISO/IEC 9545

- a) application-entity-invocation
- b) application-process-invocation
- c) application-service-element

3.1.3 Terms and definitions from ISO/IEC 8824-1

- a) object identifier
- b) type
- c) value
- d) simple type

- e) structured type
- f) component type
- g) tag
- h) true
- i) false
- j) integer type
- k) octet string type
- m) null type

3.1.4 Terms and definitions from ISO/IEC 8825-1

- a) encoding (of a data value)
- b) data value
- c) identifier octets (the singular form is used in this standard)
- d) length octet(s) (both singular and plural forms are used in this standard)
- e) contents octets

3.2 IEC 61158-1 terms

3.2.1

application

function or data structure for which data is consumed or produced

3.2.2

application object

object class that manages and provides the run time exchange of messages across the network and within the network device

Note 1 to entry: Multiple types of application object classes may be defined.

3.2.3

application process

part of a distributed application on a network, which is located on one device and unambiguously addressed

3.2.4

application process object

component of an application process that is identifiable and accessible through an FAL application relationship

Note 1 to entry: Application process object definitions are composed of a set of values for the attributes of their class (see the definition for Application Process Object Class Definition). Application process object definitions may be accessed remotely using the services of the FAL Object Management ASE. FAL Object Management services can be used to load or update object definitions, to read object definitions, and to dynamically create and delete application objects and their corresponding definitions.

3.2.5

application process object class

class of application process objects defined in terms of the set of their network-accessible attributes and services

3.2.6

application relationship

cooperative association between two or more application-entity-invocations for the purpose of exchange of information and coordination of their joint operation

Note 1 to entry: This relationship is activated either by the exchange of application-protocol-data-units or as a result of pre-configuration activities.

3.2.7**application relationship endpoint**

context and behavior of an application relationship as seen and maintained by one of the application processes involved in the application relationship

Note 1 to entry: Each application process involved in the application relationship maintains its own application relationship endpoint.

3.2.8**attribute**

description of an externally visible characteristic or feature of an object

Note 1 to entry: The attributes of an object contain information about variable portions of an object. Typically, they provide status information or govern the operation of an object. Attributes may also affect the behaviour of an object. Attributes are divided into class attributes and instance attributes.

3.2.9**behaviour**

indication of how the object responds to particular events

Note 1 to entry: Its description includes the relationship between attribute values and services.

3.2.10**class**

set of objects, all of which represent the same kind of system component

Note 1 to entry: A class is a generalisation of the object; a template for defining variables and methods. All objects in a class are identical in form and behaviour, but usually contain different data in their attributes.

3.2.11**class attributes**

attribute that is shared by all objects within the same class

3.2.12**class code**

unique identifier assigned to each object class

3.2.13**class specific service**

service defined by a particular object class to perform a required function which is not performed by a common service

Note 1 to entry: A class specific object is unique to the object class which defines it.

3.2.14**client**

- (a) an object which uses the services of another (server) object to perform a task
- (b) an initiator of a message to which a server reacts, such as the role of an AR endpoint in which it issues confirmed service request APDUs to a single AR endpoint acting as a server

3.2.15**comm error**

detectable error in receiving a PhPDU or DLPDU, also 'Communication error code' octet of APDU

3.2.16**conveyance path**

unidirectional flow of APDUs across an application relationship

3.2.17

cyclic

term used to describe events which repeat in a regular and repetitive manner

3.2.18

endpoint

one of the communicating entities involved in a connection

3.2.19

error

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

3.2.20

error code

identification of a specific type of error within an error class

3.2.21

management information

network-accessible information that supports managing the operation of the fieldbus system, including the application layer

Note 1 to entry: Managing includes functions such as controlling, monitoring, and diagnosing.

3.2.22

server

- (a) role of an AREP in which it returns a confirmed service response APDU to the client that initiated the request
- (b) an object which provides services to another (client) object

3.2.23

service

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

Note 1 to entry: A set of common services is defined and provisions for the definition of object-specific services are provided. Object-specific services are those which are defined by a particular object class to perform a required function which is not performed by a common service.

3.3 Type 20 fieldbus application-layer specific definitions

3.3.1

analog channel

continuously varying electrical signal connecting a field device to the remainder of the data acquisition or control system

Note 1 to entry: Some field devices support multiple analog channels (input or output) Each analog channel transmits a single dynamic variable to or from the field device.

3.3.2

broadcast

process of sending a PDU to all devices that are connected to the network and are able to receive the transmission

3.3.3

broadcast address

address used by a master to send a command to all devices

3.3.4**burst mode**

initiation of communication activity by a slave device at cyclic interval without request from a master

3.3.5**busy**

state of a device which is active on different tasks and cannot execute a command at the time

3.3.6**comm error**

detectable error in receiving a PhPDU or DLPDU, also 'Communication error code' octet of APDU

3.3.7**device**

any entity containing an implementation of Type 20 fieldbus

3.3.8**device ID**

serial number for a device that is unique among all instances of one type of device

Note 1 to entry: The manufacturer is required to assigned unique value for every device that has the identical values for Manufacturer ID and Device Type.

3.3.9**device type**

manufacturer's type of a device, e.g. its product name

Note 1 to entry: The value of this attribute is unique among all manufacturers and all type of devices. Its value specifies the set of commands and data objects supported by the device.

3.3.10**device variable**

uniquely defined data item within a Field Device that is always associated with the process-related information

Note 1 to entry: A device variable's value varies in response to changes and variations in the process to which the device is connected.

3.3.11**dynamic variable**

a device variable that is assigned as the dynamic variable and possibly associated with an analog channel

Note 1 to entry: A device may contain up to four variables – primary, secondary, tertiary, and quaternary variables. These are collectively called the dynamic variables.

3.3.12**expanded device type**

the manufacturer's type of the device

Note 1 to entry: The value of this attribute is unique among all manufacturers and all type of devices. Its value specifies the set of commands and data objects supported by the device.

3.3.13**field device**

physical entity that is connected to the process or to plant equipment and has at least one signalling element that communicates with other signalling element(s) via the network

Note 1 to entry: It directly connects to the sensor or actuator or performs process control function and it is directly connected to the physical layer specified in this standard. It may generate or receive an analog signal in addition to a digital signal.

3.3.14

long tag

32-character restricted ISO Latin-1 string used to identify a field device

3.3.15

loop current

value measured by a milli-ammeter in series with the field device

Note 1 to entry: The loop current is a near DC analog 4-20 mA signal used to communicate a single value between the control system and the field device. Voltage mode devices use "Volts DC" as their engineering units where "loop current" values are used.

3.3.16

manufacturer ID

string identifying the manufacturer that produced the device

Note 1 to entry: A manufacturer is required to use the value assigned to it and is not permitted to use the value assigned to another manufacturer.

3.3.17

master

device that initiates communication activity by sending request PDUs to another device and expecting a response frame from that device

3.3.18

network

a single pair of cable, connectors, associated signaling elements by which a given set of signaling devices are interconnected and non-signaling elements that are attached to the same pair of cable

Note 1 to entry: An installation using multiple-pair wire and a common network power supply is considered as multiple networks.

3.3.19

polling address

identifier assigned to a device such that it is unique within the network to which the device is connected

Note 1 to entry: The polling address is used to construct a one-octet address.

3.3.20

primary master

master device that can always initiate the communication

3.3.21

secondary master

master device that can initiate the communication only through an arbitration process and when primary master has relinquished the initiation of the communication

3.3.22

slave

device that initiates communication activity only after it receives a request PDU from a master device, and that is required to send a response to that request

3.3.23

tag

8-character ASCII string used to identify a field device

3.3.24**transaction**

exchange of related, consecutive frames between two peer medium access control entities, required for a successful transmission

Note 1 to entry: A transaction consists of either (a) a single PhPDU transmission from a source device, or (b) one PhPDU from the source device followed by a second, link-level acknowledgement PhPDU from the destination device.

3.3.25**unique ID**

identifier assigned to a device which is unique among all instances of the devices compliant to this standard

3.4 Abbreviations and symbols

AE	Application entity
AL	Application layer
AP	Application process
APDU	Application protocol data unit
APO	Application object
AR	Application relationship
AREP	Application relationship endpoint
ARPM	Application Relationship Protocol Machine
ASCII	American Standard Code for Information Interchange
ASE	Application Service Element
Cnf	Confirmation
DL-	(as a prefix) data-link-
DLC	Data-link connection
DLE	DL entity (the local active instance of the data-link layer)
DLL	Data-link layer
DLM	Data-link-management
DLME	Data-link-management entity
DLPPDU	Data link protocol data unit
DLSAP	Data-link service access point
DLSDU	DL-service-data-unit
DR	Delayed response
DRM	Delayed response mechanism
FAL	Fieldbus application layer
HCF	HART™ Communication Foundation
ID	Identifier
IEC	International Electrotechnical Commission
Ind	Indication
LRV	Low range value
LSO	Least significant octet
MSO	Most significant octet

OSI	Open Systems Interconnect
PDU	Protocol data unit
PhL-	Physical layer (as a prefix)
PhPDU	PhL-protocol-data-unit
PV	Primary variable
QV	Quaternary variable
Req	Request
Rsp	Response
SAP	Service access point
SDU	Service data unit
SN	Sign bit
SOP	Standard Operating Procedure
SV	Secondary variable
TV	Tertiary variable
URV	Upper range value
VFD	Virtual field device

3.5 Conventions

3.5.1 General concept

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 three parts: its class definitions, its services, and its protocol specification. The first two are contained in IEC 61158-5-20. The protocol specification for each of the ASEs is defined in this standard.

The class definitions define the attributes of the classes supported by each ASE. The service specification defines the services that are provided by the ASE.

3.5.2 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 defined in IEC 61158-5-20, 3.5.2.

3.5.3 Abstract syntax conventions

When the "optionalParametersMap" parameter is used, a bit number which corresponds to each OPTIONAL or DEFAULT production is given as a comment.

3.6 Conventions used in state machines

3.6.1 State machine diagram

This standard uses a diagram to represent a state machine. The following convention is used.

- The labeled rectangles represent states that a component can be in.
- State transitions are directed lines. They show which state a component leaves and which state it transitions to.
- A transition is labeled with the events that caused it and the corresponding actions (this may be empty). The events (above the line) causing the transition is separated from the resulting actions (below the line) by a horizontal line.
- A tilde (" ~ ") means the one's complement or negation.

3.6.2 State machine table

The state machines are described in Table 1.

Table 1 – Conventions used for state machines

#	Current state	Event / condition => action	Next state
Name of this transition	The current state to which this state transition applies	Events or conditions that trigger this state transaction. => The actions that are taken when the above events or conditions are met. The actions are always indented below events or conditions	The next state after the actions in this transition is taken

NOTE The name for the transition is not included in all state machine specifications.

The conventions used in the state machines are as follows:

:= Value of an item on the left is replaced by value of an item on the right. If an item on the right is a parameter, it comes from the primitive shown as an input event.

xxx A parameter name.

Example:

Identifier := reason

means value of a 'reason' parameter is assigned to a parameter called 'Identifier.'

"xxx" Indicates fixed value.

Example:

Identifier := "abc"

means value "abc" is assigned to a parameter named 'Identifier.'

= A logical condition to indicate an item on the left is equal to an item on the right.

< A logical condition to indicate an item on the left is less than the item on the right.

> A logical condition to indicate an item on the left is greater than the item on the right.

<> A logical condition to indicate an item on the left is not equal to an item on the right.

&& Logical "AND"

|| Logical "OR"

This construct allows the execution of a sequence of actions in a loop within one transition. The loop is executed for all values from start_value to end_value.

Example:

```
for (Identifier := start_value to end_value)
  actions
endfor
```

This construct allows the execution of alternative actions depending on some condition (which might be the value of some identifier or the outcome of a previous action) within one transition.

Example:

```
If (condition)
  actions
else
  actions
endif
```

4 Abstract syntax

The abstract syntax of APDUs is combined with their transfer syntax and is specified in Clause 5.

5 Transfer syntax

5.1 Common APDU fields

5.1.1 Command number

It is command number that is used to represent the Numeric identifier of Read, Write or Information report service. It is Unsigned8 field and it identifies the type of service and the object in the Slave that has to be accessed or action that has to be performed.

5.1.2 Extended command number

It is command number that is used to extend the Numeric Identifier of Read, Write or Information report service to Unsigned16 value. If the extended command number is used then command number field shall be set to '31'. It is Unsigned16 field and it identifies the type of service and the object in the Slave that has to be accessed or action that has to be performed.

5.1.3 Octet count

This field represents the total number of octets in all of the fields that follow this field. The data type of this field is Unsigned8 and its value can be '0' to '255'. If the value of this field is zero, then this is the last field in the APDU.

5.1.4 Response code

Response code field indicates either the successful completion of the requested command, or completion of the requested command with a warning or an error in performing the requested command, as shown in Table 2. Some of the values depend upon the value of Command field. Each service-specific APDU structure shows the response code values assigned to that service. A device shall use values specified in the service-specific APDU. The device can respond with an unused value as long it indicates a condition that is not defined in this standard. It is one octet Enumeration data type field.

Table 2 – Response code values

Value	Class	Definition
0	Success	Command (read or Write) was executed properly
1 – 127	Warning	Command (Write) was executed with the deviation as described in response (for example a value was set to the nearest allowed value)
1 – 127	Error	Command execution was not properly completed and the Response code indicates the reason (for example the device is in Write Protect mode)
1 – 127	Reserved	A code may be reserved for future use by this standard protocol. A device is not permitted to return reserved value

5.1.5 Value

The Value field is the user data which is transferred between application layer and its user. The ALE assembles it from the parameters of a service primitive or parses it into parameters of a service primitive. Its structure depends upon the type of service-specific APDU.

5.1.6 Application process status

This field is provided by the FAL user. This field is one octet Bit Field data type. Its value should be set by the FAL user as shown in Table 3, so that all user applications can interoperate with field devices compliant to this standard.

NOTE Application process status is also known as Device status, because it indicates the current operating status of the field device as a whole and is not associated with the completion of any command.

Table 3 – Application process status values

Bit mask	Name	Description
0x80	Device malfunction	The device detected a serious error or failure that compromises device operation.
0x40	Configuration changed	An operation was performed that changed the device's configuration.
0x20	Cold start	A power failure or device reset has occurred.
0x10	More status available	More status information is available via another Read service.
0x08	Loop current fixed	The Loop current or signalling voltage (for a device using voltage output) is being held at a fixed value and is not responding to process variations.
0x04	Loop current saturated	The Loop current or signalling voltage (for a device using voltage output) has reached its upper (or lower) endpoint limit and cannot increase (or decrease) any further.
0x02	Non-primary variable out of limits	A Device variable that is not mapped to the Primary variable is beyond its operating limits.
0x01	Primary variable out of limits	The Primary variable is beyond its operating limits.

5.1.7 Extended status

This field indicates additional status of the Application process. The possible values of this field are shown in Table 4. Its data type is one octet Bit Field. This field is used as part of the FAL user data or as part of the header in some of the APDUs.

Table 4 – Extended status values

Code	Value	Description
0x01	Maintenance required	The device requires maintenance, although it has not malfunctioned.
0x02	Device variable alert	One or more of the 'Device variables' are in an Alarm or Warning state.
0x04	Critical power failure	The available power is becoming critically low; used by devices that operate from stored power.

NOTE Extended status is also known as Extended device status, because it indicates the current operating status of the field device as a whole and is not associated with the completion of any command.

5.1.8 Formats of APDU structures

The sending application layer prepares an APDU to transfer it to the receiving application layer. It uses the parameters from the service primitives to do so. There are several formats of the APDU:

- request APDU from Master to Slave device using one octet command number,
- request APDU from Master to Slave device using extended command number,

- normal response from Slave to Master device using one octet command number,
- normal response from Slave to Master device using extended command number,
- command error response from Slave to Master device using one octet command number, and
- command error response from Slave to Master device using extended command number.

The common format and coding rules for all APDU structures are specified in 5.2. The formats for service-specific APDU structures are specified in 5.3. The command number field is included in the request and response both. Therefore, the requester does not have to store the request APDU to associate it with the response APDU.

5.2 Common APDU structure

5.2.1 Request APDU

The request APDU shall have a structure as shown in Figure 1. The Figure 1 (a) shows the APDU for Command number values that are one octet long and its value is not thirty-one (31). The Figure 1 (b) shows the APDU for Command values that are two octets long, also called extended command number.

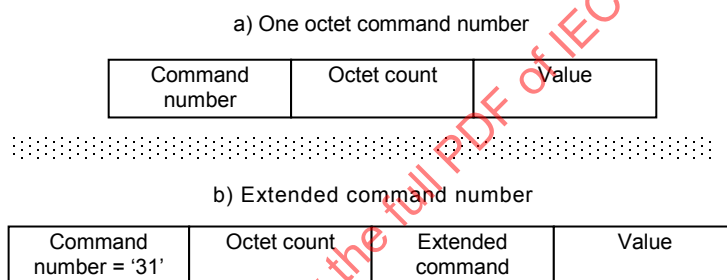


Figure 1 – Request APDU

For a Read service, Value field shall be either Null or a Variable Sub Index. For a Write service, Value field shall be Variable Sub Index, if required followed by the value that has to be written to the Slave objects identified by the Command. The ALE assembles it from the parameters of a service primitive. Its structure depends upon the service-specific APDU.

5.2.2 Normal response APDU

Figure 2 shows the structure of the application layer fields that are used for a response from Slave to Master without any error in executing the requested command. The Figure 2 (a) shows the APDU for Command number values that are one octet long and its value is not thirty-one (31). The Figure 2 (b) shows the APDU for Command values that are two octets long, also called extended command number.

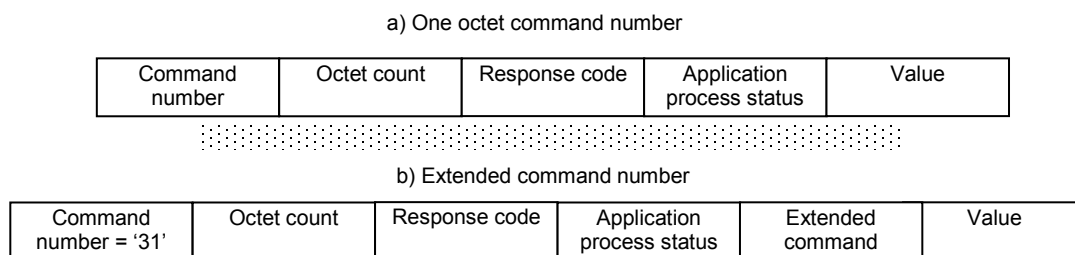


Figure 2 – Normal response APDU

The Slave sends this APDU in response to a request from Master. The value of Command field in this APDU is identical to the Command field and Extended command field (if present) in the corresponding request APDU. The minimum value of Octet count field is two '2', because Response code and Application process status fields are always included.

If the response is for a Read request, then Value field is used for the value of the objects that are read from Slave device. If the response is for a Write request, then this field is used for the value of the objects that were written in the Slave device. In the response to the Write request, it is same as the value received in the request, unless the value received was not permitted and the server wrote a different value. In that case, it is the value written in the Slave device.

5.2.3 Command error response APDU

Figure 3 shows the structure of the application layer fields that are used for a response from Slave to Master, when there is an error in executing the command. The Figure 3 (a) shows the APDU for Command number values that are one octet long and its value is not thirty-one (31). The Figure 3 (b) shows the APDU for Command values that are two octets long, also called extended command number. The Value field does not exist. The value of Octet count field shall be '2' for Command number values that are one octet long and it shall be '4' for extended command number.

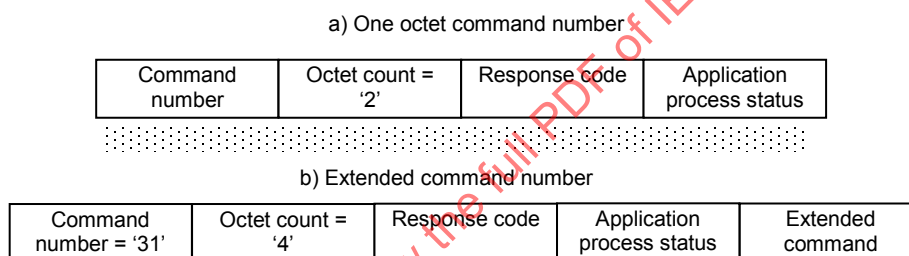


Figure 3 – Command error response from slave to master

The responding device shall send this APDU in response to a request from Master. The value of Command field in this APDU shall be identical to the Command number field in the corresponding request APDU. The Application process status shall be the current status.

5.2.4 Information report APDU

A slave device can publish data without requiring a request from Master for every publishing. It shall use the information report service for this purpose using the response APDU specified in 5.2.2 or command aggregation APDU specified in 5.2.5. The publishing is setup to use one of the command numbers (see IEC 62591:2010, 8.2.3.32).

5.2.5 Aggregated command APDU

This standard allows multiple commands and associated data to be transported in a single request and response APDU. The command aggregation is especially useful when reading device configurations.

The request APDU format shall be as shown in Figure 1 (a) with Command number equal to '78'. The octet count shall be the length of the value field as shown in Figure 4 (a). The Request count is equal to the number of individual requests in the value field. The format of individual request is shown in Figure 4 (b). The Command number field in each command request shall be two octets long; the Octet count shall be the length of the individual request value field.

The response APDU format shall be as shown in Figure 2 (a) with Command number equal to '78'. The octet count shall be the length of the value field as shown in Figure 4 (c). The

Response count is equal to the number of individual responses in the value field. The format of individual response is shown in Figure 4 (d). The Command number field in each command response shall be two octets long; the Octet count shall be the length of the individual response code and value fields combined.

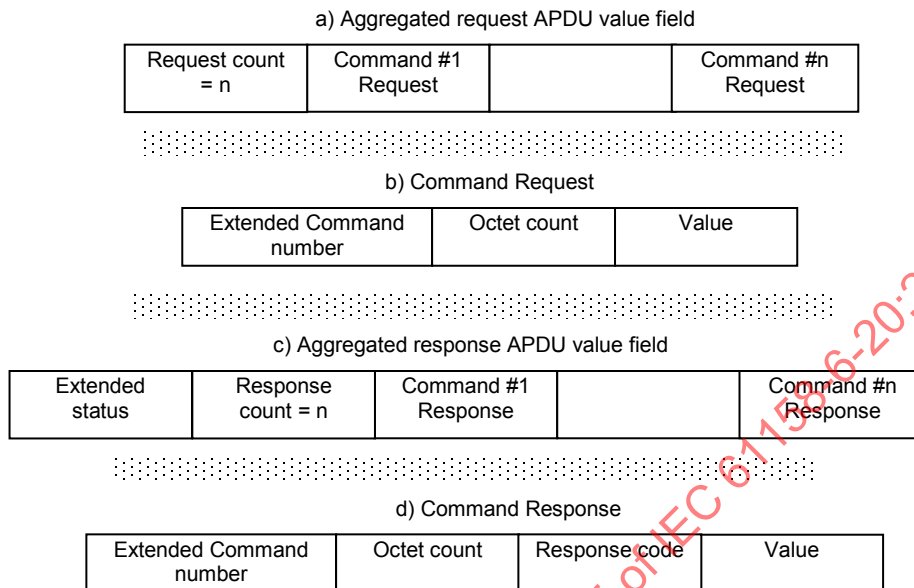


Figure 4 – Aggregated command APDU

If the device has been configured to publish more than one messages (see IEC 62591:2010, 8.2.3.32), it can combine these into one message using command aggregation, except that “Publish data Message 0” shall never be aggregated.

Whenever possible, the field device shall aggregate multiple pending “Publish data Message” into one transaction using Information report request. If the multiple “Publish data Message” does not fit into a single transaction, then these can be sent in separately.

5.3 Device application service-specific APDU structures

5.3.1 Identify FAL PDU

5.3.1.1 Request primitive

The value of Command field can be either 0 or 11 or 21. The APDU format is shown in Table 5.

Table 5 – Identify request APDU

Command	Octet count	Data
0	0	Null
11	6	Tag (Packed ASCII)
21	32	Long Tag (Latin-1)

If the value of Command field is 0, then Value field is null. If the value of Command field is 11, then Value field contains six octet value of Tag parameter. If the value of Command field is 21, then Value field contains 32 octet value of Long Tag parameter.

5.3.1.2 Response primitive

The Value field of Identify response is shown in Table 6. The Response code values are shown in Table 7.

Table 6 – Identify response value field

Octet offset	Data type	Parameter name	Description
0	Unsigned8	None	is always 254
1	Enumeration	Expanded Device Type	is a 2-octet enumeration and uniquely identifies the type of the device
3	Unsigned8	Preamble Count	indicates the minimum number of Preambles to be sent with the request message from the Master to the responding device
4	Unsigned8	Command Rev	indicates the major revision level of the Protocol supported by the device
5	Unsigned8	Device Rev	describes the revision level of the device. The value of this parameter shall be defined by the manufacturer.
6	Unsigned8	Soft Rev	indicates the revision level of the firmware in the device. The manufacturer shall increment the value of this parameter for every new release of the device's firmware
7	Bit Field	Hard Rev Phy Type	a) the five most significant bits of this octet represent the revision level of the device hardware. The manufacturer shall increment the value of this attribute for every major change of the device's hardware. It is not necessary to track individual hardware component changes. b) the least three significant bits of this octet represent the type of PhL signalling used by the device
8	Bit Field	Device Flag	indicates other information about the device such as multi-sensor, non-volatile memory control, protocol bridge, etc.
9	Unsigned24	Device ID	indicates a serial number for the device. The manufacturer shall assign a unique value for every device that has the identical values for Manufacturer ID and Expanded Device Type
12	Unsigned8	Preamble Count	indicates the minimum number of preambles to be sent with the response message from the slave to the master
13	Unsigned8	Variable Count	specifies the maximum number of objects (variables) that can be accessed from the device. The value of this parameter indicates the last variable code that a Client application can expect to be found in the device
14	Unsigned16	Configuration Change Count	keeps track of number of device configuration changes. The device shall increment the value of this parameter every time it receives a request to change the configuration using application layer services, or a user of the device changes the device configuration using local mean such as local operator's interface
16	Bit Field	Device ExtStatus	indicates the extended operational status of the device
17	Enumeration	Manufacturer ID	is a 2 octet enumeration and indicates the manufacturer that produced the device
19	Enumeration	Distributor code	is a 2 octet enumeration and indicates the private label manufacturer that distributed the device
21	Enumeration	Device profile	specifies the class to which the device belongs as shown in IEC 62591:2010, Table F.32.

Table 7 – Identify command specific response codes

Value	Class	Description
0	Success	No error

5.3.1.3 Procedure at responding device

If the request contains a Tag or Long Tag, then the device shall compare it with the Tag or Long Tag stored in the device. The device shall respond only if the value in request matches the value stored in the device.

5.3.2 Read primary variable FAL PDU

5.3.2.1 Request primitive

The value of Command is one (1); the value of Octet Count field is zero (0) and Data field is null.

5.3.2.2 Response primitive

The Value field of response is shown in Table 8. The Response Code values are shown in Table 9.

Table 8 – Read primary variable response value field

Octet offset	Data type	Parameter name	Description
0	Engineering unit	Primary Variable Unit	This parameter represents the engineering unit of primary variable.
1 – 4	Float32	Primary Variable	It is the current value of primary variable.

Table 9 – Read primary variable command specific response codes

Value	Class	Description
0	Success	No error
6	Error	Device Specific Command Error
8	Warning	Update Failure
16	Error	Access Restricted

5.3.3 Read loop current and percent of range FAL PDU

5.3.3.1 Request primitive

The value of Command is two (2); the value of Octet Count field is zero (0) and Data field is null.

5.3.3.2 Response primitive

The Value field of response is shown in Table 10. The Response Code values are shown in Table 11.

Table 10 – Read loop current and percent of range value field

Octet offset	Data type	Parameter name	Description
0 – 3	Float32	Loop Current	It is the current value of the output of the device – either loop current in milli-amperes or voltage output in volts.
4 – 7	Float32	Primary Variable	It is the current value of primary variable in percent of range. It tracks the primary input to the device. Its value can be outside the normal range of 0 % to 100 %.

Table 11 – Read loop current and percent of range command specific response codes

Value	Class	Description
0	Success	No error
6	Error	Device-Specific Command Error
8	Warning	Update Failure
16	Error	Access Restricted

5.3.4 Read dynamic variables and loop current FAL PDU

5.3.4.1 Request primitive

The value of Command is three (3); the value of Octet Count field is zero (0) and Data field is null.

5.3.4.2 Response primitive

The Value field of response is shown in Table 12. The Response Code values are shown in Table 13. If a device does not support all four variables, then the response contains only the number of variables that the device supports. In that case, the remaining fields are absent from the response.

Table 12 – Read dynamic variables and loop current value field

Octet offset	Data type	Parameter name	Description
0 – 3	Float32	Loop Current	It is the current value of the output of the device – either loop current in milli-amperes or voltage output in volts.
4	Engineering unit	Primary Variable Unit	This parameter represents the engineering unit of primary variable.
5 – 8	Float32	Primary Variable	It is the current value of primary variable.
9	Engineering unit	Secondary Variable Unit	This parameter represents the engineering unit of secondary variable.
10 – 13	Float32	Secondary Variable	It is the current value of secondary variable.
14	Engineering unit	Tertiary Variable Unit	This parameter represents the engineering unit of tertiary variable.
15 – 18	Float32	Tertiary Variable	It is the current value of tertiary variable.
19	Engineering unit	Quaternary Variable Unit	This parameter represents the engineering unit of quaternary variable.
20 – 23	Float32	Quaternary Variable	It is the current value of quaternary variable.

Table 13 – Read dynamic variables and loop current command specific response codes

Value	Class	Description
0	Success	No error
6	Error	Device-Specific Command Error
8	Warning	Update Failure
16	Error	Access Restricted

5.3.5 Write loop configuration FAL PDU

5.3.5.1 Request primitive

The value of Command is six (6); the value of Octet Count field is two (2) and Data field is as shown in Table 14.

Table 14 – Write loop configuration value field

Octet offset	Data type	Parameter name	Description
0	Unsigned8	Polling Address	This parameter represents the data-link address of the device.
1	Enumeration	Loop Current Mode	This parameter encodes the loop current mode as shown in Table C.6.

5.3.5.2 Response primitive

The Value field of response is shown in Table 14. The Response Code values are shown in Table 15.

Table 15 – Write loop configuration command specific response codes

Value	Class	Description
0	Success	No error
2	Error	Invalid Poll Address
5	Error	Too few data octets received
6	Error	Device-Specific Command Error
7	Error	In write protect mode
12	Error	Invalid mode selection
16	Error	Access Restricted
32	Error	Busy

5.3.5.3 Procedure at responding device

A devices shall disable loop current signalling when requested by the Master. When current signalling is disabled, the loop current shall be set to the minimum value required for field device operation. The Application process status bit 4 as shown in Table 3, Loop Current Fixed, shall be set to '1'.

5.3.6 Read loop configuration FAL PDU

5.3.6.1 Request primitive

The value of Command is seven (7); the value of Octet Count field is zero (0) and Data field is null.

5.3.6.2 Response primitive

The Value field of response is shown in Table 16. The Response Code values are shown in Table 17.

Table 16 – Read loop configuration value field

Octet offset	Data type	Parameter name	Description
0	Unsigned8	Polling Address	This parameter represents the data-link address of the device.
1	Enumeration	Loop Current Mode	This parameter encodes the loop current mode as shown in Table C.6.

Table 17 – Read loop configuration command specific response codes

Value	Class	Description
0	Success	No error
16	Error	Access Restricted

5.3.7 Read dynamic variable families classifications FAL PDU**5.3.7.1 Request primitive**

The value of Command is eight (8); the value of Octet Count field is zero (0) and Data field is null.

5.3.7.2 Response primitive

The Value field of response is shown in Table 18. The Response Code values are shown in Table 19.

Table 18 – Read dynamic variable families classifications value field

Octet offset	Data type	Parameter name	Description
0	Enumeration	Primary Variable Classification	This parameter indicates the function performed by the device variable as specified in IEC 62591:2010, Table F.5.
1	Enumeration	Secondary Variable Classification	This parameter indicates the function performed by the device variable as specified in IEC 62591:2010, Table F.5.
2	Enumeration	Tertiary Variable Classification	This parameter indicates the function performed by the device variable as specified in IEC 62591:2010, Table F.5.
3	Enumeration	Quaternary Variable Classification	This parameter indicates the function performed by the device variable as specified in IEC 62591:2010, Table F.5.

Table 19 – Read dynamic variable families classifications command specific response codes

Value	Class	Description
0	Success	No error
16	Error	Access Restricted

5.3.8 Read device variables with status FAL PDU**5.3.8.1 Request primitive**

The value of Command is nine (9); the value of Octet count field is one to eight (1 – 8) and it is equal to number of requested variables. The request Value field is shown in Table 20.

Table 20 – Read device variables with status request value field

Octet offset	Data type	Parameter name	Description
0	Unsigned8	Slot 0	This parameter is the code assigned to a device variable
1	Unsigned8	Slot 1	This parameter is the code assigned to a device variable
2	Unsigned8	Slot 2	This parameter is the code assigned to a device variable
3	Unsigned8	Slot 3	This parameter is the code assigned to a device variable
4	Unsigned8	Slot 4	This parameter is the code assigned to a device variable
5	Unsigned8	Slot 5	This parameter is the code assigned to a device variable
6	Unsigned8	Slot 6	This parameter is the code assigned to a device variable
7	Unsigned8	Slot 7	This parameter is the code assigned to a device variable

5.3.8.2 Response primitive

The Value field of response is shown in Table 21. The Response code values are shown in Table 22.

Table 21 – Read device variables with status value field

Octet offset	Data type	Parameter name	Description
0	Bit Field	Device ExtdStatus	This parameter indicates the extended operational status of the device.
1	Unsigned8	Variable Code	This is the code received in request at Value field for slot 0.
2	Enumeration	Variable classification	This indicates the function performed by the device variable for slot 0 as specified in IEC 62591:2010, Table F.5.
3	Engineering unit	Variable Unit	This represents the engineering unit of variable for slot 0.
4	Float32	Variable Value	It is the current value of variable for slot 0.
8	Bit Field	Variable Status	This represents the status of variable for slot 0 as specified in Clause B.6.
9	Unsigned8	Variable Code	This is the code received in request at Value field for slot 1.
10	Enumeration	Variable classification	This indicates the function performed by the device variable for slot 1 as specified in IEC 62591:2010, Table F.5.
11	Engineering unit	Variable Unit	This represents the engineering unit of variable for slot 1.
12	Float32	Variable Value	It is the current value of variable for slot 1.
16	Bit Field	Variable Status	This represents the status of variable for slot 1 as specified in Clause B.6.
17	Unsigned8	Variable Code	This is the code received in request at Value field for slot 2.
18	Enumeration	Variable classification	This indicates the function performed by the device variable for slot 2 as specified in IEC 62591:2010, Table F.5.
19	Engineering unit	Variable Unit	This represents the engineering unit of variable for slot 2.
20	Float32	Variable Value	It is the current value of variable for slot 2.
24	Bit Field	Variable Status	This represents the status of variable for slot 2 as specified in Clause B.6.
25	Unsigned8	Variable Code	This is the code received in request at Value field for slot 3.
26	Enumeration	Variable classification	This indicates the function performed by the device variable for slot 3 as specified in IEC 62591:2010, Table F.5.
27	Engineering unit	Variable Unit	This represents the engineering unit of variable for slot 3.

Octet offset	Data type	Parameter name	Description
28	Float32	Variable Value	It is the current value of variable for slot 3.
32	Bit Field	Variable Status	This represents the status of variable for slot 3 as specified in Clause B.6.
33	Unsigned8	Variable Code	This is the code received in request at Value field for slot 4.
34	Enumeration	Variable classification	This indicates the function performed by the device variable for slot 4 as specified in IEC 62591:2010, Table F.5.
35	Engineering unit	Variable Unit	This represents the engineering unit of variable for slot 4.
36	Float32	Variable Value	It is the current value of variable for slot 4.
40	Bit Field	Variable Status	This represents the status of variable for slot 4 as specified in Clause B.6.
41	Unsigned8	Variable Code	This is the code received in request at Value field for slot 5.
42	Enumeration	Variable classification	This indicates the function performed by the device variable for slot 5 as specified in IEC 62591:2010, Table F.5.
43	Engineering unit	Variable Unit	This represents the engineering unit of variable for slot 5.
44	Float32	Variable Value	It is the current value of variable for slot 5.
48	Bit Field	Variable Status	This represents the status of variable for slot 5 as specified in Clause B.6.
49	Unsigned8	Variable Code	This is the code received in request at Value field for slot 6.
50	Enumeration	Variable classification	This indicates the function performed by the device variable for slot 6 as specified in IEC 62591:2010, Table F.5.
51	Engineering unit	Variable Unit	This represents the engineering unit of variable for slot 6.
52	Float32	Variable Value	It is the current value of variable for slot 6.
56	Bit Field	Variable Status	This represents the status of variable for slot 6 as specified in Clause B.6.
57	Unsigned8	Variable Code	This is the code received in request at Value field for slot 7.
58	Enumeration	Variable classification	This indicates the function performed by the device variable for slot 7 as specified in IEC 62591:2010, Table F.5.
59	Engineering unit	Variable Unit	This represents the engineering unit of variable for slot 7.
60	Float32	Variable Value	It is the current value of variable for slot 7.
64	Bit Field	Variable Status	This represents the status of variable for slot 7 as specified in Clause B.6.
65	Time	Time stamp	Slot 0 data time stamp

Table 22 – Read device variables with status command specific response codes

Value	Class	Description
0	Success	No error
2	Error	Invalid selection
5	Error	Too few data octets received
6	Error	Device-Specific Command Error
8	Warning	Update Failure
14	Warning	Dynamic variables returned for device variables
16	Error	Access restricted
30	Warning	Command response truncated

5.3.8.3 Procedure at responding device

If the request does not contain any data octets (Value field) then the device shall respond with error code equal to 5; otherwise the respond code shall be a value other than 5.

The device shall respond with values of exactly as many variables as in the request.

If a variable requested is not supported in the device, then in the response APDU, the corresponding:

Variable classification field shall be set to '0';

Variable Unit field shall be set to '250';

Variable Value field shall be set to a floating point value of 'Not-a-number' (0x7E, 0xA0, 0x00, 0x00); and

Variable Status field shall be set to 0x30.

If the Variable classification is not supported for a requested variable, then Variable classification field in the response APDU for that variable shall be set to '0' (not yet classified) and the respond code shall be set to '14'.

The device shall support at least four slots.

5.3.9 Read message FAL PDU

5.3.9.1 Request primitive

The value of Command is twelve (12); the value of Octet Count field is zero (0) and Data field is null.

5.3.9.2 Response primitive

The Value field of response is shown in Table 23. The Response Code values are shown in Table 24.

Table 23 – Read message response value field

Octet offset	Data type	Parameter name	Description
0 – 23	Packed ASCII	Message	This is the value of Message stored in the device.

Table 24 – Read message command specific response codes

Value	Class	Description
0	Success	No error
16	Error	Access Restricted
32	Error	Busy

5.3.10 Read tag, descriptor and date FAL PDU

5.3.10.1 Request primitive

The value of Command is thirteen (13); the value of Octet Count field is zero (0) and Data field is null.

5.3.10.2 Response primitive

The Value field of response is shown in Table 25. The Response Code values are shown in Table 26.

Table 25 – Read tag, descriptor, date response value field

Octet offset	Data type	Parameter name	Description
0 – 5	Packed ASCII	Tag	This is the value of Tag stored in the device.
6 – 17	Packed ASCII	Descriptor	This is the value of Descriptor stored in the device.
18 – 20	Date	Date	This is the value of Date stored in the device.

Table 26 – Read tag, descriptor, date command specific response codes

Value	Class	Description
0	Success	No error
16	Error	Access Restricted
32	Error	Busy

5.3.11 Read primary variable transducer information FAL PDU

5.3.11.1 Request primitive

The value of Command is fourteen (14); the value of Octet Count field is zero (0) and Data field is null.

5.3.11.2 Response primitive

The Value field of response is shown in Table 27. The Response Code values are shown in Table 28.

Table 27 – Read primary variable transducer information response value field

Octet offset	Data type	Parameter name	Description
0 – 2	Unsigned24	Transducer Serial Number	This is the value of Transducer Serial Number.
3	Engineering unit	Transducer Unit	This is the code for Transducer Limits and Minimum Span Units
4 – 7	Float32	Upper Transducer Limit	It is the value of Upper Transducer Limit variable.
8 – 11	Float32	Lower Transducer Limit	It is the value of Lower Transducer Limit variable.
12 – 15	Float32	Minimum Span	It is the value of Minimum Span variable.

**Table 28 – Read primary variable transducer information
command specific response codes**

Value	Class	Description
0	Success	No error
16	Error	Access Restricted
32	Error	Busy

5.3.11.3 Procedure at responding device

If the Transducer Serial Number is not applicable to the device, then the device shall respond with following values:

- Transducer Serial Number equal to '0';
- Transducer Unit equal to '250';
- all other values equal to a floating point value of 'Not-a-number' (0x7F, 0xA0, 0x00, 0x00).

5.3.12 Read device information FAL PDU

5.3.12.1 Request primitive

The value of Command is fifteen (15); the value of Octet Count field is zero (0) and Data field is null.

5.3.12.2 Response primitive

The Value field of response is shown in Table 29. The Response Code values are shown in Table 30.

Table 29 – Read device information response value field

Octet offset	Data type	Parameter name	Description
0	Enumeration	PV Alarm Selection	This is the value of PV Alarm Selection as shown in Table C.2. It indicates the action taken by the device under error conditions. For transmitters, the code indicates the action taken by the Loop Current. For Actuators, the action taken by the positioner is indicated.
1	Enumeration	PV Transfer Function	This is the value of PV Transfer Function as shown in Table C.1. If the device does not support transfer functions, then this value is '0'.
2	Engineering unit	PV Range Unit	This is the value of PV upper and lower Range Unit.
3 – 6	Float32	PV Upper Range	PV Upper Range Value
7 – 10	Float32	PV Lower Range	PV Lower Range Value
11 – 14	Float32	PV Damping	PV Damping Value in seconds
15	Enumeration	Write Protect	This is the value of Write Protect as shown in Table C.3. If the device does not implement write protection, then this value is '251'.
16	Enumeration	Reserved	This field is reserved for future use. Its value shall be '250'.
17	Bit Field	PV Analog Channel	This is the value of PV Analog Channel flags as shown in Table C.8.

Table 30 – Read device information command specific response codes

Value	Class	Description
0	Success	No error
16	Error	Access Restricted
32	Error	Busy

5.3.13 Read final assembly number FAL PDU**5.3.13.1 Request primitive**

The value of Command is sixteen (16); the value of Octet Count field is zero (0) and Data field is null.

5.3.13.2 Response primitive

The Value field of response is shown in Table 31. The Response Code values are shown in Table 32.

Table 31 – Read final assembly number response value field

Octet offset	Data type	Parameter name	Description
0 – 2	Unsigned24	Final Assembly Number	This is the value of Final Assembly Number

Table 32 – Read final assembly number command specific response codes

Value	Class	Description
0	Success	No error
16	Error	Access Restricted
32	Error	Busy

5.3.14 Write message FAL PDU**5.3.14.1 Request primitive**

The value of Command is seventeen (17); the value of Octet Count field is twenty-four (24) and Data field is as shown in Table 33.

Table 33 – Write message value field

Octet offset	Data type	Parameter name	Description
0 – 23	Packed ASCII	Message	This is a string used by a Master for record keeping

5.3.14.2 Response primitive

The Value field of response is shown in Table 33. The Response Code values are shown in Table 34.

Table 34 – Write message command specific response codes

Value	Class	Description
0	Success	No error
2	Error	Invalid Poll Address
5	Error	Too few data octetsreceived
6	Error	Device-Specific Command Error
7	Error	In write protect mode
16	Error	Access Restricted
32	Error	Busy

5.3.15 Write tag, descriptor and date FAL PDU

5.3.15.1 Request primitive

The value of Command is eighteen (18); the value of Octet Count field is twenty-one (21) and Data field is as shown in Table 35.

Table 35 – Write tag, descriptor, date value field

Octet offset	Data type	Parameter name	Description
0 – 5	Packed ASCII	Tag	This is the value of Tag to be stored in the device.
6 – 17	Packed ASCII	Descriptor	This is the value of Descriptor to be stored in the device.
18 – 20	Date	Date	This is the value of Date to be stored in the device.

5.3.15.2 Response primitive

The Value field of response is shown in Table 35. The Response Code values are shown in Table 36.

Table 36 – Write tag, descriptor, date command specific response codes

Value	Class	Description
0	Success	No error
5	Error	Too few data octetsreceived
6	Error	Device-Specific Command Error
7	Error	In write protect mode
9	Error	Invalid Date code
16	Error	Access Restricted
32	Error	Busy

5.3.16 Write final assembly number FAL PDU

5.3.16.1 Request primitive

The value of Command is nineteen (19); the value of Octet Count field is three (3) and Data field is as shown in Table 37.

Table 37 – Write final assembly number value field

Octet offset	Data type	Parameter name	Description
0 – 2	Unsigned24	Final Assembly Number	This is the value of Final Assembly Number. It is normally changed when electronics or other device components are upgraded in the field.

5.3.16.2 Response primitive

The Value field of response is shown in Table 37. The Response Code values are shown in Table 38.

Table 38 – Write final assembly number command specific response codes

Value	Class	Description
0	Success	No error
5	Error	Too few data octets received
6	Error	Device-Specific Command Error
7	Error	In write protect mode
16	Error	Access Restricted
32	Error	Busy

5.3.17 Read long tag FAL PDU**5.3.17.1 Request primitive**

The value of Command is twenty (20); the value of Octet Count field is zero (0) and Data field is null.

5.3.17.2 Response primitive

The Value field of response is shown in Table 39. The Response Code values are shown in Table 40.

Table 39 – Read long tag response value field

Octet offset	Data type	Parameter name	Description
0 – 31	Latin-1	Long Tag	This is the value of Long Tag stored in the device.

Table 40 – Read long tag command-specific response codes

Value	Class	Description
0	Success	No error
16	Error	Access Restricted
32	Error	Busy

5.3.18 Write long tag FAL PDU**5.3.18.1 Request primitive**

The value of Command is twenty-two (22); the value of Octet Count field is thirty-two (32) and Data field is as shown in Table 41.

Table 41 – Write long tag value field

Octet offset	Data type	Parameter name	Description
0 – 31	Latin-1	Long Tag	This is the value of Long Tag to be stored in the device.

5.3.18.2 Response primitive

The Value field of response is shown in Table 41. The Response Code values are shown in Table 42.

Table 42 – Write long tag command specific Response codes

Value	Class	Description
0	Success	No error
5	Error	Too few data octets received
6	Error	Device-Specific Command Error
7	Error	In write protect mode
16	Error	Access Restricted
32	Error	Busy
33	Error	DR initiated
34	Error	DR running
35	Error	DR dead
36	Error	DR conflict

5.3.19 Reset configuration changed flag FAL PDU

This Command is specified in IEC 62591:2010, 8.2.3.19.

5.3.20 Perform self test FAL PDU

This Command is specified in IEC 62591:2010, 8.2.3.20.

5.3.21 Perform device reset FAL PDU

This Command is specified in IEC 62591:2010, 8.2.3.21.

5.3.22 Read additional device status FAL PDU

This Command is specified in IEC 62591:2010, 8.2.3.22.

5.3.23 Reset more status available FAL PDU

This Command is specified in IEC 62591:2010, 8.2.3.23.

5.3.24 Read device variable information FAL PDU

This Command is specified in IEC 62591:2010, 8.2.3.24.

5.3.25 Write device variable FAL PDU

This Command is specified in IEC 62591:2010, 8.2.3.25.

5.3.26 Write primary variable range FAL PDU

5.3.26.1 Request primitive

The value of Command is thirty-five (35); the value of Octet count field is nine (9) and Value field is as shown in Table 43.

Table 43 – Write primary variable range value field

Octet offset	Data type	Parameter name	Description
0	Engineering unit	Range unit	Measurement unit of the Range value
1	Float32	Upper range value	Range value to be written
5	Float32	Lower range value	Range value to be written

5.3.26.2 Response primitive

The Value field of response is shown in Table 43. The Response code values are shown in Table 44.

Table 44 – Write primary variable range command specific response codes

Value	Class	Description
0	Success	No error
2	Error	Invalid selection
5	Error	Too few data octets received
6	Error	Device-specific Command error
7	Error	In write protect mode
8	Warning	Set to the nearest possible value
9	Error	Lower range value too high
10	Error	Lower range value too low
11	Error	Upper range value too high
12	Error	Upper range value too low
13	Error	Upper and Lower range values out of limits
14	Warning	Span too small (device accuracy may be impaired).
16	Error	Access Restricted
18	Error	Invalid Units code
29	Error	Invalid span
32	Error	Busy

5.3.26.3 Procedure at responding device

The Upper range value of the Primary variable can be lower than its Lower range value, enabling the device to be operated with reverse action.

NOTE The device-specific document will indicate if this capability has not been implemented.

The range values shall be returned in the same units as received. The value returned in the response shall be the rounded or truncated value actually used by the device. The Primary variable range Unit code received with this command shall not affect the Primary variable Unit of the device. For a current output device, the Range values allow the Primary variable value to be converted to a percent for transmission via the Loop current. For a current input device,

the Range values allow the Loop current to be converted to a percent for use by the actuator (e.g., to use as the actuator setpoint).

5.3.27 Enter-exit fixed current mode FAL PDU

5.3.27.1 Request primitive

The value of Command is forty (40); the value of Octet count field is four (4) and Value field is as shown in Table 45.

Table 45 – Enter-exit fixed current mode request value field

Octet offset	Data type	Parameter name	Description
0	Float32	Fixed current level	Loop current value to be set

5.3.27.2 Response primitive

The Value field of response is shown in Table 46. The Response code values are shown in Table 47.

Table 46 – Enter-exit fixed current mode response value field

Octet offset	Data type	Parameter name	Description
0	Float32	Actual current level	Actual current value

Table 47 – Enter-exit fixed current mode command specific response codes

Value	Class	Description
0	Success	No error
3	Error	Fixed current level too high
4	Error	Fixed current level too low
5	Error	Too few data octets received
6	Error	Device-specific Command error
7	Error	In write protect mode
11	Error	Loop current not active (device in multidrop mode)
16	Error	Access Restricted
32	Error	Busy

5.3.27.3 Procedure at responding device

If the Fixed current level value in the request is zero, then the device shall exit the fixed Loop current mode. Else, if the value is valid then the device shall be put in the fixed Loop current mode using the value in the request. The value returned in the response shall be the rounded or truncated value actually used by the device.

Fixed Loop current mode shall be exited when power is removed and reapplied to the device.

5.3.28 Write primary variable unit FAL PDU

5.3.28.1 Request primitive

The value of Command is forty (44); the value of Octet count field is one (1) and Value field is as shown in Table 48.

Table 48 – Write primary variable unit value field

Octet offset	Data type	Parameter name	Description
0	Engineering unit	Primary variable unit code	Primary variable measurement unit code

5.3.28.2 Response primitive

The Value field of response is shown in Table 48. The Response code values are shown in Table 49.

Table 49 – Write primary variable unit command specific response codes

Value	Class	Description
0	Success	No error
2	Error	Invalid selection
5	Error	Too few data octets received
6	Error	Device-specific Command error
7	Error	In write protect mode
16	Error	Access Restricted
32	Error	Busy

5.3.28.3 Procedure at responding device

The device shall select the units in which the Primary variable and its range are measured. This command also selects the unit for transducer limits and minimum span.

5.3.29 Trim loop current zero FAL PDU**5.3.29.1 Request primitive**

The value of Command is forty (45); the value of Octet count field is four (4) and Value field is as shown in Table 50. In the request PDU it is the externally measured Loop current.

Table 50 – Trim loop current zero request value field

Octet offset	Data type	Parameter name	Description
0	Float32	Zero loop current	Loop current level in mA

5.3.29.2 Response primitive

The Value field of response is shown in Table 50. In the response PDU it is the actual Loop current used by the device. The Response code values are shown in Table 51.

Table 51 – Trim loop current zero command specific response codes

Value	Class	Description
0	Success	No error
3	Error	Zero loop current level too high
4	Error	Zero loop current level too low
5	Error	Too few data octets received
6	Error	Device-specific Command error

Value	Class	Description
7	Error	In write protect mode
9	Error	Incorrect Loop current mode or value
11	Error	Loop current not active (device in multidrop mode)
16	Error	Access Restricted
32	Error	Busy

5.3.29.3 Procedure at responding device

The Field device shall trim its zero calibration of the Loop current so that the Loop current value matches the value received from the Master. The value returned in the response shall be the rounded or truncated value actually used by the device.

NOTE This trim is typically performed by adjusting the Loop current to 4,00 mA and sending the measured value to the Field device.

Response code 9 (Incorrect Loop current mode or value) shall be returned if the device is not in the proper mode to allow the Loop current to be calibrated or if the current was not set to exactly the minimum value.

If the device is a voltage output device, then it shall interpret the Loop current value as output value in Volts DC.

5.3.30 Trim loop current gain FAL PDU

5.3.30.1 Request primitive

The value of Command is forty (46); the value of Octet count field is four (4) and Value field is as shown in Table 52. In the request PDU it is the externally measured Loop current.

Table 52 – Trim loop current gain request value field

Octet offset	Data type	Parameter name	Description
0	Float32	Gain loop current	Loop current level in mA

5.3.30.2 Response primitive

The Value field of response is shown in Table 52. In the response PDU it is the actual Loop current used by the device. The Response code values are shown in Table 53.

Table 53 – Trim loop current gain command specific response codes

Value	Class	Description
0	Success	No error
3	Error	Gain loop current level too high
4	Error	Gain loop current level too low
5	Error	Too few data octets received
6	Error	Device-specific Command error
7	Error	In write protect mode
9	Error	Incorrect Loop current mode or value
11	Error	Loop current not active (device in multidrop mode)
16	Error	Access Restricted

Value	Class	Description
32	Error	Busy

5.3.30.3 Procedure at responding device

The Field device shall trim its gain calibration of the Loop current so that the Loop current value matches the value received from the Master. The value returned in the response shall be the rounded or truncated value actually used by the device.

NOTE This trim is typically performed by adjusting the Loop current to 20,00 mA and sending the measured value to the Field device.

Response code 9 (Incorrect Loop current mode or value) shall be returned if the device is not in the proper mode to allow the Loop current to be calibrated or if the current was not set to exactly the maximum value.

If the device is a voltage output device, then it shall interpret the Loop current value as output value in Volts DC.

5.3.31 Read dynamic variable assignment FAL PDU

5.3.31.1 Request primitive

The value of Command is fifty (50); the value of Octet count field is zero (0) and Value field is null.

5.3.31.2 Response primitive

The value of Octet count field is four (4) and Value field is shown in Table 54. The Response code values are shown in Table 55.

Table 54 – Read dynamic variable assignment response value field

Octet offset	Data type	Parameter name	Description
0	Unsigned8	Primary variable assignment	Device variable assigned to the Primary variable as specified in Table C.9.
1	Unsigned8	Secondary variable assignment	Device variable assigned to the Secondary variable as specified in Table C.9.
2	Unsigned8	Tertiary variable assignment	Device variable assigned to the Tertiary variable as specified in Table C.9.
3	Unsigned8	Quaternary variable assignment	Device variable assigned to the Quaternary variable as specified in Table C.9.

Table 55 – Read dynamic variable assignment command specific response codes

Value	Class	Description
0	Success	No error
6	Error	Device-specific Command error
16	Error	Access Restricted

5.3.31.3 Procedure at responding device

The device shall respond with the Device variable numbers that are assigned to the Primary, Secondary, Tertiary, and Quaternary Variables. The Field device shall return all response data octets. Unsupported Dynamic variable assignment shall be set to "250" (not used).

5.3.32 Write dynamic variable assignment FAL PDU

5.3.32.1 Request primitive

The value of Command is fifty-one (51); the value of Octet count field is four (4) and Value field is as shown in Table 56.

Table 56 – Write dynamic variable assignment value field

Octet offset	Data type	Parameter name	Description
0	Unsigned8	Primary variable assignment	Device variable assigned to the Primary variable as specified in Table C.9.
1	Unsigned8	Secondary variable assignment	Device variable assigned to the Secondary variable as specified in Table C.9.
2	Unsigned8	Tertiary variable assignment	Device variable assigned to the Tertiary variable as specified in Table C.9.
3	Unsigned8	Quaternary variable assignment	Device variable assigned to the Quaternary variable as specified in Table C.9.

5.3.32.2 Response primitive

The Value field of response is shown in Table 56. The Response code values are shown in Table 57.

Table 57 – Write dynamic variable assignment command specific response codes

Value	Class	Description
0	Success	No error
2	Error	Invalid selection
5	Error	Too few data octets received
6	Error	Device-specific Command error
7	Error	In write protect mode
16	Error	Access Restricted
32	Error	Busy

5.3.32.3 Procedure at responding device

The device shall write the Device variable numbers that are assigned to the Primary, Secondary, Tertiary and Quaternary variables. If the request has one or more data octets, it shall not respond with "Too few data octets received". The variable assignment not included in the request may be set to any valid Device variable in the Field device. The Field device shall always return all four response data octets. Unsupported Dynamic variable assignment shall be set to "250" (not used).

5.3.33 Write number of response preambles FAL PDU

5.3.33.1 Request primitive

The value of Command is fifty-nine (59); the value of Octet count field is one (1) and Value field is as shown in Table 58.

Table 58 – Write number of response preambles value field

Octet offset	Data type	Parameter name	Description
0	Unsigned8	Preamble count	Number of preambles characters to be sent in the PhPDU from the Slave to the Master

5.3.33.2 Response primitive

The Value field of response is shown in Table 58. The Response code values are shown in Table 59.

Table 59 – Write number of response preambles command specific response codes

Value	Class	Description
0	Success	No error
3	Error	Preamble count too high
4	Error	Preamble count too low
5	Error	Too few data octets received
6	Error	Device-specific Command error
7	Error	In write protect mode
8	Warning	Set to the nearest possible value
16	Error	Access Restricted
32	Error	Busy

5.3.33.3 Procedure at responding device

The device shall use the Preamble count parameter received in this request as the length of preamble in all PhPDU that is sent by the device. If the received value is less than five (5) or more than twenty (20), then the device shall use either reject the command and send appropriate response code.

5.3.34 Read device variable trim points FAL PDU**5.3.34.1 Request primitive**

The value of Command is eighty (80); the value of Octet count field is one (1) and Value field is as shown in Table 60.

Table 60 – Read device variable trim points request value field

Octet offset	Data type	Parameter name	Description
0	Unsigned8	Device variable	Device variable code as specified in Table C.9

5.3.34.2 Response primitive

The value of Octet count field is ten (10) and Value field is shown in Table 61. The Response code values are shown in Table 62.

Table 61 – Read device variable trim points response value field

Octet offset	Data type	Parameter name	Description
0	Unsigned8	Device variable	Device variable code as specified in Table C.9
1	Engineering unit	Trim point unit	Measurement unit code of trim points
2	Float32	Lower trim point	Lower or single trim point value
6	Float32	Upper trim point	Upper trim point value

Table 62 – Read device variable trim points command specific response codes

Value	Class	Description
0	Success	No error
5	Error	Too few data octets received
6	Error	Device-specific Command error
17	Error	Invalid Device variable code i.e. the requested Device variable does not exist.
19	Error	Device Variable code not allowed for this command
32	Error	Busy

5.3.34.3 Procedure at responding device

The device shall respond with the most recent value used trim point values. If the Device variable does not support a trim operation, then both the upper and lower trim points shall be 'Not-A-Number'. If the Device variable supports only a single trim point, then the upper trim point shall be 'Not-A-Number'.

5.3.35 Read device variable trim guidelines FAL PDU**5.3.35.1 Request primitive**

The value of Command is eighty-one (81); the value of Octet count field is one (1) and Value field is as shown in Table 63.

Table 63 – Read device variable trim guidelines request value field

Octet offset	Data type	Parameter name	Description
0	Unsigned8	Device variable	Device variable code as specified in Table C.9

5.3.35.2 Response primitive

The value of Octet count field is twenty-three (23) and Value field is shown in Table 64. The Response code values are shown in Table 65.

Table 64 – Read device variable trim guidelines response value field

Octet offset	Data type	Parameter name	Description
0	Unsigned8	Device variable	Device variable code as specified in Table C.9
1	Enumeration	Trim points supported	Trim points that can be set in the device as enumerated in Table C.7
2	Engineering unit	Trim point unit	Measurement unit code of trim points

Octet offset	Data type	Parameter name	Description
3	Float32	Minimum lower trim point	Minimum value that will be accepted by the device during a lower trim procedure
7	Float32	Maximum lower trim point	Maximum value that will be accepted by the device during a lower trim procedure
11	Float32	Minimum upper trim point	Minimum value that will be accepted by the device during a upper trim procedure
15	Float32	Maximum upper trim point	Maximum value that will be accepted by the device during a upper trim procedure
19	Float32	Minimum trim point difference	Minimum difference between upper and lower trim points that will be accepted by the device during a trim procedure

Table 65 – Read device variable trim points command specific response codes

Value	Class	Description
0	Success	No error
5	Error	Too few data octets received
6	Error	Device-specific Command error
17	Error	Invalid Device variable code i.e. the requested Device variable does not exist.
19	Error	Device Variable code not allowed for this command
32	Error	Busy

5.3.35.3 Procedure at responding device

The device shall respond with the information that a Host will need to guide a user through a correct selection of trim points. If the Device variable does not support a trim operation, then Trim points supported shall be zero (0). Trim point unit shall be 'Not used' and all trim point values shall be 'Not-A-Number'. If the Device variable supports only a single trim point, then the upper trim point and trim point difference shall be 'Not-A-Number'.

5.3.36 Write device variable trim point FAL PDU

5.3.36.1 Request primitive

The value of Command is eighty-two (82); the value of Octet count field is seven (7) and Value field is as shown in Table 66.

Table 66 – Write device variable trim point value field

Octet offset	Data type	Parameter name	Description
0	Unsigned8	Device variable	Device variable code as specified in Table C.9
1	Enumeration	Trim points type	Trim points that can be set in the device as enumerated in Table C.7
2	Engineering unit	Trim point unit	Measurement unit code of trim point
3	Float32	Trim point value	Trim point value to be written

5.3.36.2 Response primitive

The Value field of response is shown in Table 66. The Response code values are shown in Table 67.

Table 67 – Write device variable trim point command specific response codes

Value	Class	Description
0	Success	No error
2	Error	Invalid selection
3	Error	Trim point value too high
4	Error	Trim point value too low
5	Error	Too few data octets received
6	Error	Device-specific Command error
7	Error	In write protect mode
9	Error	Applied process too high
10	Error	Applied process too low
11	Error	Trim error, excess correction attempted
13	Error	Computation error, trim values were not changed
14	Warning	Span too small
16	Error	Access Restricted
17	Error	Invalid Device variable code i.e. the requested Device variable does not exist.
18	Error	Invalid Units code
19	Error	Device Variable code not allowed for this command
32	Error	Busy – a delayed response could not be started.
33	Error	DR initiated
34	Error	DR running
35	Error	DR dead
36	Error	DR conflict

5.3.36.3 Procedure at responding device

This command performs a calibration adjustment for the requested Device variable. The trim point value that is sent in the request should represent the presently applied process variable value. On receipt of this request, the device shall check the value for validity (i.e. within acceptable limits), then attempt to adjust the indicated point in its calibrated calculation so that the resulting digital process value matches the value supplied in the request.

If the device cannot complete the trim adjustment within the time allowed by the DLE for a response, the device shall use the delayed response mechanism to inform the Master that the calibration is continuing.

5.3.37 Reset device variable trim FAL PDU

5.3.37.1 Request primitive

The value of Command is eighty-three (83); the value of Octet count field is seven (1) and Value field is as shown in Table 68.

Table 68 – Reset device variable trim value field

Octet offset	Data type	Parameter name	Description
0	Unsigned8	Device variable	Device variable code as specified in Table C.9

5.3.37.2 Response primitive

The Value field of response is shown in Table 68. The Response code values are shown in Table 69.

Table 69 – Reset device variable trim command specific response codes

Value	Class	Description
0	Success	No error
2	Error	Invalid selection
5	Error	Too few data octets received
7	Error	In write protect mode
16	Error	Access Restricted
17	Error	Invalid Device variable code i.e. the requested Device variable does not exist.
19	Error	Device Variable code not allowed for this command
29	Error	Invalid span
32	Error	Busy – a delayed response could not be started.
33	Error	DR initiated
34	Error	DR running
35	Error	DR dead
36	Error	DR conflict

5.3.37.3 Procedure at responding device

The device shall reset the trim points for the requested Device variable to the factory default values. If the device cannot complete the trim adjustment within the time allowed by the DLE for a response, the device shall use the delayed response mechanism to inform the Master that the calibration is continuing.

5.3.38 Write publish data period FAL PDU

This Command is specified in IEC 62591:2010, 8.2.3.27, except that the Minimum update period shall be 0,500 s and there is no default value.

NOTE The minimum time between two publish mode message is equal to RT2 time plus the time required to transmit one publish mode PhPDU. This time is at least 375 ms for publishing the command 3 – one of the commonly used commands for this purpose.

5.3.39 Write publish data trigger FAL PDU

This Command is specified in IEC 62591:2010, 8.2.3.28.

5.3.40 Read publish data mode configuration FAL PDU

This Command is specified in IEC 62591:2010, 8.2.3.29.

5.3.41 Flush delayed responses FAL PDU

This Command is specified in IEC 62591:2010, 8.2.3.30.

5.3.42 Write publish data device variables FAL PDU

This Command is specified in IEC 62591:2010, 8.2.3.31.

5.3.43 Write publish data mode command number FAL PDU

This Command is specified in IEC 62591:2010, 8.2.3.32.

5.3.44 Write publish data mode control FAL PDU

This Command is specified in IEC 62591:2010, 8.2.3.33, except that the clause '8.2.3.33.3 Procedure at responding device' shall be replaced by the following.

This command is used to enable or disable the Publish data mode for devices. This command affects data-link layer operation. All data-link layer requirements for entering and exiting Publish Data mode shall be met.

NOTE In the Data link layer specifications, the Publish Data mode is referred to as Burst mode.

If 'Publish data mode control' in the request is either '1' or '3', the request is for enabling the publishing, then the responding device shall perform the following procedure.

- It shall set the Burst mode variable to 'Enable'.
- The device shall send a response indicating 'Success'.

If 'Publish data mode control' in the request is either '0' or '2', the request is for disabling the publishing, then the responding device shall perform the following procedure.

- It shall set the Burst mode variable to 'Disable'.
- The responding device shall respond with 'Success'.

If there is only one octet in the request, then that octet is 'Publish data mode control', the device shall assume that Publish data message '0' is being activated or deactivated for publishing.

5.3.45 Aggregated command

5.3.45.1 Request primitive

The value of Command is seventy-eight (78); the value of Octet count field shall be less than or equal to 255 and Value is shown in 5.2.5. Although a read command may be aggregated with a write command, this practice is not recommended. Commands may be aggregated as long as they do not have dependencies on each other. For example, several read commands may be aggregated in one transaction. Some restrictions may be imposed in the requirements for specific commands. For example, a host application (FAL-user) for device calibration may prevent command aggregation. The protocol itself does not put any other restriction on the command.

5.3.45.2 Response primitive

The value of Command is seventy-eight (78); the value of Octet count field shall be less than or equal to 255 and Value is shown in 5.2.5. The Response code values are shown in Table 70.

Table 70 – Aggregated command specific response codes

Value	Class	Description
0	Success	No error
2	Error	Invalid selection
5	Error	Too few data octets received
6	Error	Device-specific Command error
8	Warning	Update failure

Value	Class	Description
9	Error	Invalid command requested
16	Error	Access Restricted
30	Warning	Command response truncated - one or more commands response missing
32	Error	Busy – a delayed response could not be started.
33	Error	DR initiated
34	Error	DR running
35	Error	DR dead
36	Error	DR conflict

5.3.45.3 Procedure at responding device

If a device receives an aggregated request APDU, then it shall provide response to all “Command” fields in the request in one aggregated response APDU, unless the length of value field for the aggregated response exceeds 255. In that case, the device shall truncate the response and send the response code value of ‘30’. If any one of the commands in the request is invalid, then the device shall not perform any command execution and send the response code value of ‘9’.

5.4 Data coding rules

5.4.1 General

If the user data is always a simple (primitive) component. It is encoded as shown in Figure 5. The semantics of the user data are known implicitly from the position in the PDU, the length is fixed and implicitly known.

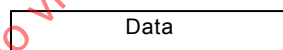


Figure 5 – Coding without identification

5.4.2 Coding for integer types

Integer values are signed quantities as shown in Figure 6 and Figure 7. The MSO of this data type is transferred first followed by the next octets and the LSO is transferred last.

Notation:	Integer8, Integer16, Integer24, Integer32		
Range of Values:	Data Type	range of values	length
	Integer8	$-128 \leq i \leq 127$	1 octet
	Integer16	$-32768 \leq i \leq 32767$	2 octets
	Integer24	$-2^{23} \leq i \leq 2^{23} - 1$	3 octets
	Integer32	$-2^{31} \leq i \leq 2^{31} - 1$	4 octets
Coding:	In two's complement representation the most significant bit is the bit after the sign bit (SN) in the first octet. SN = 0: positive numbers and z SN = 1: negative numbers		

Figure 6 – Coding of Integer type data

	bits	8	7	6	5	4	3	2	1
octets	1	SN	2^{14}	2^{13}	2^{12}	2^{11}	2^{10}	2^9	2^8
	2	2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0

Figure 7 – Coding of Integer16 type data

5.4.3 Coding for unsigned Types

Unsigned Values are encoded as shown in Figure 8 and Figure 9. The MSO of this data type is transferred first followed by the next octets and the LSO is transferred last.

Notation:	Unsigned8, Unsigned16, Unsigned24, Unsigned32		
Range of Values:	Data Type	range of values	length
	Unsigned8	$0 \leq i \leq 255$	1 octet
	Unsigned16	$0 \leq i \leq 65535$	2 octets
	Unsigned24	$0 \leq i \leq 2^{24} - 1$	3 octets
	Unsigned32	$0 \leq i \leq 2^{32} - 1$	4 octets
Coding	Binary		

Figure 8 – Coding of Unsigned type data

	bits	8	7	6	5	4	3	2	1
octets	1	2^{15}	2^{14}	2^{13}	2^{12}	2^{11}	2^{10}	2^9	2^8
	2	2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0

Figure 9 – Coding of Unsigned16 type data

5.4.4 Coding for floating point Type

Floating Point values are encoded as shown in Figure 10 and Figure 11. The sign and MSO of the exponent is transferred first followed by the balance of the exponent and the MSO to LSO of the fraction. If the value of an object of floating point data type is not known, then the value transferred is 0x7F, 0xA0, followed by all zeros (0x00); this value represents a 'Not-a-number'.

Notation:	Floating-Point (4 octet)							
Range of Values:	see ISO/IEC/IEEE 60559, Short Real Number (32 bits)							
Coding:	see ISO/IEC/IEEE 60559, Short Real Number (32 bits)							
bits	8	7	6	5	4	3	2	1
octets	Exponent (E)							
1	SN	2^7	2^6	2^5	2^4	2^3	2^2	2^1
2	(E)	Fraction (F)						
	2^0	2^{-1}	2^{-2}	2^{-3}	2^{-4}	2^{-5}	2^{-6}	2^{-7}
3	Fraction (F)							
	2^{-8}	2^{-9}	2^{-10}	2^{-11}	2^{-12}	2^{-13}	2^{-14}	2^{-15}
4	Fraction (F)							
	2^{-16}	2^{-17}	2^{-18}	2^{-19}	2^{-20}	2^{-21}	2^{-22}	2^{-23}

SN: sign 0 = positive, 1 = negative

SN: sign 0 = positive, 1 = negative

Figure 10 – Coding of single precision Floating Point type data

Notation:		Floating-Point (8 octet)							
Range of Values:		see ISO/IEC/IEEE 60559, Long Real Number (64 bits)							
Coding:		see ISO/IEC/IEEE 60559, Long Real Number (64 bits)							
bits		8	7	6	5	4	3	2	1
octets		Exponent (E)							
1		SN	2^{10}	2^9	2^8	2^7	2^6	2^5	2^4
2		Exponent (E)				Fraction (F)			
		2^3	2^2	2^1	2^0	2^{-1}	2^{-2}	2^{-3}	2^{-4}
3		Fraction (F)							
		2^{-5}	2^{-6}	2^{-7}	2^{-8}	2^{-9}	2^{-10}	2^{-11}	2^{-12}
4		Fraction (F)							
		2^{-13}	2^{-14}	2^{-15}	2^{-16}	2^{-17}	2^{-18}	2^{-19}	2^{-20}
5		Fraction (F)							
		2^{-21}	2^{-22}	2^{-23}	2^{-24}	2^{-25}	2^{-26}	2^{-27}	2^{-28}
6		Fraction (F)							
		2^{-29}	2^{-30}	2^{-31}	2^{-32}	2^{-33}	2^{-34}	2^{-35}	2^{-36}
7		Fraction (F)							
		2^{-37}	2^{-38}	2^{-39}	2^{-40}	2^{-41}	2^{-42}	2^{-43}	2^{-44}
8		Fraction (F)							
		2^{-45}	2^{-46}	2^{-47}	2^{-48}	2^{-49}	2^{-50}	2^{-51}	2^{-52}
SN: sign 0 = positive, 1 = negative									

Figure 11 – Coding of double precision Floating Point type data

5.4.5 Coding for date type

The data type Date consists of a calendar date as shown in Table 71 and Figure 12. The first octet representing day of month is transferred first, followed by the month octet and the year octet is transferred last.

Notation: Date

Range of Values: 1 January 1900 to 31 December 2155

Coding: in 3 octets

Table 71 – Coding for Date type

Parameter	Range of values	Meaning of the parameters
d. of m.	1...31	day of month
months	1...12	months
years	0...255	years – 1900

bits	8	7	6	5	4	3	2	1	
octets	RSV	RSV	RSV	2^4	2^3	2^2	2^1	2^0	1...31 d. of m.
1									
2	RSV	RSV	RSV	RSV	2^3	2^2	2^1	2^0	1...12 months
3	2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0	0 ... 255 years

Figure 12 – Coding of Date type data

5.4.6 Coding for enumeration

The data type Enumeration is used to associate specific user definitions to numeric codes. The definitions are as shown in Table 72. The definitions for codes 250–255 is always as shown. It is not necessary to define all possible values in the table. All undefined codes in the table are reserved. Reserved values shall not be used, returned or transferred by any device. For every object of Enumeration type, there is such table that is known to the users by associating the table with a variable index.

Range of Values: 0 to 255
Coding: in 1 octet

Table 72 – Coding for one octet Enumerated Type

Unsigned value	Definition
0	Specified definition 0
1	Specified definition 1
⋮	⋮
n	Specified definition n
n+1	Reserved
n+2	Reserved
⋮	⋮
249	Reserved
250	"Not Used"
251	"None"
252	"Unknown"
253	"Special"
254	"Expansion"
255	Reserved

5.4.7 Coding for bit field

The Bit Field type of data is used to transfer objects encoded as single-bit data. Individual bits are packed together, and reserved bits are added as necessary to transfer one octet. A lookup table is used to define the meaning of each bit in the bit field as shown in Table 73. In other words, each bit shall be either true (set to a 1) or false (set to a 0). Any unused bit is reserved. For every object of Bit Field type, there is such table that is known to the users by associating the table with a variable index.

Range of Values: 0 to 255
Coding: in 1 octet

Table 73 – One octet bit field

Bit mask	Definition
0x01	For Bit 1 (least significant bit)
0x02	For Bit 2
0x04	For Bit 3
0x08	For Bit 4
0x10	For Bit 5
0x20	For Bit 6
0x40	For Bit 7

Bit mask	Definition
0x80	For Bit 8 (most significant bit)

5.4.8 Coding for time type

The data type Time is 32-bit unsigned integer and represents time in the increments of 1/32 of a millisecond. If this data type is used to represent time of day, then it indicates number of 1/32 of milliseconds since midnight. The MSO of this data type is transferred first followed by the next octets and the LSO is transferred last.

5.4.9 Coding for engineering unit type

The data type Engineering unit is one octet long. The coding of this data type is same as that of Unsigned8.

5.4.10 Coding for packed ASCII string type

Packed ASCII is a 6-bit character code representing a subset of the ASCII character code set as shown in Table 74. It is produced by compressing four Packed ASCII characters into three octets. This type of string is always a multiple of 4 characters (3 octets) and is padded out to the end of the data item with SP characters. The length of the string is not transferred along with the string. The length is predetermined for each type of object, which is identified by object index. If the number of characters in the value of a string object is less than its predetermined length, then unused characters at the end of the string are set to SP (0x20). For example, 4 SP characters at the end of a string would appear as the 3 octets: 0x82, 0x08, and 0x20. The first character of this string is transferred first, followed by the next character and the last character is transferred last.

Table 74 – Packed ASCII character set

Bits 4 and 5	Bits 0 to 3															
	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	@	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
1	P	Q	R	S	T	U	V	W	X	Y	Z	[	\	]	^	_
2	SP ^a	!	"	#	\$	%	&	'	(	)	*	+	,	-	.	/
3	0	1	2	3	4	5	6	7	8	9	:	;	<	=	>	?
NOTE Most significant hexadecimal digit is top to bottom; least significant is left to right.																
^a SP indicates a space character.																

5.4.11 Coding for restricted ISO Latin-1 string type

ISO Latin-1 (ISO/IEC 8859-1) string consists of one character per octet. A restricted version that omits rows 0, 1, 8 and 9 in the Latin-1 table is used, as shown in Table 75.

The length of the string is not transferred along with the string. The length is predetermined for each type of object, which is identified by object index. If the number of characters in the value of a string object is less than its predetermined length, then unused characters at the end of the string are set to zeros (0x00). One zero (0x00) at the end of a string ISO Latin-1 data item may not be sufficient to meet this requirement. A valid restricted ISO Latin-1 character, other than zero, is permitted in the last octet of an restricted ISO Latin-1 data object. The first character of this string is transferred first, followed by the next character and the last character is transferred last.

Table 75 – Acceptable subset of ISO Latin-1 characters

Bits 4 to 7	Bits 0 to 3															
	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0																
1																
2	SP ^a	!	"	#	\$	%	&	'	(	)	*	+	,	-	.	/
3	0	1	2	3	4	5	6	7	8	9	:	;	<	=	>	?
4	@	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
5	P	Q	R	S	T	U	V	W	X	Y	Z	[	\	]	^	_
6	`	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o
7	p	q	r	s	t	u	v	w	x	y	z	{		}	~	
8																
9																
A	NBSP ^b	ı	ç	£	¤	¥	¦	§	¨	©	ª	«	¬	SHY ^c	®	¯
B	°	±	²	³	´	µ	¶	·	¸	¹	º	»	¼	½	¾	¿
C	À	Á	Â	Ã	Ä	Å	Æ	Ç	È	É	Ê	Ë	Ì	Í	Î	Ï
D	Ð	Ñ	Ò	Ó	Ô	Õ	Ö	×	Ø	Ù	Ú	Û	Ü	Ý	Þ	ß
E	à	á	â	ã	ä	å	æ	ç	è	é	ê	ë	ì	í	î	ï
F	ð	ñ	ò	ó	ô	õ	ö	÷	ø	ù	ú	û	ü	ý	þ	ÿ

NOTE Most significant hexadecimal digit is top to bottom; least significant is left to right. Grayed out cells means that no character is assigned to this code.

^a SP indicates a space character.
^b NBSP indicates a non-breaking space character.
^c SHY indicates a soft hyphen.

6 Common procedures

6.1 Delayed response

Delayed response procedure is specified in IEC 62591:2010, 8.3.1.

6.2 Publish mode procedure

Based on Trigger mode specified in 5.3.39, if there is new Publish data message ready for sending, the ALE shall use DL-CYCLIC-DATA request to update the buffer in the DLE. If there is more than one Publish data message, then the ALE shall aggregate these as specified in 5.2.5.

7 FAL protocol state machines

7.1 General

The behaviour of the FAL is described by the Application Relationship Protocol Machine (ARPM). The ARPM describes the exchange of FAL-PDUs with a remote ARPM. There are two types of ARPM – Client and Server. In any FAL, there can be only one instance of ARPM. The type of ARPM depends upon the device class. The Master class device has Client ARPM; the Slave class device has Server ARPM. The ARPM is responsible for the following activities:

- to accept the FAL service primitive from the FAL user, to perform state transition, to prepare DLL service primitives, and send them to the DLL;
- to receive DLL indication or confirmation primitives from the DLL, to perform state transition and deliver the FAL service primitive to the FAL users.

NOTE The state machines specified in the 7.3 and 7.4 only define the valid events for each. It is a local matter to handle the invalid events.

7.2 AREP mapping to data link layer

7.2.1 General

Subclause 7.2 describes the mapping of the AL to the Fieldbus data-link layer. It does not redefine the DLSAP attributes or DLME attributes that are defined in the data-link layer specification; rather, it defines how they are used by each of the AR classes. A means to configure and monitor the values of these attributes is provided by Network Management.

The following class definitions describe the DLSAP attributes and the DLME attributes required to support each of the AREP classes.

7.2.2 DLL mapping of client AREP class

7.2.2.1 Client AREP class formal model

The DLL Mapping attributes and their permitted values and the DLL services used with the Client AREP class are defined in 7.2.2.

CLASS: Client
PARENT CLASS: Top
ATTRIBUTES:
 1. (m) KeyAttribute: Destination address
 2. (m) Attribute: Preamble length
DLL SERVICES:
 1. (m) OpsService: DL-DATA-EXCHANGE request
 2. (m) OpsService: DL-DATA-EXCHANGE confirm
 3. (m) OpsService: DL-RECEIVE indication

7.2.2.2 Attributes

7.2.2.2.1 Destination address

This attribute specifies the remote address to which request APDU is sent, or from which response APDUs is received.

7.2.2.2.2 Preamble length

This parameter the length of Preamble that the DLL has to transmit when sending a DLPDU to the address specified by Destination address.

7.2.2.3 DLL services

7.2.2.3.1 DL-DATA-EXCHANGE request

This service is used in Master device to transfer an APDU to the destination Slave DLL user and to receive response APDU from the destination DLL user.

7.2.2.3.2 DL-DATA-EXCHANGE confirm

The DLL uses this service to transfer the response APDU from a remote Slave DLL user.

7.2.2.3.3 DL-RECEIVE indication

The Master DLL uses this service to indicate the reception of a publish mode APDU.

7.2.3 DLL mapping of server AREP class

7.2.3.1 Server AREP class formal model

The DLL Mapping attributes and their permitted values and the DLL services used with the Server AREP class are defined in 7.2.3.

CLASS: Server

PARENT CLASS: Top

ATTRIBUTES:

1. (m) KeyAttribute: Address

DLL SERVICES:

1. (m) OpsService: DL-DATA-EXCHANGE indication
2. (m) OpsService: DL-DATA-EXCHANGE response
3. (m) OpsService: DL-CYCLIC-DATA request

7.2.3.2 Attributes

7.2.3.2.1 Address

This attribute specifies the address that the Slave device uses to receive and send DLPDU.

7.2.3.3 DLL services

7.2.3.3.1 DL-DATA-EXCHANGE indication

The DLL uses this service to transfer the indication APDU from a remote Master DLL user.

7.2.3.3.2 DL-DATA-EXCHANGE response

This service is used in Slave device to transfer an APDU to the remote Master DLL user.

7.2.3.3.3 DL-CYCLIC-DATA request

This service is used in Slave device to transfer an APDU for publishing.

7.3 Client ARPM

Figure 13 shows the FAL ARPM in a Client ALE. The machine stays in 'Idle' state until it receives a service request from the AL-user. After sending the ALPDU to the Server using DLL service; the machine transitions to 'Wait for response' state. When it receives the response from the Server, the machine completes the request and returns to 'Idle' state. The ALE services only one request at a time. The transitions are shown in Table 76. All read and write services have identical state transition and identical actions. Therefore, Table 76 does not list every service request and response separately.

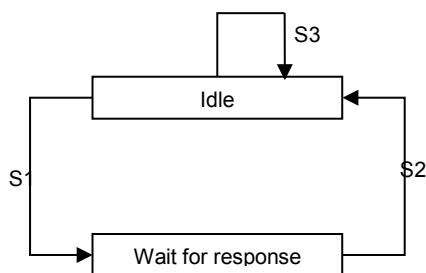


Figure 13 – Client state machine

Table 76 – Client machine state transitions

#	Current state	Event or condition => action	Next state
S1	Idle	Identify.req => DL-DATA-EXCHANGE request { Destination address := AREP_ID, Preamble length := Preamble count, DLS-user-data := FormReqApdu (Identify.req) }, Pending command := Identify	Wait for response
S1	Idle	Read(any).req => DL-DATA-EXCHANGE request { Destination address := AREP_ID, Preamble length := Preamble count, DLS-user-data := FormReqApdu (Read.req) }, Pending command := Command(Read.req)	Wait for response
S1	Idle	Write(any).req => DL-DATA-EXCHANGE request { Destination address := AREP_ID, Preamble length := Preamble count, DLS-user-data := FormReqApdu (Write.req) }, Pending command := Command(Write.req)	Wait for response
S1	Idle	Action(any).req => DL-DATA-EXCHANGE request { Destination address := AREP_ID, Preamble length := Preamble count, DLS-user-data := FormReqApdu (Action.req) }, Pending command := Command(Action.req)	Wait for response
S2	Wait for response	DL-DATA-EXCHANGE confirm && Status == 'Success' && CommErr (DLS-user-data) = 'False' && Command (DLS-user-data) == 0 11 21 => Identify.cnf(+) Response Code := RespCode(DLS-user-data), Application process status := ApStatus(DLS-user-data), Variable value := Value(DLS-user-data) }	Idle
S2	Wait for response	DL-DATA-EXCHANGE confirm && Status == 'Success' && CommErr (DLS-user-data) = 'False' && Command (DLS-user-data) != 0 11 21 => Read.cnf(+) Write.cnf(+) Action.cnf(+) { Response Code := RespCode (DLS-user-data), Application process status := ApStatus(DLS-user-data), Variable value := Value(DLS-user-data) }	Idle
S2	Wait for response	DL-DATA-EXCHANGE confirm && Status == 'Success' && CommErr (DLS-user-data) = 'False' && RespCode (DLS-user-data) == 'Error' && Command (DLS-user-data) == 0 11 21 => Identify.cnf(-) { Response Code := RespCode (DLS-user-data), Application process status := ApStatus(DLS-user-data) }	Idle
S2	Wait for response	DL-DATA-EXCHANGE confirm && Status == 'Success' && CommErr (DLS-user-data) = 'False' && RespCode (DLS-user-data) == 'Error' && Command (DLS-user-data) != 0 11 21 => Read.cnf(-) Write.cnf(-) Action.cnf(-) { Response Code := RespCode (DLS-user-data), Application process status := ApStatus(DLS-user-data) }	Idle

#	Current state	Event or condition => action	Next state
S2	Wait for response	DL-DATA-EXCHANGE confirm && Status == 'Success' && CommErr (DLS-user-data) = 'True' && Command (DLS-user-data) == 0 11 21 => Identify.cnf(-) { Communication status := CommCode (DLS-user-data), Application process status := ApStatus(DLS-user-data) }	Idle
S2	Wait for response	DL-DATA-EXCHANGE confirm && Status == 'Success' && CommErr (DLS-user-data) = 'True' && Command (DLS-user-data) != 0 11 21 => Read.cnf(-) Write.cnf(-) Action.cnf(-) { Communication status := CommCode (DLS-user-data), Application process status := ApStatus(DLS-user-data) }	Idle
S2	Wait for response	DL-DATA-EXCHANGE confirm && Status == 'No response' && Pending command == 0 11 21 => Identify.cnf(-) { Communication status := Device not available }	Idle
S2	Wait for response	DL-DATA-EXCHANGE confirm && Status == 'No response' && Pending command != 0 11 21 => Read.cnf(-) Write.cnf(-) Action.cnf(-) { Communication status := Device not available }	Idle
S3	Idle	DL-RECEIVE indication && Status = 'Success' => Information report.ind { AREP_ID := Source address, Numeric Identifier := Command (DLS-user-data), Response Code := RespCode(DLS-user-data), Application process status := ApStatus(DLS-user-data), Variable value := Value(DLS-user-data) }	Idle
NOTE Read (any) includes all AL Read services; Write (any) includes all AL Write services, Action (any) includes all AL Action services.			

7.4 Server ARPM

Figure 14 shows the FAL ARPM in a Server ALE. The machine transitions to 'Wait for response' state when it receives DL-DATA-EXCHANGE indication. It provides the service indication to the AL-user and waits for a response from the AL-user. When it receives the response, the machine completes the processing by using DL-DATA-EXCHANGE response service and returns to 'Idle' state. The transitions are shown in Table 77. All read and write services have identical state transition and identical actions. Therefore, this table does not list every service request and response separately.

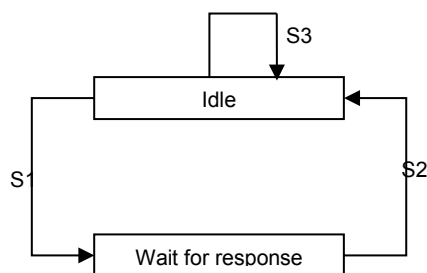


Figure 14 – Server state machine

Table 77 – Server machine state transitions

#	Current state	Event or condition => action	Next state
S1	Idle	DL-DATA-EXCHANGE indication => al_service.ind { AREP_ID := Source address, Numeric Identifier := Command (DLS-user-data), Variable value := Value(DLS-user-data) }	Wait for response
S2	Wait for response	al_service.rsp => DL-DATA-EXCHANGE response { DLS-user-data := FormRspAdu (al_service.rsp) }	Idle
S3	Idle	Information report.req => DL-CYCLIC-DATA request { DLS-user-data := FormRspAdu (Information report.req) }	Idle
NOTE al_service includes all AL services.			

7.5 Functions used by FAL state machines

Table 78 to Table 84 describe the functions used by FAL state machines.

Table 78 – Function FormReqAdu

Name	FormReqAdu()
Input	Output
AL-user request	APDU
Function	
Uses the parameters to form the request APDU as specified in 5.2.	

Table 79 – Function Command

Name	Command()
Input	Output
AL-user request	Command
Function	
Retrieve the variable index (command number) from the AL-user request.	

Table 80 – Function CommErr

Name	CommErr()
Input	Output
DLS-user-data	True / False
Function	
If the bit-7 of the first octet of DLS-user-data is '1' then the output shall be 'True' else the output shall be 'False'.	

Table 81 – Function RespCode

Name	RespCode()		
Input		Output	
DLS-user-data		Response Code	
Function			
This function returns Response code field (first octet of DLS-user-data) in the APDU. This function should be used only if the bit-7 of the first octet of DLS-user-data is '0'; otherwise first octet of DLS-user-data is not Response code.			

Table 82 – Function Commcode

Name	Commcode()		
Input		Output	
DLS-user-data		Communication error code	
Function			
This function returns Communication code field (first octet of DLS-user-data) in the APDU. This function should be used only if the bit-7 of the first octet of DLS-user-data is '1'; otherwise first octet of DLS-user-data is not Communication error code.			

Table 83 – Function ApStatus

Name	ApStatus()	
Input		Output
DLS-user-data		Application process status
Function		
This function returns Application process status field (second octet of DLS-user-data) in the APDU.		

Table 84 – Function Value

Name	Value()	
Input		Output
DLS-user-data		Variable value
Function		
This function returns Variable value field (DLS-user-data third octet to the end) in the APDU.		

Annex A (normative)

Application process status

A.1 General

Application process status is part of all response APDUs from a Slave field device to a Master device. It is also known as Device status, because it indicates the current operating status of the field device as a whole and is not associated with the completion of any command. This status shall be meaningful even if a communication error response is sent by DLE. The bits in this field are described in 5.1.6. This annex defines conditions under which each bit shall be set to '1'.

A.2 Device malfunction

This bit shall be used to indicate that an error has been detected that prevents proper operation of the field device (for example, at least one of the Device variables is incorrect). Host application should consider this as an alarm condition and refrain from using the device for process control applications. Devices supporting Command 48 shall provide diagnostic information in the Command 48 response value field. The Device malfunction bit shall remain '0' until a device completes its power-on self tests. If possible, the device shall respond to all requests even if a Device malfunction is detected.

A.3 Configuration changed

A field device shall contain two 'Configuration changed' bits – separate bits for the primary and the secondary master. Both bits shall be set to '1' when any configuration item in that field device is changed. These bits are used along with the 'Configuration change count' (see Identify FAL PDU Table 6) to allow a host application to monitor the configuration of the field device. The 'Configuration change counter' is a 16-bit counter that shall be incremented once for every command received that changes the devices configuration. The commands that change the configuration are shown in Table A.1. The value of the 'Configuration changed' bits and 'Configuration change counter' shall be maintained even if power is removed from the device or a device reset is performed.

If 'Configuration changed' is '1', then the client should read other variables from the slave device, if necessary and then use 'Reset configuration changed flag' service (IEC 62591:2010, 7.3.2.4.18) to reset this status bit.

Table A.1 – Commands that cause configuration change

Command number	Command	Reference
35	Write primary variable range	5.3.26
44	Write primary variable unit	5.3.28
45	Trim loop current zero	5.3.29
46	Trim loop current gain	5.3.30
51	Write dynamic variable assignment	5.3.32
59	Write number of response preambles	5.3.33
79	Write device variable	IEC 62591:2010, 8.2.3.25

Command number	Command	Reference
82	Write device variable trim point	5.3.36
83	Reset device variable trim	5.3.37
103	Write publish data period	IEC 62591:2010, 8.2.3.27
104	Write publish data trigger	IEC 62591:2010, 8.2.3.28
107	Write publish data device variables	IEC 62591:2010, 8.2.3.31
108	Write publish data mode command number	IEC 62591:2010, 8.2.3.32
109	Write publish data mode control	IEC 62591:2010, 8.2.3.33

A.4 Cold start

A field device shall contain two 'Cold start' bits – separate bits for the primary and the secondary master. Both bits shall be set to '1' when the field device is powered up or after a device reset. The first request APDU received by the device from a primary or secondary master shall reset the corresponding 'Cold start' bit to '0'.

A.5 More status available

'More status available' bit shall be set to '1' when there is more diagnostics information available in the device. If this bit is '1', then the host application should use 'Read additional device status' service (IEC 62591:2010, 7.3.2.4.21) to get the information not included in the 'Application process status'. When this bit is '1', the measurements may still be correct and suitable for use by control systems.

Since setting 'More status available' to '1' will cause most host applications to use "Read additional device status" service, thus decreasing available communication capacity, field device designers should consider which diagnostics is used to set this bit.

A.6 Loop current fixed

When the Loop current is directly (e.g., using Command 40 as specified in 5.3.27) or indirectly (e.g., using Command 79 as specified in IEC 62591:2010, 8.2.3.25 Write device variable FAL PDU) fixed and does not respond to the process changes, the 'Loop current fixed' bit shall be set to '1'. Setting this bit indicates to host applications that the Loop current should not be used for analog signalling of the Primary variable between the host control system and the field device.

NOTE A field device may be in multi-drop with the Loop current still responsive to changing process conditions (see 5.3.5 Write loop configuration FAL PDU and Table C.6).

A.7 Loop current saturated

The Loop current varies from 4,00 to 20,0 mA. The Upper and Lower range values set the 4,00 mA and 20,0 mA points (see 5.3.26 and 5.3.36). Most field devices allow the loop current to exceed these endpoints by some small amount. If any of this electrical limit established by the field device is exceeded, the 'Loop current saturated' bit shall be set. For transmitter, its analog output becomes saturated. For actuator, its input is over or under range. Figure A.1 shows the Loop current operation as it reaches saturation. The alarm level set by the device (e.g., when the Device malfunction bit is set to '1') shall be sufficiently different from the Loop current saturated levels to allow differentiation by devices monitoring the Loop current.

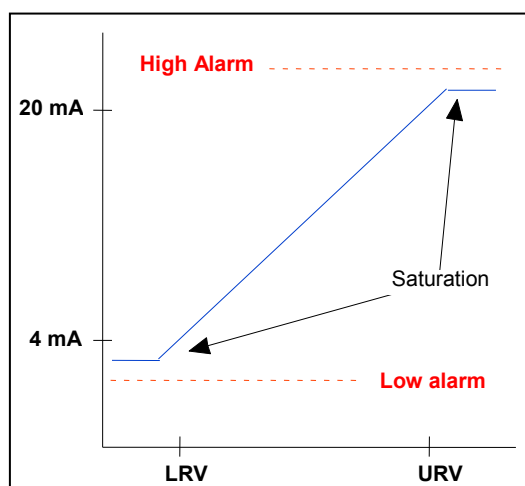


Figure A.1 – Loop current saturation and alarm levels

A.8 Non-primary variable out of limits

The process variable can be measured accurately as long as it is within the Upper and Lower transducer limits (see 5.3.11). The limits may be electrical (e.g., in a transmitter) or mechanical (e.g., actuator at its travel limit as indicated by a limit switch). When the process variable reaches one of these limits, a device variable may also reach its limit of measurement. If the device variable is outside of the Upper or Lower range Value, but within the Upper and Lower transducer limits, the device variable shall maintain valid value. If the device variable is outside the Upper or Lower transducer limit, and if that variable is not mapped into the PV, then 'Non-primary variable out of limits' bit shall be set to '1'.

A.9 Primary variable out of limits

If the device variable is outside the Upper or Lower transducer limit, and if that variable is mapped into the PV, then 'Primary variable out of limits' bit shall be set to '1'.

Annex B (normative)

Device and dynamic variable

B.1 Device variable

Device Variable is uniquely defined data item within a Field device that is always associated with process-related information as shown in Figure B.1. A Device variable's value changes as its connected process varies. A code number is assigned to each Device variable and this assignment shall never be changed for a given Device type. The device manufacturer shall specify these codes as shown in Table C.9. This table shall always include the Device variable code table found in the manufacturer's device-specific document. The number of Device variables available in a device is returned in Identity commands (see Identify FAL PDU Table 6). Device variables shall be numbered consecutively starting from zero (0). Each Device variable represents a direct or indirect connection between the process and the field device. An indirect connection infers the process state or level via calculations using direct process values from other Device variables, as shown for the Device variable number '4' in Figure B.1. In any case, a field device has a fixed number of Device variables. The Device variable family codes are specified in IEC 62591:2010, F.2.3. The Device variable classification codes are specified in IEC 62591:2010, F.2.4.

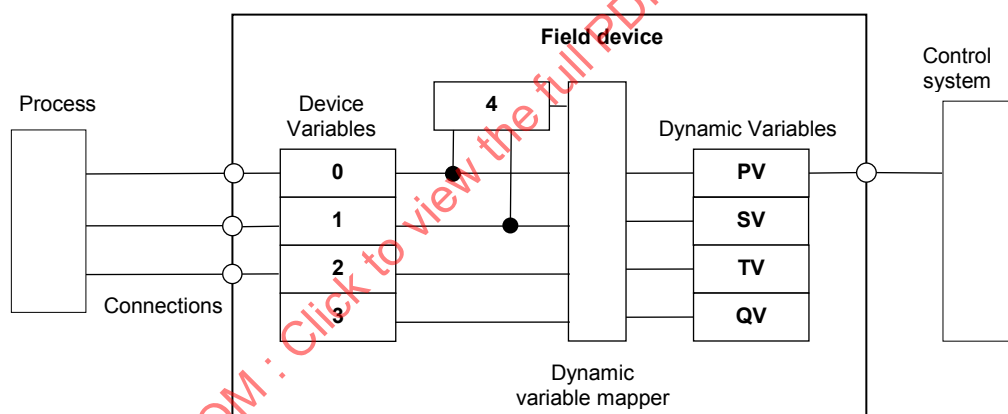


Figure B.1 – Device and Dynamic variables

B.2 Dynamic variable

Dynamic variable is a device variable that is assigned as the dynamic variable and possibly associated with an analog channel. Up to four of the Device variables can be mapped to the Dynamic variables using the appropriate Device variable number as shown in Figure B.1 (see dynamic variable assignment commands in 5.3.31 and 5.3.32). These dynamic variables are named as primary (PV), secondary (SV), tertiary (TV) and quaternary (QV) variable. A Field device shall have at least one dynamic variable – PV. The PV shall be connected to the first analog channel and this channel shall be used for digital signalling. The other three dynamic variables may also be connected to analog channels. The value of the dynamic variables can be read by using Command 1 (5.3.2), 2 (5.3.3) or 3 (5.3.4). The Dynamic variables are assigned Device variable codes as shown in Table C.9. Therefore, all devices shall support access to the Dynamic variables and the Loop current in all commands accessing Device variable data or properties.

NOTE The Dynamic variables allow access to process-related data via Commands that are required for all devices compliant to this standard. However, Device variables are accessed via optional Commands. This means that even though all devices have Device variables, the device developer may choose to not expose the underlying

Device variables. As a result, a simple device may support only the Dynamic variables and not implement the Commands that access Device variables.

B.3 Primary variable

Primary Variable refers the Device variable and analog channel associated with the Loop current. The commands that access the Primary Variable, allow convenient access to all configuration properties and process-related values connected with the Loop current. In addition, this association of the Loop current to the Primary variable allows the Loop current to serve as a communication channel that transmits the Primary variable's value from the field device to the measurement or control system.

The Primary Variable's attributes can be segregated into two domains, as shown in Figure B.2.

- Transducer: This domain characterizes the connection between the field device and the process. The attributes of this domain are:
 - Upper transducer limit,
 - Lower transducer limit,
 - Transducer serial number,
 - Transducer trim points and
 - Damping factor.
- Analog Channel: This domain is used for conversion between the digital value and engineering units of the Device variable domain and the milliamp signal of the Current loop. This domain has two sub-domains.
 - Range conversion: This domain is used for conversion between the Percent range and the Primary variable's value. The attributes of this sub-domain are:
 - Upper Range Value,
 - Lower Range Value and
 - Transfer Function.
 - Data acquisition: This DAQ domain converts between the physical analog signal value that can be measured on the loop and the Loop current value returned in Command 2 (5.3.3) and Command 3 (5.3.4). The attributes of this sub-domain are:
 - Alarm code,
 - the DAQ's zero and gain and
 - Loop current damping.

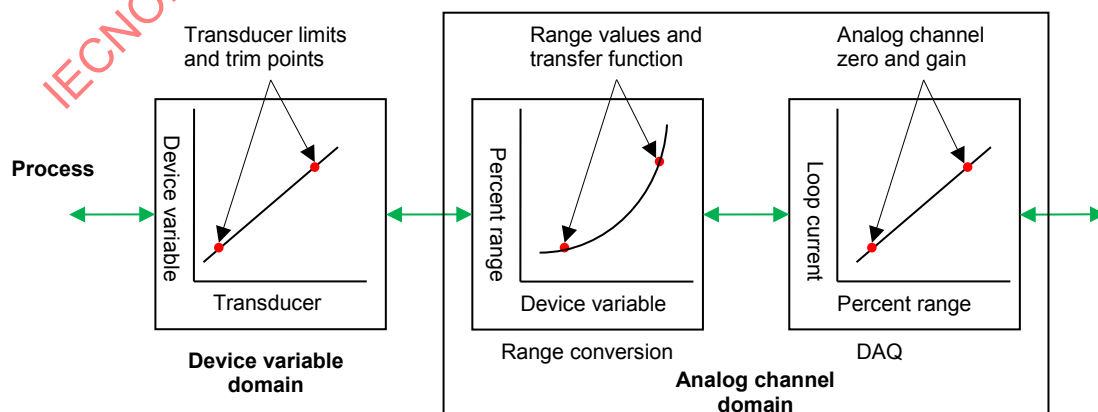


Figure B.2 – Primary variable domains

Although the analog signal is named as Loop current, it can be either current in mA unit or voltage in V unit. The type of signaling is specified in Table C.4. The Loop current shall always refer to the first analog channel in a Field device and it shall be the Primary variable.

B.4 Device variable classification

Device variables can be classified by the function performed (see IEC 62591:2010, F.2.4). The Device variable classification attribute can be read with Command 54 (see IEC 62591:2010, 8.2.3.24 Read device variable information FAL PDU). Once defined, the classification of Device variable shall not change for that device type. The Device variable classification indicates the type of process connection the Device variable characterizes and the engineering units that are supported by that Device variable. If a Device variable that does not use any of the classification defined by this standard, its classification shall be set to zero ("0") indicating that the base unit code table shall be used by the Client.

B.5 Device families

A device variable may belong to a specific device family – see IEC 62591:2010, F.2.3. Each device family has following characteristics.

- narrowly defined specified scope,
- the attributes for configuration or commissioning of the device variable,
- classification of the device variable attributes as optional or mandatory,
- group of the attributes into Read commands to optimize upload speeds,
- write commands for all attributes in the collection,
- recommended Standard Operating Procedure (SOP) for managing the device variable,
- collection of commands specific to the Device variable family and
- device variable status bit definition.

These Device family specific commands are for specific types of process connection or for specific process related functions.

NOTE 1 The device family specific commands are outside the scope of this standard. These commands allow the setup and parameterization of field devices without requiring device-specific commands.

NOTE 2 There are a limited number of commands and a large number of potential device families. Therefore, device family commands use Extended command numbers, which are two octets long. The most significant octet of this Extended command number is the Device variable Classification. Thus, there can be up to 256 commands per device family.

NOTE 3 If a device variable belongs to a device family, then the Field device supports all required commands for that device family.

B.6 Device variable status

All Device variables and Dynamic variables (cyclical process data) shall include a Device variable status octet as shown in Figure B.3. The various sub-fields are described below.

a) The bits 7 and 6 specify the Process data status:

- 11: Good,
- 01: Poor accuracy,
- 10: Manual or fixed,
- 00: Bad.

b) The bits 5 and 4 specify whether the Device variable value is limited (i.e., not responding to the process):

11: Constant,
 01: Low limited,
 10: High limited,
 00: Not limited.

- c) The bit 3 if '1' indicates that additional device family-specific status is available via the appropriate device family command.
- d) The bits 2, 1 and 0 indicate Device variable family specific status.

NOTE 1 If the Field device ever sets any of the least significant 4 bits then the Field device is expected to support Command 54 - see IEC 62591:2010, 8.2.3.24, and the appropriate device family documentation.

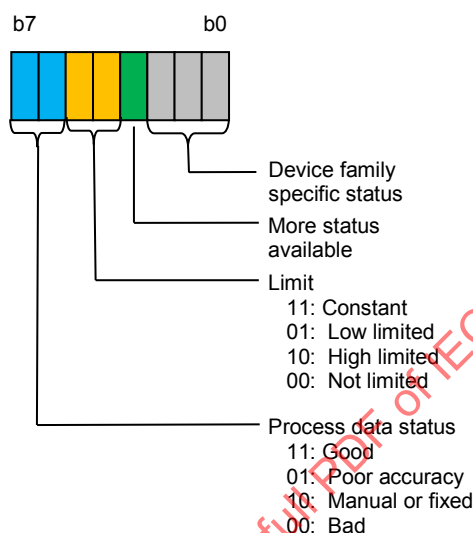


Figure B.3 – Device variable status

NOTE 2 The bits 7 to 4 provide useful status about the Device variable's value. For example, if the Process data status is 'Manual or fixed' and the Limit status is 'Not limited' then the value is being manually controlled. Limit status cannot be 'Constant' because the value can be changed at any time by the user.

Annex C (normative)

Common tables

C.1 Overview

C.1.1 General

The protocol of this standard uses lookup tables to associate specific definitions to numeric codes or to define the interpretation of an individual bit that is set to '1'. If the protocol references a table, the numeric values have specific definitions that allow all hosts to consistently utilize that data. This annex contains lookup tables (for example Device variable family and classification code, Standardized status codes etc.) used for Enumeration or Bit Field data types.

If one of the tables in this annex is referenced by the protocol specification, then the table or a subset of that table shall be used exactly as specified in this annex and undefined (i.e., unlisted) values shall not be used, unless specified otherwise in the definition of a table in this annex.

C.1.2 Enumeration

The Enumeration shall be as specified in IEC 62591:2010, F.1.2.

C.1.3 Bit Field

The Bit Field shall be as specified in IEC 62591:2010, F.1.3.

C.2 Table definitions

C.2.1 Publish mode control codes

The Publish mode control codes shall be as specified in IEC 62591:2010, F.2.1.

C.2.2 Write device variable codes

The Write device variable codes shall be as specified in IEC 62591:2010, F.2.2.

C.2.3 Device variable family codes

The Device variable family codes shall be as specified in IEC 62591:2010, F.2.3.

C.2.4 Device variable classification codes

The Device variable classification codes shall be as specified in IEC 62591:2010, F.2.4.

C.2.5 Analog channel saturated codes

The Analog channel saturated codes shall be as specified in IEC 62591:2010, F.2.5.

C.2.6 Analog channel fixed codes

The Analog channel fixed codes shall be as specified in IEC 62591:2010, F.2.6.

C.2.7 Standardized status 0 codes

The Standardized status 0 codes shall be as specified in IEC 62591:2010, F.2.7.

C.2.8 Standardized status 1 codes

The Standardized status 1 codes shall be as specified in IEC 62591:2010, F.2.8.

C.2.9 Standardized status 2 codes

The Standardized status 2 codes shall be as specified in IEC 62591:2010, F.2.9.

C.2.10 Standardized status 3 codes

The Standardized status 3 codes shall be as specified in IEC 62591:2010, F.2.10.

C.2.11 Publish trigger mode codes

The Publish trigger mode codes shall be as specified in IEC 62591:2010, F.2.11.

C.2.12 Transfer function codes

It is 1-octet enumeration data as shown in Table C.1.

Table C.1 – Transfer function codes

Code	Transfer function	Equation
0	Linear	$y = mx + b$
1	Square root	$y = \sqrt{x}$
2	Square root third power	$y = \sqrt{x^3}$
3	Square root fifth power	$y = \sqrt{x^5}$
4	Special curve	
5	Square	$y = x^2$
230	Discrete (Switch)	Binary (on/off)
231	Reserved	
232	Reserved	
233	Reserved	
240 - 249	Manufacturer specific definitions	
250	Not Used	
251	None	
252	Unknown	
253	Special	

C.2.13 Alarm Selection Codes

It is 1-octet enumeration data as shown in Table C.2.

Table C.2 – Alarm Selection codes

Code	Alarm Selection
0	High
1	Low
239	Hold last output value
240 - 249	Manufacturer specific definitions
250	Not Used
251	None
252	Unknown
253	Special

C.2.14 Write Protect Codes

It is 1-octet enumeration data as shown in Table C.3.

Table C.3 – Write Protect codes

Code	Write Protect
0	No - not write protected
1	Yes - write protected
250	Not Used
251	None
252	Unknown
253	Special

C.2.15 Physical layer signalling codes

It is 1-octet enumeration data as shown in Table C.4.

Table C.4 – Physical layer signalling codes

Code	Physical layer signalling
0	Bell 202 current
1	Bell 202 voltage
2	RS-485
3	RS-232
4	Wireless
6	Special

NOTE Only codes '0' and '1' apply to this standard.

C.2.16 Flag Assignment codes

It is 1-octet Bit field data as shown in Table C.5.

Table C.5 – Flag Assignment codes

Code	Flag Assignment
0x80	Reserved
0x40	Reserved
0x20, 0x10	Undefined
0x08	IEEE 802.15.4 2.4GHz DSSS with O-QPSK modulation
0x04	Protocol bridge device
0x02	EEPROM Control
0x01	Multi-sensor Field device

NOTE Only bit masks 0x01 and 0x02 apply to this standard.

C.2.17 Loop current mode codes

It is 1-octet enumeration data as shown in Table C.6 and it applies to the loop current signalling state of the device.

Table C.6 – Loop current mode codes

Code	Loop current mode
0	Disabled
1	Enabled

C.2.18 Trim point codes

It is 1-octet Bit field data as shown in Table C.7 and it indicates trim points that are supported by the Field device.

Table C.7 – Trim point codes

Code	Trim point
0	Reserved
1	Lower trim point supported
2	Upper trim point supported
3	Lower and upper trim point supported

C.2.19 Analog channel flag codes

It is 1-octet Bit field data as shown in Table C.8 and it indicates analog channel function.

Table C.8 – Analog channel flag codes

Code	Flag description
0x01	If this bit is '1' then the analog channel is the Field device input channel.

C.2.20 Device variable codes

The code for every Device variable available in a device shall be specified in this manufacturer's device-specific table. Every device shall include the codes and the variables

shown in Table C.9 and codes for all other device variables that can be accessed in the device.

NOTE The Device variables 244 to 249 shown in this table are not included in the count returned by "Maximum number of Device variables" in Command 0.

Table C.9 – Device variable codes

Code	Device variable
244	Percent range
245	Loop current
246	Primary variable
247	Secondary variable
248	Tertiary variable
249	Quaternary variable

C.2.21 Device profile codes

The Device profile codes shall be as specified in IEC 62591:2010, F.2.31.

IECNORM.COM : Click to view the full PDF of IEC 61158-6-20:2014

Annex D (normative)

Command requirements

D.1 General

A device can support new Commands that are specific to that Expanded device type. To insure consistent implementations, all commands shall comply with the requirements specified in this annex. These commands are referred to as device-specific commands.

NOTE The Commands specified in this standard comply with these requirements.

D.2 Stateless request and response

The commands shall be designed to be autonomous and allow stateless operation of the device's application layer. The execution of a command in master and slave shall not depend upon the execution of a prior command. The response to be supplied by a slave device shall be clearly and uniquely specified by the corresponding master request alone.

D.3 Read command

Read command shall not contain any data in the request Value field, other than that is required by the responding device to return the desired data items. The only data item that may be in the request is an index referencing an array entry in a field device (e.g., the device variable number). Read commands should have several related data items in the response Value field in order to minimize host upload times. Read command shall not change the operation of the field device in any way or change any data item stored in the field device.

D.4 Write command

Write command shall contain the identical data items in the request and response Value fields. The responding device shall return Value field of the response with the actual value of the data item as used by the device in the same engineering unit as in the request. Index data items and unit code shall not be stored or changed in the responding device as the result of a Write command. For every Write command, the Device shall support a Read command that can be used by a host client to determine the Device configuration and save it.

Unless otherwise stated in the Command specification of the Device, all data written in a Device by using a Write command shall be retained through the Device reset, Self test or removal of power from that device.

D.5 Action command

A Command to perform an action may or may not have data items in the request or response Value field. Action Command may affect the operation or configuration of the responding device, or data items in that device.

Unless otherwise stated in the Command Specification, all device operations or configurations affected by an Action command shall be retained through the Device reset, Self test or removal of power from that device.

D.6 Indexed command

Command shall contain indices allowing access to arrays or tables of data stored in a Device. The index shall be an unsigned integer. This allows single command access to an array of data. The number and the type of data items and the sequential order of the data items in the response shall be same as in the request, for all possible values of the index. An indexed command shall be specified with a rigid packet of information and, for each index value, the structure of this packet shall be identical. There shall be only one set of command-specific Response codes for all values of the index.

D.7 Multi-transaction command

Multi-transaction command has a sub command number in the request and response Value field to increase the number of device-specific commands. Multi-transaction command should be used only if a Device uses all of the allowed set of device-specific commands. All transactions shall perform the same Read, Write or Action operation (see Clause D.3, D.4 and D.5). Unlike indexed commands, the number and the type of data items in the request and response Value field can vary by transaction number. As a result, the command specification shall include a separate command-specific Response code specification for each transaction. Both the request and response Value field shall include the transaction number so that it can be stateless transaction as specified in Clause D.2.

IECNORM.COM : Click to view the full PDF of IEC 61158-6-20:2014

Bibliography

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

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

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

ISO/IEC 8825-1, *Information technology – ASN.1 encoding rules: Specification of Basic Encoding Rules (BER), Canonical Encoding Rules (CER) and Distinguished Encoding Rules (DER)*

HCF_SPEC-183, *Common Tables Specification*
available at <www.hartcomm.org>

IECNORM.COM : Click to view the full PDF of IEC 61158-6-20:2014

SOMMAIRE

AVANT-PROPOS	83
INTRODUCTION.....	85
1 Domaine d'application	86
2 Références normatives.....	86
3 Termes, définitions, symboles, abréviations et conventions	87
3.1 Termes et définitions provenant d'autres normes ISO/CEI	87
3.2 Termes de la CEI 61158-1	88
3.3 Définitions relatives à la couche Application de bus de terrain de Type 20	90
3.4 Abréviations et symboles.....	93
3.5 Conventions	94
3.6 Conventions utilisées dans les diagrammes d'états	95
4 Syntaxe abstraite	96
5 Syntaxe de transfert	96
5.1 Champs d'unité APDU communs	96
5.2 Structure d'unité APDU commune	98
5.3 Structures d'unité APDU spécifique à un service d'application d'appareil	101
5.4 Règles de codage des données.....	130
6 Procédures communes	135
6.1 Réponse différée	135
6.2 Procédure de mode Publish.....	135
7 Diagrammes d'états de protocole de la couche FAL.....	135
7.1 Généralités.....	135
7.2 Mapping de point AREP avec la couche Liaison de données	136
7.3 Machine ARPM Client.....	137
7.4 Machine ARPM Serveur.....	139
7.5 Fonctions utilisées par les diagrammes d'états FAL	140
Annexe A (normative) État de processus d'application	142
A.1 Généralités.....	142
A.2 Device malfunction	142
A.3 Configuration changed	142
A.4 Cold start	143
A.5 More status available	143
A.6 Loop current fixed	143
A.7 Loop current saturated	143
A.8 Non-primary variable out of limits	144
A.9 Primary variable out of limits	144
Annexe B (normative) Variable d'appareil et variable dynamique	145
B.1 Variable d'appareil	145
B.2 Variable dynamique.....	146
B.3 Variable primaire.....	146
B.4 Classification de variable d'appareil	147
B.5 Familles d'appareils	148
B.6 État de variable d'appareil.....	148
Annexe C (normative) Tables communes	150
C.1 Présentation	150

C.1.1	Généralités	150
C.1.2	Enumeration	150
C.1.3	Bit Field	150
C.2	Définitions des tables	150
C.2.1	Codes de commande du mode édition (Publishmode control codes)	150
C.2.2	Write device variable codes	150
C.2.3	Codes de famille de variables Device	150
C.2.4	Codes de classification de variables Device	150
C.2.5	Codes saturés de voie analogique	151
C.2.6	Codes fixes de voie analogique	151
C.2.7	Codes Standardized status 0	151
C.2.8	Codes Standardized status 1	151
C.2.9	Codes Standardized status 2	151
C.2.10	Codes Standardized status 3	151
C.2.11	Codes de mode déclenchement d'édition	151
C.2.12	Codes de fonction de transfert	151
C.2.13	Codes de sélection d'alarme	152
C.2.14	Codes de protection en écriture	152
C.2.15	Codes de signalisation de couche physique	152
C.2.16	Codes d'attribution d'indicateur	153
C.2.17	Codes de mode en courant de ligne	153
C.2.18	Codes de point d'ajustement	153
C.2.19	Codes d'indicateur de voie analogique	153
C.2.20	Codes de variable d'appareil	154
C.2.21	Codes de profil d'appareil	154
Annexe D (normative)	Exigences relatives aux commandes	155
D.1	Généralités	155
D.2	Demande et réponse sans état	155
D.3	Commande Read (lecture)	155
D.4	Commande Write (écriture)	155
D.5	Commande Action	155
D.6	Commande Indexed (Indexation)	156
D.7	Commande Multi-transaction	156
Bibliographie		157
Figure 1	Unité APDU de demande	99
Figure 2	Unité APDU de réponse normale	99
Figure 3	Réponse d'erreur de commande de l'esclave au maître	100
Figure 4	Unité APDU de commande Aggregated	101
Figure 5	Codage sans identification	130
Figure 6	Codage des données de type Integer	130
Figure 7	Codage des données de type Integer16	130
Figure 8	Codage des données de type Unsigned	131
Figure 9	Codage des données de type Unsigned16	131
Figure 10	Codage des données de type Floating Point simple précision	131
Figure 11	Codage des données de type Floating Point double précision	132
Figure 12	Codage des données de type Date	132

Figure 13 – Diagramme d'états Client	137
Figure 14 – Diagramme d'états Serveur	139
Figure A.1 – Saturation de courant de ligne et niveaux d'alarme	144
Figure B.1 – Variables d'appareil et variables dynamiques.....	145
Figure B.2 – Domaines de variable primaire.....	147
Figure B.3 – État de variable d'appareil	149
Tableau 1 – Conventions utilisées dans les diagrammes d'états	95
Tableau 2 – Valeurs du code de réponse	97
Tableau 3 – Valeurs d'état de processus d'application	97
Tableau 4 – Valeurs de l'état étendu.....	98
Tableau 5 – Unité APDU de demande Identify	101
Tableau 6 – Champ de valeur de la réponse Identify.....	102
Tableau 7 – Codes de réponse spécifiques à la commande Identify.....	102
Tableau 8 – Champ de valeur de la réponse Read primary variable.....	103
Tableau 9 – Codes de réponse spécifiques à la commande Read primary variable	103
Tableau 10 – Champ de valeur de Read loop current and percent of range	103
Tableau 11 – Codes de réponse spécifiques à la commande Read loop current and percent of range	104
Tableau 12 – Champ de valeur de Read dynamic variables and loop current.....	104
Tableau 13 – Codes de réponse spécifiques à la commande Read dynamic variables and loop current.....	104
Tableau 14 – Champ de valeur de Write loop configuration.....	105
Tableau 15 – Codes de réponse spécifiques à la commande Write loop configuration	105
Tableau 16 – Champ de valeur de Read loop configuration	106
Tableau 17 – Codes de réponse spécifiques à la commande Read loop configuration	106
Tableau 18 – Champ de valeur de Read dynamic variable families classifications	106
Tableau 19 – Codes de réponse spécifiques à la commande Read dynamic variable families classifications	106
Tableau 20 – Champ de valeur de la demande Read device variables with status	107
Tableau 21 – Champ de valeur de Read device variables with status.....	107
Tableau 22 – Codes de réponse spécifiques à la commande Read device variables with status	109
Tableau 23 – Champ de valeur de la réponse Read message	110
Tableau 24 – Codes de réponse spécifiques à la commande Read message	110
Tableau 25 – Champ de valeur de la réponse Read tag, descriptor, date.....	110
Tableau 26 – Codes de réponse spécifiques à la commande Read tag, descriptor, date	111
Tableau 27 – Champ de valeur de la réponse Read primary variable transducer information.....	111
Tableau 28 – Codes de réponse spécifiques à la commande Read primary variable transducer information	111
Tableau 29 – Champ de valeur de la réponse Read device information.....	112
Tableau 30 – Codes de réponse spécifiques à la commande Read device information.....	112
Tableau 31 – Champ de valeur de la réponse Read final assembly number	113

Tableau 32 – Codes de réponse spécifiques à la commande Read final assembly number	113
Tableau 33 – Champ de valeur de Write message	113
Tableau 34 – Codes de réponse spécifiques à la commande Write message	113
Tableau 35 – Champ de valeur de Write tag, descriptor, date	114
Tableau 36 – Codes de réponse spécifiques à la commande Write tag, descriptor, date	114
Tableau 37 – Champ de valeur de Write final assembly number	114
Tableau 38 – Codes de réponse spécifiques à la commande Write final assembly number	114
Tableau 39 – Champ de valeur de la réponse Read long tag	115
Tableau 40 – Codes de réponse spécifiques à la commande Read long tag	115
Tableau 41 – Champ de valeur de Write long tag	115
Tableau 42 – Codes de réponse spécifiques à la commande Write long tag	115
Tableau 43 – Champ de valeur de Write primary variable range	116
Tableau 44 – Codes de réponse spécifiques à la commande Write primary variable range	117
Tableau 45 – Champ de valeur de la demande Enter-exit fixed current mode	118
Tableau 46 – Champ de valeur de la réponse Enter-exit fixed current mode	118
Tableau 47 – Codes de réponse spécifiques à la commande Enter-exit fixed current mode	118
Tableau 48 – Champ de valeur de Write primary variable unit	119
Tableau 49 – Codes de réponse spécifiques à la commande Write primary variable unit	119
Tableau 50 – Champ de valeur de la demande Trim loop current zero	119
Tableau 51 – Codes de réponse spécifiques à la commande Trim loop current zero	120
Tableau 52 – Champ de valeur de la demande Trim loop current gain	120
Tableau 53 – Codes de réponse spécifiques à la commande Trim loop current gain	121
Tableau 54 – Champ de valeur de la réponse Read dynamic variable assignment	121
Tableau 55 – Codes de réponse spécifiques à la commande Read dynamic variable assignment	122
Tableau 56 – Champ de valeur de Write dynamic variable assignment	122
Tableau 57 – Codes de réponse spécifiques à la commande Write dynamic variable assignment	122
Tableau 58 – Champ de valeur de Write number of response preambles	123
Tableau 59 – Codes de réponse spécifiques à la commande Write number of response preambles	123
Tableau 60 – Champ de valeur de la demande Read device variable trim points	124
Tableau 61 – Champ de valeur de la réponse Read device variable trim points	124
Tableau 62 – Codes de réponse spécifiques à la commande Read device variable trim points	124
Tableau 63 – Champ de valeur de la demande Read device variable trim guidelines	125
Tableau 64 – Champ de valeur de la réponse Read device variable trim guidelines	125
Tableau 65 – Codes de réponse spécifiques à la commande Read device variable trim points	125
Tableau 66 – Champ de valeur de Write device variable trim point	126
Tableau 67 – Codes de réponse spécifiques à la commande Write device variable trim point	126

Tableau 68 – Reset device variable trim champ de valeur	127
Tableau 69 – Codes de réponse spécifiques à la commande Reset device variable trim	127
Tableau 70 – Codes de réponse spécifiques à la commande Aggregated	129
Tableau 71 – Codage du type Date	132
Tableau 72 – Codage du type Enumeration sur un octet	133
Tableau 73 – Type Bit Field sur un octet	133
Tableau 74 – Jeu de caractères Packed ASCII	134
Tableau 75 – Jeu de caractères ISO Latin-1 partiel admis	135
Tableau 76 – Passages d'état de la machine Client	138
Tableau 77 – Passages d'état de la machine Serveur	140
Tableau 78 – Fonction FormReqApu	140
Tableau 79 – Fonction Command	140
Tableau 80 – Fonction CommErr	140
Tableau 81 – Fonction RespCode	141
Tableau 82 – Fonction Commcode	141
Tableau 83 – Fonction ApStatus	141
Tableau 84 – Fonction Value	141
Tableau A.1 – Commandes modifiant la configuration	142
Tableau C.1 – Codes de fonction de transfert	151
Tableau C.2 – Codes de sélection d'alarme	152
Tableau C.3 – Codes de protection en écriture	152
Tableau C.4 – Codes de signalisation de couche physique	152
Tableau C.5 – Codes d'attribution d'indicateur	153
Tableau C.6 – Codes de mode en courant de ligne	153
Tableau C.7 – Codes de point d'ajustement	153
Tableau C.8 – Codes d'indicateur de voie analogique	153
Tableau C.9 – Codes de variable d'appareil	154

COMMISSION ÉLECTROTECHNIQUE INTERNATIONALE

**RÉSEAUX DE COMMUNICATION INDUSTRIELS –
SPÉCIFICATIONS DES BUS DE TERRAIN –****Partie 6-20: Spécification du protocole de la couche application –
Éléments de type 20**

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 elle-même ne fournit aucune attestation de conformité. Des organismes de certification indépendants fournissent des services d'évaluation de conformité et, dans certains secteurs, accèdent aux marques de conformité de la CEI. La CEI n'est responsable d'aucun des services effectués par les organismes de certification indépendants.
- 6) Tous les utilisateurs doivent s'assurer qu'ils sont en possession de la dernière édition de cette publication.
- 7) Aucune responsabilité ne doit être imputée à la CEI, à ses administrateurs, employés, auxiliaires ou mandataires, y compris ses experts particuliers et les membres de ses comités d'études et des Comités nationaux de la CEI, pour tout préjudice causé en cas de dommages corporels et matériels, ou de tout autre dommage de quelque nature que ce soit, directe ou indirecte, ou pour supporter les coûts (y compris les frais de justice) et les dépenses découlant de la publication ou de l'utilisation de cette Publication de la CEI ou de toute autre Publication de la CEI, ou au crédit qui lui est accordé.
- 8) L'attention est attirée sur les références normatives citées dans cette publication. L'utilisation de publications référencées est obligatoire pour une application correcte de la présente publication.
- 9) L'attention est attirée sur le fait que certains des éléments de la présente Publication de la CEI peuvent faire l'objet de droits de brevet. La CEI ne saurait être tenue pour responsable de ne pas avoir identifié de tels droits de brevets et de ne pas avoir signalé leur existence.

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

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

La Norme internationale CEI 61158-6-20 a été établie par le sous-comité 65C: Réseaux industriels, du comité d'études 65 de la CEI: Mesure, commande et automation dans les processus industriels.

Cette troisième édition annule et remplace la deuxième édition parue en 2010. Cette édition constitue une révision technique.

Les modifications majeures par rapport à l'édition précédente sont les suivantes:

- a) ajout d'un protocole pour les nouveaux services ajoutés à la CEI 61158-5-20;
- b) ajout d'annexes normatives;
- c) mise à jour des références normatives, termes, définitions, symboles, abréviations;
- d) correction des erreurs rédactionnelles et syntaxiques.

Le texte de cette norme est issu des documents suivants:

FDIS	Rapport de vote
65C/764/FDIS	65C/774/RVD

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

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

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

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

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

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

INTRODUCTION

La présente partie de la norme CEI 61158 s'inscrit dans une série créée pour faciliter l'interconnexion des composants de systèmes d'automatisation. Elle renvoie aux autres normes de l'ensemble défini par le modèle de référence de bus de terrain "à trois couches" décrit dans la CEI 61158-1.

Le protocole d'application fournit le service d'application au moyen des services disponibles au niveau de la couche Liaison de données ou de la couche immédiatement inférieure. Le principal objectif de la présente norme est de définir un ensemble de règles de communication, exprimées en termes de procédures que doivent suivre les entités d'application (*Application Entity*, AE) homologues au moment de la communication. Ces règles de communication visent à fournir une base saine pour le développement, dans divers buts:

- en tant que guide pour les développeurs et les concepteurs;
- dans une optique d'utilisation lors des essais et de l'achat de matériel;
- dans le cadre d'un accord pour l'admission de systèmes dans l'environnement de systèmes ouverts;
- en tant que précision apportée à la compréhension des communications prioritaires dans le modèle OSI.

Cette norme traite, en particulier, de la communication et de l'interfonctionnement des capteurs, effecteurs et autres appareils d'automatisation. L'utilisation conjointe de la présente norme avec d'autres normes entrant dans les modèles de référence OSI ou de bus de terrain permet à des systèmes qui ne pourraient pas, sans cela, de fonctionner ensemble dans toute combinaison.

RÉSEAUX DE COMMUNICATION INDUSTRIELS – SPÉCIFICATIONS DES BUS DE TERRAIN –

Partie 6-20: Spécification du protocole de la couche application – Eléments de type 20

1 Domaine d'application

La couche Application de bus de terrain (*Fieldbus Application Layer*, FAL) procure aux programmes de l'utilisateur un moyen d'accès à l'environnement de communication des bus de terrain. À cet égard, la FAL peut être considérée comme une "fenêtre entre programmes d'application correspondants".

La présente norme fournit des éléments communs pour les communications prioritaires ou non prioritaires entre des programmes d'application dans un environnement et avec un matériel d'automatisation spécifiques aux bus de terrain de Type 20. Le terme "communication prioritaire" signale l'existence d'une fenêtre temporelle dans laquelle est exigée la réalisation d'une ou de plusieurs actions spécifiées, avec un niveau de certitude défini. La non-réalisation des actions spécifiées dans la fenêtre temporelle induit un risque de défaillance des applications qui demandent ces actions, avec les risques afférents pour l'équipement, les installations et éventuellement la vie humaine.

La présente norme définit de manière abstraite le comportement assuré par la couche Application de bus de terrain de Type 20, en termes

- a) de syntaxe abstraite définissant les unités de données de protocole de couche Application, transmises entre les entités d'application en communication;
- b) de syntaxe de transfert définissant les unités de données de protocole de couche Application, transmises entre les entités d'application en communication;
- c) de diagrammes d'états de contexte d'application définissant le comportement de service d'application visible entre les entités d'application en communication; et
- d) de diagrammes d'états de relations entre applications définissant le comportement de communication visible entre les entités d'application en communication.

La présente norme vise à définir le protocole mis en place pour définir

- a) la représentation filaire des primitives de service définies dans la CEI 61158-5-20; et
- b) le comportement, visible par un observateur externe, associé à leur transfert.

La présente norme spécifie le protocole de la couche Application de bus de terrain CEI de Type 20, en conformité avec le modèle de référence de base OSI (ISO/CEI 7498-1) et avec la structure de la couche Application OSI (ISO/CEI 9545).

2 Références normatives

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

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

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

CEI 61158-5-20:2014, *Réseaux de communication industriels – Spécifications des bus de terrain – Partie 5-20: Définition des services de la couche application – Eléments de type 20*

CEI 62591:2010, *Réseaux de communication industriels – Réseau de communications sans fil et profils de communication – WirelessHART™*

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

ISO/IEC 8824-1, *Information technology – Abstract Syntax Notation One (ASN.1): Specification of basic notation* (disponible en anglais seulement)

ISO/IEC 8859-1, *Information technology – 8-bit single-byte coded graphic character sets – Part 1: Latin alphabet No. 1* (disponible en anglais seulement)

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

ISO/IEC/IEEE 60559, *Information technology – Microprocessor Systems – Floating-Point arithmetic* (disponible en anglais seulement)

IEEE 802.15.4: *IEEE Standard for Local and metropolitan area networks – Part 15.4: Low-Rate Wireless Personal Area Networks (LR-WPANs)*

3 Termes, définitions, symboles, abréviations et conventions

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

3.1 Termes et définitions provenant d'autres normes ISO/CEI

3.1.1 Termes et définitions de l'ISO/CEI 7498-1

- a) syntaxe abstraite
- b) entité d'application
- c) processus d'application
- d) unité de données de protocole d'application
- e) élément de service d'application
- f) invocation d'entités d'application
- g) invocation de processus d'application
- h) syntaxe de transfert

3.1.2 Termes et définitions de l'ISO/CEI 9545

- a) invocation d'entités d'application
- b) invocation de processus d'application
- c) élément de service d'application

3.1.3 Termes et définitions de l'ISO/CEI 8824-1

- a) identifiant d'objet
- b) type

- c) valeur
- d) type simple
- e) type structuré
- f) type composant
- g) indicateur (ou balise)
- h) true (vrai)
- i) false (faux)
- j) type entier
- k) type chaîne d'octets
- m) type null (nul)

3.1.4 Termes et définitions de l'ISO/CEI 8825-1

- a) codage (d'une valeur de donnée)
- b) valeur de données
- c) octet d'identification (la présente norme utilise le singulier)
- d) octet(s) de longueur (la présente norme utilise à la fois le singulier et le pluriel)
- e) octets de contenu

3.2 Termes de la CEI 61158-1

3.2.1

application

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

3.2.2

objet d'application

classe d'objets qui gère et assure un échange de messages pendant le mode exécution à travers le réseau et à l'intérieur de l'appareil de réseau

Note 1 à l'article: De nombreux types de classes d'objets d'application peuvent être définis.

3.2.3

processus d'application

partie d'une application répartie sur un réseau, située sur un appareil et comportant une adresse non ambiguë

3.2.4

objet du processus d'application

composant d'un processus d'application, identifiable et accessible par une relation entre applications FAL

Note 1 à l'article: Les définitions de l'objet du processus d'application sont constituées d'un ensemble de valeurs pour les attributs de leur classe (voir la définition de la Définition de classe d'objets du processus d'application). Les définitions de l'objet du processus d'application peuvent être accessibles à distance au moyen des services de l'ASE de gestion des objets FAL. Ces services peuvent être utilisés pour charger ou actualiser les définitions d'objets, pour lire les définitions d'objets et pour créer et supprimer de manière dynamique ces mêmes objets et leurs définitions correspondantes.

3.2.5

classe d'objets de processus d'application

classe des objets du processus d'application définis en termes d'ensemble de leurs attributs et services accessibles par le réseau

3.2.6**relation entre applications**

association coopérative entre deux ou plusieurs invocations d'entités d'application (application-entity-invocation) à des fins d'échange d'informations et de coordination de leur fonctionnement conjoint

Note 1 à l'article: Cette relation est activée soit par l'échange d'unités de données de protocole d'application, soit à la suite d'activités de préconfiguration.

3.2.7**point d'extrémité de relation entre applications**

contexte et comportement d'une relation entre applications tels que vus et maintenus par l'un des processus d'application impliqués dans la relation entre applications

Note 1 à l'article: Chaque processus d'application impliqué dans la relation entre applications maintient son propre point d'extrémité de relation entre applications.

3.2.8**attribut**

description d'une caractéristique ou fonction visible de l'extérieur d'un objet

Note 1 à l'article: Les attributs d'un objet contiennent des informations relatives à des parties variables d'un objet. Typiquement, ils fournissent des informations de statut ou régissent le fonctionnement d'un objet. Des attributs peuvent aussi avoir une incidence sur le comportement d'un objet. Les attributs se répartissent en attributs de classes et attributs d'instances.

3.2.9**comportement**

indication de la façon dont l'objet réagit à des événements particuliers

Note 1 à l'article: Sa description inclut la relation entre les valeurs et services d'attributs.

3.2.10**classe**

ensemble d'objets, qui représentent tous le même type de composant système

Note 1 à l'article: Une classe est une généralisation de l'objet; un modèle pour définir des variables et des méthodes. Tous les objets dans une classe ont une forme et un comportement identiques, mais contiennent en général des données différentes dans leurs attributs.

3.2.11**attribut de classe**

attribut qui est partagé par tous les objets au sein de la même classe

3.2.12**code de classe**

identifiant unique attribué à chaque classe d'objets

3.2.13**service spécifique à une classe**

service défini par une classe d'objets particulière pour accomplir une fonction requise qui n'est pas accomplie par un service commun

Note 1 à l'article: Un objet spécifique à une classe est unique pour la classe d'objets qui le définit.

3.2.14**client**

- (a) objet qui utilise les services d'un autre objet (serveur) pour accomplir une tâche
- (b) initiateur d'un message auquel un serveur réagit, comme par exemple un point d'extrémité d'AR qui émet des APDU de demande de service confirmées vers un point d'extrémité d'AR unique agissant comme serveur

3.2.15

erreur de communication

erreur détectable à la réception d'une PhPDU ou DLPDU, également octet 'Code d'erreur de communication' de l'unité APDU

3.2.16

trajet d'acheminement

flux unidirectionnel d'APDU dans le cadre d'une relation entre applications

3.2.17

cyclique

terme utilisé pour décrire les événements qui se répètent de manière régulière et répétitive

3.2.18

point d'extrémité

une des entités de communication impliquées dans une connexion

3.2.19

erreur

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

3.2.20

code d'erreur

identification d'un type spécifique d'erreur dans une classe d'erreurs

3.2.21

informations de gestion

informations accessibles sur le réseau qui prennent en charge la gestion du fonctionnement du système de bus de terrain, y compris la couche application

Note 1 à l'article: La gestion comprend des fonctions telles que le contrôle, la surveillance et le diagnostic.

3.2.22

serveur

- (a) rôle d'un AREP dans lequel il retourne une APDU de réponse de service confirmée au client qui a émis la demande
- (b) objet qui fournit des services à un autre objet (client)

3.2.23

service

opération ou fonction réalisée par un objet et/ou une classe d'objets sur demande d'un autre objet et/ou d'une autre classe d'objets

Note 1 à l'article: Un ensemble de services communs est défini et des dispositions sont prévues pour la définition des services spécifiques à l'objet. Les services spécifiques à l'objet sont ceux définis par une classe d'objets particulière afin d'exécuter une fonction requise non exécutée par un service commun.

3.3 Définitions relatives à la couche Application de bus de terrain de Type 20

3.3.1

voie analogique

signal électrique à variation continue permettant de relier un appareil de terrain au reste du système d'acquisition ou de contrôle des données

Note 1 à l'article: Certains appareils de terrain prennent en charge plusieurs voies analogiques (entrée ou sortie). Chaque voie analogique transmet une seule variable dynamique vers ou en provenance de l'appareil de terrain.

3.3.2**diffusion**

processus de transmission d'une PDU (unité de données de protocole) à tous les appareils qui sont connectés au réseau et qui sont en mesure de recevoir la transmission

3.3.3**adresse de diffusion**

adresse utilisée par un maître pour transmettre une commande à tous les appareils

3.3.4**mode en rafale**

initiation de l'activité de communication par un appareil esclave à intervalles cycliques sans demande de la part d'un maître

3.3.5**occupé**

état d'un appareil qui est actif sur différentes tâches et ne peut exécuter une commande en même temps

3.3.6**erreur de communication**

erreur détectable lors de la réception d'une PhPDU ou DLPDU, également l'octet du "code d'erreur de communication" de l'APDU

3.3.7**appareil**

toute entité contenant une mise en œuvre de bus de terrain de Type 20

3.3.8**ID d'appareil**

numéro de série d'un appareil qui est unique parmi toutes les instances d'un type d'appareil

Note 1 à l'article: Le fabricant est tenu d'attribuer une valeur unique pour chaque appareil dont les valeurs sont identiques pour l'ID du fabricant et le type d'appareil.

3.3.9**type d'appareil**

type de fabrication d'un appareil, par exemple son nom de produit

Note 1 à l'article: La valeur de cet attribut est unique parmi tous les fabricants et tous les types d'appareils. Sa valeur spécifie l'ensemble des commandes et des objets de données pris en charge par l'appareil.

3.3.10**variable d'appareil**

élément de donnée à définition unique dans un appareil de terrain, qui est toujours associé aux informations relatives au processus

Note 1 à l'article: Une valeur de variable d'appareil varie suite à des modifications et des variations du processus auquel l'appareil est relié.

3.3.11**variable dynamique**

variable d'appareil qui est attribuée comme la valeur dynamique et éventuellement associée à une voie analogique

Note 1 à l'article: Un appareil peut contenir jusqu'à quatre variables - variables primaires, secondaires, tertiaires et quaternaires. Ces variables sont désignées collectivement par le terme "dynamiques".

3.3.12**type d'appareil étendu**

type de fabrication de l'appareil

Note 1 à l'article: La valeur de cet attribut est unique parmi tous les fabricants et tous les types d'appareils. Sa valeur spécifie l'ensemble des commandes et des objets de données pris en charge par l'appareil.

3.3.13

appareil de terrain

entité physique connectée au processus ou à l'équipement et possédant au moins un élément de signalisation qui communique avec un ou des autre(s) élément(s) de signalisation via un câble

Note 1 à l'article: Il est directement connecté au capteur ou à la grille d'entraînement ou il exécute la fonction de commande de processus et est directement connecté à la couche physique spécifiée dans la présente norme. Il peut générer ou recevoir un signal analogique en plus d'un signal numérique.

3.3.14

indicateur long (ou balise longue)

chaîne ISO Latin-1 restreinte à 32 caractères servant à identifier un appareil de terrain

3.3.15

courant de ligne

valeur mesurée par un mA en série avec l'appareil de terrain

Note 1 à l'article: Le courant de ligne est un signal 4 mA à 20 mA analogique en courant continu servant à communiquer une valeur unique entre le système de commande et l'appareil de terrain. Les appareils en mode tension utilisent "volts en courant continu" comme unités techniques lorsque les valeurs de "courant de ligne" sont utilisées.

3.3.16

ID du fabricant

chaîne identifiant le fabricant qui a produit un appareil

Note 1 à l'article: Le fabricant est tenu d'utiliser la valeur qui lui est attribuée et n'est pas autorisé à utiliser la valeur attribuée à un autre fabricant.

3.3.17

maître

appareil qui initie une activité de communication en envoyant une trame de demande à un autre appareil et en attendant une trame de réponse de la part de cet appareil

3.3.18

réseau

paire unique de câbles, connecteurs, éléments de signalisation associés au moyen desquels un ensemble d'appareils de signalisation sont interconnectés, et les éléments ne servant pas à la signalisation qui sont raccordés à cette même paire de câbles

Note 1 à l'article: Une installation qui utilise plusieurs paires de fils et une alimentation de réseau commune est considérée comme plusieurs réseaux.

3.3.19

adresse d'interrogation

identifiant attribué à un appareil de manière unique dans le réseau auquel l'appareil est connecté

Note 1 à l'article: L'adresse d'interrogation permet de construire une adresse codée sur un octet.

3.3.20

maître primaire

appareil maître qui peut toujours initier la communication

3.3.21

maître secondaire

appareil maître qui ne peut initier la communication que par un processus d'arbitrage et lorsque l'appareil maître a abandonné l'initiation de la communication

3.3.22**esclave**

appareil qui déclenche les activités de communication uniquement après avoir reçu une unité PDU de demande émise par un appareil maître et qui est tenu d'envoyer une réponse à cette demande

3.3.23**indicateur (ou balise)**

chaîne ASCII de 8 caractères servant à identifier un appareil de terrain

3.3.24**transaction**

échange de trames liées consécutives entre deux entités de contrôle d'accès au support homologues, nécessaires pour réussir la transmission

Note 1 à l'article: Une transaction comprend (a) une seule transmission PhPDU à partir d'un appareil source, ou (b) une PhPDU à partir de l'appareil source suivie par une deuxième PhPDU d'acquittement de niveau de liaison depuis l'appareil de destination.

3.3.25**ID unique**

identifiant attribué à un appareil, unique parmi toutes les instances des appareils conformes à la présente norme

3.4 Abréviations et symboles

AE	Entité d'application (<i>Application Entity</i>)
AL	Couche Application (<i>Application Layer</i>)
AP	Processus d'application (<i>Application Process</i>)
APDU	Unité de données de protocole d'application (<i>Application Protocol Data Unit</i>)
APO	Objet d'application (<i>Application Object</i>)
AR	Relation entre applications (<i>Application Relationship</i>)
AREP	Point d'extrémité de relation entre applications (<i>Application Relationship EndPoint</i>)
ARPM	Machine protocolaire de relations entre applications (<i>Application Relationship Protocol Machine</i>)
ASCII	American Standard Code for Information Interchange (<i>Code américain normalisé pour l'échange d'informations</i>)
ASE	Élément de service d'application (<i>Application Service Element</i>)
Cnf	Confirmation
DL-	Préfixe désignant la couche Liaison de données (<i>Data Link-</i>)
DLC	Connexion de couche Liaison de données (<i>Data-Link Connection</i>)
DLE	Entité DL (instance active locale de la couche Liaison de données) (<i>DL entity</i>)
DLL	Couche Liaison de données (<i>Data-Link Layer</i>)
DLM	Gestion de couche Liaison de données (<i>Data-Link-Management</i>)
DLME	Entité de gestion de couche Liaison d données (<i>Data-link-management entity</i>)
DLPDU	Unité de données de protocole de liaison de données (<i>Data link protocol data unit</i>)
DLSAP	Point d'accès au service de couche Liaison de données (<i>Data-Link Service Access Point</i>)
DLSDU	Unité de données de service de couche Liaison de données (<i>DL-Service-Data-Unit</i>)
DR	Réponse différée (<i>Delayed response</i>)

DRM	Mécanisme de réponse différée (<i>Delayed response mechanism</i>)
FAL	Couche Application de bus de terrain (<i>Fieldbus Application Layer</i>)
HCF	HART™ Communication Foundation
ID	Identifiant
CEI	Commission Electrotechnique Internationale
Ind	Indication
LRV	Valeur de plage inférieure (<i>Low range value</i>)
LSO	Octet de poids faible (<i>Least significant octet</i>)
MSO	Octet de poids fort (<i>Most significant octet</i>)
OSI	Interconnexion de systèmes ouverts (<i>Open Systems Interconnect</i>)
PDU	Unité de données de protocole (<i>Protocol data unit</i>)
PhL-	Couche physique (<i>Physical layer</i>) (comme préfixe)
PhPDU	Unité de données de protocole PhL (<i>PhL-protocol-data-unit</i>)
PV	Variable primaire (<i>Primary variable</i>)
QV	Variable quaternaire (<i>Quaternary variable</i>)
Req	Demande (<i>Request</i>)
Rsp	Réponse (<i>Response</i>)
SAP	Point d'accès au service (<i>Service Access Point</i>)
SDU	Unité de données de service (<i>Service Data Unit</i>)
SN	Bit de signe (<i>Sign bit</i>)
SOP	Procédure normale d'exploitation (<i>Standard Operating Procedure</i>)
SV	Variable secondaire (<i>Secondary variable</i>)
TV	Variable tertiaire (<i>Tertiary variable</i>)
URV	Valeur de plage supérieure (<i>Upper range value</i>)
VFD	Appareil de terrain virtuel (<i>Virtual Field Device</i>)

3.5 Conventions

3.5.1 Concept général

La couche FAL se compose d'un ensemble d'éléments ASE orientés objet. Chaque élément ASE est défini dans un paragraphe distinct. Chaque spécification d'élément ASE est divisée en trois parties: ses définitions de classe, ses services et la spécification de ses protocoles. Les deux premiers éléments sont contenus dans la CEI 61158-5-20. La spécification des protocoles pour chacun des éléments ASE est définie dans la présente norme.

Les définitions de classe définissent les attributs des classes prises en charge par chaque élément ASE. La spécification de services définit les services fournis par l'élément ASE.

3.5.2 Conventions relatives aux définitions de classe

Les définitions de classe sont décrites au moyen de modèles. Chaque modèle se compose d'une liste d'attributs associés à la classe. La forme générale des modèles est définie dans la CEI 61158-5-20, 3.5.2.

3.5.3 Conventions de syntaxe abstraite

Lorsque le paramètre "optionalParametersMap" est utilisé, un numéro de bit correspondant à chaque production OPTIONAL ou DEFAULT est donné en commentaire.

3.6 Conventions utilisées dans les diagrammes d'états

3.6.1 Diagramme d'états

La présente norme utilise un diagramme pour représenter un diagramme d'états. La convention suivante est appliquée.

- Les rectangles marqués représentent les états que peut prendre un composant.
- Les passages d'état sont représentés par des lignes droites. Ils indiquent l'état que quitte un composant et l'état vers lequel il passe.
- Un passage est marqué avec les événements qu'il a engendrés et les actions correspondantes (ceci peut être vide). Les événements (au-dessus de la ligne) engendrant le passage sont séparés des actions résultantes (sous la ligne) par un trait horizontal.
- Un tilde (" ~ ") indique le complément à un ou la négation.

3.6.2 Table de diagrammes d'états

Les diagrammes d'états sont décrits dans le Tableau 1.

Tableau 1 – Conventions utilisées dans les diagrammes d'états

N°	État courant	Événement / condition => action	État suivant
Nom du passage	L'état courant dans lequel se trouve le diagramme d'états lors de ce passage d'état	Événements ou conditions déclenchant ce passage d'état. => Actions effectuées lorsque les événements ou conditions ci-dessus sont réunis. Elles figurent toujours au-dessous des événements ou conditions	Etat suivant dans lequel passe le diagramme d'états une fois les actions effectuées

NOTE Le nom du passage ne figure pas dans toutes les spécifications de diagramme d'états.

Les conventions utilisées dans les diagrammes d'états sont les suivantes:

:= la valeur d'un élément, à gauche, est remplacée par la valeur d'un élément, à droite. Si un élément à droite est un paramètre, il provient de la primitive indiquée comme événement d'entrée.

xxx indique un nom de paramètre.

Exemple:

Identifieur := reason

signifie que la valeur d'un paramètre "reason" est attribuée à un paramètre appelé "Identifieur".

"xxx" indique une valeur fixe.

Exemple:

Identifieur := "abc"

signifie que la valeur "abc" est attribuée à un paramètre appelé "Identifieur".

= condition logique indiquant qu'un élément à gauche est égal à un élément à droite.

< condition logique indiquant qu'un élément à gauche est inférieur à l'élément à droite.

> condition logique indiquant qu'un élément à gauche est supérieur à l'élément à droite.

<> condition logique indiquant qu'un élément à gauche est différent d'un élément à droite.

&& indique le "ET" logique

|| indique le "OU" logique.

Cette construction permet d'exécuter une séquence d'actions en boucle au sein d'un passage. La boucle est exécutée pour toutes les valeurs comprises entre start_value et end_value.

Exemple:
 for (Identifiant := start_value to end_value)
 actions
endfor

Cette construction permet d'exécuter d'autres actions dépendant d'une condition (qui peut être la valeur d'un identifiant ou le résultat d'une action antérieure) au sein d'un passage.

Exemple:
 If (condition)
 actions
else
 actions
endif

4 Syntaxe abstraite

La syntaxe abstraite des unités APDU est associée à leur syntaxe de transfert; elle est spécifiée à l'Article 5.

5 Syntaxe de transfert

5.1 Champs d'unité APDU communs

5.1.1 Numéro de commande

Numéro de commande utilisé pour représenter l'identifiant numérique du service Read, Write ou Information report. Il s'agit d'un champ Unsigned8 qui identifie le type de service et l'objet dans l'esclave devant être atteint ou l'action devant être entreprise.

5.1.2 Numéro de commande étendu

Numéro de commande utilisé pour étendre l'identifiant numérique du service Read, Write ou Information report à la valeur Unsigned16. Si le numéro de commande étendu est utilisé, le champ de numéro de commande doit être défini sur '31'. Il s'agit d'un champ Unsigned16 qui identifie le type de service et l'objet dans l'esclave devant être atteint ou l'action devant être entreprise.

5.1.3 Nombre d'octets

Ce champ représente le nombre total d'octets dans tous les champs qui suivent ce champ. Le data type de ce champ est Unsigned8 et sa valeur peut être comprise entre '0' et '255'. Si ce champ prend la valeur zéro, il s'agit du dernier champ dans l'unité APDU.

5.1.4 Code de réponse

Le champ de code de réponse indique la réussite de la commande demandée, ou la réalisation de la commande demandée générant un avertissement ou une erreur, comme indiqué dans le Tableau 2. Certaines valeurs dépendent de la valeur du champ de commande. Chaque structure d'unité APDU spécifique à un service montre les valeurs de code de réponse attribuées à ce service. Un appareil doit utiliser les valeurs spécifiées dans l'unité APDU spécifique à un service. L'appareil peut répondre à l'aide d'une valeur inutilisée, pourvu qu'il indique une condition non définie dans la présente norme. Le champ du data type Enumeration contient un octet.

Tableau 2 – Valeurs du code de réponse

Valeur	Classe	Définition
0	Success	La commande (de lecture ou d'écriture) a été exécutée de manière correcte
1 – 127	Warning	La commande (d'écriture) a été exécutée avec l'écart décrit dans la réponse (par exemple, une valeur a été définie sur sa valeur conforme la plus proche).
1 – 127	Error	L'exécution de la commande ne s'est pas déroulée correctement. Le code de réponse indique la cause de l'erreur (par exemple, l'appareil est protégé en écriture).
1 – 127	Reserved	Un code peut être réservé pour utilisation future par le protocole défini par la présente norme. Il n'est pas permis qu'un appareil renvoie la valeur Reserved.

5.1.5 Valeur

Le champ de valeur représente les données d'utilisateur transférées entre la couche application et son utilisateur. L'ALE les assemble à partir des paramètres d'une primitive de service ou les décompose en paramètres appartenant à une primitive de service. Sa structure dépend du type d'unité APDU spécifique à un service.

5.1.6 État de processus d'application

Ce champ est fourni par l'utilisateur FAL. Le data type de ce champ est Bit Field sur un octet. Il convient que sa valeur soit définie par l'utilisateur FAL comme indiqué dans le Tableau 3, afin que toutes les applications utilisateur puissent interopérer avec les appareils de terrain conformes à la présente norme.

NOTE L'état de processus d'application est également appelé état de l'appareil, car il indique l'état de fonctionnement actuel de l'appareil de terrain dans son ensemble. Il n'est associé à la réussite d'aucune commande.

Tableau 3 – Valeurs d'état de processus d'application

Masque de bits	Nom	Description
0x80	Device malfunction	L'appareil a détecté une erreur ou une défaillance grave qui compromet son fonctionnement.
0x40	Configuration changed	une action qui a modifié la configuration de l'appareil a été exécutée.
0x20	Cold start	une défaillance électrique ou une réinitialisation de l'appareil est survenue.
0x10	More status available	d'autres informations d'état sont accessibles via une autre commande de lecture.
0x08	Loop current fixed	le courant de ligne ou tension de signalisation (pour un appareil utilisant la tension de sortie) est maintenu à une valeur fixe et ne réagit pas aux variations de processus.
0x04	Loop current saturated	le courant de ligne ou tension de signalisation (pour un appareil utilisant la tension de sortie) a atteint sa limite de point d'extrémité supérieure (ou inférieure) et ne peut plus augmenter (ou diminuer).
0x02	Non-primary variable out of limits	une variable d'appareil non mappée avec la variable primaire a dépassé ses limites de fonctionnement..
0x01	Primary variable out of limits	la variable primaire a dépassé sa limite de fonctionnement.

5.1.7 Etat étendu

Ce champ indique l'état supplémentaire du processus d'application. Les valeurs possibles de ce champ sont indiquées dans le Tableau 4. Son data type est Bit Field sur un octet. Ce champ est utilisé dans le cadre des données d'utilisateur FAL ou de l'en-tête de certaines unités APDU.

Tableau 4 – Valeurs de l'état étendu

Code	Valeur	Description
0x01	Maintenance requise	L'appareil nécessite une maintenance, bien qu'il ne soit pas défaillant
0x02	Alerte de variable d'appareil	Une ou plusieurs 'Variables d'appareil' sont à l'état Alarm (alarme) ou Warning (Avertissement).
0x04	Panne d'alimentation critique	L'alimentation disponible devient extrêmement faible et critique; utilisé par les appareils fonctionnant par alimentation stockée.

NOTE L'état étendu est également appelé État d'appareil étendu, car il indique l'état fonctionnel du courant de l'appareil de terrain dans son ensemble et n'est pas associé à la réalisation d'une commande.

5.1.8 Formats des structures d'unité APDU

La couche Application émettrice prépare une unité APDU pour la transférer à la couche Application réceptrice. Elle utilise pour cela les paramètres issus des primitives de service. Il existe plusieurs formats d'unité APDU:

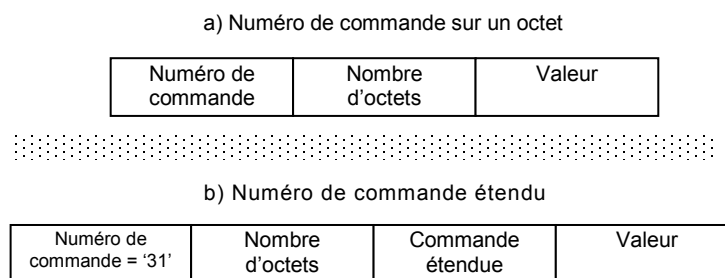
- unité APDU de demande de l'appareil maître à l'appareil esclave utilisant le numéro de commande sur un octet,
- unité APDU de demande de l'appareil maître à l'appareil esclave utilisant le numéro de commande étendu,
- réponse normale de l'appareil esclave à l'appareil maître utilisant le numéro de commande sur un octet,
- réponse normale de l'appareil esclave à l'appareil maître utilisant le numéro de commande étendu,
- réponse d'erreur de commande de l'appareil esclave à l'appareil maître utilisant le numéro de commande sur un octet, et
- réponse d'erreur de commande de l'appareil esclave à l'appareil maître utilisant le numéro de commande étendu.

Le format commun et les règles de codage de toutes les structures d'unités APDU sont spécifiés en 5.2. Les formats des structures d'unité APDU spécifique à un service sont spécifiés en 5.3. Le champ de numéro de commande est inclus dans la demande et dans la réponse. Par conséquent, le demandeur n'a pas besoin de stocker l'unité APDU de demande pour l'associer à l'unité APDU de réponse.

5.2 Structure d'unité APDU commune

5.2.1 Unité APDU de demande

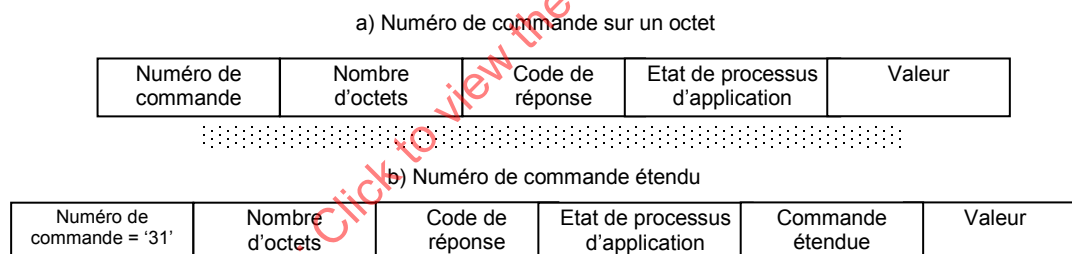
L'unité APDU de demande doit avoir une structure telle que représentée à la Figure 1. La Figure 1 (a) illustre l'unité APDU pour les valeurs de numéro de commande sur un octet de long et sa valeur n'est pas égale à trente et un (31). La Figure 1 (b) illustre l'unité APDU pour les valeurs de commande sur deux octets de long, également appelée numéro de commande étendu.

**Figure 1 – Unité APDU de demande**

Pour un service de lecture, le champ de valeur doit être Null ou correspondre au sous-index de variable. Pour un service d'écriture, le champ de valeur doit correspondre au sous-index de variable, suivi si nécessaire par la valeur devant être écrite dans les objets esclave identifiés par la Commande. L'ALE l'assemble à partir des paramètres d'une primitive de service. Sa structure dépend de l'unité APDU spécifique à un service.

5.2.2 Unité APDU de réponse normale

La Figure 2 représente la structure des champs de couche Application utilisés pour une réponse de l'esclave au maître en l'absence d'erreurs lors de l'exécution de la commande demandée. La Figure 2 (a) illustre l'unité APDU pour les valeurs de numéro de commande sur un octet de long et sa valeur n'est pas égale à trente et un (31). La Figure 2 (b) illustre l'unité APDU pour les valeurs de commande sur deux octets de long, également appelée numéro de commande étendu.

**Figure 2 – Unité APDU de réponse normale**

L'esclave envoie cette unité APDU en réponse à une demande du maître. La valeur du champ de commande de cette unité APDU est identique à celle du champ de commande et du champ de commande étendu (s'il existe) de l'unité APDU de demande correspondante. La valeur minimale du Champ de nombre d'octets est deux '2', car les champs de code de réponse et d'état de processus d'application sont toujours inclus.

Si la réponse correspond à une demande de lecture, le champ de valeur est utilisé pour définir la valeur des objets qui sont lus sur l'appareil esclave. Si la réponse correspond à une demande d'écriture, ce champ est utilisé pour définir la valeur des objets qui ont été écrits dans l'appareil esclave. Dans la réponse à la demande Write (écriture), il s'agit de la même valeur que celle reçue dans la demande, à moins que cette dernière ne soit pas admise et que le serveur en ait écrit une autre. Dans ce cas, il s'agit de la valeur écrite dans l'appareil esclave.

5.2.3 Unité APDU de réponse d'erreur de commande

La Figure 3 représente la structure des champs de couche Application utilisés pour une réponse de l'esclave au maître, en cas d'erreur lors de l'exécution de la commande. La Figure 3 (a) illustre l'unité APDU pour les valeurs de numéro de commande sur un octet de

long et sa valeur n'est pas égale à trente et un (31). La Figure 3 (b) illustre l'unité APDU pour les valeurs de commande sur deux octets de long, également appelée numéro de commande étendu. Le champ de valeur n'existe pas. La valeur du champ de nombre d'octets doit être '2' pour les valeurs de numéro de commande sur un octet de long et doit être '4' pour le numéro de commande étendu.

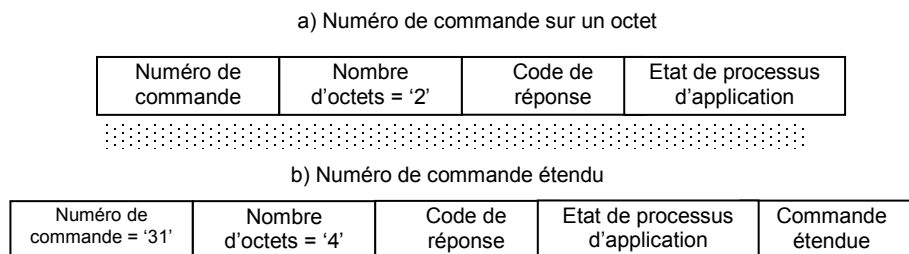


Figure 3 – Réponse d'erreur de commande de l'esclave au maître

L'appareil répondeur doit envoyer cette unité APDU en réponse à une demande du maître. La valeur du champ de commande de cette unité APDU doit être identique à celle du champ de numéro de commande de l'unité APDU de demande correspondante. L'état de processus d'application doit être l'état réel.

5.2.4 Unité APDU « Information report »

Un appareil esclave peut éditer des données sans nécessiter une demande du maître à chaque publication. A cet effet, il doit utiliser le service Information report en utilisant l'unité APDU de réponse spécifiée en 5.2.2 ou l'unité APDU de commande Aggregation spécifiée en 5.2.5. La publication est définie pour utiliser l'un des numéros de commande (voir CEI 62591:2010, 8.2.3.32).

5.2.5 Unité APDU de commande Aggregated

La présente norme permet d'acheminer plusieurs commandes et données associées dans une seule unité APDU de demande et de réponse. La commande Aggregation se révèle tout particulièrement utile pour la lecture des configurations d'appareil.

Le format d'unité APDU de demande doit être tel que présenté à la Figure 1 (a) avec le numéro de commande égal à '78'. Le nombre d'octets doit correspondre à la longueur du champ de valeur tel que présenté à la Figure 4 (a). Le nombre de demandes est égal au nombre de demandes individuelles dans le champ de valeur. Le format de demande individuelle est présenté à la Figure 4 (b). Le champ de numéro de commande dans chaque commande Request (demande) doit être sur deux octets de long; le nombre d'octets doit correspondre à la longueur du champ de valeur de la demande individuelle.

Le format d'unité APDU de réponse doit être tel que présenté à la Figure 2 (a) avec le numéro de commande égal à '78'. Le nombre d'octets doit correspondre à la longueur du champ de valeur tel que présenté à la Figure 4 (c). Le nombre de réponses est égal au nombre de réponses individuelles dans le champ de valeur. Le format de réponse individuelle est présenté à la Figure 4 (d). Le champ de numéro de commande dans chaque commande Response (réponse) doit être sur deux octets de long; le nombre d'octets doit correspondre à la longueur du code de réponse individuelle et des champs de valeur combinés.

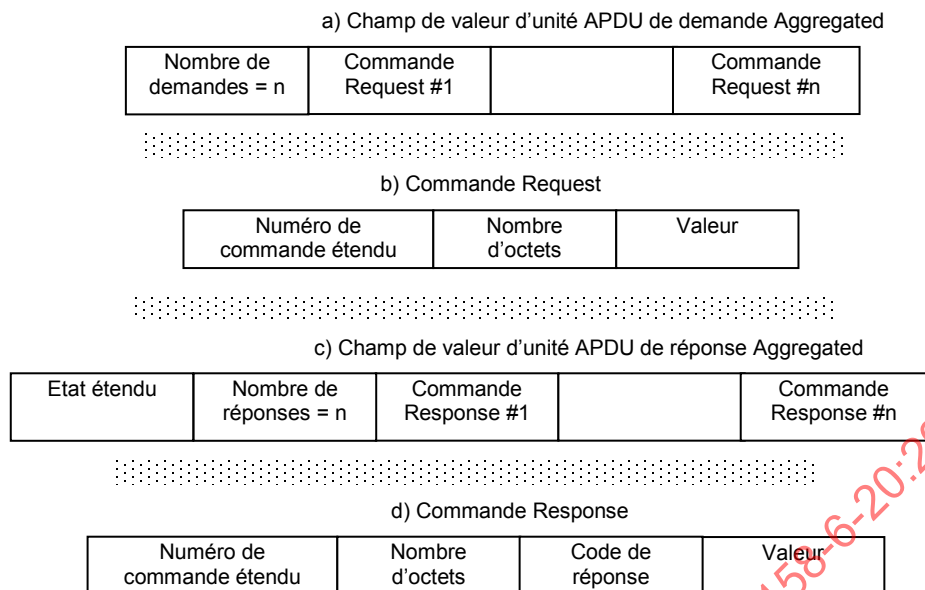


Figure 4 – Unité APDU de commande Aggregated

Si l'appareil a été configuré pour éditer plusieurs messages (voir 8.2.3.32 de la CEI 62591:2010), il peut les combiner dans un message en utilisant la commande Aggregation, à l'exception du fait que "Publish data Message 0" ne doit jamais être agrégé.

Dans toute la mesure du possible, l'appareil de terrain doit agréger plusieurs "Publish data Message" en attente dans une transaction en utilisant la demande Information report. Si la transaction ne peut contenir tous les messages "Publish data Message", ces derniers peuvent être envoyés séparément.

5.3 Structures d'unité APDU spécifique à un service d'application d'appareil

5.3.1 Identify FAL PDU

5.3.1.1 Primitive de demande

Le champ de commande peut prendre la valeur 0, 11 ou 21. Le format des unités APDU est décrit dans le Tableau 5.

Tableau 5 – Unité APDU de demande Identify

Commande	Nombre d'octets	Données
0	0	Null
11	6	Tag (Packed ASCII)
21	32	Long Tag (Latin-1)

Si la valeur du champ de commande est 0, le champ de valeur est null. Si la valeur du champ de commande est 11, le champ de valeur contient la valeur de six octets du paramètre Tag. Si la valeur du champ de commande est 21, le champ de valeur contient la valeur de 32 octets du paramètre Long Tag.

5.3.1.2 Primitive de réponse

Le champ de valeur de la réponse Identify est décrit dans le Tableau 6. Les valeurs du code de réponse sont indiquées dans le Tableau 7.

Tableau 6 – Champ de valeur de la réponse Identify

Position relative en octets	Data type	Nom du paramètre	Description
0	Unsigned8	None	toujours égal à 254
1	Enumeration	Expanded Device Type	énumération sur 2 octets qui identifie de manière unique le type d'appareil
3	Unsigned8	Preamble Count	indique le nombre minimum de préambules que le maître doit envoyer avec le message de demande à l'appareil répondeur
4	Unsigned8	Command Rev	indique le niveau de révision majeure du protocole pris en charge par l'appareil.
5	Unsigned8	Device Rev	décrit le niveau de révision de l'appareil. La valeur de ce paramètre doit être définie par le fabricant.
6	Unsigned8	Soft Rev	indique le niveau de révision du microprogramme de l'appareil. Le fabricant doit augmenter la valeur de ce paramètre à chaque nouvelle version du microprogramme de l'appareil.
7	Bit Field	Hard Rev Phy Type	a) les cinq bits de poids fort de cet octet représentent le niveau de révision du matériel de l'appareil. Le fabricant doit augmenter la valeur de cet attribut à chaque modification majeure du matériel de l'appareil. Il n'est pas nécessaire d'assurer le suivi des modifications apportées à chaque composant matériel. b) les trois bits de poids faible de cet octet représentent le type de transmission de signaux PHL utilisé par l'appareil.
8	Bit Field	Device Flag	indique d'autres informations concernant l'appareil (multi-capteur, contrôle de mémoire non volatile, pont de protocole, etc.).
9	Unsigned24	Device ID	indique le numéro de série de l'appareil. Le fabricant doit attribuer une valeur unique à tous les appareils possédant les mêmes valeurs pour les paramètres Manufacturer ID et Expanded Device Type.
12	Unsigned8	Preamble Count	indique le nombre minimum de préambules que l'esclave doit envoyer avec le message de réponse au maître.
13	Unsigned8	Variable Count	spécifie le nombre maximum d'objets (variables) accessibles depuis l'appareil. La valeur de ce paramètre indique le dernier code de variable qu'une application client peut s'attendre à trouver dans l'appareil.
14	Unsigned16	Configuration Change Count	assure le suivi du nombre de modifications apportées à la configuration de l'appareil. L'appareil doit augmenter la valeur de ce paramètre chaque fois qu'il reçoit une demande de modification de la configuration au moyen des services de couche Application ou qu'un utilisateur de l'appareil en modifie la configuration au moyen d'un outil local tel que l'interface d'opérateur local.
16	Bit Field	Device ExtStatus	indique l'état de fonctionnement étendu de l'appareil.
17	Enumeration	Manufacturer ID	énumération sur 2 octets désignant le fabricant qui a produit l'appareil. Cette valeur doit être unique pour tous les appareils de Type 20 qui se trouvent sur le même réseau et à portée d'autres réseaux.
19	Enumeration	Distributor code	énumération sur 2 octets qui désigne le fabricant de marque privée qui a distribué l'appareil. Cette valeur doit être unique pour tous les appareils de Type 20 qui se trouvent sur le même réseau et à portée d'autres réseaux.
21	Enumeration	Device profile	spécifie la classe à laquelle appartient l'appareil comme indiqué dans la CEI 62591:2010, Tableau F.32.

Tableau 7 – Codes de réponse spécifiques à la commande Identify

Valeur	Classe	Description
0	Success	Absence d'erreurs

5.3.1.3 Procédure au niveau de l'appareil répondeur

Si la demande contient une valeur Tag ou Long Tag, l'appareil doit la comparer à la valeur Tag ou Long Tag stockée dans l'appareil. Si l'appareil doit répondre uniquement si la valeur figurant dans la demande correspond à la valeur stockée dans l'appareil.

5.3.2 Read primary variable FAL PDU

5.3.2.1 Primitive de demande

La valeur du champ de commande est un (1); la valeur du champ de nombre d'octets est zéro (0) et le champ de données est null.

5.3.2.2 Primitive de réponse

Le champ de valeur de la réponse est décrit dans le Tableau 8. Les valeurs du code de réponse sont indiquées dans le Tableau 9.

Tableau 8 – Champ de valeur de la réponse Read primary variable

Position relative en octets	Data type	Nom du paramètre	Description
0	Engineering unit	Primary Variable Unit	Ce paramètre représente l'unité technique de la variable primaire.
1 – 4	Float32	Primary Variable	Ce paramètre indique la valeur actuelle de la variable primaire.

Tableau 9 – Codes de réponse spécifiques à la commande Read primary variable

Valeur	Classe	Description
0	Success	Absence d'erreurs
6	Error	Erreur de commande spécifique à l'appareil
8	Warning	Échec de mise à jour
16	Error	Accès restreint

5.3.3 Read loop current and percent of range FAL PDU

5.3.3.1 Primitive de demande

La valeur du champ de commande est deux (2); la valeur du champ de nombre d'octets est zéro (0) et le champ de données est null.

5.3.3.2 Primitive de réponse

Le champ de valeur de la réponse est décrit dans le Tableau 10. Les valeurs du code de réponse sont indiquées dans le Tableau 11.

Tableau 10 – Champ de valeur de Read loop current and percent of range

Position relative en octets	Data type	Nom du paramètre	Description
0 – 3	Float32	Loop Current	Ce paramètre indique la valeur actuelle de la sortie de l'appareil – courant de ligne en milliampères ou tension de sortie en volts.
4 – 7	Float32	Primary Variable	Ce paramètre indique la valeur actuelle de la variable primaire exprimée en pourcentage de la gamme. Cette valeur suit l'entrée primaire de l'appareil. Elle peut se trouver à l'extérieur de la gamme, comprise entre 0 % et 100 %.

**Tableau 11 – Codes de réponse spécifiques à la commande
Read loop current and percent of range**

Valeur	Classe	Description
0	Success	Absence d'erreurs
6	Error	Erreur de commande spécifique à l'appareil
8	Warning	Échec de mise à jour
16	Error	Accès restreint

5.3.4 Read dynamic variables and loop current FAL PDU

5.3.4.1 Primitive de demande

La valeur du champ de commande est trois (3); la valeur du champ de nombre d'octets est zéro (0) et le champ de données est null.

5.3.4.2 Primitive de réponse

Le champ de valeur de la réponse est décrit dans le Tableau 12. Les valeurs du code de réponse sont indiquées dans le Tableau 13. Si un appareil ne prend pas en charge les quatre variables, la réponse contient uniquement le nombre de variables pris en charge par l'appareil. Dans ce cas, les champs restants ne figurent pas dans la réponse.

Tableau 12 – Champ de valeur de Read dynamic variables and loop current

Position relative en octets	Data type	Nom du paramètre	Description
0 – 3	Float32	Loop Current	Ce paramètre indique la valeur actuelle de la sortie de l'appareil – courant de ligne en milliampères ou tension de sortie en volts.
4	Engineering unit	Primary Variable Unit	Ce paramètre représente l'unité technique de la variable primaire.
5 – 8	Float32	Primary Variable	Ce paramètre indique la valeur actuelle de la variable primaire.
9	Engineering unit	Secondary Variable Unit	Ce paramètre représente l'unité technique de la variable secondaire.
10 – 13	Float32	Secondary Variable	Ce paramètre indique la valeur actuelle de la variable secondaire.
14	Engineering unit	Tertiary Variable Unit	Ce paramètre représente l'unité technique de la variable tertiaire.
15 – 18	Float32	Tertiary Variable	Ce paramètre indique la valeur actuelle de la variable tertiaire.
19	Engineering unit	Quaternary Variable Unit	Ce paramètre représente l'unité technique de la variable quaternaire.
20 – 23	Float32	Quaternary Variable	Ce paramètre indique la valeur actuelle de la variable quaternaire.

**Tableau 13 – Codes de réponse spécifiques à la commande
Read dynamic variables and loop current**

Valeur	Classe	Description
0	Success	Absence d'erreurs
6	Error	Erreur de commande spécifique à l'appareil
8	Warning	Échec de mise à jour
16	Error	Accès restreint

5.3.5 Write loop configuration FAL PDU

5.3.5.1 Primitive de demande

La valeur du champ de commande est six (6); la valeur du champ de nombre d'octets est deux (2) et le champ de données est tel qu'indiqué dans le Tableau 14.

Tableau 14 – Champ de valeur de Write loop configuration

Position relative en octets	Data type	Nom du paramètre	Description
0	Unsigned8	Polling Address	Ce paramètre représente l'adresse de liaison de données de l'appareil.
1	Enumeration	Loop Current Mode	Ce paramètre code le mode de courant de ligne comme indiqué dans le Tableau C.6.

5.3.5.2 Primitive de réponse

Le champ de valeur de la réponse est décrit dans le Tableau 14. Les valeurs du code de réponse sont indiquées dans le Tableau 15.

Tableau 15 – Codes de réponse spécifiques à la commande Write loop configuration

Valeur	Classe	Description
0	Success	Absence d'erreurs
2	Error	Adresse d'interrogation non valide
5	Error	Nombre d'octets de données reçus trop faible
6	Error	Erreur de commande spécifique à l'appareil
7	Error	Protégé en écriture
12	Error	Choix de mode non valide
16	Error	Accès restreint
32	Error	Occupé

5.3.5.3 Procédure au niveau de l'appareil répondeur

Un appareil doit désactiver la transmission de signaux de courant de ligne lorsque le maître le demande. Lorsque la transmission de signaux de courant est désactivée, le courant de ligne doit être défini sur la valeur minimum nécessaire au fonctionnement de l'appareil de terrain. Le bit d'état 4 du processus d'application (voir Tableau 3, Loop Current Fixed) doit être défini sur '1'.

5.3.6 Read loop configuration FAL PDU

5.3.6.1 Primitive de demande

La valeur du champ de commande est sept (7); la valeur du champ de nombre d'octets est zéro (0) et le champ de données est null.

5.3.6.2 Primitive de réponse

Le champ de valeur de la réponse est décrit dans le Tableau 16. Les valeurs du code de réponse sont indiquées dans le Tableau 17.

Tableau 16 – Champ de valeur de Read loop configuration

Position relative en octets	Data type	Nom du paramètre	Description
0	Unsigned8	Polling Address	Ce paramètre représente l'adresse de liaison de données de l'appareil.
1	Enumeration	Loop Current Mode	Ce paramètre code le mode de courant de ligne comme indiqué dans le Tableau C.6.

Tableau 17 – Codes de réponse spécifiques à la commande Read loop configuration

Valeur	Classe	Description
0	Success	Absence d'erreurs
16	Error	Accès restreint

5.3.7 Read dynamic variable families classifications FAL PDU

5.3.7.1 Primitive de demande

La valeur du champ de commande est huit (8); la valeur du champ de nombre d'octets est zéro (0) et le champ de données est null.

5.3.7.2 Primitive de réponse

Le champ de valeur de la réponse est décrit dans le Tableau 18. Les valeurs du code de réponse sont indiquées dans le Tableau 19.

Tableau 18 – Champ de valeur de Read dynamic variable families classifications

Position relative en octets	Data type	Nom du paramètre	Description
0	Enumeration	Primary Variable Classification	Ce paramètre indique la fonction remplie par la variable de l'appareil selon les spécifications du Tableau F.5 de la CEI 62591:2010.
1	Enumeration	Secondary Variable Classification	Ce paramètre indique la fonction remplie par la variable de l'appareil selon les spécifications du Tableau F.5 de la CEI 62591:2010.
2	Enumeration	Tertiary Variable Classification	Ce paramètre indique la fonction remplie par la variable de l'appareil selon les spécifications du Tableau F.5 de la CEI 62591:2010.
3	Enumeration	Quaternary Variable Classification	Ce paramètre indique la fonction remplie par la variable de l'appareil selon les spécifications du Tableau F.5 de la CEI 62591:2010.

Tableau 19 – Codes de réponse spécifiques à la commande Read dynamic variable families classifications

Valeur	Classe	Description
0	Success	Absence d'erreurs
16	Error	Accès restreint

5.3.8 Read device variables with status FAL PDU

5.3.8.1 Primitive de demande

La valeur du champ de commande est neuf (9); la valeur du champ de nombre d'octets est comprise entre un et huit (1 – 8) et est égale au nombre de variables demandées. Le champ de valeur de la demande est décrit dans le Tableau 20.

Tableau 20 – Champ de valeur de la demande Read device variables with status

Position relative en octets	Data type	Nom du paramètre	Description
0	Unsigned8	Slot 0	Ce paramètre indique le code attribué à une variable d'appareil
1	Unsigned8	Slot 1	Ce paramètre indique le code attribué à une variable d'appareil
2	Unsigned8	Slot 2	Ce paramètre indique le code attribué à une variable d'appareil
3	Unsigned8	Slot 3	Ce paramètre indique le code attribué à une variable d'appareil
4	Unsigned8	Slot 4	Ce paramètre indique le code attribué à une variable d'appareil
5	Unsigned8	Slot 5	Ce paramètre indique le code attribué à une variable d'appareil
6	Unsigned8	Slot 6	Ce paramètre indique le code attribué à une variable d'appareil
7	Unsigned8	Slot 7	Ce paramètre indique le code attribué à une variable d'appareil

5.3.8.2 Primitive de réponse

Le champ de valeur de la réponse est décrit dans le Tableau 21. Les valeurs du code de réponse sont indiquées dans le Tableau 22.

Tableau 21 – Champ de valeur de Read device variables with status

Position relative en octets	Data type	Nom du paramètre	Description
0	Bit Field	Device Extd Status	Ce paramètre indique l'état de fonctionnement étendu de l'appareil.
1	Unsigned8	Variable Code	Ce paramètre indique le code reçu en tant que demande dans le champ de valeur associé au connecteur 0.
2	Enumeration	Variable classification	Ce paramètre indique la fonction remplie par la variable de l'appareil associée au connecteur 0 selon les spécifications du Tableau F.5 de la CEI 62591:2010.
3	Engineering unit	Variable Unit	Ce paramètre représente l'unité technique de la variable associée au connecteur 0.
4	Float32	Variable Value	Ce paramètre indique la valeur actuelle de la variable associée au connecteur 0.
8	Bit Field	Variable Status	Ce paramètre représente l'état de la variable associée au connecteur 0, comme spécifié à l'Article B.6.
9	Unsigned8	Variable Code	Ce paramètre indique le code reçu en tant que demande dans le champ de valeur associé au connecteur 1.
10	Enumeration	Variable classification	Ce paramètre indique la fonction remplie par la variable de l'appareil associée au connecteur 1 selon les spécifications du Tableau F.5 de la CEI 62591:2010.
11	Engineering unit	Variable Unit	Ce paramètre représente l'unité technique de la variable associée au connecteur 1.
12	Float32	Variable Value	Ce paramètre indique la valeur actuelle de la variable associée au connecteur 1.
16	Bit Field	Variable Status	Ce paramètre représente l'état de la variable associée au connecteur 1, comme spécifié à l'Article B.6.

Position relative en octets	Data type	Nom du paramètre	Description
17	Unsigned8	Variable Code	Ce paramètre indique le code reçu en tant que demande dans le champ de valeur associé au connecteur 2.
18	Enumeration	Variable classification	Ce paramètre indique la fonction remplie par la variable de l'appareil associée au connecteur 2 selon les spécifications du Tableau F.5 de la CEI 62591:2010.
19	Engineering unit	Variable Unit	Ce paramètre représente l'unité technique de la variable associée au connecteur 2.
20	Float32	Variable Value	Ce paramètre indique la valeur actuelle de la variable associée au connecteur 2.
24	Bit Field	Variable Status	Ce paramètre représente l'état de la variable associée au connecteur 2, comme spécifié à l'Article B.6.
25	Unsigned8	Variable Code	Ce paramètre indique le code reçu en tant que demande dans le champ de valeur associé au connecteur 3.
26	Enumeration	Variable classification	Ce paramètre indique la fonction remplie par la variable de l'appareil associée au connecteur 3 selon les spécifications du Tableau F.5 de la CEI 62591:2010.
27	Engineering unit	Variable Unit	Ce paramètre représente l'unité technique de la variable associée au connecteur 3.
28	Float32	Variable Value	Ce paramètre indique la valeur actuelle de la variable associée au connecteur 3.
32	Bit Field	Variable Status	Ce paramètre représente l'état de la variable associée au connecteur 3, comme spécifié à l'Article B.6.
33	Unsigned8	Variable Code	Ce paramètre indique le code reçu en tant que demande dans le champ de valeur associé au connecteur 4.
34	Enumeration	Variable classification	Ce paramètre indique la fonction remplie par la variable de l'appareil associée au connecteur 4 selon les spécifications du Tableau F.5 de la CEI 62591:2010.
35	Engineering unit	Variable Unit	Ce paramètre représente l'unité technique de la variable associée au connecteur 4.
36	Float32	Variable Value	Ce paramètre indique la valeur actuelle de la variable associée au connecteur 4.
40	Bit Field	Variable Status	Ce paramètre représente l'état de la variable associée au connecteur 4, comme spécifié à l'Article B.6.
41	Unsigned8	Variable Code	Ce paramètre indique le code reçu en tant que demande dans le champ de valeur associé au connecteur 5.
42	Enumeration	Variable classification	Ce paramètre indique la fonction remplie par la variable de l'appareil associée au connecteur 5 selon les spécifications du Tableau F.5 de la CEI 62591:2010.
43	Engineering unit	Variable Unit	Ce paramètre représente l'unité technique de la variable associée au connecteur 5.
44	Float32	Variable Value	Ce paramètre indique la valeur actuelle de la variable associée au connecteur 5.
48	Bit Field	Variable Status	Ce paramètre représente l'état de la variable associée au connecteur 5, comme spécifié à l'Article B.6.
49	Unsigned8	Variable Code	Ce paramètre indique le code reçu en tant que demande dans le champ de valeur associé au connecteur 6.
50	Enumeration	Variable classification	Ce paramètre indique la fonction remplie par la variable de l'appareil associée au connecteur 6 selon les spécifications du Tableau F.5 de la CEI 62591:2010.
51	Engineering unit	Variable Unit	Ce paramètre représente l'unité technique de la variable associée au connecteur 6.
52	Float32	Variable Value	Ce paramètre indique la valeur actuelle de la variable associée au connecteur 6.

Position relative en octets	Data type	Nom du paramètre	Description
56	Bit Field	Variable Status	Ce paramètre représente l'état de la variable associée au connecteur 6, comme spécifié à l'Article B.6.
57	Unsigned8	Variable Code	Ce paramètre indique le code reçu en tant que demande dans le champ de valeur associé au connecteur 7.
58	Enumeration	Variable classification	Ce paramètre indique la fonction remplie par la variable de l'appareil associée au connecteur 7 selon les spécifications du Tableau F.5 de la CEI 62591:2010.
59	Engineering unit	Variable Unit	Ce paramètre représente l'unité technique de la variable associée au connecteur 7.
60	Float32	Variable Value	Ce paramètre indique la valeur actuelle de la variable associée au connecteur 7.
64	Bit Field	Variable Status	Ce paramètre représente l'état de la variable associée au connecteur 7, comme spécifié à l'Article B.6.
65	Time	Time stamp	Horodatage des données du connecteur 0

**Tableau 22 – Codes de réponse spécifiques à la commande
Read device variables with status**

Valeur	Classe	Description
0	Success	Absence d'erreurs
2	Error	Choix non valide
5	Error	Nombre d'octets de données reçus trop faible
6	Error	Erreur de commande spécifique à l'appareil
8	Warning	Échec de mise à jour
14	Warning	Variables dynamiques renvoyées pour les variables d'appareil
16	Error	Accès restreint
30	Warning	Réponse à la commande tronquée

5.3.8.3 Procédure au niveau de l'appareil répondeur

Si la demande ne contient aucun octet de données (Champ de valeur), l'appareil doit renvoyer le code d'erreur 5; dans le cas contraire, le code de réponse doit être différent de 5.

L'appareil doit renvoyer un nombre de valeurs exactement égal au nombre de variables figurant dans la demande.

Si une variable demandée n'est pas prise en charge par l'appareil, les champs de l'unité APDU de réponse sont définis comme suit:

le champ Variable Classification doit être défini sur '0';

le champ Variable Unit doit être défini sur '250';

le champ Variable Value doit être défini sur une valeur en virgule flottante de 'Not-a-number' (0x7F, 0xA0, 0x00, 0x00); et

Champ Variable Status doit être défini sur 0x30.

Si le paramètre Variable classification n'est pas pris en charge pour une variable demandée, le champ Variable Classification de l'unité APDU de réponse correspondant à cette variable doit être défini sur '0' (pas encore classée) et le code de réponse doit être défini sur '14'.

L'appareil doit prendre en charge au moins quatre connecteurs.

5.3.9 Read message FAL PDU

5.3.9.1 Primitive de demande

La valeur du champ de commande est 12 (12); la valeur du champ de nombre d'octets est zéro (0) et le champ de données est null.

5.3.9.2 Primitive de réponse

Le champ de valeur de la réponse est décrit dans le Tableau 23. Les valeurs du code de réponse sont indiquées dans le Tableau 24.

Tableau 23 – Champ de valeur de la réponse Read message

Position relative en octets	Data type	Nom du paramètre	Description
0 – 23	Packed ASCII	Message	Indique la valeur du paramètre Message stocké dans l'appareil.

Tableau 24 – Codes de réponse spécifiques à la commande Read message

Valeur	Classe	Description
0	Success	Absence d'erreurs
16	Error	Accès restreint
32	Error	Occupé

5.3.10 Read tag, descriptor and date FAL PDU

5.3.10.1 Primitive de demande

La valeur du champ de commande est treize (13); la valeur du champ de nombre d'octets est zéro (0) et le champ de données est null.

5.3.10.2 Primitive de réponse

Le champ de valeur de la réponse est décrit dans le Tableau 25. Les valeurs du code de réponse sont indiquées dans le Tableau 26.

Tableau 25 – Champ de valeur de la réponse Read tag, descriptor, date

Position relative en octets	Data type	Nom du paramètre	Description
0 – 5	Packed ASCII	Tag	Indique la valeur du paramètre Tag stocké dans l'appareil.
6 – 17	Packed ASCII	Descriptor	Indique la valeur du paramètre Descriptor stocké dans l'appareil.
18 – 20	Date	Date	Indique la valeur du paramètre Date stocké dans l'appareil.

Tableau 26 – Codes de réponse spécifiques à la commande Read tag, descriptor, date

Valeur	Classe	Description
0	Success	Absence d'erreurs
16	Error	Accès restreint
32	Error	Occupé

5.3.11 Read primary variable transducer information FAL PDU**5.3.11.1 Primitive de demande**

La valeur du champ de commande est quatorze (14); la valeur du champ de nombre d'octets est zéro (0) et le champ de données est null.

5.3.11.2 Primitive de réponse

Le champ de valeur de la réponse est décrit dans le Tableau 27. Les valeurs du code de réponse sont indiquées dans le Tableau 28.

Tableau 27 – Champ de valeur de la réponse Read primary variable transducer information

Position relative en octets	Data type	Nom du paramètre	Description
0 – 2	Unsigned24	Transducer Serial Number	Indique la valeur du paramètre Transducer Serial Number.
3	Engineering unit	Transducer Unit	Indique le code associé aux paramètres Transducer Limits et Minimum-Span Units
4 – 7	Float32	Upper Transducer Limit	Indique la valeur de la variable Upper Transducer Limit.
8 – 11	Float32	Lower Transducer Limit	Indique la valeur de la variable Lower Transducer Limit.
12 – 15	Float32	Minimum Span	Indique la valeur de la variable Minimum Span.

Tableau 28 – Codes de réponse spécifiques à la commande Read primary variable transducer information

Valeur	Classe	Description
0	Success	Absence d'erreurs
16	Error	Accès restreint
32	Error	Occupé

5.3.11.3 Procédure au niveau de l'appareil répondeur

Si le paramètre Transducer Serial Number n'est pas applicable à l'appareil, ce dernier doit renvoyer les valeurs suivantes:

- Transducer Serial Number égale à '0';
- Transducer Unit égale à '250';
- toutes les autres valeurs égales à une valeur en virgule flottante de 'Not-a-number' (0x7F, 0xA0, 0x00, 0x00).

5.3.12 Read device information FAL PDU

5.3.12.1 PrIMITIVE DE DEMANDE

La valeur du champ de commande est quinze (15); la valeur du champ de nombre d'octets est zéro (0) et le champ de données est null.

5.3.12.2 PrIMITIVE DE RÉPONSE

Le champ de valeur de la réponse est décrit dans le Tableau 29. Les valeurs du code de réponse sont indiquées dans le Tableau 30.

Tableau 29 – Champ de valeur de la réponse Read device information

Position relative en octets	Data type	Nom du paramètre	Description
0	Enumeration	PV Alarm Selection	Indique la valeur du paramètre PV Alarm Selection comme indiqué dans le Tableau C.2. Cette valeur décrit l'action entreprise par l'appareil en cas d'erreur. Pour les émetteurs, le code indique l'action entreprise par le courant de ligne. Pour les actionneurs, il indique l'action entreprise par le positionneur.
1	Enumeration	PV Transfer Function	Indique la valeur du paramètre PV Transfer Function comme indiqué dans le Tableau C.1. Si l'appareil ne prend pas en charge les fonctions de transfert, cette valeur est égale à '0'.
2	Engineering unit	PV Range Unit	Indique la valeur des limites supérieure et inférieure du paramètre PV Range Unit.
3 – 6	Float32	PV Upper Range	Indique la valeur du paramètre PV Upper Range.
7 – 10	Float32	PV Lower Range	Indique la valeur du paramètre PV Lower Range.
11 – 14	Float32	PV Damping	Indique la valeur du paramètre PV Damping en secondes.
15	Enumeration	Write Protect	Indique la valeur du paramètre Write Protect comme indiqué dans le Tableau C.3. Si l'appareil ne fournit pas de fonction de protection en écriture, cette valeur est égale à '251'.
16	Enumeration	Reserved	Ce champ est réservé pour utilisation future. Sa valeur doit être égale à '250'.
17	Bit Field	PV Analog Channel	Indique la valeur des indicateurs PV Analog Channel comme indiqué dans le Tableau C.8.

Tableau 30 – Codes de réponse spécifiques à la commande Read device information

Valeur	Classe	Description
0	Success	Absence d'erreurs
16	Error	Accès restreint
32	Error	Occupé

5.3.13 Read final assembly number FAL PDU

5.3.13.1 PrIMITIVE DE DEMANDE

La valeur du champ de commande est seize (16); la valeur du champ de nombre d'octets est zéro (0) et le champ de données est null.

5.3.13.2 PrIMITIVE DE RÉPONSE

Le champ de valeur de la réponse est décrit dans le Tableau 31. Les valeurs du code de réponse sont indiquées dans le Tableau 32.

Tableau 31 – Champ de valeur de la réponse Read final assembly number

Position relative en octets	Data type	Nom du paramètre	Description
0 – 2	Unsigned24	Final Assembly Number	Indique la valeur du paramètre Final Assembly Number

Tableau 32 – Codes de réponse spécifiques à la commande Read final assembly number

Valeur	Classe	Description
0	Success	Absence d'erreurs
16	Error	Accès restreint
32	Error	Occupé

5.3.14 Write message FAL PDU

5.3.14.1 Primitive de demande

La valeur du champ de commande est dix-sept (17); la valeur du champ de nombre d'octets est vingt-quatre (24) et le champ de données est tel qu'indiqué dans le Tableau 33.

Tableau 33 – Champ de valeur de Write message

Position relative en octets	Data type	Nom du paramètre	Description
0 – 23	Packed ASCII	Message	Chaîne utilisée par un maître à des fins de tenue des dossiers

5.3.14.2 Primitive de réponse

Le champ de valeur de la réponse est décrit dans le Tableau 33. Les valeurs du code de réponse sont indiquées dans le Tableau 34.

Tableau 34 – Codes de réponse spécifiques à la commande Write message

Valeur	Classe	Description
0	Success	Absence d'erreurs
2	Error	Adresse d'interrogation non valide
5	Error	Nombre d'octets de données reçus trop faible
6	Error	Erreur de commande spécifique à l'appareil
7	Error	Protégé en écriture
16	Error	Accès restreint
32	Error	Occupé

5.3.15 Write tag, descriptor and date FAL PDU

5.3.15.1 Primitive de demande

La valeur du champ de commande est dix-huit (18); la valeur du champ de nombre d'octets est vingt-et-un (21) et le champ de données est tel qu'indiqué dans le Tableau 35.

Tableau 35 – Champ de valeur de Write tag, descriptor, date

Position relative en octets	Data type	Nom du paramètre	Description
0 – 5	Packed ASCII	Tag	Indique la valeur du paramètre Tag à stocker dans l'appareil.
6 – 17	Packed ASCII	Descriptor	Indique la valeur du paramètre Descriptor à stocker dans l'appareil.
18 – 20	Date	Date	Indique la valeur du paramètre Date à stocker dans l'appareil.

5.3.15.2 Primitive de réponse

Le champ de valeur de la réponse est décrit dans le Tableau 35. Les valeurs du code de réponse sont indiquées dans le Tableau 36.

Tableau 36 – Codes de réponse spécifiques à la commande Write tag, descriptor, date

Valeur	Classe	Description
0	Success	Absence d'erreurs
5	Error	Nombre d'octets de données reçus trop faible
6	Error	Erreur de commande spécifique à l'appareil
7	Error	Protégé en écriture
9	Error	Code date non valide
16	Error	Accès restreint
32	Error	Occupé

5.3.16 Write final assembly number FAL PDU

5.3.16.1 Primitive de demande

La valeur du champ de commande est dix-neuf (19); la valeur du champ de nombre d'octets est trois (3) et le champ de données est tel qu'indiqué dans le Tableau 37.

Tableau 37 – Champ de valeur de Write final assembly number

Position relative en octets	Data type	Nom du paramètre	Description
0 – 2	Unsigned24	Final Assembly Number	Indique la valeur du paramètre Final Assembly Number. Cette valeur est normalement modifiée lors de la modernisation sur le terrain de composants électroniques ou autres de l'appareil.

5.3.16.2 Primitive de réponse

Le champ de valeur de la réponse est décrit dans le Tableau 37. Les valeurs du code de réponse sont indiquées dans le Tableau 38.

Tableau 38 – Codes de réponse spécifiques à la commande Write final assembly number

Valeur	Classe	Description
0	Success	Absence d'erreurs
5	Error	Nombre d'octets de données reçus trop faible
6	Error	Erreur de commande spécifique à l'appareil
7	Error	Protégé en écriture

Valeur	Classe	Description
16	Error	Accès restreint
32	Error	Occupé

5.3.17 Read long tag FAL PDU

5.3.17.1 Primitive de demande

La valeur du champ de commande est vingt (20); la valeur du champ de nombre d'octets est zéro (0) et le champ de données est null.

5.3.17.2 Primitive de réponse

Le champ de valeur de la réponse est décrit dans le Tableau 39. Les valeurs du code de réponse sont indiquées dans le Tableau 40.

Tableau 39 – Champ de valeur de la réponse Read long tag

Position relative en octets	Data type	Nom du paramètre	Description
0 – 31	Latin-1	Long Tag	Indique la valeur du paramètre Long Tag stocké dans l'appareil.

Tableau 40 – Codes de réponse spécifiques à la commande Read long tag

Valeur	Classe	Description
0	Success	Absence d'erreurs
16	Error	Accès restreint
32	Error	Occupé

5.3.18 Write long tag FAL PDU

5.3.18.1 Primitive de demande

La valeur du champ de commande est vingt-deux (22); la valeur du champ de nombre d'octets est trente-deux (32) et le champ de données est tel qu'indiqué dans le Tableau 41.

Tableau 41 – Champ de valeur de Write long tag

Position relative en octets	Data type	Nom du paramètre	Description
0 – 31	Latin-1	Long Tag	Indique la valeur du paramètre Long Tag à stocker dans l'appareil.

5.3.18.2 Primitive de réponse

Le champ de valeur de la réponse est décrit dans le Tableau 41. Les valeurs du code de réponse sont indiquées dans le Tableau 42.

Tableau 42 – Codes de réponse spécifiques à la commande Write long tag

Valeur	Classe	Description
0	Success	Absence d'erreurs
5	Error	Nombre d'octets de données reçus trop faible
6	Error	Erreur de commande spécifique à l'appareil

Valeur	Classe	Description
7	Error	Protégé en écriture
16	Error	Accès restreint
32	Error	Occupé
33	Error	DR initiée
34	Error	DR en fonctionnement
35	Error	DR arrêtée
36	Error	DR en conflit

5.3.19 Reset configuration changed flag FAL PDU

Cette Commande est spécifiée dans la CEI 62591:2010, 8.2.3.19.

5.3.20 Perform self test FAL PDU

Cette Commande est spécifiée dans la CEI 62591:2010, 8.2.3.20.

5.3.21 Perform device reset FAL PDU

Cette Commande est spécifiée dans la CEI 62591:2010, 8.2.3.21.

5.3.22 Read additional device status FAL PDU

Cette Commande est spécifiée dans la CEI 62591:2010, 8.2.3.22.

5.3.23 Reset more status available FAL PDU

Cette Commande est spécifiée dans la CEI 62591:2010, 8.2.3.23.

5.3.24 Read device variable information FAL PDU

Cette Commande est spécifiée dans la CEI 62591:2010, 8.2.3.24.

5.3.25 Write device variable FAL PDU

Cette Commande est spécifiée dans la CEI 62591:2010, 8.2.3.25.

5.3.26 Write primary variable range FAL PDU

5.3.26.1 Primitive de demande

La valeur du champ de commande est trente-cinq (35); la valeur du champ de nombre d'octets est neuf (9) et le champ de valeur est tel qu'indiqué dans le Tableau 43.

Tableau 43 – Champ de valeur de Write primary variable range

Position relative en octets	Data type	Nom du paramètre	Description
0	Engineering unit	Range unit	Unité de mesure de la valeur Range
1	Float32	Upper range value	Valeur Range (plage) à écrire
5	Float32	Lower range value	Valeur Range (plage) à écrire

5.3.26.2 Primitive de réponse

Le champ de valeur de la réponse est décrit dans le Tableau 43. Les valeurs du code de réponse sont indiquées dans le Tableau 44.

**Tableau 44 – Codes de réponse spécifiques à la commande
Write primary variable range**

Valeur	Classe	Description
0	Success	Absence d'erreurs
2	Error	Choix non valide
5	Error	Nombre d'octets de données reçus trop faible
6	Error	Erreur de commande spécifique à l'appareil
7	Error	Protégé en écriture
8	Warning	Défini sur la valeur possible la plus proche
9	Error	Valeur de Lower range (plage inférieure) trop élevée
10	Error	Valeur de plage inférieure trop faible
11	Error	Valeur de Upper range (plage supérieure) trop élevée
12	Error	Valeur de plage supérieure trop faible
13	Error	Valeurs de plage supérieure et inférieure hors limite
14	Warning	Variable Span (étendue) trop petite (la précision de l'appareil peut être compromise).
16	Error	Accès restreint
18	Error	Code Units non valide
29	Error	Variable Span (étendue) non valide
32	Error	Occupé

5.3.26.3 Procédure au niveau de l'appareil répondeur

La valeur de plage supérieure de la variable primaire peut être inférieure à sa valeur de plage inférieure, permettant à l'appareil de fonctionner par action inverse.

NOTE Le document spécifique à l'appareil indique si cette capacité n'a pas été mise en œuvre.

Les valeurs de Range (plage) doivent être renvoyées telles que reçues dans les mêmes unités. La valeur renvoyée dans la réponse doit être arrondie ou correspondre à la valeur tronquée réellement utilisée par l'appareil. Le code Range Unit de la variable primaire reçu avec cette commande ne doit pas affecter le Unit de la variable primaire de l'appareil. Pour un appareil à courant de sortie, les valeurs de Range permettent de convertir la valeur de la variable primaire en un pourcentage de transmission via le courant de ligne. Pour un appareil à courant d'entrée, les valeurs de Range values permettant de convertir le courant de ligne en un pourcentage à utiliser par l'actionneur (par exemple, à utiliser comme le point de consigne de l'actionneur).

5.3.27 Enter-exit fixed current mode FAL PDU

5.3.27.1 Primitive de demande

La valeur du champ de commande est quarante (40); la valeur du champ de nombre d'octets est quatre (4) et le champ de valeur est tel qu'indiqué dans le Tableau 45.

Tableau 45 – Champ de valeur de la demande Enter-exit fixed current mode

Position relative en octets	Data type	Nom du paramètre	Description
0	Float32	Fixed current level	Valeur de courant de ligne à définir

5.3.27.2 Primitive de réponse

Le champ de valeur de la réponse est décrit dans le Tableau 46. Les valeurs du code de réponse sont indiquées dans le Tableau 47.

Tableau 46 – Champ de valeur de la réponse Enter-exit fixed current mode

Position relative en octets	Data type	Nom du paramètre	Description
0	Float32	Actual current level	Valeur de courant réelle

Tableau 47 – Codes de réponse spécifiques à la commande Enter-exit fixed current mode

Valeur	Classe	Description
0	Success	Absence d'erreurs
3	Error	Niveau de courant fixe trop élevé
4	Error	Niveau de courant fixe trop faible
5	Error	Nombre d'octets de données reçus trop faible
6	Error	Erreur de commande spécifique à l'appareil
7	Error	Protégé en écriture
11	Error	Courant de ligne non actif (appareil en mode multipoint)
16	Error	Accès restreint
32	Error	Occupé

5.3.27.3 Procédure au niveau de l'appareil répondeur

Si la valeur de Fixed current level (Niveau de courant fixe) dans la demande est zéro, l'appareil doit quitter le mode fixed Loop current (courant de ligne fixe). Par ailleurs, si la valeur est valide, l'appareil doit être placé en mode fixed Loop current en utilisant la valeur dans la demande. La valeur renvoyée dans la réponse doit être arrondie ou correspondre à la valeur tronquée réellement utilisée par l'appareil.

Le mode Fixed Loop current doit être quitté lorsque l'alimentation est désactivée puis réappliquée à l'appareil.

5.3.28 Write primary variable unit FAL PDU

5.3.28.1 Primitive de demande

La valeur du champ de commande est quarante-quatre (44); la valeur du champ de nombre d'octets est un (1) et le champ de valeur est tel qu'indiqué dans le Tableau 48.

Tableau 48 – Champ de valeur de Write primary variable unit

Position relative en octets	Data type	Nom du paramètre	Description
0	Engineering unit	Primary variable unit code	Code d'unité de mesure de la variable primaire

5.3.28.2 Primitive de réponse

Le champ de valeur de la réponse est décrit dans le Tableau 48. Les valeurs du code de réponse sont indiquées dans le Tableau 49.

Tableau 49 – Codes de réponse spécifiques à la commande Write primary variable unit

Valeur	Classe	Description
0	Success	Absence d'erreurs
2	Error	Choix non valide
5	Error	Nombre d'octets de données reçus trop faible
6	Error	Erreur de commande spécifique à l'appareil
7	Error	Protégé en écriture
16	Error	Accès restreint
32	Error	Occupé

5.3.28.3 Procédure au niveau de l'appareil répondeur

L'appareil doit sélectionner les unités de mesure de la variable primaire et de sa plage. Cette commande sélectionne également l'unité applicable aux limites du transducteur et à l'étendue minimale.

5.3.29 Trim loop current zero FAL PDU**5.3.29.1 Primitive de demande**

La valeur du champ de commande est quarante-cinq (45); la valeur du champ de nombre d'octets est quatre (4) et le champ de valeur est tel qu'indiqué dans le Tableau 50. Dans l'unité APDU de demande, il s'agit du courant de ligne mesuré à l'extérieur.

Tableau 50 – Champ de valeur de la demande Trim loop current zero

Position relative en octets	Data type	Nom du paramètre	Description
0	Float32	Zero loop current	Niveau du courant de ligne, en mA

5.3.29.2 Primitive de réponse

Le champ de valeur de la réponse est décrit dans le Tableau 50. Dans l'unité PDU de réponse, il s'agit du courant de ligne réel utilisé par l'appareil. Les valeurs du code de réponse sont indiquées dans le Tableau 51.

Tableau 51 – Codes de réponse spécifiques à la commande Trim loop current zero

Valeur	Classe	Description
0	Success	Absence d'erreurs
3	Error	Niveau de courant de ligne zéro trop élevé
4	Error	Niveau de courant de ligne zéro trop faible
5	Error	Nombre d'octets de données reçus trop faible
6	Error	Erreur de commande spécifique à l'appareil
7	Error	Protégé en écriture
9	Error	mode ou valeur de courant de ligne incorrect(e)
11	Error	Courant de ligne non actif (appareil en mode multipoint)
16	Error	Accès restreint
32	Error	Occupé

5.3.29.3 Procédure au niveau de l'appareil répondeur

L'appareil de terrain doit ajuster son étalonnage à zéro du courant de ligne de sorte que la valeur de courant de ligne corresponde à la valeur reçue du maître. La valeur renvoyée dans la réponse doit être arrondie ou correspondre à la valeur tronquée réellement utilisée par l'appareil.

NOTE Cet ajustement est généralement réalisé en réglant le courant de ligne à 4,00 mA et en envoyant la valeur mesurée à l'appareil de terrain.

Le code de réponse 9 (mode ou valeur de courant de ligne incorrect(e)) doit être renvoyé si l'appareil n'est pas dans le mode approprié permettant d'étalonner le courant de ligne ou si le courant n'a pas été défini de manière précise à la valeur minimale.

Si l'appareil fonctionne à la tension de sortie, il doit interpréter la valeur de courant de ligne comme valeur de sortie en Volts continus.

5.3.30 Trim loop current gain FAL PDU

5.3.30.1 Primitive de demande

La valeur du champ de commande est quarante-six (46); la valeur du champ de nombre d'octets est quatre (4) et le champ de valeur est tel qu'indiqué dans le Tableau 52. Dans l'unité PDU de demande, il s'agit du courant de ligne mesuré à l'extérieur.

Tableau 52 – Champ de valeur de la demande Trim loop current gain

Position relative en octets	Data type	Nom du paramètre	Description
0	Float32	Gain loop current	Niveau du courant de ligne, en mA

5.3.30.2 Primitive de réponse

Le champ de valeur de la réponse est décrit dans le Tableau 52. Dans l'unité PDU de réponse, il s'agit du courant de ligne réel utilisé par l'appareil. Les valeurs du code de réponse sont indiquées dans le Tableau 53.

Tableau 53 – Codes de réponse spécifiques à la commande Trim loop current gain

Valeur	Classe	Description
0	Success	Absence d'erreurs
3	Error	Niveau de courant de ligne de gain trop élevé
4	Error	Niveau de courant de ligne de gain trop faible
5	Error	Nombre d'octets de données reçus trop faible
6	Error	Erreur de commande spécifique à l'appareil
7	Error	Protégé en écriture
9	Error	Mode ou valeur de courant de ligne incorrect(e)
11	Error	Courant de ligne non actif (appareil en mode multipoint)
16	Error	Accès restreint
32	Error	Occupé

5.3.30.3 Procédure au niveau de l'appareil répondeur

L'appareil de terrain doit ajuster son étalonnage de gain du courant de ligne de sorte que la valeur de courant de ligne corresponde à la valeur reçue du maître. La valeur renvoyée dans la réponse doit être arrondie ou correspondre à la valeur tronquée réellement utilisée par l'appareil.

NOTE Cet ajustement est généralement réalisé en réglant le courant de ligne à 20,00 mA et en envoyant la valeur mesurée à l'appareil de terrain.

Le code de réponse 9 (mode ou valeur de courant de ligne incorrect(e)) doit être renvoyé si l'appareil n'est pas dans le mode approprié permettant d'étalonner le courant de ligne ou si le courant n'a pas été défini de manière précise à la valeur maximale.

Si l'appareil fonctionne à la tension de sortie, il doit interpréter la valeur de courant de ligne comme valeur de sortie en Volts continus.

5.3.31 Read dynamic variable assignment FAL PDU

5.3.31.1 Primitive de demande

La valeur du champ de commande est cinquante (50); la valeur du champ de nombre d'octets est zéro (0) et le champ de valeur est null.

5.3.31.2 Primitive de réponse

La valeur du champ de nombre d'octets est quatre (4) et le champ de valeur est tel qu'indiqué dans le Tableau 54. Les valeurs du code de réponse sont indiquées dans le Tableau 55.

Tableau 54 – Champ de valeur de la réponse Read dynamic variable assignment

Position relative en octets	Data type	Nom du paramètre	Description
0	Unsigned8	Primary variable assignment	Variable d'appareil attribuée à la variable primaire comme spécifié dans le Tableau C.9.
1	Unsigned8	Secondary variable assignment	Variable d'appareil attribuée à la variable secondaire comme spécifié dans le Tableau C.9.
2	Unsigned8	Tertiary variable assignment	Variable d'appareil attribuée à la variable tertiaire comme spécifié dans le Tableau C.9.
3	Unsigned8	Quaternary variable	Variable d'appareil attribuée à la variable quaternaire

Position relative en octets	Data type	Nom du paramètre	Description
		assignment	comme spécifié dans le Tableau C.9.

Tableau 55 – Codes de réponse spécifiques à la commande Read dynamic variable assignment

Valeur	Classe	Description
0	Success	Absence d'erreurs
6	Error	Erreur de commande spécifique à l'appareil
16	Error	Accès restreint

5.3.31.3 Procédure au niveau de l'appareil répondeur

L'appareil doit répondre avec les numéros de variable d'appareil attribués aux variables primaire, secondaire, tertiaire et quaternaire. L'appareil de terrain doit renvoyer tous les octets de données de réponse. La commande Dynamic variable assignment non prise en charge doit être définie sur "250" (non utilisé).

5.3.32 Write dynamic variable assignment FAL PDU

5.3.32.1 Primitive de demande

La valeur du champ de commande est cinquante et un (51); la valeur du champ de nombre d'octets est quatre (4) et le champ de valeur est tel qu'indiqué dans le Tableau 56.

Tableau 56 – Champ de valeur de Write dynamic variable assignment

Position relative en octets	Data type	Nom du paramètre	Description
0	Unsigned8	Primary variable assignment	Variable d'appareil attribuée à la variable primaire comme spécifié dans le Tableau C.9.
1	Unsigned8	Secondary variable assignment	Variable d'appareil attribuée à la variable secondaire comme spécifié dans le Tableau C.9.
2	Unsigned8	Tertiary variable assignment	Variable d'appareil attribuée à la variable tertiaire comme spécifié dans le Tableau C.9.
3	Unsigned8	Quaternary variable assignment	Variable d'appareil attribuée à la variable quaternaire comme spécifié dans le Tableau C.9.

5.3.32.2 Primitive de réponse

Le champ de valeur de la réponse est décrit dans le Tableau 56. Les valeurs du code de réponse sont indiquées dans le Tableau 57.

Tableau 57 – Codes de réponse spécifiques à la commande Write dynamic variable assignment

Valeur	Classe	Description
0	Success	Absence d'erreurs
2	Error	Choix non valide
5	Error	Nombre d'octets de données reçus trop faible
6	Error	Erreur de commande spécifique à l'appareil

7	Error	Protégé en écriture
16	Error	Accès restreint
32	Error	Occupé

5.3.32.3 Procédure au niveau de l'appareil répondeur

L'appareil doit écrire les numéros de variable d'appareil attribués aux variables primaire, secondaire, tertiaire et quaternaire. Si la demande dispose de un ou plusieurs octets de données, elle ne doit pas répondre avec "Nombre d'octets de données reçus trop faible". La variable Assignment (attribution) non incluse dans la demande peut être définie sur toute variable d'appareil valide dans l'appareil de terrain. L'appareil de terrain doit toujours renvoyer les quatre octets de données de réponse. La commande Dynamic variable assignment non prise en charge doit être définie sur "250" (non utilisé).

5.3.33 Write number of response preambles FAL PDU

5.3.33.1 Primitive de demande

La valeur du champ de commande est cinquante-neuf (59); la valeur du champ de nombre d'octets est un (1) et le champ de valeur est tel qu'indiqué dans le Tableau 58.

Tableau 58 – Champ de valeur de Write number of response preambles

Position relative en octets	Data type	Nom du paramètre	Description
0	Unsigned8	Preamble count	Nombre de caractères de préambule à envoyer dans la PhPDU de l'esclave au maître

5.3.33.2 Primitive de réponse

Le champ de valeur de la réponse est décrit dans le Tableau 58. Les valeurs du code de réponse sont indiquées dans le Tableau 59.

Tableau 59 – Codes de réponse spécifiques à la commande Write number of response preambles

Valeur	Classe	Description
0	Success	Absence d'erreurs
3	Error	Nombre de préambules trop élevé
4	Error	Nombre de préambules trop faible
5	Error	Nombre d'octets de données reçus trop faible
6	Error	Erreur de commande spécifique à l'appareil
7	Error	Protégé en écriture
8	Warning	Défini sur la valeur possible la plus proche
16	Error	Accès restreint
32	Error	Occupé

5.3.33.3 Procédure au niveau de l'appareil répondeur

L'appareil doit utiliser le paramètre Preamble count reçu dans cette demande comme la longueur du préambule dans toutes les PhPDU envoyée par l'appareil. Si la valeur reçue est inférieure à cinq (5) ou supérieure à vingt (20), l'appareil doit rejeter la commande et envoyer le code de réponse approprié.

5.3.34 Read device variable trim points FAL PDU

5.3.34.1 Primitive de demande

La valeur du champ de commande est quatre-vingt (80); la valeur du champ de nombre d'octets est un (1) et le champ de valeur est tel qu'indiqué dans le Tableau 60.

Tableau 60 – Champ de valeur de la demande Read device variable trim points

Position relative en octets	Data type	Nom du paramètre	Description
0	Unsigned8	Device variable	Code de variable d'appareil comme spécifié dans le Tableau C.9

5.3.34.2 Primitive de réponse

La valeur du champ de nombre d'octets est dix (10) et le champ de valeur est tel qu'indiqué dans le Tableau 61. Les valeurs du code de réponse sont indiquées dans le Tableau 62.

Tableau 61 – Champ de valeur de la réponse Read device variable trim points

Position relative en octets	Data type	Nom du paramètre	Description
0	Unsigned8	Device variable	Code de variable d'appareil comme spécifié dans le Tableau C.9
1	Engineering unit	Trim point unit	Code d'unité de mesure des points d'ajustement
2	Float32	Lower trim point	Valeur de point d'ajustement unique ou inférieure
6	Float32	Upper trim point	Valeur de point d'ajustement supérieure

Tableau 62 – Codes de réponse spécifiques à la commande Read device variable trim points

Valeur	Classe	Description
0	Success	Absence d'erreurs
5	Error	Nombre d'octets de données reçus trop faible
6	Error	Erreur de commande spécifique à l'appareil
17	Error	Code de variable d'appareil non valide, c'est-à-dire la variable d'appareil demandée n'existe pas.
19	Error	Code de variable d'appareil non admis pour cette commande
32	Error	Occupé

5.3.34.3 Procédure au niveau de l'appareil répondeur

L'appareil doit répondre avec la valeur la plus récente utilisée pour les valeurs de point d'ajustement. Si la variable d'appareil ne prend pas en charge l'ajustement, les points d'ajustement supérieurs et inférieurs doivent être 'Not-A-Number'. Si la variable d'appareil ne prend en charge qu'un point d'ajustement unique, le point d'ajustement supérieur doit être 'Not-A-Number'.

5.3.35 Read device variable trim guidelines FAL PDU

5.3.35.1 Primitive de demande

La valeur du champ de commande est quatre-vingt-un (81); la valeur du champ de nombre d'octets est un (1) et le champ de valeur est tel qu'indiqué dans le Tableau 63.

Tableau 63 – Champ de valeur de la demande Read device variable trim guidelines

Position relative en octets	Data type	Nom du paramètre	Description
0	Unsigned8	Device variable	Code de variable d'appareil comme spécifié dans le Tableau C.9

5.3.35.2 Primitive de réponse

La valeur du champ de nombre d'octets est vingt-trois (23) et le champ de valeur est tel qu'indiqué dans le Tableau 64. Les valeurs du code de réponse sont indiquées dans le Tableau 65.

Tableau 64 – Champ de valeur de la réponse Read device variable trim guidelines

Position relative en octets	Data type	Nom du paramètre	Description
0	Unsigned8	Device variable	Code de variable d'appareil comme spécifié dans le Tableau C.9
1	Enumeration	Trim points supported	Points d'ajustement pouvant être définis dans l'appareil comme « enumerated » dans le Tableau C.7
2	Engineering unit	Trim point unit	Code d'unité de mesure des points d'ajustement
3	Float32	Minimum lower trim point	Valeur minimale acceptée par l'appareil au cours d'une procédure d'ajustement inférieur
7	Float32	Maximum lower trim point	Valeur maximale acceptée par l'appareil au cours d'une procédure d'ajustement inférieur
11	Float32	Minimum upper trim point	Valeur minimale acceptée par l'appareil au cours d'une procédure d'ajustement supérieur
15	Float32	Maximum upper trim point	Valeur maximale acceptée par l'appareil au cours d'une procédure d'ajustement supérieur
19	Float32	Minimum trim point difference	Différence minimale acceptée entre les points d'ajustement supérieur et inférieur acceptée par l'appareil au cours d'une procédure d'ajustement.

Tableau 65 – Codes de réponse spécifiques à la commande Read device variable trim points

Valeur	Classe	Description
0	Success	Absence d'erreurs
5	Error	Nombre d'octets de données reçus trop faible
6	Error	Erreur de commande spécifique à l'appareil
17	Error	Code de variable d'appareil non valide, c'est-à-dire la variable d'appareil demandée n'existe pas.
19	Error	Code de variable d'appareil non admis pour cette commande
32	Error	Occupé

5.3.35.3 Procédure au niveau de l'appareil répondeur

L'appareil doit répondre avec l'information indiquant qu'un Hôte nécessite de guider l'utilisateur pour réaliser une bonne sélection des points d'ajustement. Si la variable d'appareil ne prend pas en charge l'ajustement, les points d'ajustement pris en charge doivent être zéro (0). L'unité de point d'ajustement doit être 'Not used' (non utilisé) et toutes les valeurs de point d'ajustement doivent être 'Not-A-Number'. Si la variable d'appareil ne prend en charge qu'un point d'ajustement unique, la différence entre le point d'ajustement supérieur et le point d'ajustement doit être 'Not-A-Number'.

5.3.36 Write device variable trim point FAL PDU

5.3.36.1 Primitive de demande

La valeur du champ de commande est quatre-vingt-deux (82); la valeur du champ de nombre d'octets est sept (7) et le champ de valeur est tel qu'indiqué dans le Tableau 66.

Tableau 66 – Champ de valeur de Write device variable trim point

Position relative en octets	Data type	Nom du paramètre	Description
0	Unsigned8	Device variable	Code de variable d'appareil comme spécifié dans le Tableau C.9
1	Enumeration	Trim points type	Points d'ajustement pouvant être définis dans l'appareil comme « enumerated » dans le Tableau C.7
2	Engineering unit	Trim point unit	Code d'unité de mesure du point d'ajustement
3	Float32	Trim point value	Valeur de point d'ajustement à écrire

5.3.36.2 Primitive de réponse

Le champ de valeur de la réponse est décrit dans le Tableau 66. Les valeurs du code de réponse sont indiquées dans le Tableau 67.

Tableau 67 – Codes de réponse spécifiques à la commande Write device variable trim point

Valeur	Classe	Description
0	Success	Absence d'erreurs
2	Error	Choix non valide
3	Error	Valeur de point d'ajustement trop élevée
4	Error	Valeur de point d'ajustement trop faible
5	Error	Nombre d'octets de données reçus trop faible
6	Error	Erreur de commande spécifique à l'appareil
7	Error	Protégé en écriture
9	Error	Processus appliqué trop élevé
10	Error	Processus appliqué trop faible
11	Error	Erreur d'ajustement, tentative de correction supplémentaire
13	Error	Erreur de calcul, valeurs d'ajustement non modifiées
14	Warning	Variable Span (étendue) trop petite
16	Error	Accès restreint
17	Error	Code de variable d'appareil non valide, c'est-à-dire la variable d'appareil demandée n'existe pas.

Valeur	Classe	Description
18	Error	Code Units non valide
19	Error	Code de variable d'appareil non admis pour cette commande
32	Error	Occupé – une réponse différée ne peut être démarrée.
33	Error	DR initiée
34	Error	DR en fonctionnement
35	Error	DR arrêtée
36	Error	DR en conflit

5.3.36.3 Procédure au niveau de l'appareil répondeur

Cette commande réalise un ajustement d'étalonnage de la variable d'appareil demandée. Il convient que la valeur de point d'ajustement envoyée dans la demande représente la valeur de variable de processus réellement appliquée. A réception de cette demande, l'appareil doit vérifier la validité de la valeur (c'est-à-dire dans les limites acceptables), puis tenter d'ajuster le point indiqué dans son calcul étalonné de sorte que la valeur de processus numérique qui en résulte corresponde à la valeur fournie dans la demande.

Si l'appareil ne peut pas réaliser l'ajustement au cours de la période de réponse autorisée par la DLE, l'appareil doit utiliser le mécanisme de réponse différée pour informer le maître que l'étalonnage se poursuit.

5.3.37 Reset device variable trim FAL PDU

5.3.37.1 Primitive de demande

La valeur du champ de commande est quatre vingt-trois (83); la valeur du champ de nombre d'octets est sept (7) et le champ de valeur est tel qu'indiqué dans le Tableau 68.

Tableau 68 – Reset device variable trim champ de valeur

Position relative en octets	Data type	Nom du paramètre	Description
0	Unsigned8	Device variable	Code de variable d'appareil comme spécifié dans le Tableau C.9

5.3.37.2 Primitive de réponse

Le champ de valeur de la réponse est décrit dans le Tableau 68. Les valeurs du code de réponse sont indiquées dans le Tableau 69.

Tableau 69 – Codes de réponse spécifiques à la commande Reset device variable trim

Valeur	Classe	Description
0	Success	Absence d'erreurs
2	Error	Choix non valide
5	Error	Nombre d'octets de données reçus trop faible
7	Error	Protégé en écriture
16	Error	Accès restreint
17	Error	Code de variable d'appareil non valide, c'est-à-dire la variable d'appareil demandée n'existe pas.
19	Error	Code de variable d'appareil non admis pour cette commande
29	Error	Variable Span non valide