

INTERNATIONAL STANDARD

NORME INTERNATIONALE

**Electronic railway equipment – Train communication network (TCN) –
Part 3-3: CANopen Consist Network (CCN)**

**Matériel électronique ferroviaire – Réseau embarqué de train (TCN) –
Partie 3-3: Réseau de rame CANopen (CCN)**





THIS PUBLICATION IS COPYRIGHT PROTECTED

Copyright © 2012 IEC, Geneva, Switzerland

All rights reserved. Unless otherwise specified, no part of this publication may be reproduced or utilized in any form or by any means, electronic or mechanical, including photocopying and microfilm, without permission in writing from either IEC or IEC's member National Committee in the country of the requester.

If you have any questions about IEC copyright or have an enquiry about obtaining additional rights to this publication, please contact the address below or your local IEC member National Committee for further information.

Droits de reproduction réservés. Sauf indication contraire, aucune partie de cette publication ne peut être reproduite ni utilisée sous quelque forme que ce soit et par aucun procédé, électronique ou mécanique, y compris la photocopie et les microfilms, sans l'accord écrit de la CEI ou du Comité national de la CEI du pays du demandeur.

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

IEC Central Office
3, rue de 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.

Useful links:

IEC publications search - www.iec.ch/searchpub

The advanced search enables you to find IEC publications by a variety of criteria (reference number, text, technical committee,...).

It also gives information on projects, replaced and withdrawn publications.

IEC Just Published - webstore.iec.ch/justpublished

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

Electropedia - www.electropedia.org

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

Customer Service Centre - webstore.iec.ch/csc

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

A propos de la CEI

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

A propos des publications CEI

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

Liens utiles:

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

La recherche avancée vous permet de trouver des publications CEI en utilisant différents critères (numéro de référence, texte, comité d'études,...).

Elle donne aussi des informations sur les projets et les publications remplacées ou retirées.

Just Published CEI - webstore.iec.ch/justpublished

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

Electropedia - www.electropedia.org

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

Service Clients - webstore.iec.ch/csc

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



IEC 61375-3-3

Edition 1.0 2012-06

INTERNATIONAL STANDARD

NORME INTERNATIONALE

**Electronic railway equipment – Train communication network (TCN) –
Part 3-3: CANopen Consist Network (CCN)**

**Matériel électronique ferroviaire – Réseau embarqué de train (TCN) –
Partie 3-3: Réseau de rame CANopen (CCN)**

INTERNATIONAL
ELECTROTECHNICAL
COMMISSION

COMMISSION
ELECTROTECHNIQUE
INTERNATIONALE

PRICE CODE
CODE PRIX

XE

ICS 45.060

ISBN 978-2-88912-072-7

**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.....	10
INTRODUCTION.....	12
1 Scope.....	13
2 Normative references	13
3 Terms, definitions and abbreviations	14
3.1 Terms and definitions	14
3.2 Abbreviations	15
3.3 Conventions	15
4 Architecture.....	15
4.1 Content	15
4.2 Logical CANopen-based consist network	15
4.3 Network topology.....	16
4.4 Addressing	16
4.5 Data classes	17
5 Physical layer.....	17
5.1 Content	17
5.2 Cabling.....	17
5.3 Connector	17
5.4 Physical medium attachment	19
5.5 Physical signaling.....	19
6 Data Link layer	19
6.1 Content	19
6.2 CANopen data link layer	20
7 CANopen application layer	20
7.1 Content	20
7.2 Reference model	20
7.3 Field device model	20
7.4 CANopen communication objects	22
7.5 CANopen object dictionary	22
7.6 Predefined CANopen communication objects	24
7.6.1 Content	24
7.6.2 Object 1000 _h : Device type	24
7.6.3 Object 1001 _h : Error register.....	24
7.6.4 Object 1014 _h : COB-ID emergency object.....	24
7.6.5 Object 1017 _h : Heartbeat producer	24
7.6.6 Object 1018 _h : Identity object.....	24
7.6.7 Object 1029 _h : Error behavior	24
7.6.8 Object 67FF _h : Device type.....	25
7.6.9 Service data objects (SDOs).....	25
7.6.10 Process data objects (PDOs).....	25
8 Application data.....	25
8.1 Content	25
8.2 CANopen application data representation.....	25
8.3 Recommended representation principle of application data	25
8.3.1 Content	25

8.3.2	Application data for door control	25
8.3.3	Consumed door control application objects	26
8.3.4	Produced door control application objects	27
9	CANopen network management	29
9.1	Content	29
9.2	CANopen NMT slave functionality	30
9.3	CANopen manager functionality	30
9.3.1	General	30
9.3.2	Object dictionary usage	31
9.3.3	Redundant networks	31
9.4	CANopen NMT start-up	32
9.4.1	NMT startup	32
9.4.2	NMT startup simple	35
9.4.3	Start process boot NMT slave	36
9.5	Boot NMT slave	37
9.5.1	Check configuration	42
9.5.2	Check NMT state	43
9.5.3	NMT flying master start up	43
9.5.4	Error status	44
9.6	Error control	45
9.6.1	Start error control	45
9.6.2	Error handler	46
9.6.3	Bootup handler	47
9.7	Additional NMT master services and protocols	47
9.8	Object dictionary entries	47
9.8.1	Object 1020 _h : Verify configuration	47
9.8.2	Object 102A _h : NMT inhibit time	48
9.8.3	Object 1F20 _h : Store DCF	49
9.8.4	Object 1F22 _h : Concise DCF	50
9.8.5	Object 1F26 _h : Expected configuration date	52
9.8.6	Object 1F27 _h : Expected configuration time	53
9.8.7	Object 1F80 _h : NMT startup	54
9.8.8	Object 1F81 _h : NMT slave assignment	56
9.8.9	Object 1F82 _h : Request NMT	58
9.8.10	Object 1F83 _h : Request node guarding	61
9.8.11	Object 1F84 _h : Device type identification	63
9.8.12	Object 1F85 _h : Vendor identification	64
9.8.13	Object 1F86 _h : Product code	65
9.8.14	Object 1F87 _h : Revision number	66
9.8.15	Object 1F88 _h : Serial number	67
9.8.16	Object 1F89 _h : Boot time	68
9.8.17	Object 1F8A _h : Restore configuration	69
9.8.18	Object 1F91 _h : Self starting nodes timing parameters	70
10	Gateway functions	71
10.1	Content	71
10.2	Gateway architecture	72
10.3	General principles and services	73
10.3.1	Content	73
10.3.2	Gateway class definitions	73

10.3.3	Service primitives definitions	73
10.4	Network access service specification.....	73
10.4.1	SDO access services.....	73
10.4.2	PDO access services.....	75
10.4.3	CANopen NMT services.....	78
10.4.4	Device failure management services	81
10.4.5	CANopen interface configuration services	82
10.4.6	Gateway management services	84
10.4.7	Manufacturer-specific services	85
10.5	ASCII mapping of network access services	85
10.5.1	Content	85
10.5.2	Definitions	86
10.5.3	Network access command specification.....	89
11	Train network management	97
11.1	Content	97
11.2	Manager, Agents and interfaces (informative).....	98
11.3	Management message protocol (informative)	98
11.4	Object interfaces (informative).....	98
11.5	CANopen-specific management services	98
11.5.1	General	98
11.5.2	Agent interfaces on a Station connected to CANopen consist network	98
11.5.3	Management message structure for CANopen consist networks	99
11.5.4	Notation for the CANopen specific SIF_codes	99
11.5.5	Notation for a call CANopen management message	100
11.5.6	Notation for a reply CANopen management message	100
11.5.7	Notation for the TNM CANopen services command codes	100
11.6	TNM CANopen services	101
11.6.1	Content	101
11.6.2	Call_Write_CANopen_Command (with reservation)	101
11.6.3	Reply_Write_CANopen_Command (with reservation)	102
11.6.4	Call_Read_CANopen_Command (without reservation)	102
11.6.5	Reply_Read_CANopen_Command (without reservation).....	103
12	CANopen management message data handling.....	103
12.1	General.....	103
12.2	Message data format.....	105
12.3	Requirements for message data communication within CANopen networks	105
12.4	Object 1F78 _h : CANopen message data reception	106
13	Conformance testing	107
	Bibliography.....	108
	Figure 1 – Logical network architecture of the consist network.....	16
	Figure 2 – Network topology of CANopen-based consist network.....	16
	Figure 3 – 9-pin D-sub connector.....	18
	Figure 4 – 5-pin micro style connector	18
	Figure 5 – Field device model	20
	Figure 6 – Minimum field device.....	21
	Figure 7 – CANopen device structure.....	22

Figure 8 – Structure of the device type object	24
Figure 9 – Object structure	26
Figure 10 – Object structure	27
Figure 11 – Object structure	28
Figure 12 – NMT startup, part 1	32
Figure 13 – NMT startup, part 2	34
Figure 14 – NMT startup simple	35
Figure 15 – Start process boot NMT slave	36
Figure 16 – Boot NMT slave, part 1	37
Figure 17 – Boot NMT slave, part 2	39
Figure 18 – Boot NMT slave, part 3	40
Figure 19 – Check configuration	42
Figure 20 – Check NMT state	43
Figure 21 – Start error control	45
Figure 22 – Error handler	46
Figure 23 – Bootup handler	47
Figure 24 – Data stream definition of concise DCF	51
Figure 25 – Object structure	54
Figure 26 – Bit structure of the configuration value	54
Figure 27 – Object structure of the value	56
Figure 28 – Bit structure of the configuration value	57
Figure 29 – Gateway between Train backbone and CANopen consist network	72
Figure 30 – Management messages (informative)	97
Figure 31 – Agent interface on a CANopen (gateway) station for message data	99
Figure 32 – Call_Write_CANopen_Command	102
Figure 33 – Reply_Write_CANopen_Command	102
Figure 34 – Call_Read_CANopen_Command (without reservation)	103
Figure 35 – Reply_Read_CANopen_command (without reservation)	103
Figure 36 – CANopen device capable to handle TNM management messages	104
Figure 37 – Message data format comparison	105
Table 1 – Pinning for 9-pin D-sub connector	18
Table 2 – Pinning for 5-pin micro style connector	19
Table 3 – Bit timing	19
Table 4 – CANopen object dictionary structure	23
Table 5 – Value definition	26
Table 6 – Object description	26
Table 7 – Entry description	27
Table 8 – Value definition	27
Table 9 – Object description	27
Table 10 – Entry description	28
Table 11 – Value definition	29
Table 12 – Object description	29

Table 13 – Entry description	29
Table 14 – Error status	44
Table 15 – Object description	48
Table 16 – Entry description	48
Table 17 – Object description	49
Table 18 – Entry description	49
Table 19 – Object description	49
Table 20 – Entry description	50
Table 21 – Object description	51
Table 22 – Entry description	52
Table 23 – Object description	52
Table 24 – Entry description	53
Table 25 – Object description	53
Table 26 – Entry description	54
Table 27 – Value NMT master (bit: 0)	55
Table 28 – Value Start all nodes (bit: 1)	55
Table 29 – Value NMT master start (bit: 2)	55
Table 30 – Value Start node (bit: 3)	55
Table 31 – Reset all nodes (bit: 4)	55
Table 32 – Flying master (bit: 5)	55
Table 33 – Stop all nodes (bit: 6)	55
Table 34 – Exceptions for NMT start-up capable devices	56
Table 35 – Object description	56
Table 36 – Entry description	56
Table 37 – NMT slave (bit: 0)	57
Table 38 – NMT boot slave (bit: 2)	57
Table 39 – Mandatory (bit: 3)	57
Table 40 – Reset communication (bit: 4)	57
Table 41 – Software version (bit: 5)	57
Table 42 – Software update (bit: 6)	57
Table 43 – Restore (bit: 7)	58
Table 44 – Object description	58
Table 45 – Entry description	58
Table 46 – Value definition	60
Table 47 – Object description	60
Table 48 – Entry description	61
Table 49 – Value definition	62
Table 50 – Object description	62
Table 51 – Entry description	63
Table 52 – Object description	64
Table 53 – Entry description	64
Table 54 – Object description	65
Table 55 – Entry description	65

Table 56 – Object description	66
Table 57 – Entry description	66
Table 58 – Object description	67
Table 59 – Entry description	67
Table 60 – Object description	68
Table 61 – Entry description	68
Table 62 – Object description	69
Table 63 – Entry description	69
Table 64 – Object description	69
Table 65 – Entry description	70
Table 66 – Object description	70
Table 67 – Entry description	71
Table 68 – Upload SDO service	74
Table 69 – Download SDO parameters	75
Table 70 – Configure SDO timeout parameters	75
Table 71 – Configure RPDO service parameters	76
Table 72 – Configure TPDO service parameters	77
Table 73 – Read PDO data service parameters	77
Table 74 – Write PDO data service parameters.....	78
Table 75 – RPDO received service parameters.....	78
Table 76 – Start node service parameters.....	78
Table 77 – Stop node service parameters	79
Table 78 – Set node to pre-operational service parameters	79
Table 79 – Reset node service parameters	79
Table 80 – Reset communication service parameters.....	80
Table 81 – Enable node guarding service parameters	80
Table 82 – Disable node guarding service parameters	80
Table 83 – Start heartbeat consumer service parameters	81
Table 84 – Disable heartbeat consumer service parameters	81
Table 85 – Error control event received parameters	81
Table 86 – Read device error service parameters	82
Table 87 – Emergency event received service parameters.....	82
Table 88 – Initialize gateway service parameters	82
Table 89 – Store configuration service parameters	83
Table 90 – Restore configuration service parameters.....	83
Table 91 – Set heartbeat producer service parameters	83
Table 92 – Set node-ID service parameters	84
Table 93 – Start emergency consumer service parameters	84
Table 94 – Stop emergency consumer service parameters.....	84
Table 95 – Set default network service parameters	85
Table 96 – Start default node-ID service parameters	85
Table 97 – Get version service parameters	85
Table 98 – Syntax and CANopen data types	86

Table 99 – Command notation in BNF.....	87
Table 100 – Response notation.....	88
Table 101 – Internal error code (InEC).....	88
Table 102 – Notation for event triggered messages	88
Table 103 – Syntax for upload SDO command	89
Table 104 – Examples for upload SDO command	89
Table 105 – Syntax for Download SDO command	89
Table 106 – Examples for download SDO command	89
Table 107 – Syntax for configure SDO timeout command.....	89
Table 108 – Syntax for configure RPDO command.....	90
Table 109 – Examples for configure RPDO command	90
Table 110 – Syntax for configure TPDO command	90
Table 111 – Examples for configure TPDO command.....	90
Table 112 – Syntax for read PDO data command.....	91
Table 113 – Response syntax for read PDO data command.....	91
Table 114 – Syntax for write PDO data command	91
Table 115 – Syntax for RPDO receive command.....	91
Table 116 – Examples RPDO received command	91
Table 117 – Syntax for start node command	91
Table 118 – Syntax for stop node command	92
Table 119 – Syntax set node to pre-operational command.....	92
Table 120 – Syntax reset node command	92
Table 121 – Syntax reset communication command.....	92
Table 122 – Syntax enable node guarding command	92
Table 123 – Syntax disable node guarding command.....	93
Table 124 – Syntax start heartbeat consumer command	93
Table 125 – Syntax disable heartbeat consumer command	93
Table 126 – Syntax for error control event received command	93
Table 127 – Syntax for read device error command	94
Table 128 – Syntax for emergency event received command	94
Table 129 – Syntax for initialize gateway command	94
Table 130 – Bit rate indices	94
Table 131 – Syntax for store configuration command.....	95
Table 132 – Storage specifier	95
Table 133 – Syntax restore configuration command.....	95
Table 134 – Syntax set heartbeat producer command.....	95
Table 135 – Syntax set node-ID command.....	95
Table 136 – Syntax set default network command.....	96
Table 137 – Syntax set default node-ID command	96
Table 138 – Syntax for get version command.....	96
Table 139 – Response syntax for get version command.....	96
Table 140 – Example for get version response.....	97
Table 141 – Management message structure.....	99

Table 142 – CANopen specific SIF_codes	100
Table 143 – Notation for a call CANopen management message	100
Table 144 – Notation for a reply CANopen management message	100
Table 145 – TNM CANopen services command codes (reservation required)	101
Table 146 – TNM CANopen services command codes (reservation not required)	101
Table 147 – Value definition for Call_Write_CANopen_Command	102
Table 148 – Value definition Reply_Write_CANopen_Command	102
Table 149 – Value definition for Call_Read_CANopen_Command (without reservation)	103
Table 150 – Value definition for Reply_Read_CANopen_Command (without reservation)	103
Table 151 – Object description	106
Table 152 – Entry description	106

INTERNATIONAL ELECTROTECHNICAL COMMISSION

**ELECTRONIC RAILWAY EQUIPMENT –
TRAIN COMMUNICATION NETWORK (TCN) –****Part 3-3: CANopen Consist Network (CCN)**

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.

International Standard IEC 61375-3-3 has been prepared by IEC technical committee 9: Electrical equipment and systems for railways.

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

FDIS	Report on voting
9/1646/FDIS	9/1670/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 the ISO/IEC Directives, Part 2.

A list of all parts of IEC 61375 series, under the general title *Electronic railway equipment – Train Communication Network (TCN)*, can be found on the IEC website.

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.

INTRODUCTION

TCN is an International Standard with the aim of defining interfaces so as to achieve plug-in compatibility:

- a) between equipment located in different vehicles or consists, and
- b) between equipment and devices located within the same vehicle or consist.

One of the key success factors for the deployment of any technology is standardization and ensuring interoperability among various implementations. To facilitate interoperability a conformance test should be implemented.

In this part of IEC 61375, the TCN deals with:

the consist network based on CANopen.

In addition gateway devices between the Train Backbone and the CANopen-based consist network are considered.

This standard is structured into 13 clauses.

ELECTRONIC RAILWAY EQUIPMENT – TRAIN COMMUNICATION NETWORK (TCN) –

Part 3-3: CANopen Consist Network (CCN)

1 Scope

This part of IEC 61375 specifies the data communication bus inside consists that are based on CANopen. CANopen was developed for use in, but is not limited to, industrial automation applications. These applications may include devices such as input/output modules, motion controllers, human machine interfaces, sensors, closed-loop controllers, encoders, hydraulic valves or programmable controllers.

In the application field of rail vehicles CANopen networks are utilized to network subsystems in consists such as e.g. brake control system, diesel engine control system and interior or exterior lighting control system. In addition CANopen is utilized as consist network to enable the data exchange between the different subsystems within one single rail vehicle or a group of rail vehicles sharing the same Consist Network.

This part of IEC 61375 applies to all equipment and devices operated on a CANopen-based consist network within TCN architecture as described in IEC 61375-1.

The applicability of this standard to a TCN implementation allows for individual conformance checking of the implementation itself and is a pre-requisite for further interoperability checking between different TCN implementations. In any case, proof of compatibility between Train Backbone and the Consist Network will have to be brought by the supplier.

This part of IEC 61375 applies to the architecture of communication systems in Open trains. In addition it may be applicable to closed trains and multiple unit trains when so agreed between purchaser and supplier.

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.

IEC 61131 (all parts): *Programmable controllers*

IEC 61375-1, ed3, *Electronic railway equipment – Train Communication Network (TCN) – Part 1: General Architecture*

IEC 61375-2-1, *Electronic railway equipment – Train Communication Network (TCN) – Part 2-1: Wire Train Bus (WTB)*

IEC 61375-2-2, *Electronic railway equipment – Train Communication Network (TCN) – Part 2-2: WTB – Wire Train Bus conformance testing*

ISO/IEC 646:1991 *Information technology – ISO 7-bit coded character set for information interchange*

ISO/IEC 9899:1999, *Programming languages – C*

ISO 11898-1:2003, *Road vehicles – Controller area network (CAN) – Part 1: Data link layer and physical signaling*

ISO 11898-2:2003, *Road vehicles – Controller area network (CAN) – Part 2: High-speed medium access unit*

EN 50325-4:2002, *Industrial communication subsystems based on ISO 11898 (CAN) for controller- device interfaces – Part 4: CANopen*

3 Terms, definitions and abbreviations

3.1 Terms and definitions

For the purposes of this document, the terms and definitions given in ISO 11898-1, ISO 11898-2, IEC 61375-1, IEC 61375-2-1 and EN 50325-4, as well as the following apply.

3.1.1

ASCII

7-bit coded character set according to ISO/IEC 646

3.1.2

CANopen device

End Device connected to a CANopen-based consist network

3.1.3

Field device

networked independent physical entity of an automation system capable of performing specified functions in a particular context and delimited by its interfaces

3.1.4

logical consist network

the way the data passes through the consist network without considering the physical interconnection of the End Devices

3.1.5

logical device

representation of a field device in terms of its objects and behavior according to a field device model that describes the device's data and behavior as viewed through a network

3.1.6

Layer Setting Services

services for adjusting bit rate and Node-ID via the communication interface of the CANopen device

3.1.7

Node-ID

- a) network-wide unique identifier for each CANopen device
- b) 7-bit coded Device Address in CANopen-based consist networks

3.1.8

Object

entity with a well-defined boundary and identity that encapsulates state and behavior

3.1.9

virtual device

entity of software capable of accomplishing a functional element of a field device

3.2 Abbreviations

For the purposes of this document, abbreviations given in ISO 11898-1, ISO 11898-2, IEC 61375-1, IEC 61375-2-1 and EN 50325-4, as well as the following apply.

ASCII	American Standard Code for Information Interchange
BNF	Backus Naur Form
CAN	Controller Area Network
CPU	Central Processing Unit
CR	Carriage Return
CRLF	Carriage Return and Line Feed
DCF	Device configuration file
FSA	Finite state automaton
ID	Identifier
LF	Line Feed
LSS	Layer Setting Services
NMT	Network management
OSI	Open system interconnect
PAS	PDO access service
PDO	Process data object
SAS	Service data object access service
SRD	SDO requesting device

3.3 Conventions

The conventions given in IEC 61375-1 shall apply.

4 Architecture

4.1 Content

This clause provides the definition of the architecture for Consist Networks based on CANopen.

4.2 Logical CANopen-based consist network

The overall Logical CANopen consist network, connecting several Virtual Devices, is illustrated in Figure 1. The CANopen-based Consist Network interconnects several Virtual Devices and sub-systems such as e.g. Train operating system (TOS), Monitoring and safety system (MSS), Auxiliary operating system (AUX), Power (drive) system (PDS), Running gear system (RGS), Brake control system (BCS), Ancillary operating system (ANC), Vehicle linkage system (VLS), Exterior lighting system (ELS), Interior lighting system (ILS), Door control system (DCS), HVAC system (HS), Passenger information system (PIS), Diagnostic system (DS), or Train-to-ground communication system (TCS).

The Logical CANopen-based consist network is connected to the Train Backbone via a Gateway. The Gateway manages the information exchange as well as the process data marshalling between the Train Backbone and the Consist Network.

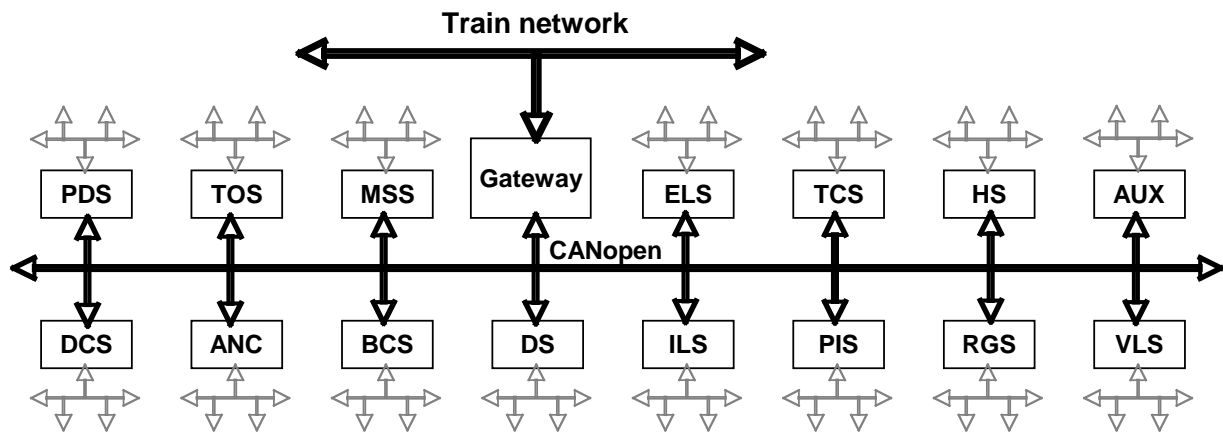


Figure 1 – Logical network architecture of the consist network

4.3 Network topology

The network topology shall be as close as possible to a single line structure with termination resistors on both sides of the network as suggested by ISO11898-2. The total network length shall not exceed 450 m, if a transmission rate of 125 kbit/s is used.

The accumulated stub length shall not exceed 110 m, and the single stub length shall not exceed 22 m, in case a transmission rate of 125 kbit/s is used. It is recommended to keep the stub lengths as short as possible. The CAN transceiver chips shall be galvanically isolated. Optocouplers are placed between CAN controller and CAN transceiver. This affects the maximum bus length depending upon the propagation delay of the optocouplers. The termination resistor on both ends of the bus lines shall be 120 Ω or higher.

The network topology is illustrated in Figure 2.

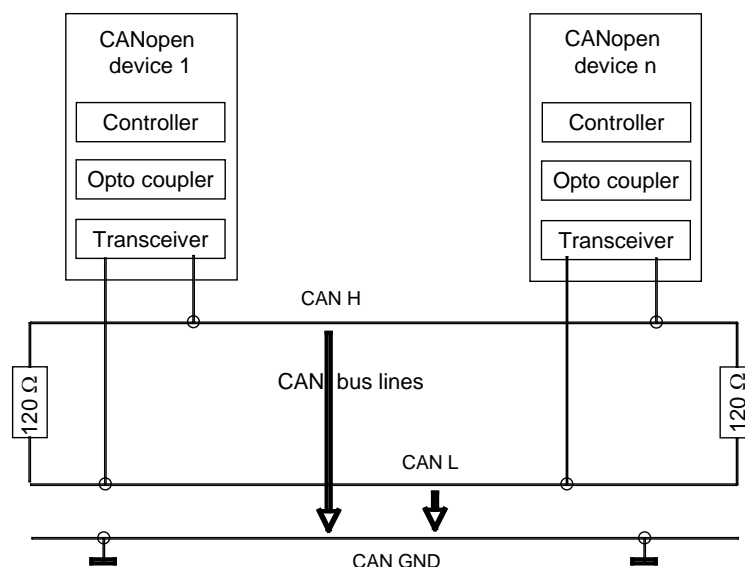


Figure 2 – Network topology of CANopen-based consist network

4.4 Addressing

Any End Device that is connected to a CANopen-based Consist Network requires a unique CANopen Node-ID in the range from 01_h to 7F_h. The CANopen Node-ID is either adjusted via hardware switches or via software. In case the CANopen Node-ID is adjusted via the CANopen communication interface of the device, the CANopen Layer Setting Services shall

be used. The CANopen Node-ID shall not be adjusted via the CANopen object dictionary of the related device.

The definition of the Layer Setting Services (LSS) is not in the scope of this standard

NOTE LSS are specified in the document CiA 305.

4.5 Data classes

To be able to exchange meaningful data across the CANopen-based Consist Network, it is necessary that the format of this data and its meaning is known by the producer and the consumer(s). This specification models this by the concept of data types.

The encoding rules define the representation of values of data types and the transfer syntax for the representations. Values are represented as bit sequences. Bit sequences are transferred in sequences of octets (bytes). For numerical data types the encoding is little endian style.

Applications often require data types beyond the basic data types. Using the compound data type mechanism, it is possible to extend the list of available data types. Some general extended data types are defined as “Visible String” or “Time of Day”. Compound data types are a means to implement user defined “DEFTYPES” in the terminology of the EN 50325-4 and not “DEFSTRUCTS”.

The data types and encoding rules provided in EN 50325-4 shall apply.

5 Physical layer

5.1 Content

This clause provides definitions for the physical layer of a CANopen-based consist network.

5.2 Cabling

The main trunk circuit shall be at least a single twisted pair of nominal characteristic impedance as defined in ISO 11898-2. In addition a CAN_Ground line as clear reference level for CAN_H and CAN_L voltage level shall be applied.

5.3 Connector

It is recommended that devices connected to a CANopen-based consist network support either the 9-pin D-sub connector or the 5-pin micro style (M12) connector. The connectors are illustrated in Figure 3 and Figure 4. The 9-pin D-sub connector shall support a pinning as defined in Table 1. The 5-pin micro style connector (M12) shall support a pinning as defined in Table 2. As the CANopen devices shall be galvanically isolated, the optional V+ line shall only be used for supply of CAN transceiver and optocoupler, in case no additional power supply is used.

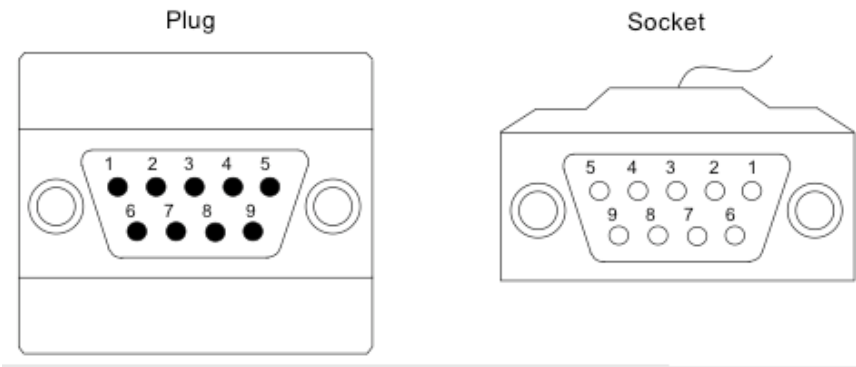


Figure 3 – 9-pin D-sub connector

Table 1 – Pinning for 9-pin D-sub connector

Pin	Signal	Description
1	-	Reserved
2	CAN_L	CAN_L bus line (dominant low)
3	CAN_GND	CAN ground
4	-	Reserved
5	(CAN_SHLD)	Optional CAN shield
6	(GND)	Optional ground
7	CAN_H	CAN_H bus line (dominant high)
8	-	Reserved
9	(CAN_V+)	Optional CAN external positive supply (dedicated for supply of transceiver and optocouplers for galvanically isolated bus nodes)

NOTE It is highly recommended not to connect CAN_GND and GND.

In case shielding is required, it is recommended to connect a shield via the metal housing of the used connector. Optionally the CAN_SHLD pin can be used.

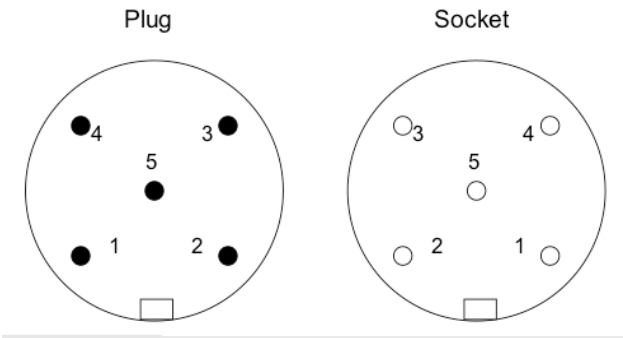


Figure 4 – 5-pin micro style connector

Table 2 – Pinning for 5-pin micro style connector

Pin	Signal	Description
1	(CAN_SHLD)	Optional CAN shield
2	(CAN_V+)	Optional CAN external positive supply (dedicated for supply of transceiver and optocouplers, if galvanic isolation of the bus node applies)
3	CAN_GND	Ground / 0V / V-
4	CAN_H	CAN_H bus line (dominant high)
5	CAN_L	CAN_L bus line (dominant low)

5.4 Physical medium attachment

The physical medium for a device connected to CANopen-based consist networks shall be a differentially driven two-wire bus line with common return according to high-speed transmission specification in ISO 11898-2.

Using the high-speed transceiver according to ISO 11898-2 the maximum rating for V_{CAN_H} and V_{CAN_L} shall be +16 V. Galvanic isolation between CANopen devices is optional. It is recommended to use a CAN transceiver that is capable of sustaining misconnection of any of the wires of the connector including the optional V+ voltages of up to 30 V.

5.5 Physical signaling

The bit encoding/decoding and synchronization shall meet the requirements defined in ISO 11898-1.

The bit timing shall meet the requirements defined in ISO 11898-1 and the definitions as given in Table 3. The recommended location of the sample point is as close as possible to 87,5 % of the related bit time.

Table 3 – Bit timing

Bit rate	Nominal bit time t_b μs	Valid range for location of sample point %
1 Mbit/s	1	75 to 90
800 kbit/s	1,25	75 to 90
500 kbit/s	2	85 to 90
250 kbit/s	4	85 to 90
125 kbit/s	8	85 to 90
50 kbit/s	20	85 to 90
20 kbit/s	50	85 to 90
10 kbit/s	100	85 to 90

Devices connected to a CANopen-based consist network shall support a bit rate of 125 kbit/s. Optionally further bit rates as given in Table 3 may be supported.

6 Data Link layer

6.1 Content

This clause provides definitions for the data link layer of a CANopen-based consist network.

6.2 CANopen data link layer

The described CANopen-based consist network shall be based on a data link layer and its sub-layers according to ISO 11898-1.

This specification is based on the CAN base frame format with 11-bit CAN-Identifier. Therefore it is not required to support the CAN extended frame format with 29-bit CAN-Identifier.

NOTE As certain applications require the usage of the CAN extended frame format, the network can be operated in this mode as well, in case all connected CANopen devices support this format.

7 CANopen application layer

7.1 Content

This clause provides definitions with regard to the upper layers of the ISO OSI reference model for devices connected to a CANopen-based consist network.

7.2 Reference model

Devices connected to a CANopen-based consist network compliant to this specification shall use the reference model, as defined in EN 50325-4.

7.3 Field device model

Devices connected to a CANopen-based consist network shall comply with the field device model as given in Figure 5.

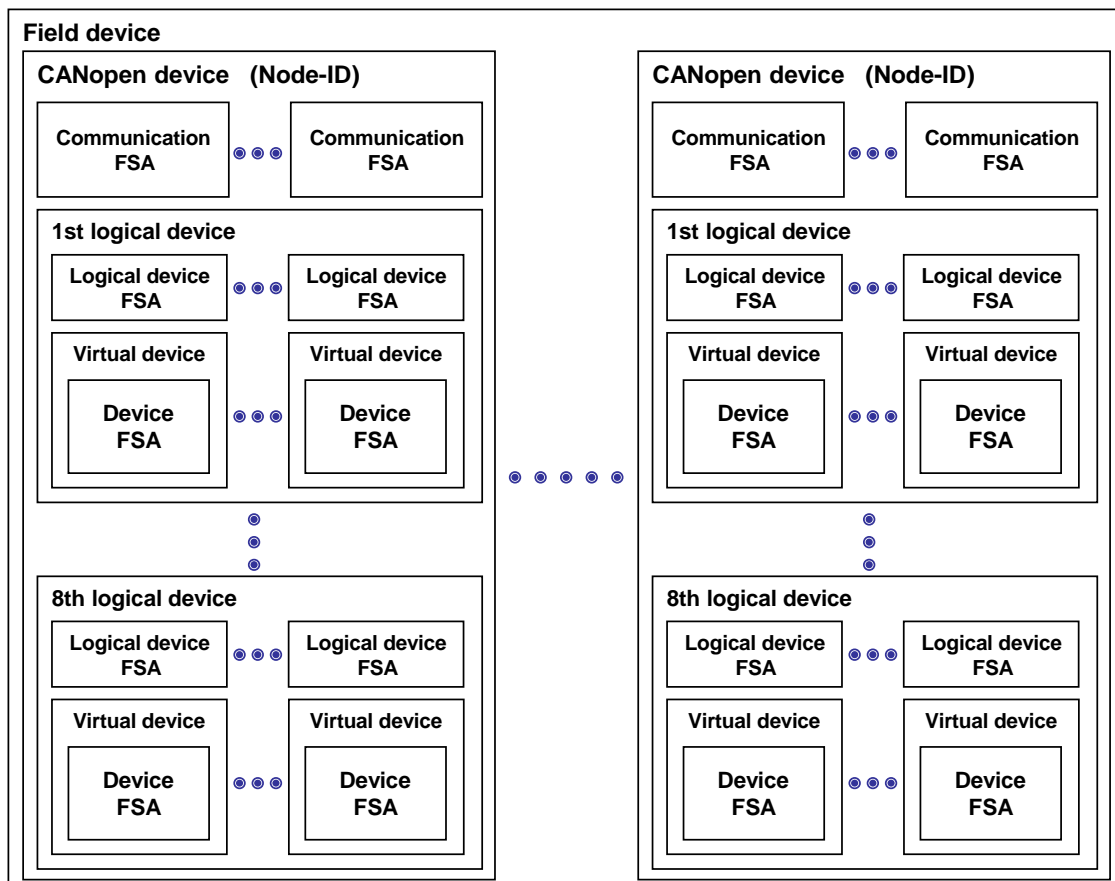


Figure 5 – Field device model

The field device as illustrated in Figure 5 shall provide at least one CANopen device. Each CANopen device within the field device shall provide at least one associated network interface comprising the data link layer protocol (see Clause 6) and the physical layer (see Clause 5), one CANopen Node-ID, and at least one communication FSA. The first communication FSA contains the NMT slave state machine as defined in EN 50325-4. Additional communication FSAs contain an emergency state machine (see EN 50325-4) and others. A CANopen device shall support at least one and may support up to eight logical devices. Each logical device may contain a number of virtual devices and optionally a logical device FSA. A virtual device contains a virtual device FSA and is not distributed to several logical devices. The minimum field device is shown in Figure 6.

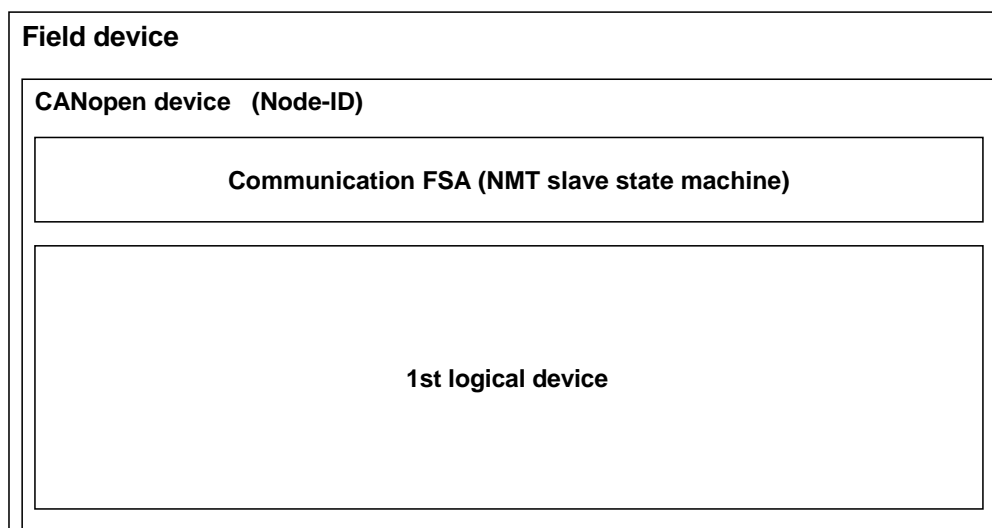


Figure 6 – Minimum field device

A CANopen device is structured as shown in Figure 7:

- **Communication** – This function unit provides the communication objects and the appropriate functionality to transport data items via the underlying network structure.
- **Object dictionary** – The object dictionary is a collection of all the data items which have an influence on the behavior of the application objects, the communication objects and the state machine used on this device.
- **Application** – The application comprises the functionality of the device with respect to the interaction with the process environment.

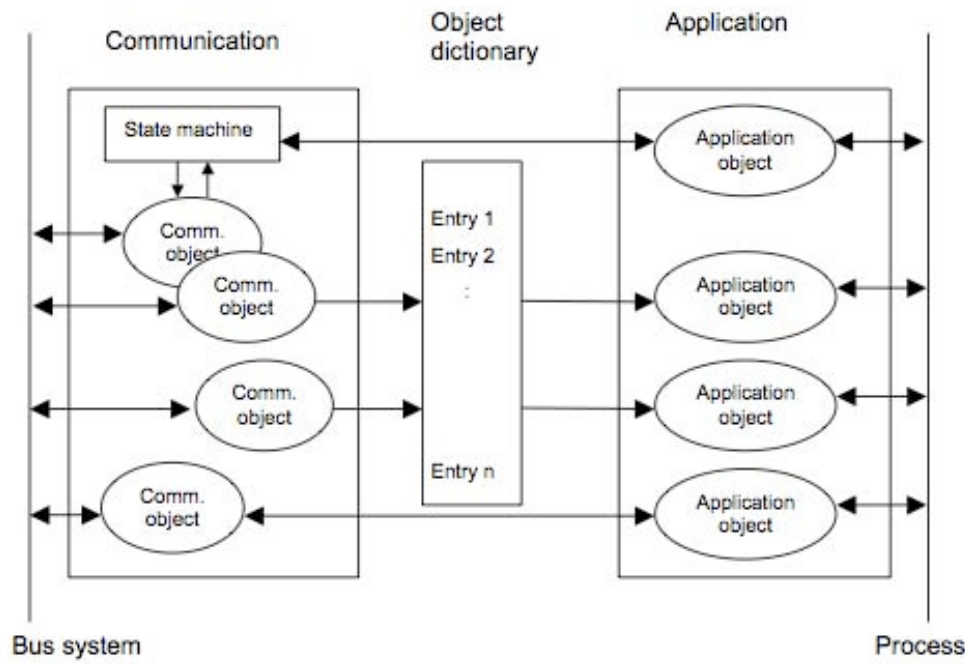


Figure 7 – CANopen device structure

7.4 CANopen communication objects

Devices connected to a CANopen-based consist network shall support all mandatory communication objects of EN 50325-4 as well as the CANopen network management state machine as defined in EN 50325-4.

In order to reduce configuration effort for simple CANopen networks a CAN-ID allocation scheme is defined. These CAN-IDs shall be available in the NMT state Pre-operational directly after the NMT state Initialization (if no modifications have been stored). The objects SYNC, TIME, EMCY write and PDO may be deleted and re-created with new CAN-IDs by means of dynamic distribution. A CANopen device shall provide the corresponding CAN-IDs only for the supported communication objects. The CAN-ID allocation scheme is defined in EN 50325-4.

7.5 CANopen object dictionary

The CANopen object dictionary, as defined in EN 50325-4, contains a maximum of 65536 objects that are addressed through a 16-bit index and up to 256 sub-indices per object, which are addressed through an 8-bit sub-index.

The static data types at indices from 0001_h to 001F_h contain type definitions for standard data types like BOOLEAN, INTEGER, UNSIGNED, floating point, string, etc.

Complex data types at indices from 0020_h to 003F_h are pre-defined structures that are composed of standard data types and are common to all CANopen devices.

Manufacturer-specific complex data types at indices from 0040_h to 005F_h are structures composed of standard data types but are specific to a particular CANopen device.

CANopen device profiles may define additional data types specific to their device type. The static data types and the complex data types defined by the CANopen device profile are listed at indices from 0060_h to 025F_h.

A CANopen device optionally provides the structure of the supported complex data types (indices from 0020_h to 005F_h and from 0060_h to 025F_h) at read access to the corresponding index. Sub-index 00_h then provides the highest sub-index supported at this index, and the following sub-indices contain the data type encoded as UNSIGNED16 according to EN 50325-4.

The communication profile area at indices from 1000_h to 1FFF_h contains the communication specific parameters. These objects are common to all CANopen devices.

The standardized profile area at indices from 6000_h to 9FFF_h contains all data objects common to a class of CANopen devices that may be read or written via the network. The CANopen device profiles use objects from 6000_h to 9FFF_h to describe parameters and functionality.

The object dictionary concept caters for optional features, which means a manufacturer may not provide certain extended functionality on his CANopen devices but if he wishes to do so he shall do it in a pre-defined fashion. Space is left in the object dictionary at indices from 2000_h to 5FFF_h for manufacturer-specific functionality.

The network variables at indices from A000_h to AFFF_h contain input variables and output variables, which are part of a programmable CANopen device.

The system variables at indices from B000_h to BFFF_h contain input variables and output variables, which are part of an underlying CANopen network in a hierarchical sense.

The general CANopen object dictionary structure is illustrated in Table 4.

Table 4 – CANopen object dictionary structure

Index	Object
0000 _h	not used
0001 _h – 001F _h	Static data types
0020 _h – 003F _h	Complex data types
0040 _h – 005F _h	Manufacturer-specific complex data types
0060 _h – 025F _h	Device profile specific data types
0260 _h – 03FF _h	reserved
0400 _h – 0FFF _h	reserved
1000 _h – 1FFF _h	Communication profile area
2000 _h – 5FFF _h	Manufacturer-specific profile area
6000 _h – 67FF _h	Standardized profile area 1 st logical device
6800 _h – 6FFF _h	Standardized profile area 2 nd logical device
7000 _h – 77FF _h	Standardized profile area 3 rd logical device
7800 _h – 7FFF _h	Standardized profile area 4 th logical device
8000 _h – 87FF _h	Standardized profile area 5 th logical device
8800 _h – 8FFF _h	Standardized profile area 6 th logical device
9000 _h – 97FF _h	Standardized profile area 7 th logical device
9800 _h – 9FFF _h	Standardized profile area 8 th logical device
A000 _h – AFFF _h	Standardized network variable area
B000 _h – BFFF _h	Standardized system variable area
C000 _h – FFFF _h	reserved

7.6 **Predefined CANopen communication objects**

7.6.1 **Content**

This clause provides the basic CANopen communication capability for CANopen devices that participate in a CANopen-based consist network according to this specification.

7.6.2 **Object 1000_h: Device type**

This object provides the type of device and its functionality. The value 0000_h for the device profile number indicates a logical device that does not follow a standardized profile. In this case the additional information is 0000_h (if no further logical device is implemented) or FFFF_h (if a further logical device is implemented).

For multiple logical device modules the additional information parameter is FFFF_h and the device profile number referenced by object 1000_h is the profile of the first logical device in the object dictionary. All other profiles of a multiple logical device module identify their profiles at objects 67FF_h + x * 800_h with x = internal number of the logical device (from 1 to 8) minus 1. These objects describe the device type of the preceding logical device, having the very same value definition as object 1000_h.

Figure 8 illustrates the structure of the object. The value definition, the object description and the entry description are specified in EN 50325-4.

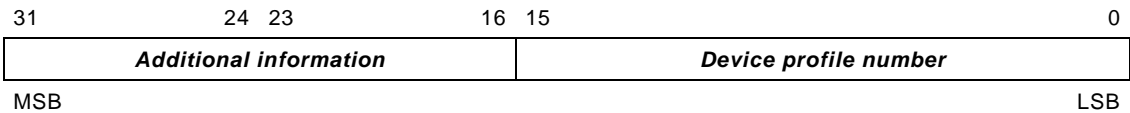


Figure 8 – Structure of the device type object

NOTE Devices connected to CANopen-based consist networks may follow the CANopen application profile for train vehicle control networks CiA 421. This CANopen application profile defines application data, based on UIC 556, that is exchanged in a CANopen-based consist network.

7.6.3 **Object 1001_h: Error register**

This object provides the error information. The CANopen device maps internal errors into this object. This error information is published as part of the emergency message. The value definition, object description and entry description are specified in EN 50325-4.

7.6.4 **Object 1014_h: COB-ID emergency object**

This object shall be implemented. It is specified in EN 50325-4. The CAN-ID, which is part of this object, shall not be changed.

7.6.5 **Object 1017_h: Heartbeat producer**

All CANopen devices that are connected to a CANopen-based consist network shall implement this object. It is specified in EN 50325-4.

The devices shall support heartbeat message transmissions from 100 ms to 1 000 ms.

7.6.6 **Object 1018_h: Identity object**

This object provides general information about the device as specified in EN 50325-4.

7.6.7 **Object 1029_h: Error behavior**

This object specifies to which state the device is set, in case a communication error or a device-internal error is detected. It is specified in EN 50325-4.

7.6.8 Object 67FF_h: Device type

This object shall describe the first logical device in a multiple device module according to EN 50325-4.

7.6.9 Service data objects (SDOs)

Any CANopen device supports the first SDO server channel. CANopen devices connected to a CANopen-based consist network may support additional SDO server or client channels.

In case an additional SDO channel is supported, the related SDO parameter set is supported in the CANopen object dictionary as defined in EN 50325-4.

There are no further SDO channels pre-defined by this specification.

7.6.10 Process data objects (PDOs)

CANopen devices operated in a CANopen-based consist network may support up to 512 PDOs in transmit as well as receive direction.

In case a CANopen device supports a PDO, the related PDO communication parameter and mapping entries are supported in the CANopen object dictionary as defined in EN 50325-4.

There are no PDOs pre-defined.

8 Application data

8.1 Content

This clause provides the representation of application data that is communicated between a CANopen-based consist network and a Train Backbone via a Gateway.

NOTE 1 The application data is described in application profile of TCN.

In general application data is managed in the CANopen object dictionary index ranges 2000_h to 5FFF_h and 6000_h to 9FFF_h. A manufacturer specific device behavior is controlled via the index range 2000_h to 5FFF_h. A standardized CANopen device behavior is controlled via objects in the index range 6000_h to 9FFF_h.

NOTE 2 Object dictionary entries in the index range 6000h to 9FFFh are specified in CANopen device and application profiles and available at CAN in Automation.

8.2 CANopen application data representation

As the application data for TCN is not defined yet, the representation of the TCN application data in the standardized object dictionary index range is not defined.

8.3 Recommended representation principle of application data

8.3.1 Content

In this clause a representation principle of application data is provided for the representation of application data in CANopen-based consist networks. Therefore it is possible to map the process to CANopen process data objects (PDOs) for process data transfer.

8.3.2 Application data for door control

This clause provides the recommended representation principle by means of door control information, which is communicated between a CANopen-based consist network and a Train

Backbone via a Gateway. The represented process data is therefore available in the CANopen-based consist network and can be communicated via the CANopen communication objects.

The access type is given for the interface of the gateway device to the CANopen-based consist network. Other devices such as e.g. door controllers or door units, which are connected to a CANopen-based consist network manage the data as defined in this clause, at the given CANopen object dictionary index. The access type of the object may be changed to the appropriate access type as defined in EN 50325-4.

NOTE The illustrated application data is derived from the document UIC 556 and is given in the representation of the CANopen application profile CiA 421 train vehicle control network.

8.3.3 Consumed door control application objects

8.3.3.1 Object 6007_h: External door status word export

This object shall indicate the status of the external doors of the local rail vehicle. Via the gateway to the Train Backbone, this information is available in the TCN. Figure 9 specifies the object structure and Table 5 specifies the value definition. Table 6 specifies the object description and Table 7 specifies the entry description.

NOTE The object shall correspond to R3-telegram octet 20, which is defined in UIC 556.

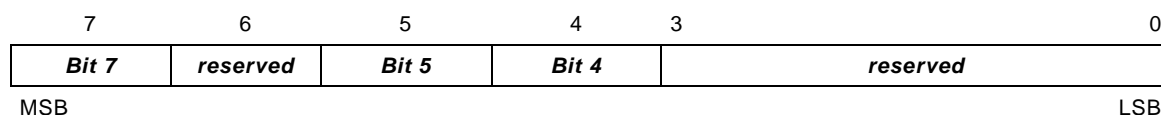


Figure 9 – Object structure

Table 5 – Value definition

Bit	Value	Value definition
Bit4	0	At least one left door is open
	1	All left doors locked
Bit 5	0	At least one right door is open
	1	All right doors locked
Bit 7	0	Side selective door blocking is not in operation
	1	Side selective door blocking is in operation
Reserved		Reserved (shall be ignored)

Table 6 – Object description

Attribute	Value
INDEX	6007 _h
Name	External doors status word export
Object code	Variable
Data type	UNSIGNED8
Category	Optional

Table 7 – Entry description

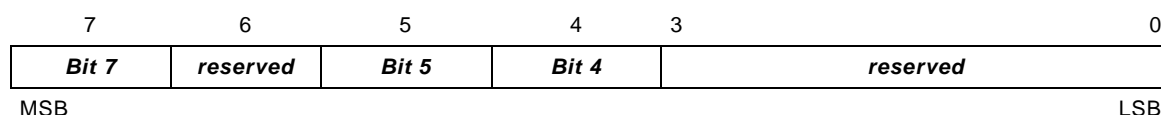
Attribute	Value
Sub-index	00 _h
Access	rw
PDO mapping	Optional
Value range	See value definition
Default value	Manufacturer specific

8.3.4 Produced door control application objects

8.3.4.1 Object 6006_h: External door status word import

This object shall provide the status of the external doors of other rail vehicles. Each sub-index shall provide the door status of that vehicle, which corresponds with the UIC vehicle number. Therefore the external door status of other vehicles is available within the CANopen-based consist network. The overall door status of all vehicles is provided in sub-index 21_h. Figure 10 specifies the object structure and Table 8 specifies the value definition. Table 9 specifies the object description and Table 10 specifies the entry description.

NOTE The object shall correspond to R3-telegram octet 20, which is defined in UIC 556.

**Figure 10 – Object structure****Table 8 – Value definition**

Bit	Value	Value definition
Bit4	0	At least one left door is open
	1	All left doors locked
Bit 5	0	At least one right door is open
	1	All right doors locked
Bit 7	0	Side selective door blocking is not in operation
	1	Side selective door blocking is in operation
Reserved		Reserved (shall be ignored)

Table 9 – Object description

Attribute	Value
INDEX	6006 _h
Name	External doors status word import
Object code	ARRAY
Data type	UNSIGNED8
Category	Optional

Table 10 – Entry description

Attribute	Value
Sub-index	00 _h
Description	Highest subindex supported
Entry category	Mandatory
Access	ro
PDO mapping	No
Value range	01 _h to 21 _h
Default value	No
Sub-index	01 _h
Description	Door status UIC vehicle 1
Entry category	Optional
Access	ro
PDO mapping	Optional
Value range	see Table 46
Default value	No
to	
Sub-index	20 _h
Description	Door status UIC vehicle 32
Entry category	Optional
Access	ro
PDO mapping	No
Value range	Optional
Default value	No
Sub-index	21 _h
Description	Door status train
Entry category	Optional
Access	ro
PDO mapping	No
Value range	Optional
Default value	No

8.3.4.2 Object 6001_h: External door command

This object shall provide the first control word for the external doors. Figure 11 specifies the object structure and Table 11 specifies the value definition. Table 12 specifies the object description and Table 13 specifies the entry description.

NOTE The object shall correspond to R3-telegram octet 20, which is defined in UIC 556.

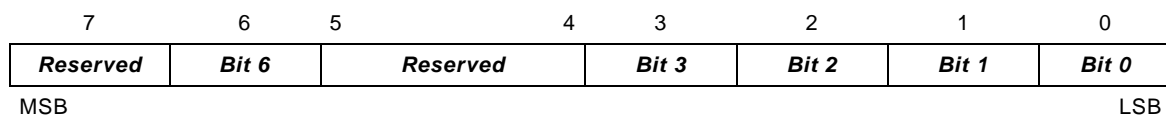


Figure 11 – Object structure

Table 11 – Value definition

Bit	Value	Value definition
Bit 0	0	Inactive
	1	All doors close
Bit 1	0	Inactive
	1	Interrupt door closing
Bit 2	0	Release all left doors
	1	Lock all left doors
Bit 3	0	Release all right doors
	1	Lock all right doors
Bit 6	0	Not extend footstep
	1	Extend footstep
Reserved		Reserved (shall be ignored)

Table 12 – Object description

Attribute	Value
INDEX	6001 _h
Name	External door command 1
Object code	Variable
Data type	UNSIGNED8
Category	Optional

Table 13 – Entry description

Attribute	Value
Sub-index	00 _h
Access	ro
PDO mapping	Optional
Value range	See value definition
Default value	No

9 CANopen network management

9.1 Content

This clause provides the network management for CANopen-based consist networks. The definition of the network management includes the definition of the network startup behavior as well as definitions that are related to networks that operate without NMT master, networks with one CANopen device capable of the NMT master mode. These definitions are intended to be an add-on to the CANopen application layer and communication profile provided in EN 50325-4.

CANopen consist networks with more than one CANopen device capable of the NMT master mode (NMT flying master) are not in the scope of this specification. However such networks are not excluded.

NOTE With regard to an increased availability, within CANopen-based consist networks several CANopen devices with NMT master functionality may reside. As at a certain time, only one active NMT master is allowed, mechanisms are required, that enable overtaking of the NMT master functionality from one CANopen device to another. These services are part of the CANopen flying master functionality that is described in CiA 302.

9.2 CANopen NMT slave functionality

Devices operated in a CANopen-based consist network shall provide CANopen NMT-slave functionality and shall implement all mandatory communication services and protocols as well as all mandatory objects as specified in EN 50325-4. The minimum communication objects are defined in 7.6.

Optionally further communication objects may be supported. In addition the CANopen manager functionality may be supported.

9.3 CANopen manager functionality

9.3.1 General

This subclause provides the definition of the CANopen manager functionality for CANopen-based consist networks.

Besides the application process several different additional functionalities exist in a CANopen network. These functionalities are referred to by different terms. This subclause is intended to clarify these terms.

Within a distributed system the application process is divided into several parts running on different CANopen devices. From the applications point of view usually one CANopen device is responsible for the control of the system. This CANopen device is called *application master*.

From the network's point of view there are several additional functionalities, which not directly deal with the application but provide application-supporting functions. These additional functionalities are based on a master/slave, client/server or producer/consumer relationship.

As it is common to combine several of the additional functionalities in one CANopen device the term CANopen Manager is introduced.

A CANopen device is referred to as CANopen manager, if it provides the NMT master functionality and at least one of the functionalities SDO manager or Configuration manager.

9.3.1.1 NMT master

The network management (NMT) provides services for controlling the network behavior of CANopen devices as defined in EN 50325-4. All CANopen devices that are participating in a CANopen consist network, referred to as NMT slaves, are controlled by services provided by an NMT master. Usually the NMT master application is also part of the application master.

The device, in which the NMT master functionality resides is a fully CANopen device. In addition to the NMT master functionality it supports all functions and objects, which are indicated as mandatory in EN 50325-4.

9.3.1.2 Flying master

The flying master mechanism provides services for a hot stand-by NMT master within a CANopen network. The flying master is an optional functionality within a CANopen device. The CANopen device implementing the flying master implements the NMT master functionality. The definition of the responsibilities, functionalities, and services of the flying master is not in the scope of this document.

NOTE Definitions for the flying master functionality are provided in CiA 302.

9.3.1.3 SDO manager

The SDO manager is an optional functionality, responsible for handling of the dynamic establishment of SDO connections. If an SDO manager is present in a CANopen network it resides together with the NMT master on the same CANopen device.

NOTE Definitions for the SDO manager functionality are provided in CiA 302.

9.3.1.4 Configuration manager

The Configuration Manager is an optional functionality, which provides mechanisms for configuration of CANopen devices in a CANopen network during boot-up. The mechanisms are called Configuration Management (CMT). If the Configuration Manager is present in a CANopen network it resides together with the NMT master on the same CANopen device.

NOTE Definitions for the Configuration manager functionality are provided in CiA 302.

9.3.1.5 SYNC producer

The SYNC producer is an optional functionality, which is responsible for transmitting the SYNC object. It may reside on any CANopen device in the CANopen-based consist network. The relevant CANopen object dictionary entries are defined in EN 50325-4.

9.3.1.6 TIME producer

The TIME producer is an optional functionality, which is responsible for transmitting the TIME STAMP object. It may reside on any CANopen device in the CANopen-based consist network. The time stamp object is defined in EN 50325-4.

9.3.1.7 LSS master

The layer setting services (LSS) provide services for configuring layer 2 (bit timing) and NMT (CANopen Node-ID) via CAN. The LSS master configures LSS slaves by executing LSS services. The definition of the layer setting services is not in the scope of this document.

NOTE Definitions for the layer setting services are provided in CiA 305.

9.3.2 Object dictionary usage

Several objects relating to the configuration and validation of CANopen devices are of object type ARRAY. The sub-index of the entries of such an ARRAY corresponds to the CANopen Node-ID of the CANopen device. Those objects may have less than 127 entries. In that case all of these objects shall have the very same set of supported entries. Sub-index 00_h shall provide the highest sub-index supported at this index. Related object description in Clause 9 refer to this as *SupportedNodeID* condition.

NOTE For a CANopen manager to be universally usable it is recommended to support the full sub-index range of 01_h to $7F_h$.

9.3.3 Redundant networks

Redundant networks are required for high-availability applications. Redundant networks consist of at least two CAN lines. This allows communication between CANopen devices, in case of a single failure within the physical interconnection between the CANopen devices. The definition of the responsibilities, functionalities, and services for redundant networks is not in the scope of this document.

NOTE Definitions for setting up redundant CANopen networks are provided in CiA 302.

9.4 CANopen NMT start-up

9.4.1 NMT startup

CANopen managers shall behave according to the NMT slave state machine as defined in EN 50325-4. Before transition from NMT state *Pre-operational* to NMT state *Operational* of the CANopen manager, all assigned NMT slaves shall be booted. The main flow chart for the procedure is defined in Figure 12 and Figure 13. The simplest flow chart for the startup is defined in Figure 14.

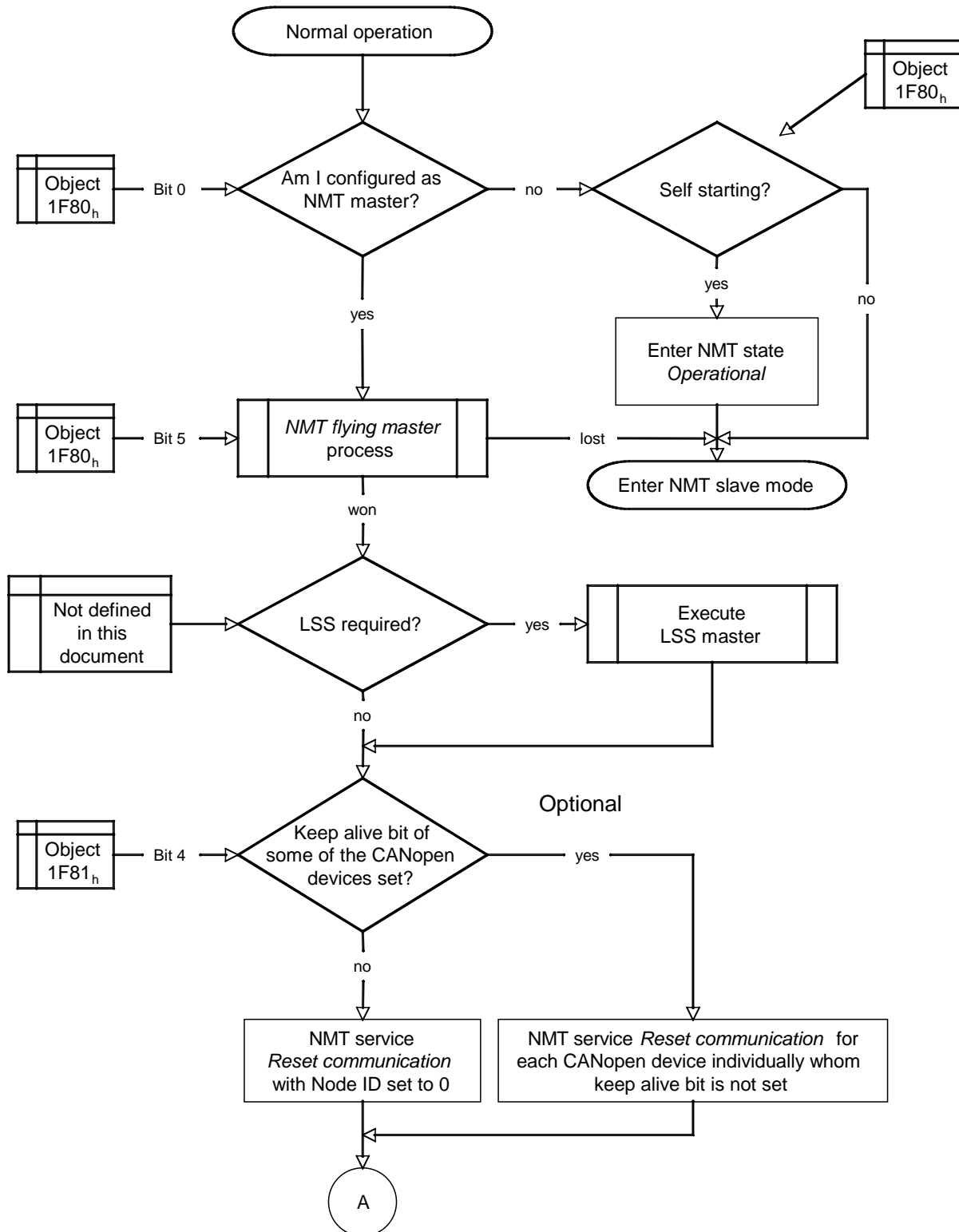


Figure 12 – NMT startup, part 1

The process NMT startup as shown in Figure 12 consists of the following basic steps:

NOTE 1 The process execute LSS master is not in the scope of this standard.

- a) Bit 0 of object 1F80_h (see clause 9.8.7) is used to decide whether this CANopen device shall be the NMT master. If the CANopen device is NMT master the process shall continue. If the CANopen device is configured as self-starting device, it shall enter the NMT state Operational automatically. In case the CANopen device is not the NMT master the process shall end.
- b) Bit 5 of object 1F80_h (see clause 9.8.7) is used to decide whether this CANopen device shall participate in the service *NMT flying master negotiation*. If the CANopen device shall participate in the service *NMT flying master negotiation* and the CANopen device lost the service *NMT flying master negotiation* the CANopen device shall not become NMT master. The definition of the NMT flying master negotiation is not in the scope of this document.

NOTE 2 The description of the NMT flying master negotiation is provided in CiA 302.

- c) If LSS is required to set the CANopen Node-ID and bit rate of other CANopen devices within the network the NMT master shall execute the LSS master services. The LSS master services may be executed at any time. The precise definition of the LSS master services is not in the scope of this document.

NOTE 3 The description of the layer setting services is provided in CiA 305.

- d) Bit 4 of all entries of object 1F81_h (see clause 9.8.8) is used to decide whether the NMT master shall perform the NMT service *Reset communication* with CANopen Node-ID set to 0 or if the NMT service *Reset communication* shall be performed for each CANopen device in the network individually. If at least for one entry of object 1F81_h bit 4 is set to 1_b and the corresponding CANopen device is in NMT state *Operational* the NMT master shall not issue the NMT service *Reset communication* with CANopen Node-ID set to 0. In this case each CANopen device shall be reset individually. This shall also include all CANopen Node-IDs that are not part of the slave list 1F81_h.

NOTE 4 This will force potentially existing CANopen devices that are not configured in the slave list to transmit a boot-up message. By this the CANopen Manager will recognize unconfigured CANopen devices via the boot-up handler (Figure 23). The NMT service *Reset communication* shall not apply to the NMT master itself.

If after step d) the NMT service *Reset communication* is necessary, for any reason, then the NMT service *Reset communication* shall be executed with the CANopen Node-ID of that NMT slave. This shall apply independently from the configuration for that NMT slave in object 1F81_h. The process NMT startup resumes in Figure 13.

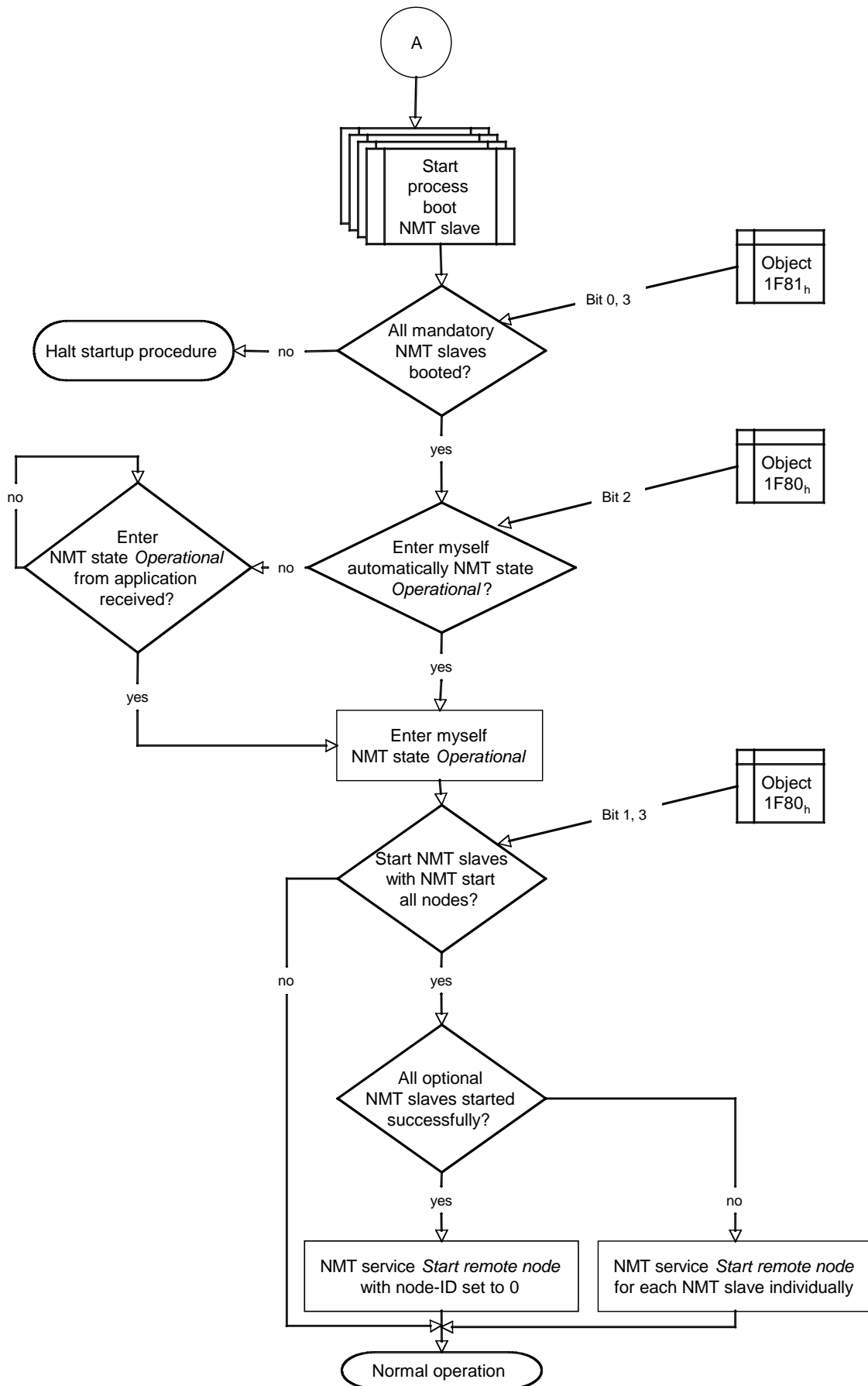


Figure 13 – NMT startup, part 2

- e) The NMT master shall start the process *start process boot NMT slave* for all NMT slaves as shown in Figure 15. For all NMT slaves that are marked as mandatory (bit 0 and bit 3 of object 1F81_h; see 9.8.8) the process *start process boot NMT slave* shall terminate successfully.
- f) If an error is detected during process *boot slave* for the NMT slaves that are marked as mandatory the process *NMT startup* shall be stopped.
- g) Bit 2 of object 1F80_h (see 9.8.7) is used to decide whether the NMT master shall enter the NMT state *Operational* automatically by itself or shall wait until it is requested by the application running on the very same CANopen device.
- h) Under the following conditions
- Bit 3 of object 1F80_h is set to 0_b,
 - Bit 1 of object 1F80_h is set to 1_b,
 - and all NMT slaves listed in 1F81_h booted successfully
- the NMT service *Start remote node* shall be performed with CANopen Node-ID set to 0.
- Under the following conditions
- Bit 3 of object 1F80_h is set to 0_b,
 - Bit 1 of object 1F80_h is set to 1_b,
 - and not all NMT slaves listed in 1F81_h booted successfully
- the NMT service *Start remote node* shall be performed for each NMT slave individually.
- i) The process *NMT startup* ended successfully and the NMT master shall proceed with normal operation.

Occurrence and detection of NMT slaves not listed in 1F81_h falls into the responsibility of the application.

9.4.2 NMT startup simple

Since nearly all objects and features are optional it is possible to implement a basic NMT master, which may make sense for some applications. Removing all optional parts in the definitions above results in the process *NMT startup simple* as shown in Figure 14. Even a simple NMT master supports all mandatory objects, which are specified in EN 50325-4 and which are defined in this standard.

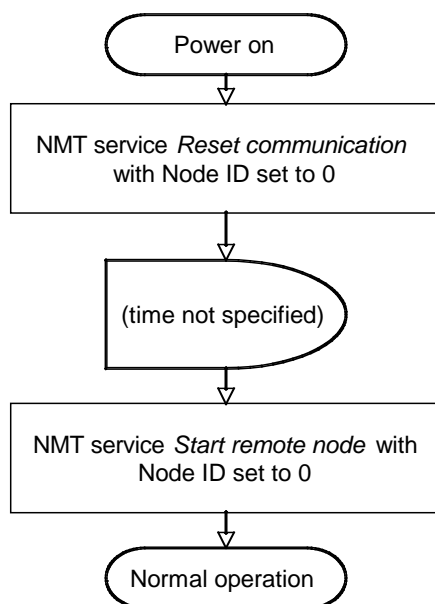


Figure 14 – NMT startup simple

9.4.3 Start process boot NMT slave

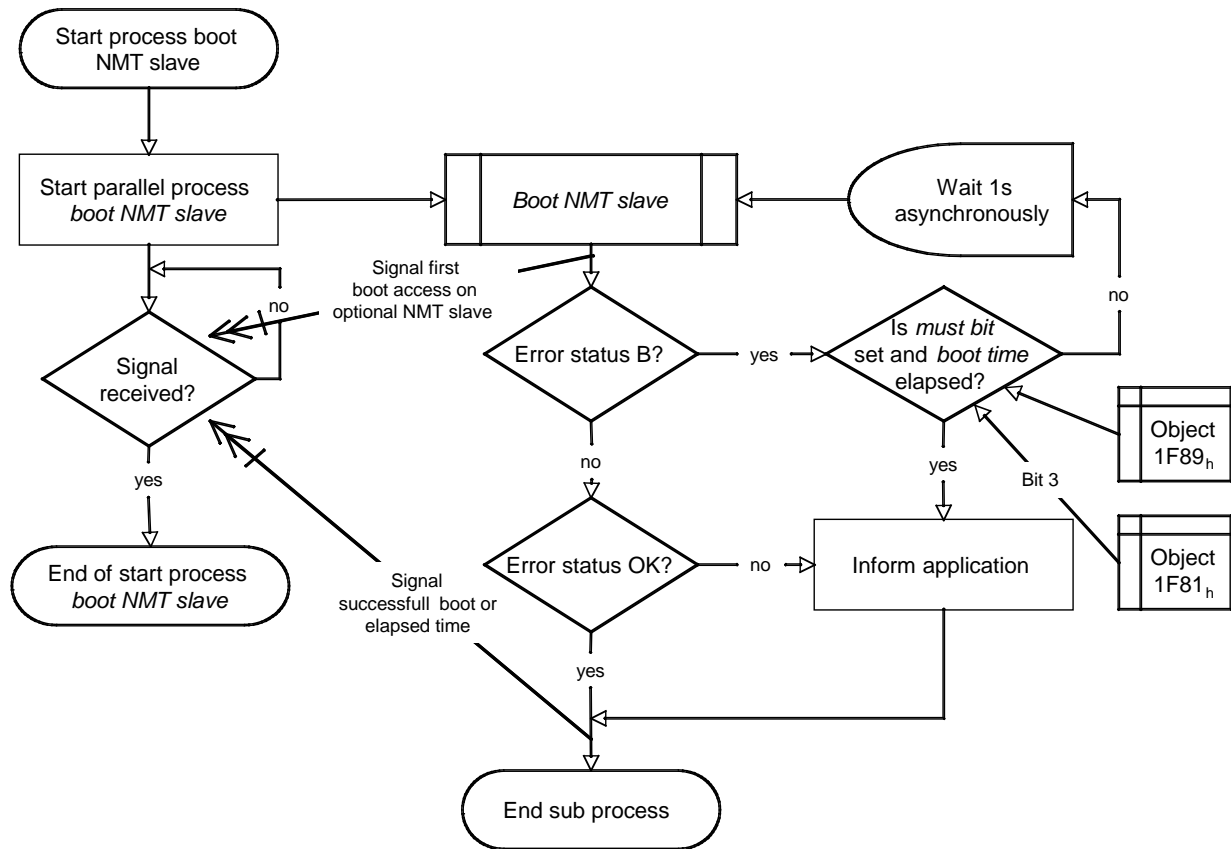


Figure 15 – Start process boot NMT slave

The process *start process boot NMT slave* as shown in Figure 15 shall include the following steps:

- Start parallel process *boot NMT slave*.
- Mandatory NMT slaves: wait for completion of the process *boot NMT slave*.
Optional NMT slaves: wait for signal that the process *Boot NMT slave* was performed.

The parallel process shall

- Perform the process boot NMT slave (see 9.5).
- Create signal for every try of the process *boot NMT slave*.

If the process *boot NMT slave* returned with status OK the process shall terminate. This process shall run endlessly for any optional NMT slave until the process *Boot NMT slave* finishes with status OK.

NOTE The recommended cycle time is 1 s for bit rate higher than 125 kbit/s.

If the process *boot NMT slave* returned with error status B for mandatory NMT slaves and the elapsed time is greater than the configured value of object 1F89_h then the application shall be informed and this sub-process shall end.

The sub-process of the process *start process boot NMT slave* shall run asynchronously to other processes.

9.5 Boot NMT slave

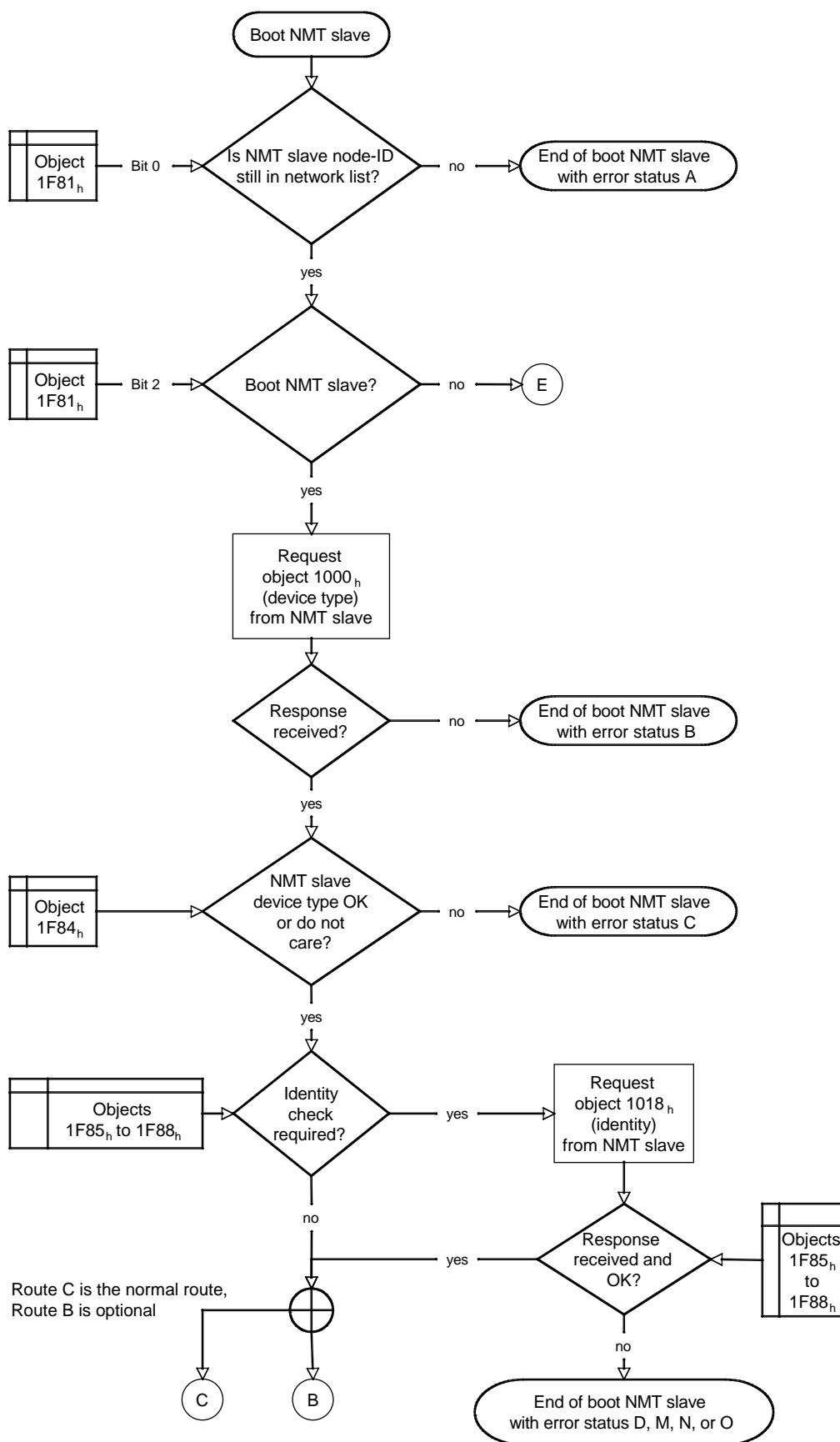


Figure 16 – Boot NMT slave, part 1

The process *boot NMT slave* as shown in Figure 16 consists of the following steps:

- a) Bit 0 of object 1F81_h is used to decide upon whether the NMT slave shall be processed or if the process shall terminate with error status.
- b) Bit 2 of object 1F81_h is used to decide upon whether the NMT slave shall be configured and started.
- c) Upload object 1000_h from the NMT slave. In case no response is received the process shall terminate with error status.
- d) In case the value of object 1F84_h for the NMT slave is unequal 0 the value shall be checked against object 1000_h. In case both values are different the process shall terminate with error status.
- e) In case the values of objects 1F85_h to 1F88_h are unequal 0, the particular object values shall be checked against their corresponding values of the object 1018_h from the NMT slave. In case one of the values of object 1F85_h to 1F88_h is different to the corresponding values of the object 1018_h from the NMT slave the process shall terminate with error status.

The process *boot NMT slave* may continue with an optional part 2 (see Figure 17), which introduces two optional features:

- keeping alive CANopen devices initially in NMT state *Operational*, and
- managing application software versions including automatic update of the application software.

If the keep alive bit of a CANopen device is set the NMT master shall not issue the NMT services *Reset node* and *Reset communication* for this CANopen device.

NOTE 1 Such situations may occur, in case the NMT master encounters a failure with a subsequent re-start, e.g. power fail.

Software version control may be used in systems where the version of the application software running on the CANopen device is checked for correctness. The process check and update software version may be used to automatically download the most current version of the application software. CANopen devices that detect an error in their application software, for example by calculating a checksum during startup, may force a download of the most current application software by responding with wrong version information.

Both or only one of the two features may be implemented.

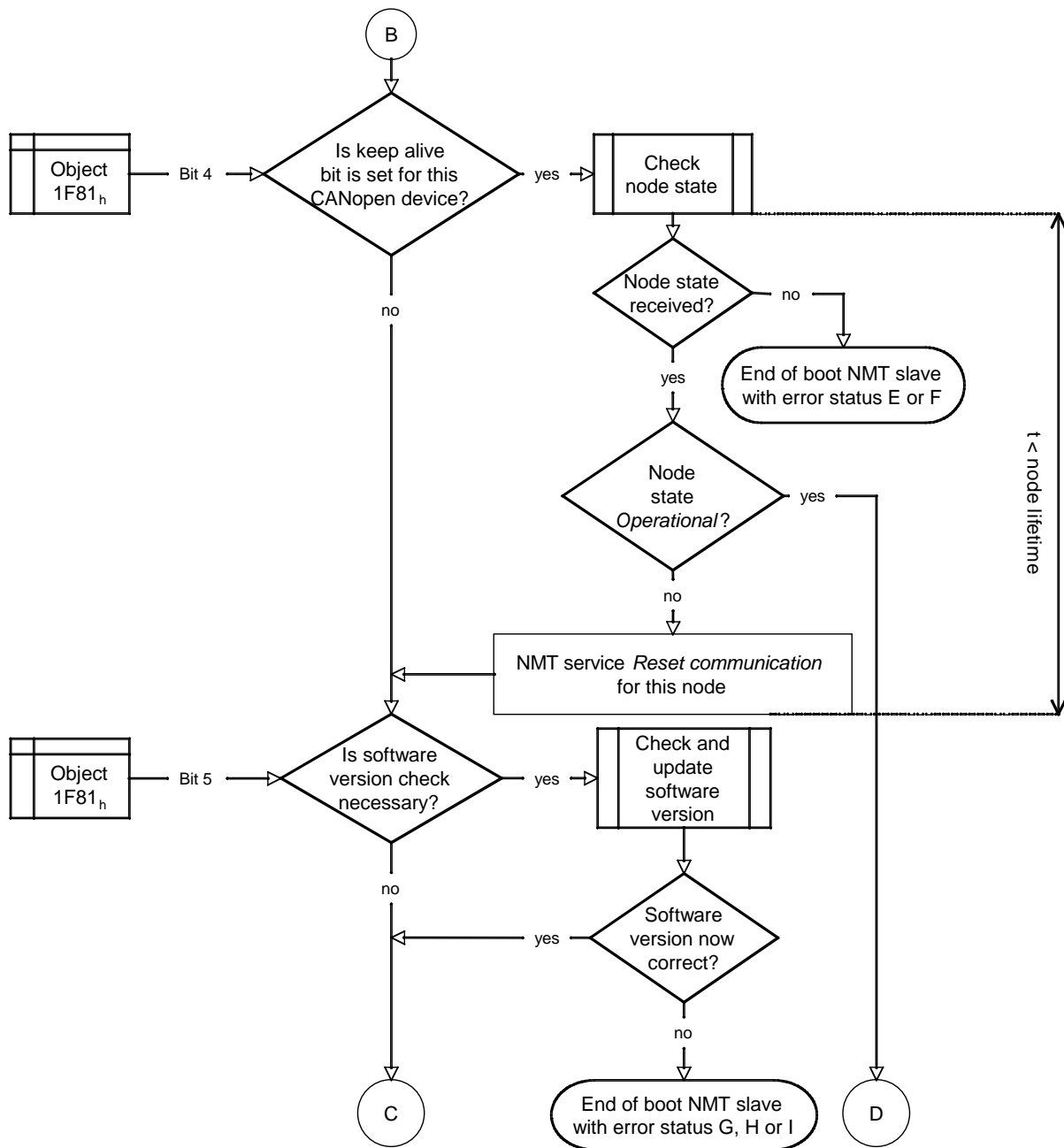


Figure 17 – Boot NMT slave, part 2

- f) Bit 4 of object 1F81_h is used to decide whether keep alive is requested.

In case keep alive is requested the NMT master shall request the current NMT state of the NMT slave for which the *boot NMT slave* is performed. In case no current NMT state is received the process shall finish with error status. In case the current NMT state is *Operational* the process shall continue with D. In case the current NMT state is not the NMT state *Operational* the NMT master shall perform the NMT service *Reset communication* for that NMT slave.

The process check NMT state is defined in 9.5.2.

NOTE 2 In case the NMT slave supports Node and Life guarding only the NMT master takes care that the time from requesting the current NMT state from the NMT slave to issuing the NMT service *Reset communication* is less than the life time as set in the NMT slave.

- g) Bit 5 of object 1F81_h is used to decide whether the application software verification shall be performed.

In case application software verification is requested, the process check and update software version shall be performed.

In case the software version stays incorrect the process Boot NMT slave shall end with error status.

The process *boot NMT slave* continues in part 3 (see Figure 18) and is mandatory.

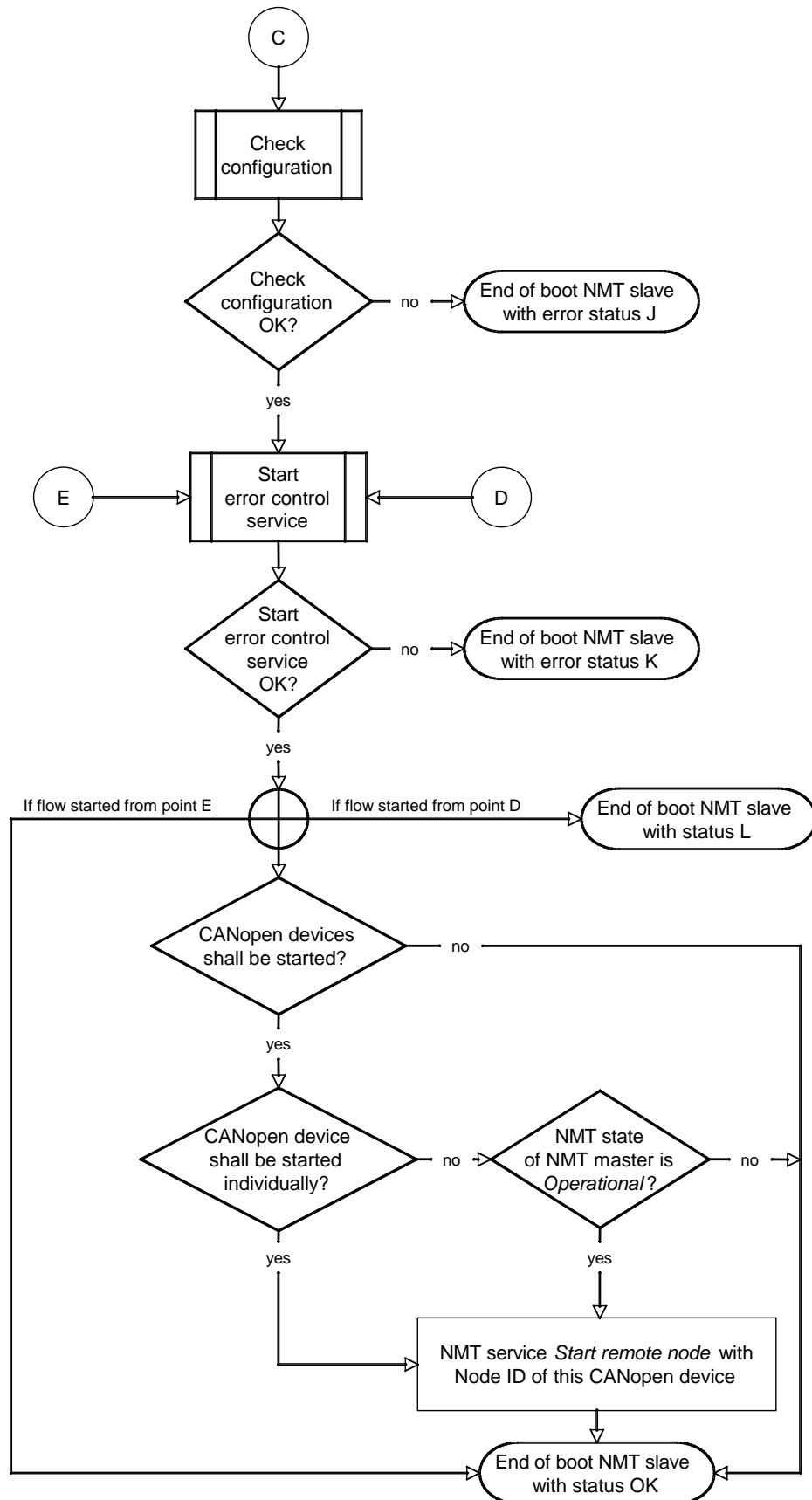


Figure 18 – Boot NMT slave, part 3

- h) The process check configuration (see 9.5.1) shall be performed.
 - i) In case the process check configuration finished unsuccessfully the process *boot NMT slave* shall end with an error status.
 - j) The process *start error control* (see 9.6.1) shall be performed.
 - k) In case the process *start error control* finished unsuccessfully the process *boot NMT slave* shall end with an error status.
 - l) In case the NMT slave is in NMT state *Operational* the process *boot NMT slave* shall finish successfully.
 - m) Bit 3 of object 1F80_h is used to decide whether the NMT master shall execute the NMT service *Start remote node*.
 - n) In case the NMT master shall not execute the NMT service *Start remote node* the process *boot NMT slave* shall end successfully with status.
 - o) Bit 1 of object 1F80_h is used to decide upon whether the NMT master shall execute the NMT service *Start remote node* with CANopen Node-ID set to 0 or for each NMT slave in the network individually.
 - p) In case the NMT master shall execute the NMT service *Start remote node* with CANopen Node-ID set to 0 and the NMT master is not in NMT state *Operational* the process *boot NMT slave* shall finish successfully with status OK.
- NOTE 3 If the NMT master is in NMT state *Operational* it is regarded as an indication that the initial NMT startup has been completed prior to the startup of this NMT slave.
- q) In case the NMT master executes the NMT service *Start remote node* for each NMT slave in the network individually or in case the NMT master executes the NMT service *Start remote node* with CANopen Node-ID set to 0 and the NMT master is in NMT state *Operational* the NMT master shall execute the NMT service *Start remote node* with CANopen Node-ID set to the appropriate value.
 - r) The process *boot NMT slave* shall finish successfully with status.

9.5.1 Check configuration

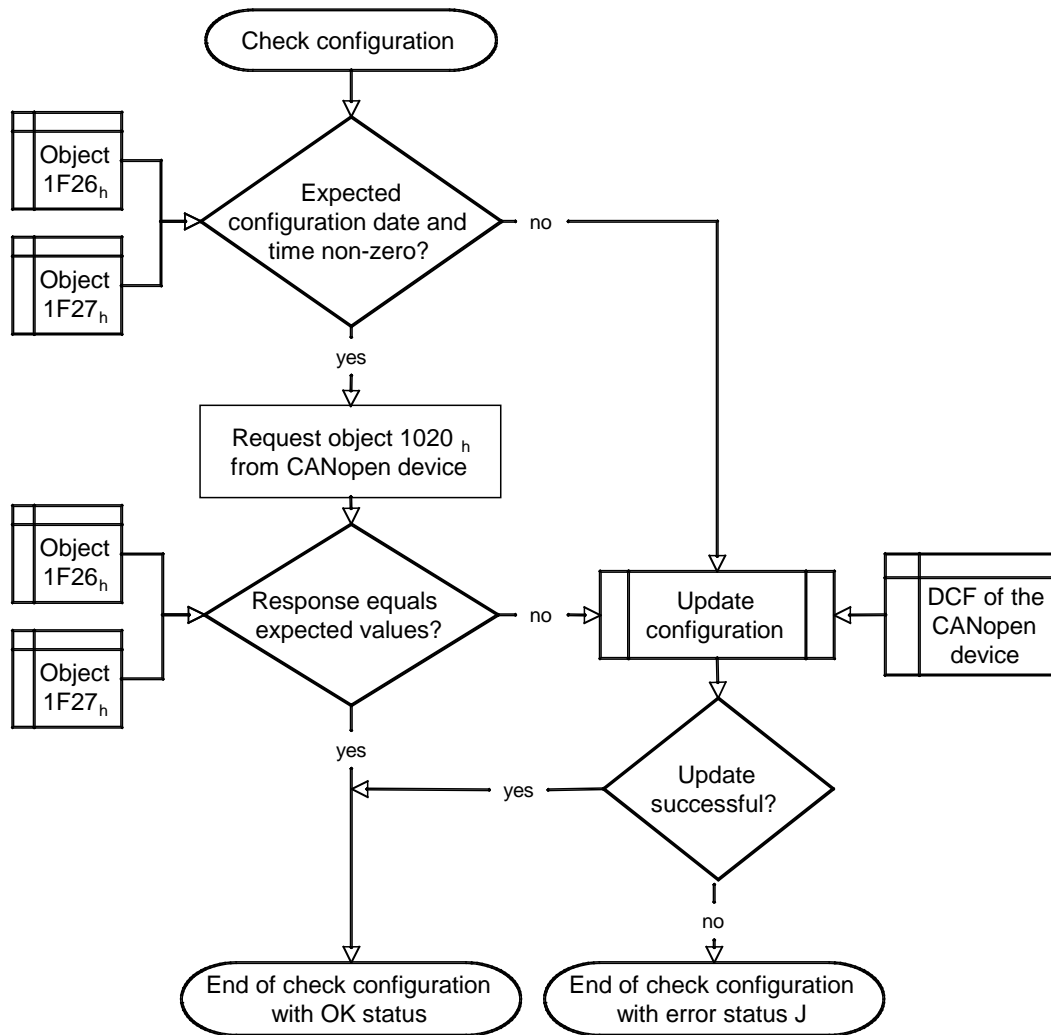


Figure 19 – Check configuration

The process *check configuration* as shown in Figure 19 consists of the following steps:

- The corresponding entries (sub-index shall equal CANopen Node-ID) of object 1F26_h and object 1F27_h are used to decide whether the CANopen devices configuration is verified. If the values are equal to 0 the process shall go to step d). If the values are not equal to 0 the process shall go to step b).
- Object 1020_h shall be requested from the CANopen device.
- The values of the corresponding entries (sub-index shall equal CANopen Node-ID) of object 1F26_h and object 1F27_h shall be compared to the values of the received object 1020_h. If the values are equivalent the process shall finish with status OK. If the values are not equivalent the process shall go to step d).
- The configuration shall be updated on the CANopen device.
Depending on the settings of 1F81_h and 1F8A_h the configured restore operation shall be executed. Only after the restore operation the NMT service reset communication or NMT service reset node for the corresponding NMT slave shall be issued.
The configuration may be derived from any kind of DCF provided for that CANopen device. The DCF may be read from a local file system, object 1F20_h, or object 1F22_h on the configuration manager.
- If the download is successful then the process shall finish with status OK. If the download is not successful then the process shall finish with error status.

The process *check configuration* may be skipped if the keep-alive is enabled for that CANopen device and the CANopen device is in the NMT state *Operational*. If this occurs the process boot NMT slave shall finish with error status (L — "NMT slave was initially operational").

9.5.2 Check NMT state

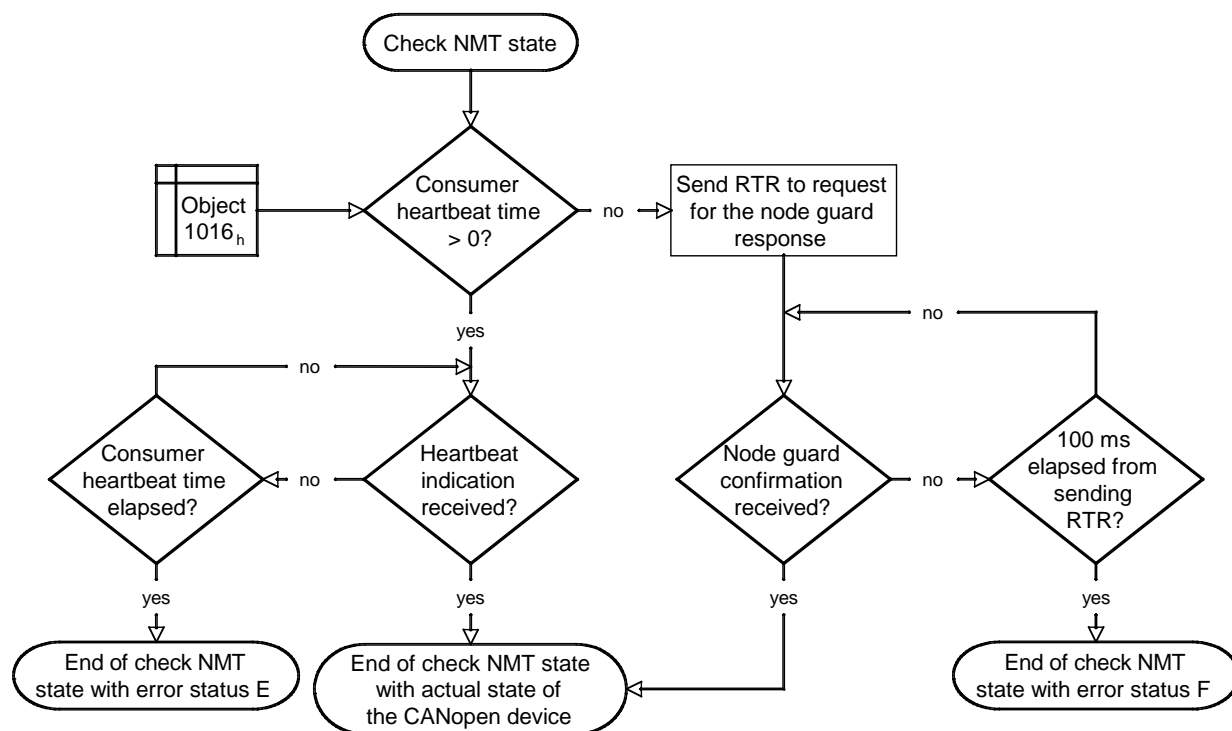


Figure 20 – Check NMT state¹

The process *check NMT state* as shown in Figure 20 consists of the following steps:

- Object 1016_h (see EN 50325-4) shall be used to decide whether the CANopen device is set up for heartbeat.
- In case the CANopen device is set up for heartbeat the NMT master shall check if it received a heartbeat indication in time (heartbeat consumer time is not elapsed). If the heartbeat indication is not received in time the process shall end with an error status. If the heartbeat indication is received the process shall end with the actual NMT state of the CANopen device that is checked.
- In case the CANopen device that is checked is not set up for heartbeat it is assumed that the CANopen device is set up for CANopen device guarding. In this case the NMT master shall request the actual NMT state of the CANopen device that is checked, the NMT service CANopen device guarding shall be used. In case no confirmation is received and more than 100 ms are elapsed the process shall end with an error status. In case a confirmation is received the process shall end with the actual NMT state of the CANopen device that is checked.

9.5.3 NMT flying master start up

The definition of the flying NMT master is not in the scope of this standard.

NOTE Definitions for the flying NMT master functionality are provided in CiA 302.

¹ Required, if bit 4 of object 1F81_h is set (keep alive supported).

9.5.4 Error status

The errors as shown in Table 14 shall be signaled within the NMT master during startup.

Table 14 – Error status

Error status	Description
A	The CANopen device is not listed in object 1F81 _h .
B	No response received for upload request of object 1000 _h .
C	Value of object 1000 _h from CANopen device is different to value in object 1F84 _h (Device type).
D	Value of object 1018 _h sub-index 01 _h from CANopen device is different to value in object 1F85 _h (Vendor ID).
E	Heartbeat event. No heartbeat message received from CANopen device.
F	Node guarding event. No confirmation for guarding request received from CANopen device.
G	Object 1F53 _h or object 1F54 _h not configured for that CANopen device.
H	Values of object 1F52 _h from that CANopen device are different to value of object 1F53 _h or value of object 1F54 _h and bit 6 of object 1F81 _h is not set.
I	Values of object 1F52 _h from that CANopen device are different to value of object 1F53 _h or value of object 1F54 _h and download failed.
J	Configuration download failed.
K	Heartbeat event during start error control service. No heartbeat message received from CANopen device during start error control service.
L	NMT slave was initially operational. (CANopen manager may resume operation with other CANopen devices)
M	Value of object 1018 _h sub-index 02 _h from CANopen device is different to value in object 1F86 _h (Product code).
N	Value of object 1018 _h sub-index 03 _h from CANopen device is different to value in object 1F87 _h (Revision number).
O	Value of object 1018 _h sub-index 04 _h from CANopen device is different to value in object 1F88 _h (Serial number).

9.6 Error control

9.6.1 Start error control

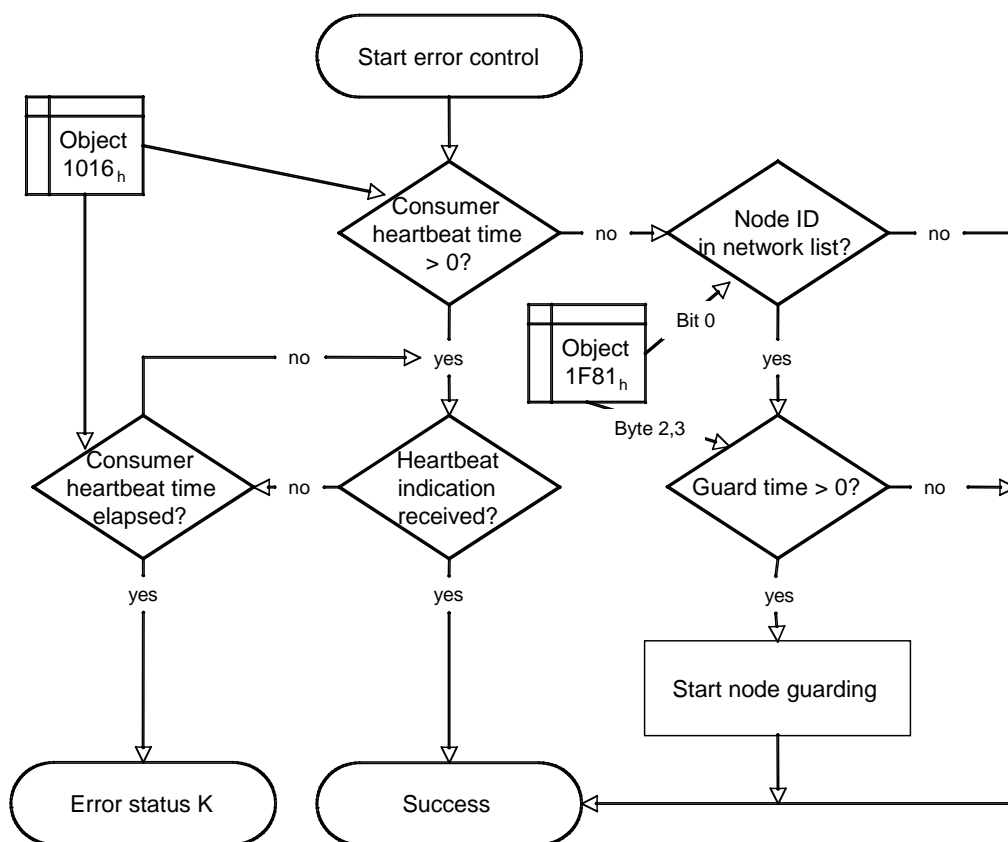


Figure 21 – Start error control

The process *start error control* as shown in Figure 21 consists of the following steps:

- Object 1016_h (see EN 50325-4) shall be checked to decide whether the CANopen device is set up for heartbeat.
- In case the CANopen device is set up for heartbeat and no heartbeat indication is received in time the process shall end with an error status. The timer for timeout checking shall be started immediately with the process itself.
- In case the CANopen device is set up for heartbeat and a heartbeat indication is received in time the process shall end successfully.
- In case the CANopen device is not set up for heartbeat bit 0 of object 1F81_h is used to decide whether the CANopen device shall be guarded.
- In case the CANopen device is not in the network list the process shall end successfully.
- In case the CANopen device is in the network list the value of byte 2 and 3 of object 1F81_h is used to decide whether the CANopen device shall be guarded.
- In case the value is greater than 0 guarding shall be started for that CANopen device and the process shall end successfully.
- In case the value equals 0 the process shall end successfully.

9.6.2 Error handler

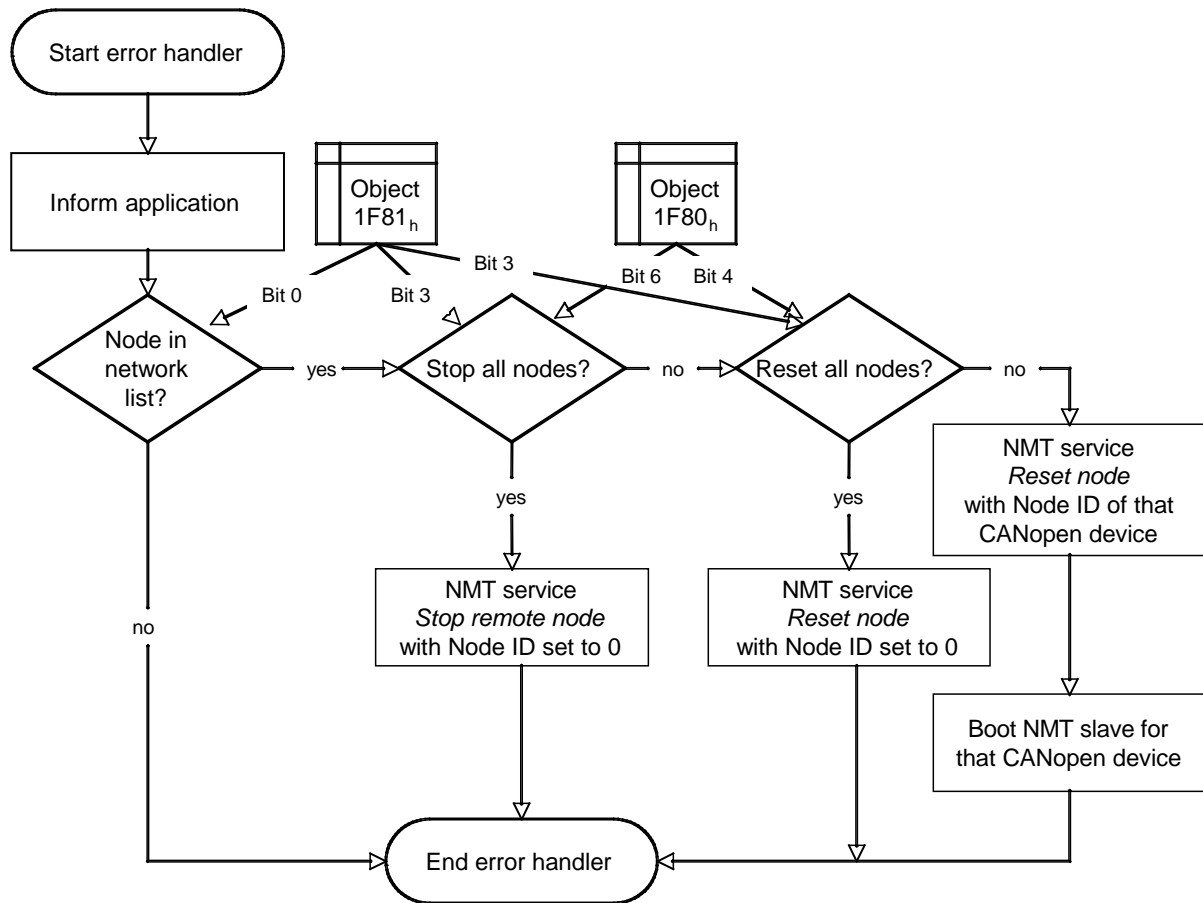


Figure 22 – Error handler

The process *error handler* as defined in Figure 22 shall be initiated any time an NMT error event occurs.

9.6.3 Bootup handler

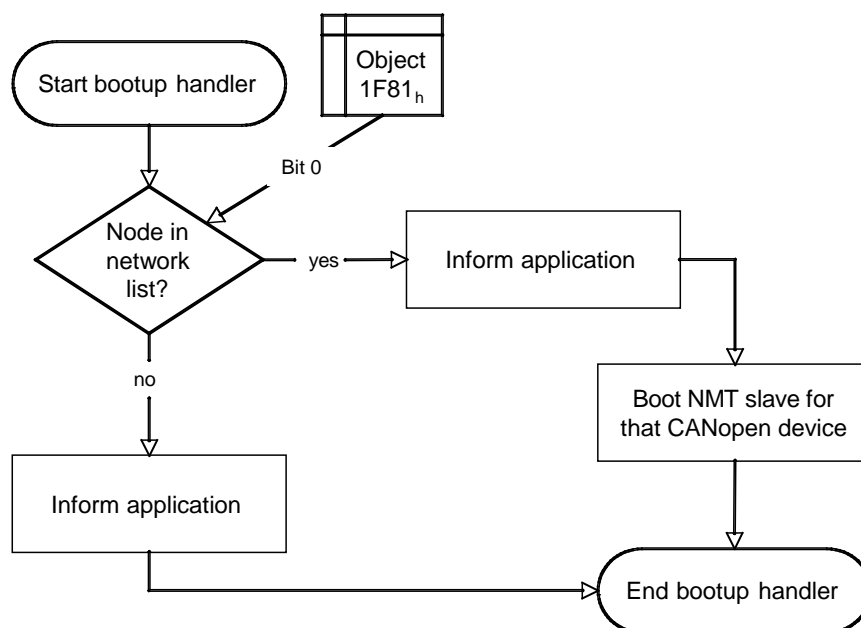


Figure 23 – Bootup handler

The process *bootup handler* as defined in Figure 23 shall be initiated any time a bootup event occurs (see EN 50325-4).

9.7 Additional NMT master services and protocols

The definition of additional NMT master services and protocols is not in the scope of this specification.

NOTE Additional NMT master services and protocols such as NMT master negotiation or NMT master detection services are provided in CiA 302.

9.8 Object dictionary entries

9.8.1 Object 1020_h: Verify configuration

This object shall indicate the downloaded configuration date and time. If a CANopen device supports the saving of parameters in non-volatile memory, a network configuration tool or a CANopen manager uses this object to verify the configuration after a CANopen device reset and to check if a reconfiguration is necessary. The configuration tool stores the date and time in that object and stores the same values in the DCF. Now the configuration tool lets the CANopen device save its configuration by writing to index 1010_h sub-index 01_h the signature "save". After a reset the CANopen device shall restore the last configuration and the signature automatically or by request. If any other command changes boot-up configuration values, the CANopen device shall reset the object Verify Configuration to 0.

The Configuration Manager compares signature and configuration with the value from the DCF and decides if a reconfiguration is necessary or not.

NOTE The usage of this object allows a significant speed-up of the boot-up process. If it is used, the system integrator considers that an user changes a configuration value and afterwards activate the command store configuration 1010_h without changing the value of 1020_h. So the system integrator ensures a 100 % consequent usage of this feature.

Sub-index 01_h (configuration date) shall contain the number of days since January 1, 1984. Sub-index 02_h (configuration time) shall be the number of ms after midnight. Table 15 and Table 16 provide the object description and the entry description.

Table 15 – Object description

Attribute	Value
Index	1020 _h
Name	Verify configuration
Object code	ARRAY
Data type	UNSIGNED32
Category	Optional

Table 16 – Entry description

Attribute	Value
Sub-index	00 _h
Description	Highest sub-index supported
Entry category	Mandatory
Access	const
PDO mapping	No
Value range	02 _h
Default value	02 _h
Sub-index	01 _h
Description	Configuration date
Entry category	Mandatory
Access	rw
PDO mapping	No
Value range	UNSIGNED32
Default value	Manufacturer-specific
to	
Sub-index	02 _h
Description	Configuration time
Entry category	Mandatory
Access	rw
PDO mapping	No
Value range	UNSIGNED32
Default value	Manufacturer-specific

9.8.2 Object 102A_h: NMT inhibit time

This object shall indicate the configured inhibit time between two subsequent NMT messages. The outstanding NMT services shall be queued and shall be issued in order of their occurrence respecting the configured inhibit time. Table 17 and Table 18 define the object description and the entry description.

The value shall be given in multiples of 100 µs. The value 0 shall disable the inhibit time.

Table 17 – Object description

Attribute	Value
Index	102A _h
Name	NMT inhibit time
Object code	VAR
Data type	UNSIGNED16
Category	Mandatory

Table 18 – Entry description

Attribute	Value
Sub-index	00 _h
Access	rw
PDO mapping	No
Value range	UNSIGNED16
Default value	0

9.8.3 Object 1F20_h: Store DCF

This object shall be used to save the current configuration for a specific CANopen device in the network. Table 19 defines the object description and Table 20 defines the entry description.

The sub-index of an entry shall correspond to the CANopen Node-ID of the CANopen device in the network. The sub-index of the entry corresponding to its own CANopen Node-ID shall be used for self-configuration. If no DCF is saved previously a read access shall be responded with an SDO abort message (error code: 0800 0024_h or 0800 0000_h). The definition of the DCF format is not in the scope of this specification.

NOTE The format of DCF is provided in CiA 302.

Table 19 – Object description

Attribute	Value
Index	1F20 _h
Name	Store DCF
Object code	ARRAY
Data type	DOMAIN
Category	Optional

Table 20 – Entry description

Attribute	Value
Sub-index	00 _h
Description	Number of entries
Entry category	Mandatory
Access	const
PDO mapping	No
Value range	7F _h
Default value	7F _h
Sub-index	01 _h
Description	CANopen Node-ID 1
Entry category	Mandatory
Access	rw
PDO mapping	No
Value range	<i>See value definition</i>
Default value	Manufacturer-specific
to	
Sub-index	7F _h
Description	CANopen Node-ID 127
Entry category	Mandatory
Access	rw
PDO mapping	No
Value range	<i>See value definition</i>
Default value	Manufacturer-specific

9.8.4 Object 1F22_h: Concise DCF

This object shall be used to save the current configuration for a specific CANopen device in the network. Table 21 defines the object description and Table 22 defines the entry description.

The sub-index of an entry shall correspond to the CANopen Node-ID of the CANopen device in the network. The sub-index of the entry corresponding to its own CANopen Node-ID shall be used for self-configuration. The format of the Device Configuration File (DCF) shall be as defined in Figure 24. If no DCF was saved previously the number of entries shall be 0. If the DCF for a specific entry shall be deleted the number of entries shall be set to 0.

Number (n) of entries		UNSIGNED32
Entry 1	Index	UNSIGNED16
	Sub-index	UNSIGNED8
	Size (m) of data in bytes	UNSIGNED32
	Data byte 1	UNSIGNED8
	Data byte 2	UNSIGNED8

	Data byte m	UNSIGNED8
Entry 2	Index	UNSIGNED16
	Sub-index	UNSIGNED8
	Size (m) of data bytes	UNSIGNED32
	Data byte 1	UNSIGNED8
	Data byte 2	UNSIGNED8

	Data byte m	UNSIGNED8
.....		
Entry n	Index	UNSIGNED16
	Sub-index	UNSIGNED8
	Size (m) of data bytes	UNSIGNED32
	Data byte 1	UNSIGNED8
	Data byte 2	UNSIGNED8

	Data byte m	UNSIGNED8

Figure 24 – Data stream definition of concise DCF

Table 21 – Object description

Attribute	Value
Index	1F22 _h
Name	Concise DCF
Object code	ARRAY
Data type	DOMAIN
Category	Optional

Table 22 – Entry description

Attribute	Value
Sub-index	00 _h
Description	Number of entries
Entry category	Mandatory
Access	const
PDO mapping	No
Value range	7F _h
Default value	7F _h
Sub-index	01 _h
Description	CANopen Node-ID 1
Entry category	Mandatory
Access	rw
PDO mapping	No
Value range	See definition
Default value	Manufacturer-specific
to	
Sub-index	7F _h
Description	CANopen Node-ID 127
Entry category	Mandatory
Access	rw
PDO mapping	No
Value range	See definition
Default value	Manufacturer-specific

9.8.5 Object 1F26_h: Expected configuration date

This object shall be used for verification of the configuration date of the CANopen devices in the network.

The configuration date (see 9.8.1 object 1020_h sub-index 01_h) of the CANopen devices in the network shall be matched against the value of this object in case the value is unequal to 0. In case the value of this object is 0 the CANopen device shall be configured.

The sub-index of an entry shall correspond to the CANopen Node-ID of the CANopen devices in the network.

Table 23 – Object description

Attribute	Value
Index	1F26 _h
Name	Expected configuration date
Object code	ARRAY
Data type	UNSIGNED32
Category	Optional

Table 24 – Entry description

Attribute	Value
Sub-index	00 _h
Description	Number of entries
Entry category	Mandatory
Access	const
PDO mapping	No
Value range	7F _h
Default value	7F _h
Sub-index	01 _h
Description	CANopen Node-ID 1
Entry category	Mandatory
Access	rw
PDO mapping	No
Value range	See definition
Default value	Manufacturer-specific
to	
Sub-index	7F _h
Description	CANopen Node-ID 127
Entry category	Mandatory
Access	rw
PDO mapping	No
Value range	See definition
Default value	Manufacturer-specific

9.8.6 Object 1F27_h: Expected configuration time

This object shall be used for verification of the configuration time of the CANopen devices in the network.

The configuration time of the CANopen devices (object 1020_h sub-index 02_h) in the network shall be matched against the value of this object in case the value is unequal to 0. In case the value of this object is 0 the CANopen device shall be configured.

The sub-index of an entry shall correspond to the CANopen Node-ID of the CANopen devices in the network.

NOTE Object description and entry description of index 1020_h sub-index 02_h are provided in the latest CANopen version; document CiA 301.

Table 25 – Object description

Attribute	Value
Index	1F27 _h
Name	Expected configuration time
Object code	ARRAY
Data type	UNSIGNED32
Category	Optional

Table 26 – Entry description

Attribute	Value
Sub-index	00 _h
Description	Number of entries
Entry category	Mandatory
Access	const
PDO mapping	No
Value range	7F _h
Default value	7F _h
Sub-index	01 _h
Description	CANopen Node-ID 1
Entry category	Mandatory
Access	rw
PDO mapping	No
Value range	See value definition
Default value	Manufacturer-specific
to	
Sub-index	7F _h
Description	CANopen Node-ID 127
Entry category	Mandatory
Access	rw
PDO mapping	No
Value range	See definition
Default value	Manufacturer-specific

9.8.7 Object 1F80_h: NMT startup

This object shall configure the startup behavior of a CANopen device. Internal state transitions shall not change the value of this object. An attempt to change a bit of a functionality that is not supported by the CANopen device shall be responded with an abort message (abort code: 0800 0000_h or 0609 0030_h). Figure 25 and Figure 26 define the bit-oriented structure of the value. Table 27, Table 28, Table 29, Table 30, Table 31, Table 32, and Table 33 define the allowed values. Table 34 defines exceptions for start-up capable devices. Table 35 and Table 36 define the object description and the entry description.

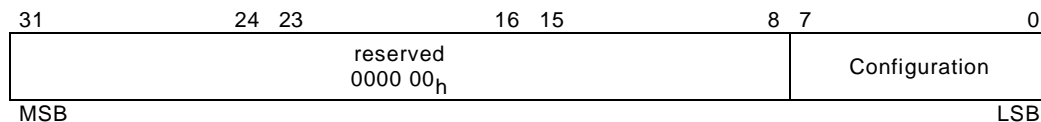


Figure 25 – Object structure

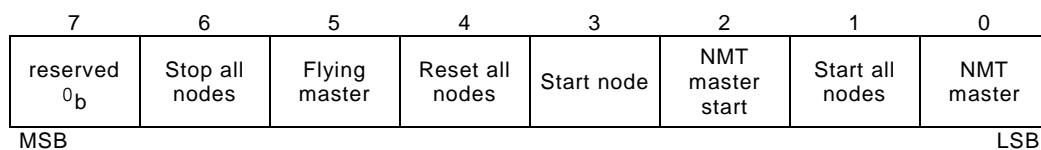


Figure 26 – Bit structure of the configuration value

Table 27 – Value NMT master (bit: 0)

Value	Description
0 _b	CANopen device is not NMT master The entries of the object 1F81 _h shall be ignored. All other bits of object 1F80 _h shall be ignored with the exceptions defined in Table 34.
1 _b	CANopen device is the NMT master

Table 28 – Value Start all nodes (bit: 1)

Value	Description
0 _b	NMT service <i>start remote node</i> for each CANopen Node-ID
1 _b	NMT service <i>start remote node</i> with CANopen Node-ID = 0

Table 29 – Value NMT master start (bit: 2)

Value	Description
0 _b	Shall switch into NMT state <i>Operational</i> in the process NMT startup (see Figure 13)
1 _b	Shall not switch into the NMT state <i>Operational</i> by itself.

Table 30 – Value Start node (bit: 3)

Value	Description
0 _b	The NMT master shall start the NMT slaves.
1 _b	The NMT master shall not start the NMT slaves and the application may start the NMT slaves.

Table 31 – Reset all nodes (bit: 4)

Value	Description
0 _b	In case of error control event of a CANopen device defined as mandatory (see object 1F81 _h) the NMT service <i>reset node</i> with CANopen Node-ID of the CANopen device that caused the error control event shall be executed.
1 _b	In case of error control event of a CANopen device defined as mandatory (see object 1F81 _h) the NMT service <i>reset node</i> with CANopen Node-ID = 0 shall be executed.

Table 32 – Flying master (bit: 5)

Value	Description
0 _b	CANopen device shall not participate the NMT flying master negotiation
1 _b	Reserved (for NMT flying master capable devices)

NOTE The definition of the NMT flying master functionality is not in the scope of this standard. Definitions are provided in CiA 302.

Table 33 – Stop all nodes (bit: 6)

Value	Description
0 _b	In case of error control event of a CANopen device defined as mandatory (see object 1F81 _h) the action as defined by bit 4 shall be executed.
1 _b	In case of error control event of a CANopen device defined as mandatory (see object 1F81 _h) the NMT service <i>Stop remote node</i> with CANopen Node-ID = 0 shall be executed. Bit 4 shall be ignored.

Table 34 – Exceptions for NMT start-up capable devices

Value	Description
00000000 00000000 00000000 00001000 _b	NMT slave that shall enter the NMT state <i>Operational</i> after the NMT state <i>Initialisation</i> autonomously (self starting)
00000000 00000000 00000000 00000010 _b	NMT slave that shall execute the NMT service <i>start remote node</i> with CANopen Node-ID set to 0

Table 35 – Object description

Attribute	Value
Index	1F80 _h
Name	NMT startup
Object code	VAR
Data type	UNSIGNED32
Category	Conditional; Mandatory if the CANopen device is a CANopen manager or a start-up capable CANopen device.

Table 36 – Entry description

Attribute	Value
Sub-index	00 _h
Access	rw
PDO mapping	No
Value range	See value definition
Default value	Manufacturer-specific

9.8.8 Object 1F81_h: NMT slave assignment

This object shall assign CANopen devices to the CANopen manager, the device that shall implement this object. Each sub-index of this object shall correspond to the CANopen Node-ID of the related CANopen device in the network. The sub-index corresponding to its own CANopen Node-ID shall be ignored. An attempt to change a bit of a functionality that is not supported by the CANopen device shall be responded with an abort message (abort code: 0800 0000_h or 0609 0030_h). Figure 27 and Figure 28 define the bit-oriented structure of the value. Table 37, Table 38, Table 39, Table 40, Table 41, Table 42, and Table 43 define the value contents. Table 44 and Table 45 define the object description and the entry description.

The value for the retry factor indicates the number of retries the NMT master shall issue in case of a node guarding event. The value 0 shall disable node guarding for the CANopen device.

The value for the guard time shall indicate the cycle time for the node guarding of the CANopen device. The value shall be indicated in multiples of ms. The value 0 shall disable node guarding for the CANopen device.

NOTE If the heartbeat consumer object is configured to a value unequal to 0, then the heartbeat mechanism will have priority over node guarding.

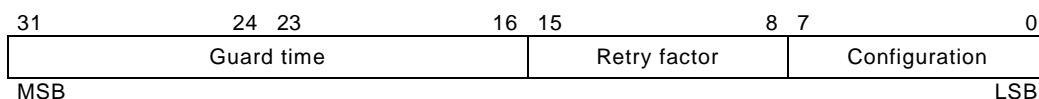


Figure 27 – Object structure of the value

7	6	5	4	3	2	1	0
Restore	Software update	Software version	Reset communication	Mandatory	NMT boot slave	reserved 0 _b	NMT slave
MSB							LSB

Figure 28 – Bit structure of the configuration value**Table 37 – NMT slave (bit: 0)**

Value	Description
0 _b	NMT master or not available in the network
1 _b	NMT slave and available in the network

Table 38 – NMT boot slave (bit: 2)

Value	Description
0 _b	Configuration and NMT service <i>Start remote node</i> shall not be allowed in case of error control event or NMT service <i>Bootup</i> NOTE The application is responsible for the NMT slave startup.
1 _b	Configuration and NMT service <i>Start remote node</i> shall be performed in case of error control event or NMT service <i>Bootup</i>

Table 39 – Mandatory (bit: 3)

Value	Description
0 _b	CANopen device may be present prior to network startup (CANopen device is optional)
1 _b	CANopen device shall be present prior to network startup (CANopen device is mandatory)

Table 40 – Reset communication (bit: 4)

Value	Description
0 _b	NMT service <i>Reset communication</i> may be executed for the CANopen device at any time
1 _b	NMT service <i>Reset communication</i> shall not be executed for the CANopen device in case the CANopen device is in NMT state <i>Operational</i>

Table 41 – Software version (bit: 5)

Value	Description
0 _b	Software version verification shall not be performed for the CANopen device
1 _b	Software version verification shall be performed for the CANopen device

Table 42 – Software update (bit: 6)

Value	Description
0 _b	Software update shall not be performed for the CANopen device
1 _b	Software update shall be performed for the CANopen device

Table 43 – Restore (bit: 7)

Value	Description
0 _b	CANopen device may be used without prior resetting.
1 _b	CANopen device shall be reset to factory defaults by issuing a restore to defaults (object 1011 _h)

Table 44 – Object description

Attribute	Value
Index	1F81 _h
Name	NMT slave assignment
Object code	ARRAY
Data type	UNSIGNED32
Category	Conditional; mandatory if the CANopen device is a CANopen manager

Table 45 – Entry description

Attribute	Value
Sub-index	00 _h
Description	Number of entries
Entry category	Mandatory
Access	const
PDO mapping	No
Value range	7F _h
Default value	7F _h
Sub-index	01 _h
Description	CANopen Node-ID 1
Entry category	Mandatory
Access	rw
PDO mapping	No
Value range	see Figure 27, Figure 28, Table 37, Table 38, Table 39, Table 40, Table 41, Table 42, and Table 43
Default value	0000 0000 _h
to	
Sub-index	7F _h
Description	CANopen Node-ID 127
Entry category	Mandatory
Access	rw
PDO mapping	No
Value range	see Figure 27, Figure 28, Table 37, Table 38, Table 39, Table 40, Table 41, Table 42, and Table 43
Default value	0000 0000 _h

9.8.9 Object 1F82_h: Request NMT

This object shall request a specific NMT service for a unique CANopen device in the network or for all CANopen devices in the network in case the CANopen device implementing this

object is in NMT master mode. Normally, the request is issued by another CANopen device (e.g. configuration tool) in the network or by the application on the very same CANopen device (e.g. in an IEC 61131 environment).

The sub-index shall correspond to the CANopen Node-ID of the CANopen devices in the network. These requests may be applied for the NMT master itself. Table 46 defines the value contents. Table 47 and Table 48 define the object description and the entry description.

The values from 84_h to $8F_h$ require the knowledge of the CANopen Node-ID of the requesting CANopen device. If the CANopen Node-ID is unknown the service shall be aborted (abort code: $0800\ 0000_h$ or $0609\ 0030_h$).

An attempt to download a value that is reserved shall be responded with an SDO abort message (abort code: $0800\ 0000_h$ or $0609\ 0030_h$).

NOTE The values from 00_h to $7F_h$ have to be applied carefully to not unintentionally affect the NMT master or the requesting CANopen device itself.

Table 46 – Value definition

Value	Description	
	on upload (read)	on download (write)
00 _h	NMT state unknown	reserved
01 _h	CANopen device missing	reserved
02 _h	reserved	
03 _h	reserved	
04 _h	NMT state <i>Stopped</i>	NMT service <i>Stop remote node</i>
05 _h	NMT state <i>Operational</i>	NMT service <i>Start remote node</i>
06 _h	reserved	NMT service <i>Reset node</i>
07 _h	reserved	NMT service <i>Reset communication</i>
08 _h	reserved	
.....	
7E _h	reserved	
7F _h	NMT state <i>Pre-operational</i>	NMT service <i>Enter pre-operational</i>
80 _h	reserved	
.....	
83 _h	reserved	
84 _h	NMT state <i>Stopped</i>	NMT service <i>Stop remote node</i> (excluding NMT master and requesting CANopen device)
85 _h	NMT state <i>Operational</i>	NMT service <i>Start remote node</i> (excluding NMT master and requesting CANopen device)
86 _h	reserved	NMT service <i>Reset node</i> (excluding NMT master and requesting CANopen device)
87 _h	reserved	NMT service <i>Reset communication</i> (excluding NMT master and requesting CANopen device)
88 _h	reserved	
.....	
8E _h	reserved	
8F _h	NMT state <i>Pre-operational</i>	NMT service <i>Enter pre-operational</i> (excluding NMT master and requesting CANopen device)
90 _h	reserved	
.....	
FF _h	reserved	

Table 47 – Object description

Attribute	Value
Index	1F82 _h
Name	Request NMT
Object code	ARRAY
Data type	UNSIGNED8
Category	Optional

Table 48 – Entry description

Attribute	Value
Sub-index	00 _h
Description	Number of entries
Entry category	Mandatory
Access	const
PDO mapping	No
Value range	80 _h
Default value	80 _h
Sub-index	01 _h
Description	CANopen Node-ID 1
Entry category	Mandatory
Access	rw
PDO mapping	No
Value range	see Table 46
Default value	00 _h
to	
Sub-index	7F _h
Description	CANopen Node-ID 127
Entry category	Mandatory
Access	rw
PDO mapping	No
Value range	see Table 46
Default value	00 _h
Sub-index	80 _h
Description	All nodes
Entry category	Mandatory
Access	wo
PDO mapping	No
Value range	see Table 46
Default value	00 _h

9.8.10 Object 1F83_h: Request node guarding

This object shall request node guarding for a unique CANopen device in the network or for all CANopen devices in the network in case the CANopen device implementing this object is in NMT master mode and node guarding is enabled. Normally, the request is issued by another CANopen device (e.g. configuration tool) in the network or by the application on the very same CANopen device (e.g. in an IEC 61131 environment).

The sub-index shall correspond to the CANopen Node-ID of the CANopen devices in the network. The sub-index corresponding to its own CANopen Node-ID shall be ignored. Table 49 defines the value contents. Table 50 and Table 51 define the object description and the entry description.

An attempt to download a value that is reserved shall be responded with an abort message (abort code: 0800 0000_h or 0609 0030_h).

NOTE Heartbeat consumer is covered by object 1016_h (see CiA301).

Table 49 – Value definition

Value	Description	
	on download (write)	on upload (read)
00 _h	Stop node guarding	Node guarding stopped
01 _h	Start node guarding	Node guarding started
02 _h	reserved	
.....	
FF _h	reserved	

Table 50 – Object description

Attribute	Value
Index	1F83 _h
Name	Request node guarding
Object code	ARRAY
Data type	UNSIGNED8
Category	Optional

Table 51 – Entry description

Attribute	Value
Sub-index	00 _h
Description	Number of entries
Entry category	Mandatory
Access	const
PDO mapping	No
Value range	80 _h
Default value	80 _h
Sub-index	01 _h
Description	CANopen Node-ID 1
Entry category	Mandatory
Access	rw
PDO mapping	No
Value range	see Table 49
Default value	00 _h
Sub-index	7F _h
Description	CANopen Node-ID 127
Entry category	Mandatory
Access	rw
PDO mapping	No
Value range	see Table 49
Default value	00 _h
to	
Sub-index	80 _h
Description	All nodes
Entry category	Mandatory
Access	wo
PDO mapping	No
Value range	see Table 49
Default value	00 _h

9.8.11 Object 1F84_h: Device type identification

This object is used for verification of the device type of the CANopen devices in the network.

The device type (object 1000_h – see EN 50325-4) of the CANopen device in the network shall be matched against the value of this object in case the value is unequal to 0. An error event shall be generated if the values mismatch. In case the value of this object is 0 the device type of the CANopen device in the network may not be verified. Table 52 and Table 53 define the object description and the entry description.

The sub-index shall correspond to the CANopen Node-ID of the CANopen devices in the network. The sub-index corresponding to its own CANopen Node-ID shall be ignored.

Table 52 – Object description

Attribute	Value
Index	1F84 _h
Name	Device type identification
Object code	ARRAY
Data type	UNSIGNED32
Category	Optional

Table 53 – Entry description

Attribute	Value
Sub-index	00 _h
Description	Number of entries
Entry category	Mandatory
Access	const
PDO mapping	No
Value range	7F _h
Default value	7F _h
Sub-index	01 _h
Description	CANopen Node-ID 1
Entry category	Mandatory
Access	rw
PDO mapping	No
Value range	See value definition of object 1000 _h – EN 50325-4
Default value	0000 0000 _h
to	
Sub-index	7F _h
Description	CANopen Node-ID 127
Entry category	Mandatory
Access	rw
PDO mapping	No
Value range	See value definition in EN 50325-4 of object 1000 _h
Default value	0000 0000 _h

9.8.12 Object 1F85_h: Vendor identification

This object shall be used for verification of the Vendor ID of the CANopen devices in the network.

The Vendor ID (object 1018_h sub-index 01_h – see EN 50325-4) of the CANopen device in the network shall be matched against the value of this object in case the value is unequal to 0. An error event shall be generated if the values mismatch. In case the value of this object is 0 the Vendor ID of the CANopen device in the network may not be verified. Table 54 and Table 55 define the object description and the entry description.

The sub-index shall correspond to the CANopen Node-ID of the CANopen devices in the network. The sub-index corresponding to its own CANopen Node-ID shall be ignored.

Table 54 – Object description

Attribute	Value
Index	1F85 _h
Name	Vendor identification
Object code	ARRAY
Data type	UNSIGNED32
Category	Optional

Table 55 – Entry description

Attribute	Value
Sub-index	00 _h
Description	Number of entries
Entry category	Mandatory
Access	const
PDO mapping	No
Value range	7F _h
Default value	7F _h
Sub-index	01 _h
Description	CANopen Node-ID 1
Entry category	Mandatory
Access	rw
PDO mapping	No
Value range	See value definition in EN 50325-4 object 1018 _h sub-index 01 _h
Default value	0000 0000 _h
to	
Sub-index	7F _h
Description	CANopen Node-ID 127
Entry category	Mandatory
Access	rw
PDO mapping	No
Value range	See value definition in EN 50325-4 object 1018 _h sub-index 01 _h
Default value	0000 0000 _h

9.8.13 Object 1F86_h: Product code

This object shall be used for verification of the product code of the CANopen devices in the network.

The product code (object 1018_h sub-index 02_h – see EN 50325-4) of the CANopen device in the network shall be matched against the value of this object in case the value is unequal to 0. An error event shall be generated if the values mismatch. In case the value of this object is 0 the product code of the CANopen device in the network may not be verified. Table 56 and Table 57 define the object description and the entry description.

The sub-index shall correspond to the CANopen Node-ID of the CANopen devices in the network. The sub-index corresponding to its own CANopen Node-ID shall be ignored.

Table 56 – Object description

Attribute	Value
Index	1F86 _h
Name	Product code
Object code	ARRAY
Data type	UNSIGNED32
Category	Optional

Table 57 – Entry description

Attribute	Value
Sub-index	00 _h
Description	Number of entries
Entry category	Mandatory
Access	const
PDO mapping	No
Value range	7F _h
Default value	7F _h
Sub-index	01 _h
Description	CANopen Node-ID 1
Entry category	Mandatory
Access	rw
PDO mapping	No
Value range	See value definition of EN 50325-4 object 1018 _h sub-index 02 _h
Default value	0000 0000 _h
to	
Sub-index	7F _h
Description	CANopen Node-ID 127
Entry category	Mandatory
Access	rw
PDO mapping	No
Value range	See value definition in EN 50325-4 object 1018 _h sub-index 02 _h
Default value	0000 0000 _h

9.8.14 Object 1F87_h: Revision number

This object shall be used for verification of the revision number of the CANopen devices in the network.

The revision number (object 1018_h sub-index 03_h – see EN 50325-4) of the CANopen device in the network shall be matched against the value of this object in case the value is unequal to 0. An error event shall be generated if the values mismatch. A mismatch is defined as:

- the major revision number is unequal to the expected major revision number, or
- the minor revision number is less than the expected minor revision number,

In case the value of this object is 0 the revision number of the CANopen device in the network shall not be verified. Table 58 and Table 59 define the object description and the entry description.

The sub-index shall correspond to the CANopen Node-ID of the CANopen devices in the network. The sub-index corresponding to its own CANopen Node-ID shall be ignored.

Table 58 – Object description

Attribute	Value
Index	1F87 _h
Name	Revision number
Object code	ARRAY
Data type	UNSIGNED32
Category	Optional

Table 59 – Entry description

Attribute	Value
Sub-index	00 _h
Description	Number of entries
Entry category	Mandatory
Access	const
PDO mapping	No
Value range	7F _h
Default value	7F _h
Sub-index	01 _h
Description	CANopen Node-ID 1
Entry category	Mandatory
Access	rw
PDO mapping	No
Value range	See value definition of object 1018 _h sub-index 03 _h – EN 50325-4
Default value	0000 0000 _h
to	
Sub-index	7F _h
Description	CANopen Node-ID 127
Entry category	Mandatory
Access	rw
PDO mapping	No
Value range	See value definition of object 1018 _h sub-index 03 _h – EN 50325-4
Default value	0000 0000 _h

9.8.15 Object 1F88_h: Serial number

This object shall be used for verification of the serial number of the CANopen devices in the network.

The serial number (object 1018_h sub-index 04_h – see EN 50325-4) of the CANopen device in the network shall be matched against the value of this object in case the value is unequal to 0. An error event shall be generated if the values mismatch. In case the value of this object is 0 the serial number of the CANopen device in the network may not be verified. Table 60 and Table 61 define the object description and the entry description.

The sub-index shall correspond to the CANopen Node-ID of the CANopen devices in the network. The sub-index corresponding to its own CANopen Node-ID shall be ignored.

Table 60 – Object description

Attribute	Value
Index	1F88 _h
Name	Serial number
Object code	ARRAY
Data type	UNSIGNED32
Category	Optional

Table 61 – Entry description

Attribute	Value
Sub-index	00 _h
Description	Number of entries
Entry category	Mandatory
Access	const
PDO mapping	No
Value range	7F _h
Default value	7F _h
Sub-index	01 _h
Description	CANopen Node-ID 1
Entry category	Mandatory
Access	rw
PDO mapping	No
Value range	See value definition of EN 50325-4 object 1018 _h sub-index 04 _h
Default value	0000 0000 _h
to	
Sub-index	7F _h
Description	CANopen Node-ID 127
Entry category	Mandatory
Access	rw
PDO mapping	No
Value range	See value definition of EN 50325-4 object 1018 _h sub-index 04 _h
Default value	0000 0000 _h

9.8.16 Object 1F89_h: Boot time

The object defines the time out between start of the process *Start process boot NMT slave* and signaling of successful boot of all mandatory NMT slaves. The details are defined in 9.4.3.

The value shall be given in multiples of ms. The value 0 shall disable the timer. Table 62 provides the object description and Table 63 the entry description.

Table 62 – Object description

Attribute	Value
Index	1F89 _h
Name	Boot time
Object code	VAR
Data type	UNSIGNED32
Category	Optional

Table 63 – Entry description

Attribute	Value
Sub-index	00 _h
Access	rw
PDO mapping	No
Value range	UNSIGNED32
Default value	0000 0000 _h

9.8.17 Object 1F8A_h: Restore configuration

This object is used to define the allowed restore procedure for a CANopen device during startup.

The restore procedure configured in this object shall be used to restore, during startup, each CANopen device in the network (object 1011_h – see EN 50325-4) according to the object entries. Table 64 and Table 65 define the object description and the entry description.

The sub-index shall correspond to the CANopen Node-ID of the CANopen devices in the network. The sub-index corresponding to its own CANopen Node-ID shall be ignored.

Table 64 – Object description

Attribute	Value
Index	1F8A _h
Name	Restore configuration
Object code	ARRAY
Data type	UNSIGNED8
Category	Optional

Table 65 – Entry description

Attribute	Value
Sub-index	00 _h
Description	Number of entries
Entry category	Mandatory
Access	const
PDO mapping	No
Value range	7F _h
Default value	7F _h
Sub-index	01 _h
Description	CANopen Node-ID 1
Entry category	Mandatory
Access	rw
PDO mapping	No
Value range	01 _h to 7F _h
Default value	01 _h
to	
Sub-index	7F _h
Description	CANopen Node-ID 127
Entry category	Mandatory
Access	rw
PDO mapping	No
Value range	01 _h to 7F _h
Default value	01 _h

9.8.18 Object 1F91_h: Self-starting nodes timing parameters

This object defines parameters that shall be configured to apply a determined behavior for self-starting CANopen devices. Table 66 and Table 67 define the object description and entry description.

The NMT master detection timeout, NMT master request delay time and node time slot shall be given in multiples of ms.

Table 66 – Object description

Attribute	Value
Index	1F91 _h
Name	Self-starting nodes timing parameters
Object code	ARRAY
Data type	UNSIGNED16
Category	Optional; Mandatory for CANopen devices supporting self-starting

Table 67 – Entry description

Attribute	Value
Sub-index	00 _h
Description	Number of entries
Entry category	Mandatory
Access	const
PDO mapping	No
Value range	03 _h
Default value	03 _h
Sub-index	01 _h
Description	NMT master detection timeout
Entry category	Mandatory
Access	rw
PDO mapping	No
Value range	UNSIGNED16
Default value	100
Sub-index	02 _h
Description	NMT master request delay time
Entry category	Mandatory
Access	rw
PDO mapping	No
Value range	UNSIGNED16
Default value	500
Sub-index	03 _h
Description	Node time slot
Entry category	Mandatory
Access	rw
PDO mapping	No
Value range	UNSIGNED 16
Default value	15

10 Gateway functions

10.1 Content

This clause provides the gateway functions for gateways between a Train Backbone and a CANopen-based consist network.

Network access services are specified to enable the access to CANopen-based consist networks. In addition a mapping of these CANopen access services to an ASCII protocol is defined. In order to fit into a TCN architecture, the mapping of this protocol to management messages is defined in Clause 11.

Depending on the implemented capabilities, this clause introduces also several gateway classes.

10.2 Gateway architecture

The architecture of a gateway between a Train backbone and a CANopen consist network is shown in Figure 29. In the illustration the Train backbone is implemented as WTB.

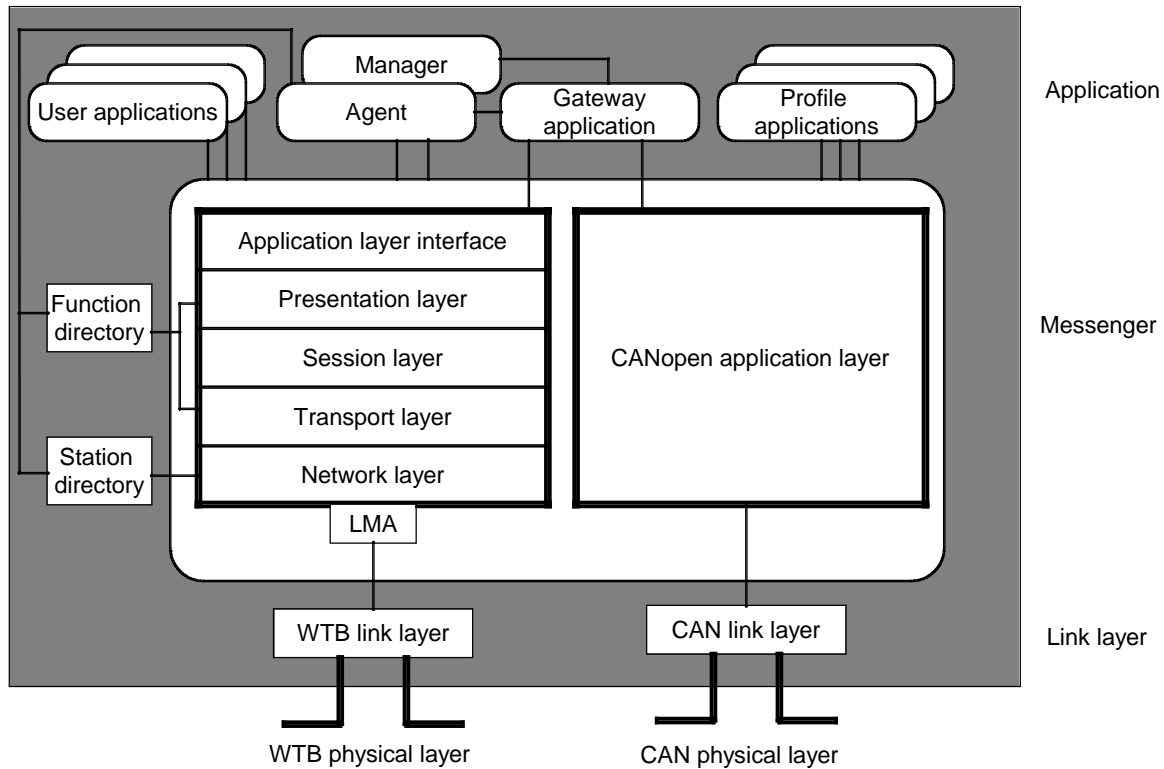


Figure 29 – Gateway between Train backbone and CANopen consist network

A gateway interprets between a Train Backbone and a CANopen-based consist network. It transfers the process data via the gateway application and ensures the correct endian style for the data representation on both sides of the gateway.

NOTE The ordering scheme for transferring data in CANopen is little endian in contrast to WTB, which uses big endian style.

By means of TNM management messages TNM services may be requested from both sides of the gateway. In addition to the services specified in IEC 61375-2-1, further TNM services are defined (see clause 11.5) that enable full control of the sub-layered CANopen consist network (see network access services as defined in 10.4). The network access services provided by a gateway - between the Train Backbone and a CANopen-based consist network - give one Node, which is connected to the Train Backbone, access to End Devices connected to a CANopen-based consist network.

As a mapping of TNM management messages to CANopen communication objects (COBs) is defined, the TNM management services are available within CANopen-based consist networks as well.

As shown in Figure 29, the Manager functionality as well as the Agent functionality reside on application level, as defined in IEC 61375-2-1. In addition manufacturer-specific user applications as well as standardized applications according to CANopen profiles (profile applications) may reside on this level.

10.3 General principles and services

10.3.1 Content

This subclause provides the network access services provided by a gateway device that enables to access from the Train Backbone the CANopen-based consist network.

10.3.2 Gateway class definitions

A CANopen device, in which the gateway functionality to the Train Backbone resides, may support one or more of the following classes:

Class 0: The gateway is a device, acting as network slave (NMT slave functionality) within the CANopen network. The device supports only the process data transfer capability. A gateway of this class is not capable to communicate TNM management messages via the CANopen network.

Class 1: The gateway is a device, acting as network slave (NMT slave functionality) within the CANopen network. The device shall provide, in addition to the functionality defined for gateways of Class 0, the SDO client functionality.

Class 2: The gateway is a device implementing the functionality of a Class 1 device, which additionally implements SDO requesting device (SRD) functionality.

Class 3: The gateway is a device within the CANopen network acting as the CANopen manager. The CANopen manager functionality shall be achieved by supporting the NMT master functionality as well as the SDO manager functionality.

NOTE Only gateway devices of class 1, 2 or 3 are capable to communicate TNM management messages within CANopen-based consist networks.

10.3.3 Service primitives definitions

Via service primitives the services are utilized by which the gateway application and the network application layer interact. The service primitives are defined in IEC 61375-1.

10.4 Network access service specification

10.4.1 SDO access services

10.4.1.1 Content

The services specified in this clause are used to initiate and configure SDO services accessing any object in the object dictionary of any node on any of the CANopen networks linked to the gateway device.

10.4.1.2 Upload SDO

This service shall initiate an SDO upload service. The parameters for this service are defined in Table 68.

Table 68 – Upload SDO service

Parameter	Indication	Response
Argument Network Node-ID Multiplexor Data type BOOLEAN UNSIGNED8 UNSIGNED16 UNSIGNED24 UNSIGNED32 UNSIGNED40 UNSIGNED48 UNSIGNED56 UNSIGNED64 INTEGER8 INTEGER16 INTEGER24 INTEGER32 INTEGER40 INTEGER48 INTEGER56 INTEGER64 REAL32 REAL64 TIME_OF_DAY TIME_DIFFERENCE OCTET_STRING VISIBLE_STRING UNICODE_STRING DOMAIN Offset Length	Mandatory Optional Optional Mandatory Optional Selection Optional Optional	
Remote result Success Data Length Failure Reason		Mandatory Selection Mandatory Optional Selection Mandatory

10.4.1.3 Download SDO

This service shall initiate an SDO download service. The parameters for this service are defined in Table 69.

Table 69 – Download SDO parameters

Parameter	Indication	Response
Argument Network Node-ID Multiplexor Data type BOOLEAN UNSIGNED8 UNSIGNED16 UNSIGNED24 UNSIGNED32 UNSIGNED40 UNSIGNED48 UNSIGNED56 UNSIGNED64 INTEGER8 INTEGER16 INTEGER24 INTEGER32 INTEGER40 INTEGER48 INTEGER56 INTEGER64 REAL32 REAL64 TIME_OF_DAY TIME_DIFFERENCE OCTET_STRING VISIBLE_STRING UNICODE_STRING DOMAIN Offset Length Data	Mandatory Optional Optional Mandatory Optional Selection Selection Selection Selection Selection Selection Selection Selection Selection Selection Selection Selection Selection Selection Selection Selection Selection Selection Selection Optional Optional Mandatory	
Remote result Success Failure Reason		Mandatory Selection Selection Mandatory

10.4.1.4 Configure SDO timeout

This service shall configure the time-out for all Client-SDOs on the gateway device. The parameters for this service are defined in Table 70.

Table 70 – Configure SDO timeout parameters

Parameter	Indication	Response
Argument Network SDO Timeout	Mandatory Optional Mandatory	
Remote result Success Failure Reason		Mandatory Selection Selection Mandatory

10.4.2 PDO access services

10.4.2.1 Content

The services specified in this clause are used to configure and initiate PDO services in the gateway device. They include:

- Configure a receive PDO

- Configure a transmit PDO
- Request to read a PDO
- Request to write a PDO
- Indicate a received PDO

The two PDO configuration services are intended to create PDOs in the gateway device. If the gateway device implements an object dictionary, the PDO communication and mapping parameter entries shall be set-up accordingly.

The two PDO request services are intended to control the PDOs in accordance to the configured PDO transmission type.

The data types `VISIBLE_STRING`, `OCTET_STRING`, and `UNICODE_STRING`, as well as `DOMAIN` shall not be used as PAS data type parameter.

NOTE The PAS services are not intended to configure the PDO communication and mapping entries of the object dictionary of the remote nodes. Accessing the CANopen nodes individually by means of SAS services may do the PDO configuration.

10.4.2.2 Configure RPDO

This service shall create an RPDO in the gateway device. Table 71 defines the parameters for this service.

Table 71 – Configure RPDO service parameters

<i>Parameter</i>	<i>Indication</i>	<i>Response</i>
Argument Network PDO number COB-ID TxType Nbr_objects 1 st mapped object Data type Multiplexor 2 nd mapped object Data type Multiplexor ... 64 th mapped object Data type Multiplexor	Mandatory Optional Mandatory Mandatory Mandatory Mandatory Mandatory Selection Selection Optional Selection Selection ... Optional Selection Selection	
Remote result Success Failure Reason		Mandatory Selection Selection Mandatory

10.4.2.3 Configure TPDO

This service shall create a TPDO in the gateway device. Table 72 defines the parameters for this service.

Table 72 – Configure TPDO service parameters

Parameter	Indication	Response
Argument Network PDO number COB-ID TxType Nbr_objects 1 st mapped object Data type Multiplexor 2 nd mapped object Data type Multiplexor ... 64 th mapped object Data type Multiplexor	Mandatory Optional Mandatory Mandatory Mandatory Mandatory Selection Selection Optional Selection Selection ... Optional Selection Selection	
Remote result Success Failure Reason		Mandatory Selection Selection Mandatory

10.4.2.4 Read PDO data

This service shall read the data received by an RPDO. If an RPDO is configured with transmission type 252 or 253, the gateway devices shall trigger it by means of an RTR.

NOTE It is not recommended to use RTR.

The received data shall be transmitted in the remote result. Table 73 defines the parameters for this service.

Table 73 – Read PDO data service parameters

Parameter	Indication	Response
Argument Network PDO number	Mandatory Optional Mandatory	
Remote result Success Network PDO number Nbr_objects Data 1 st object ... Data 64 th object Failure Reason		Mandatory Selection Optional Mandatory Mandatory Conditional ... Conditional Selection Mandatory

10.4.2.5 Write PDO data

This service shall trigger the transmission of a PDO. The actual transmission of the PDO shall be triggered according to the configured PDO transmission type.

The parameters of this service are defined in Table 74.

Table 74 – Write PDO data service parameters

Parameter	Indication	Response
Argument Network PDO number Nbr_objects Data 1 st object ... Data 64 th object	Mandatory Optional Mandatory Mandatory Conditional ... Conditional	
Remote result Success Failure Reason		Mandatory Selection Selection Mandatory

10.4.2.6 RPDO received

This service shall signal that new PDO data has been received and shall provide the received data.

The parameters of this service are defined in Table 75.

Table 75 – RPDO received service parameters

Parameter	Request
Argument Network PDO number Nbr_objects 1 st object value ... 64 th object value	Mandatory Optional Mandatory Mandatory Conditional ... Conditional

10.4.3 CANopen NMT services

10.4.3.1 Content

The services specified in this clause are used to control a CANopen device or a CANopen network and associated error control services.

10.4.3.2 Start node

This service shall set CANopen nodes into NMT state OPERATIONAL. For Class 1 and Class 2 devices this service shall trigger a CANopen Request NMT service. For Class 3 devices the service shall trigger a Start Remote Node service. For Class 0 devices this shall trigger a failure notification as remote result. Table 76 defines the parameters for this service.

Table 76 – Start node service parameters

Parameter	Indication	Response
Argument Network Node-ID All	Mandatory Optional Selection Selection	
Remote result Success Failure Reason		Mandatory Selection Selection Mandatory

10.4.3.3 Stop node

This service shall set CANopen nodes into NMT state STOPPED. For class 1 and class 2 devices this service shall trigger a CANopen request NMT service. For class 3 devices the service shall trigger a stop remote node service. For Class 0 devices this shall trigger a failure notification as remote result. Table 77 defines the parameters for this service.

NOTE The remote result for class 1 and 2 devices is only the confirmation of the SDO request; for class 3 devices the remote result is based on the error control services as defined in EN 50325-4.

Table 77 – Stop node service parameters

Parameter	Indication	Response
Argument Network Node-ID All	Mandatory Optional Selection Selection	
Remote result Success Failure Reason		Mandatory Selection Selection Mandatory

10.4.3.3.1 Set node to pre-operational

This service shall set CANopen nodes into NMT state PRE-OPERATIONAL. For class 1 and class 2 devices this service shall trigger a CANopen request NMT service. For class 3 devices the service shall trigger an Enter Pre-operational service. For Class 0 devices this shall trigger a failure notification as remote result. Table 78 defines the parameters for this service.

Table 78 – Set node to pre-operational service parameters

Parameter	Indication	Response
Argument Network Node-ID All	Mandatory Optional Selection Selection	
Remote result Success Failure Reason		Mandatory Selection Selection Mandatory

10.4.3.4 Reset node

This service shall set CANopen nodes into NMT state RESET APPLICATION. For class 1 and class 2 devices this service shall trigger a CANopen request NMT service. For class 3 devices the service shall trigger a reset node service. For Class 0 devices this shall trigger a failure notification as remote result. Table 79 defines the parameters for this service.

Table 79 – Reset node service parameters

Parameter	Indication	Response
Argument Network Node-ID All	Mandatory Optional Selection Selection	
Remote result Success Failure Reason		Mandatory Selection Selection Mandatory

10.4.3.5 Reset communication

This service shall set CANopen nodes into NMT state RESET COMMUNICATION. For class 1 and class 2 devices this service shall trigger a CANopen request NMT service. For class 3 devices the service shall trigger a reset communication service. For Class 0 devices this shall trigger a failure notification as remote result. Table 80 defines the parameters for this service.

Table 80 – Reset communication service parameters

Parameter	Indication	Response
Argument Network Node-ID All	Mandatory Optional Selection Selection	
Remote result Success Failure Reason		Mandatory Selection Selection Mandatory

10.4.3.6 Enable node guarding

This service is only available for class 3 devices. It shall start node guarding for the device specified by CANopen Node-ID with the parameters given by GuardTime and LifeTimeFactor. If heartbeat is already activated on the addressed node, the service request shall be rejected. Table 81 defines the parameters for this service.

Table 81 – Enable node guarding service parameters

Parameter	Indication	Response
Argument Network Node-ID GuardTime LifeTimefactor	Mandatory Optional Mandatory Mandatory Mandatory	
Remote result Success Failure Reason		Mandatory Selection Selection Mandatory

10.4.3.7 Disable node guarding

This service is only available for class 3 devices. It shall stop node guarding for the device specified by CANopen Node-ID. Table 82 defines the parameters for this service.

Table 82 – Disable node guarding service parameters

Parameter	Indication	Response
Argument Network Node-ID	Mandatory Optional Mandatory	
Remote result Success Failure Reason		Mandatory Selection Selection Mandatory

10.4.3.8 Start heartbeat consumer

This service shall start the consumption of heartbeat messages transmitted by a CANopen device specified by CANopen Node-ID. Table 83 defines parameters for this service.

Table 83 – Start heartbeat consumer service parameters

Parameter	Indication	Response
Argument Network Node-ID HeartbeatConsumerTime	Mandatory Optional Mandatory Mandatory	
Remote result Success Failure Reason		Mandatory Selection Selection Mandatory

10.4.3.9 Disable heartbeat consumer

This service shall stop the consumption of heartbeat messages transmitted by a CANopen device specified by CANopen Node-ID. Table 84 defines parameters for this service.

Table 84 – Disable heartbeat consumer service parameters

Parameter	Indication	Response
Argument Network Node-ID	Mandatory Optional Mandatory	
Remote result Success Failure Reason		Mandatory Selection Selection Mandatory

10.4.3.10 Error control event received

This service shall signal the NMT status or error control events received from a CANopen node specified by CANopen Node-ID. Table 85 defines the parameters for this service.

Table 85 – Error control event received parameters

Parameter	Request
Argument Network Node-ID Status Error Code	Mandatory Optional Mandatory Selection Selection

10.4.4 Device failure management services**10.4.4.1 General**

The services specified in this clause are used to manage failures within the gateway device or within any other CANopen device.

10.4.4.2 Read device error

This service shall read EMCY message information received from the CANopen device specified by the CANopen Node-ID parameter. Table 86 defines the parameters for this service.

Table 86 – Read device error service parameters

Parameter	Indication	Response
Argument Network Node-ID	Mandatory Optional Optional	
Remote result Success Network Node-ID Error Error Msg number Error Msg text Emergency Emergency code Error register Manufacturer error Failure Reason		Mandatory Selection Optional Mandatory Selection Mandatory Optional Selection Mandatory Optional Optional Selection Mandatory

10.4.4.3 Emergency event received

This service shall signal the reception of an emergency message in the gateway device transmitted by a CANopen device specified by the CANopen Node-ID. Table 87 defines the parameters for this service.

Table 87 – Emergency event received service parameters

Parameter	Request
Argument Network Node-ID Emergency code Error register Manufacturer error	Mandatory Optional Mandatory Mandatory Mandatory Optional

10.4.5 CANopen interface configuration services

10.4.5.1 General

The services described in this clause are used to configure and parameterize the CANopen interface of the gateway device.

10.4.5.2 Initialize gateway

This service shall initiate the CANopen initialization of the gateway device. It shall perform a power-on equivalent reset of the CANopen interface. It is used to initialize the bit-timing parameters. Table 88 defines the parameters for this service.

Table 88 – Initialize gateway service parameters

Parameter	Indication	Response
Argument Network CAN bit timing	Mandatory Optional Optional	
Remote result Success Failure Reason		Mandatory Selection Selection Mandatory

10.4.5.3 Store configuration

This service shall command the gateway device to store its CANopen interface configuration. Table 89 defines the parameters for this service.

Table 89 – Store configuration service parameters

Parameter	Indication	Response
Argument Network Storage specifier	Mandatory Optional Optional	
Remote result Success Failure Reason		Mandatory Selection Selection Mandatory

10.4.5.4 Restore configuration

This service shall command the gateway device to restore its CANopen interface configuration. Table 90 defines the parameters for this service.

Table 90 – Restore configuration service parameters

Parameter	Indication	Response
Argument Network Storage specifier	Mandatory Optional Optional	
Remote result Success Failure Reason		Mandatory Selection Selection Mandatory

10.4.5.5 Set heartbeat producer

This service shall set the CANopen heartbeat producer time in the gateway device. Table 91 defines the parameters for this service.

Table 91 – Set heartbeat producer service parameters

Parameter	Indication	Response
Argument Network Node-ID HeartbeatProducerTime	Mandatory Optional Mandatory Mandatory	
Remote result Success Failure Reason		Mandatory Selection Selection Mandatory

10.4.5.6 Set node-ID

This service shall set the CANopen Node-ID in the gateway device for the CANopen network given in the *network* parameter. Table 92 defines the parameters for this service.

Table 92 – Set node-ID service parameters

Parameter	Indication	Response
Argument Network Node-ID	Mandatory Optional Mandatory	
Remote result Success Failure Reason		Mandatory Selection Selection Mandatory

10.4.5.7 Start emergency consumer

This service shall start the consumption of emergency messages. The relation between CANopen Node-ID producing an emergency message and the COB-ID has to be explicitly known. Table 93 defines the parameters for this service.

Table 93 – Start emergency consumer service parameters

Parameter	Indication	Response
Argument Network Node-ID COB-ID	Mandatory Optional Mandatory Mandatory	
Remote result Success Failure Reason		Mandatory Selection Selection Mandatory

10.4.5.8 Stop emergency consumer

This service shall stop the consumption of emergency messages. The relation between CANopen Node-ID producing an emergency message and the COB-ID has to be explicitly known. Table 94 defines the parameters for this service.

Table 94 – Stop emergency consumer service parameters

Parameter	Indication	Response
Argument Network Node-ID COB-ID	Mandatory Optional Mandatory Mandatory	
Remote result Success Failure Reason		Mandatory Selection Selection Mandatory

10.4.6 Gateway management services

10.4.6.1 General

The services specified in this clause are used to manage the gateway device.

10.4.6.2 Set default network

This service shall set the default network number, which shall be used for all services. Table 95 defines the parameters for this service.

Table 95 – Set default network service parameters

Parameter	Indication	Response
Argument DefaultNetwork	Mandatory Mandatory	
Remote result Success Failure Reason		Mandatory Selection Selection Mandatory

10.4.6.3 Set default node-ID

This service shall set the default CANopen Node-ID, which shall be used for all services. Table 96 defines the parameters for this service.

Table 96 – Start default node-ID service parameters

Parameter	Indication	Response
Argument Default node-ID	Mandatory Mandatory	
Remote result Success Failure Reason		Mandatory Selection Selection Mandatory

10.4.6.4 Get version

This service shall get information on the gateway device and its CANopen interface. Table 97 defines the parameters for this service.

Table 97 – Get version service parameters

Parameter	Indication	Response
Argument Network	Mandatory Optional	
Remote Result Success Vendor-ID Product code Revision number Serial number Gateway class Protocol version Implementation class Failure Reason		Mandatory Selection Mandatory Mandatory Mandatory Mandatory Mandatory Mandatory Mandatory Selection Mandatory

10.4.7 Manufacturer-specific services

The manufacturer of a gateway device may define additional services.

10.5 ASCII mapping of network access services**10.5.1 Content**

This clause defines the mapping of the network access services to an ASCII-based communication syntax for CANopen gateway devices. This protocol is mapped to TCN message data.

10.5.2 Definitions

10.5.2.1 Command

A Command controls the gateway and interacts with CANopen devices. It may have a long form and a short form. The short form is a one or two letter abbreviation of the long form. The long form is obtained by concatenating the short form and the string enclosed in brackets “[”, “]”.

NOTE In the given examples it is assumed that network address and node address are preset.

10.5.2.2 Data type syntax

The mandatory data types, provided in Table 98, shall be supported.

Table 98 – Syntax and CANopen data types

Syntax	CANopen Type	Category
b	Boolean	Mandatory
u8	UNSIGNED8	Mandatory
u16	UNSIGNED16	Mandatory
u24	UNSIGNED24	Optional
u32	UNSIGNED32	Mandatory
u40	UNSIGNED40	Optional
u48	UNSIGNED48	Optional
u56	UNSIGNED56	Optional
u64	UNSIGNED64	Optional
i8	INTEGER8	Mandatory
i16	INTEGER16	Mandatory
i24	INTEGER24	Optional
i32	INTEGER32	Mandatory
i40	INTEGER40	Optional
i48	INTEGER48	Optional
i56	INTEGER56	Optional
i64	INTEGER64	Optional
r32	REAL32	Optional
r64	REAL64	Optional
t	Time of day (with two arguments: day ms)	Optional
td	Time difference	Optional
vs	Visible string	Optional
os	Octet string	Optional
us	Unicode string	Optional
d	Domain	Optional

NOTE The value of the data type *domain*, *octet string*, and *unicode string* is encoded in mime-base64 as described in RFC 2045.

All wrapping CRLF shall be stripped from the encoded data to have one long string.

10.5.2.3 Whitespace

Whitespaces as specified in ISO/IEC 9899 except of CR and LF.

NOTE A visible string with whitespace is enclosed with double quotes to denote it as single argument of the command. If a double quote is used within the string, the quotes are escaped by a second quotes, e.g. "Hello ""World""", CANopen is great".

10.5.2.4 Command structure

10.5.2.4.1 General

The principle communication is based on non-case-sensitive ASCII strings according to ISO/IEC 646 instead of architecture and CPU/compiler depending binary structures. Due to this, no application handles with things like endianness, data size and byte alignment. In all cases where numbers are used, the typical representation is as specified in ISO/IEC 9899.

100	- decimal, starting with a number
0x64	- hexadecimal, starting with the string 0x
1.22	- float
.22e10	- float
22e3	- float

10.5.2.4.2 Request

The CANopen gateway is controlled by commands. A command is composed of tokens, which are separated by any number of whitespaces and is closed with a CRLF.

All commands are confirmed. Commands start with a sequence number which is enclosed by square brackets []. The sequence number is a 4-byte value. It is not used for event-triggered messages. According to the addressing principle, a network number and a node number follow the sequence number. Network number and node number are optional, when the CANopen gateway only provides one Train Backbone to CANopen Consist Network interface or when a client presets them. Commands that affect only the server not a remote node but a net and node are given, net and node are ignored. In BNF notation the command notation is provided in Table 99:

Table 99 – Command notation in BNF

<command-request>	::= "["<sequence>"]" [[<net>] <node>] <command>
<sequence>	::= UNSIGNED32
<net>	::= UNSIGNED8
<node>	::= UNSIGNED8
<command>	::= <command-specifier> <compound-command>
<compound-command>	::= <command-specifier> <parameter>
<parameter>	::= <value> <compound-parameter>
<compound-parameter>	::= <value> <parameter>
<map-object>	::= <datatype> <multiplexor>
<datatype>	::= 'b' 'u8' 'u16' 'u32' 'u40' 'u48' 'u56' 'u64' 'i8' 'i16' 'i24' 'i32' 'i40' 'i48' 'i56' 'i64' 'r32' 'r64' 't' 'td' 'vs' 'os' 'us' 'd'
<multiplexor>	::= <index> <subindex>
<index>	::= UNSIGNED16
<subindex>	::= UNSIGNED8

Net numbers are starting with 1. Node numbers are starting with 1. The value 0 for net or node is used to address all networks or all nodes.

The token <value> designates a value of the possible CANopen data types. Within the description of the commands the sequence number is omitted for reasons of readability.

10.5.2.4.3 Response

The CANopen gateway shall respond with the same sequence number at the first position as given by the request. This number shall be given in decimal format. There shall be only one response to a request. The response notation is provided in Table 100.

Table 100 – Response notation

<command-response>	::= "["<sequence>"]" <response>
<response>	::= <value> <error-string> <emcy-list> "OK"
<error-string>	::= "Error:" <error code>
<error-code>	::= <internal-error-code> <sdo-abort-code>
<internal-error-code>	::= see table 2
<sdo-abort-code>	::= UNSIGNED32
<emcy-list>	::= [<emcy1> " " ..<emcy254>]

The sdo-abort-codes (SAC) are defined in EN 50325-4. Allowed internal-error-codes (InEC) are listed in Table 101:

Table 101 – Internal error code (InEC)

InEC	Message text
100	Request not supported
101	Syntax error
102	Request not processed due to internal state
103	Time-out (where applicable)
200	Lost guarding message
201	Lost connection
202	Heartbeat started
203	Heartbeat lost
205	Boot-up
300	Error passive
301	Bus off
303	CAN buffer overflow
304	CAN init
305	CAN active (at init or start-up)
400	PDO already used
401	PDO length exceeded

NOTE After bus-off, the command init should be invoked to reset the CAN controller.

10.5.2.4.4 Event triggered messages

Messages due to errors in the CANopen network or the occurrence of communication objects using the producer-consumer principle shall not use a sequence number. The notation for event-triggered messages is provided in Table 102.

Table 102 – Notation for event triggered messages

<event-triggered-message>	::= [[net] node] <event-specifier> <parameter>
<event-specifier>	::= "EMCY" "ERROR" "PDO" "SYNC" "USER"

The content of event-triggered messages is described within the command description that enables the specific service.

10.5.3 Network access command specification

10.5.3.1 SDO access commands

10.5.3.1.1 General

The following command definitions shall be used to implement the SDO access services as defined in 10.4.

SDO access services are addressing a specific object at an SDO server via index and sub-index and a transfer data type.

10.5.3.1.2 Upload SDO command

Indication syntax is defined in Table 103.

Table 103 – Syntax for upload SDO command

<code>[[net] node] r[ead] <multiplexor> <datatype></code>

Examples are provided in Table 104.

Table 104 – Examples for upload SDO command

<code>[21] r 0x1000 0 u32</code>
<code>[4096] read 0x1008 0 vs</code>

Response syntax:

See 10.5.2.4.3.

10.5.3.1.3 Download SDO command

Indication syntax is defined in Table 105.

Table 105 – Syntax for Download SDO command

<code>[[net] node] w[rite] <multiplexor> <datatype> <value></code>
--

Examples are provided in Table 106.

Table 106 – Examples for download SDO command

<code>[20] 1 23 w 0x1016 0 u16 100</code>
<code>[23] write 0x1016 0 u16 0x64</code>

Response syntax:

See 10.5.2.4.3.

10.5.3.1.4 Configure SDO timeout command

The timeout delay time for abort error code ‘SDO protocol timed out’, used by the gateway’s SDO client, may be set.

Indication syntax is defined in Table 107.

Table 107 – Syntax for configure SDO timeout command

<code>[net] set sdo_timeout <ms></code>

Response syntax:

See chapter 10.5.2.4.3.

10.5.3.2 PDO access commands

10.5.3.2.1 General

The following command definitions shall be used to implement the PDO access services as defined in 10.4. Normally a PDO is first configured before transmission and reception is possible.

A PDO is seen from the view of the gateway. An RPDO therefore receives data from the CANopen network and a TPDO sends the data into the CANopen network.

NOTE In order to disable/delete a gateway PDO, the bit 31 in the COB-ID is used. For details refer to EN 50325-4.

10.5.3.2.2 Configure RPDO command

Indication syntax is defined in Table 108.

Table 108 – Syntax for configure RPDO command

<pre>[[net] node] set rpdo <nr> <COB> <tx-type> <nr-of-data> <map-obj1> [..<map-obj64>] "event"="" "sync<0..240>"<="" <tx-type>="" ::="rtr" pre="" =""> </map-obj64>]></pre>

NOTE In case a <map-obj> is given in form of index and sub-index, e.g. <multiplexor>, they are counted as 1 in the <nr-of-data>. The <nr> is an offset; it starts with 1.

Examples are provided in Table 109.

Table 109 – Examples for configure RPDO command

<pre>[12] set rpdo 1 0x180 event 3 u8 u8 u16 [24] 2 set rpdo 1 0x180 event 3 u8 u8 i16</pre>
--

Response syntax:

See 10.5.2.4.3.

10.5.3.2.3 Configure TPDO command

Indication syntax is provided in Table 110.

Table 110 – Syntax for configure TPDO command

<pre>[[net] node] set tpdo <nr> <COB> <tx-type> <nr-of-data> <map-obj1> [..<map-obj64>]< pre=""> </map-obj64>]<></pre>
--

NOTE In case a <map-obj> is given in form of index and subindex, i.e. <multiplexor>, they are counted as 1 in the <nr-of-data>. The <nr> is an offset; it starts with 1.

Example is provided in Table 111.

Table 111 – Examples for configure TPDO command

<pre>[13] set tpdo 1 0x201 rtr 4 u8 u16 u16 u8</pre>
--

Response syntax:

See 10.5.2.4.3.

NOTE It is not recommended to support RTR. Therefore it is recommended to set bit 30 of the COB-ID accordingly. For details see EN 50325-4.

10.5.3.2.4 Read PDO data command

Indication syntax is defined in Table 112.

Table 112 – Syntax for read PDO data command

[net] r[ead] p[do] <nr>

If the transmission type is different than RTR, the gateway answers with the values of the mapped objects.

Response syntax is defined in Table 113.

Table 113 – Response syntax for read PDO data command

[net] pdo <nr> <nr-of-data> <value1>[.. <value64>]< td=""> </value64>]<>
--

10.5.3.2.5 Write PDO data command

Indication syntax is defined in Table 114.

Table 114 – Syntax for write PDO data command

[net] w[rite] p[do] <nr> <nr-of-data> <value1>[.. <value64>]< td=""> </value64>]<>
--

Response syntax:

See 10.5.2.4.3.

10.5.3.2.6 RPDO received command

Indication syntax is defined in Table 115.

Table 115 – Syntax for RPDO receive command

[net] pdo <nr> <nr-of-data> <value1>[.. <value64>]< td=""> </value64>]<>
--

Examples are provided in Table 116.

Table 116 – Examples RPDO received command

1 pdo 1 2 123 4	;	# gateway with more than one network
	;	# received RPDO1 at net 1 with two objects
	;	# mapped
pdo 2 1 1234	;	# RPDO2 with one object mapped
pdo 2 3 100 2 4	;	# three objects mapped

10.5.3.3 CANopen NMT commands

10.5.3.3.1 General

The following command definitions shall be used to implement the CANopen NMT services as defined in 10.4. The supported services depend on the gateway class.

10.5.3.3.2 Start node command

Indication syntax is defined in Table 117.

Table 117 – Syntax for start node command

[[net] node] start

Response syntax:

See 10.5.2.4.3.

10.5.3.3.3 Stop node command

Indication syntax is defined in Table 118.

Table 118 – Syntax for stop node command

[[net] node] stop

Response syntax:

See 10.5.2.4.3.

10.5.3.3.4 Set node to pre-operational command

Indication syntax is defined in Table 119.

Table 119 – Syntax set node to pre-operational command

[[net] node] preop[erational]

Response syntax:

See 10.5.2.4.3.

10.5.3.3.5 Reset node command

Indication syntax is defined in Table 120.

Table 120 – Syntax reset node command

[[net] node] reset node

Response syntax:

See 10.5.2.4.3.

10.5.3.3.6 Reset communication command

Indication syntax is defined in Table 121.

Table 121 – Syntax reset communication command

[[net] node] reset comm[unication]

Response syntax:

See 10.5.2.4.3.

10.5.3.3.7 Enable node guarding command

Activating node-guarding functionality enables another event-triggered response message to the clients of the gateway. Only in the case that one of the monitored CANopen nodes violates the guarding protocol, an event message shall be sent to the clients.

Indication syntax is defined in Table 122.

Table 122 – Syntax enable node guarding command

[[net] node] enable guarding <guardingtime> <lifetimefactor>
--

Response syntax:

See 10.5.2.4.3.

10.5.3.3.8 Disable node guarding command

Indication syntax is defined in Table 123.

Table 123 – Syntax disable node guarding command

<code>[[net] node] disable guarding</code>
--

Response syntax:

See 10.5.2.4.3.

10.5.3.3.9 Start heartbeat consumer command

Activating the heartbeat consumer at the gateway enables another event-triggered response message to the gateway’s clients. Only in the case that one of the monitored CANopen nodes violates the guarding protocol, an event message shall be sent to the clients.

Indication syntax is defined in Table 124.

Table 124 – Syntax start heartbeat consumer command

<code>[[net] node] enable heartbeat <heartbeattime></code>
--

Response syntax:

See 10.5.2.4.3.

10.5.3.3.10 Disable heartbeat consumer command

Indication syntax is defined in Table 125.

Table 125 – Syntax disable heartbeat consumer command

<code>[[net] node] disable heartbeat</code>

Response syntax:

See 10.5.2.4.3.

10.5.3.3.11 Error control event received command

Response syntax is defined in Table 126.

Table 126 – Syntax for error control event received command

<code>[[net] node] ERROR <internal-error-code></code>

10.5.3.4 Device failure management commands

10.5.3.4.1 General

The following command definitions shall be used to implement the device failure management services as defined in 10.4.4.

10.5.3.4.2 Read device error command

Indication syntax is defined in Table 127.

Table 127 – Syntax for read device error command

```
[[net] node] r[ead] error
```

Response syntax:

See 10.5.2.4.3.

10.5.3.4.3 Emergency event received command

Response syntax is defined in Table 128.

Table 128 – Syntax for emergency event received command

```
[[net] node] EMCY <emcy-code> <error-register> <m-error-code>
```

The (manufacturer)-error-code shall be returned as five decimal values corresponding to the manufacturer-specific error code in the EMCY message.

10.5.3.5 CANopen interface configuration commands

10.5.3.5.1 General

The following command definitions shall be used to implement the CANopen interface configuration services as defined in 10.4.5. Settings are valid for the default network, if no network address is given. The node address shall be omitted, in case it is given.

10.5.3.5.2 Initialize gateway command

Indication syntax is defined in Table 129.

Table 129 – Syntax for initialize gateway command

```
[net] init <bitrate>
```

The bit rate shall be given as table index of the standard CANopen bit rate table specified in Table 130.

Table 130 – Bit rate indices

Index	Bit rate
0	1 Mbit/s
1	800 kbit/s
2	500 kbit/s
3	250 kbit/s
4	125 kbit/s
5	reserved
6	50 kbit/s
7	20 kbit/s
8	10 kbit/s
9	Automatic bit rate detection

Response syntax:

See 10.5.2.4.3.

10.5.3.5.3 **Store configuration command**

All settings may be stored. Storage of settings may be selective to a special service. If no argument is given all settings shall be stored.

Indication syntax is defined in Table 131. The storage identifier is specified in Table 132.

Table 131 – Syntax for store configuration command

[net] store <storage-specifier>

Table 132 – Storage specifier

Storage specifier	Description
CFG	id, bitrate, default node, default net
PDO	PDO-number, transmission type, number of mapping, mapping
SDO	sdo timeout
NMT	- Node guarding: node, guarding time, life time factor - Heartbeat: node, heartbeat time - Heartbeat time of the server

Response syntax:

See 10.5.2.4.3.

10.5.3.5.4 **Restore configuration command**

Indication syntax is defined in Table 133.

Table 133 – Syntax restore configuration command

[net] restore <storage-specifier>

Response syntax:

See chapter 10.5.2.4.3.

10.5.3.5.5 **Set heartbeat producer command**

Indication syntax is defined in Table 134.

Table 134 – Syntax set heartbeat producer command

[net] set heartbeat <ms>

Response syntax:

See 10.5.2.4.3.

10.5.3.5.6 **Set node-ID command**

Indication syntax is defined in Table 135.

Table 135 – Syntax set node-ID command

[net] set id <value>

Response syntax:

See 10.5.2.4.3.

10.5.3.5.7 Start emergency consumer command

This command is not specified.

10.5.3.5.8 Stop emergency consumer command

This command is not specified.

10.5.3.6 Gateway management commands

10.5.3.6.1 General

The following command definitions shall be used to implement the gateway management services as defined in 10.4.6.

10.5.3.6.2 Set default network command

Indication syntax is defined in Table 136.

Table 136 – Syntax set default network command

[net] set network <value>

Response syntax:

See 10.5.2.4.3.

10.5.3.6.3 Set default node-ID command

Indication syntax is defined in Table 137.

Table 137 – Syntax set default node-ID command

[net] set node <value>

Response syntax:

See chapter 10.5.2.4.3.

10.5.3.6.4 Get version command

Indication syntax is defined in Table 138.

Table 138 – Syntax for get version command

info version

The response to the command is the current version information of the gateway as a whitespace separated list. The first list elements are values normally contained in the gateway's object dictionary at object 1018_h. The following listed elements contain the gateway class, as a combination of possible classes and the version number of the implemented protocol corresponded to 10.4.6.4.

Response syntax is defined in Table 139.

Table 139 – Response syntax for get version command

<version-string>::=	<vendor-id> <product-code> <version-high>.<version-low> <serial-number> <gateway-class> <protocol-version> <implementation-class>
---------------------	--

Result example is provided in Table 140.

Table 140 – Example for get version response

[1234]	52	100	1.01	1234567	128	0.85	0.10
--------	----	-----	------	---------	-----	------	------

10.5.3.7 Manufacturer-specific commands

The following command definitions shall be used to implement the manufacturer-specific services. Gateway manufacturer may add commands to their gateway to provide further features. In order to avoid syntax errors due to an unknown command all user-specific commands shall be prepended with an underscore “_”.

If there is the need for event-triggered messages not specified in this part of the specification the event specifier “USER” shall be used.

11 Train network management

11.1 Content

This clause extends the set of train network management (TNM) services for a WTB Node specified in IEC 61375-2-1. Train network management specifies services to assist commissioning, testing, operation and maintenance of a TCN. These services comprise station identification and control, distribution of routing and topography, remote reading and forcing variables, downloading and uploading, and management of the train network as well as of the consist network’s link layers. A Manager process requests these services remotely and each station through an Agent process executes them.

For informative reasons, the handling of management messages is illustrated in Figure 30, as defined in IEC 61375-2-1.

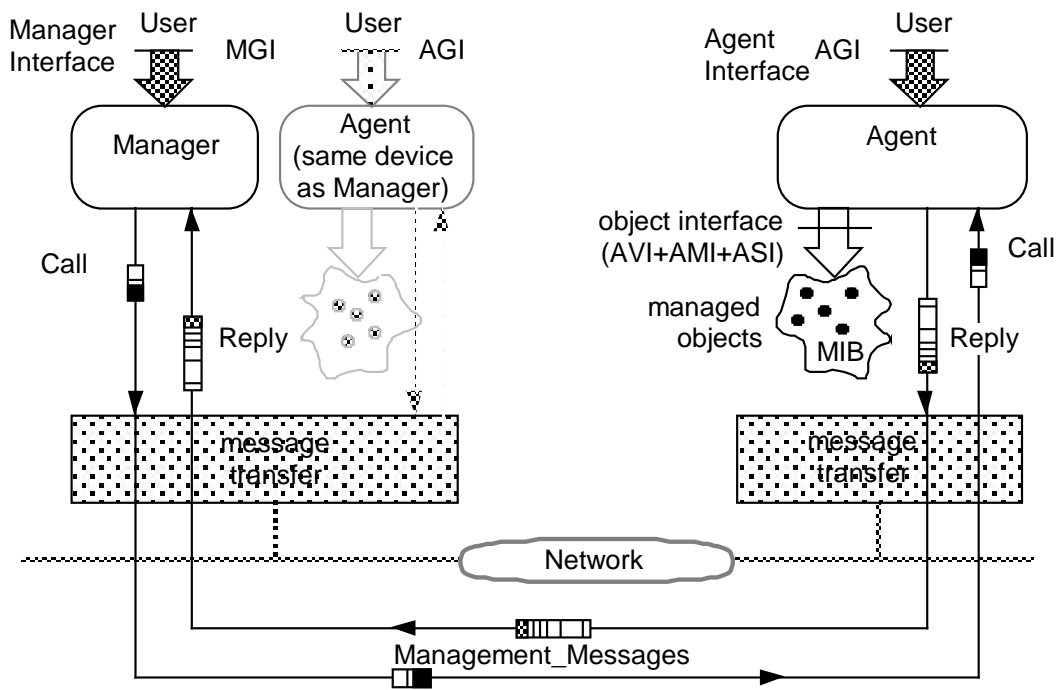


Figure 30 – Management messages (informative)

The additional services are specific for Stations connected to a CANopen consist network. A Manager shall request these services by sending a management Call_Message to the Agent in the target Station. The Agent shall execute the service and shall respond by means of a management Reply_Message with the result of the service. The CANopen-related services, which may be requested at the Agent, are specified in 10.4.

11.2 Manager, Agents and interfaces (informative)

As defined in IEC 61375-2-1 the Train Network Management services are provided in each Station by an Agent. The Station_Id of the Station on which it resides identifies the Agent. A Manager requests the Train Network Management services.

11.3 Management message protocol (informative)

For the purpose of Train Network Management, Manager and Agents communicate over the network by exchanging management messages, using the Messages Services of the Train Communication Network, as illustrated in Figure 30 and defined in IEC 61375-2-1. The Manager acts as a Caller and the Agent as a Replier.

The Manager accesses a remote object in two steps; the Manager sends a management Call_Message and the Agent decodes the message, accesses the actual object and sends back a management Reply_Message, with the result of the service.

11.4 Object interfaces (informative)

According to IEC 61375-2-1, TCN Communication objects are related to the network communication, while TCN non-communication objects are related to other properties of a Station.

NOTE1 Examples are provided in IEC 61375-2-1.

The Agent accesses communication objects through the interfaces defined for general access in this standard, and in particular through the:

- a) AVI (Application_Variables_Interface) for Variables;
- b) AMI (Application_Messages_Interface) for Messages;
- c) ASI (Application_Supervisory_Interface) for objects not accessible by user processes.

The ASI provides access to objects located in the different communication layers. The Agent accesses these objects through the Layer Management Interface of the layer in which they reside.

NOTE 2 The entity, which effectively accesses the objects in each layer, is called the Layer Management Entity, or LME.

The Agent shall also be able to access non-communication objects. The interface for doing this is not specified.

11.5 CANopen-specific management services

11.5.1 General

CANopen-specific management messages identify a service by two identifiers, the SIF_code and the Command_code. The SIF_code determines one of two groups of services – services that are executed with and without the reservation by the Manager for its exclusive use. The Command_code selects the particular service of the group. The payload of the call message is the request for a service coded as an ASCII string, the payload of the reply message is the response coded in the same way. The ASCII coding is specified in 10.5.

11.5.2 Agent interfaces on a Station connected to CANopen consist network

The definitions for the Agent interfaces, provided in IEC 61375-2-1 apply for CANopen devices, that communicate TCN message data, as well. The Agent interfaces are illustrated in Figure 31 for a gateway between the Train backbone and CANopen-based consist network. In Figure 31 the Train Backbone is implemented as WTB.

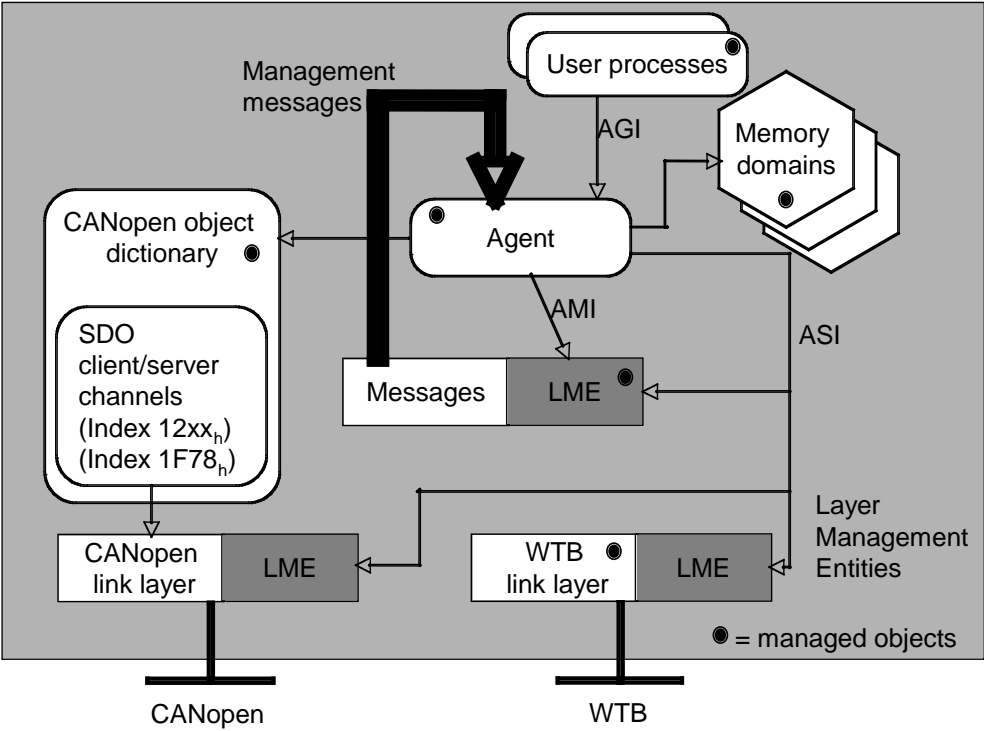


Figure 31 – Agent interface on a CANopen (gateway) station for message data

On a CANopen station the AVI is not used. The application data is directly represented in dedicated objects in the manufacturer specific as well as in the standardized CANopen object dictionary index range, as described in 8.2.

11.5.3 Management message structure for CANopen consist networks

Each service shall be invoked by a Call/Reply Management message exchange with the format defined in Table 141.

Table 141 – Management message structure

Management_Message_Can ::= RECORD			
{			
tnm_key	ENUM8 --	first octet	
{			
CALL	('02'H),	--	Call (request)
REPLY	('82'H)	--	Reply (response)
},			
message	ONE_OF [tnm_key]	--	selects call or reply
{			
[CALL]	Call_Mgt_Message_Can,	--	see 11.5.5
[REPLY]	Reply_Mgt_Message_Can	--	see 11.5.6
}			
}			

The most significant bit of 'tnm_key' shall indicate if this is a Call or a Reply message.

11.5.4 Notation for the CANopen specific SIF_codes

In management message services, the SIF_code indicates the requested services. In case of CANopen consist networks, the SIF_code indicates the group of requested CANopen service. As defined in Table 142, the least significant bit indicates if this is a read or a write (modifying) group of services. Read services may be used without previous reservation.

Table 142 – CANopen specific SIF_codes

```
Sif_Code_Can ::= ENUM8 -- choice of the group of services
{
  WRITE_CANOPEN_COMMAND          (91),
  READ_CANOPEN_COMMAND           (90)
}
```

11.5.5 Notation for a call CANopen management message

The notation for a call CANopen management message is defined in Table 143.

Table 143 – Notation for a call CANopen management message

```
Call_Mgt_Message_Can ::= RECORD
{
  sif_code Sif_Code_Can, -- the second octet is the
    sif_code
  message_body ONE_OF [sif_code]
  {
    [WRITE_CANOPEN_COMMAND] Call_Write_CANopen_Command,
    [READ_CANOPEN_COMMAND]  Call_Read_CANopen_Command
  }
}
```

11.5.6 Notation for a reply CANopen management message

The notation for a reply CANopen management message is defined in Table 144.

Table 144 – Notation for a reply CANopen management message

```
Reply_Mgt_Message_Can ::= RECORD
{
  sif_code Sif_Code_Can, -- the second octet is the
    sif_code
  message_body ONE_OF [sif_code]
  {
    [WRITE_CANOPEN_COMMAND] Reply_Write_CANopen_Command,
    [READ_CANOPEN_COMMAND]  Reply_Read_CANopen_Command
  }
}
```

11.5.7 Notation for the TNM CANopen services command codes

The Command code determines the particular service of the group of services identified by SIF_code. Table 145 and Table 146 define the TNM CANopen services command codes for services that require reservation or not.

Table 145 – TNM CANopen services command codes (reservation required)

Cmd_Code_R ::= ENUM8 -- choice of the service that needs reservation	
{	
UPLOAD_SDO	(1),
DOWNLOAD_SDO	(2),
CONFIGURE_SDO_TIMEOUT	(3),
CONFIGURE_RPDO	(4),
CONFIGURE_TPDO	(5),
WRITE_PDO_DATA	(6),
START_NODE	(7),
STOP_NODE	(8),
RESET_NODE	(9),
RESET_COMMUNICATION	(10),
ENABLE_NODE_GUARDING	(11),
DISABLE_NODE_GUARDING	(12),
START_HEARTBEAT_CONSUMER	(13),
DISABLE_HEARTBEAT_CONSUMER	(14),
INITIALIZE_GATEWAY	(15),
STORE_CONFIGURATION	(16),
RESTORE_CONFIGURATION	(17),
SET_HEARTBEAT_PRODUCER	(18),
SET_NODE_ID	(19),
SET_DEFAULT_NETWORK	(20),
SET_DEFAULT_NODE	(21)
}	

NOTE Notification services are not considered, as they cannot be implemented (request, response).

Table 146 – TNM CANopen services command codes (reservation not required)

Cmd_Code_NR ::= ENUM8 -- choice of the service that does not need reservation	
{	
READ_PDO_DATA	(30),
READ_DEVICE_ERROR	(31),
GET_VERSION	(32)
}	

11.6 TNM CANopen services

11.6.1 Content

This subclause defines the messages, identified by command code and specified by service command string transferred in the body of the message, by means of which the related TNM CANopen service shall be requested and responded.

11.6.2 Call_Write_CANopen_Command (with reservation)

By this message the services of the group “with reservation” (see Table 145) identified by the command code and specified by the service command string transferred in the body of the message shall be requested. Figure 32 provides the command structure and Table 147 provides the value definition.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
tnm_key								sif_code = 91							
reserved1								cmd_code							
reserved2															
string_size															
command: ARRAY ALIGN16 [string_size] OF															
(CHARACTER8)								CHARACTER8 or '00'H							

Figure 32 – Call_Write_CANopen_Command

Table 147 – Value definition for Call_Write_CANopen_Command

Call_Write_CANopen_Command ::= RECORD	
{	
reserved1 WORD8 (=0),	-- reserved
cmd_code Cmd_Code_R,	-- command code of the service
reserved2 WORD16 (=0),	-- reserved
string_size UNSIGNED16,	-- up to 65535 characters
command ARRAY ALIGN16 [string_size] OF CHARACTER8	-- service command string
}	

11.6.3 Reply_Write_CANopen_Command (with reservation)

By this message the Call_Write_CANopen command of the group “with reservation” (see 11.6.2) identified by the command code and specified by the service command string transferred in the body of the message shall be replied. Figure 33 provides the command structure and Table 148 provides the value definition.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
tnm_key								sif_code = 91							
reserved1								get_sif_code							
reserved2															
string_size															
response: ARRAY ALIGN16 [string_size] OF															
(CHARACTER8)								CHARACTER8 or '00'H							

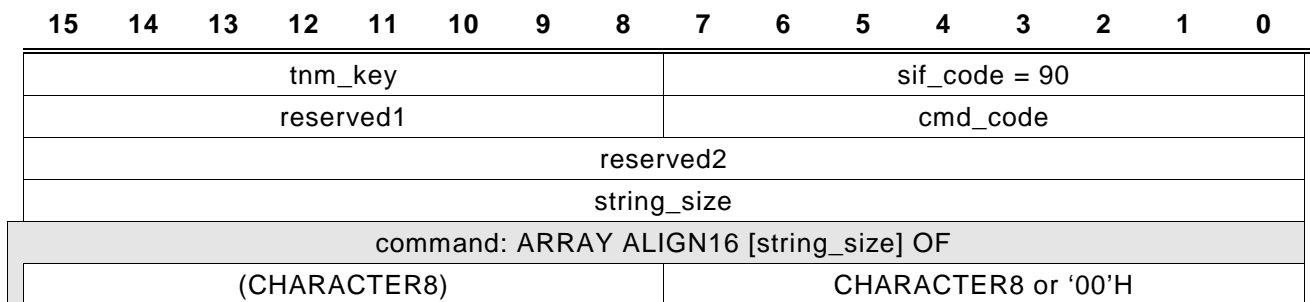
Figure 33 – Reply_Write_CANopen_Command

Table 148 – Value definition Reply_Write_CANopen_Command

Reply_Write_CANopen_Command ::= RECORD	
{	
reserved1 WORD8 (=0),	-- reserved
cmd_code Cmd_Code_R,	-- command code of the service
reserved2 WORD16 (=0),	-- reserved
string_size UNSIGNED16,	-- up to 65535 characters
response ARRAY ALIGN16 [string_size] OF CHARACTER8	-- service response string
}	

11.6.4 Call_Read_CANopen_Command (without reservation)

By this message the services of the group “without reservation” (see Table 146) identified by the command code and specified by the service command string transferred in the body of the message shall be requested. Figure 34 provides the command structure and Table 149 provides the value definition.

**Figure 34 – Call_Read_CANopen_Command (without reservation)****Table 149 – Value definition for Call_Read_CANopen_Command (without reservation)**

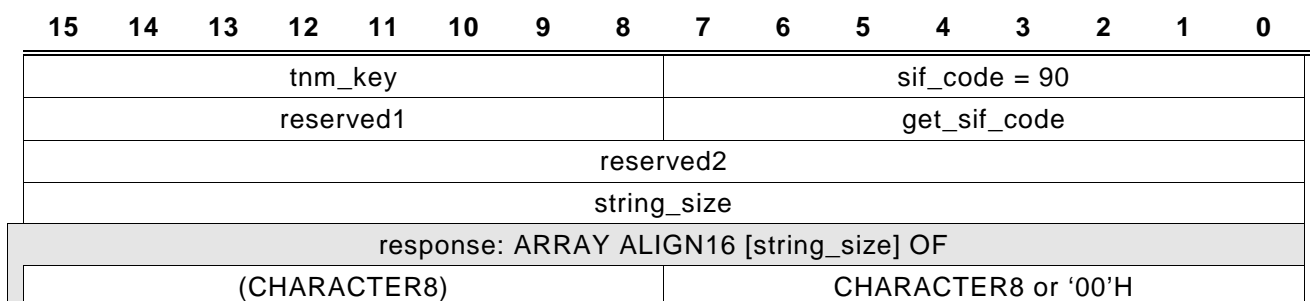
```

Call_Read_CANopen_Command ::= RECORD
{
  reserved1 WORD8 (=0),          -- reserved
  cmd_code Cmd_Code_NR,         -- command code of the service
  reserved2 WORD16 (=0),        -- reserved
  string_size UNSIGNED16, -- up to 65535 characters
  command ARRAY ALIGN16 [string_size] OF CHARACTER8 -- service
                           command string
}

```

11.6.5 Reply_Read_CANopen_Command (without reservation)

By this message the Call_Write_CANopen command of the group “without reservation” (see 11.6.4) identified by the command code and specified by the service command string transferred in the body of the message shall be replied. Figure 35 provides the command structure and Table 150 provides the value definition.

**Figure 35 – Reply_Read_CANopen_command (without reservation)****Table 150 – Value definition for Reply_Read_CANopen_Command (without reservation)**

```

Reply_Read_CANopen_Command ::= RECORD
{
  reserved1 WORD8 (=0),          -- reserved
  cmd_code Cmd_Code_NR,         -- command code of the service
  reserved2 WORD16 (=0),        -- reserved
  string_size UNSIGNED16, -- up to 65535 characters
  response ARRAY ALIGN16 [string_size] OF CHARACTER8 -- service
                           response string
}

```

12 CANopen management message data handling**12.1 General**

CANopen management messages may be also communicated between CANopen devices within a CANopen-based consist network. Figure 36 shows a CANopen device that is capable to communicate message data via the CANopen based consist network.

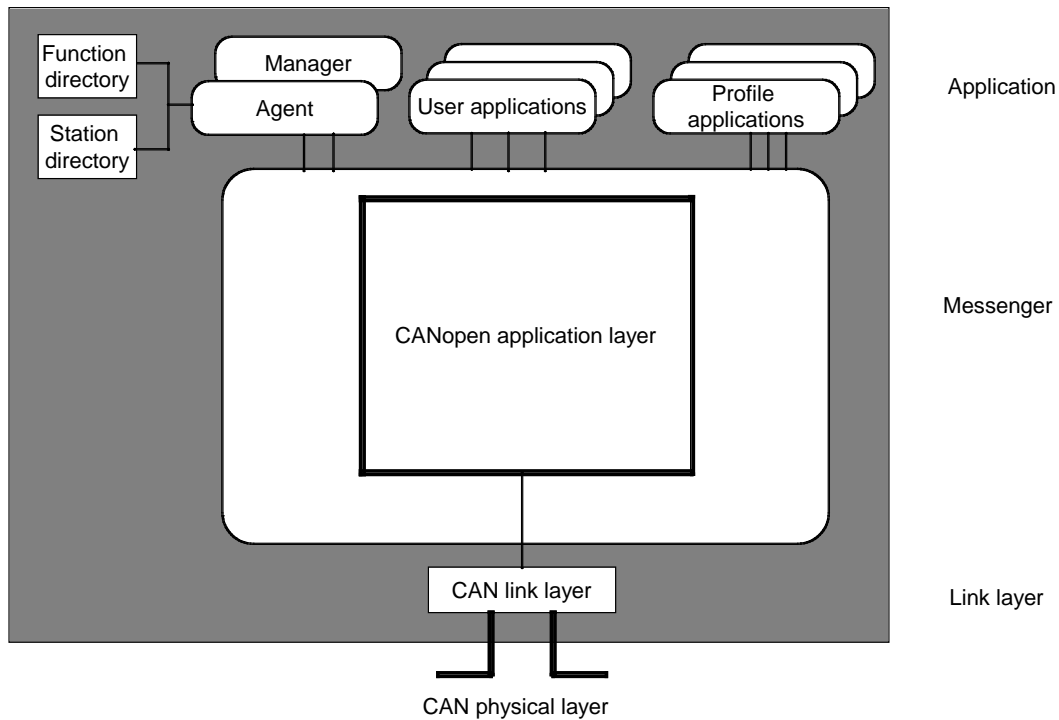


Figure 36 – CANOpen device capable to handle TNM management messages

In addition to user- and CANopen profile-specific applications the Manager- and Agent-application reside in such a CANopen device. In addition the Function- and Station-directory are supported within the CANopen device's application. For transmission and reception of message data, the CANopen device, as shown in Figure 36, supports SDO server and client channels and the CANopen object dictionary entry 1F78_h.

As defined and described in IEC 61375-2-1, Message Data is transmitted as datagrams. A datagram is similar to a letter: each datagram carries all addresses needed to route it from end to end (and send the acknowledgements back). This scheme is advantageous when several busses are interconnected, since the routers do not need to keep knowledge about previous messages.

Each Message Data frame carries two types of addresses: the source and destination Device Addresses for the communication within one bus (Device Addresses), and the addresses of the origin source and final destination (Network Addresses).

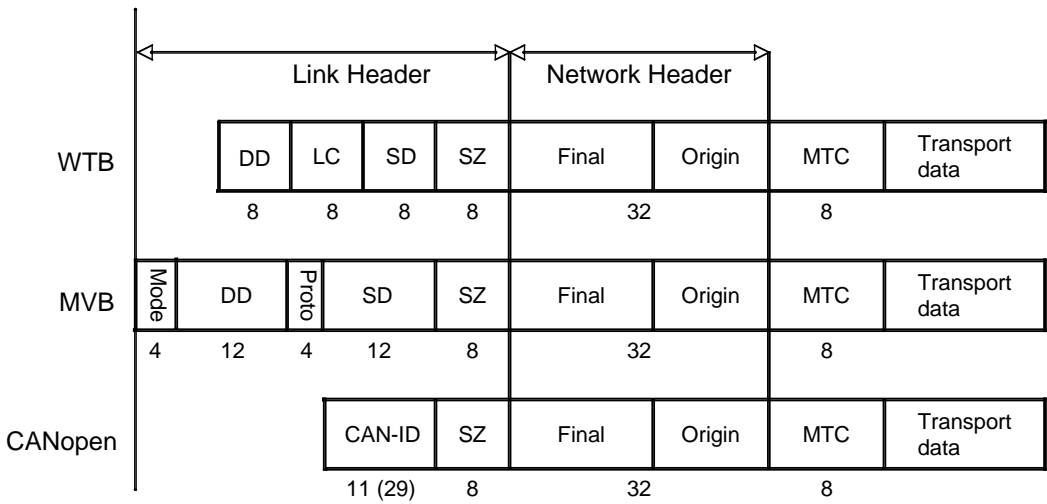
The source and destination Device Addresses apply only within the same bus. When Message Data is sent to another consist network, the destination device is the Node at the Train Backbone, which acts as Gateway. When the Node receives Message Data from another consist vehicle, it inserts its Device Address as source device.

Message Data carry two types of addresses:

- a source address and a destination address, which identify the devices communicating over the same bus and which are bus-specific. These addresses are specific to the Link Layer;
- an Origin Address and a Final Address, which identify the Stations communicating over the network and which are known in the whole network. The Origin Address and the Final Address identify the Producer and the Consumer, and in turn the Caller and the Replier. These addresses belong to the Network Layer.

12.2 Message data format

As defined in IEC 61375-2-1, with regard to Message Data, the frames on the WTB, the MVB, CANopen or any other bus system, shall only differ in their Link Header as shown in Figure 37.



DD = destination Device Address LC = Link Control SZ = Link Data Size
SD = source Device Address MTC Message Transport Control Proto = Protocol Type

CAN-ID = CAN Identifier; determined by selected CANopen SDO channel
SDO channel = Communication channel between DD and SD

Figure 37 – Message data format comparison

With regard to CANopen, the Link header comprises the CAN-Identifier. As message data shall be communicated via SDO – a confirmed point-to-point communication - the destination as well as the source device address is implicitly determined by the selection of the SDO communication channel. Each SDO communication channel provides, by means of the SDO parameter set, an unique CAN-Identifier for the request as well as for the response direction.

12.3 Requirements for message data communication within CANopen networks

Any CANopen device, connected to a CANopen-based consist network, which shall communicate Message Data, shall support SDO client as well as SDO server functionality.

Each CANopen device that shall call services at another device within the CANopen consist network, shall support a SDO client channel to call a service. By means of an SDO write access to object 1F78_h (see clause 12.4), via the SDO server channel of the “calling CANopen device”, the CANopen device that executes and replies to received message data shall transfer the reply.

Each CANopen device that shall receive a request of a service from another CANopen device within the CANopen consist network, shall support a SDO server channel to receive the request of the service. By means of an SDO write access via its SDO client channel this CANopen device shall transmit the response to the “calling CANopen device”.

NOTE It is recommended to support as many SDO client and -server channels in such a CANopen device, as device are in the CANopen network, that are capable to communicate message data. To each of these devices a pair of client/server channels is pre-configured.

The Message Data may be transferred either by SDO segmented transfer or SDO block transfer as defined in EN 50325-4. The message data shall be received within the appropriate sub-index of CANopen object dictionary index 1F78_h (see 12.4). As each sub-index of this object is related to one SDO server channel, the sub-index to be written shall be equal to the number of the utilized SDO server channel.

12.4 Object 1F78_h: CANopen message data reception

TCN Message Data shall be received via SDO write accesses to this object as defined in 12.3. Each sub-index of this object shall be related to a supported SDO server channel.

The data written to this object is of type domain and shall be interpreted as defined in IEC 61375-2-1 TCN message data.

Object and entry description are provided in Table 151 and Table 152.

Table 151 – Object description

Attribute	Value
Index	1F78 _h
Name	CANopen message data reception
Object code	ARRAY
Data type	DOMAIN
Category	Conditional; mandatory if message data communication is supported

Table 152 – Entry description

Attribute	Value
Sub-index	00 _h
Description	Number of entries
Entry category	Mandatory
Access	const
PDO mapping	No
Value range	01 _h to 80 _h
Default value	Device-specific
Sub-index	01 _h
Description	TCN message data received via SDO server channel 1 specified by Index 1200h
Entry category	Mandatory
Access	wo
PDO mapping	No
Value range	See Clause 11 and IEC 61375-2-1
Default value	Manufacturer specific

Attribute	Value
Sub-index	02 _h
Description	TCN message data received via SDO server channel 2 specified by Index 1201h
Entry category	Conditional; mandatory if message data communication is supported via SDO server channel 2
Access	wo
PDO mapping	No
Value range	See Clause 11 and IEC 61375-2-1
Default value	Manufacturer specific
to	
Sub-index	80 _h
Description	TCN message data received via SDO server channel 128 specified by Index 127Fh
Entry category	Conditional; mandatory if message data communication is supported via SDO server channel 128
Access	wo
PDO mapping	No
Value range	See Clause 11 and IEC 61375-2-1
Default value	Manufacturer specific

13 Conformance testing

The relevant controller-device interface standards shall specify the type tests required to verify compliance of the design of a device to this standard. The equipment to be tested shall include

- power supply,
- network device,
- controller,
- communication medium,
- electromechanics.

These tests shall include electrical tests, electromagnetic compatibility tests, and logical tests. For gateway devices between the Train Backbone and the CANopen-based consist network, all communication interfaces shall be tested.

The conformance test plan for WTB is provided in IEC 61375-2-2. The conformance test plan for CANopen devices is not in the scope of this standard.

NOTE 1 The CANopen conformance test plan is specified in the document CiA 310.

NOTE 2 Information on conformance testing services is offered at CAN in Automation (CiA) GmbH.

Bibliography

IEC 61375-2-4, *Electronic railway equipment – Train Communication Network (TCN) – Part 2-4: Application profile*

CiA 301, *CANopen application layer and communication profile*. CAN in Automation e.V., Nuremberg

CiA 302, *CANopen additional application layer functions*. CAN in Automation e.V., Nuremberg

CiA 305, *CANopen layer setting services (LSS) and protocols*. CAN in Automation e.V., Nuremberg

CiA 303-1, *CANopen additional specification – Part 1: Cabling and connector pin assignment*. CAN in Automation e.V., Nuremberg

CiA 310, *CANopen conformance test plan*. CAN in Automation e.V., Nuremberg

CiA 421, *CANopen application profile for train vehicle control networks*. CAN in Automation e.V., Nuremberg

RFC 2045; *RFC 2045 Multipurpose Internet mail extensions*. www.rfc.net

UIC 556, *Information transmission in trains (train bus)*. International Union of Railways (UIC), Brussels

SOMMAIRE

AVANT-PROPOS	118
INTRODUCTION.....	120
1 Domaine d'application	121
2 Références normatives	121
3 Termes, définitions et abréviations	122
3.1 Termes et définitions	122
3.2 Abréviations	123
3.3 Conventions	123
4 Architecture.....	123
4.1 Teneur	123
4.2 Réseau de rame logique basé sur CANopen	123
4.3 Topologie de réseau.....	124
4.4 Adressage	125
4.5 Classes de données	125
5 Couche physique.....	126
5.1 Teneur	126
5.2 Câblage	126
5.3 Connecteur	126
5.4 Fixation du support physique.....	127
5.5 Signalisation physique.....	128
6 Couche de Liaison de Données	128
6.1 Teneur	128
6.2 Couche de liaison de données CANopen.....	128
7 Couche d'application CANopen	129
7.1 Teneur	129
7.2 Modèle de référence	129
7.3 Modèle de dispositif de terrain.....	129
7.4 Objets de communication CANopen	131
7.5 Dictionnaire d'objets CANopen	131
7.6 Objets de communication CANopen prédéfinis	133
7.6.1 Teneur.....	133
7.6.2 Objet 1000 _h : Type de dispositif.....	133
7.6.3 Objet 1001 _h : Registre d'erreurs	134
7.6.4 Objet 1014 _h : Objet situation critique COB-ID	134
7.6.5 Objet 1017 _h : Producteur de pulsation (heartbeat)	134
7.6.6 Objet 1018 _h : Objet Identité	134
7.6.7 Objet 1029 _h : Comportement à l'erreur	134
7.6.8 Objet 67FF _h : Type de dispositif	134
7.6.9 Objets de données de service (SDO)	134
7.6.10 Objets de données de processus (PDO)	135
8 Données d'application	135
8.1 Teneur	135
8.2 Représentation des données d'application CANopen.....	135
8.3 Principe de représentation recommandé des données d'application	135
8.3.1 Teneur.....	135

8.3.2	Données d'application pour la commande des portes	135
8.3.3	Objets d'application des commandes des portes utilisées	136
8.3.4	Objets d'application des commandes des portes produites	137
9	Gestion de réseau CANopen	139
9.1	Teneur	139
9.2	Fonctionnalité Esclave NMT CANopen	140
9.3	Fonctionnalité Gestionnaire CANopen	140
9.3.1	Généralités.....	140
9.3.2	Emploi du dictionnaire d'objets	141
9.3.3	Réseaux redondants.....	141
9.4	Démarrage du NMT CANopen	142
9.4.1	Démarrage du NMT	142
9.4.2	Démarrage du NMT simple	148
9.4.3	Démarrage du processus d'initialisation de l'esclave NMT	149
9.5	Initialisation de l'esclave NMT	151
9.5.1	Vérification de la configuration	157
9.5.2	Vérification de l'état NMT	158
9.5.3	Démarrage du maître volant NMT	159
9.5.4	Etat d'erreur	159
9.6	Contrôle d'erreurs	161
9.6.1	Démarrage du contrôle d'erreurs	161
9.6.2	Gestionnaire de traitement d'erreur	162
9.6.3	Gestionnaire de traitement d'initialisation	163
9.7	Services et protocoles de maître NMT supplémentaires	164
9.8	Entrées du dictionnaire d'objets.....	164
9.8.1	Objet 1020 _h : Vérification de configuration.....	164
9.8.2	Objet 102A _h : Temps de blocage NMT	165
9.8.3	Objet 1F20 _h : Enregistrement DCF	166
9.8.4	Objet 1F22 _h : DCF concis	167
9.8.5	Objet 1F26 _h : Date de configuration prévue	169
9.8.6	Objet 1F27 _h : Heure de configuration prévue	170
9.8.7	Objet 1F80 _h : Démarrage du NMT	171
9.8.8	Objet 1F81 _h : Attribution de l'esclave NMT	173
9.8.9	Objet 1F82 _h : Requête NMT	176
9.8.10	Objet 1F83 _h : Requête de protection de nœud	178
9.8.11	Objet 1F84 _h : Identification du type de dispositif	180
9.8.12	Objet 1F85 _h : Identification du fournisseur	181
9.8.13	Objet 1F86 _h : Code produit.....	182
9.8.14	Objet 1F87 _h : Numéro de révision.....	183
9.8.15	Objet 1F88 _h : Numéro de série	185
9.8.16	Objet 1F89 _h : Temps d'initialisation	186
9.8.17	Objet 1F8A _h : Restauration de la configuration	186
9.8.18	Objet 1F91 _h : Paramètres de synchronisation de nœuds de démarrage automatique.....	187
10	Fonctions passerelle	189
10.1	Teneur	189
10.2	Architecture de passerelle	189
10.3	Principes généraux et services	191
10.3.1	Teneur.....	191

10.3.2	Définitions des classes de passerelle	191
10.3.3	Définitions des primitives de service	191
10.4	Spécification des services d'accès au réseau	191
10.4.1	Services d'accès SDO	191
10.4.2	Services d'accès PDO	193
10.4.3	Services NMT CANopen	196
10.4.4	Services de gestion de défaillance d'un dispositif	200
10.4.5	Services de configuration d'interface CANopen	200
10.4.6	Services de gestion de la passerelle.....	202
10.4.7	Services spécifiques au fabricant	203
10.5	Mise en correspondance ASCII des services d'accès au réseau	204
10.5.1	Teneur.....	204
10.5.2	Définitions	204
10.5.3	Spécifications des commandes d'accès au réseau.....	208
11	Gestion de réseau de train	216
11.1	Teneur	216
11.2	Gestionnaire, Agents et interfaces (informatif)	217
11.3	Protocole des messages de gestion (informatif)	217
11.4	Interfaces d'objets (informatif)	217
11.5	Services de gestion spécifiques CANopen.....	217
11.5.1	Généralités.....	217
11.5.2	Interfaces de l'Agent sur une Station connectée à un réseau de rame CANopen.....	218
11.5.3	Structure des messages de gestion pour les réseaux de rame CANopen.....	219
11.5.4	Notation pour les SIF_codes spécifiques CANopen	219
11.5.5	Notation pour un message de gestion CANopen d'appel.....	219
11.5.6	Notation pour un message de gestion CANopen de réponse.....	219
11.5.7	Notation pour les codes de commande de services Canopen TNM	220
11.6	Services CANopen TNM	221
11.6.1	Teneur.....	221
11.6.2	Call_Write_CANopen_Command (avec réservation)	221
11.6.3	Reply_Write_CANopen_Command (avec réservation)	221
11.6.4	Call_Read_CANopen_Command (sans réservation)	222
11.6.5	Reply_Read_CANopen_Command (sans réservation)	222
12	Traitement des données de messagerie de gestion CANopen	223
12.1	Généralités.....	223
12.2	Format des données de messagerie	224
12.3	Exigences relatives à la communication des données de messagerie dans les réseaux CANopen.....	226
12.4	Objet 1F78 _h : Réception des données de messagerie CANopen.....	226
13	Essais de conformité	227
	Bibliographie.....	229
	Figure 1 – Architecture de réseau logique du Réseau de Rame	124
	Figure 2 – Topologie de réseau du Réseau de Rame basé sur CANopen.....	125
	Figure 3 – Connecteur D-sub à 9 broches.....	126
	Figure 4 – Microconnecteur à 5 broches	127

Figure 5 – Modèle de dispositif de terrain	129
Figure 6 – Dispositif de terrain minimal	130
Figure 7 – Structure du dispositif CANopen	131
Figure 8 – Structure d'objet « Type de dispositif »	134
Figure 9 – Structure d'objet.....	136
Figure 10 – Structure d'objet.....	137
Figure 11 – Structure d'objet.....	138
Figure 12 – Démarrage NMT, partie 1	144
Figure 13 – Démarrage NMT, partie 2.....	147
Figure 14 – Démarrage du NMT simple.....	148
Figure 15 – Processus « Démarrer le processus d'initialisation de l'esclave NMT »	149
Figure 16 – Initialisation de l'esclave NMT, partie 1	152
Figure 17 – Initialisation de l'esclave NMT, partie 2	154
Figure 18 – Initialisation de l'esclave NMT, partie 3	156
Figure 19 – Vérification de la configuration	157
Figure 20 – Vérification de l'état NMT	159
Figure 21 – Démarrage du contrôle d'erreurs.....	161
Figure 22 – Gestionnaire de traitement d'erreur	163
Figure 23 – Gestionnaire de traitement d'initialisation	164
Figure 24 – Définition du flux de données d'un DCF concis.....	168
Figure 25 – Structure d'objet.....	171
Figure 26 – Structure de bit de la valeur de configuration	171
Figure 27 – Structure d'objet de la valeur	174
Figure 28 – Structure de bit de la valeur de configuration	174
Figure 29 – Passerelle entre le Réseau Central de Train et le réseau de rame CANopen	190
Figure 30 – Messages de gestion (informatif).....	216
Figure 31 – Interface de l'Agent sur une station (passerelle) CANopen pour les données de messagerie	218
Figure 32 – Call_Write_CANopen_Command.....	221
Figure 33 – Reply_Write_CANopen_Command	221
Figure 34 – Call_Read_CANopen_Command (sans réservation)	222
Figure 35 – Reply_Read_CANopen_Command (sans réservation)	222
Figure 36 – Dispositif CANopen capable de traiter des messages de gestion TNM	223
Figure 37 – Comparaison des formats de données de messagerie	225
 Tableau 1 – Brochage du connecteur D-sub à 9 broches	 127
Tableau 2 – Brochage du microconnecteur à 5 broches	127
Tableau 3 – Synchronisation des bits.....	128
Tableau 4 – Structure du dictionnaire d'objets CANopen.....	133
Tableau 5 – Définition de valeur	136
Tableau 6 – Description d'objet.....	136
Tableau 7 – Description d'entrée	137
Tableau 8 – Définition de valeur	137

Tableau 9 – Description d'objet.....	137
Tableau 10 – Description d'entrée	138
Tableau 11 – Définition de valeur.....	139
Tableau 12 – Description d'objet.....	139
Tableau 13 – Description d'entrée	139
Tableau 14 – Etat d'erreur	160
Tableau 15 – Description d'objet.....	165
Tableau 16 – Description d'entrée	165
Tableau 17 – Description d'objet.....	166
Tableau 18 – Description d'entrée	166
Tableau 19 – Description d'objet.....	166
Tableau 20 – Description d'entrée	167
Tableau 21 – Description d'objet.....	168
Tableau 22 – Description d'entrée	169
Tableau 23 – Description d'objet.....	169
Tableau 24 – Description d'entrée	170
Tableau 25 – Description d'objet.....	170
Tableau 26 – Description d'entrée	171
Tableau 27 – Valeur Maître NMT (bit: 0)	172
Tableau 28 – Valeur Démarrage de tous les nœuds (bit: 1).....	172
Tableau 29 – Valeur Démarrage du Maître NMT (bit: 2)	172
Tableau 30 – Valeur Démarrage du nœud (bit: 3)	172
Tableau 31 – Initialisation de tous les nœuds (bit: 4)	172
Tableau 32 – Maître volant (bit: 5)	172
Tableau 33 – Arrêt de tous les nœuds (bit: 6)	173
Tableau 34 – Exceptions pour les dispositifs capables de démarrage NMT	173
Tableau 35 – Description d'objet.....	173
Tableau 36 – Description d'entrée	173
Tableau 37 – Esclave NMT (bit: 0)	174
Tableau 38 – Esclave d'initialisation NMT (bit: 2)	174
Tableau 39 – Obligatoire (bit: 3)	174
Tableau 40 – Initialisation de la communication (bit: 4)	174
Tableau 41 – Version logicielle (bit: 5)	175
Tableau 42 – Mise à jour du logiciel (bit: 6).....	175
Tableau 43 – Restauration (bit: 7).....	175
Tableau 44 – Description d'objet.....	175
Tableau 45 – Description d'entrée	176
Tableau 46 – Définition de valeur.....	177
Tableau 47 – Description d'objet.....	177
Tableau 48 – Description d'entrée	178
Tableau 49 – Définition de valeur.....	179
Tableau 50 – Description d'objet.....	179
Tableau 51 – Description d'entrée	180

Tableau 52 – Description d'objet.....	181
Tableau 53 – Description d'entrée	181
Tableau 54 – Description d'objet.....	182
Tableau 55 – Description d'entrée	182
Tableau 56 – Description d'objet.....	183
Tableau 57 – Description d'entrée	183
Tableau 58 – Description d'objet.....	184
Tableau 59 – Description d'entrée	184
Tableau 60 – Description d'objet.....	185
Tableau 61 – Description d'entrée	185
Tableau 62 – Description d'objet.....	186
Tableau 63 – Description d'entrée	186
Tableau 64 – Description d'objet.....	187
Tableau 65 – Description d'entrée	187
Tableau 66 – Description d'objet.....	188
Tableau 67 – Description d'entrée	188
Tableau 68 – Service de téléchargement SDO amont	192
Tableau 69 – Paramètres de téléchargement SDO aval	193
Tableau 70 – Paramètres de configuration de temporisation SDO.....	193
Tableau 71 – Paramètres de configuration des services RPDO.....	194
Tableau 72 – Paramètres de configuration des services TPDO	195
Tableau 73 – Paramètres de service de lecture des données PDO	195
Tableau 74 – Paramètres de service d'écriture des données PDO	196
Tableau 75 – Paramètres de service des RPDO reçus.....	196
Tableau 76 – Paramètres de service de démarrage de nœud.....	196
Tableau 77 – Paramètres de service d'arrêt de nœud	197
Tableau 78 – Paramètres de service de mise du nœud à l'état préopérationnel	197
Tableau 79 – Paramètres de service d'initialisation de nœud	197
Tableau 80 – Paramètres de service d'initialisation de la communication.....	198
Tableau 81 – Paramètres de service d'activation de la protection de nœud	198
Tableau 82 – Paramètres de service de désactivation de la protection de nœud.....	199
Tableau 83 – Paramètres de service de démarrage du consommateur de pulsation.....	199
Tableau 84 – Paramètres de service de désactivation du consommateur de pulsation	199
Tableau 85 – Paramètres d'événement contrôle d'erreurs reçu	199
Tableau 86 – Paramètres de service de lecture d'erreur de dispositif.....	200
Tableau 87 – Paramètres de service d'événement situation critiquereçu.....	200
Tableau 88 – Paramètres de service d'initialisation de la passerelle	201
Tableau 89 – Paramètres de service d'enregistrement de configuration	201
Tableau 90 – Paramètres de service de restauration de configuration.....	201
Tableau 91 – Paramètres de service de réglage du producteur de pulsation	201
Tableau 92 – Paramètres de service de réglage de l'indicatif de nœud	202
Tableau 93 – Paramètres de service de démarrage du consommateur de messages de situation critique	202

Tableau 94 – Paramètres de service d'arrêt du consommateur de messages de de situation critique	202
Tableau 95 – Paramètres de service de réglage du réseau par défaut	203
Tableau 96 – Paramètres de service de réglage de l'indicatif de nœud par défaut	203
Tableau 97 – Paramètres de service d'obtention de la version	203
Tableau 98 – Syntaxe et types de données CANopen	205
Tableau 99 – Notation d'une commande au format FBN.....	206
Tableau 100 – Notation d'une réponse.....	207
Tableau 101 – Code d'erreurs internes (InEC)	207
Tableau 102 – Notation applicable aux messages déclenchés par un événement	207
Tableau 103 – Syntaxe d'une commande SDO de téléchargement amont	208
Tableau 104 – Exemples d'une commande SDO de téléchargement amont	208
Tableau 105 – Syntaxe d'une commande SDO de téléchargement aval	208
Tableau 106 – Exemples d'une commande SDO de téléchargement aval.....	208
Tableau 107 – Syntaxe de configuration de la commande de temporisation SDO.....	208
Tableau 108 – Syntaxe de configuration de la commande RPDO	209
Tableau 109 – Exemples de configuration de la commande RPDO	209
Tableau 110 – Syntaxe de configuration de la commande TPDO	209
Tableau 111 – Exemples de configuration de la commande TPDO.....	209
Tableau 112 – Syntaxe de la commande de lecture des données PDO	210
Tableau 113 – Syntaxe de réponse pour la commande de lecture des données PDO.....	210
Tableau 114 – Syntaxe de la commande d'écriture des données PDO	210
Tableau 115 – Syntaxe de la commande RPDO reçu	210
Tableau 116 – Exemples de commande RPDO reçu	210
Tableau 117 – Syntaxe de la commande de démarrage de nœud	210
Tableau 118 – Syntaxe de la commande d'arrêt de nœud.....	211
Tableau 119 – Syntaxe de la commande de mise du nœud à l'état préopérationnel	211
Tableau 120 – Syntaxe de la commande d'initialisation de nœud.....	211
Tableau 121 – Syntaxe de la commande d'initialisation de la communication.....	211
Tableau 122 – Syntaxe de la commande d'activation de la protection de nœud	211
Tableau 123 – Syntaxe de la commande de désactivation de la protection de nœud.....	211
Tableau 124 – Syntaxe de la commande de démarrage du consommateur de pulsation.....	212
Tableau 125 – Syntaxe de la commande de désactivation du consommateur de pulsation.....	212
Tableau 126 – Syntaxe de la commande Événement contrôle d'erreurs reçu	212
Tableau 127 – Syntaxe de la commande de lecture d'erreur de dispositif.....	212
Tableau 128 – Syntaxe de la commande Événement situation critique reçu	212
Tableau 129 – Syntaxe de la commande d'initialisation de la passerelle	213
Tableau 130 – Index de débit binaire	213
Tableau 131 – Syntaxe de la commande d'enregistrement de la configuration	213
Tableau 132 – Spécificateur d'enregistrement	214
Tableau 133 – Syntaxe de la commande de restauration de la configuration.....	214
Tableau 134 – Commande de réglage du producteur de pulsation	214
Tableau 135 – Syntaxe de la commande de réglage de l'indicatif de nœud	214

Tableau 136 – Syntaxe de la commande de réglage du réseau par défaut	215
Tableau 137 – Syntaxe de la commande de réglage de l'indicatif de nœud par défaut	215
Tableau 138 – Syntaxe de la commande d'obtention de la version.....	215
Tableau 139 – Syntaxe de réponse de la commande d'obtention de la version	215
Tableau 140 – Exemple de réponse de la commande d'obtention de la version	215
Tableau 141 – Structure des messages de gestion	219
Tableau 142 – SIF_codes spécifiques CANopen.....	219
Tableau 143 – Notation pour un message de gestion CANopen d'appel.....	219
Tableau 144 – Notation pour un message de gestion CANopen de réponse.....	220
Tableau 145 – Codes de commande de services CANopen TNM (réservation nécessaire)	220
Tableau 146 – Codes de commande de services CANopen TNM (réservation non nécessaire)	220
Tableau 147 – Définition de valeur pour Call_Write_CANopen_Command	221
Tableau 148 – Définition de valeur pour Reply_Write_CANopen_Command	222
Tableau 149 – Définition de valeur pour Call_Read_CANopen_Command (sans réservation)	222
Tableau 150 – Définition de valeur pour Reply_Read_CANopen_Command (sans réservation)	223
Tableau 151 – Description d'objet.....	226
Tableau 152 – Description d'entrée.....	227

COMMISSION ÉLECTROTECHNIQUE INTERNATIONALE

MATÉRIEL ÉLECTRONIQUE FERROVIAIRE – RÉSEAU EMBARQUÉ DE TRAIN (TCN) –

Partie 3-3: Réseau de rame CANopen (CCN)

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.

La Norme internationale CEI 61375-3-3 a été établie par le comité d'études 9 de la CEI: Matériels et systèmes électriques ferroviaires.

Le texte de cette norme est issu des documents suivants:

FDIS	Rapport de vote
9/1646/FDIS	9/1670/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 61375, présentées sous le titre général *Matériel électronique ferroviaire – Réseau embarqué de train (TCN)*, 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.

INTRODUCTION

Le TCN est une Norme internationale dont l'objet est de définir les interfaces permettant d'obtenir la compatibilité de connexion:

- a) entre des équipements situés dans des véhicules ou des rames différents, et
- b) entre des équipements et dispositifs situés à l'intérieur du même véhicule ou de la même rame.

Le succès du déploiement d'une technologie repose en partie sur la normalisation et l'interopérabilité des différentes mises en œuvre. Pour faciliter l'interopérabilité, il convient d'effectuer un essai de conformité.

Dans la présente partie de la CEI 61375, le TCN traite:

du Réseau de Rame basé sur CANopen.

De plus, les passerelles situées entre le réseau central de train et le Réseau de Rame basé sur CANopen sont prises en considération.

La présente norme est structurée en 13 articles.

MATÉRIEL ÉLECTRONIQUE FERROVIAIRE – RÉSEAU EMBARQUÉ DE TRAIN (TCN) –

Partie 3-3: Réseau de rame CANopen (CCN)

1 Domaine d'application

La présente partie de la CEI 61375 spécifie les bus de communication de données à l'intérieur des rames basés sur CANopen. CANopen a été développé pour une utilisation dans les applications d'automatisation industrielle, sans toutefois s'y limiter. Ces applications peuvent inclure des dispositifs tels que les modules d'entrée/sortie, contrôleurs de mouvement, interfaces homme-machine, capteurs, contrôleurs en boucle fermée, codeurs, valves hydrauliques ou contrôleurs programmables.

Dans le domaine d'application des véhicules ferroviaires, les réseaux CANopen sont utilisés pour les sous-systèmes de réseaux de rame tels que, par exemple, système de commande de frein, système de commande de moteurs diesels et système de commande d'éclairage intérieur ou extérieur. De plus, CANopen est utilisé comme Réseau de Rame destiné à permettre l'échange de données entre les différents sous-systèmes d'un seul véhicule ferroviaire ou d'un groupe de véhicules ferroviaires qui partagent le même Réseau de Rame.

La présente partie de la CEI 61375 s'applique à tous les équipements et dispositifs utilisés sur un Réseau de Rame basé sur CANopen au sein d'une architecture TCN comme décrit dans la CEI 61375-1.

L'applicabilité de la présente norme à la mise en œuvre d'un TCN permet de procéder à une vérification de conformité individuelle de la mise en œuvre elle-même et constitue une condition préalable à un contrôle approfondi de l'interopérabilité des différentes mises en œuvre du TCN. Dans tous les cas, le fournisseur aura à faire la preuve de la compatibilité entre le Réseau Central de Train et le Réseau de Rame.

La présente partie de la CEI 61375 s'applique à l'architecture des systèmes de communication des trains à composition variable. De plus, elle peut s'appliquer aux trains indéformables et aux trains à unité multiple après accord entre acheteur et fournisseur.

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).

CEI 61131 (toutes les parties), *Automates programmables*

CEI 61375-1, Ed.3, *Matériel électronique ferroviaire – Réseau embarqué de train (TCN) – Partie 1: Architecture générale*

CEI 61375-2-1, *Matériel électronique ferroviaire – Réseau embarqué de train (TCN) – Partie 2-1: Bus de train filaire (WTB)*

CEI 61375-2-2, *Matériel électronique ferroviaire – Réseau embarqué de train (TCN) – Partie 2-2: Bus de train filaire – Essais de conformité*

ISO/CEI 646:1991, *Technologies de l'information – Jeu ISO de caractères codés à 7 éléments pour l'échange d'informations*

ISO/CEI 9899:1999, *Langages de programmation – C*

ISO 11898-1:2003, *Véhicules routiers – Gestionnaire de réseau de communication (CAN) – Partie 1: Couche liaison de données et signalisation physique*

ISO 11898-2:2003, *Véhicules routiers – Gestionnaire de réseau de communication (CAN) – Partie 2: Unité d'accès au support à grande vitesse*

EN 50325-4:2002, *Sous-systèmes de communication industriels basés sur l'ISO 11898 (CAN) pour les interfaces des dispositifs de commande – Partie 4: protocole CANopen*

3 Termes, définitions et abréviations

3.1 Termes et définitions

Pour les besoins du présent document, les termes et définitions donnés dans l'ISO 11898-1, l'ISO 11898-2, la CEI 61375-1, la CEI 61375-2-1, la EN 50325-4 s'appliquent, ainsi que les suivants.

3.1.1

ASCII

jeu de caractères codés à 7 éléments conformément à l'ISO/CEI 646

3.1.2

dispositif CANopen

dispositif terminal relié à un Réseau de Rame basé sur CANopen

3.1.3

dispositif de terrain

entité physique indépendante mise en réseau d'un système d'automatisation capable d'exécuter des fonctions spécifiées dans un contexte particulier et délimité par ses interfaces

3.1.4

Réseau de Rame logique

moyen de circulation des données dans le Réseau de Rame sans tenir compte de l'interconnexion physique des Dispositifs Terminaux

3.1.5

dispositif logique

représentation d'un dispositif de terrain en termes de ses objets et de son comportement selon un modèle de dispositif de terrain qui décrit les données et le comportement du dispositif observés par l'intermédiaire d'un réseau

3.1.6

services de Stratification

services permettant d'ajuster le débit binaire et l'indicatif de nœud via l'interface de communication du dispositif CANopen

3.1.7

Indicatif de Nœud

a) indicatif unique sur l'ensemble du réseau pour chaque dispositif CANopen

b) adresse de dispositif codée à 7 éléments dans les réseaux de rame basés sur CANopen

3.1.8**Objet**

entité à limite bien définie et dont l'identité encapsule l'état et le comportement

3.1.9**équipement virtuel**

élément de logiciel capable d'accomplir un élément fonctionnel d'un dispositif de terrain

3.2 Abréviations

Pour les besoins du présent document, les abréviations données dans l'ISO 11898-1, l'ISO 11898-2, la CEI 61375-1, la CEI 61375-2-1, la EN 50325-4 s'appliquent, ainsi que les suivantes.

ASCII	American Standard Code for Information Interchange (Code américain normalisé pour l'échange d'information)
FBN	Forme de Backus Naur
CAN	Controller Area Network (Gestionnaire de réseau de communication)
UC	Unité centrale
CR	Carriage Return (Retour de chariot)
CRLF	Carriage Return and Line Feed (Retour de chariot et interligne)
DCF	Device configuration file (Fichier de configuration des dispositifs)
FSA	Finite state automaton (Automate d'états finis)
ID	Indicatif
LF	Line Feed (Interligne)
LSS	Layer Setting Services (Services de stratification)
NMT	Network management (Gestion de réseau)
OSI	Open system interconnect (Interconnexion de systèmes ouverts)
PAS	PDO access service (Service d'accès PDO)
PDO	Process data object (Objet de données de processus)
SAS	Service data object access service (Service d'accès d'objet de données de service)
SRD	SDO requesting device (Dispositif de requête SDO)

3.3 Conventions

Les conventions données dans la CEI 61375-1 doivent s'appliquer.

4 Architecture**4.1 Teneur**

Le présent article donne la définition de l'architecture des Réseaux de Rame basés sur CANopen.

4.2 Réseau de rame logique basé sur CANopen

Le Réseau de Rame logique global basé sur CANopen, qui relie plusieurs Equipements Virtuels, est illustré à la Figure 1. Le Réseau de Rame basé sur CANopen relie entre eux plusieurs Equipements Virtuels et sous-systèmes, tels que, par exemple Système de contrôle du train (TOS), système de contrôle et de sécurité (MSS), système d'exploitation auxiliaire (AUX), système d'entraînement (PDS), système d'organes de roulement (RGS), système de commande de frein (BCS), système d'exploitation accessoires (ANC), système de liaison de véhicule (VLS), système d'éclairage extérieur (ELS), système d'éclairage intérieur (ILS),

système de commande de portes (DCS), système de chauffage, ventilation et climatisation (HS), système d'information aux voyageurs (PIS), système de diagnostic (DS) ou système de communication train-sol (TCS).

Le Réseau de Rame logique basé sur CANopen est connecté au Réseau Central de Train par l'intermédiaire d'une Passerelle. La Passerelle gère l'échange d'informations ainsi que la sérialisation des données de processus entre le Réseau Central de Train et le Réseau de Rame.

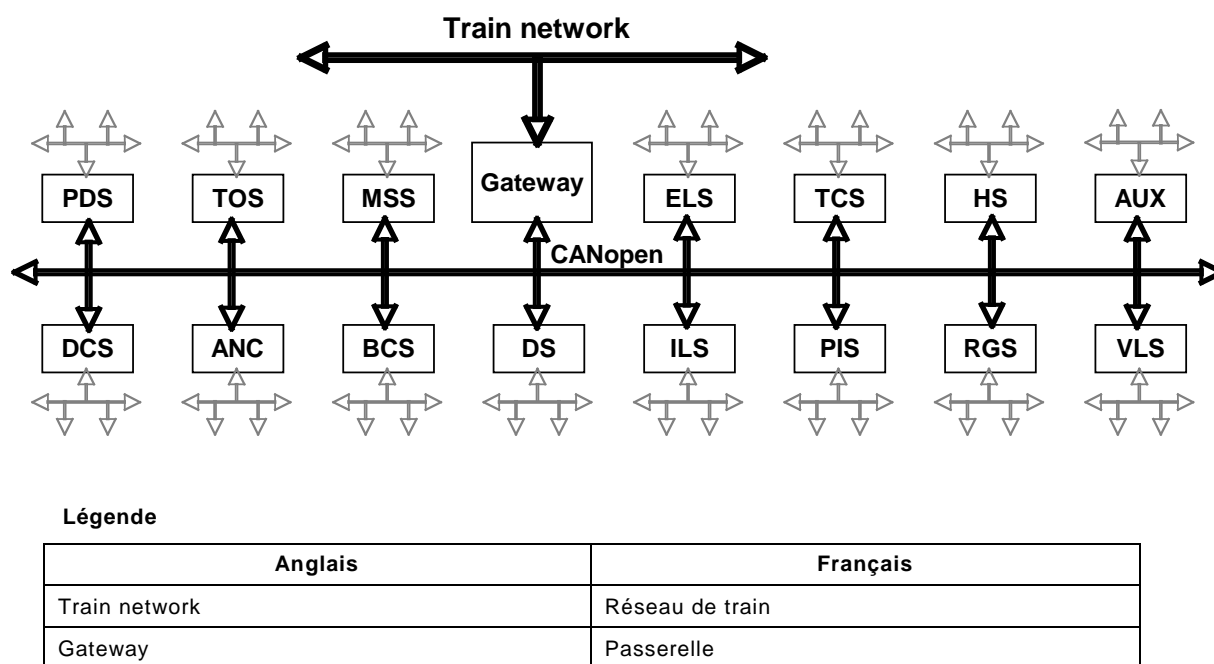


Figure 1 – Architecture de réseau logique du Réseau de Rame

4.3 Topologie de réseau

La topologie de réseau doit être le plus proche possible d'une structure à ligne simple avec présence de résistances de terminaison des deux côtés du réseau comme le suggère l'ISO 11898-2. La longueur totale de réseau ne doit pas dépasser 450 m, si une vitesse de transmission de 125 kbit/s est utilisée.

La longueur cumulée des branches ne doit pas dépasser 110 m, et la longueur de branche simple ne doit pas dépasser 22 m, dans le cas d'une vitesse de transmission de 125 kbit/s. Il est recommandé que les longueurs de branche soient les plus courtes possible. Les puces de l'émetteur-récepteur CAN doivent être isolées galvaniquement. Des coupleurs optoélectroniques sont placés entre le contrôleur et l'émetteur-récepteur CAN. Ceci affecte la longueur de bus maximale selon le délai de propagation des coupleurs optoélectroniques. La résistance de terminaison placée aux deux extrémités des lignes du bus doit être égale ou supérieure à 120 Ω .

La topologie de réseau est illustrée à la Figure 2.

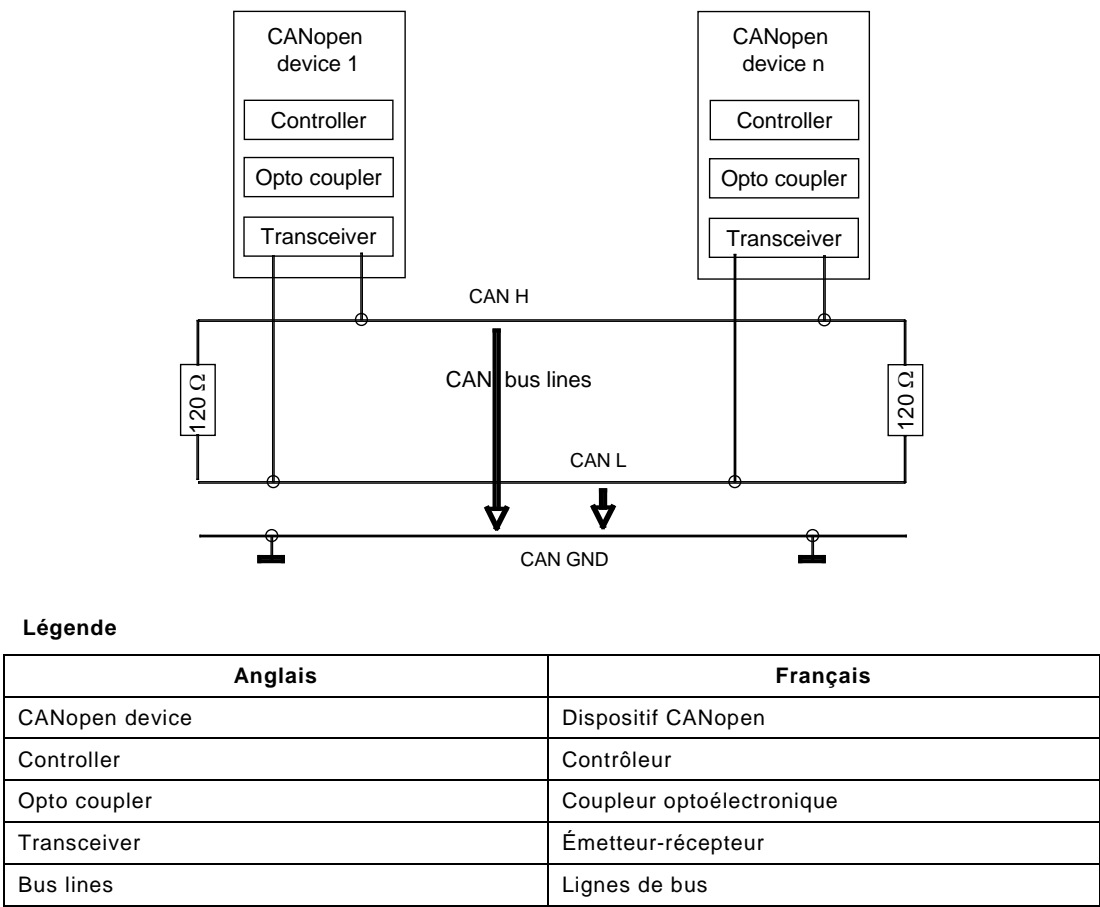


Figure 2 – Topologie de réseau du Réseau de Rame basé sur CANopen

4.4 Adressage

Tout Dispositif Terminal connecté à un Réseau de Rame basé sur CANopen nécessite un indicatif de nœud CANopen unique dans la plage comprise entre 01_h et 7F_h. L'indicatif de nœud CANopen est ajusté par l'intermédiaire de commutateurs matériels ou d'un logiciel. Les Services de Stratification CANopen doivent être utilisés en cas d'ajustement de l'indicatif de nœud CANopen via l'interface de communication CANopen du dispositif. L'indicatif de nœud CANopen ne doit pas être ajusté via le dictionnaire d'objets CANopen du dispositif associé.

La définition des Services de Stratification (LSS) ne relève pas du domaine d'application de la présente norme.

NOTE Les LSS sont spécifiés dans le document CiA 305.

4.5 Classes de données

Il est nécessaire, pour pouvoir échanger des données significatives dans le Réseau de Rame basé sur CANopen, que le format et la signification de ces données soient connus par le producteur et le(s) consommateur(s). La présente spécification modélise cette nécessité par le concept des types de données.

Les règles de codage définissent la représentation des valeurs des types de données et la syntaxe de transfert applicable aux représentations. Les valeurs sont représentées sous forme de séquences de bits. Les séquences de bits sont transférées en séquences d'octets. Le codage est de type little-endian (petit boutiste) pour les types de données numériques.

Les applications exigent souvent des types de données autres que les types de données de base. Le mécanisme de type de données composées permet d'étendre la liste des types de données disponibles. Certains types de données étendues générales sont définis comme « Chaîne Visible » ou « Heure ». Les types de données composées constituent un moyen de mise en œuvre des "DEFTYPES" définis par l'utilisateur dans la terminologie de la EN 50325-4 et non les "DEFSTRUCTS".

Les types de données et les règles de codage fournis dans la EN 50325-4 doivent s'appliquer.

5 Couche physique

5.1 Teneur

Le présent article donne les définitions afférentes à la couche physique d'un réseau de rame basé sur CANopen.

5.2 Câblage

Le circuit tronc principal doit être au moins une paire de câbles torsadés d'impédance caractéristique nominale comme défini dans l'ISO 11898-2. De plus, une ligne CAN_Ground doit être appliquée comme niveau de référence clairement établi pour le niveau de tension CAN_H et CAN_L.

5.3 Connecteur

Il est recommandé que les dispositifs connectés à un réseau de rame basé sur CANopen prennent en charge le connecteur D-sub à 9 broches ou le microconnecteur à 5 broches (M12). Les connecteurs sont illustrés aux Figures 3 et 4. Le connecteur D-sub à 9 broches doit prendre en charge un brochage comme défini dans le Tableau 1. Le microconnecteur à 5 broches (M12) doit prendre en charge un brochage comme défini dans le Tableau 2. Dans la mesure où les dispositifs CANopen doivent être isolés galvaniquement, la ligne V+ facultative doit être utilisée uniquement pour l'alimentation de l'émetteur-récepteur et du coupleur optoélectronique CAN, en l'absence de toute alimentation supplémentaire.

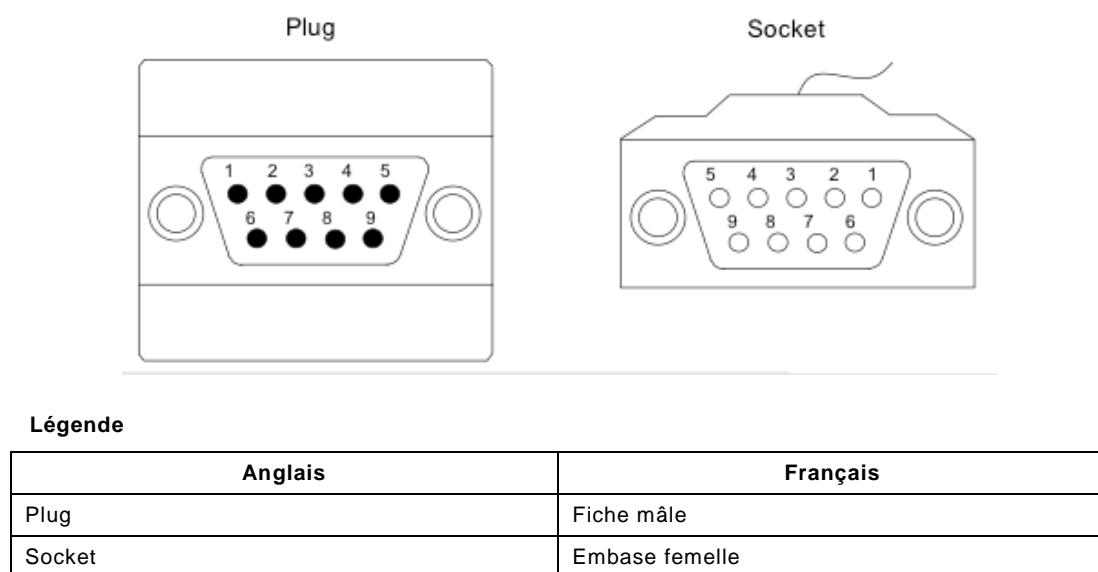


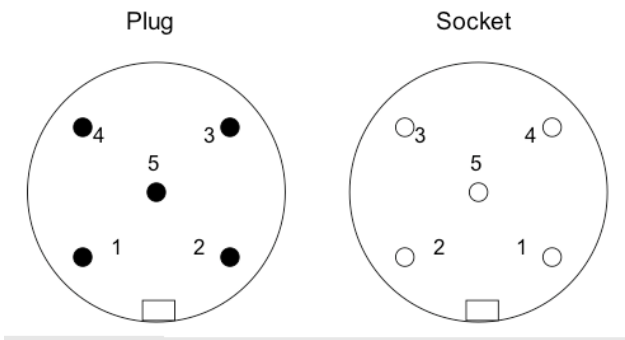
Figure 3 – Connecteur D-sub à 9 broches

Tableau 1 – Brochage du connecteur D-sub à 9 broches

Broche	Signal	Description
1	-	Réservée
2	CAN_L	Ligne de bus CAN_L (dominante faible)
3	CAN_GND	Masse CAN
4	-	Réservée
5	(CAN_SHLD)	Blindage CAN facultatif
6	(GND)	Masse facultative
7	CAN_H	Ligne de bus CAN_H (dominante élevée)
8	-	Réservée
9	(CAN_V+)	Alimentation positive CAN externe facultative (dédiée à l'alimentation de l'émetteur-récepteur et des coupleurs optoélectroniques pour les nœuds de bus à isolation galvanique)

NOTE Il est fortement recommandé de ne pas connecter CAN_GND et GND.

Il est recommandé, si une protection par blindage est exigée, de relier un blindage par l'intermédiaire du boîtier métallique du connecteur utilisé. La broche CAN_SHLD peut également être utilisée.



Légende

Anglais	Français
Plug	Fiche mâle
Socket	Embase femelle

Figure 4 – Microconnecteur à 5 broches

Tableau 2 – Brochage du microconnecteur à 5 broches

Broche	Signal	Description
1	(CAN_SHLD)	Blindage CAN facultatif
2	(CAN_V+)	Alimentation positive CAN externe facultative (dédiée à l'alimentation de l'émetteur-récepteur et des coupleurs optoélectroniques, si l'isolation galvanique du nœud de bus s'applique)
3	CAN_GND	Masse / 0V / V-
4	CAN_H	Ligne de bus CAN_H (dominante élevée)
5	CAN_L	Ligne de bus CAN_L (dominante faible)

5.4 Fixation du support physique

Le support physique d'un dispositif connecté à des réseaux de rame basés sur CANopen doit être une ligne de bus à deux fils séparés avec un retour commun selon la spécification de transmission à grande vitesse définie dans l'ISO 11898-2.

Les caractéristiques assignées maximales de V_{CAN_H} et V_{CAN_L} doivent être de +16 V si l'on utilise l'émetteur-récepteur à grande vitesse conformément à l'ISO 11898-2. L'isolation galvanique entre les dispositifs CANopen est facultative. Il est recommandé d'utiliser un émetteur-récepteur CAN capable de résister à tout défaut de connexion de l'un des fils du connecteur, y compris les tensions V+ facultatives jusqu'à 30 V.

5.5 Signalisation physique

Le codage/décodage et la synchronisation des bits doivent satisfaire aux exigences définies dans l'ISO 11898-1.

La synchronisation des bits doit satisfaire aux exigences définies dans l'ISO 11898-1 ainsi qu'aux définitions données dans le Tableau 3. L'emplacement recommandé du point échantillon est le plus proche possible de 87,5 % du temps de bit associé.

Tableau 3 – Synchronisation des bits

Débit binaire	Temps de bit nominal t_b µs	Plage valide d'emplacement du point échantillon %
1 Mbit/s	1	75 à 90
800 kbit/s	1,25	75 à 90
500 kbit/s	2	85 à 90
250 kbit/s	4	85 à 90
125 kbit/s	8	85 à 90
50 kbit/s	20	85 à 90
20 kbit/s	50	85 à 90
10 kbit/s	100	85 à 90

Les dispositifs connectés à un réseau de rame basé sur CANopen doivent prendre en charge un débit binaire de 125 kbit/s. D'autres débits binaires donnés dans le Tableau 3 peuvent également être pris en charge.

6 Couche de Liaison de Données

6.1 Teneur

Le présent article donne les définitions afférentes à la couche de liaison de données d'un réseau de rame basé sur CANopen.

6.2 Couche de liaison de données CANopen

Le réseau de rame basé sur CANopen décrit doit être basé sur une couche de liaison de données et ses sous-couches conformément à l'ISO 11898-1.

Cette spécification est basée sur le format de trame de base CAN avec un indicatif CAN de 11 bits. Il n'est par conséquent pas nécessaire de prendre en charge le format de trame étendu CAN avec un indicatif CAN de 29 bits.

NOTE Dans la mesure où certaines applications exigent l'utilisation du format de trame étendu CAN, le réseau peut être exploité également dans ce mode, dans le cas où tous les dispositifs CANopen connectés prennent en charge ce format.

7 Couche d'application CANopen

7.1 Teneur

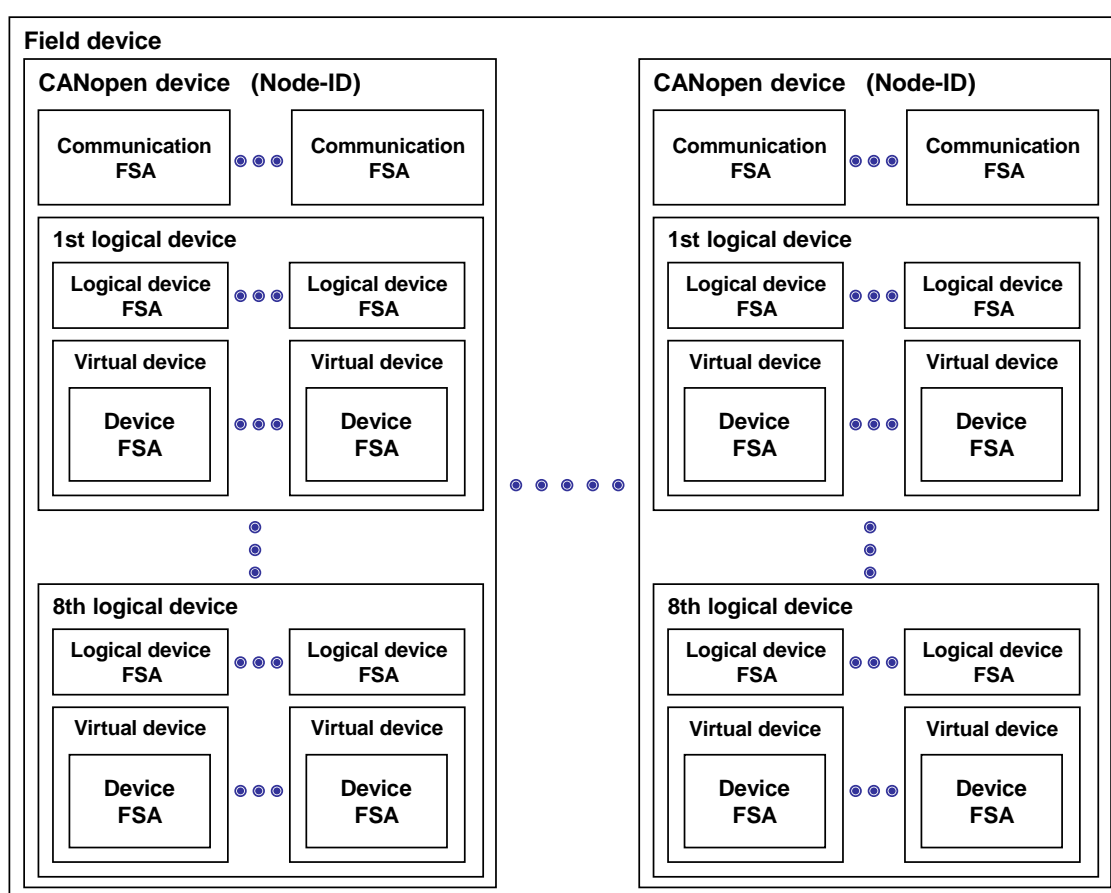
Le présent article donne les définitions concernant les couches supérieures du modèle de référence OSI de l'ISO pour les dispositifs connectés à un réseau de rame basé sur CANopen.

7.2 Modèle de référence

Les dispositifs connectés à un réseau de rame basé sur CANOpen conforme à la présente spécification doivent utiliser le modèle de référence, comme défini dans la EN 50325-4.

7.3 Modèle de dispositif de terrain

Les dispositifs connectés à un réseau de rame basé sur CANopen doivent être conformes au modèle de dispositif de terrain donné à la Figure 5.

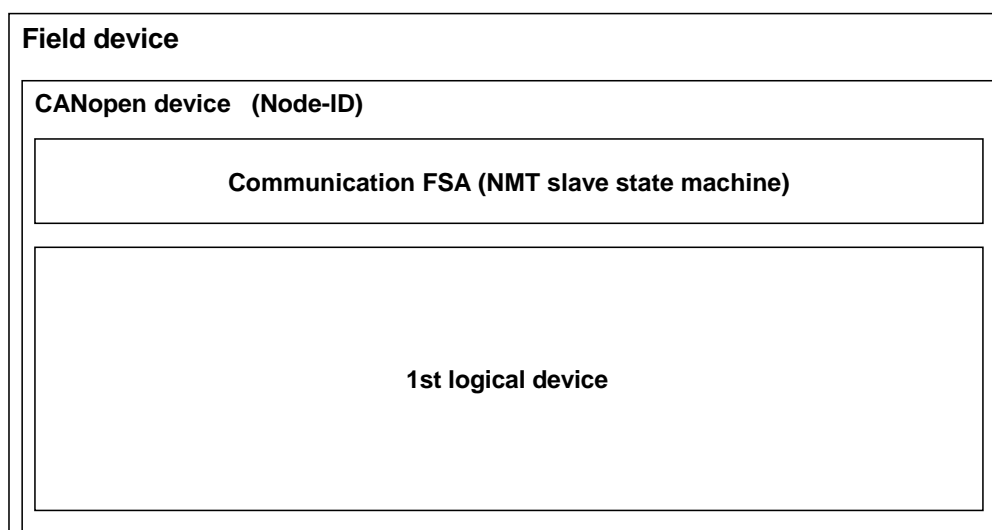


Légende

Anglais	Français
Field device	Dispositif de terrain
CANopen device (Node-ID)	Dispositif CANopen (indicatif de nœud)
1 st logical device	1 ^{er} dispositif logique
Communication FSA	FSA de communication
Logical device	Dispositif logique
Virtual device	Équipement virtuel
Device	Dispositif
8 th logical device	8 ^{ème} dispositif logique

Figure 5 – Modèle de dispositif de terrain

Le dispositif de terrain illustré à la Figure 5 doit fournir au moins un dispositif CANopen. Chaque dispositif CANopen contenu dans le dispositif de terrain doit fournir au moins une interface de réseau associée comprenant le protocole de couche de liaison de données (voir Article 6) et la couche physique (voir Article 5), un indicatif de nœud CANopen et au moins un FSA de communication. Le premier FSA de communication contient la machine d'état esclave NMT comme défini dans la EN 50325-4. Les autres FSA de communication contiennent une machine d'état de situation critique (voir EN 50325-4) et d'autres machines. Un dispositif CANopen doit prendre en charge au moins un dispositif logique et peut en prendre en charge jusqu'à huit. Chaque dispositif logique peut contenir plusieurs équipements virtuels et éventuellement un dispositif logique FSA. Un équipement virtuel contient un équipement virtuel FSA et n'est pas distribué à plusieurs dispositifs logiques. Le dispositif de terrain minimal est présenté à la Figure 6.



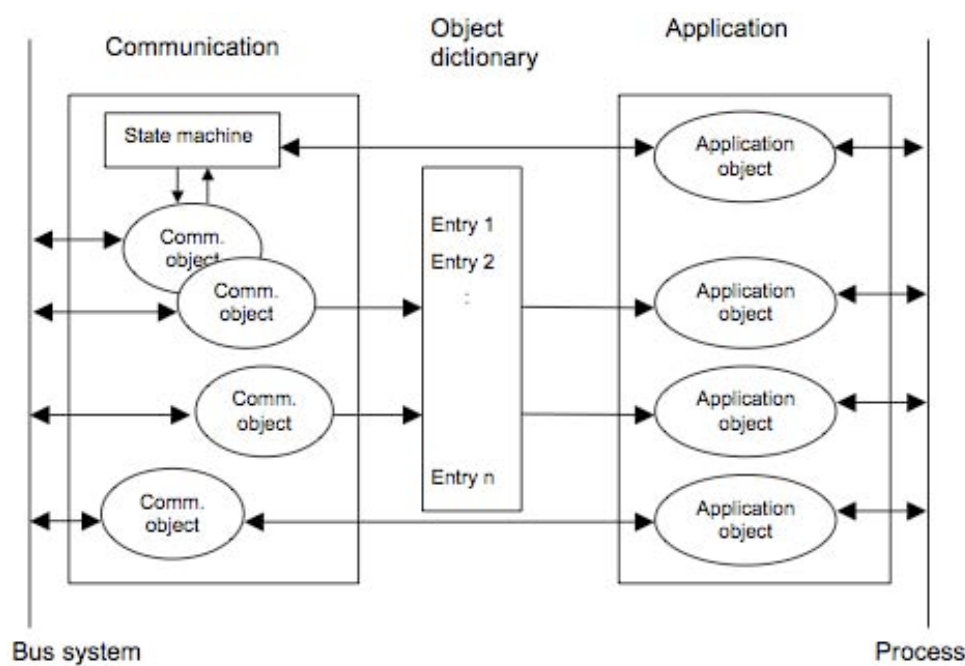
Légende

Anglais	Français
Field device	Dispositif de terrain
CANopen device (Node-ID)	Dispositif CANopen (Indicatif de nœud)
Communication FSA (NMT slave state machine)	FSA de communication (machine d'état esclave NMT)
1 st logical device	1 ^{er} dispositif logique

Figure 6 – Dispositif de terrain minimal

Un dispositif CANopen est structuré comme illustré à la Figure 7:

- Communication – Cette unité fonctionnelle fournit les objets de communication et la fonctionnalité appropriée pour le transport des éléments de données par l'intermédiaire de la structure de réseau sous-jacente.
- Dictionnaire d'objets – Constitue un ensemble de tous les éléments de données exerçant une influence sur le comportement des objets d'application et de communication et de la machine d'état utilisés sur ce dispositif.
- Application – Comprend la fonctionnalité du dispositif par rapport à l'interaction avec l'environnement de processus.



Légende

Anglais	Français
Object dictionary	Dictionnaire d'objets
State machine	Machine d'état
Application object	Objet d'application
Communication object	Objet de communication
Entry	Entrée
Bus system	Système de bus
Process	Processus

Figure 7 – Structure du dispositif CANopen

7.4 Objets de communication CANopen

Les dispositifs connectés à un réseau de rame basé sur CANopen doivent prendre en charge tous les objets de communication obligatoires de la EN 50325-4, ainsi que la machine d'état de gestion de réseau CANopen définie dans la EN 50325-4.

Une méthode de répartition CAN-ID est définie afin de réduire l'effort de configuration applicable aux réseaux CANopen simples. Ces indicateurs CAN (CAN-ID) doivent être disponibles dans l'état NMT « Préopérationnel » directement après l'état NMT Initialisation (si aucune modification n'a été enregistrée). Les objets SYNC, TIME, EMCY write et PDO peuvent être supprimés et régénérés avec de nouveaux indicateurs CAN par une répartition dynamique. Un dispositif CANopen doit fournir les indicateurs CAN correspondants pour les seuls objets de communication pris en charge. La méthode de répartition CAN-ID est définie dans la EN 50325-4.

7.5 Dictionnaire d'objets CANopen

Le dictionnaire d'objets CANopen, défini dans la EN 50325-4, contient un nombre maximal de 65536 objets dont l'adressage s'effectue par un index de 16 bits et jusqu'à 256 sous-index par objet, dont l'adressage s'effectue par un sous-index de 8 bits.

Les types de données statiques aux index compris entre 0001_h to 001F_h contiennent des définitions de type pour les types de données normales tels que BOOLEAN, INTEGER, UNSIGNED, virgule flottante, chaîne, etc.

Les types de données complexes aux index compris entre 0020_h et 003F_h sont des structures prédéfinies composées de types de données normales et communes à tous les dispositifs CANopen.

Les types de données complexes, spécifiques au fabricant, aux index compris entre 0040_h et 005F_h sont des structures composées de types de données normales, mais spécifiques à un dispositif CANopen particulier.

Les profils de dispositifs CANopen peuvent définir des types de données supplémentaires spécifiques à leur type de dispositif. Les types de données statiques et les types de données complexes définis par le profil de dispositif CANopen sont énumérés aux index compris entre 0060_h et 025F_h.

Un dispositif CANopen fournit éventuellement la structure des types de données complexes pris en charge (index compris entre 0020_h et 005F_h et entre 0060_h et 025F_h) à accès lecture à l'index correspondant. Le sous-index 00_h fournit alors le sous-index le plus élevé pris en charge à cet index, les sous-index suivants contenant le type de données codé comme UNSIGNED16 conformément à la EN 50325-4.

Le domaine de profil de communication aux index compris entre 1000_h et 1FFF_h contient les paramètres spécifiques de communication. Ces objets sont communs à tous les dispositifs CANopen.

Le domaine de profil normalisé aux index compris entre 6000_h et 9FFF_h contient tous les objets de données communs à une classe de dispositifs CANopen pouvant être lus ou établis via le réseau. Les profils de dispositifs CANopen utilisent des objets compris entre 6000_h et 9FFF_h pour décrire les paramètres et la fonctionnalité.

Le concept de dictionnaire d'objets traite des caractéristiques facultatives, ce qui signifie qu'un fabricant peut ne pas fournir certaines fonctionnalités étendues pour ses dispositifs CANopen, mais en revanche s'il le souhaite, doit le faire de manière prédéfinie. Le dictionnaire d'objets comporte un espace, aux index compris entre 2000_h et 5FFF_h, pour la fonctionnalité spécifique au fabricant.

Les variables de réseau aux index compris entre A000_h et AFFF_h contiennent des variables d'entrée et de sortie, qui font partie intégrante d'un dispositif CANopen programmable.

Les variables de système aux index compris entre B000_h et BFFF_h contiennent des variables d'entrée et de sortie, qui font partie intégrante d'un réseau CANopen sous-jacent en termes hiérarchiques.

La structure générale du dictionnaire d'objets CANopen est illustrée au Tableau 4.

Tableau 4 – Structure du dictionnaire d'objets CANopen

Index	Objet
0000 _h	non utilisé
0001 _h – 001F _h	Type de données statiques
0020 _h – 003F _h	Types de données complexes
0040 _h – 005F _h	Types de données complexes spécifiques au fabricant
0060 _h – 025F _h	Types de données spécifiques au profil de dispositif
0260 _h – 03FF _h	réservé
0400 _h – 0FFF _h	réservé
1000 _h – 1FFF _h	Domaine de profil de communication
2000 _h – 5FFF _h	Domaine de profil spécifique au fabricant
6000 _h – 67FF _h	1 ^{er} dispositif logique de domaine de profil normalisé
6800 _h – 6FFF _h	2 ^{ème} dispositif logique de domaine de profil normalisé
7000 _h – 77FF _h	3 ^{ème} dispositif logique de domaine de profil normalisé
7800 _h – 7FFF _h	4 ^{ème} dispositif logique de domaine de profil normalisé
8000 _h – 87FF _h	5 ^{ème} dispositif logique de domaine de profil normalisé
8800 _h – 8FFF _h	6 ^{ème} dispositif logique de domaine de profil normalisé
9000 _h – 97FF _h	7 ^{ème} dispositif logique de domaine de profil normalisé
9800 _h – 9FFF _h	8 ^{ème} dispositif logique de domaine de profil normalisé
A000 _h – AFFF _h	Domaine variable de réseau normalisé
B000 _h – BFFF _h	Domaine variable de système normalisé
C000 _h – FFFF _h	réservé

7.6 Objets de communication CANopen prédéfinis

7.6.1 Teneur

Le présent article spécifie la capacité de communication CANopen de base pour les dispositifs CANopen participant à un réseau de rame basé sur CANopen selon la présente spécification.

7.6.2 Objet 1000_h: Type de dispositif

Cet objet donne le type de dispositif et sa fonctionnalité. La valeur 0000_h relative au numéro de profil du dispositif désigne un dispositif logique qui ne suit pas un profil normalisé. Dans ce cas, l'information supplémentaire est égale à 0000_h (si aucun dispositif logique supplémentaire n'est mis en œuvre) ou FFFF_h (si un dispositif logique supplémentaire est mis en œuvre).

Le paramètre d'information supplémentaire est égal à FFFF_h pour les modules de dispositifs logiques multiples, le numéro de profil de dispositif référencé par l'objet 1000_h étant le profil du premier dispositif logique dans le dictionnaire d'objets. Tous les autres profils d'un module de dispositif logique multiple identifient leurs profils aux objets 67FF_h + x * 800_h avec x = numéro interne du dispositif logique (compris entre 1 et 8) moins 1. Ces objets décrivent le type de dispositif du dispositif logique précédent, ayant exactement la même définition de valeur que l'objet 1000_h.

La Figure 8 illustre la structure d'objet. La définition de valeur, la description d'objet et la description d'entrée sont spécifiées dans la EN 50325-4.

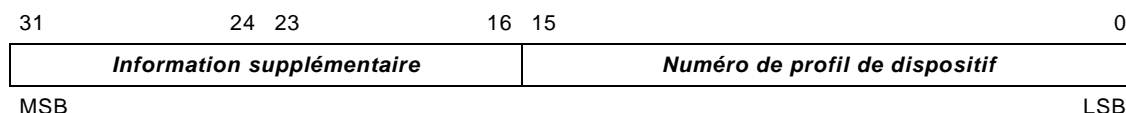


Figure 8 – Structure d'objet « Type de dispositif »

NOTE Les dispositifs connectés aux réseaux de rame basés sur CANopen peuvent suivre le profil d'application CANopen pour les réseaux de commande de véhicules ferroviaires CiA 421. Ce profil d'application CANopen définit les données d'application, sur la base de l'UIC 556, échangées dans un réseau de rame basé sur CANopen.

7.6.3 Objet 1001_h: Registre d'erreurs

Cet objet fournit les informations d'erreur. Le dispositif CANopen met en correspondance les erreurs internes dans cet objet. Ces informations d'erreur sont publiées comme partie intégrante du message de situation critique. La définition de valeur, la description d'objet et la description d'entrée sont spécifiées dans la EN 50325-4.

7.6.4 Objet 1014_h: Objet situation critique COB-ID

Cet objet doit être mis en œuvre. Il est spécifié dans la EN 50325-4. Le CAN-ID, qui fait partie intégrante de cet objet, ne doit pas être modifié.

7.6.5 Objet 1017_h: Producteur de pulsation (heartbeat)

Tous les dispositifs CANopen qui sont connectés à un réseau de rame basé sur CANopen doivent mettre en œuvre cet objet. Il est spécifié dans la EN 50325-4.

Les dispositifs doivent prendre en charge les transmissions de messages de pulsation entre 100 ms et 1 000 ms.

7.6.6 Objet 1018_h: Objet Identité

Cet objet fournit des informations générales concernant le dispositif, comme spécifié dans la EN 50325-4.

7.6.7 Objet 1029_h: Comportement à l'erreur

Cet objet spécifie l'état de réglage du dispositif, en cas de détection d'une erreur de communication ou d'une erreur interne au dispositif. Il est spécifié dans la EN 50325-4.

7.6.8 Objet 67FF_h: Type de dispositif

Cet objet doit décrire le premier dispositif logique dans un module de dispositifs multiples conformément à la EN 50325-4.

7.6.9 Objets de données de service (SDO)

Tout dispositif CANopen prend en charge le premier canal serveur SDO. Les dispositifs CANopen connectés à un réseau de rame basé sur CANopen peuvent prendre en charge des canaux serveur ou client SDO supplémentaires.

En cas de prise en charge d'un canal SDO supplémentaire, l'ensemble de paramètres SDO associé est pris en charge dans le dictionnaire d'objets CANopen comme défini dans la EN 50325-4.

La présente spécification ne prédéfinit aucun canal SDO supplémentaire.

7.6.10 Objets de données de processus (PDO)

Les dispositifs CANopen exploités dans un réseau de rame basé sur CANopen peuvent prendre en charge jusqu'à 512 PDO dans les directions de transmission et de réception.

Dans le cas où un dispositif CANopen prend en charge un PDO, le paramètre de communication PDO et les entrées de mise en correspondance associés sont pris en charge dans le dictionnaire d'objets CANopen comme défini dans la EN 50325-4.

Aucun PDO n'est prédéfini.

8 Données d'application

8.1 Teneur

Le présent article spécifie la représentation des données d'application communiquées entre un réseau de rame basé sur CANopen et un Réseau Central de Train par l'intermédiaire d'une Passerelle.

NOTE 1 Les données d'application sont décrites dans le profil d'application du TCN.

En règle générale, les données d'application sont gérées dans les intervalles d'index du dictionnaire d'objets CANopen 2000_h à $5FFF_h$ et 6000_h à $9FFF_h$. Le comportement d'un dispositif spécifique au fabricant est contrôlé par le biais de l'intervalle d'index 2000_h à $5FFF_h$. Le comportement d'un dispositif CANopen normalisé est contrôlé par l'intermédiaire d'objets compris dans l'intervalle d'index 6000_h à $9FFF_h$.

NOTE 2 Les entrées du dictionnaire d'objets comprises dans l'intervalle d'index 6000_h à $9FFF_h$ sont spécifiées dans le dispositif CANopen et les profils d'application, et sont disponibles au format CAN en mode d'automatisation.

8.2 Représentation des données d'application CANopen

Dans la mesure où les données d'application relatives au TCN ne sont pas encore définies, la représentation des données d'application TCN dans l'intervalle d'index du dictionnaire d'objets normalisé n'est également pas définie.

8.3 Principe de représentation recommandé des données d'application

8.3.1 Teneur

Le présent article spécifie un principe de représentation des données d'application pour la représentation des données d'application dans les réseaux de rame basés sur CANopen. Il est par conséquent possible de mettre en correspondance le processus d'application des objets de données de processus (PDO) CANopen pour le transfert des données de processus.

8.3.2 Données d'application pour la commande des portes

Le présent article spécifie le principe de représentation recommandé utilisant les informations relatives à la commande des portes, ces informations étant communiquées entre un réseau de rame basé sur CANopen et un Réseau Central de Train par l'intermédiaire d'une Passerelle. Les données de processus représentées sont par conséquent disponibles dans le réseau de rame basé sur CANopen et peuvent être communiquées par l'intermédiaire des objets de communication CANopen.

Le type d'accès est spécifié pour l'interface du dispositif passerelle et du réseau de rame basé sur CANopen. D'autres dispositifs tels que, par exemple, des commandes de portes ou des unités de porte, qui sont connectés à un réseau de rame basé sur CANopen, gèrent les données comme défini dans le présent article, à l'index de dictionnaire d'objets CANopen

donné. Le type d'accès de l'objet peut être modifié en type d'accès approprié comme défini dans la EN 50325-4.

NOTE Les données d'application illustrées sont issues du document UIC 556 et sont spécifiées dans la représentation du profil d'application CANopen CiA 421 du réseau de commande de véhicules ferroviaires.

8.3.3 Objets d'application des commandes des portes utilisées

8.3.3.1 Objet 6007_h: Exportation du mot d'état des portes extérieures

Cet objet doit indiquer l'état des portes extérieures du véhicule ferroviaire local. Cette information est disponible dans le TCN par l'intermédiaire de la passerelle d'accès au Réseau Central de Train. La Figure 9 spécifie la structure d'objet et le Tableau 5 spécifie la définition de valeur. Le Tableau 6 spécifie la description d'objet et le Tableau 7 spécifie la description d'entrée.

NOTE L'objet doit correspondre à l'octet 20 de télégramme R3, défini dans le document UIC 556.

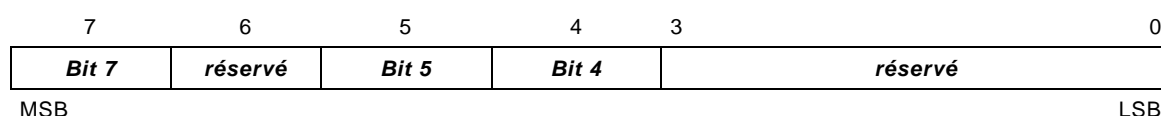


Figure 9 – Structure d'objet

Tableau 5 – Définition de valeur

Bit	Valeur	Définition de valeur
Bit 4	0	Au moins une porte gauche est ouverte
	1	Toutes les portes gauches sont verrouillées
Bit 5	0	Au moins une porte droite est ouverte
	1	Toutes les portes droites sont verrouillées
Bit 7	0	Le verrouillage sélectif des portes latérales n'est pas activé
	1	Le verrouillage sélectif des portes latérales est activé
Réservé		Réservée (doit être ignorée)

Tableau 6 – Description d'objet

Attribut	Valeur
INDEX	6007 _h
Nom	Exportation du mot d'état des portes extérieures
Code d'objet	Variable
Type de données	UNSIGNED8
Catégorie	Facultatif

Tableau 7 – Description d'entrée

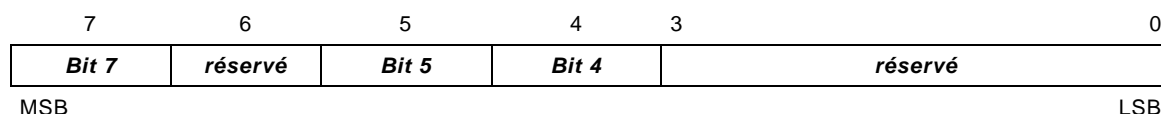
Attribut	Valeur
Sous-index	00 _h
Accès	rw
Mise en correspondance PDO	Facultatif
Plage de valeurs	Voir définition de valeur
Valeur par défaut	Spécifique au fabricant

8.3.4 Objets d'application des commandes des portes produites

8.3.4.1 Objet 6006_h: Importation du mot d'état des portes extérieures

Cet objet doit indiquer l'état des portes extérieures des autres véhicules ferroviaires. Chaque sous-index doit fournir l'état des portes du véhicule concerné, qui correspond au numéro de véhicule UIC. Par conséquent, l'état des portes extérieures des autres véhicules est disponible dans le réseau de rame basé sur CANopen. L'état général des portes de tous les véhicules est indiqué dans le sous-index 21_h. La Figure 10 spécifie la structure d'objet et le Tableau 8 spécifie la définition de valeur. Le Tableau 9 spécifie la description d'objet et le Tableau 10 spécifie la description d'entrée.

NOTE L'objet doit correspondre à l'octet 20 de télégramme R3, défini dans le document UIC 556.

**Figure 10 – Structure d'objet****Tableau 8 – Définition de valeur**

Bit	Valeur	Définition de valeur
Bit 4	0	Au moins une porte gauche est ouverte
	1	Toutes les portes gauches sont verrouillées
Bit 5	0	Au moins une porte droite est ouverte
	1	Toutes les portes droites sont verrouillées
Bit 7	0	Le verrouillage sélectif des portes latérales n'est pas activé
	1	Le verrouillage sélectif des portes latérales est activé
Réservé		Réservée (doit être ignorée)

Tableau 9 – Description d'objet

Attribut	Valeur
INDEX	6006 _h
Nom	Importation du mot d'état des portes extérieures
Code d'objet	ARRAY
Type de données	UNSIGNED8
Catégorie	Facultatif

Tableau 10 – Description d'entrée

Attribut	Valeur
Sous-index	00 _h
Description	Sous-index le plus élevé pris en charge
Catégorie d'entrée	Obligatoire
Accès	ro
Mise en correspondance PDO	Non
Plage de valeurs	01 _h à 21 _h
Valeur par défaut	Non
Sous-index	01 _h
Description	Etat des portes véhicule UIC 1
Catégorie d'entrée	Facultatif
Accès	ro
Mise en correspondance PDO	Facultatif
Plage de valeurs	Voir Tableau 46
Valeur par défaut	Non
vers	
Sous-index	20 _h
Description	Etat des portes véhicule UIC 32
Catégorie d'entrée	Facultatif
Accès	ro
Mise en correspondance PDO	Non
Plage de valeurs	Facultatif
Valeur par défaut	Non
Sous-index	21 _h
Description	Etat des portes du train
Catégorie d'entrée	Facultatif
Accès	ro
Mise en correspondance PDO	Non
Plage de valeurs	Facultatif
Valeur par défaut	Non

8.3.4.2 Objet 6001_h: Commande des portes extérieures

Cet objet doit fournir le premier mot de commande des portes extérieures. La Figure 11 spécifie la structure d'objet et le Tableau 11 spécifie la définition de valeur. Le Tableau 12 spécifie la description d'objet et le Tableau 13 spécifie la description d'entrée.

NOTE L'objet doit correspondre à l'octet 20 de télégramme R3, défini dans le document UIC 556.

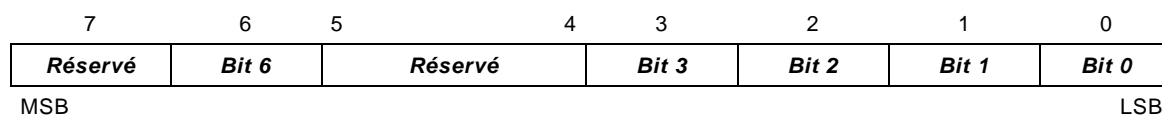


Figure 11 – Structure d'objet

Tableau 11 – Définition de valeur

Bit	Valeur	Définition de valeur
Bit 0	0	Inactif
	1	Toutes les portes sont fermées
Bit 1	0	Inactif
	1	Interruption fermeture des portes
Bit 2	0	Libérer toutes les portes gauches
	1	Verrouiller toutes les portes gauches
Bit 3	0	Libérer toutes les portes droites
	1	Verrouiller toutes les portes droites
Bit 6	0	Ne pas déployer la marche
	1	Déployer la marche
Réservé		Réservée (doit être ignorée)

Tableau 12 – Description d'objet

Attribut	Valeur
INDEX	6001 _h
Nom	Commande 1 des portes extérieures
Code d'objet	Variable
Type de données	UNSIGNED8
Catégorie	Facultatif

Tableau 13 – Description d'entrée

Attribut	Valeur
Sous-index	00 _h
Accès	ro
Mise en correspondance PDO	Facultatif
Plage de valeurs	Voir définition de valeur
Valeur par défaut	Non

9 Gestion de réseau CANopen

9.1 Teneur

Le présent article spécifie la gestion de réseau applicable aux réseaux de rame basés sur CANopen. La définition de la gestion de réseau inclut la définition du comportement de démarrage du réseau ainsi que les définitions associées aux réseaux qui fonctionnent sans maître NMT et aux réseaux dont un dispositif CANopen peut fonctionner en mode maître NMT. Ces définitions sont destinées à compléter la couche d'application et le profil de communication CANopen fournis dans la EN 50325-4.

Les réseaux de rame CANopen dont deux dispositifs CANopen ou plus peuvent fonctionner en mode maître NMT (maître volant NMT) ne relèvent pas du domaine d'application de la présente spécification. Ces réseaux ne sont toutefois pas exclus.

NOTE Concernant une plus grande disponibilité, les réseaux de rame basés sur CANopen peuvent contenir plusieurs dispositifs CANopen comprenant une fonctionnalité Maître NMT. A un moment donné, seul un maître NMT actif est admis. Des mécanismes doivent être utilisés qui permettent de transmettre la fonctionnalité Maître NMT d'un dispositif CANopen à un autre. Ces services font partie intégrante de la fonctionnalité Maître volant CANopen décrite dans le document CiA 302.

9.2 Fonctionnalité Esclave NMT CANopen

Les dispositifs exploités dans un réseau de rame basé sur CANopen doivent fournir la fonctionnalité Esclave NMT CANopen et doivent mettre en œuvre tous les services et protocoles de communication obligatoires, ainsi que tous les objets obligatoires spécifiés dans la EN 50325-4. Les objets de communication minimum sont définis en 7.6.

D'autres objets de communication peuvent également être pris en charge. Par ailleurs, la fonctionnalité Gestionnaire CANopen peut être prise en charge.

9.3 Fonctionnalité Gestionnaire CANopen

9.3.1 Généralités

Le présent paragraphe donne la définition de la fonctionnalité Gestionnaire CANopen pour les réseaux de rame basés sur CANopen.

Outre le processus d'application, un réseau CANopen comporte plusieurs fonctionnalités supplémentaires différentes. Différents termes font référence à ces fonctionnalités. Le présent paragraphe a pour objet de clarifier ces termes.

Au sein d'un système réparti, le processus d'application est divisé en plusieurs parties fonctionnant sur différents dispositifs CANopen. Habituellement, et du point de vue de l'application, un dispositif CANopen est chargé de la commande du système. Ce dispositif CANopen est appelé *maître d'application*.

Du point de vue du réseau, il existe plusieurs fonctionnalités supplémentaires qui ne traitent pas directement de l'application mais fournissent des fonctions d'aide à cette dernière. Ces fonctionnalités supplémentaires sont basées sur une relation maître/esclave, client/serveur ou producteur/consommateur.

Le terme Gestionnaire CANopen est utilisé dans la mesure où il est courant de combiner plusieurs fonctionnalités supplémentaires en un dispositif CANopen.

Un dispositif CANopen est désigné comme gestionnaire CANopen s'il fournit la fonctionnalité Maître NMT et au moins une des fonctionnalités Gestionnaire SDO ou gestionnaire de configuration.

9.3.1.1 Maître NMT

Le gestionnaire de réseau (NMT) fournit des services de contrôle du comportement de réseau des dispositifs CANopen comme défini dans la EN 50325-4. Tous les dispositifs CANopen qui participent à un réseau de rame CANopen, et désignés comme des esclaves NMT, sont contrôlés par des services assurés par un maître NMT. Habituellement, l'application maître NMT fait également partie intégrante du maître d'application.

Le dispositif qui comporte la fonctionnalité Maître NMT est un dispositif CANopen complet. Il prend en charge, outre la fonctionnalité Maître NMT, toutes les fonctions et tous les objets signalés comme obligatoires dans la EN 50325-4.

9.3.1.2 Maître volant

Le mécanisme de maître volant fournit des services pour un maître NMT de réserve dynamique dans un réseau CANopen. Le maître volant est une fonctionnalité facultative d'un dispositif CANopen. Le dispositif CANopen qui met en œuvre le maître volant met en œuvre la fonctionnalité Maître NMT. La définition des responsabilités, fonctionnalités et services du maître volant ne relève pas du domaine d'application du présent document.

NOTE Les définitions applicables à la fonctionnalité Maître volant sont fournies dans le document CiA 302.

9.3.1.3 Gestionnaire SDO

Le gestionnaire SDO constitue une fonctionnalité facultative, chargée de la gestion de la réalisation dynamique des connexions SDO. Si un réseau CANopen comporte un gestionnaire SDO, ce dernier coexiste avec le maître NMT sur le même dispositif CANopen.

NOTE Les définitions applicables à la fonctionnalité Gestionnaire SDO sont fournies dans le document CiA 302.

9.3.1.4 Gestionnaire de configuration

Le gestionnaire de configuration est une fonctionnalité facultative qui fournit les mécanismes de configuration des dispositifs CANopen dans un réseau CANopen lors de l'initialisation. Les mécanismes sont appelés Gestion de configuration (CMT). Si un réseau CANopen comporte le Gestionnaire de configuration, ce dernier coexiste avec le maître NMT sur le même dispositif CANopen.

NOTE Les définitions applicables à la fonctionnalité Gestionnaire de configuration sont fournies dans le document CiA 302.

9.3.1.5 Producteur SYNC

Le producteur SYNC est une fonctionnalité facultative chargée de la transmission de l'objet SYNC. Cette fonctionnalité peut être associée à tout dispositif CANopen du réseau de rame basé sur CANopen. Les entrées appropriées du dictionnaire d'objets CANopen sont définies dans la EN 50325-4.

9.3.1.6 Producteur TIME

Le producteur TIME est une fonctionnalité facultative chargée de la transmission de l'objet TIME STAMP. Cette fonctionnalité peut être associée à tout dispositif CANopen du réseau de rame basé sur CANopen. L'objet « TIME STAMP » est défini dans la EN 50325-4.

9.3.1.7 Maître LSS

Les Services de Stratification (LSS) fournissent des services de configuration de la couche 2 (synchronisation des bits) et de NMT (Indicatif de nœud CANopen) via CAN. Les services LSS permettent au maître LSS de configurer les esclaves LSS. La définition des Services de Stratification ne relève pas du domaine d'application du présent document.

NOTE Les définitions applicables aux Services de Stratification sont fournies dans le document CiA 305.

9.3.2 Emploi du dictionnaire d'objets

Plusieurs objets liés à la configuration et à la validation des dispositifs CANopen relèvent du type d'objet ARRAY. Le sous-index des entrées de ce type de ARRAY correspond à l'indicatif de nœud CANopen du dispositif correspondant. Ces objets peuvent avoir moins de 127 entrées. Dans ce cas, tous ces objets doivent avoir exactement le même ensemble d'entrées prises en charge. Le sous-index 00_h doit fournir le sous-index le plus élevé pris en charge à l'index concerné. La description d'objet associée spécifiée à l'Article 9 désigne cet objet comme la condition *SupportedNodeID*.

NOTE Pour qu'un gestionnaire CANopen puisse être utilisé en de façon universelle, il est recommandé de prendre en charge tout l'intervalle de sous-index compris entre 01_h et 7F_h.

9.3.3 Réseaux redondants

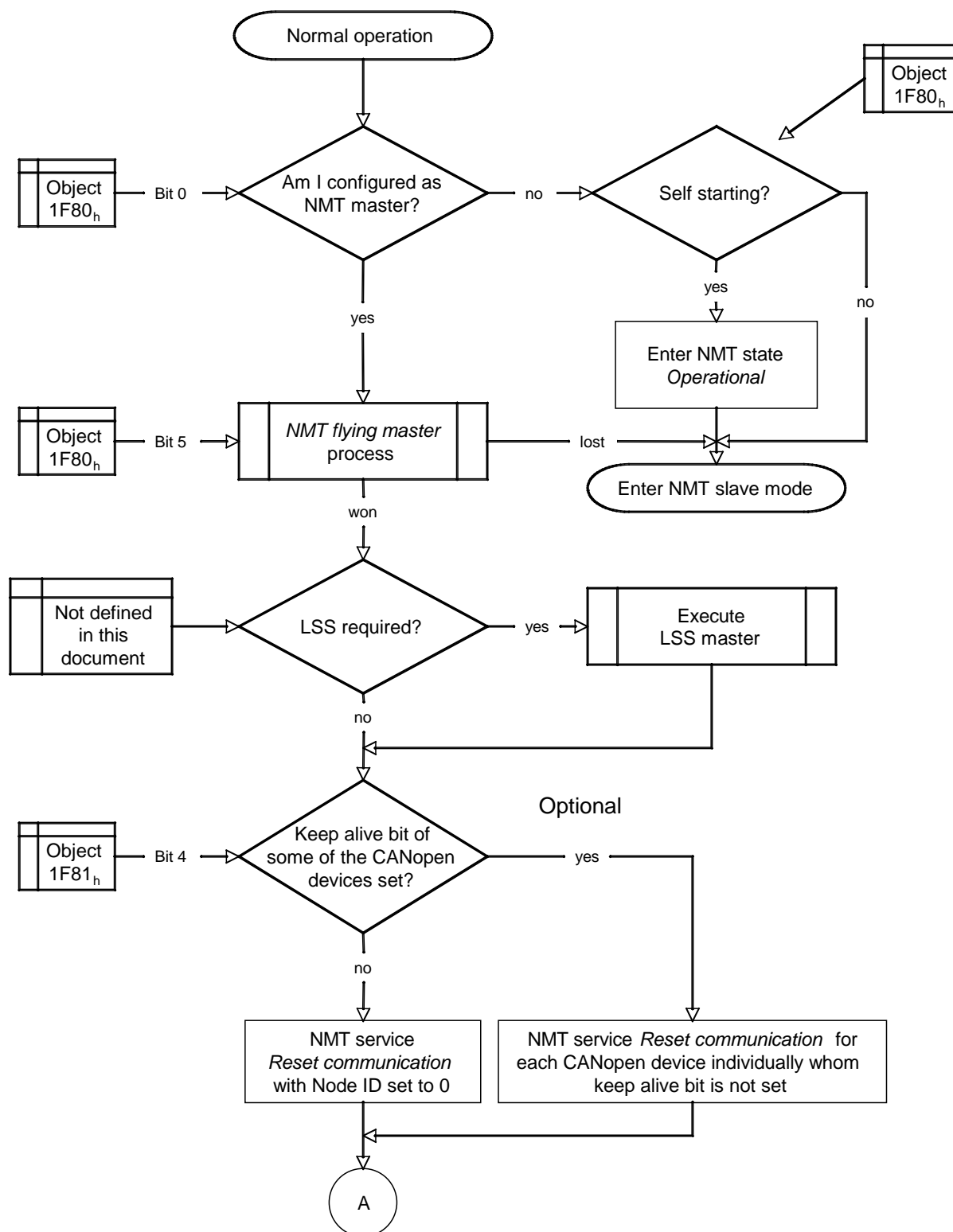
Des réseaux redondants doivent être utilisés pour des applications à grande disponibilité. Les réseaux redondants comportent au moins deux lignes CAN. Ceci permet la communication entre les dispositifs CANopen, dans le cas d'une défaillance simple de l'interconnexion physique entre les dispositifs CANopen. La définition des responsabilités, fonctionnalités et services applicables aux réseaux redondants ne relève pas du domaine d'application du présent document.

NOTE Les définitions applicables à l'établissement des réseaux CANopen redondants sont fournies dans le document CiA 302.

9.4 Démarrage du NMT CANopen

9.4.1 Démarrage du NMT

Les gestionnaires CANopen doivent se comporter selon la machine d'état esclave NMT comme défini dans la EN 50325-4. Tous les esclaves NMT attribués doivent être initialisés préalablement à la transition entre l'état NMT *Préopérationnel* et l'état NMT *Opérationnel* du gestionnaire CANopen. Le principal organigramme de la procédure est défini aux Figures 12 et 13. L'organigramme de démarrage le plus simple est défini à la Figure 14.



Légende

Anglais	Français
Normal operation	Fonctionnement normal
Object	Objet
Am I configured as NMT master?	Ma configuration est-elle celle d'un maître NMT ?
no	non
Self starting?	Démarrage automatique ?
yes	oui
Enter NMT state <i>Operational</i>	Adoption de l'état NMT <i>Opérationnel</i>
NMT <i>flying master process</i>	Processus <i>Maître volant NMT</i>
lost	perte
Enter NMT slave mode	Adoption du mode Esclave NMT
won	Gain
LSS required?	LSS nécessaires ?
Not defined in this document	Non défini dans le présent document
Execute LSS master	Exécution du maître LSS
Optional	Facultatif
Keep alive bit of some of the CANopen devices set?	Le bit d'entretien de certains dispositifs CANopen est-il mis à 1 ?
NMT service Reset communication with Node ID set to 0	Service NMT <i>Initialisation de la communication</i> avec indicatif de nœud mis à 0
NMT service Reset communication for each CANopen device individually whom keep alive bit is not set	Service NMT <i>Initialisation de la communication</i> pour chaque dispositif CANopen individuel, dont le bit d'entretien n'est pas mis à 1

Figure 12 – Démarrage NMT, partie 1

Le processus Démarrage du NMT, illustré à la Figure 12, comprend les étapes de base suivantes:

NOTE 1 Le processus d'exécution du maître LSS ne relève pas du domaine d'application de la présente norme.

- Le bit 0 de l'objet 1F80_h (voir 9.8.7) permet de déterminer si ce dispositif CANopen doit être le maître NMT. Si le dispositif CANopen est le maître NMT, le processus doit se poursuivre. Si le dispositif CANopen est configuré comme dispositif à démarrage automatique, il doit adopter l'état NMT « Opérationnel » de manière automatique. Si le dispositif CANopen n'est pas le maître NMT, le processus doit s'interrompre.
- Le bit 5 de l'objet 1F80_h (voir 9.8.7) permet de déterminer si ce dispositif CANopen doit participer au service « *Négociation du maître volant NMT* ». Si le dispositif CANopen doit participer au service « *Négociation du maître volant NMT* », et s'il a perdu le service, il ne doit pas devenir maître NMT. La définition de la négociation du maître volant NMT ne relève pas du domaine d'application du présent document.

NOTE 2 La description de la négociation du maître volant NMT est fournie dans le document CiA 302.

- Si les services LSS sont nécessaires pour le réglage de l'indicateur de nœud CANopen et du débit binaire des autres dispositifs CANopen dans le réseau, le maître NMT doit exécuter les services de maître LSS. Ces services peuvent être exécutés à tout moment. La définition précise des services de maître LSS ne relève pas du domaine d'application du présent document.

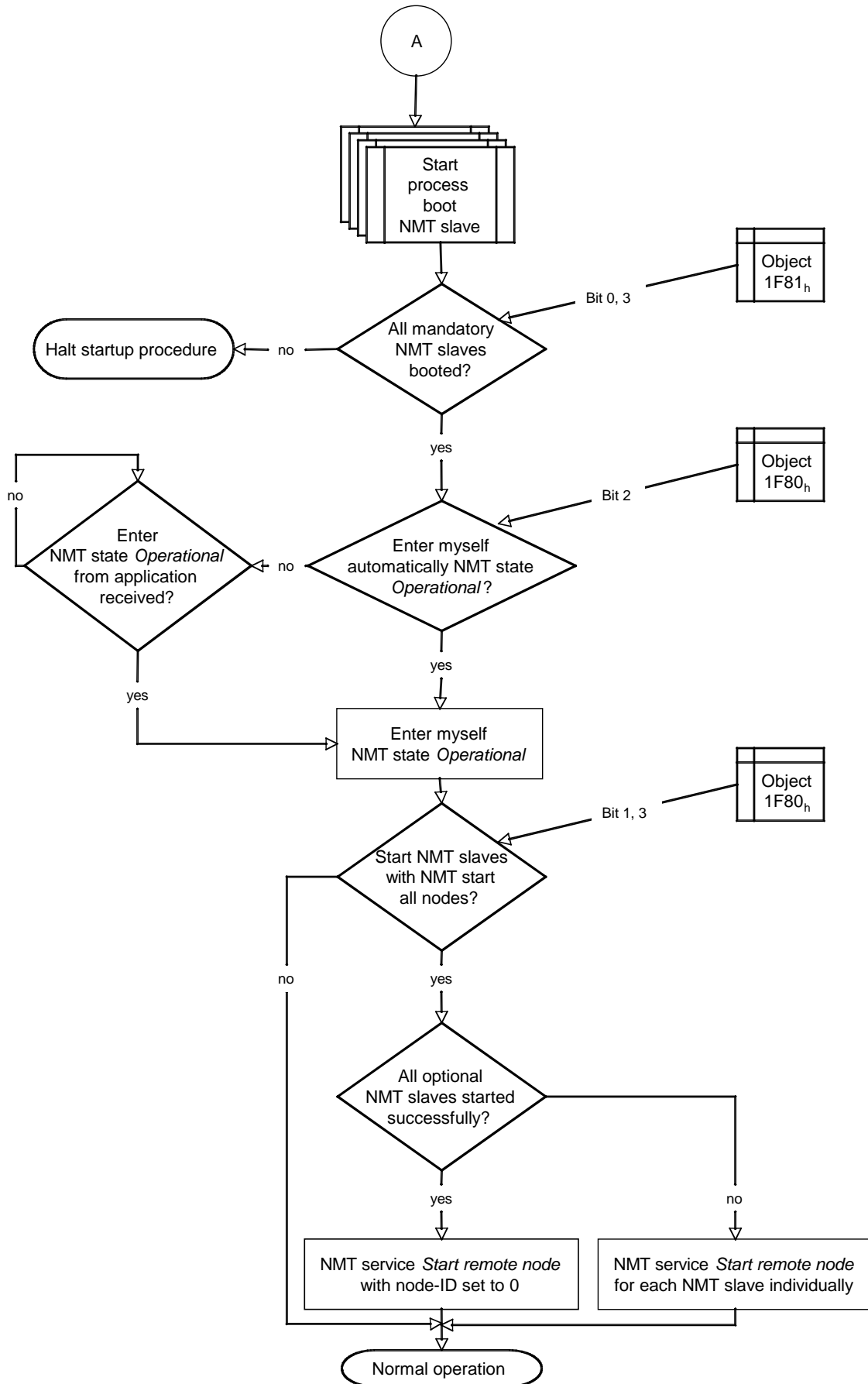
NOTE 3 La description des Services de Stratification est fournie dans le document CiA 305.

- Le bit 4 de toutes les entrées de l'objet 1F81_h (voir 9.8.8) permet de déterminer si le maître NMT doit exécuter le service NMT *Initialisation de la communication* avec l'indicateur de nœud CANopen mis à 0 ou si le service NMT *Initialisation de la communication* doit être exécuté pour chaque dispositif CANopen du réseau de manière individuelle. Si, au

moins pour une entrée de l'objet 1F81_h, le bit 4 est mis à 1_b et le dispositif Canopen correspondant est à l'état NMT *Opérationnel*, le maître NMT ne doit pas exécuter le service NMT *Initialisation de la communication* avec l'indicatif de nœud CANopen mis à 0. Dans ce cas, chaque dispositif CANopen doit être réinitialisé de manière individuelle. Ceci doit également inclure tous les indicatifs de nœuds CANopen qui ne font pas partie intégrante de la liste d'esclaves 1F81_h.

NOTE 4 Ceci contraint les dispositifs CANopen potentiellement existants qui ne sont pas configurés dans la liste d'esclaves à transmettre un message d'initialisation. Cette action permet au gestionnaire CANopen de reconnaître les dispositifs CANopen non configurés via le pilote d'initialisation (Figure 23). Le service NMT « Initialisation de la communication » ne doit pas s'appliquer au maître NMT proprement dit.

Si pour quelque raison que ce soit, le service NMT *Initialisation de la communication* se révèle nécessaire après l'étape d), il doit être exécuté avec l'indicatif de nœud CANopen de l'esclave NMT. Ceci doit s'appliquer indépendamment de la configuration applicable à cet esclave NMT dans l'objet 1F81_h. Le processus *Démarrage du NMT* reprend à la Figure 13.



Légende

Anglais	Français
Start process boot NMT slave	Démarrer le processus Initialisation de l'esclave NMT
Halt startup procedure	Arrêter la procédure de démarrage
no	non
All mandatory NMT slaves booted?	Tous les esclaves NMT obligatoires ont-ils été initialisés ?
Object	Objet
yes	Oui
Enter NMT state <i>Operational</i> from application received?	Adoption de l'état NMT <i>Opérationnel</i> à partir de l'application reçue ?
Enter myself automatically NMT state <i>Operational</i> ?	Adoption de l'état NMT <i>Opérationnel</i> automatique personnel ?
Enter myself NMT state <i>Operational</i> ?	Adoption de l'état NMT <i>Opérationnel</i> personnel ?
Start NMT slaves with NMT start all nodes?	Démarrer les esclaves NMT avec démarrage de tous les nœuds NMT ?
All optional NMT slaves started successfully?	Tous les esclaves NMT facultatifs ont-ils été démarrés avec succès ?
NMT service <i>Start remote node</i> with node-ID set to 0	Service NMT <i>Démarrage du nœud distant</i> avec indicatif de nœud mis à 0
NMT service <i>Start remote node</i> for each NMT slave individually	Service NMT <i>Démarrage du nœud distant</i> pour chaque esclave NMT de manière individuelle
Normal operation	Fonctionnement normal

Figure 13 – Démarrage NMT, partie 2

- e) Le maître NMT doit démarrer le processus *Démarrer le processus d'initialisation de l'esclave NMT* pour tous les esclaves NMT comme illustré à la Figure 15. Pour tous les esclaves NMT marqués comme obligatoires (bit 0 et bit 3 de l'objet 1F81_h; voir 9.8.8), le processus *Démarrer le processus d'initialisation de l'esclave NMT* doit s'interrompre de manière satisfaisante.
- f) Toute détection d'erreur au cours de l'intervention du processus *Initialisation de l'esclave* pour les esclaves NMT marqués comme obligatoires doit entraîner l'interruption du processus *Démarrage NMT*.
- g) Le bit 2 de l'objet 1F80_h (voir 9.8.7) est utilisé pour déterminer si le maître NMT doit adopter l'état NMT *Opérationnel* de manière automatique par lui-même, ou doit attendre jusqu'à ce que l'application qui fonctionne sur exactement le même dispositif CANopen le demande.
- h) Dans les conditions suivantes
- Le bit 3 de l'objet 1F80_h est mis à 0_b,
 - Le bit 1 de l'objet 1F80_h est mis à 1_b,
 - et tous les esclaves NMT énumérés en 1F81_h sont initialisés avec succès

le service NMT *Démarrage du nœud distant* doit être exécuté avec l'indicatif de nœud CANopen mis à 0.

Dans les conditions suivantes

- Le bit 3 de l'objet 1F80_h est mis à 0_b,
- Le bit 1 de l'objet 1F80_h est mis à 1_b,
- et les esclaves NMT énumérés en 1F81_h ne sont pas tous initialisés avec succès

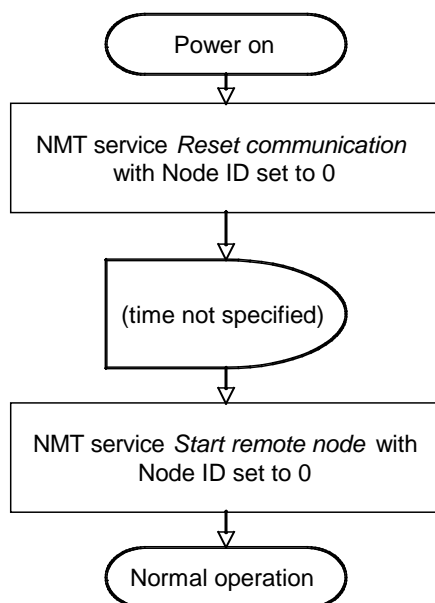
le service NMT *Démarrage du nœud distant* doit être exécuté pour chaque esclave NMT de manière individuelle.

- i) Le processus *Démarrage du NMT* s'étant achevé avec succès, le maître NMT doit alors fonctionner normalement.

L'occurrence et la détection des esclaves NMT non énumérés en 1F81_h relèvent de la responsabilité de l'application.

9.4.2 Démarrage du NMT simple

Dans la mesure où la quasi-totalité des objets et caractéristiques sont facultatifs, il est possible de mettre en œuvre un maître NMT de base, qui peut se révéler adapté à certaines applications. Le retrait de tous les éléments facultatifs des définitions susmentionnées aboutit au processus *Démarrage du NMT simple* illustré à la Figure 14. Un maître NMT simple lui-même prend en charge tous les objets obligatoires qui sont spécifiés dans la EN 50325-4 et qui sont définis dans la présente norme.

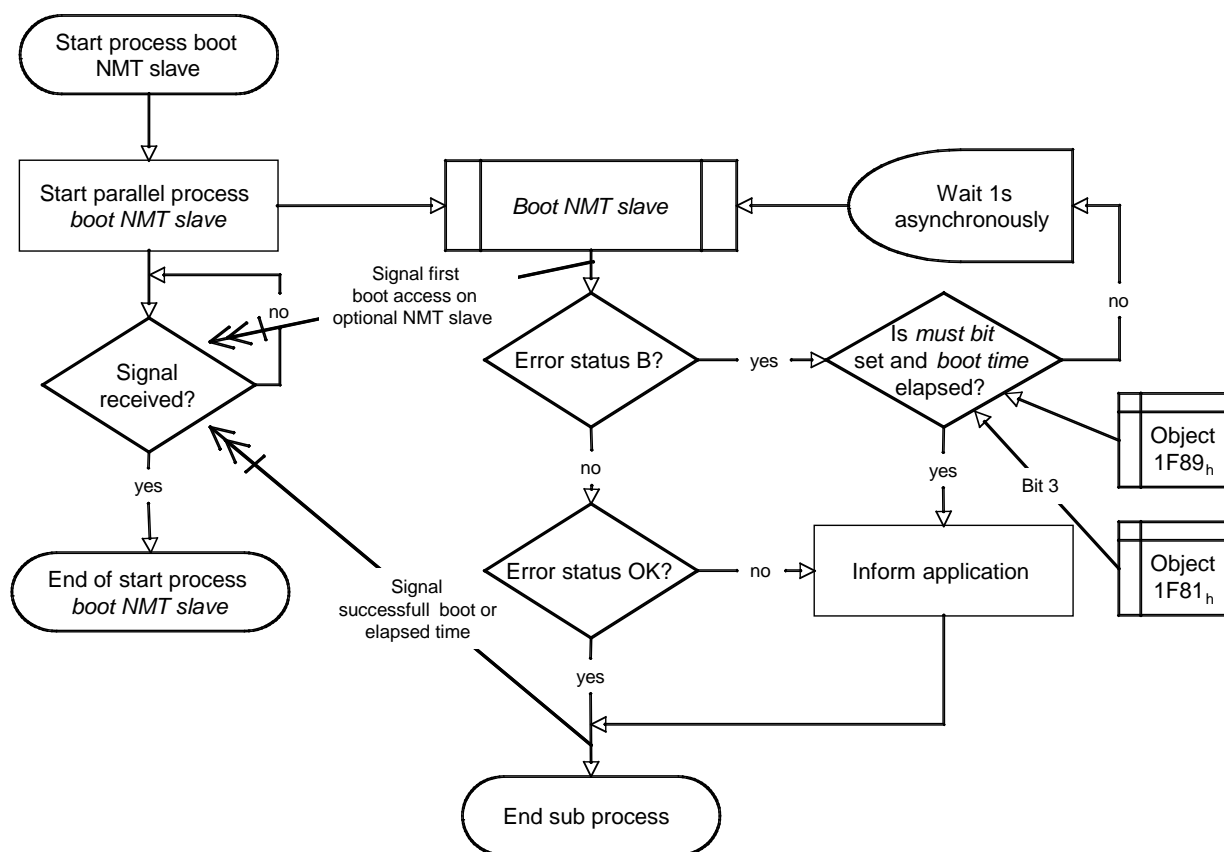


Légende

Anglais	Français
Power on	Mise sous tension
NMT service <i>Reset communication</i> with Node ID set to 0	Service NMT <i>Initialisation de la communication</i> avec indicatif de nœud mis à 0
(time not specified)	Durée non spécifiée
NMT service <i>Start remote node</i> with Node ID set to 0	Service NMT <i>Démarrage du nœud distant</i> avec indicatif de nœud mis à 0
Normal operation	Fonctionnement normal

Figure 14 – Démarrage du NMT simple

9.4.3 Démarrage du processus d'initialisation de l'esclave NMT



Légende

Anglais	Français
Start process boot NMT slave	Démarrage du processus d'initialisation de l'esclave NMT
Start parallel process <i>boot NMT slave</i>	Démarrage du processus parallèle d'initialisation de l'esclave NMT
Boot NMT slave	Initialisation de l'esclave NMT
Wait 1 s asynchronously	Attendre 1 s (processus asynchrone)
Signal first boot access on optional NMT slave	Indiquer le premier accès en initialisation de l'esclave NMT facultatif
no	non
Signal received?	Réception d'un signal ?
Error status?	Etat d'erreur ?
Yes	Oui
Is must bit set and boot time elapsed?	Le <i>must bit</i> est-il établi et le temps d'initialisation (<i>boot time</i>) est-il écoulé ?
Object	Objet
End of start process boot NMT slave	Fin du processus de démarrage de l'initialisation de l'esclave NMT
Signal successfull boot or elapsed time	Indication d'une initialisation réussie ou du temps écoulé
Error status OK ?	Etat d'erreur OK ?
Inform application	Information de l'application
End sub process	Fin du sous-processus

Figure 15 – Processus « Démarrer le processus d'initialisation de l'esclave NMT »

Le processus *Démarrer le processus d'initialisation de l'esclave NMT* illustré à la Figure 15 doit inclure les étapes suivantes:.

- a) Démarrer le processus parallèle *Initialisation de l'esclave NMT*.
- b) Esclaves NMT obligatoires: attendre l'achèvement du processus *Initialisation de l'esclave NMT*.
Esclaves NMT facultatifs: attendre le signal de confirmation d'exécution du processus *Initialisation de l'esclave NMT*.

Le processus parallèle doit

- c) Exécuter le processus *Initialisation de l'esclave NMT* (voir 9.5).
- d) Produire un signal pour chaque tentative d'exécution du processus *Initialisation de l'esclave NMT*.

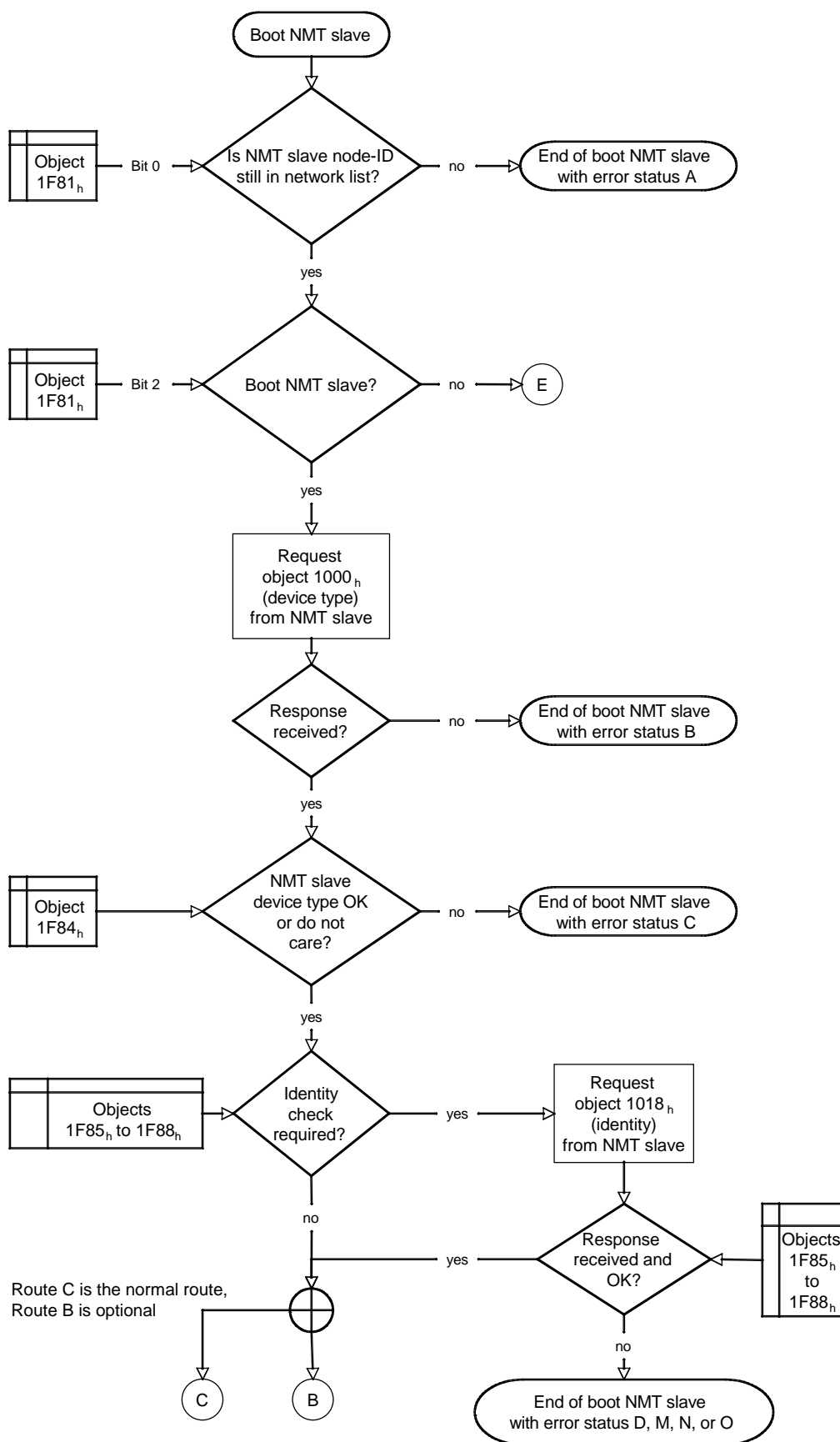
Si le processus *Initialisation de l'esclave NMT* recouvre l'état OK, le processus doit s'achever. Ce processus doit se poursuivre sans fin pour tout esclave NMT facultatif, jusqu'à ce que le processus *Initialisation de l'esclave NMT* s'achève avec l'état OK.

NOTE Le temps de cycle recommandé est de 1 s pour un débit binaire supérieur à 125 kbit/s.

Si le processus *Initialisation de l'esclave NMT* recouvre l'état d'erreur B pour les esclaves NMT obligatoires, et si le temps écoulé est supérieur à la valeur configurée de l'objet 1F89_h, l'application doit en être informée et ce sous-processus doit s'achever.

Le sous-processus du processus *Démarrer le processus Initialisation de l'esclave NMT* doit être asynchrone par rapport aux autres processus.

9.5 Initialisation de l'esclave NMT



Légende

Anglais	Français
Boot NMT slave	Initialisation de l'esclave NMT
Object	Objet
Is NMT slave node-ID still in network list?	L'indicatif de nœud de l'esclave NMT figure-t-il toujours dans la liste de réseau ?
no	non
End of boot NMT slave with error status A	Fin de l'initialisation de l'esclave NMT avec état d'erreur A
yes	Oui
Boot NMT slave?	Initialisation de l'esclave NMT ?
Request object 1000 _h (device type) from NMT slave	Demande de l'objet 1000 _h (type de dispositif) depuis l'esclave NMT
Response received?	Réception de la réponse ?
End of boot NMT slave with error status	Fin de l'initialisation de l'esclave NMT avec état d'erreur
NMT slave device type OK or do not care?	Type de dispositif de l'esclave NMT OK ou aucune importance ?
Identity check required?	Vérification de l'identité nécessaire ?
Request object.. (identity) from NMT slave	Demande de l'objet ... (identité) auprès de l'esclave NMT
Response received and OK?	Réponse reçue et OK ?
to	À
Route C is the normal route	Le parcours C est le parcours normal
Route B is optional	Le parcours B est facultatif
or	ou

Figure 16 – Initialisation de l'esclave NMT, partie 1

Le processus *Initialisation de l'esclave NMT*, illustré à la Figure 16, comprend les étapes suivantes:

- Le bit 0 de l'objet 1F81_h permet de déterminer si l'esclave NMT doit être traité ou si le processus doit s'achever par l'état d'erreur.
- Le bit 2 de l'objet 1F81_h permet de déterminer si l'esclave NMT doit être configuré et initialisé.
- Charger l'objet 1000_h à partir de l'esclave NMT. Le processus doit s'achever par l'état d'erreur en cas d'absence de réponse.
- Lorsque la valeur de l'objet 1F84_h pour l'esclave NMT est différente de 0, elle doit être vérifiée par rapport à l'objet 1000_h. Le processus doit s'achever par l'état d'erreur dans le cas où les deux valeurs sont différentes.
- Lorsque les valeurs des objets 1F85_h à 1F88_h sont différentes de 0, les valeurs des objets particuliers doivent être vérifiées par rapport à leurs valeurs correspondantes de l'objet 1018_h à partir de l'esclave NMT. Dans le cas où une des valeurs des objets 1F85_h à 1F88_h est différente des valeurs correspondantes de l'objet 1018_h à partir de l'esclave NMT, le processus doit s'achever par l'état d'erreur.

Le processus *Initialisation de l'esclave NMT* peut se poursuivre avec une partie facultative 2 (voir Figure 17), qui introduit deux fonctions facultatives:

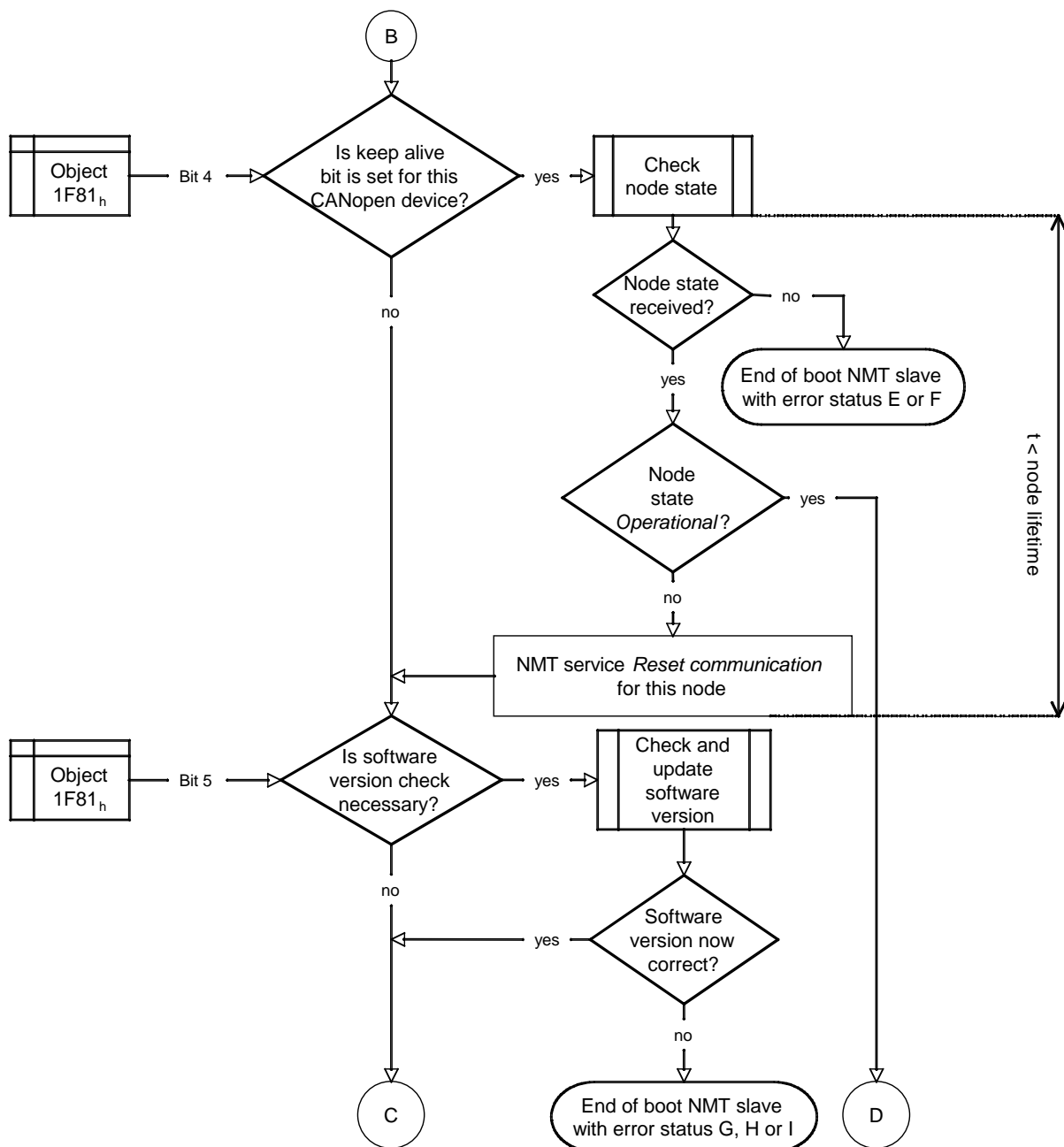
- maintenir sous tension les dispositifs CANopen initialement à l'état NMT *Opérationnel*, et
- gérer les versions logicielles d'application, y compris la mise à jour automatique du logiciel d'application.

Si le bit d'entretien d'un dispositif CANopen est mis à 1, le maître NMT ne doit pas exécuter les services NMT *Initialisation du nœud* et *Initialisation de la communication* pour ce dispositif CANopen.

NOTE 1 De telles situations peuvent se produire au cas où le maître NMT connaît une défaillance avec un redémarrage ultérieur, par exemple, panne d'alimentation.

Un contrôle de la version logicielle peut être utilisé dans les systèmes pour lesquels la version du logiciel d'application utilisé sur le dispositif CANopen fait l'objet d'une vérification d'exactitude. Le processus de vérification et de mise à jour de la version logicielle peut être utilisé pour télécharger automatiquement la version du logiciel d'application la plus récente. Les dispositifs CANopen qui détectent une erreur dans leur logiciel d'application, par exemple par le calcul d'une somme de contrôle lors du démarrage, peuvent forcer le téléchargement du logiciel d'application le plus récent en répondant par des informations erronées concernant la version de logiciel.

Les deux fonctions, ou une seule fonction, peuvent être mises en œuvre.



Légende

Anglais	Français
Object	Objet
Is keep alive bit set for this CANopen device?	Le bit d'entretien est-il mis à 1 pour ce dispositif CANopen ?
yes	oui
Check node state	Vérification de l'état de nœud
Node state received?	Réception de l'état de nœud ?
no	Non
End of boot NMT slave with error status ... or ..	Fin de l'initialisation de l'esclave NMT avec état d'erreur ... ou ...
Node state <i>operational</i> ?	Etat de nœud <i>Opérationnel</i> ?
Node lifetime	Durée de vie de nœud
NMT service Reset communication for this node	Service NMT <i>Initialisation de la communication</i> pour ce nœud
Is software version check necessary?	Le contrôle de la version logicielle est-il nécessaire ?
Check and update software version	Vérification et mise à jour de la version logicielle
Software version now correct?	Version logicielle correcte à présent?

Figure 17 – Initialisation de l'esclave NMT, partie 2

- f) Le bit 4 de l'objet 1F81_h permet de déterminer si le maintien sous tension est nécessaire. Si tel est le cas, le maître NMT doit demander l'état NMT actuel de l'esclave NMT pour lequel le processus *Initialisation de l'esclave NMT* est exécuté. Le processus doit s'achever par l'état d'erreur en cas de non-réception de l'état NMT actuel. Le processus doit se poursuivre avec D dans le cas où l'état NMT actuel est *Opérationnel*. Dans le cas où l'état NMT actuel n'est pas l'état NMT *Opérationnel*, le maître NMT doit exécuter le service NMT *Initialisation de la communication* pour l'esclave NMT concerné.

Le processus de vérification de l'état NMT est défini en 9.5.2.

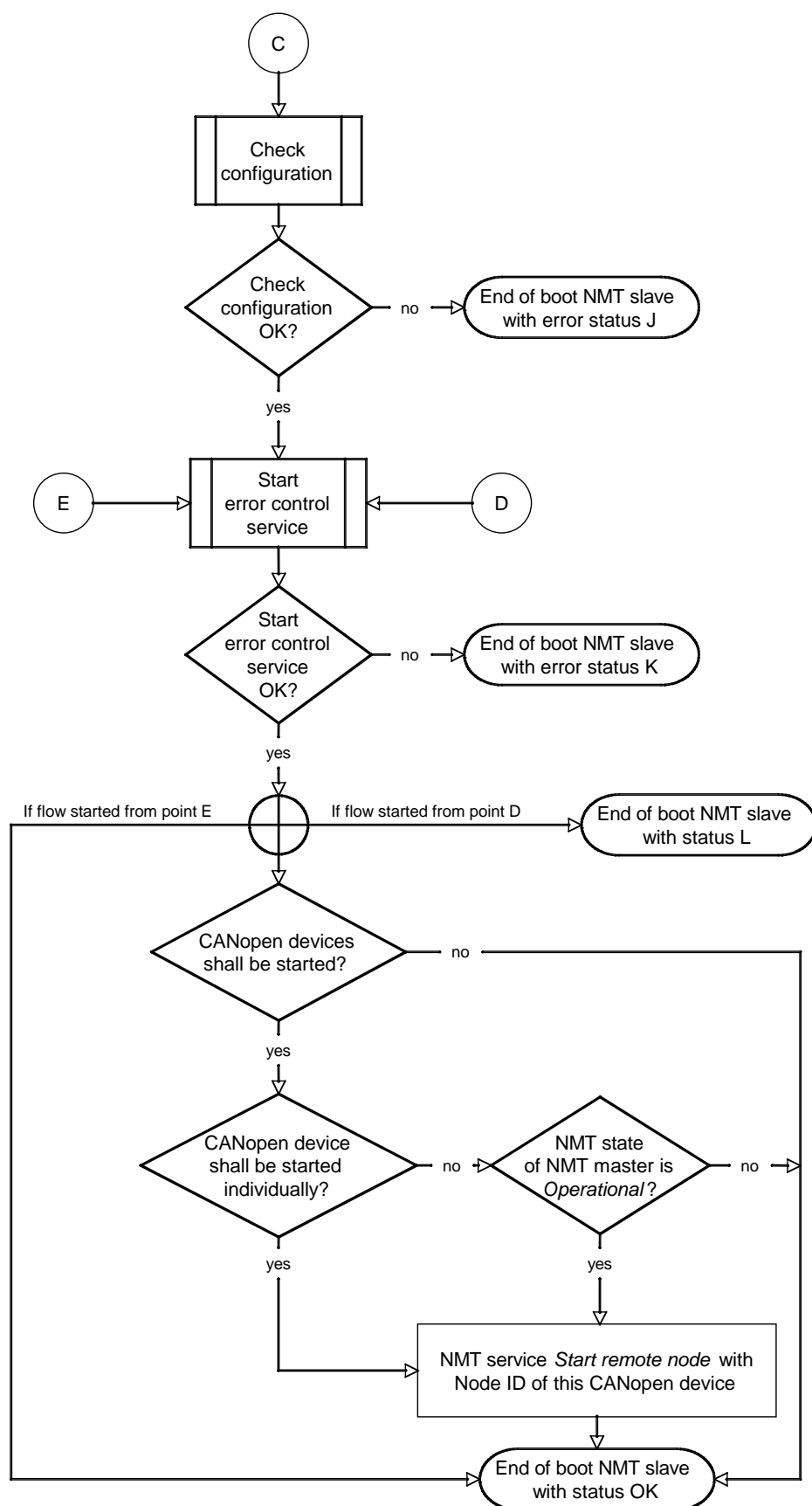
NOTE 2 Dans le cas où l'esclave NMT prend en charge uniquement la sauvegarde de nœud et de la durée de vie, le maître NMT veille à ce que le temps qui s'écoule entre la demande de l'état NMT actuel auprès de l'esclave NMT et l'exécution du service NMT *Initialisation de la communication* soit inférieur à la durée de vie utile définie dans l'esclave NMT.

- g) Le bit 5 de l'objet 1F81_h permet de déterminer si la vérification du logiciel d'application doit être effectuée.

Lorsque la vérification du logiciel d'application se révèle nécessaire, le processus de vérification et de mise à jour de la version logicielle doit être exécuté.

Dans le cas où la version logicielle demeure incorrecte, le processus *Initialisation de l'esclave NMT* doit s'achever par l'état d'erreur.

Le processus *Initialisation de l'esclave NMT* se poursuit dans la partie 3 (voir Figure 18); il est obligatoire.



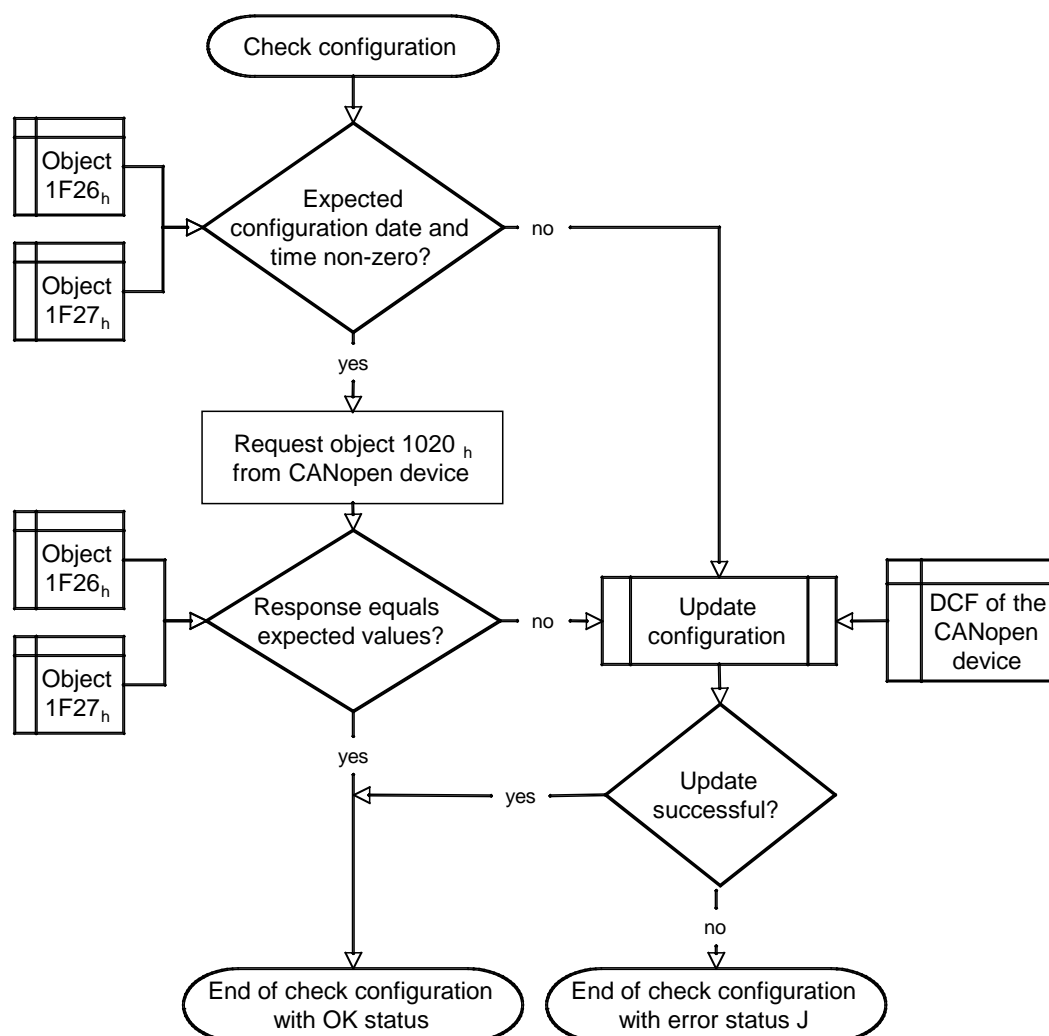
Légende

Anglais	Français
Check configuration	Vérification de la configuration
no	non
End of boot NMT slave with error status	Fin de l'initialisation de l'esclave NMT avec état d'erreur
yes	oui
Start error control service	Démarrage du service de contrôle d'erreurs
Start error control service OK?	Démarrage du service de contrôle d'erreurs OK ?
If flow started from point	Si démarrage du flux au point
End of boot NMT slave with status	Fin de l'initialisation de l'esclave NMT avec état
CANopen devices shall be started?	Les dispositifs CANopen doivent-ils être démarrés ?
CANopen device shall be started individually?	Les dispositifs CANopen doivent-ils être démarrés de manière individuelle ?
NMT state of NMT master is <i>Operational</i> ?	L'état NMT du maître NMT est-il <i>Opérationnel</i> ?
NMT service <i>Start remote node</i> with Node ID of this CANopen device	Service NMT <i>Démarrage du nœud distant</i> avec indicatif de nœud de ce dispositif CANopen

Figure 18 – Initialisation de l'esclave NMT, partie 3

- h) Le processus de vérification de la configuration (voir 9.5.1) doit être exécuté.
 - i) Dans le cas où le processus de vérification de la configuration s'est achevé sur un échec, le processus *Initialisation de l'esclave NMT* doit s'achever par un état d'erreur.
 - j) Le processus *Démarrage du contrôle d'erreurs* (voir 9.6.1) doit être exécuté.
 - k) Dans le cas où le processus *Démarrage du contrôle d'erreurs* s'est achevé sur un échec, le processus *Initialisation de l'esclave NMT* doit s'achever par un état d'erreur.
 - l) Dans le cas où l'esclave NMT est dans l'état *Opérationnel*, le processus *Initialisation de l'esclave NMT* doit s'achever avec succès.
 - m) Le bit 3 de l'objet 1F80_h permet de déterminer si le maître NMT doit exécuter le service NMT *Démarrage du nœud distant*.
 - n) Dans le cas où le maître NMT ne doit pas exécuter le service NMT *Démarrage du nœud distant*, le processus *Initialisation de l'esclave NMT* doit s'achever avec succès avec indication de l'état.
 - o) Le bit 1 de l'objet 1F80_h permet de déterminer si le maître NMT doit exécuter le service NMT *Démarrage du nœud distant* avec l'indicatif de nœud CANopen mis à 0 ou pour chaque esclave NMT dans le réseau de manière individuelle.
 - p) Dans le cas où le maître NMT doit exécuter le service NMT *Démarrage du nœud distant* avec l'indicatif de nœud CANopen mis à 0 et où ce même maître n'est pas à l'état NMT *Opérationnel*, le processus *Initialisation de l'esclave NMT* doit s'achever avec succès avec indication de l'état OK.
- NOTE 3 Si le maître NMT est à l'état NMT *Opérationnel*, il est considéré comme une indication de l'exécution du Démarrage du NMT initial préalablement au démarrage de cet esclave NMT.
- q) Dans le cas où le maître NMT exécute le service NMT *Démarrage du nœud distant* pour chaque esclave NMT dans le réseau de manière individuelle ou dans le cas où ce même maître exécute ledit service avec l'indicatif de nœud CANopen mis à 0 et où il est dans l'état NMT *Opérationnel*, il doit exécuter le service NMT *Démarrage du nœud distant* avec l'indicatif de nœud CANopen mis à la valeur appropriée.
 - r) Le processus *Initialisation de l'esclave NMT* doit s'achever avec succès avec indication de l'état.

9.5.1 Vérification de la configuration



Légende

Anglais	Français
Check configuration	Vérification de la configuration
Object	Objet
Expected configuration date and time non-zero?	Date et heure de configuration prévues non mises à 0 ?
no	non
yes	oui
Request object ... from CANopen device	Demande d'objet ... depuis le dispositif CANopen
Response equals expected values?	Réponse égale aux valeurs attendues ?
Update configuration	Mise à jour de la configuration
DCF of the CANopen device	DCF du dispositif CANopen
Update successful?	Mise à jour réussie ?
End of check configuration with OK status	Fin de la vérification de la configuration avec état OK
End of check configuration with error status	Fin de la vérification de la configuration avec état d'erreur

Figure 19 – Vérification de la configuration

Le processus *Vérification de la configuration* illustré à la Figure 19 comprend les étapes suivantes:

- Les entrées conformes (le sous-index doit être équivalent à l'indicatif de nœud CANopen) des objets 1F26_h et 1F27_h permettent de déterminer si la configuration des dispositifs CANopen fait l'objet d'une vérification. Si les valeurs sont égales à 0, le processus doit passer à l'étape d). Si les valeurs ne sont pas égales à 0, le processus doit passer à l'étape b).
- L'objet 1020_h doit être demandé auprès du dispositif CANopen.
- Les valeurs des entrées conformes (le sous-index doit être équivalent à l'indicatif de nœud CANopen) des objets 1F26_h et 1F27_h doivent être comparées aux valeurs de l'objet reçu 1020_h. Si les valeurs sont équivalentes, le processus doit s'achever par l'état OK.

Si les valeurs ne sont pas équivalentes, le processus doit passer à l'étape d).

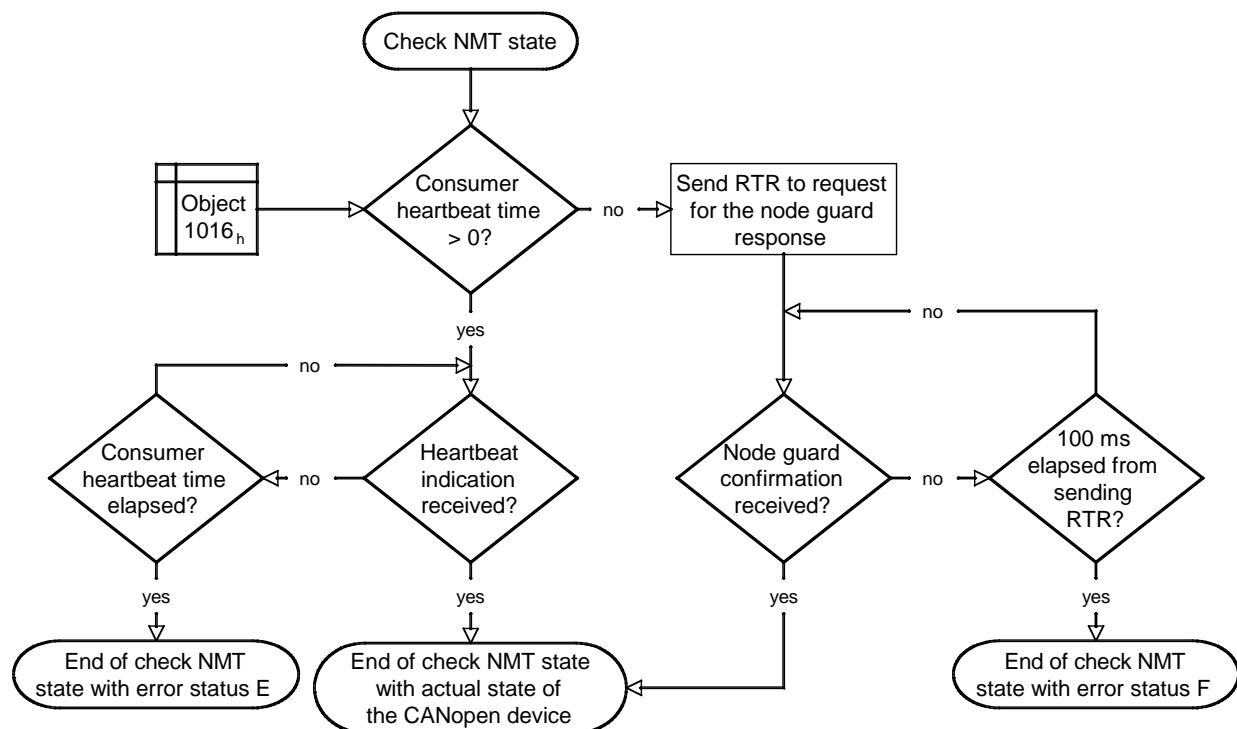
- La configuration doit être actualisée sur le dispositif CANopen. L'opération de restauration configurée doit être exécutée selon les paramètres de 1F81_h et 1F8A_h. Le service NMT Initialisation de la communication ou Initialisation du nœud pour l'esclave NMT correspondant doit être exécuté uniquement après l'opération de restauration.

La configuration peut être déduite de tout type de DCF prévu pour le dispositif CANopen concerné. Le DCF peut être lu à partir d'un système de fichiers local, objet 1F20_h, ou objet 1F22_h sur le gestionnaire de configuration.

- Si le téléchargement est achevé avec succès, le processus doit alors s'achever par l'état OK. Si le téléchargement ne s'est pas achevé avec succès, le processus doit alors s'achever par l'état d'erreur.

Le processus *Vérification de la configuration* peut être omis si l'état Maintien sous tension est activé pour le dispositif CANopen concerné et si ce dernier est dans l'état NMT *Opérationnel*. Si cela se produit, le processus Initialisation de l'esclave NMT doit s'achever par l'état d'erreur (L – « l'esclave NMT était opérationnel à l'origine »).

9.5.2 Vérification de l'état NMT



Légende

Anglais	Français
Check NMT state	Vérification de l'état NMT
Object	Objet
Consumer heartbeat time > 0?	Temps consommateur de pulsation > 0 ?
no	non
Send RTR to request for the node guard response	Envoi du RTR pour demande de la réponse relative à la protection de nœud
yes	Oui
Consumer heartbeat time elapsed?	Temps consommateur de pulsation écoulé ?
heartbeat indication received?	Réception de l'indication de la pulsation ?
Node guard confirmation received?	Réception de la confirmation de la protection de nœud ?
100 ms elapsed from sending RTR ?	Écoulement de 100 ms depuis l'envoi du RTR ?
End of check NMT state with error status	Fin de la vérification de l'état NMT avec état d'erreur
End of check NMT state with actual state of the CANopen device	Fin de la vérification de l'état NMT avec état réel du dispositif CANopen

Figure 20 – Vérification de l'état NMT¹

Le processus *Vérification de l'état NMT* illustré à la Figure 20 comprend les étapes suivantes:

- L'objet 1016_h (voir EN 50325-4) doit être utilisé pour déterminer si le dispositif CANopen est réglé pour la pulsation.
- Dans le cas où le dispositif CANopen est réglé pour la pulsation, le maître NMT doit vérifier s'il a reçu une indication de la pulsation dans le délai imparti (le temps consommateur de pulsation n'est pas écoulé). Si l'indication de la pulsation n'est pas reçue dans le délai imparti, le processus doit s'achever par un état d'erreur. Si l'indication de la pulsation est reçue, le processus doit s'achever par l'état NMT réel du dispositif CANopen faisant l'objet de la vérification.
- Dans le cas où le dispositif CANopen vérifié n'est pas réglé pour la pulsation, il est supposé que le dispositif CANopen est réglé pour la protection du dispositif CANopen. Dans ce cas, le maître NMT doit demander l'état NMT réel du dispositif CANopen faisant l'objet de la vérification, la protection du dispositif CANopen de service NMT doit être appliquée. En l'absence de réception d'une confirmation et si plus de 100 ms se sont écoulées, le processus doit s'achever par un état d'erreur. En cas de réception d'une confirmation, le processus doit s'achever par l'état NMT réel du dispositif CANopen faisant l'objet de la vérification.

9.5.3 Démarrage du maître volant NMT

La définition du maître volant NMT ne relève pas du domaine d'application de la présente norme.

NOTE Les définitions applicables à la fonctionnalité Maître NMT volant sont fournies dans le document CiA 302.

9.5.4 Etat d'erreur

Les erreurs indiquées dans le Tableau 14 doivent être signalées dans le maître NMT en cours de démarrage.

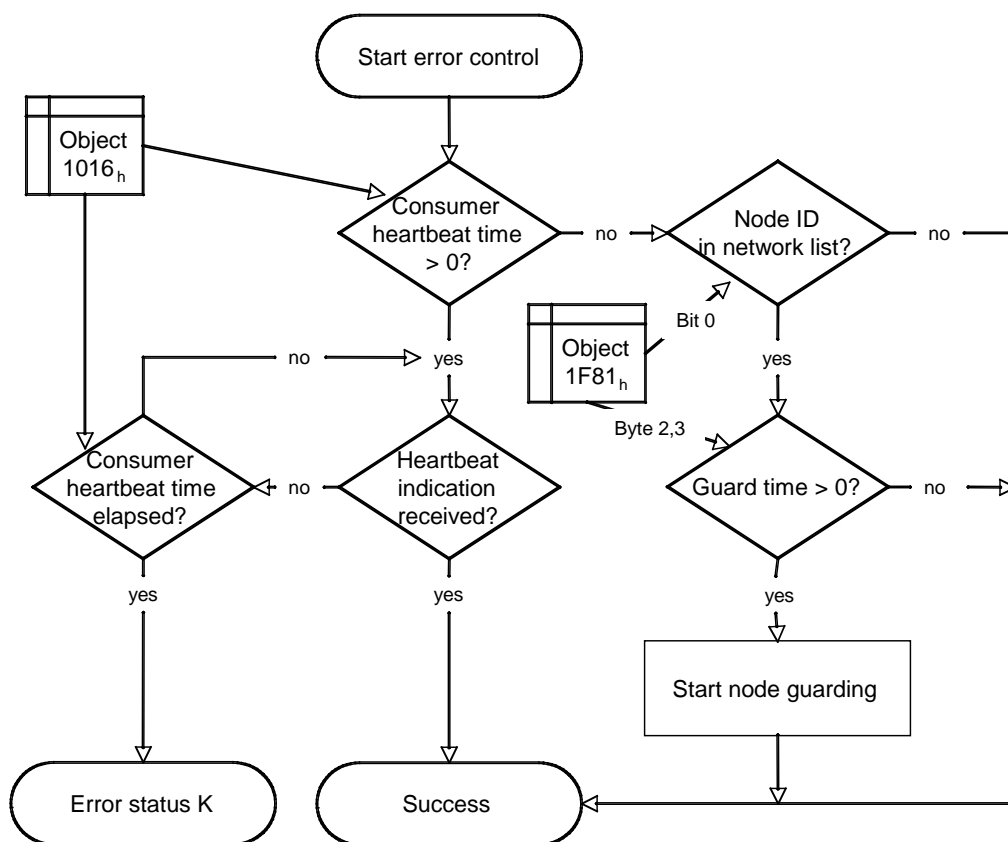
¹ Nécessaire, si le bit 4 de l'objet 1F81_h est mis à 1 (bit de maintien pris en charge).

Tableau 14 – Etat d'erreur

Etat d'erreur	Description
A	Le dispositif CANopen n'est pas énuméré dans l'objet 1F81 _h .
B	Aucune réponse reçue pour la demande de téléchargement de l'objet 1000 _h .
C	La valeur de l'objet 1000 _h du dispositif CANopen est différente de la valeur de l'objet 1F84 _h (Type de dispositif).
D	La valeur de l'objet 1018 _h sous-index 01 _h du dispositif CANopen est différente de la valeur de l'objet 1F85 _h (indicatif du fournisseur).
E	Événement de pulsation. Aucun message de pulsation reçu du dispositif CANopen.
F	Événement de protection de nœud. Aucune confirmation de demande de protection reçue du dispositif CANopen.
G	Objet 1F53 _h ou objet 1F54 _h non configuré pour le dispositif CANopen concerné.
H	Les valeurs de l'objet 1F52 _h par rapport à ce dispositif CANopen sont différentes de la valeur de l'objet 1F53 _h ou de la valeur de l'objet 1F54 _h , le bit 6 de l'objet 1F81 _h n'étant par ailleurs pas mis à 1.
I	Les valeurs de l'objet 1F52 _h par rapport à ce dispositif CANopen sont différentes de la valeur de l'objet 1F53 _h ou de la valeur de l'objet 1F54 _h , le téléchargement n'ayant par ailleurs pas abouti.
J	Non aboutissement du téléchargement de configuration.
K	Événement de pulsation lors du service « Démarrage du contrôle d'erreurs. Aucun message de pulsation reçu du dispositif CANopen lors du service « Démarrage du contrôle d'erreurs.
L	Esclave NMT opérationnel à l'origine. (le gestionnaire CANopen peut reprendre ses activités avec d'autres dispositifs CANopen)
M	La valeur de l'objet 1018 _h , sous-index 02 _h , du dispositif CANopen est différente de la valeur de l'objet 1F86 _h (code produit).
N	La valeur de l'objet 1018 _h , sous-index 03 _h , du dispositif CANopen est différente de la valeur de l'objet 1F87 _h (numéro de révision).
O	La valeur de l'objet 1018 _h , sous-index 04 _h , du dispositif CANopen est différente de la valeur de l'objet 1F88 _h (numéro de série).

9.6 Contrôle d'erreurs

9.6.1 Démarrage du contrôle d'erreurs



Légende

Anglais	Français
Start error control	Démarrage du contrôle d'erreurs
Object	Objet
Consumer heartbeat time >0?	Temps consommateur de pulsation > 0
no	Non
yes	Oui
Node ID in network list?	Indicatif de nœud figurant dans la liste réseau ?
Consumer heartbeat time elapsed?	Temps consommateur de pulsation écoulé ?
Heartbeat indication received?	Réception de l'indication de la pulsation ?
Guard time	Temps de protection
Start node guarding	Démarrage de la protection de nœud
Error status	État d'erreur
Success	Succès

Figure 21 – Démarrage du contrôle d'erreurs

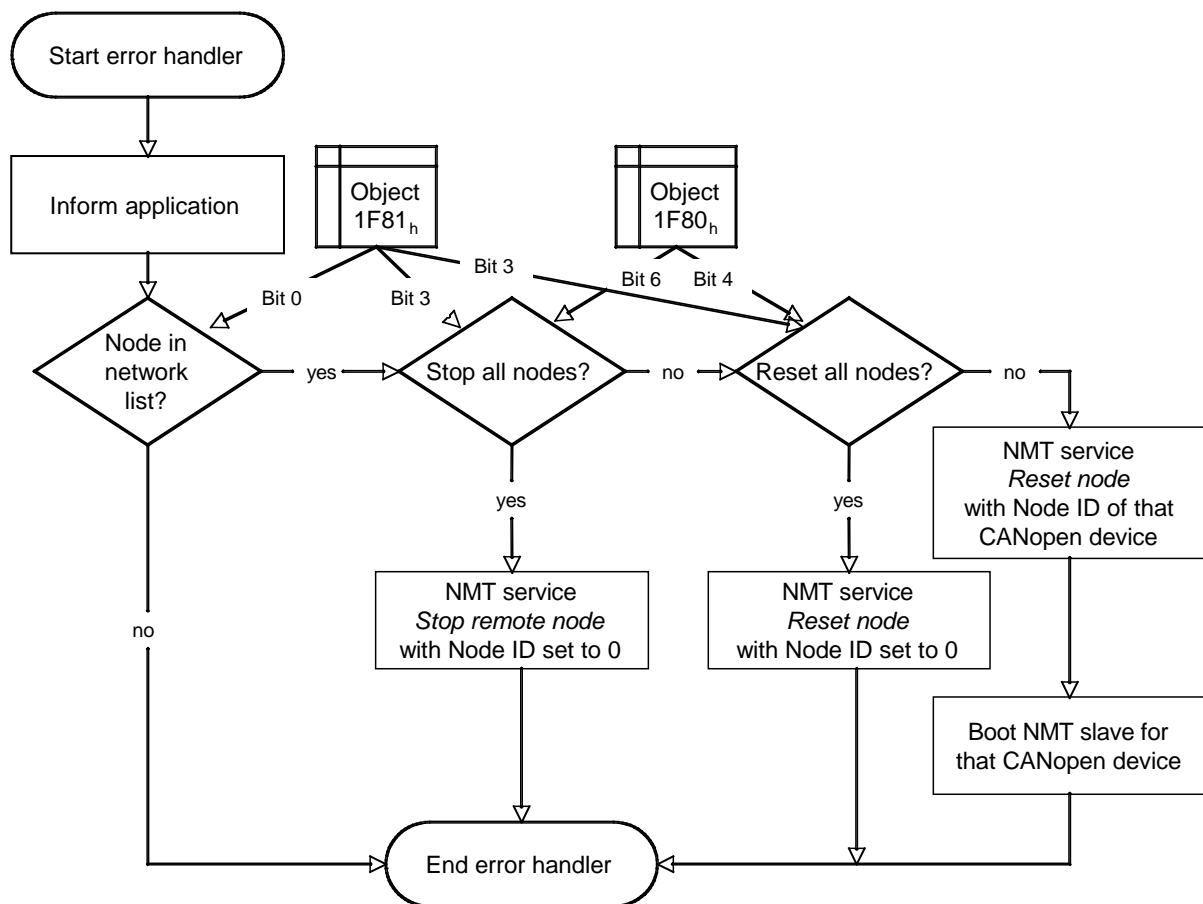
Le processus *Démarrage du contrôle d'erreurs*, comme illustré à la Figure 21, comprend les étapes suivantes:

- L'objet 1016h (voir EN 50325-4) doit être vérifié pour déterminer si le dispositif CANopen est réglé pour la pulsation.
- Dans le cas où le dispositif CANopen est réglé pour la pulsation et où aucune indication de pulsation n'est reçue dans le délai imparti, le processus doit d'achever par un état

d'erreur. Le temporisateur utilisé pour la vérification de la temporisation doit être lancé conjointement au processus lui-même.

- c) Dans le cas où le dispositif CANopen est réglé pour la pulsation et où une indication de pulsation est reçue dans le délai imparti, le processus doit d'achever avec succès.
- d) Dans le cas où le dispositif CANopen n'est pas réglé pour la pulsation, le bit 0 de l'objet 1F81h permet de déterminer si le dispositif CANopen doit être protégé.
- e) Dans le cas où le dispositif CANopen ne figure pas dans la liste de réseau, le processus doit s'achever avec succès.
- f) Dans le cas où le dispositif CANopen figure dans la liste de réseau, la valeur des octets 2 et 3 de l'objet 1F81h est utilisée pour déterminer si le dispositif CANopen doit être protégé.
- g) Dans le cas où la valeur est supérieure à 0, la protection doit être initialisée pour le dispositif CANopen concerné et le processus doit s'achever avec succès.
- h) Dans le cas où la valeur est nulle, le processus doit s'achever avec succès.

9.6.2 Gestionnaire de traitement d'erreur

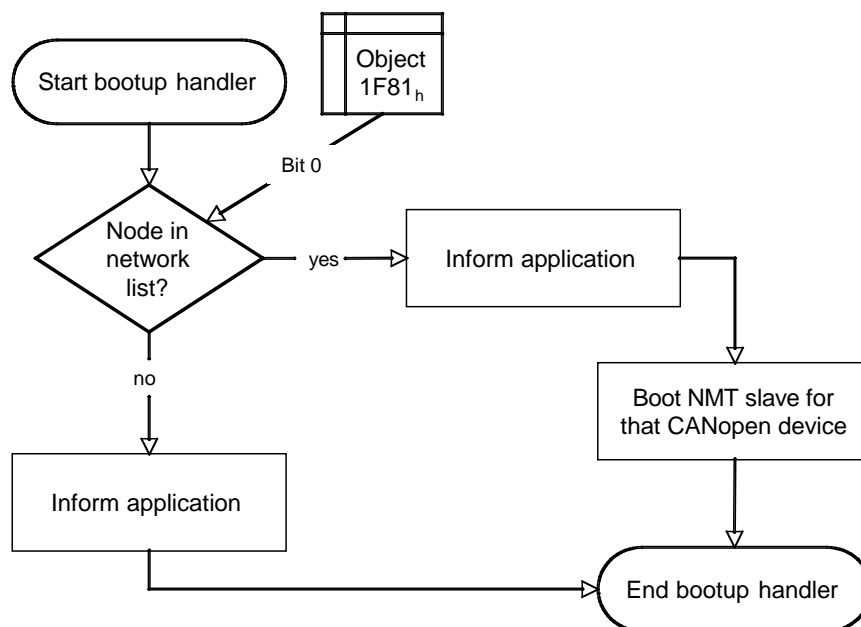


Légende

Anglais	Français
Start error handler	Démarrage du gestionnaire de traitement d'erreur
Inform application	Information de l'application
Object	Objet
Node in network list?	Nœud figurant dans la liste de réseau ?
yes	oui
no	Non
Stop all nodes?	Arrêt de tous les nœuds ?
Reset all nodes?	Initialisation de tous les nœuds ?
NMT service <i>Reset node</i> with Node ID of that CANopen device	Service NMT <i>Initialisation de nœud</i> avec indicatif de nœud du dispositif CANopen concerné
NMT service <i>Stop remote node</i> with Node ID set to 0	Service NMT <i>Arrêt de nœud distant</i> avec indicatif de nœud mis à 0
NMT service <i>Reset node</i> with Node ID set to 0	Service NMT <i>Initialisation de nœud</i> avec indicatif de nœud mis à 0
Boot NMT slave for that CANopen device	Initialisation de l'esclave NMT pour le dispositif CANopen concerné
End error handler	Arrêt du gestionnaire de traitement d'erreur

Figure 22 – Gestionnaire de traitement d'erreur

Le processus *Gestionnaire de traitement d'erreur* défini à la Figure 22 doit être déclenché à chaque occurrence d'un événement d'erreur NMT.

9.6.3 Gestionnaire de traitement d'initialisation

Légende

Anglais	Français
Start bootup handler	Démarrage du gestionnaire de traitement d'initialisation
Object	Objet
Node in network list?	Nœud figurant dans liste de réseau ?
yes	Oui
no	Non
Inform application	Information de l'application
Boot NMT slave for that CANopen device	Initialisation de l'esclave NMT pour le dispositif CANopen concerné
End bootup handler	Arrêt du gestionnaire de traitement d'initialisation

Figure 23 – Gestionnaire de traitement d'initialisation

Le processus *Gestionnaire de traitement d'initialisation* défini à la Figure 23 doit être déclenché à chaque occurrence d'un événement initialisation (voir EN 50325-4).

9.7 Services et protocoles de maître NMT supplémentaires

La définition des services et protocoles de maître NMT supplémentaires ne relève pas du domaine d'application de la présente spécification.

NOTE Des services et protocoles de maître NMT supplémentaires tels que la négociation ou les services de détection de maître NMT sont fournis dans le document CiA 302.

9.8 Entrées du dictionnaire d'objets

9.8.1 Objet 1020_h: Vérification de configuration

Cet objet doit indiquer la date et l'heure de téléchargement de la configuration. Si un dispositif CANopen prend en charge la sauvegarde des paramètres dans une mémoire non volatile, un outil de configuration de réseau ou un gestionnaire CANopen utilise cet objet pour vérifier la configuration après une réinitialisation du dispositif CANopen et pour vérifier si une reconfiguration se révèle nécessaire. L'outil de configuration enregistre la date et l'heure dans cet objet et enregistre les mêmes valeurs dans le DCF. Il permet alors au dispositif CANopen de sauvegarder sa configuration en imputant à l'index 1010_h sous-index 01_h, la signature "sauvegarde". Après une initialisation, le dispositif CANopen doit restaurer la dernière configuration et la signature de manière automatique, ou sur requête. Si toute autre commande modifie les valeurs de configuration de l'initialisation, le dispositif CANopen doit mettre à 0 l'objet Vérification de configuration.

Le gestionnaire de configuration compare la signature et la configuration à la valeur du DCF et détermine si une reconfiguration est nécessaire ou non.

NOTE L'application de cet objet permet d'accélérer le processus d'initialisation de manière significative. L'intégrateur de système, s'il est utilisé, considère qu'un utilisateur modifie une valeur de configuration, puis active la commande Enregistrer la configuration 1010_h sans modifier la valeur de 1020_h. L'intégrateur de système garantit ainsi une utilisation ultérieure à 100 % de cette fonction.

Le sous-index 01_h (date de configuration) doit comporter le nombre de jours depuis le 1er janvier 1984. Le sous-index 02_h (heure de configuration) doit correspondre au nombre de ms après minuit. Les Tableaux 15 et 16 spécifient la description d'objet et la description d'entrée.

Tableau 15 – Description d'objet

Attribut	Valeur
Index	1020 _h
Nom	Vérification de la configuration
Code d'objet	ARRAY
Type de données	UNSIGNED32
Catégorie	Facultatif

Tableau 16 – Description d'entrée

Attribut	Valeur
Sous-index	00 _h
Description	Sous-index le plus élevé pris en charge
Catégorie d'entrée	Obligatoire
Accès	const
Mise en correspondance PDO	Non
Plage de valeurs	02 _h
Valeur par défaut	02 _h
Sous-index	01 _h
Description	Date de configuration
Catégorie d'entrée	Obligatoire
Accès	rw
Mise en correspondance PDO	Non
Plage de valeurs	UNSIGNED32
Valeur par défaut	Spécifique au fabricant
vers	
Sous-index	02 _h
Description	Heure de configuration
Catégorie d'entrée	Obligatoire
Accès	rw
Mise en correspondance PDO	Non
Plage de valeurs	UNSIGNED32
Valeur par défaut	Spécifique au fabricant

9.8.2 Objet 102A_h: Temps de blocage NMT

Cet objet doit indiquer le temps de blocage configuré entre deux messages NMT subséquents. Les services NMT en cours doivent être mis en file d'attente et doivent être exécutés dans l'ordre de leur occurrence compte tenu du temps de blocage configuré. Les Tableaux 17 et 18 spécifient la description d'objet et la description d'entrée.

La valeur doit être indiquée en multiples de 100 µs. La valeur 0 doit désactiver le temps de blocage.

Tableau 17 – Description d'objet

Attribut	Valeur
Index	102A _h
Nom	Temps de blocage NMT
Code d'objet	VAR
Type de données	UNSIGNED16
Catégorie	Obligatoire

Tableau 18 – Description d'entrée

Attribut	Valeur
Sous-index	00 _h
Accès	rw
Mise en correspondance PDO	Non
Plage de valeurs	UNSIGNED16
Valeur par défaut	0

9.8.3 Objet 1F20_h: Enregistrement DCF

Cet objet doit être utilisé pour sauvegarder la configuration actuelle d'un dispositif CANopen spécifique dans le réseau. Le Tableau 19 spécifie la description d'objet et le Tableau 20 spécifie la description d'entrée.

Le sous-index d'une entrée doit correspondre à l'indicatif de nœud CANopen du dispositif CANopen dans le réseau. Le sous-index d'entrée correspondant à son propre indicatif de nœud CANopen doit être utilisé pour la configuration automatique. Si aucun DCF n'est sauvegardé précédemment, un message d'annulation SDO doit répondre à un accès en lecture (code d'erreur: 0800 0024_h ou 0800 0000_h). La définition du format DCF ne relève pas du domaine d'application de la présente spécification.

NOTE Le format DCF est fourni dans le document CiA 302.

Tableau 19 – Description d'objet

Attribut	Valeur
Index	1F20 _h
Nom	Enregistrement DCF
Code d'objet	ARRAY
Type de données	DOMAIN
Catégorie	Facultatif

Tableau 20 – Description d'entrée

Attribut	Valeur
Sous-index	00 _h
Description	Nombre d'entrées
Catégorie d'entrée	Obligatoire
Accès	const
Mise en correspondance PDO	Non
Plage de valeurs	7Fh
Valeur par défaut	7Fh
Sous-index	01 _h
Description	Indicatif de nœud CANopen 1
Catégorie d'entrée	Obligatoire
Accès	rw
Mise en correspondance PDO	Non
Plage de valeurs	<i>Voir définition de valeur</i>
Valeur par défaut	Spécifique au fabricant
vers	
Sous-index	7F _h
Description	Indicatif de nœud CANopen 127
Catégorie d'entrée	Obligatoire
Accès	rw
Mise en correspondance PDO	Non
Plage de valeurs	<i>Voir définition de valeur</i>
Valeur par défaut	Spécifique au fabricant

9.8.4 Objet 1F22_h: DCF concis

Cet objet doit être utilisé pour sauvegarder la configuration actuelle d'un dispositif CANopen spécifique dans le réseau. Le Tableau 21 spécifie la description d'objet et le Tableau 22 spécifie la description d'entrée.

Le sous-index d'une entrée doit correspondre à l'indicatif de nœud CANopen du dispositif CANopen dans le réseau. Le sous-index d'entrée correspondant à son propre indicatif de nœud CANopen doit être utilisé pour la configuration automatique. Le format du Fichier de configuration des dispositifs (DCF) doit être comme défini à la Figure 24. Si aucun DCF n'a été sauvegardé préalablement, le nombre d'entrées doit être égal à 0. Si le DCF pour une entrée spécifique doit être supprimé, le nombre d'entrées doit être mis à 0.

Nombre (n) d'entrées		UNSIGNED32
Entrée 1	Index	UNSIGNED16
	Sous-index	UNSIGNED8
	Taille (m) des données en octets	UNSIGNED32
	Octet de données 1	UNSIGNED8
	Octet de données 2	UNSIGNED8

	Octet de données m	UNSIGNED8
Entrée 2	Index	UNSIGNED16
	Sous-index	UNSIGNED8
	Taille (m) des données en octets	UNSIGNED32
	Octet de données 1	UNSIGNED8
	Octet de données 2	UNSIGNED8

	Octet de données m	UNSIGNED8
.....		
Entrée n	Index	UNSIGNED16
	Sous-index	UNSIGNED8
	Taille (m) des données en octets	UNSIGNED32
	Octet de données 1	UNSIGNED8
	Octet de données 2	UNSIGNED8

	Octet de données m	UNSIGNED8

Figure 24 – Définition du flux de données d'un DCF concis

Tableau 21 – Description d'objet

Attribut	Valeur
Index	1F22 _n
Nom	DCF concis
Code d'objet	ARRAY
Type de données	DOMAIN
Catégorie	Facultatif

Tableau 22 – Description d'entrée

Attribut	Valeur
Sous-index	00 _h
Description	Nombre d'entrées
Catégorie d'entrée	Obligatoire
Accès	const
Mise en correspondance PDO	Non
Plage de valeurs	7F _h
Valeur par défaut	7F _h
Sous-index	01 _h
Description	Indicatif de nœud CANopen 1
Catégorie d'entrée	Obligatoire
Accès	rw
Mise en correspondance PDO	Non
Plage de valeurs	Voir définition
Valeur par défaut	Spécifique au fabricant
vers	
Sous-index	7F _h
Description	Indicatif de nœud CANopen 127
Catégorie d'entrée	Obligatoire
Accès	rw
Mise en correspondance PDO	Non
Plage de valeurs	Voir définition
Valeur par défaut	Spécifique au fabricant

9.8.5 Objet 1F26_h: Date de configuration prévue

Cet objet doit être utilisé pour vérifier la date de configuration des dispositifs CANopen dans le réseau.

La date de configuration (voir 9.8.1, objet 1020_h sous-index 01_h) des dispositifs CANopen dans le réseau doit être adaptée à la valeur de cet objet dans le cas où cette dernière n'est pas nulle. Dans le cas contraire (valeur nulle), le dispositif CANopen doit être configuré.

Le sous-index d'une entrée doit correspondre à l'indicatif de nœud CANopen des dispositifs CANopen dans le réseau.

Tableau 23 – Description d'objet

Attribut	Valeur
Index	1F26 _h
Nom	Date de configuration prévue
Code d'objet	ARRAY
Type de données	UNSIGNED32
Catégorie	Facultatif

Tableau 24 – Description d'entrée

Attribut	Valeur
Sous-index	00 _h
Description	Nombre d'entrées
Catégorie d'entrée	Obligatoire
Accès	const
Mise en correspondance PDO	Non
Plage de valeurs	7F _h
Valeur par défaut	7F _h
Sous-index	01 _h
Description	Indicatif de nœud CANopen 1
Catégorie d'entrée	Obligatoire
Accès	rw
Mise en correspondance PDO	Non
Plage de valeurs	Voir définition
Valeur par défaut	Spécifique au fabricant
vers	
Sous-index	7F _h
Description	Indicatif de nœud CANopen 127
Catégorie d'entrée	Obligatoire
Accès	rw
Mise en correspondance PDO	Non
Plage de valeurs	Voir définition
Valeur par défaut	Spécifique au fabricant

9.8.6 Objet 1F27_h: Heure de configuration prévue

Cet objet doit être utilisé pour vérifier l'heure de configuration des dispositifs CANopen dans le réseau.

L'heure de configuration des dispositifs CANopen (objet 1020_h, sous-index 02_h) dans le réseau doit être adaptée à la valeur de cet objet dans le cas où cette dernière n'est pas nulle. Dans le cas contraire (valeur nulle), le dispositif CANopen doit être configuré.

Le sous-index d'une entrée doit correspondre à l'indicatif de nœud CANopen des dispositifs CANopen dans le réseau.

NOTE Les descriptions d'objet et d'entrée de l'index 1020_h sous-index 02_h sont fournies dans la toute dernière version CANopen (document CiA 301).

Tableau 25 – Description d'objet

Attribut	Valeur
Index	1F27 _h
Nom	Heure de configuration prévue
Code d'objet	ARRAY
Type de données	UNSIGNED32
Catégorie	Facultatif

Tableau 26 – Description d'entrée

Attribut	Valeur
Sous-index	00 _h
Description	Nombre d'entrées
Catégorie d'entrée	Obligatoire
Accès	const
Mise en correspondance PDO	Non
Plage de valeurs	7F _h
Valeur par défaut	7F _h
Sous-index	01 _h
Description	Indicatif de nœud CANopen 1
Catégorie d'entrée	Obligatoire
Accès	rw
Mise en correspondance PDO	Non
Plage de valeurs	Voir définition de valeur
Valeur par défaut	Spécifique au fabricant
vers	
Sous-index	7F _h
Description	Indicatif de nœud CANopen 127
Catégorie d'entrée	Obligatoire
Accès	rw
Mise en correspondance PDO	Non
Plage de valeurs	Voir définition
Valeur par défaut	Spécifique au fabricant

9.8.7 Objet 1F80_h: Démarrage du NMT

Cet objet doit configurer le comportement au démarrage d'un dispositif CANopen. Les transitions d'état internes ne doivent pas modifier la valeur de cet objet. Un message d'annulation (code d'annulation 0800 0000_h ou 0609 0030_h) doit répondre à toute tentative de modification d'un bit d'une fonctionnalité non prise en charge par le dispositif CANopen. Les Figures 25 et 26 définissent la structure orientée bit de la valeur. Les Tableaux 27, 28, 29, 30, 31, 32 et 33 définissent les valeurs admises. Le Tableau 34 définit les exceptions des dispositifs aptes au démarrage. Les Tableaux 35 et 36 spécifient la description d'objet et la description d'entrée.

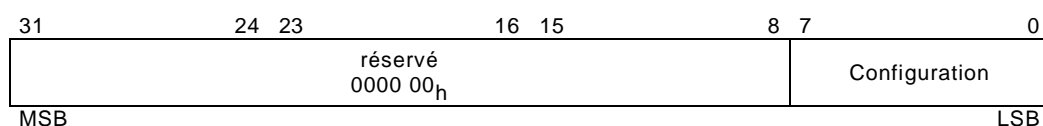
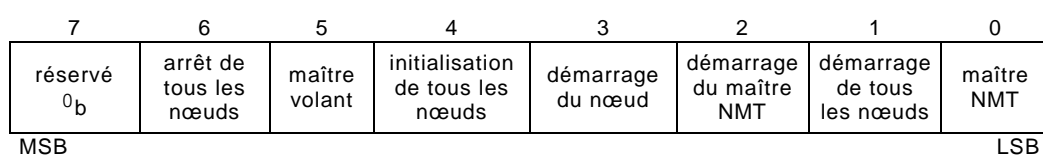
**Figure 25 – Structure d'objet****Figure 26 – Structure de bit de la valeur de configuration**

Tableau 27 – Valeur Maître NMT (bit: 0)

Valeur	Description
0 _b	Le dispositif CANopen n'est pas le maître NMT. Les entrées de l'objet 1F81 _h doivent être ignorées. Tous les autres bits de l'objet 1F80 _h doivent être ignorés avec les exceptions définies dans le Tableau 34.
1 _b	Le dispositif CANopen est le maître NMT

Tableau 28 – Valeur Démarrage de tous les nœuds (bit: 1)

Valeur	Description
0 _b	Service NMT <i>Démarrage du nœud distant</i> pour chaque indicatif de nœud CANopen
1 _b	Service NMT <i>Démarrage du nœud distant</i> avec indicatif de nœud CANopen = 0

Tableau 29 – Valeur Démarrage du Maître NMT (bit: 2)

Valeur	Description
0 _b	Doit commuter sur l'état NMT <i>Opérationnel</i> dans le processus Démarrage du NMT (voir Figure 13)
1 _b	Ne doit pas commuter sur l'état NMT <i>Opérationnel</i> de lui-même.

Tableau 30 – Valeur Démarrage du nœud (bit: 3)

Valeur	Description
0 _b	Le maître NMT doit démarrer les esclaves NMT.
1 _b	Le maître NMT ne doit pas démarrer les esclaves NMT, l'application pouvant par ailleurs démarrer ces derniers.

Tableau 31 – Initialisation de tous les nœuds (bit: 4)

Valeur	Description
0 _b	Dans le cas d'un événement contrôle d'erreurs d'un dispositif CANopen défini comme obligatoire (voir objet 1F81 _h), le service NMT <i>Initialisation du nœud</i> avec l'indicatif de nœud CANopen du dispositif CANopen à l'origine dudit événement, doit être exécuté.
1 _b	Dans le cas d'un événement contrôle d'erreurs d'un dispositif CANopen défini comme obligatoire (voir objet 1F81 _h), le service NMT <i>Initialisation du nœud</i> avec l'indicatif de nœud CANopen = 0 doit être exécuté.

Tableau 32 – Maître volant (bit: 5)

Valeur	Description
0 _b	Le dispositif CANopen ne doit pas prendre part à la négociation du maître volant NMT
1 _b	Réservé (pour les dispositifs avec la capacité de maître volant NMT)

NOTE La définition de la fonctionnalité Maître volant NMT ne relève pas du domaine d'application de la présente norme. Des définitions sont fournies dans le document CiA 302.

Tableau 33 – Arrêt de tous les nœuds (bit: 6)

Valeur	Description
0 _b	Dans le cas d'un événement contrôle d'erreurs d'un dispositif CANopen défini comme obligatoire (<i>voir objet 1F81_h</i>), l'action définie par le bit 4 doit être exécutée.
1 _b	Dans le cas d'un événement contrôle d'erreurs d'un dispositif CANopen défini comme obligatoire (<i>voir objet 1F81_h</i>), le service NMT <i>Arrêt du nœud distant</i> avec l'indicatif de nœud CANopen = 0 doit être exécuté. Le bit 4 doit être ignoré.

Tableau 34 – Exceptions pour les dispositifs capables de démarrage NMT

Valeur	Description
00000000 00000000 00000000 00001000 _b	Esclave NMT qui doit adopter de manière autonome l'état NMT <i>Opérationnel</i> après l'état NMT <i>Initialisation</i> (démarrage automatique)
00000000 00000000 00000000 00000010 _b	Esclave NMT qui doit exécuter le service NMT <i>Démarrage du nœud distant</i> avec l'indicatif de nœud CANopen mis à 0

Tableau 35 – Description d'objet

Attribut	Valeur
Index	1F80 _h
Nom	Démarrage du NMT
Code d'objet	VAR
Type de données	UNSIGNED32
Catégorie	Conditionnel; Obligatoire si le dispositif CANopen est un gestionnaire CANopen ou un dispositif CANopen de démarrage.

Tableau 36 – Description d'entrée

Attribut	Valeur
Sous-index	00 _h
Accès	rw
Mise en correspondance PDO	Non
Plage de valeurs	Voir définition de valeur
Valeur par défaut	Spécifique au fabricant

9.8.8 Objet 1F81_h: Attribution de l'esclave NMT

Cet objet doit attribuer des dispositifs CANopen au gestionnaire CANopen, qui représente le dispositif qui doit mettre en œuvre cet objet. Chaque sous-index de cet objet doit correspondre à l'indicatif de nœud CANopen du dispositif CANopen correspondant dans le réseau. Le sous-index correspondant à son propre indicatif de nœud CANopen doit être ignoré. Un message d'annulation (code d'annulation 0800 0000_h ou 0609 0030_h) doit répondre à toute tentative de modification d'un bit d'une fonctionnalité non prise en charge par le dispositif CANopen. Les Figures 27 et 28 définissent la structure orientée bit de la valeur. Les Tableaux 37, 38, 39, 40, 41, 42 et 43 définissent le contenu des valeurs. Les Tableaux 44 et 45 spécifient la description d'objet et la description d'entrée.

La valeur relative au facteur de ré-essai indique le nombre de ré-essais que le maître NMT doit exécuter dans le cas d'un événement protection de nœud. La valeur 0 doit désactiver la protection de nœud pour le dispositif CANopen.

La valeur relative au temps de protection doit indiquer le temps de cycle afférent à la protection de nœud du dispositif CANopen. La valeur doit être indiquée en multiples de ms. La valeur 0 doit désactiver la protection de nœud pour le dispositif CANopen.

NOTE Si l'objet consommateur de pulsation est configuré à une valeur non nulle, le mécanisme de pulsation est prioritaire sur la protection de nœud.

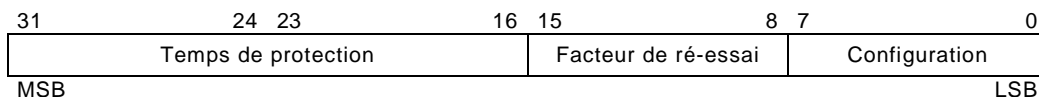


Figure 27 – Structure d'objet de la valeur

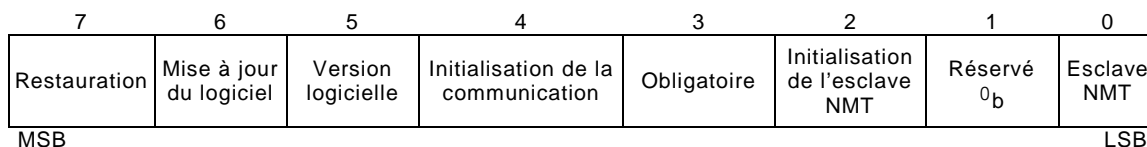


Figure 28 – Structure de bit de la valeur de configuration

Tableau 37 – Esclave NMT (bit: 0)

Valeur	Description
0_b	Maître NMT ou non disponible dans le réseau
1_b	Esclave NMT et disponible dans le réseau

Tableau 38 – Esclave d'initialisation NMT (bit: 2)

Valeur	Description
0_b	La configuration et le service NMT <i>Démarrage du nœud distant</i> ne doivent pas être autorisés en cas d'événement contrôle d'erreurs ou service NMT <i>Initialisation</i> NOTE L'application est chargée du démarrage de l'esclave NMT.
1_b	La configuration et le service NMT <i>Démarrage du nœud distant</i> doivent être exécutés en cas d'événement contrôle d'erreurs ou service NMT <i>Initialisation</i>

Tableau 39 – Obligatoire (bit: 3)

Valeur	Description
0_b	Un dispositif CANopen peut être présent préalablement au démarrage du réseau (dispositif CANopen facultatif)
1_b	Un dispositif CANopen doit être présent préalablement au démarrage du réseau (dispositif CANopen obligatoire)

Tableau 40 – Initialisation de la communication (bit: 4)

Valeur	Description
0_b	Le service NMT <i>Initialisation de la communication</i> peut être exécuté pour le dispositif CANopen à tout moment
1_b	Le service NMT <i>Initialisation de la communication</i> ne doit pas être exécuté pour le dispositif CANopen dans le cas où ce dernier est à l'état NMT <i>Opérationnel</i>

Tableau 41 – Version logicielle (bit: 5)

Valeur	Description
0 _b	La vérification de la version logicielle ne doit pas être effectuée pour le dispositif CANopen
1 _b	La vérification de la version logicielle doit être effectuée pour le dispositif CANopen

Tableau 42 – Mise à jour du logiciel (bit: 6)

Valeur	Description
0 _b	La mise à jour du logiciel ne doit pas être effectuée pour le dispositif CANopen
1 _b	La mise à jour du logiciel doit être effectuée pour le dispositif CANopen

Tableau 43 – Restauration (bit: 7)

Valeur	Description
0 _b	Le dispositif CANopen peut être utilisé sans réinitialisation préalable.
1 _b	Le dispositif CANopen doit être réinitialisé aux valeurs par défaut d'usine par l'exécution d'une restauration des valeurs par défaut (objet 1011 _h)

Tableau 44 – Description d'objet

Attribut	Valeur
Index	1F81 _h
Nom	Attribution de l'esclave NMT
Code d'objet	ARRAY
Type de données	UNSIGNED32
Catégorie	Conditionnel; obligatoire si le dispositif CANopen est un gestionnaire CANopen

Tableau 45 – Description d'entrée

Attribut	Valeur
Sous-index	00 _h
Description	Nombre d'entrées
Catégorie d'entrée	Obligatoire
Accès	const
Mise en correspondance PDO	Non
Plage de valeurs	7F _h
Valeur par défaut	7F _h
Sous-index	01 _h
Description	Indicatif de nœud CANopen 1
Catégorie d'entrée	Obligatoire
Accès	rw
Mise en correspondance PDO	Non
Plage de valeurs	voir Figures 27 et 28 et Tableaux 37, 38, 39, 40, 41, 42 et 43
Valeur par défaut	0000 0000 _h
vers	
Sous-index	7F _h
Description	Indicatif de nœud CANopen 127
Catégorie d'entrée	Obligatoire
Accès	rw
Mise en correspondance PDO	Non
Plage de valeurs	voir Figures 27 et 28 et Tableaux 37, 38, 39, 40, 41, 42 et 43
Valeur par défaut	0000 0000 _h

9.8.9 Objet 1F82_h: Requête NMT

Cet objet doit demander un service NMT spécifique pour un dispositif CANopen unique dans le réseau ou pour tous les dispositifs CANopen dans ce même réseau, dans le cas où le dispositif CANopen qui met en œuvre ledit objet est en mode maître NMT. En règle générale, la requête est émise par un autre dispositif CANopen (par exemple, outil de configuration) dans le réseau ou par l'application sur le même dispositif CANopen (par exemple, dans un environnement conforme à la CEI 61131).

Le sous-index doit correspondre à l'indicatif de nœud CANopen des dispositifs CANopen dans le réseau. Ces requêtes peuvent être émises pour le maître NMT proprement dit. Le Tableau 46 définit le contenu des valeurs. Les Tableaux 47 et 48 spécifient la description d'objet et la description d'entrée.

Les valeurs comprises entre 84_h et 8F_h nécessitent de connaître l'indicatif de nœud CANopen du dispositif CANopen demandeur. Le service doit être annulé si l'indicatif de nœud CANopen n'est pas connu (code d'annulation: 0800 0000_h ou 0609 0030_h).

Un message d'annulation SDO (code d'annulation 0800 0000_h ou 0609 0030_h) doit répondre à toute tentative de téléchargement d'une valeur réservée.

NOTE Les valeurs comprises entre 00_h et 7F_h doivent être appliquées avec prudence afin de ne pas affecter de manière non délibérée le maître NMT ou le dispositif CANopen demandeur proprement dit.

Tableau 46 – Définition de valeur

Valeur	Description	
	téléchargement amont (lecture)	téléchargement aval (écriture)
00 _h	Etat NMT inconnu	réservé
01 _h	Absence de dispositif CANopen	réservé
02 _h	réservé	
03 _h	réservé	
04 _h	Etat NMT <i>Interrompu</i>	Service NMT <i>Arrêt du nœud distant</i>
05 _h	Etat NMT <i>Opérationnel</i>	Service NMT <i>Démarrage du nœud distant</i>
06 _h	réservé	Service NMT <i>Initialisation du nœud</i>
07 _h	réservé	Service NMT <i>Initialisation de la communication</i>
08 _h	réservé	
.....	
7E _h	réservé	
7F _h	Etat NMT <i>Préopérationnel</i>	Service NMT <i>Adoption de l'état préopérationnel</i>
80 _h	réservé	
.....	
83 _h	réservé	
84 _h	Etat NMT <i>Interrompu</i>	Service NMT <i>Arrêt du nœud distant</i> (à l'exclusion du maître NMT et du dispositif CANopen demandeur)
85 _h	Etat NMT <i>Opérationnel</i>	Service NMT <i>Démarrage du nœud distant</i> (à l'exclusion du maître NMT et du dispositif CANopen demandeur)
86 _h	réservé	Service NMT <i>Initialisation du nœud</i> (à l'exclusion du maître NMT et du dispositif CANopen demandeur)
87 _h	réservé	Service NMT <i>Initialisation de la communication</i> (à l'exclusion du maître NMT et du dispositif CANopen demandeur)
88 _h	réservé	
.....	
8E _h	réservé	
8F _h	Etat NMT <i>Préopérationnel</i>	Service NMT <i>Adoption de l'état préopérationnel</i> (à l'exclusion du maître NMT et du dispositif CANopen demandeur)
90 _h	réservé	
.....	
FF _h	réservé	

Tableau 47 – Description d'objet

Attribut	Valeur
Index	1F82 _h
Nom	Requête NMT
Code d'objet	ARRAY
Type de données	UNSIGNED8
Catégorie	Facultatif

Tableau 48 – Description d'entrée

Attribut	Valeur
Sous-index	00 _h
Description	Nombre d'entrées
Catégorie d'entrée	Obligatoire
Accès	const
Mise en correspondance PDO	Non
Plage de valeurs	80 _h
Valeur par défaut	80 _h
Sous-index	01 _h
Description	Indicatif de nœud CANopen 1
Catégorie d'entrée	Obligatoire
Accès	rw
Mise en correspondance PDO	Non
Plage de valeurs	Voir Tableau 46
Valeur par défaut	00 _h
vers	
Sous-index	7F _h
Description	Indicatif de nœud CANopen 127
Catégorie d'entrée	Obligatoire
Accès	rw
Mise en correspondance PDO	Non
Plage de valeurs	Voir Tableau 46
Valeur par défaut	00 _h
Sous-index	80 _h
Description	Tous les nœuds
Catégorie d'entrée	Obligatoire
Accès	wo
Mise en correspondance PDO	Non
Plage de valeurs	Voir Tableau 46
Valeur par défaut	00 _h

9.8.10 Objet 1F83_h: Requête de protection de nœud

Cet objet doit demander la protection de nœud pour un dispositif CANopen unique dans le réseau ou pour tous les dispositifs CANopen dans ce même réseau, dans le cas où le dispositif CANopen qui met en œuvre ledit objet est en mode maître NMT et où la protection de nœud est activée. En règle générale, la requête est émise par un autre dispositif CANopen (par exemple, outil de configuration) dans le réseau ou par l'application sur le même dispositif CANopen (par exemple, dans un environnement conforme à la CEI 61131).

Le sous-index doit correspondre à l'indicatif de nœud CANopen des dispositifs CANopen dans le réseau. Le sous-index correspondant à son propre indicatif de nœud CANopen doit être ignoré. Le Tableau 49 définit le contenu des valeurs. Les Tableaux 50 et 51 spécifient la description d'objet et la description d'entrée.

Un message d'annulation (code d'annulation 0800 0000_h ou 0609 0030_h) doit répondre à toute tentative de téléchargement d'une valeur réservée.

NOTE Le consommateur de pulsation est couvert par l'objet 1016_h (voir CiA301).

Tableau 49 – Définition de valeur

Valeur	Description	
	téléchargement aval (écriture)	téléchargement amont (lecture)
00 _h	Arrêt de la protection de nœud	Protection de nœud arrêtée
01 _h	Démarrage de la protection de nœud	Protection de nœud démarrée
02 _h	réservé	
.....	
FF _h	réservé	

Tableau 50 – Description d'objet

Attribut	Valeur
Index	1F83 _h
Nom	Requête de protection de nœud
Code d'objet	ARRAY
Type de données	UNSIGNED8
Catégorie	Facultatif

Tableau 51 – Description d'entrée

Attribut	Valeur
Sous-index	00 _h
Description	Nombre d'entrées
Catégorie d'entrée	Obligatoire
Accès	const
Mise en correspondance PDO	Non
Plage de valeurs	80 _h
Valeur par défaut	80 _h
Sous-index	01 _h
Description	Indicatif de nœud CANopen 1
Catégorie d'entrée	Obligatoire
Accès	rw
Mise en correspondance PDO	Non
Plage de valeurs	Voir Tableau 49
Valeur par défaut	00 _h
Sous-index	7F _h
Description	Indicatif de nœud CANopen 127
Catégorie d'entrée	Obligatoire
Accès	rw
Mise en correspondance PDO	Non
Plage de valeurs	Voir Tableau 49
Valeur par défaut	00 _h
vers	
Sous-index	80 _h
Description	Tous les nœuds
Catégorie d'entrée	Obligatoire
Accès	wo
Mise en correspondance PDO	Non
Plage de valeurs	Voir Tableau 49
Valeur par défaut	00 _h

9.8.11 Objet 1F84_h: Identification du type de dispositif

Cet objet est utilisé pour vérifier le type de dispositif des dispositifs CANopen dans le réseau.

Le type de dispositif (objet 1000_h - voir EN 50325-4) du dispositif CANopen dans le réseau doit être adapté à la valeur de cet objet dans le cas où cette dernière n'est pas nulle. La non-adaptation des valeurs doit générer un événement erreur. Le type de dispositif du dispositif CANopen dans le réseau peut ne pas être vérifié dans le cas où la valeur de cet objet est nulle. Les Tableaux 52 et 53 spécifient la description d'objet et la description d'entrée.

Le sous-index doit correspondre à l'indicatif de nœud CANopen des dispositifs CANopen dans le réseau. Le sous-index correspondant à son propre indicatif de nœud CANopen doit être ignoré.

Tableau 52 – Description d'objet

Attribut	Valeur
Index	1F84 _h
Nom	Identification du type de dispositif
Code d'objet	ARRAY
Type de données	UNSIGNED32
Catégorie	Facultatif

Tableau 53 – Description d'entrée

Attribut	Valeur
Sous-index	00 _h
Description	Nombre d'entrées
Catégorie d'entrée	Obligatoire
Accès	const
Mise en correspondance PDO	Non
Plage de valeurs	7F _h
Valeur par défaut	7F _h
Sous-index	01 _h
Description	Indicatif de nœud CANopen 1
Catégorie d'entrée	Obligatoire
Accès	rw
Mise en correspondance PDO	Non
Plage de valeurs	Voir définition de valeur de l'objet 1000 _h – EN 50325-4
Valeur par défaut	0000 0000 _h
vers	
Sous-index	7F _h
Description	Indicatif de nœud CANopen 127
Catégorie d'entrée	Obligatoire
Accès	rw
Mise en correspondance PDO	Non
Plage de valeurs	Voir définition de valeur de l'objet 1000 _h dans la EN 50325-4
Valeur par défaut	0000 0000 _h

9.8.12 Objet 1F85_h: Identification du fournisseur

Cet objet doit être utilisé pour vérifier l'indicatif du fournisseur des dispositifs CANopen dans le réseau.

L'indicatif du fournisseur (objet 1018_h sous-index 01_h – voir EN 50325-4) du dispositif CANopen dans le réseau doit être adapté à la valeur de cet objet dans le cas où cette dernière n'est pas nulle. La non-adaptation des valeurs doit générer un événement erreur. L'indicatif du fournisseur du dispositif CANopen dans le réseau peut ne pas être vérifié dans le cas où la valeur de cet objet est nulle. Les Tableaux 54 et 55 spécifient la description d'objet et la description d'entrée.

Le sous-index doit correspondre à l'indicatif de nœud CANopen des dispositifs CANopen dans le réseau. Le sous-index correspondant à son propre indicatif de nœud CANopen doit être ignoré.

Tableau 54 – Description d'objet

Attribut	Valeur
Index	1F85 _h
Nom	Identification du fournisseur
Code d'objet	ARRAY
Type de données	UNSIGNED32
Catégorie	Facultatif

Tableau 55 – Description d'entrée

Attribut	Valeur
Sous-index	00 _h
Description	Nombre d'entrées
Catégorie d'entrée	Obligatoire
Accès	const
Mise en correspondance PDO	Non
Plage de valeurs	7F _h
Valeur par défaut	7F _h
Sous-index	01 _h
Description	Indicatif de nœud CANopen 1
Catégorie d'entrée	Obligatoire
Accès	rw
Mise en correspondance PDO	Non
Plage de valeurs	Voir définition de valeur de l'objet 1018 _h sous-index 01 _h dans la EN 50325-4
Valeur par défaut	0000 0000 _h
vers	
Sous-index	7F _h
Description	Indicatif de nœud CANopen 127
Catégorie d'entrée	Obligatoire
Accès	rw
Mise en correspondance PDO	Non
Plage de valeurs	Voir définition de valeur de l'objet 1018 _h sous-index 01 _h dans la EN 50325-4
Valeur par défaut	0000 0000 _h

9.8.13 Objet 1F86_h: Code produit

Cet objet doit être utilisé pour vérifier le code produit des dispositifs CANopen dans le réseau.

Le code produit (objet 1018_h sous-index 02_h – voir EN 50325-4) du dispositif CANopen dans le réseau doit être adapté à la valeur de cet objet dans le cas où cette dernière n'est pas nulle. La non-adaptation des valeurs doit générer un événement erreur. Le code produit du dispositif CANopen dans le réseau peut ne pas être vérifié dans le cas où la valeur de cet

objet est nulle. Les Tableaux 56 et 57 spécifient la description d'objet et la description d'entrée.

Le sous-index doit correspondre à l'indicatif de nœud CANopen des dispositifs CANopen dans le réseau. Le sous-index correspondant à son propre indicatif de nœud CANopen doit être ignoré.

Tableau 56 – Description d'objet

Attribut	Valeur
Index	1F86 _h
Nom	Code produit
Code d'objet	ARRAY
Type de données	UNSIGNED32
Catégorie	Facultatif

Tableau 57 – Description d'entrée

Attribut	Valeur
Sous-index	00 _h
Description	Nombre d'entrées
Catégorie d'entrée	Obligatoire
Accès	const
Mise en correspondance PDO	Non
Plage de valeurs	7F _h
Valeur par défaut	7F _h
Sous-index	01 _h
Description	Indicatif de nœud CANopen 1
Catégorie d'entrée	Obligatoire
Accès	rw
Mise en correspondance PDO	Non
Plage de valeurs	Voir définition de valeur de l'objet 1018 _h sous-index 02 _h dans la EN 50325-4
Valeur par défaut	0000 0000 _h
vers	
Sous-index	7F _h
Description	Indicatif de nœud CANopen 127
Catégorie d'entrée	Obligatoire
Accès	rw
Mise en correspondance PDO	Non
Plage de valeurs	Voir définition de valeur de l'objet 1018 _h sous-index 02 _h dans la EN 50325-4
Valeur par défaut	0000 0000 _h

9.8.14 Objet 1F87_h: Numéro de révision

Cet objet doit être utilisé pour vérifier le numéro de révision des dispositifs CANopen dans le réseau.

Le numéro de révision (objet 1018_h sous-index 03_h – voir EN 50325-4) du dispositif CANopen dans le réseau doit être adapté à la valeur de cet objet dans le cas où cette dernière n'est pas nulle. La non-adaptation des valeurs doit générer un événement erreur. Une non-adaptation est définie comme suit:

- le numéro de révision principal n'est pas égal au numéro de révision principal prévu, ou
- le numéro de révision secondaire est inférieur au numéro de révision secondaire prévu,

Le numéro de révision du dispositif CANopen dans le réseau ne doit pas être vérifié dans le cas où la valeur de cet objet est nulle. Les Tableaux 58 et 59 spécifient la description d'objet et la description d'entrée.

Le sous-index doit correspondre à l'indicatif de nœud CANopen des dispositifs CANopen dans le réseau. Le sous-index correspondant à son propre indicatif de nœud CANopen doit être ignoré.

Tableau 58 – Description d'objet

Attribut	Valeur
Index	1F87 _h
Nom	Numéro de révision
Code d'objet	ARRAY
Type de données	UNSIGNED32
Catégorie	Facultatif

Tableau 59 – Description d'entrée

Attribut	Valeur
Sous-index	00 _h
Description	Nombre d'entrées
Catégorie d'entrée	Obligatoire
Accès	const
Mise en correspondance PDO	Non
Plage de valeurs	7F _h
Valeur par défaut	7F _h
Sous-index	01 _h
Description	Indicatif de nœud CANopen 1
Catégorie d'entrée	Obligatoire
Accès	rw
Mise en correspondance PDO	Non
Plage de valeurs	Voir définition de valeur de l'objet 1018 _h sous-index 03 _h – EN 50325-4
Valeur par défaut	0000 0000 _h
vers	

Attribut	Valeur
Sous-index	7F _h
Description	Indicatif de nœud CANopen 127
Catégorie d'entrée	Obligatoire
Accès	rw
Mise en correspondance PDO	Non
Plage de valeurs	Voir définition de valeur de l'objet 1018 _h sous-index 03 _h – EN 50325-4
Valeur par défaut	0000 0000 _h

9.8.15 Objet 1F88_h: Numéro de série

Cet objet doit être utilisé pour vérifier le numéro de série des dispositifs CANopen dans le réseau.

Le numéro de série (objet 1018_h sous-index 04_h – voir EN 50325-4) du dispositif CANopen dans le réseau doit être adapté à la valeur de cet objet dans le cas où cette dernière n'est pas nulle. La non-adaptation des valeurs doit générer un événement erreur. Le numéro de série du dispositif CANopen dans le réseau peut ne pas être vérifié dans le cas où la valeur de cet objet est nulle. Les Tableaux 60 et 61 spécifient la description d'objet et la description d'entrée.

Le sous-index doit correspondre à l'indicatif de nœud CANopen des dispositifs CANopen dans le réseau. Le sous-index correspondant à son propre indicatif de nœud CANopen doit être ignoré.

Tableau 60 – Description d'objet

Attribut	Valeur
Index	1F88 _h
Nom	Numéro de série
Code d'objet	ARRAY
Type de données	UNSIGNED32
Catégorie	Facultatif

Tableau 61 – Description d'entrée

Attribut	Valeur
Sous-index	00 _h
Description	Nombre d'entrées
Catégorie d'entrée	Obligatoire
Accès	const
Mise en correspondance PDO	Non
Plage de valeurs	7F _h
Valeur par défaut	7F _h

Attribut	Valeur
Sous-index	01 _h
Description	Indicatif de nœud CANopen 1
Catégorie d'entrée	Obligatoire
Accès	rw
Mise en correspondance PDO	Non
Plage de valeurs	Voir définition de valeur de l'objet 1018 _h sous-index 04 _h dans la EN 50325-4
Valeur par défaut	0000 0000 _h
vers	
Sous-index	7F _h
Description	Indicatif de nœud CANopen 127
Catégorie d'entrée	Obligatoire
Accès	rw
Mise en correspondance PDO	Non
Plage de valeurs	Voir définition de valeur de l'objet 1018 _h sous-index 04 _h dans la EN 50325-4
Valeur par défaut	0000 0000 _h

9.8.16 Objet 1F89_h: Temps d'initialisation

L'objet définit la temporisation entre le début du processus *Démarrer le processus d'initialisation de l'esclave NMT* et le signallement de l'initialisation réussie de tous les esclaves NMT obligatoires. Les détails sont définis en 9.4.3.

La valeur doit être indiquée en multiples de ms. La valeur 0 doit désactiver le temporisateur. Le Tableau 62 spécifie la description d'objet et le Tableau 63 spécifie la description d'entrée.

Tableau 62 – Description d'objet

Attribut	Valeur
Index	1F89 _h
Nom	Temps d'initialisation
Code d'objet	VAR
Type de données	UNSIGNED32
Catégorie	Facultatif

Tableau 63 – Description d'entrée

Attribut	Valeur
Sous-index	00 _h
Accès	rw
Mise en correspondance PDO	Non
Plage de valeurs	UNSIGNED32
Valeur par défaut	0000 0000 _h

9.8.17 Objet 1F8A_h: Restauration de la configuration

Cet objet est utilisé pour définir la procédure de restauration admise d'un dispositif CANopen lors du démarrage.

La procédure de restauration configurée dans cet objet doit être utilisée lors du démarrage pour restaurer chaque dispositif CANopen concerné dans le réseau (objet 1011_h – voir EN 50325-4). Les Tableaux 64 et 65 spécifient la description d'objet et la description d'entrée.

Le sous-index doit correspondre à l'indicatif de nœud CANopen des dispositifs CANopen dans le réseau. Le sous-index correspondant à son propre indicatif de nœud CANopen doit être ignoré.

Tableau 64 – Description d'objet

Attribut	Valeur
Index	1F8A _h
Nom	Restauration de la configuration
Code d'objet	ARRAY
Type de données	UNSIGNED8
Catégorie	Facultatif

Tableau 65 – Description d'entrée

Attribut	Valeur
Sous-index	00 _h
Description	Nombre d'entrées
Catégorie d'entrée	Obligatoire
Accès	const
Mise en correspondance PDO	Non
Plage de valeurs	7F _h
Valeur par défaut	7F _h
Sous-index	01 _h
Description	Indicatif de nœud CANopen 1
Catégorie d'entrée	Obligatoire
Accès	rw
Mise en correspondance PDO	Non
Plage de valeurs	01 _h à 7F _h
Valeur par défaut	01 _h
vers	
Sous-index	7F _h
Description	Indicatif de nœud CANopen 127
Catégorie d'entrée	Obligatoire
Accès	rw
Mise en correspondance PDO	Non
Plage de valeurs	01 _h à 7F _h
Valeur par défaut	01 _h

9.8.18 Objet 1F91_h: Paramètres de synchronisation de nœuds de démarrage automatique

Cet objet définit les paramètres qui doivent être configurés pour appliquer un comportement déterminé pour les dispositifs CANopen à démarrage automatique. Les Tableaux 66 et 67 spécifient la description d'objet et la description d'entrée.

La temporisation de détection du maître NMT, le temps de retard de la requête du maître NMT et le créneau de temps de nœud doivent être indiqués en multiples de ms.

Tableau 66 – Description d'objet

Attribut	Valeur
Index	1F91 _h
Nom	Paramètres de synchronisation de nœuds de démarrage automatique
Code d'objet	ARRAY
Type de données	UNSIGNED16
Catégorie	Facultatif Obligatoire pour les dispositifs CANopen prenant en charge le démarrage automatique

Tableau 67 – Description d'entrée

Attribut	Valeur
Sous-index	00 _h
Description	Nombre d'entrées
Catégorie d'entrée	Obligatoire
Accès	const
Mise en correspondance PDO	Non
Plage de valeurs	03 _h
Valeur par défaut	03 _h
Sous-index	01 _h
Description	Temporisation de détection du maître NMT
Catégorie d'entrée	Obligatoire
Accès	rw
Mise en correspondance PDO	Non
Plage de valeurs	UNSIGNED16
Valeur par défaut	100
Sous-index	02 _h
Description	Temps de retard de la requête du maître NMT
Catégorie d'entrée	Obligatoire
Accès	rw
Mise en correspondance PDO	Non
Plage de valeurs	UNSIGNED16
Valeur par défaut	500

Attribut	Valeur
Sous-index	03 _h
Description	Créneau de temps de nœud
Catégorie d'entrée	Obligatoire
Accès	rw
Mise en correspondance PDO	Non
Plage de valeurs	UNSIGNED16
Valeur par défaut	15

10 Fonctions passerelle

10.1 Teneur

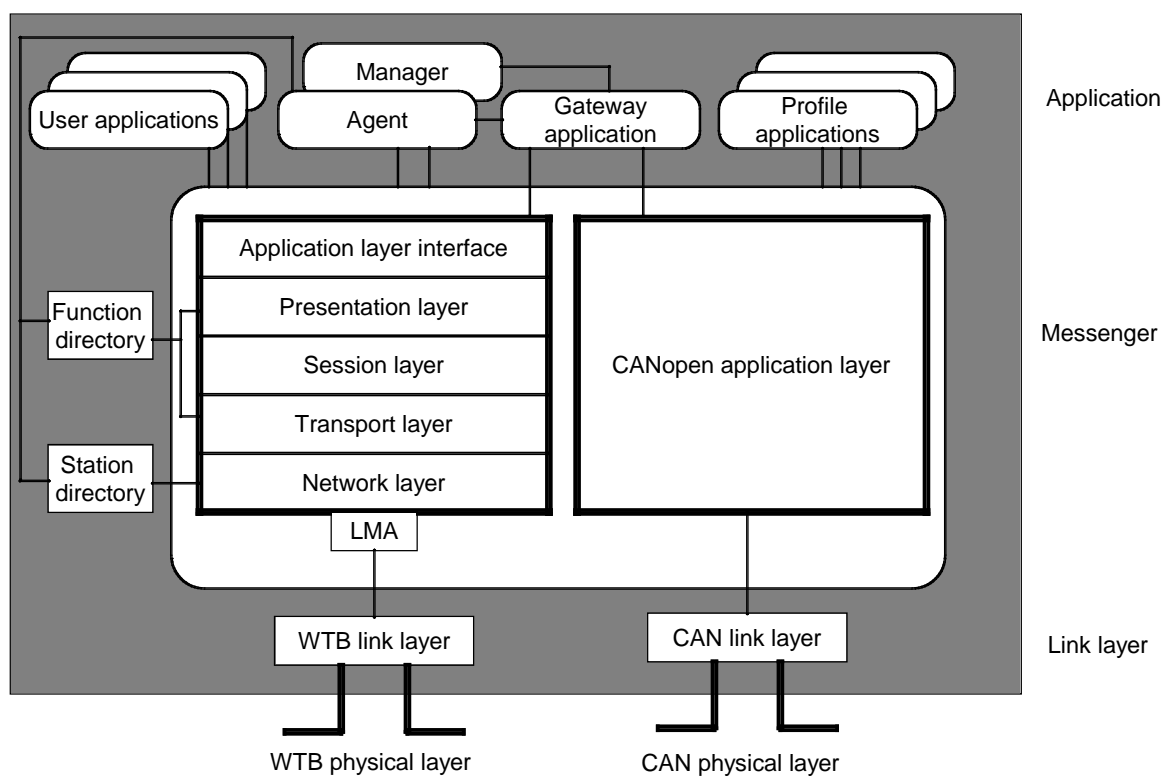
Le présent article spécifie les fonctions passerelle pour les passerelles placées entre un Réseau Central de Train et un réseau de rame basé sur CANopen.

Les services d'accès au réseau sont spécifiés pour permettre l'accès aux réseaux de rame basés sur CANopen. Une mise en correspondance de ces services d'accès CANopen avec un protocole ASCII est également définie. La mise en correspondance de ce protocole avec le traitement des messages de gestion est définie à l'Article 11 pour intégration à une architecture TCN.

Selon les capacités mises en œuvre, le présent article introduit également plusieurs classes de passerelle.

10.2 Architecture de passerelle

L'architecture d'une passerelle entre un Réseau Central de Train et un réseau de rame CANopen est illustrée à la Figure 29. Dans l'illustration, le Réseau Central de Train est mis en œuvre comme bus de train filaire (WTB).



Légende

Anglais	Français
User applications	Applications utilisateur
Manager	Gestionnaire
Agent	Agent
Gateway application	Application passerelle
Profile applications	Applications de profil
Function directory	Répertoire de fonctions
Station directory	Répertoire de stations
Application layer interface	Interface couche Application
Presentation layer	Couche Présentation
Session layer	Couche Session
Transport layer	Couche Transport
Network layer	Couche Réseau
CANopen application layer	Couche Application CANopen
Messenger	Messenger
WTB link layer	Couche Liaison WTB
CAN link layer	Couche Liaison CAN
Link layer	Couche Liaison
WTB physical layer	Couche physique WTB
CAN physical layer	Couche physique CAN

Figure 29 – Passerelle entre le Réseau Central de Train et le réseau de rame CANopen

Une passerelle sert d'intermédiaire entre un Réseau Central de Train et un réseau de rame basé sur CANopen. Elle transfère les données de processus via l'application passerelle et garantit l'adoption du style endian correct pour la représentation des données des deux côtés de la passerelle.

NOTE L'ordonnancement pour le transfert des données au format CANopen est de type little-endian (petit boutiste) par opposition au bus de train filaire qui utilise le style big-endian (gros boutiste).

Des services TNM peuvent être requis des deux côtés de la passerelle au moyen de messages de gestion TNM. Outre les services spécifiés dans la CEI 61375-2-1, d'autres services TNM sont définis (voir 11.5) qui permettent un contrôle total du réseau de rame CANopen à sous-couches (voir les services d'accès au réseau définis en 10.4). Les services d'accès au réseau fournis par une passerelle – entre le Réseau Central de Train et un réseau de rame basé sur CANopen – génèrent un Nœud, qui est connecté au Réseau Central de Train, et permet l'accès aux Dispositifs Terminaux connectés à un réseau de rame basé sur CANopen.

Dans la mesure où une mise en correspondance des messages de gestion TNM avec les objets de communication CANopen (COB) est définie, les services de gestion TNM sont également disponibles au sein des réseaux de rame basés sur CANopen.

La Figure 29 montre que la fonctionnalité Gestionnaire et la fonctionnalité Agent sont disponibles au niveau de l'application, comme défini dans la CEI 61375-2-1. De plus, des applications utilisateur spécifiques au fabricant, ainsi que des applications normalisées selon les profils CANopen (applications de profil) peuvent être disponibles à ce même niveau.

10.3 Principes généraux et services

10.3.1 Teneur

Le présent paragraphe spécifie les services d'accès au réseau fournis par une passerelle qui permet un accès au réseau de rame basé sur CANopen à partir du Réseau Central de Train.

10.3.2 Définitions des classes de passerelle

Un dispositif CANopen, qui comporte la fonctionnalité passerelle avec le Réseau Central de Train, peut prendre en charge une ou plusieurs des classes suivantes:

- Classe 0: La passerelle est un dispositif qui agit en tant qu'esclave de réseau (fonctionnalité Esclave NMT) au sein du réseau CANopen. Le dispositif prend en charge uniquement la capacité de transfert des données de processus. Une passerelle appartenant à cette classe n'est pas capable de transmettre des messages de gestion TNM via le réseau CANopen.
- Classe 1: La passerelle est un dispositif qui agit en tant qu'esclave de réseau (fonctionnalité Esclave NMT) au sein du réseau CANopen. Le dispositif doit fournir, outre la fonctionnalité définie pour les passerelles de classe 0, la fonctionnalité Client SDO.
- Classe 2: La passerelle est un dispositif qui met en œuvre la fonctionnalité d'un dispositif de classe 1, et qui en plus met en œuvre la fonctionnalité Dispositif de requête (SRD) SDO.
- Classe 3: La passerelle est un dispositif qui agit en tant que gestionnaire CANopen au sein du réseau CANopen. La fonctionnalité Gestionnaire CANopen doit être exécutée par une prise en charge de la fonctionnalité Maître NMT, ainsi que la fonctionnalité Gestionnaire SDO.

NOTE Seules les passerelles de classe 1, 2 ou 3 sont capables de transmettre des messages de gestion TNM au sein de réseaux de rame basés sur CANopen.

10.3.3 Définitions des primitives de service

Les primitives de service constituent le moyen d'interaction de la couche application de passerelle et de la couche application de réseau. Les primitives de service sont définies dans la CEI 61375-1.

10.4 Spécification des services d'accès au réseau

10.4.1 Services d'accès SDO

10.4.1.1 Teneur

Les services spécifiés dans le présent article permettent d'initier et de configurer les services SDO d'accès à tout objet dans le dictionnaire d'objets propre à tout nœud de l'un quelconque des réseaux CANopen reliés à la passerelle.

10.4.1.2 Téléchargement SDO amont

Ce service doit initier un service de téléchargement SDO amont. Les paramètres applicables à ce service sont définis dans le Tableau 68.

Tableau 68 – Service de téléchargement SDO amont

Paramètre	Indication	Réponse
Argument Réseau Indicatif de nœud Multiplexeur Type de données BOOLEAN UNSIGNED8 UNSIGNED16 UNSIGNED24 UNSIGNED32 UNSIGNED40 UNSIGNED48 UNSIGNED56 UNSIGNED64 INTEGER8 INTEGER16 INTEGER24 INTEGER32 INTEGER40 INTEGER48 INTEGER56 INTEGER64 REAL32 REAL64 TIME_OF_DAY TIME_DIFFERENCE OCTET_STRING VISIBLE_STRING UNICODE_STRING DOMAIN Décalage Longueur	Obligatoire Facultatif Facultatif Obligatoire Facultatif Sélection Facultatif Facultatif	
Résultat distant Succès Données Longueur Echec Motif		Obligatoire Sélection Obligatoire Facultatif Sélection Obligatoire

10.4.1.3 Téléchargement SDO aval

Ce service doit initier un service de téléchargement SDO aval. Les paramètres applicables à ce service sont définis dans le Tableau 69.

Tableau 69 – Paramètres de téléchargement SDO aval

Paramètre	Indication	Réponse
Argument	Obligatoire	
Réseau	Facultatif	
Indicatif de nœud	Facultatif	
Multiplexeur	Obligatoire	
Type de données	Facultatif	
BOOLEAN	Sélection	
UNSIGNED8	Sélection	
UNSIGNED16	Sélection	
UNSIGNED24	Sélection	
UNSIGNED32	Sélection	
UNSIGNED40	Sélection	
UNSIGNED48	Sélection	
UNSIGNED56	Sélection	
UNSIGNED64	Sélection	
INTEGER8	Sélection	
INTEGER16	Sélection	
INTEGER24	Sélection	
INTEGER32	Sélection	
INTEGER40	Sélection	
INTEGER48	Sélection	
INTEGER56	Sélection	
INTEGER64	Sélection	
REAL32	Sélection	
REAL64	Sélection	
TIME_OF_DAY	Sélection	
TIME_DIFFERENCE	Sélection	
OCTET_STRING	Sélection	
VISIBLE_STRING	Sélection	
UNICODE_STRING	Sélection	
DOMAIN	Sélection	
Décalage	Facultatif	
Longueur	Facultatif	
Données	Obligatoire	
Résultat distant		Obligatoire
Succès		Sélection
Echec		Sélection
Motif		Obligatoire

10.4.1.4 Configuration de temporisation SDO

Ce service doit configurer la temporisation pour tous les SDO client présents sur la passerelle. Les paramètres applicables à ce service sont définis dans le Tableau 70.

Tableau 70 – Paramètres de configuration de temporisation SDO

Paramètre	Indication	Réponse
Argument Réseau Temporisation SDO	Obligatoire Facultatif Obligatoire	
Résultat distant Succès Echec Motif		Obligatoire Sélection Sélection Obligatoire

10.4.2 Services d'accès PDO

10.4.2.1 Teneur

Les services spécifiés dans le présent article permettent de configurer et d'initier des services PDO dans la passerelle. Ces services incluent:

- Configuration d'un PDO reçu
- Configuration d'un PDO transmis
- Requête de lecture d'un PDO

- Requête d'écriture d'un PDO
- Indication d'un PDO reçu

Les deux services de configuration PDO sont destinés à générer des PDO dans la passerelle. Si la passerelle met en œuvre un dictionnaire d'objets, les entrées des paramètres de communication et de mise en correspondance PDO doivent être réglées en conséquence.

Les deux services de requête PDO sont destinés à contrôler les PDO conformément au type de transmission PDO configuré.

Les types de données `VISIBLE_STRING`, `OCTET_STRING` et `UNICODE_STRING`, ainsi que `DOMAIN`, ne doivent pas être utilisés comme paramètre de type de données PAS.

NOTE Les services PAS ne sont pas destinés à configurer les entrées de communication et de mise en correspondance PDO du dictionnaire d'objets de nœud distant. L'accès individuel aux nœuds CANopen par l'intermédiaire des services SAS peut réaliser la configuration PDO.

10.4.2.2 Configuration RPDO

Ce service doit créer un RPDO dans la passerelle. Le Tableau 71 définit les paramètres applicables à ce service.

Tableau 71 – Paramètres de configuration des services RPDO

<i>Paramètre</i>	<i>Indication</i>	<i>Réponse</i>
Argument Réseau Numéro PDO Indicatif COB TxType Nbr_objects 1 ^{er} objet mis en correspondance Type de données Multiplexeur 2 ^{ème} objet mis en correspondance Type de données Multiplexeur ... 64 ^{ème} objet mis en correspondance Type de données Multiplexeur	Obligatoire Facultatif Obligatoire Obligatoire Obligatoire Obligatoire Sélection Sélection Facultatif Sélection Sélection ... Facultatif Sélection Sélection	
Résultat distant Succès Echec Motif		Obligatoire Sélection Sélection Obligatoire

10.4.2.3 Configuration TPDO

Ce service doit créer un TPDO dans la passerelle. Le Tableau 72 définit les paramètres applicables à ce service.

Tableau 72 – Paramètres de configuration des services TPDO

Paramètre	Indication	Réponse
Argument Réseau Numéro PDO INDICATIF COB TxType Nbr_objects 1 ^{er} objet mis en correspondance Type de données Multiplexeur 2 ^{ème} objet mis en correspondance Type de données Multiplexeur ... 64 ^{ème} objet mis en correspondance Type de données Multiplexeur	Obligatoire Facultatif Obligatoire Obligatoire Obligatoire Obligatoire Obligatoire Facultatif Sélection Sélection Facultatif Sélection Sélection ... Facultatif Sélection Sélection	
Résultat distant Succès Echec Motif		Obligatoire Sélection Sélection Obligatoire

10.4.2.4 Lecture des données PDO

Ce service doit lire les données reçues par un RPDO. Si un RPDO est configuré avec le type de transmission 252 ou 253, les passerelles doivent le déclencher au moyen d'un RTR.

NOTE L'utilisation d'un RTR n'est pas recommandée.

Les données reçues doivent être transmises dans le résultat distant. Le Tableau 73 définit les paramètres applicables à ce service.

Tableau 73 – Paramètres de service de lecture des données PDO

Paramètre	Indication	Réponse
Argument Réseau Numéro PDO	Obligatoire Facultatif Obligatoire	
Résultat distant Succès Réseau Numéro PDO Nbr_objects Données 1 ^{er} objet ... Données 64 ^{ème} objet Echec Motif		Obligatoire Sélection Facultatif Obligatoire Obligatoire Conditionnel ... Conditionnel Sélection Obligatoire

10.4.2.5 Ecriture des données PDO

Ce service doit déclencher la transmission d'un PDO. La transmission réelle du PDO doit être déclenchée selon le type de transmission PDO configuré.

Les paramètres applicables à ce service sont définis dans le Tableau 74.

Tableau 74 – Paramètres de service d'écriture des données PDO

<i>Paramètre</i>	<i>Indication</i>	<i>Réponse</i>
Argument Réseau Numéro PDO Nbr_objects Données 1 ^{er} objet ... Données 64 ^{ème} objet	Obligatoire Facultatif Obligatoire Obligatoire Conditionnel ... Conditionnel	
Résultat distant Succès Echec Motif		Obligatoire Sélection Sélection Obligatoire

10.4.2.6 RPDO reçu

Ce service doit indiquer que de nouvelles données PDO ont été reçues, et fournir les données effectivement reçues.

Les paramètres applicables à ce service sont définis dans le Tableau 75.

Tableau 75 – Paramètres de service des RPDO reçus

<i>Paramètre</i>	<i>Requête</i>
Argument Réseau Numéro PDO Nbr_objects Valeur 1 ^{er} objet ... Valeur 64 ^{ème} objet	Obligatoire Facultatif Obligatoire Obligatoire Conditionnel ... Conditionnel

10.4.3 Services NMT CANopen

10.4.3.1 Teneur

Les services spécifiés dans le présent article permettent de contrôler un dispositif ou un réseau CANopen, ainsi que les services de contrôle d'erreurs associés.

10.4.3.2 Démarrage de nœud

Ce service doit mettre les nœuds CANopen à l'état NMT OPERATIONAL. Pour les dispositifs des classes 1 et 2, ce service doit déclencher un service NMT de requête CANopen. Pour les dispositifs de classe 3, il doit déclencher un service de Démarrage du nœud distant. Pour les dispositifs de classe 0, ce service doit déclencher une notification d'échec comme résultat distant. Le Tableau 76 définit les paramètres applicables à ce service.

Tableau 76 – Paramètres de service de démarrage de nœud

<i>Paramètre</i>	<i>Indication</i>	<i>Réponse</i>
Argument Réseau Indicatif de nœud Tous	Obligatoire Facultatif Sélection Sélection	
Résultat distant Succès Echec Motif		Obligatoire Sélection Sélection Obligatoire

10.4.3.3 Arrêt de nœud

Ce service doit mettre les nœuds CANopen à l'état NMT STOPPED. Pour les dispositifs des classes 1 et 2, il doit déclencher un service NMT de requête CANopen. Pour les dispositifs de classe 3, il doit déclencher un service d'arrêt de nœud distant. Pour les dispositifs de classe 0, ce service doit déclencher une notification d'échec comme résultat distant. Le Tableau 77 définit les paramètres applicables à ce service.

NOTE Le résultat distant pour les dispositifs des classes 1 et 2 constitue uniquement la confirmation de la requête SDO; pour les dispositifs de classe 3, le résultat distant est basé sur les services de contrôle d'erreurs comme défini dans la EN 50325-4.

Tableau 77 – Paramètres de service d'arrêt de nœud

Paramètre	Indication	Réponse
Argument Réseau Indicatif de nœud Tous	Obligatoire Facultatif Sélection Sélection	
Résultat distant Succès Echec Motif		Obligatoire Sélection Sélection Obligatoire

10.4.3.3.1 Mise du nœud à l'état préopérationnel

Ce service doit mettre les nœuds CANopen à l'état NMT PRE-OPERATIONAL. Pour les dispositifs des classes 1 et 2, il doit déclencher un service NMT de requête CANopen. Pour les dispositifs de classe 3, il doit déclencher un service d'adoption de l'état Préopérationnel. Pour les dispositifs de classe 0, ce service doit déclencher une notification d'échec comme résultat distant. Le Tableau 78 définit les paramètres applicables à ce service.

Tableau 78 – Paramètres de service de mise du nœud à l'état préopérationnel

Paramètre	Indication	Réponse
Argument Réseau Indicatif de nœud Tous	Obligatoire Facultatif Sélection Sélection	
Résultat distant Succès Echec Motif		Obligatoire Sélection Sélection Obligatoire

10.4.3.4 Initialisation de nœud

Ce service doit mettre les nœuds CANopen à l'état NMT RESET APPLICATION. Pour les dispositifs des classes 1 et 2, il doit déclencher un service NMT de requête CANopen. Pour les dispositifs de classe 3, il doit déclencher un service d'initialisation de nœud. Pour les dispositifs de classe 0, ce service doit déclencher une notification d'échec comme résultat distant. Le Tableau 79 définit les paramètres applicables à ce service.

Tableau 79 – Paramètres de service d'initialisation de nœud

Paramètre	Indication	Réponse
Argument Réseau Indicatif de nœud Tous	Obligatoire Facultatif Sélection Sélection	
Résultat distant Succès Echec Motif		Obligatoire Sélection Sélection Obligatoire

10.4.3.5 Initialisation de la communication

Ce service doit mettre les nœuds CANopen à l'état NMT RESET COMMUNICATION. Pour les dispositifs des classes 1 et 2, ce service doit déclencher un service NMT de requête CANopen. Pour les dispositifs de classe 3, il doit déclencher un service d'initialisation de la communication. Pour les dispositifs de classe 0, ce service doit déclencher une notification d'échec comme résultat distant. Le Tableau 80 définit les paramètres applicables à ce service.

Tableau 80 – Paramètres de service d'initialisation de la communication

Paramètre	Indication	Réponse
Argument Réseau Indicatif de nœud Tous	Obligatoire Facultatif Sélection Sélection	
Résultat distant Succès Echec Motif		Obligatoire Sélection Sélection Obligatoire

10.4.3.6 Activation de la protection de nœud

Ce service n'est disponible que pour les dispositifs de classe 3. Il doit lancer la protection de nœud pour le dispositif spécifié par l'indicatif de nœud CANopen avec les paramètres donnés par GuardTime et LifeTimeFactor. La requête du service doit être rejetée si la pulsation est déjà activée sur le nœud concerné. Le Tableau 81 définit les paramètres applicables à ce service.

Tableau 81 – Paramètres de service d'activation de la protection de nœud

Paramètre	Indication	Réponse
Argument Réseau Indicatif de nœud GuardTime LifeTimeFactor	Obligatoire Facultatif Obligatoire Obligatoire Obligatoire	
Résultat distant Succès Echec Motif		Obligatoire Sélection Sélection Obligatoire

10.4.3.7 Désactivation de la protection de nœud

Ce service n'est disponible que pour les dispositifs de classe 3. Il doit arrêter la protection de nœud pour le dispositif spécifié par l'indicatif de nœud CANopen. Le Tableau 82 définit les paramètres applicables à ce service.

Tableau 82 – Paramètres de service de désactivation de la protection de nœud

Paramètre	Indication	Réponse
Argument Réseau Indicatif de nœud	Obligatoire Facultatif Obligatoire	
Résultat distant Succès Echec Motif		Obligatoire Sélection Sélection Obligatoire

10.4.3.8 Démarrage du consommateur de pulsation

Ce service doit initier la consommation des messages de pulsation transmis par un dispositif CANopen spécifié par l'indicatif de nœud CANopen. Le Tableau 83 définit les paramètres applicables à ce service.

Tableau 83 – Paramètres de service de démarrage du consommateur de pulsation

Paramètre	Indication	Réponse
Argument Réseau Indicatif de nœud HeartbeatConsumerTime	Obligatoire Facultatif Obligatoire Obligatoire	
Résultat distant Succès Echec Motif		Obligatoire Sélection Sélection Obligatoire

10.4.3.9 Désactivation du consommateur de pulsation

Ce service doit arrêter la consommation des messages de pulsation transmis par un dispositif CANopen spécifié par l'indicatif de nœud CANopen. Le Tableau 84 définit les paramètres applicables à ce service.

Tableau 84 – Paramètres de service de désactivation du consommateur de pulsation

Paramètre	Indication	Réponse
Argument Réseau Indicatif de nœud	Obligatoire Facultatif Obligatoire	
Résultat distant Succès Echec Motif		Obligatoire Sélection Sélection Obligatoire

10.4.3.10 Événement contrôle d'erreurs reçu

Ce service doit indiquer les événements état NMT ou contrôle d'erreurs en provenance d'un nœud CANopen spécifié par l'indicatif de nœud CANopen. Le Tableau 85 définit les paramètres applicables à ce service.

Tableau 85 – Paramètres d'événement contrôle d'erreurs reçu

Paramètre	Requête
Argument Réseau Indicatif de nœud Etat Code d'erreur	Obligatoire Facultatif Obligatoire Sélection Sélection

10.4.4 Services de gestion de défaillance d'un dispositif

10.4.4.1 Généralités

Les services spécifiés dans le présent article permettent de gérer les défaillances internes à la passerelle ou à tout autre dispositif CANopen.

10.4.4.2 Lecture d'erreur de dispositif

Ce service doit lire les informations de messages EMCY transmises par le dispositif CANopen spécifié par le paramètre d'indicatif de nœud CANopen. Le Tableau 86 définit les paramètres applicables à ce service.

Tableau 86 – Paramètres de service de lecture d'erreur de dispositif

Paramètre	Indication	Réponse
Argument Réseau Indicatif de nœud	Obligatoire Facultatif Facultatif	
Résultat distant Succès Réseau Indicatif de nœud Erreur Numéro de message d'erreur Texte de message d'erreur Situation critique Code de situation critique Registre d'erreurs Erreur de fabrication Echec Motif		Obligatoire Sélection Facultatif Obligatoire Sélection Obligatoire Facultatif Sélection Obligatoire Facultatif Facultatif Sélection Obligatoire

10.4.4.3 Événement situation critique reçu

Ce service doit indiquer la réception d'un message de situation critique dans la passerelle transmis par un dispositif CANopen spécifié par l'indicatif de nœud CANopen. Le Tableau 87 définit les paramètres applicables à ce service.

Tableau 87 – Paramètres de service d'événement situation critique reçu

Paramètre	Requête
Argument Réseau Indicatif de nœud Code de situation critique Registre d'erreurs Erreur de fabrication	Obligatoire Facultatif Obligatoire Obligatoire Obligatoire Facultatif

10.4.5 Services de configuration d'interface CANopen

10.4.5.1 Généralités

Les services décrits dans le présent article permettent de configurer et de paramétrer l'interface CANopen de la passerelle.

10.4.5.2 Initialisation de la passerelle

Ce service doit déclencher l'initialisation CANopen de la passerelle. Il doit effectuer une initialisation équivalant à la mise sous tension de l'interface CANopen. Ce service est utilisé pour l'initialisation des paramètres de synchronisation des bits. Le Tableau 88 définit les paramètres applicables à ce service.

Tableau 88 – Paramètres de service d'initialisation de la passerelle

Paramètre	Indication	Réponse
Argument Réseau Synchronisation des bits CAN	Obligatoire Facultatif Facultatif	
Résultat distant Succès Echec Motif		Obligatoire Sélection Sélection Obligatoire

10.4.5.3 Enregistrement de la configuration

Ce service doit ordonner à la passerelle d'enregistrer sa configuration d'interface CANopen. Le Tableau 89 définit les paramètres applicables à ce service.

Tableau 89 – Paramètres de service d'enregistrement de configuration

Paramètre	Indication	Réponse
Argument Réseau Spécificateur d'enregistrement	Obligatoire Facultatif Facultatif	
Résultat distant Succès Echec Motif		Obligatoire Sélection Sélection Obligatoire

10.4.5.4 Restauration de la configuration

Ce service doit ordonner à la passerelle de restaurer sa configuration d'interface CANopen. Le Tableau 90 définit les paramètres applicables à ce service.

Tableau 90 – Paramètres de service de restauration de configuration

Paramètre	Indication	Réponse
Argument Réseau Spécificateur d'enregistrement	Obligatoire Facultatif Facultatif	
Résultat distant Succès Echec Motif		Obligatoire Sélection Sélection Obligatoire

10.4.5.5 Réglage du producteur de pulsation

Ce service doit régler le temps de production de la pulsation CANopen de la passerelle. Le Tableau 91 définit les paramètres applicables à ce service.

Tableau 91 – Paramètres de service de réglage du producteur de pulsation

Paramètre	Indication	Réponse
Argument Réseau Indicatif de nœud HeartbeatProducerTime	Obligatoire Facultatif Obligatoire Obligatoire	
Résultat distant Succès Echec Motif		Obligatoire Sélection Sélection Obligatoire

10.4.5.6 Réglage de l'indicatif de nœud

Ce service doit régler l'indicatif de nœud CANopen de la passerelle pour le réseau CANopen indiqué dans le paramètre de *réseau*. Le Tableau 92 définit les paramètres applicables à ce service.

Tableau 92 – Paramètres de service de réglage de l'indicatif de nœud

Paramètre	Indication	Réponse
Argument Réseau Indicatif de nœud	Obligatoire Facultatif Obligatoire	
Résultat distant Succès Echec Motif		Obligatoire Sélection Sélection Obligatoire

10.4.5.7 Démarrage du consommateur de messages de situation critique

Ce service doit initier la consommation des messages de situation critique. La relation entre l'indicatif de nœud CANopen qui génère un message de situation critique et l'indicatif COB doit être explicitement connue. Le Tableau 93 définit les paramètres applicables à ce service.

Tableau 93 – Paramètres de service de démarrage du consommateur de messages de situation critique

Paramètre	Indication	Réponse
Argument Réseau Indicatif de nœud INDICATIF COB	Obligatoire Facultatif Obligatoire Obligatoire	
Résultat distant Succès Echec Motif		Obligatoire Sélection Sélection Obligatoire

10.4.5.8 Arrêt du consommateur de messages de situation critique

Ce service doit arrêter la consommation des messages de situation critique. La relation entre l'indicatif de nœud CANopen qui génère un message de situation critique et l'indicatif COB doit être clairement connue. Le Tableau 94 définit les paramètres applicables à ce service.

Tableau 94 – Paramètres de service d'arrêt du consommateur de messages de situation critique

Paramètre	Indication	Réponse
Argument Réseau Indicatif de nœud Indicatif COB	Obligatoire Facultatif Obligatoire Obligatoire	
Résultat distant Succès Echec Motif		Obligatoire Sélection Sélection Obligatoire

10.4.6 Services de gestion de la passerelle

10.4.6.1 Généralités

Les services spécifiés dans le présent article permettent de gérer la passerelle.

10.4.6.2 Réglage du réseau par défaut

Ce service doit régler le numéro de réseau par défaut, qui doit être appliqué à tous les services. Le Tableau 95 définit les paramètres applicables à ce service.

Tableau 95 – Paramètres de service de réglage du réseau par défaut

Paramètre	Indication	Réponse
Argument DefaultNetwork	Obligatoire Obligatoire	
Résultat distant Succès Echec Motif		Obligatoire Sélection Sélection Obligatoire

10.4.6.3 Réglage de l'indicatif de nœud par défaut

Ce service doit régler l'indicatif de nœud CANopen par défaut, qui doit être appliqué à tous les services. Le Tableau 96 définit les paramètres applicables à ce service.

Tableau 96 – Paramètres de service de réglage de l'indicatif de nœud par défaut

Paramètre	Indication	Réponse
Argument Indicatif de nœud par défaut	Obligatoire Obligatoire	
Résultat distant Succès Echec Motif		Obligatoire Sélection Sélection Obligatoire

10.4.6.4 Obtention de la version

Ce service doit obtenir les informations de la passerelle et de son interface CANopen. Le Tableau 97 définit les paramètres applicables à ce service.

Tableau 97 – Paramètres de service d'obtention de la version

Paramètre	Indication	Réponse
Argument Réseau	Obligatoire Facultatif	
Résultat distant Succès Identification du fournisseur Code produit Numéro de révision Numéro de série Classe de passerelle Version de protocole Classe de mise en œuvre Echec Motif		Obligatoire Sélection Obligatoire Obligatoire Obligatoire Obligatoire Obligatoire Obligatoire Obligatoire Sélection Obligatoire

10.4.7 Services spécifiques au fabricant

Le fabricant d'une passerelle peut définir des services supplémentaires.

10.5 Mise en correspondance ASCII des services d'accès au réseau

10.5.1 Teneur

Le présent article définit la mise en correspondance des services d'accès au réseau avec une syntaxe de communication basée sur les codes ASCII pour les passerelles CANopen. Ce protocole est mis en correspondance avec les données de messagerie TCN.

10.5.2 Définitions

10.5.2.1 Commande

Une Commande contrôle la passerelle et interagit avec les dispositifs CANopen. Elle peut se présenter sous une forme longue ou courte. La forme courte est une abréviation à une ou deux lettres de la forme longue. La forme longue est obtenue par concaténation de la forme courte et de la chaîne inscrite entre crochets “[”, “]”.

NOTE Les exemples fournis supposent un prééclage de l'adresse de réseau et de l'adresse de nœud.

10.5.2.2 Syntaxe de types de données

Les types de données obligatoires, fournis dans le Tableau 98, doivent être pris en charge.

Tableau 98 – Syntaxe et types de données CANopen

Syntaxe	Type CANopen	Catégorie
b	Booléen	Obligatoire
u8	UNSIGNED8	Obligatoire
u16	UNSIGNED16	Obligatoire
u24	UNSIGNED24	Facultatif
u32	UNSIGNED32	Obligatoire
u40	UNSIGNED40	Facultatif
u48	UNSIGNED48	Facultatif
u56	UNSIGNED56	Facultatif
u64	UNSIGNED64	Facultatif
i8	INTEGER8	Obligatoire
i16	INTEGER16	Obligatoire
i24	INTEGER24	Facultatif
i32	INTEGER32	Obligatoire
i40	INTEGER40	Facultatif
i48	INTEGER48	Facultatif
i56	INTEGER56	Facultatif
i64	INTEGER64	Facultatif
r32	REAL32	Facultatif
r64	REAL64	Facultatif
t	Heure de la journée (avec deux arguments: jour ms)	Facultatif
td	Différence de temps	Facultatif
vs	Chaîne visible	Facultatif
os	Chaîne d'octets	Facultatif
us	Chaîne unicode	Facultatif
d	Domaine	Facultatif

NOTE La valeur des types de données *domaine*, *chaîne d'octets* et *chaîne unicode* est codée en mime-base64 comme décrit dans le document RFC 2045.

Tous les éléments CRLF de mise en forme doivent être retirés des données codées pour obtenir une chaîne longue.

10.5.2.3 Blanc

Blancs spécifiés dans l'ISO/CEI 9899, sauf les éléments CR et LF.

NOTE Une chaîne visible comportant un blanc est délimitée par des guillemets afin de la désigner comme argument unique de la commande. Si la chaîne comporte des guillemets, ces derniers sont délimités par d'autres guillemets, par exemple, "Hello ""World""", "CANopen is great".

10.5.2.4 Structure de commande

10.5.2.4.1 Généralités

Le principe de communication est basé sur des chaînes ASCII sans distinction majuscules/minuscules conformément à l'ISO/CEI 646, en lieu et place de structures binaires dépendant de l'architecture et de l'UC/compilateur. De ce fait, aucune application ne traite

d'éléments tels que « petit boutiste » ou « gros boutiste », taille des données et alignement d'octets. Dans tous les cas d'utilisation de nombres, la représentation typique est telle que spécifiée dans l'ISO/CEI 9899.

100	- décimal, commençant par un nombre
0x64	- hexadécimal, commençant par la chaîne 0x
1.22	- virgule flottante
.22e10	- virgule flottante
22e3	- virgule flottante

10.5.2.4.2 Requête

La passerelle CANopen est contrôlée par des commandes. Une commande se compose de jetons séparés par un nombre de blancs et est délimitée par un élément CRLF.

Toutes les commandes sont confirmées. Les commandes débutent par un numéro de séquence délimité par des crochets []. Le numéro de séquence est une valeur à 4 octets. Cette valeur n'est pas utilisée pour les messages déclenchés par un événement. Conformément au principe d'adressage, un numéro de réseau et un numéro de nœud suivent le numéro de séquence. Le numéro de réseau et le numéro de nœud sont des éléments facultatifs, lorsque la passerelle CANopen fournit uniquement une interface entre le Réseau Central de Train et le Réseau de Rame CANopen, ou lorsqu'un client règle ces éléments au préalable. Pour les commandes qui affectent uniquement le serveur et non un nœud distant, mais un réseau et un nœud sont attribués, ce réseau et ce nœud sont ignorés. Dans la notation FBN, une commande est définie comme indiqué dans le Tableau 99:

Tableau 99 – Notation d'une commande au format FBN

command-request	::=	"["<sequence>"]" [[<net>] <node>] <command>
<sequence>	::=	UNSIGNED32
<net>	::=	UNSIGNED8
<node>	::=	UNSIGNED8
<command>	::=	<command-spezifier> <compound-command>
<compound-command>	::=	<command-spezifier> <parameter>
<parameter>	::=	<value> <compound-parameter>
<compound-parameter>	::=	<value> <parameter>
<map-object>	::=	<datatype> <multiplexor>
<datatype>	::=	'b' 'u8' 'u16' 'u32' 'u40' 'u48' 'u56' 'u64' 'i8' 'i16' 'i24' 'i32' 'i40' 'i48' 'i56' 'i64' 'r32' 'r64' 't' 'td' 'vs' 'os' 'us' 'd'
<multiplexor>	::=	<index> <subindex>
<index>	::=	UNSIGNED16
<subindex>	::=	UNSIGNED8

Les numéros de réseau et les numéros de nœuds commencent par 1. La valeur 0 applicable au réseau ou au nœud est utilisée pour l'adressage de tous les réseaux ou de tous les nœuds.

La <valeur> jeton désigne une valeur des types de données CANopen potentiels. Le numéro de séquence est omis dans la description des commandes pour des raisons de lisibilité.

10.5.2.4.3 Réponse

La passerelle CANopen doit répondre avec le même numéro de séquence à la première position comme indiqué par la requête. Ce numéro doit être donné sous forme décimale. Une requête doit appeler une seule réponse. La notation de réponse est donnée dans le Tableau 100.

Tableau 100 – Notation d'une réponse

<code><command-response></code>	<code>::= "["<sequence>"]" <response></code>
<code><response></code>	<code>::= <value> <error-string> <emcy-list> "OK"</code>
<code><error-string></code>	<code>::= "Error:" <error code></code>
<code><error-code></code>	<code>::= <internal-error-code> <sdo-abort-code></code>
<code><internal-error-code></code>	<code>::= see table 2</code>
<code><sdo-abort-code></code>	<code>::= UNSIGNED32</code>
<code><emcy-list></code>	<code>::= [<emcy1> " " ..<emcy254>]</code>

Les codes d'annulation SDO (SAC) sont définis dans la EN 50325-4. Les codes d'erreurs internes admis (InEC) sont énumérés dans le Tableau 101:

Tableau 101 – Code d'erreurs internes (InEC)

InEC	Texte de message
100	Requête non prise en charge
101	Erreur de syntaxe
102	Requête non traitée en raison de l'état interne
103	Hors délai (le cas échéant)
200	Message de protection perdu
201	Connexion perdue
202	Pulsation initialisée
203	Pulsation perdue
205	Initialisation
300	Erreur passive
301	Bus hors tension
303	Débordement du tampon CAN
304	Initialisation CAN
305	CAN actif (à l'initialisation ou au démarrage)
400	PDO déjà utilisé
401	Dépassement de la longueur PDO

NOTE Après la mise hors tension du bus, il convient de solliciter la commande init pour initialiser le contrôleur CAN.

10.5.2.4.4 Messages déclenchés par un événement

Les messages dus à des erreurs internes au réseau CANopen ou à l'occurrence des objets de communication fonctionnant sur le principe producteur-consommateur ne doivent pas utiliser de numéro de séquence. La notation des messages déclenchés par un événement est donnée dans le Tableau 102.

Tableau 102 – Notation applicable aux messages déclenchés par un événement

<code><event-triggered-message></code>	<code>::= [[net] node] <event-spezifier> <parameter></code>
<code><event-spezifier></code>	<code>::= "EMCY" "ERROR" "PDO" "SYNC" "USER"</code>

Le contenu des messages déclenchés par un événement est spécifié dans la description de la commande qui active le service spécifique.

10.5.3 Spécifications des commandes d'accès au réseau

10.5.3.1 Commandes d'accès SDO

10.5.3.1.1 Généralités

Les définitions de commandes suivantes doivent être utilisées pour la mise en œuvre des services d'accès SDO définis en 10.4.

Les services d'accès SDO effectuent l'adressage d'un objet spécifique au niveau d'un serveur SDO par le biais d'un index et d'un sous-index et d'un type de données de transfert.

10.5.3.1.2 Commande SDO de téléchargement amont

La syntaxe d'expression est définie dans le Tableau 103.

Tableau 103 – Syntaxe d'une commande SDO de téléchargement amont

```
[[net] node] r[ead] <multiplexor> <datatype>
```

Des exemples sont fournis dans le Tableau 104.

Tableau 104 – Exemples d'une commande SDO de téléchargement amont

```
[21] r 0x1000 0 u32
[4096] lecture 0x1008 0 vs
```

Syntaxe de réponse:

Voir 10.5.2.4.3.

10.5.3.1.3 Commande SDO de téléchargement aval

La syntaxe d'expression est définie dans le Tableau 105.

Tableau 105 – Syntaxe d'une commande SDO de téléchargement aval

```
[[net] node] w[rite] <multiplexor> <datatype> <value>
```

Des exemples sont fournis dans le Tableau 106.

Tableau 106 – Exemples d'une commande SDO de téléchargement aval

```
[20] 1 23 w 0x1016 0 u16 100
[23] écriture 0x1016 0 u16 0x64
```

Syntaxe de réponse:

Voir 10.5.2.4.3.

10.5.3.1.4 Configuration de la commande de temporisation SDO

Le délai de temporisation applicable au code d'erreur d'annulation « Protocole SDO hors délai », utilisé par le client SDO de la passerelle, peut être mis à 1.

La syntaxe d'expression est définie dans le Tableau 107.

Tableau 107 – Syntaxe de configuration de la commande de temporisation SDO

```
[net] réglage sdo_timeout <ms>
```

Syntaxe de réponse:

Voir 10.5.2.4.3.

10.5.3.2 Commandes d'accès PDO

10.5.3.2.1 Généralités

Les définitions de commandes suivantes doivent être utilisées pour la mise en œuvre des services d'accès PDO définis en 10.4. La configuration d'une commande PDO précède normalement sa transmission et sa réception.

Un PDO est observé du point de vue de la passerelle. Un RPDO reçoit par conséquent les données émises par le réseau CANopen et un TPDO envoie les données vers le réseau CANopen.

NOTE Le bit 31 de l'indicatif COB permet de désactiver/supprimer le PDO d'une passerelle. Pour des détails, se reporter à la EN 50325-4.

10.5.3.2.2 Configuration de la commande RPDO

La syntaxe d'expression est définie dans le Tableau 108.

Tableau 108 – Syntaxe de configuration de la commande RPDO

<code>[[net] node] set rpdo <nr> <COB> <tx-type> <nr-of-data> <map-obj1> [..<map-obj64>]</code>
<code><tx-type> ::= "rtr" "event" "sync<0..240>"</code>

NOTE Dans le cas où une syntaxe <map-obj> est donnée sous forme d'index et de sous-index, par exemple, <multiplexor>, l'index et le sous-index sont comptés comme 1 dans la syntaxe <nr-of-data>. La syntaxe <nr> représente un décalage; elle commence par 1.

Des exemples sont fournis dans le Tableau 109.

Tableau 109 – Exemples de configuration de la commande RPDO

<code>[12] set rpdo 1 0x180 événement 3 u8 u8 u16</code>
<code>[24] 2 set rpdo 1 0x180 event 3 u8 u8 i16</code>

Syntaxe de réponse:
Voir 10.5.2.4.3.

10.5.3.2.3 Configuration de la commande TPDO

La syntaxe d'expression est fournie dans le Tableau 110.

Tableau 110 – Syntaxe de configuration de la commande TPDO

<code>[[net] node] set tpdo <nr> <COB> <tx-type> <nr-of-data> <map-obj1> [..<map-obj64>]</code>

NOTE Dans le cas où une syntaxe <map-obj> est donnée sous forme d'index et de sous-index, c'est-à-dire, <multiplexor>, l'index et le sous-index sont comptés comme 1 dans la syntaxe <nr-of-data>. La syntaxe <nr> représente un décalage; elle commence par 1.

Un exemple est fourni dans le Tableau 111.

Tableau 111 – Exemples de configuration de la commande TPDO

<code>[13] set tpdo 1 0x201 rtr 4 u8 u16 u16 u8</code>
--

Syntaxe de réponse:
Voir 10.5.2.4.3.

NOTE La prise en charge d'un RTR n'est pas recommandée. Il est par conséquent recommandé de régler le bit 30 de l'indicatif COB en conséquence. Pour des détails, voir la EN 50325-4.

10.5.3.2.4 Commande de lecture des données PDO

La syntaxe d'expression est définie dans le Tableau 112.

Tableau 112 – Syntaxe de la commande de lecture des données PDO

```
[net] r[ead] p[do] <nr>
```

Si le type de transmission est différent du RTR, la passerelle répond avec les valeurs des objets mis en correspondance.

La syntaxe de réponse est définie dans le Tableau 113.

Tableau 113 – Syntaxe de réponse pour la commande de lecture des données PDO

```
[net] pdo <nr> <nr-of-data> <value1>[..

```

10.5.3.2.5 Commande d'écriture des données PDO

La syntaxe d'expression est définie dans le Tableau 114.

Tableau 114 – Syntaxe de la commande d'écriture des données PDO

```
[net] w[rite] p[do] <nr> <nr-of-data> <value1>[..

```

Syntaxe de réponse:

Voir 10.5.2.4.3.

10.5.3.2.6 Commande RPDO reçu

La syntaxe d'expression est définie dans le Tableau 115.

Tableau 115 – Syntaxe de la commande RPDO reçu

```
[net] pdo <nr> <nr-of-data> <value1>[..

```

Des exemples sont fournis dans le Tableau 116.

Tableau 116 – Exemples de commande RPDO reçu

```
1 pdo 1 2 123 4      ;# passerelle avec deux réseaux ou plus
                      ;# RPDO1 reçu avec nombre net 1 et deux objets
                      ;# mis en correspondance
pdo 2 1 1234          ;# RPDO2 avec un objet mis en correspondance
pdo 2 3 100 2 4       ;# trois objets mis en correspondance
```

10.5.3.3 Commandes NMT CANopen

10.5.3.3.1 Généralités

Les définitions de commandes suivantes doivent être utilisées pour la mise en œuvre des services NMT CANopen définis en 10.4. Les services pris en charge dépendent de la classe de la passerelle.

10.5.3.3.2 Commande de démarrage de nœud

La syntaxe d'expression est définie dans le Tableau 117.

Tableau 117 – Syntaxe de la commande de démarrage de nœud

```
[[net] node] start
```

Syntaxe de réponse:

Voir 10.5.2.4.3.

10.5.3.3.3 Commande d'arrêt de nœud

La syntaxe d'expression est définie dans le Tableau 118.

Tableau 118 – Syntaxe de la commande d'arrêt de nœud

[[net] node] stop

Syntaxe de réponse:

Voir 10.5.2.4.3.

10.5.3.3.4 Commande de mise du nœud à l'état préopérationnel

La syntaxe d'expression est définie dans le Tableau 119.

Tableau 119 – Syntaxe de la commande de mise du nœud à l'état préopérationnel

[[net] node] preop[erational]

Syntaxe de réponse:

Voir 10.5.2.4.3.

10.5.3.3.5 Commande d'initialisation de nœud

La syntaxe d'expression est définie dans le Tableau 120.

Tableau 120 – Syntaxe de la commande d'initialisation de nœud

[[net] node] reset node

Syntaxe de réponse:

Voir 10.5.2.4.3.

10.5.3.3.6 Commande d'initialisation de la communication

La syntaxe d'expression est définie dans le Tableau 121.

Tableau 121 – Syntaxe de la commande d'initialisation de la communication

[net] node] reset comm[unication]

Syntaxe de réponse:

Voir 10.5.2.4.3.

10.5.3.3.7 Commande d'activation de la protection de nœud

L'activation de la fonctionnalité Protection de nœud active un autre message de réponse déclenché par un événement à destination des clients de la passerelle. Un message d'événement doit être envoyé aux clients uniquement en cas de violation du protocole de protection par un de nœud CANopen contrôlés.

La syntaxe d'expression est définie dans le Tableau 122.

Tableau 122 – Syntaxe de la commande d'activation de la protection de nœud

[[net] node] enable guarding <guardingtime> <lifetimefactor>
--

Syntaxe de réponse:

Voir 10.5.2.4.3.

10.5.3.3.8 Désactivation de la commande de protection de nœud

La syntaxe d'expression est définie dans le Tableau 123.

Tableau 123 – Syntaxe de la commande de désactivation de la protection de nœud

[[net] node] disable guarding

Syntaxe de réponse:

Voir 10.5.2.4.3.

10.5.3.3.9 Commande de démarrage du consommateur de pulsation

L'activation du consommateur de pulsation au niveau de la passerelle active un autre message de réponse déclenché par un événement à destination des clients de la passerelle. Un message d'événement doit être envoyé aux clients uniquement en cas de violation du protocole de protection par un des nœud CANopen contrôlés.

La syntaxe d'expression est définie dans le Tableau 124.

Tableau 124 – Syntaxe de la commande de démarrage du consommateur de pulsation

<code>[[net] node] enable heartbeat <heartbeattime></code>
--

Syntaxe de réponse:

Voir 10.5.2.4.3.

10.5.3.3.10 Désactivation de la commande du consommateur de pulsation

La syntaxe d'expression est définie dans le Tableau 125.

Tableau 125 – Syntaxe de la commande de désactivation du consommateur de pulsation

<code>[[net] node] disable heartbeat</code>

Syntaxe de réponse:

Voir 10.5.2.4.3.

10.5.3.3.11 Commande Événement contrôle d'erreurs reçu

La syntaxe d'expression est définie dans le Tableau 126.

Tableau 126 – Syntaxe de la commande Événement contrôle d'erreurs reçu

<code>[[net] node] ERROR <internal-error-code></code>

10.5.3.4 Commandes de gestion de défaillance d'un dispositif

10.5.3.4.1 Généralités

Les définitions de commandes suivantes doivent être utilisées pour la mise en œuvre des services de gestion de défaillance d'un dispositif définis en 10.4.4.

10.5.3.4.2 Commande de lecture d'erreur de dispositif

La syntaxe d'expression est définie dans le Tableau 127.

Tableau 127 – Syntaxe de la commande de lecture d'erreur de dispositif

<code>[[net] node] r[ead] error</code>
--

Syntaxe de réponse:

Voir 10.5.2.4.3.

10.5.3.4.3 Commande Événement de situation critique reçu

La syntaxe de réponse est définie dans le Tableau 128.

Tableau 128 – Syntaxe de la commande Événement situation critique reçu

<code>[[net] node] EMCY <emcy-code> <error-register> <m-error-code></code>
--

Le code d'erreur (de fabrication) doit être fourni sous la forme de valeurs à cinq décimales correspondant au code d'erreur spécifique au fabricant dans le message EMCY.

10.5.3.5 Commandes de configuration d'interface CANopen

10.5.3.5.1 Généralités

Les définitions de commandes suivantes doivent être utilisées pour la mise en œuvre des services de configuration d'interface CANopen définis en 10.4.5. Les paramètres de réglage sont valables pour le réseau par défaut, si aucune adresse de réseau n'est indiquée. L'adresse de nœud doit être omise si une adresse de réseau est fournie.

10.5.3.5.2 Commande d'initialisation de la passerelle

La syntaxe d'expression est définie dans le Tableau 129.

Tableau 129 – Syntaxe de la commande d'initialisation de la passerelle

```
[net] init <bitrate>
```

Le débit binaire doit être fourni comme index tabulaire de la table de débit binaire CANopen normal spécifiée dans le Tableau 130.

Tableau 130 – Index de débit binaire

Index	Débit binaire
0	1 Mbit/s
1	800 kbit/s
2	500 kbit/s
3	250 kbit/s
4	125 kbit/s
5	réservé
6	50 kbit/s
7	20 kbit/s
8	10 kbit/s
9	Détection automatique du débit binaire

Syntaxe de réponse:

Voir 10.5.2.4.3.

10.5.3.5.3 Commande d'enregistrement de configuration

Toutes les valeurs peuvent être enregistrées. L'enregistrement des valeurs peut être sélectif pour un service spécial. Toutes les valeurs doivent être enregistrées en l'absence d'indication d'argument.

La syntaxe d'expression est définie dans le Tableau 131. L'indicatif d'enregistrement est spécifié dans le Tableau 132.

Tableau 131 – Syntaxe de la commande d'enregistrement de la configuration

```
[net] store <storage-spezifier>
```

Tableau 132 – Spécificateur d'enregistrement

Spécificateur d'enregistrement	Description
CFG	indicatif, débit binaire, nœud par défaut, réseau par défaut
PDO	Numéro PDO, type de transmission, numéro de mise en correspondance, mise en correspondance
SDO	Temporisation SDO
NMT	<ul style="list-style-type: none"> - Protection de nœud: Nœud, temps de protection, facteur de durée de vie - Pulsation: nœud, durée de pulsation - Durée de pulsation du serveur

Syntaxe de réponse:

Voir 10.5.2.4.3.

10.5.3.5.4 Commande de restauration de la configuration

La syntaxe d'expression est définie dans le Tableau 133.

Tableau 133 – Syntaxe de la commande de restauration de la configuration

```
[net] restore <storage-spezifier>
```

Syntaxe de réponse:

Voir 10.5.2.4.3.

10.5.3.5.5 Commande de réglage du producteur de pulsation

La syntaxe d'expression est définie dans le Tableau 134.

Tableau 134 – Syntaxe de la commande de réglage du producteur de pulsation

```
[net] set heartbeat <ms>
```

Syntaxe de réponse:

Voir 10.5.2.4.3.

10.5.3.5.6 Commande de réglage de l'indicatif de nœud

La syntaxe d'expression est définie dans le Tableau 135.

Tableau 135 – Syntaxe de la commande de réglage de l'indicatif de nœud

```
[net] set id <value>
```

Syntaxe de réponse:

Voir 10.5.2.4.3.

10.5.3.5.7 Commande de démarrage du consommateur de messages de situation critique

Cette commande n'est pas spécifiée.

10.5.3.5.8 Commande d'arrêt du consommateur de messages de situation critique

Cette commande n'est pas spécifiée.

10.5.3.6 Commandes de gestion de la passerelle

10.5.3.6.1 Généralités

Les définitions de commandes suivantes doivent être utilisées pour la mise en œuvre des services de gestion de la passerelle définis en 10.4.6.

10.5.3.6.2 Commande de réglage du réseau par défaut

La syntaxe d'expression est définie dans le Tableau 136.

Tableau 136 – Syntaxe de la commande de réglage du réseau par défaut

```
[net] set network <value>
```

Syntaxe de réponse:

Voir 10.5.2.4.3.

10.5.3.6.3 Commande de réglage de l'indicatif de nœud par défaut

La syntaxe d'expression est définie dans le Tableau 137.

Tableau 137 – Syntaxe de la commande de réglage de l'indicatif de nœud par défaut

```
[net] set node <value>
```

Syntaxe de réponse:

Voir 10.5.2.4.3.

10.5.3.6.4 Commande d'obtention de la version

La syntaxe d'expression est définie dans le Tableau 138.

Tableau 138 – Syntaxe de la commande d'obtention de la version

```
info version
```

La réponse à la commande est l'information concernant la version actuelle de la passerelle sous la forme d'une liste séparée par des blancs. Les premiers éléments de la liste sont des valeurs présentes normalement dans le dictionnaire d'objets de la passerelle au niveau de l'objet 1018_n. La liste suivante définit les éléments qui contiennent la classe de passerelle, sous forme d'une combinaison de classes possibles et le numéro de version du protocole mis en œuvre correspondant à 10.4.6.4.

La syntaxe de réponse est définie dans le Tableau 139.

Tableau 139 – Syntaxe de réponse de la commande d'obtention de la version

```
<version-string>::= <vendor-id> <product-code>
                    <version-high>.<version-low> <serial-number>
                    <gateway-class> <protocol-version>
                    <implementation-class>
```

Un exemple de résultat est fourni dans le Tableau 140.

Tableau 140 – Exemple de réponse de la commande d'obtention de la version

```
[1234] 52 100 1.01 1234567 128 0.85 0.10
```

10.5.3.7 Commandes spécifiques au fabricant

Les définitions de commandes suivantes doivent être utilisées pour la mise en œuvre des services spécifiques au fabricant. Le fabricant de la passerelle peut ajouter des commandes à sa passerelle de manière à fournir des fonctions supplémentaires. Toutes les commandes spécifiques à l'utilisateur doivent être précédées d'un trait bas “_” afin d'éviter les erreurs de syntaxe dues à une commande inconnue.

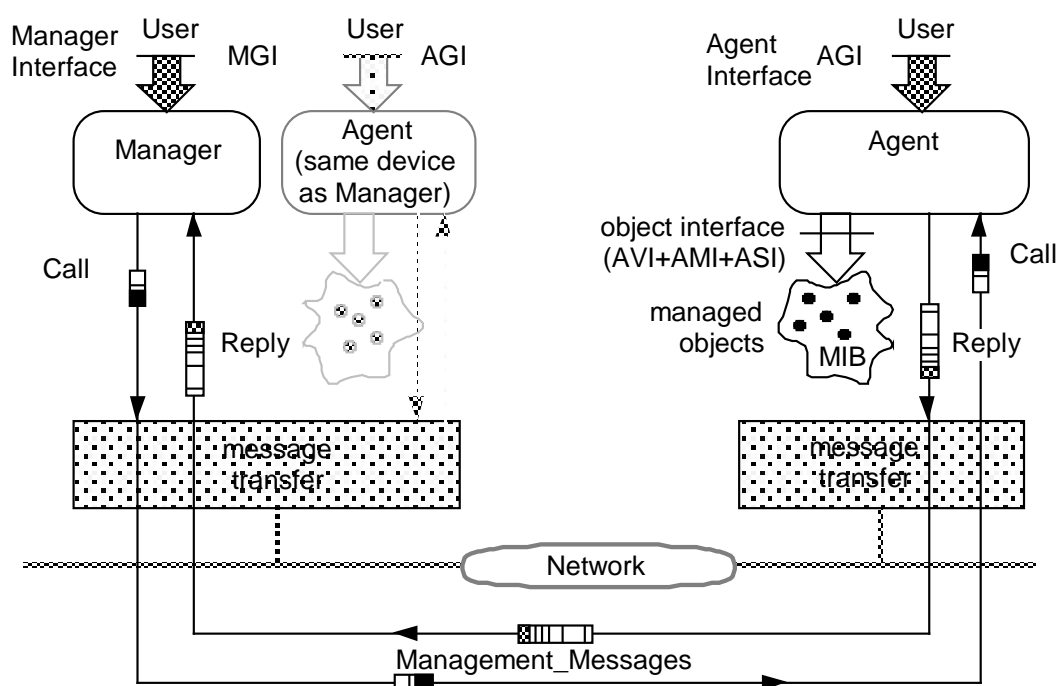
Le spécificateur d'événement « USER » doit être employé si des messages déclenchés par un événement non spécifiés dans la présente partie de la spécification doivent être utilisés.

11 Gestion de réseau de train

11.1 Teneur

Le présent article élargit l'ensemble des services de gestion de réseau de train (TNM) pour un nœud WTB spécifié dans la CEI 61375-2-1. La gestion de réseau de train définit les services d'aide à la mise en service, aux essais, à l'exploitation et à la maintenance d'un TCN. Ces services comprennent l'identification et le contrôle des stations, la distribution du routage et de la topographie, la lecture et le forçage de variables à distance, le téléchargement aval et amont et la gestion du réseau de train, ainsi que des couches de liaison du réseau de rame. Un processus de gestion demande l'exécution de ces services à distance, chaque station exécutant ces derniers par l'intermédiaire d'un processus Agent.

A titre d'information, le traitement des messages de gestion est illustré à la Figure 30, comme défini dans la CEI 61375-2-1.



Légende

Anglais	Français
Manager interface	Interface gestionnaire
User	Utilisateur
Agent interface	Interface agent
manager	gestionnaire
(same device as Manager)	(même dispositif que le Gestionnaire)
Object interface	Interface d'objets
Call	Appel
Reply	Réponse
Managed objects	Objets gérés
Message transfer	Transfert de message
Network	Réseau

Figure 30 – Messages de gestion (informatif)

Les services supplémentaires sont spécifiques aux stations connectées à un réseau de rame CANopen. Un gestionnaire doit demander l'exécution de ces services par l'envoi d'un Call_Message de gestion à l'Agent installé dans la Station cible. L'Agent doit exécuter le service et doit répondre par un Reply_Message de gestion avec indication du résultat du service. Les services relatifs à CANopen, qui peuvent être demandés à l'Agent, sont spécifiés en 10.4.

11.2 Gestionnaire, Agents et interfaces (informatif)

Les services de Gestion de Réseau de Train sont fournis dans chaque Station par un Agent, comme défini dans la CEI 61375-2-1. Le Station_Id de la Station de résidence de l'Agent identifie ce dernier. Un Gestionnaire demande les services de Gestion de Réseau de Train.

11.3 Protocole des messages de gestion (informatif)

Pour les besoins de la Gestion du Réseau de Train, le Gestionnaire et les Agents communiquent par le réseau en échangeant des messages de gestion, utilisant les services de messagerie du Réseau de Train Embarqué, comme illustré à la Figure 30 et défini dans la CEI 61375-2-1. Le Gestionnaire doit exercer les fonctions d'Appelant et l'Agent celles de Répondeur.

Le Gestionnaire accède à un objet distant en deux étapes: il envoie un Call_Message de gestion et l'Agent décode le message, puis il accède à l'objet en question et renvoie un Reply_Message de gestion, avec le résultat du service.

11.4 Interfaces d'objets (informatif)

Conformément à la CEI 61375-2-1, les objets de communication TCN se rapportent à la communication de réseau, tandis que les objets TCN de non-communication se rapportent à d'autres propriétés d'une Station.

NOTE 1 Des exemples sont fournis dans la CEI 61375-2-1.

L'Agent accède aux objets de communication par les interfaces définies pour l'accès général dans la présente norme, et en particulier à travers les interfaces:

- a) AVI (Application_Variables_Interface) pour les Variables;
- b) AMI (Application_Messages_Interface) pour les Messages;
- c) ASI (Application_Supervisory_Interface) pour les objets non accessibles par des processus d'utilisateur.

L'interface ASI fournit l'accès aux objets localisés dans les différentes couches de communication. L'Agent accède à ces objets par l'Interface de Gestion de Couche de la couche dans laquelle ils résident.

NOTE 2 L'entité qui effectivement accède aux objets dans chaque couche est appelée Entité de Gestion de Couche ou LME.

L'Agent doit également pouvoir accéder aux objets de non-communication. L'interface le permettant n'est pas spécifiée.

11.5 Services de gestion spécifiques CANopen

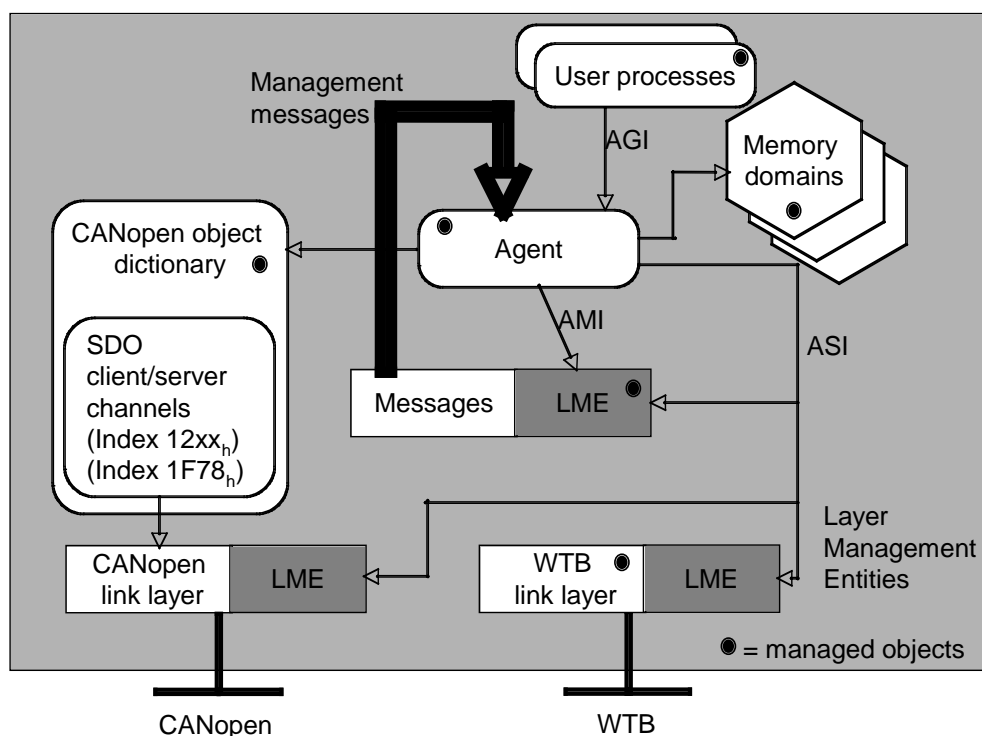
11.5.1 Généralités

Les messages de gestion spécifiques CANopen identifient un service au moyen de deux indicatifs, à savoir le SIF_code et le Command_code. Le SIF_code détermine un groupe parmi deux groupes de services - ces services sont exécutés avec et sans réservation du Gestionnaire pour son usage exclusif. Le Command_code sélectionne le service particulier du groupe. La charge utile du message d'appel représente la demande d'un service codé sous

forme de chaîne ASCII, la charge utile du message de réponse constituant la réponse codée de la même manière. Le codage ASCII est spécifié en 10.5.

11.5.2 Interfaces de l'Agent sur une Station connectée à un réseau de rame CANopen

Les définitions applicables aux interfaces de l'Agent, fournies dans la CEI 61375-2-1 s'appliquent également aux dispositifs CANopen qui communiquent les données de messagerie TCN. Les interfaces de l'Agent sont illustrées à la Figure 31 pour une passerelle entre le Réseau Central de Train et un réseau de rame basé sur CANopen. Dans la Figure 31, le Réseau Central de Train est mis en œuvre comme bus de train filaire (WTB).



Légende

Anglais	Français
Management messages	Messages de gestion
User processes	Processus utilisateur
Memory domains	Domaines de mémoire
CANopen object dictionary	Dictionnaire d'objets CANopen
Agent	Agent
SDO client/server channels (index)	Canaux client/serveur SDO (index)
CANopen link layer	Couche Liaison CANopen
WTB link layer	Couche Liaison WTB
Layer Management Entities	Entités de gestion de couches
Managed objects	Objets gérés

Figure 31 – Interface de l'Agent sur une station (passerelle) CANopen pour les données de messagerie

L'interface AVI n'est pas utilisée dans le cas d'une station CANopen. Les données d'application sont directement représentées dans des objets dédiés de l'intervalle d'index du dictionnaire d'objets spécifique au fabricant, ainsi que du dictionnaire d'objets CANopen normalisé, comme décrit en 8.2.

11.5.3 Structure des messages de gestion pour les réseaux de rame CANopen

Chaque service doit être appelé par un échange de message de gestion des appels/réponses au format défini dans le Tableau 141.

Tableau 141 – Structure des messages de gestion

```

Management_Message_Can ::= RECORD
{
  tnm_key    ENUM8 --      premier octet
  {
    CALL      ('02'H),          --      Appel (requête)
    REPLY      ('82'H),          --      Réponse (réponse)
  },
  message    ONE_OF [tnm_key] --      sélectionne appel ou réponse
  {
    [CALL]    Call_Mgt_Message_Can, --      voir 11.5.5
    [REPLY]    Reply_Mgt_Message_Can --      voir 11.5.6
  }
}

```

Le bit de poids fort de 'tnm_key' doit indiquer s'il s'agit d'un message d'appel ou de réponse.

11.5.4 Notation pour les SIF_codes spécifiques CANopen

Dans les services de messages de gestion, le SIF_code indique les services demandés. Dans le cas des réseaux de rame CANopen, le SIF_code indique le groupe de services CANopen demandés. Comme défini dans le Tableau 142, le bit de poids faible indique s'il s'agit d'un groupe de lecture ou d'écriture (de modification) de services. Les services de lecture peuvent être employés sans réservation préalable.

Tableau 142 – SIF_codes spécifiques CANopen

```

Sif_Code_Can ::= ENUM8 --      choix du groupe de services
{
  WRITE_CANOPEN_COMMAND          (91),
  READ_CANOPEN_COMMAND           (90)
}

```

11.5.5 Notation pour un message de gestion CANopen d'appel

La notation pour un message de gestion CANopen d'appel est définie dans le Tableau 143.

Tableau 143 – Notation pour un message de gestion CANopen d'appel

```

Call_Mgt_Message_Can ::= RECORD
{
  sif_code Sif_Code_Can, --      le second octet est le
    sif_code
  message_body ONE_OF [sif_code]
  {
    [WRITE_CANOPEN_COMMAND]    Call_Write_CANopen_Command,
    [READ_CANOPEN_COMMAND]     Call_Read_CANopen_Command
  }
}

```

11.5.6 Notation pour un message de gestion CANopen de réponse

La notation pour un message de gestion CANopen de réponse est définie dans le Tableau 144.

Tableau 144 – Notation pour un message de gestion CANopen de réponse

```

Reply_Mgt_Message_Can ::= RECORD
{
  sif_code Sif_Code_Can,      -- le second octet est le
                             sif_code
  message_body ONE_OF [sif_code]
  {
    [WRITE_CANOPEN_COMMAND]      Reply_Write_CANopen_Command,
    [READ_CANOPEN_COMMAND]      Reply_Read_CANopen_Command
  }
}

```

11.5.7 Notation pour les codes de commande de services Canopen TNM

Le code de Commande détermine le service particulier du groupe de services identifié par le SIF_code. Les Tableaux 145 et 146 définissent les codes de commande de services CANopen TNM pour les services nécessitant une réservation ou non.

Tableau 145 – Codes de commande de services CANopen TNM (réservation nécessaire)

```

Cmd_Code_R ::= ENUM8      -- choix du service nécessitant une réservation
{
  UPLOAD_SDO                (1),
  DOWNLOAD_SDO              (2),
  CONFIGURE_SDO_TIMEOUT     (3),
  CONFIGURE_RPDO            (4),
  CONFIGURE_TPDO            (5),
  WRITE_PDO_DATA            (6),
  START_NODE                (7),
  STOP_NODE                 (8),
  RESET_NODE                (9),
  RESET_COMMUNICATION       (10),
  ENABLE_NODE_GUARDING      (11),
  DISABLE_NODE_GUARDING     (12),
  START_HEARTBEAT_CONSUMER  (13),
  DISABLE_HEARTBEAT_CONSUMER (14),
  INITIALIZE_GATEWAY        (15),
  STORE_CONFIGURATION       (16),
  RESTORE_CONFIGURATION     (17),
  SET_HEARTBEAT_PRODUCER    (18),
  SET_NODE_ID               (19),
  SET_DEFAULT_NETWORK       (20),
  SET_DEFAULT_NODE          (21)
}

```

NOTE Les services de notification ne sont pas pris en considération, dans la mesure où ils ne peuvent pas être mis en œuvre (requête, réponse).

Tableau 146 – Codes de commande de services CANopen TNM (réservation non nécessaire)

```

Cmd_Code_NR ::= ENUM8      -- choix du service ne nécessitant pas de
réservation
{
  READ_PDO_DATA            (30),
  READ_DEVICE_ERROR        (31),
  GET_VERSION              (32)
}

```

11.6 Services CANopen TNM

11.6.1 Teneur

Le présent paragraphe définit les messages, identifiés par le code de commande et spécifiés par une chaîne de commande de service transmise dans le corps du message, au moyen duquel le service CANopen TNM associé doit être demandé et auquel une réponse doit être apportée.

11.6.2 Call_Write_CANopen_Command (avec réservation)

Ce message doit solliciter les services du groupe « avec réservation » (voir Tableau 145) identifiés par le code de commande et spécifiés par la chaîne de commandes de service transmise dans le corps du message. La Figure 32 spécifie la structure de commande et le Tableau 147 spécifie la définition des valeurs.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
tnm_key								sif_code = 91							
reserved1								cmd_code							
reserved2															
string_size															
Commande: ARRAY ALIGN16 [string_size] OF															
(CHARACTER8)								CHARACTER8 ou '00'H							

Figure 32 – Call_Write_CANopen_Command

Tableau 147 – Définition de valeur pour Call_Write_CANopen_Command

Call_Write_CANopen_Command ::= RECORD	
{	
reserved1 WORD8 (=0),	-- réservé
cmd_code Cmd_Code_R,	-- code de commande du service
reserved2 WORD16 (=0),	-- réservé
string_size UNSIGNED16,	-- jusqu'à 65535 caractères
command ARRAY ALIGN16 [string_size] OF CHARACTER8	-- chaîne de commandes de service
}	

11.6.3 Reply_Write_CANopen_Command (avec réservation)

Ce message doit répondre à la commande Call_Write_CANopen du groupe « avec réservation » (voir 11.6.2) identifiée par le code de commande et spécifiée par la chaîne de commandes de service transmise dans le corps du message. La Figure 33 spécifie la structure de commande et le Tableau 148 spécifie la définition des valeurs.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
tnm_key								sif_code = 91							
reserved1								get_sif_code							
reserved2															
string_size															
Réponse: ARRAY ALIGN16 [string_size] OF															
(CHARACTER8)								CHARACTER8 ou '00'H							

Figure 33 – Reply_Write_CANopen_Command

Tableau 148 – Définition de valeur pour Reply_Write_CANopen_Command

Reply_Write_CANopen_Command ::= RECORD	
{	
reserved1 WORD8 (=0),	-- réservé
cmd_code Cmd_Code_R,	-- code de commande du service
reserved2 WORD16 (=0),	-- réservé
string_size UNSIGNED16,	-- jusqu'à 65535 caractères
response ARRAY ALIGN16 [string_size] OF CHARACTER8	-- chaîne de réponses de service
}	

11.6.4 Call_Read_CANopen_Command (sans réservation)

Ce message doit solliciter les services du groupe « sans réservation » (voir Tableau 146) identifiés par le code de commande et spécifiés par la chaîne de commandes de service transmise dans le corps du message. La Figure 34 spécifie la structure de commande et le Tableau 149 spécifie la définition des valeurs.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
tnm_key								sif_code = 90							
reserved1								cmd_code							
reserved2															
string_size															
Commande: ARRAY ALIGN16 [string_size] OF															
(CHARACTER8)								CHARACTER8 ou '00'H							

Figure 34 – Call_Read_CANopen_Command (sans réservation)

Tableau 149 – Définition de valeur pour Call_Read_CANopen_Command (sans réservation)

Call_Read_CANopen_Command ::= RECORD	
{	
reserved1 WORD8 (=0),	-- réservé
cmd_code Cmd_Code_NR,	-- code de commande du service
reserved2 WORD16 (=0),	-- réservé
string_size UNSIGNED16,	-- jusqu'à 65535 caractères
command ARRAY ALIGN16 [string_size] OF CHARACTER8	-- chaîne de commandes de service
}	

11.6.5 Reply_Read_CANopen_Command (sans réservation)

Ce message doit répondre à la commande Call_Write_CANopen du groupe « sans réservation » (voir 11.6.4) identifiée par le code de commande et spécifiée par la chaîne de commandes de service transmise dans le corps du message. La Figure 35 spécifie la structure de commande et le Tableau 150 spécifie la définition des valeurs.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
tnm_key								sif_code = 90							
reserved1								get_sif_code							
reserved2															
string_size															
Réponse: ARRAY ALIGN16 [string_size] OF															
(CHARACTER8)								CHARACTER8 ou '00'H							

Figure 35 – Reply_Read_CANopen_Command (sans réservation)

Tableau 150 – Définition de valeur pour Reply_Read_CANopen_Command (sans réservation)

```

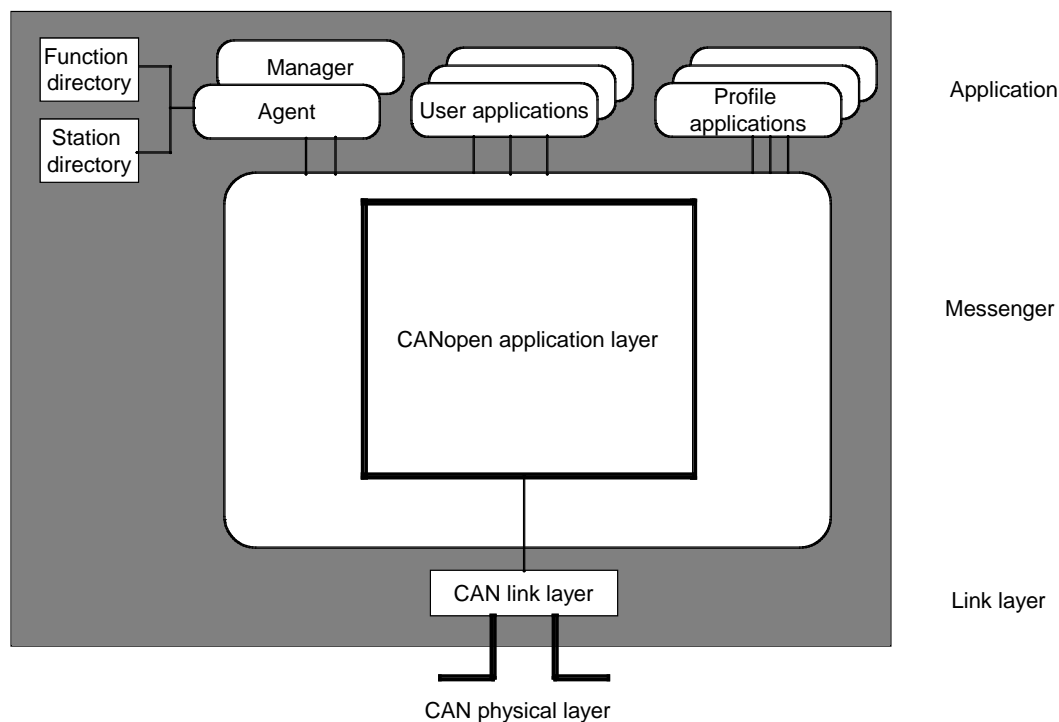
Reply_Read_CANopen_Command ::= RECORD
{
  reserved1 WORD8 (=0),           -- réservé
  cmd_code Cmd_Code_NR,           -- code de commande du service
  reserved2 WORD16 (=0),          -- réservé
  string_size UNSIGNED16,         -- jusqu'à 65535 caractères
  response ARRAY ALIGN16 [string_size] OF CHARACTER8 -- chaîne
                                de réponses de service
}

```

12 Traitement des données de messagerie de gestion CANopen

12.1 Généralités

Les messages de gestion CANopen peuvent être également communiqués entre les dispositifs CANopen d'un réseau de rame basé sur CANopen. La Figure 36 montre un dispositif CANopen capable de communiquer des données de messagerie via le réseau de rame basé sur CANopen.



Légende

Anglais	Français
Function directory	Répertoire de fonctions
Station directory	Répertoire de stations
Manager	Gestionnaire
Agent	Agent
User applications	Applications utilisateur
Profile applications	Applications de profil
Messenger	Messager
CANopen application layer	Couche Application CANopen
CAN link layer	Couche Liaison CAN
Link layer	Couche Liaison
CAN physical layer	Couche physique CAN

Figure 36 – Dispositif CANopen capable de traiter des messages de gestion TNM

Outre les applications spécifiques à l'utilisateur et au profil CANopen, un tel dispositif CANopen comporte l'application spécifique au Gestionnaire et à l'Agent. L'application du dispositif CANopen prend par ailleurs en charge le répertoire de Fonctions et de Stations. Pour la transmission et la réception des données de messagerie, le dispositif CANopen, comme illustré à la Figure 36, prend en charge les canaux serveur et client SDO, ainsi que l'entrée 1F78_h du dictionnaire d'objets CANopen.

Les données de messagerie sont transmises comme datagrammes, comme défini et décrit dans la CEI 61375-2-1. Un datagramme est similaire à une lettre: chaque datagramme comporte toutes les adresses nécessaires à son parcours de bout en bout (et au renvoi des acquittements). Ce schéma présente des avantages lorsque plusieurs bus sont connectés entre eux, dans la mesure où il n'est pas nécessaire que les routeurs conservent les informations concernant les messages précédents.

Chaque trame de données de messagerie comporte deux types d'adresses: Les adresses de dispositif émetteur et destinataire pour la communication interne à un bus (Adresse de Dispositif), et les adresses de la source d'origine et du destinataire final (Adresses de Réseau).

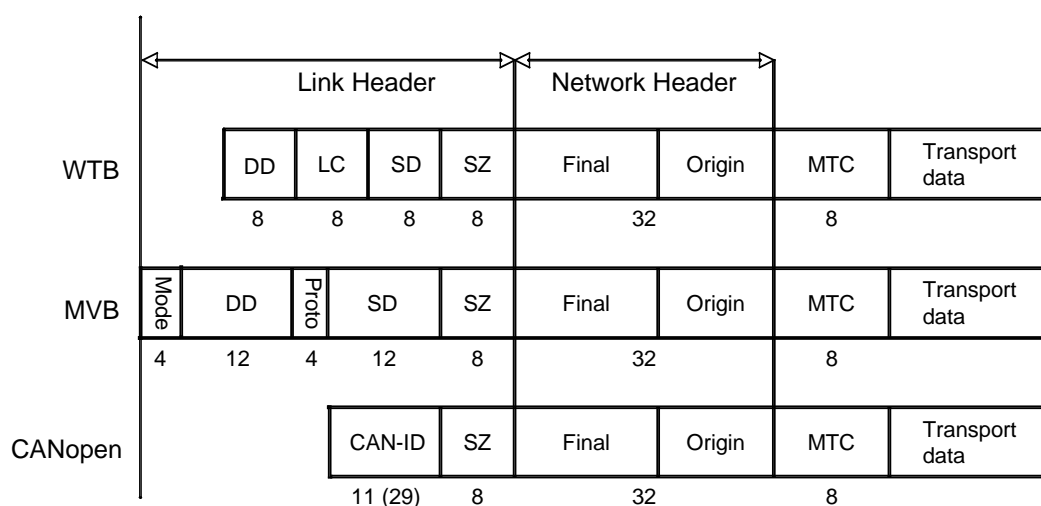
Les Adresses de Dispositif émetteur et destinataire s'appliquent uniquement au même bus. Lorsque des données de messagerie sont envoyées à un autre réseau de rame, le dispositif destinataire est le Nœud du Réseau Central de Train, qui intervient en qualité de Passerelle. Lorsque le Nœud reçoit des données de messagerie d'une autre rame, il insère son Adresse de Dispositif en tant que dispositif émetteur.

Les données de messagerie comportent deux types d'adresses:

- une adresse source et une adresse de destination, qui identifient les dispositifs communiquant sur le même bus et spécifiques à ce bus. Ces adresses sont spécifiques à la Couche de Liaison,
- une Adresse d'Origine et une Adresse Finale, qui identifient les Stations communiquant sur le réseau et qui sont connues dans l'ensemble du réseau. L'Adresse d'Origine et l'Adresse Finale identifient le Producteur et le Consommateur, puis l'Appelant et le Répondeur. Ces adresses appartiennent à la Couche Réseau.

12.2 Format des données de messagerie

Comme défini dans la CEI 61375-2-1 concernant les données de messagerie, la seule différence des trames existantes sur le WTB, le MVB, CANopen ou tout autre système de bus doit résider dans leur En-tête de Liaison, comme illustré à la Figure 37.



DD = destination Device Address LC = Link Control SZ = Link Data Size
SD = source Device Address MTC Message Transport Control Proto = Protocol Type

CAN-ID = CAN Identifier; determined by selected CANopen SDO channel
SDO channel = Communication channel between DD and SD

Légende

Anglais	Français
Link Header	En-tête de Liaison
Network Header	En-tête de réseau
Final	Final
Origin	Origine
Transport data	Données de transport
Destination Device Address	Adresse de Dispositif Destinataire
Link Control	Contrôle de Liaison
Link Data Size	Taille des données de liaison
Source Device Address	Adresse de Dispositif Emetteur
Message Transport Control	Contrôle de Transport des Messages
Protocol Type	Type de Protocole
CAN Identifier; determined by selected CANopen SDO channel	Indicatif CAN, déterminé par le canal SDO CANopen sélectionné
SDO channel	Canal SDO
Communication channel between DD and SD	Canal de communication entre DD et SD

Figure 37 – Comparaison des formats de données de messagerie

L'En-tête de Liaison comporte l'indicatif CAN (dans le cas de CANopen). Dans la mesure où les données de messagerie doivent être communiquées via SDO – communication point à point confirmée – l'adresse de dispositif destinataire ou émetteur est déterminée de manière implicite par la sélection du canal de communication SDO. Chaque canal de communication SDO fournit, par l'ensemble de paramètres SDO, un indicatif CAN unique pour les directions de requête et de réponse.

12.3 Exigences relatives à la communication des données de messagerie dans les réseaux CANopen

Tout dispositif CANopen, connecté à un réseau de rame basé sur CANopen, qui doit communiquer des données de messagerie, doit prendre en charge la fonctionnalité Client SDO, ainsi que Serveur SDO.

Chaque dispositif CANopen qui doit appeler les services à un autre dispositif de réseau de rame CANopen, doit prendre en charge un canal client SDO pour appeler un service. Le dispositif CANopen d'exécution et de réponse aux données de messagerie reçues doit transférer la réponse via un accès en écriture SDO à l'objet 1F78_h (voir 12.4), par l'intermédiaire du canal serveur SDO du "dispositif CANopen appelant".

Chaque dispositif CANopen qui doit recevoir une demande de service provenant d'un autre dispositif CANopen du réseau de rame CANopen, doit prendre en charge un canal serveur SDO pour recevoir ladite demande. Ce dispositif CANopen doit transmettre la réponse au "dispositif CANopen appelant" via un accès en écriture SDO par l'intermédiaire de son canal client SDO.

NOTE Il est recommandé que ce type de dispositif CANopen prenne en charge autant de canaux client et serveur SDO que le réseau CANopen comporte de dispositifs capables de communiquer des données de messagerie. Une paire de canaux client/serveur est pré-configurée pour chacun de ces dispositifs.

Les données de messagerie peuvent être transférées soit par un transfert à segments SDO ou par un transfert de blocs SDO, comme défini dans la EN 50325-4. Les données de messagerie doivent être reçues dans le sous-index approprié de l'index 1F78_h du dictionnaire d'objets CANopen (voir 12.4). Etant donné que chaque sous-index de cet objet est associé à un canal serveur SDO, le sous-index à écrire doit être égal au nombre de canaux serveur SDO utilisés.

12.4 Objet 1F78_h: Réception des données de messagerie CANopen

Les données de messagerie TCN doivent être reçues via les accès en écriture SDO à cet objet, comme défini en 12.3. Chaque sous-index de cet objet doit être associé à un canal serveur SDO pris en charge.

Les données constituées selon cet objet sont du type Domaine et doivent être interprétées comme défini dans la CEI 61375-2-1, données de messagerie TCN.

Les descriptions d'objet et d'entrée sont fournies dans les Tableaux 151 et 152.

Tableau 151 – Description d'objet

Attribut	Valeur
Index	1F78 _h
Nom	Réception des données de messagerie CANopen
Code d'objet	ARRAY
Type de données	DOMAIN
Catégorie	Conditionnel; obligatoire si prise en charge de la communication des données de messagerie

Tableau 152 – Description d'entrée

Attribut	Valeur
Sous-index	00 _h
Description	Nombre d'entrées
Catégorie d'entrée	Obligatoire
Accès	const
Mise en correspondance PDO	Non
Plage de valeurs	01 _h à 80 _h
Valeur par défaut	Spécifique au dispositif
Sous-index	01 _h
Description	Réception des données de messagerie TCN via le canal serveur SDO 1 spécifié par l'index 1200h
Catégorie d'entrée	Obligatoire
Accès	wo
Mise en correspondance PDO	Non
Plage de valeurs	Voir Article 11 et CEI 61375-2-1
Valeur par défaut	Spécifique au fabricant
Sous-index	02 _h
Description	Réception des données de messagerie TCN via le canal serveur SDO 2 spécifié par l'index 1201h
Catégorie d'entrée	Conditionnel; obligatoire si prise en charge de la communication des données de messagerie via le canal serveur SDO 2
Accès	wo
Mise en correspondance PDO	Non
Plage de valeurs	Voir Article 11 et CEI 61375-2-1
Valeur par défaut	Spécifique au fabricant
vers	
Sous-index	80 _h
Description	Réception des données de messagerie TCN via le canal serveur SDO 128 spécifié par l'index 127Fh
Catégorie d'entrée	Conditionnel; obligatoire si prise en charge de la communication des données de messagerie via le canal serveur SDO 128
Accès	wo
Mise en correspondance PDO	Non
Plage de valeurs	Voir Article 11 et CEI 61375-2-1
Valeur par défaut	Spécifique au fabricant

13 Essais de conformité

Les normes d'interface contrôleur-dispositif appropriées doivent spécifier les essais de type nécessaires à la vérification de la conformité de la conception d'un dispositif à la présente norme. Les équipements à soumettre à essai doivent inclure:

- l'alimentation,
- le dispositif réseau,
- le contrôleur,

- le support de communication,
- l'électromécanique.

Ces essais doivent inclure les essais électriques, les essais de compatibilité électromagnétique et les essais logiques. Toutes les interfaces de communication doivent être soumises à l'essai pour les passerelles entre le Réseau Central de Train et le réseau de rame basé sur CANopen.

La CEI 61375-2-2 spécifie le plan d'essai de conformité pour le Bus de train filaire (WTB). Le plan d'essai de conformité pour les dispositifs CANopen ne relève pas du domaine d'application de la présente norme.

NOTE 1 Le plan d'essai de conformité CANopen est spécifié dans le document CiA 310.

NOTE 2 Les informations relatives aux services associés aux essais de conformité sont disponibles auprès de la société CAN in Automation (CiA) GmbH.

Bibliographie

CEI 61375-2-4, *Matériel électronique ferroviaire – Réseau embarqué de train (TCN) – Partie 2-4: Profil d'Application*

CiA 301, *CANopen application layer and communication profile*. CAN in Automation e.V., Nuremberg

CiA 302, *CANopen additional application layer functions*. CAN in Automation e.V., Nuremberg

CiA 305, *CANopen layer setting services (LSS) and protocols*. CAN in Automation e.V., Nuremberg

CiA 303-1, *CANopen additional specification – Part 1: Cabling and connector pin assignment*. CAN in Automation e.V., Nuremberg

CiA 310, *CANopen conformance test plan*. CAN in Automation e.V., Nuremberg

CiA 421, *CANopen application profile for train vehicle control networks*. CAN in Automation e.V., Nuremberg

RFC 2045; *RFC 2045 Multipurpose Internet mail extensions*. www.rfc.net

UIC 556, *Transmission d'informations dans les trains (bus de train)*. Union Internationale des Chemins de fer (UIC), Bruxelles

INTERNATIONAL
ELECTROTECHNICAL
COMMISSION

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

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