

INTERNATIONAL STANDARD

NORME INTERNATIONALE



**Low-voltage switchgear and controlgear – Controller-device interfaces (CDIs) –
Part 3: DeviceNet**

**Appareillage à basse tension – Interfaces appareil de commande-appareil (CDI) –
Partie 3: DeviceNet**

IECNORM.COM : Click to view the full PDF of IEC 62026-3:2014



THIS PUBLICATION IS COPYRIGHT PROTECTED

Copyright © 2014 IEC, Geneva, Switzerland

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

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

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

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

About the IEC

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

About IEC publications

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

IEC Catalogue - webstore.iec.ch/catalogue

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

IEC publications search - www.iec.ch/searchpub

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

IEC Just Published - webstore.iec.ch/justpublished

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

Electropedia - www.electropedia.org

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

IEC Glossary - std.iec.ch/glossary

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

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

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

A propos de l'IEC

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

A propos des publications IEC

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

Catalogue IEC - webstore.iec.ch/catalogue

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

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

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

IEC Just Published - webstore.iec.ch/justpublished

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

Electropedia - www.electropedia.org

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

Glossaire IEC - std.iec.ch/glossary

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

Service Clients - webstore.iec.ch/csc

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

INTERNATIONAL STANDARD

NORME INTERNATIONALE



**Low-voltage switchgear and controlgear – Controller-device interfaces (CDIs) –
Part 3: DeviceNet**

**Appareillage à basse tension – Interfaces appareil de commande-appareil (CDI) –
Partie 3: DeviceNet**

INTERNATIONAL
ELECTROTECHNICAL
COMMISSION

COMMISSION
ELECTROTECHNIQUE
INTERNATIONALE

PRICE CODE
CODE PRIX

XF

ICS 29.130.20

ISBN 978-2-8322-1792-4

**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.....	8
INTRODUCTION.....	10
1 Scope.....	11
2 Normative references.....	11
3 Terms, definitions, symbols and abbreviated terms.....	12
3.1 Terms and definitions	12
3.2 Symbols and abbreviated terms	18
4 Classification.....	18
4.1 General.....	18
4.2 DeviceNet communication model	19
4.3 DeviceNet, CAN and CIP™	20
5 Characteristics	21
5.1 DeviceNet connections	21
5.1.1 General	21
5.1.2 DeviceNet's use of the CAN identifier field	21
5.1.3 Connection establishment	22
5.2 DeviceNet messaging protocol	23
5.2.1 Explicit messaging	23
5.2.2 Input/output messaging	34
5.2.3 Fragmentation/reassembly	35
5.2.4 Offline connection set	39
5.2.5 Device heartbeat.....	49
5.2.6 Device shutdown message.....	50
5.2.7 Duplicate MAC ID detection protocol	52
5.2.8 Quick connect.....	53
5.3 DeviceNet communication object classes.....	53
5.3.1 General	53
5.3.2 Identity object class definition (class ID code: 0x01)	53
5.3.3 Message router object class definition (class ID code: 0x02).....	54
5.3.4 DeviceNet object class definition (class ID code: 0x03).....	54
5.3.5 Connection object class definition (class ID code: 0x05)	54
5.3.6 Acknowledge handler object class definition (class ID code: 0x2B).....	55
5.4 Link access state machine.....	55
5.4.1 General	55
5.4.2 State transition diagram and event matrix	55
5.4.3 Duplicate MAC ID detection.....	58
5.5 Predefined master/slave connection set.....	58
5.5.1 General	58
5.5.2 Predefined master/slave connection set messages	59
5.5.3 DeviceNet object class specific services for the master/slave connection set.....	61
5.5.4 Slave connection object characteristics	68
5.5.5 Master connection object characteristics.....	73
5.5.6 Bit-strobe command/response messages.....	73
5.5.7 Poll command/response messages.....	74
5.5.8 Multicast poll command/response messages.....	75
5.5.9 Change of state/cyclic connections.....	75

5.5.10	Group 2 only devices	77
5.6	CIP Safety™ on DeviceNet.....	78
5.6.1	General	78
5.6.2	Use of CAN identifiers for CIP Safety on DeviceNet	78
5.7	Physical layer	78
5.7.1	General	78
5.7.2	Transceiver.....	80
5.7.3	Grounding.....	81
5.7.4	Isolation.....	81
5.7.5	Transmission medium	83
5.7.6	Topology.....	83
5.7.7	Link power	84
6	Product information	85
7	Normal service, mounting and transport conditions.....	85
7.1	Normal service conditions	85
7.1.1	General	85
7.1.2	Ambient air temperature.....	85
7.1.3	Altitude	86
7.1.4	Climatic conditions.....	86
7.2	Conditions during transport and storage	86
7.3	Mounting.....	86
8	Constructional and performance requirements.....	86
8.1	Indicators and configuration switches.....	86
8.2	DeviceNet cable.....	87
8.2.1	Overview	87
8.2.2	Cable profile template	87
8.2.3	Thick cable profile.....	88
8.2.4	Thin cable profile.....	92
8.2.5	Flat cable profile.....	95
8.3	Terminating resistors.....	98
8.4	Connectors	98
8.4.1	General specifications.....	98
8.4.2	Connector profile template	98
8.4.3	Open connector profile.....	99
8.4.4	Sealed mini connector profile	102
8.4.5	Sealed micro connector profile	103
8.4.6	Flat trunk connector profile.....	104
8.5	Device taps and power taps.....	107
8.5.1	Device taps.....	107
8.5.2	Power taps.....	108
8.6	Link powered devices	109
8.7	Miswiring protection	109
8.8	Power supplies.....	109
8.9	Electromagnetic compatibility (EMC).....	110
8.9.1	General	110
8.9.2	Immunity.....	110
8.9.3	Emissions	111
8.10	Additional functional safety requirements related to EMC	112
9	Tests.....	112

9.1	General.....	112
9.2	Electrical and EMC testing	112
9.2.1	Test of the DeviceNet power supply.....	112
9.2.2	Device peak current consumption	113
9.2.3	Power ON behaviour	113
9.2.4	Reverse connection of V+ and V-.....	114
9.2.5	Disconnection of V-	115
9.2.6	Differential input impedance test	115
9.2.7	Transmit levels	115
9.2.8	Acknowledge delay	116
9.2.9	CDI tests	117
9.2.10	Electromagnetic compatibility testing	118
9.3	Logical testing.....	120
9.3.1	General	120
9.3.2	Duplicate MAC ID check test	120
9.3.3	UCMM	121
9.3.4	Allocation of predefined master/slave connection set – Explicit messaging connection.....	122
9.3.5	Allocation of predefined master/slave connection set – I/O messaging connection	122
9.3.6	Logical testing of safety products	123
	Bibliography	124
	Figure 1 – Typical DeviceNet controller-device interfaces	19
	Figure 2 – DeviceNet protocol architecture compared with the OSI reference model.....	20
	Figure 3 – DeviceNet’s use of the CAN identifier field	21
	Figure 4 – Explicit message CAN data field use.....	23
	Figure 5 – Explicit message data field format.....	23
	Figure 6 – Explicit message header format.....	24
	Figure 7 – Service field format	24
	Figure 8 – Open explicit messaging connection request format	26
	Figure 9 – Open explicit messaging connection response format.....	28
	Figure 10 – Close connection request format.....	30
	Figure 11 – Close connection response format	30
	Figure 12 – Non-fragmented explicit request message format, values 0 – 3.....	32
	Figure 13 – Non-fragmented explicit request message format, value 4	33
	Figure 14 – Non-fragmented success response message format	33
	Figure 15 – Error response message.....	34
	Figure 16 – Data field of an I/O message.....	34
	Figure 17 – Format of DeviceNet fragmentation protocol.....	35
	Figure 18 – I/O message fragment format.....	36
	Figure 19 – Explicit message fragment format	36
	Figure 20 – Acknowledgement message format	38
	Figure 21 – Establishing the offline ownership	40
	Figure 22 – Multicast nature of the offline ownership	41
	Figure 23 – Offline ownership request message.....	42

Figure 24 – Offline ownership response message protocol.....	42
Figure 25 – Communication faulted request message – Multicast protocol.....	43
Figure 26 – Communication faulted request message – Point-to-point protocol.....	44
Figure 27 – Identify communication faulted request message – Multicast protocol	45
Figure 28 – Communication faulted identify response message.....	46
Figure 29 – Identify communication faulted request message – Point-to-point protocol	46
Figure 30 – Who communication faulted request message	47
Figure 31 – Who response message.....	48
Figure 32 – Change MAC ID communication faulted request message	48
Figure 33 – Device heartbeat message.....	49
Figure 34 – Device shutdown message	51
Figure 35 – Duplicate MAC ID check CAN identifier field.....	52
Figure 36 – Duplicate MAC ID check message data field format.....	52
Figure 37 – Link access state transition diagram	56
Figure 38 – Allocation choice byte contents	61
Figure 39 – Allocate_master/slave_connection_set request message.....	62
Figure 40 – Success response to allocate_master/slave_connection_set request	63
Figure 41 – Parent explicit messaging connection logic	65
Figure 42 – Release choice byte contents	66
Figure 43 – Release_master/slave_connection set request message	66
Figure 44 – Success response to release_master/slave_connection_set request.....	67
Figure 45 – Predefined master/slave I/O connection state transition diagram	69
Figure 46 – Predefined master/slave explicit messaging connection state transition diagram.....	71
Figure 47 – Physical layer block diagram.....	79
Figure 48 – Device containing a non-isolated physical layer.....	82
Figure 49 – Device containing an isolated physical layer.....	83
Figure 50 – DeviceNet medium topology	84
Figure 51 – Thick cable: physical configuration.....	91
Figure 52 – Thick cable: current available on the DeviceNet power bus.....	91
Figure 53 – Thin cable: physical configuration	94
Figure 54 – Thin cable: current available on the DeviceNet power bus	94
Figure 55 – Flat cable: physical configuration	97
Figure 56 – Flat cable: current available on the DeviceNet power bus	97
Figure 57 – Open connector pinout	101
Figure 58 – Open connector geometry.....	101
Figure 59 – Mini connector pinout	103
Figure 60 – Micro connector pinout	104
Figure 61 – Flat trunk connector layout – Part 1	106
Figure 62 – Flat trunk connector layout – Part 2	107
Figure 63 – Power supply rise time test circuit.....	112
Figure 64 – Current consumption test circuit.....	113
Figure 65 – Power ON test circuit.....	114

Figure 66 – Test circuit for reversal of V+ and V- and also V- interruption	114
Figure 67 – Differential impedance test circuit	115
Figure 68 – Transmit level test setup	116
Figure 69 – Transmit levels	116
Figure 70 – Timing test setup	117
Figure 71 – Timing	117
Figure 72 – CDI test configuration	118
Figure 73 – Test circuit for logical tests	120
Table 1 – Message body format values	27
Table 2 – Group select values	27
Table 3 – Source message ID in open explicit messaging connection request	28
Table 4 – Destination message ID in open explicit messaging connection response	29
Table 5 – UCMM error conditions/codes	31
Table 6 – Fragment type bit values	35
Table 7 – Ack status bit values	38
Table 8 – Offline connection set	40
Table 9 – Addresses reporting based upon mask	44
Table 10 – Device shutdown message shutdown code ranges	51
Table 11 – Device shutdown message “Open” shutdown codes	52
Table 12 – Link access state event matrix (1 of 2)	56
Table 13 – Predefined master/slave connection set identifier fields	60
Table 14 – Allocate_master/slave_connection_set request service data field parameters	61
Table 15 – Allocate_master/slave_connection_set response parameters	62
Table 16 – Release_master/slave_connection_set request service data field parameters	66
Table 17 – DeviceNet object specific additional error codes	68
Table 18 – Connection instance IDs for predefined master/slave connections	69
Table 19 – Predefined master/slave I/O connection state event matrix (1 of 2)	70
Table 20 – Predefined master/slave explicit messaging connection state event matrix	72
Table 21 – Predefined master/slave I/O connection object attribute access	73
Table 22 – General physical layer characteristics	79
Table 23 – Transmitter characteristics	80
Table 24 – Receiver characteristics	81
Table 25 – Load limits	85
Table 26 – Cable profile: data pair specifications	87
Table 27 – Cable profile: DC power pair specifications	87
Table 28 – Cable profile: general specifications	88
Table 29 – Cable profile: topology	88
Table 30 – Thick cable: data pair specifications	89
Table 31 – Thick cable: DC power pair specifications	89
Table 32 – Thick cable: general specifications	90
Table 33 – Thick cable: topology	90

Table 34 – Thick cable: maximum current available (A) based on network length	91
Table 35 – Thin cable: data pair specifications	92
Table 36 – Thin cable: DC power pair specifications	93
Table 37 – Thin cable: general specifications	93
Table 38 – Thin cable: topology	93
Table 39 – Thin cable: maximum current available (A) based on network length	94
Table 40 – Flat cable: data pair specifications	95
Table 41 – Flat cable: DC power pair specifications	96
Table 42 – Flat cable: general specifications	96
Table 43 – Flat cable: topology	96
Table 44 – Flat cable: maximum current available (A) based on network length	97
Table 45 – Connector profile template	99
Table 46 – Open connector	100
Table 47 – Sealed mini connector	102
Table 48 – Sealed micro connector (1 of 2)	103
Table 49 – Flat trunk connector	105
Table 50 – Internal pass-through conductor specifications	108
Table 51 – Internal drop conductor specifications	108
Table 52 – Internal pass-through conductor specifications	108
Table 53 – Internal power drop conductor specifications	109
Table 54 – Voltage regulator specifications	109
Table 55 – DeviceNet power supply specifications	110
Table 56 – Immunity performance criteria	111

INTERNATIONAL ELECTROTECHNICAL COMMISSION

**LOW-VOLTAGE SWITCHGEAR AND CONTROLGEAR –
CONTROLLER-DEVICE INTERFACES (CDIs) –****Part 3: DeviceNet****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 62026-3 has been prepared by subcommittee 121A: Low-voltage switchgear and controlgear, of IEC technical committee 121: Switchgear and controlgear and their assemblies for low voltage.

This third edition of IEC 62026-3 cancels and replaces the second edition published in 2008. This third edition constitutes a technical revision.

The main changes with respect to the previous edition are the followings:

- specification of group 4 messages (5.1.2);
- clarifications on messaging protocol (5.2);
- addition of I/O multicast poll messages (5.5.2 and 5.5.8);
- clarifications on slave behaviour (5.5.4 and 5.5.9);
- clarifications on physical layer (transceiver) in 5.7;

- miscellaneous corrections/clarifications on cable (8.2);
- clarifications on EMC testing (9.2.10) and logical testing (9.3).

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

FDIS	Report on voting
17B/1814/FDIS	121A/18/RVC

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 the IEC 62026, under the general title *Low-voltage switchgear and controlgear – Controller-device interfaces (CDIs)*, 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.

The contents of the corrigenda of March 2015 and October 2019 have been included in this copy.

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

INTRODUCTION

DeviceNet™¹ is intended for use in, but is not limited to, industrial automation applications. These applications may include devices such as limit switches, proximity sensors, electro-pneumatic valves, relays, motor starters, operator interface panels, analogue inputs, analogue outputs and controllers.

IECNORM.COM : Click to view the full PDF of IEC 62026-3:2014

¹ DeviceNet™ is a trade name of ODVA, Inc. This information is given for the convenience of users of this International Standard and does not constitute an endorsement by IEC of the trademark holder or any of its products. Compliance to this standard does not require use of the trade name DeviceNet™. Use of the trade name DeviceNet™ requires permission of ODVA, Inc.

LOW-VOLTAGE SWITCHGEAR AND CONTROLGEAR – CONTROLLER-DEVICE INTERFACES (CDIs) –

Part 3: DeviceNet

1 Scope

This part of IEC 62026 specifies an interface system between single or multiple controllers, and control circuit devices or switching elements. The interface system uses two conductor pairs within one cable – one of these pairs provides a differential communication medium and the other pair provides power to the devices. This part establishes requirements for the interoperability of components with such interfaces.

This part of IEC 62026 specifies the following particular requirements for DeviceNet:

- requirements for interfaces between controllers and switching elements;
- normal service conditions for devices;
- constructional and performance requirements;
- tests to verify conformance to requirements.

These particular requirements apply in addition to the general requirements of IEC 62026-1.

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 60529:1989, *Degrees of protection provided by enclosures (IP Code)*
IEC 60529:1989/AMD 1:1999
IEC 60529:1989/AMD 2:2013

IEC 60947-5-2:2007, *Low-voltage switchgear and controlgear – Part 5-2: Control circuit devices and switching elements – Proximity switches*
IEC 60947-5-2:2007/AMD 1:2012

IEC 61000-4-2:2008, *Electromagnetic compatibility (EMC) – Part 4-2: Testing and measurement techniques – Electrostatic discharge immunity test*

IEC 61000-4-3:2006, *Electromagnetic compatibility (EMC) – Part 4-3: Testing and measurement techniques – Radiated, radio-frequency, electromagnetic field immunity test*
IEC 61000-4-3:2006/AMD 1:2007
IEC 61000-4-3:2006/AMD 2:2010

IEC 61000-4-4:2012, *Electromagnetic compatibility (EMC) – Part 4-4: Testing and measurement techniques – Electrical fast transient/burst immunity test*

IEC 61000-4-5:2005, *Electromagnetic compatibility (EMC) – Part 4-5: Testing and measurement techniques – Surge immunity test*

IEC 61000-4-6:2013, *Electromagnetic compatibility (EMC) – Part 4-6: Testing and measurement techniques – Immunity to conducted disturbances, induced by radio-frequency fields*

IEC 61158-4-2:2014, *Industrial communication networks – Fieldbus specifications – Part 4-2: Data-link layer protocol specification – Type 2 elements*

IEC 61158-5-2:2014, *Industrial communication networks – Fieldbus specifications – Part 5-2: Application layer service definition – Type 2 elements*

IEC 61158-6-2:2014, *Industrial communication networks – Fieldbus specifications – Part 6-2: Application layer protocol specification – Type 2 elements*

IEC 61784-3-2, *Industrial communication networks – Profiles – Part 3-2: Functional safety fieldbuses – Additional specifications for CPF 2*

IEC 62026-1:2007, *Low-voltage switchgear and controlgear – Controller-device interfaces (CDIs) – Part 1: General rules*

CISPR 11:2009, *Industrial, scientific and medical equipment – Radio-frequency disturbance characteristics – Limits and methods of measurement*
CISPR 11:2009/AMD 1:2010

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

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

ANSI B93.55M-1981 (R1988), *Hydraulic Fluid Power – Solenoid Piloted Industrial Valves – Interface Dimensions for Electrical Connectors*

ASTM D 4566-942, *Standard Test Methods for Electrical Performance Properties of Insulations and Jackets for Telecommunications Wire and Cable*

3 Terms, definitions, symbols and abbreviated terms

For the purposes of this document, the terms, definitions, symbols and abbreviations given in IEC 62026-1 as well as the following apply.

3.1 Terms and definitions

3.1.1

acknowledged fragmentation

fragmentation performed on an explicit message, in which the transmission of a fragment from the transmitting object is followed by the transmission of an acknowledgement by the receiving object

Note 1 to entry: The reception of each fragment is acknowledged by the receiving object.

² A newer version of this document exists (ASTM D4566-08e1), however the listed revision applies for this standard

3.1.2**ack status**

field within an acknowledgement/response message format that indicates whether or not an error has been encountered by the receiver of a fragmented message

Note 1 to entry: This applies specifically to the DeviceNet fragmentation protocol.

3.1.3**application objects**

set of object classes and their object instances that are available within the node

Note 1 to entry: These objects manage and provide the exchange of data and messages across DeviceNet controller-device interfaces (CDIs) and within the DeviceNet compliant node.

3.1.4**attribute**

description of an externally accessible characteristic or feature of an object

Note 1 to entry: Attributes typically provide status information or govern the operation of an object.

3.1.5**bit-strobe**

communication using strobing

3.1.6**broadcast**

communication from one node to all other nodes

3.1.7**BOI attribute****bus-off interrupt attribute**

attribute of the DeviceNet object that defines the behaviour of a device after encountering a bus-off event in the CAN chip

Note 1 to entry: See IEC 61158-4-2:2014, 7.7.4.4 for further details.

3.1.8**CAN**

ISO specification that defines a generic physical layer and data link medium access procedure based on non-destructive bit-wise arbitration

Note 1 to entry: See ISO 11898-1 and ISO 11898-2.

Note 2 to entry: CAN is the abbreviation of "controller area network".

3.1.9**CAN_H**

positive half of the differential physical CAN signal

3.1.10**CAN_L**

negative half of the differential physical CAN signal

3.1.11**client**

(1) object which uses the services of another (server) object to perform a task;

(2) initiator of a message to which a server reacts

Note 1 to entry: See server (3.1.44).

[SOURCE: IEC 61158-4-2:2014, 3.4.9, MODIFIED]

3.1.12

common service

CIP service used by DeviceNet objects

Note 1 to entry: See IEC 61158-5-2:2014, 6.2.1.3 and IEC 61158-6-2:2014, 4.1.8.

3.1.13

communication objects

objects that manage and provide run-time exchange of messages across DeviceNet

3.1.14

connection

logical binding between two or more application objects

Note 1 to entry: These application objects can be located at the same node or at different nodes.

3.1.15

CID

connection ID

connection identifier assigned to all transmissions that are associated with a particular connection between multiple nodes

3.1.16

connection object

manages the communication-specific aspects associated with connections between nodes

3.1.17

consumer

end point of a connection that is responsible for receiving data

3.1.18

destination MAC ID

MAC ID of a node that is to receive a message

3.1.19

device tap

physical point of attachment from a DeviceNet device to a trunk cable or a drop cable

3.1.20

device type

identification of a collection of device-dependent information describing a viable combination of options selected for all layers in the communication stack

3.1.21

dominant

one of two complementary logic levels on the physical signal

Note 1 to entry: The dominant level is a logical "0".

3.1.22

duplicate MAC ID detection

DeviceNet-defined protocol that ensures no two nodes on the same link are assigned the same MAC ID

3.1.23**explicit messaging**

each explicit message commands the performance of a particular task and the return of the results of the task performance to the requester

3.1.24**fragmentation**

DeviceNet protocol provided by the connection object that defines a method by which data greater than eight (8) bytes in length may be transmitted

3.1.25**group 2 client**

unconnected message manager (UCMM) capable device that has gained ownership of the predefined master/slave connection set within a server such that it may act as the client on those connections

3.1.26**group 2 only client**

device that is acting as a group 2 client to a group 2 only server

Note 1 to entry: The group 2 only client provides the UCMM functionality for group 2 only servers that it has allocated.

3.1.27**group 2 server**

unconnected message manager (UCMM) capable device that has been configured to act as the server for the predefined master/slave identifier connections

3.1.28**group 2 only server**

slave device that is UCMM incapable and uses the predefined master/slave connection set to establish communications

Note 1 to entry: A group 2 only device can transmit and receive only those identifiers defined by the predefined master/slave connection set.

3.1.29**I/O connection**

connection between a producer and one or more consumers for the purpose of exchanging application-specific, time-critical I/O data

3.1.30**I/O data**

information which is transferred between I/O points and the controllers which use and set the values

3.1.31**I/O messaging**

exchange of data in a previously defined format

3.1.32**isolated device**

device in which some of its components are not referenced to the V- of the physical layer

Note 1 to entry: See non-isolated device (3.1.37).

3.1.33**master**

node which gathers and distributes I/O using the predefined master/slave connection set

3.1.34

MAC ID

link address of a DeviceNet node

Note 1 to entry: MAC ID is the abbreviation of “medium access control ID”.

3.1.35

multicast connection

logical connection from one object to multiple other objects

Note 1 to entry: A multicast connection allows data to be transferred in a single transaction from a producer to multiple consumers sharing the same connection.

3.1.36

node

DeviceNet entity which is identified at the data link level by a unique MAC ID

Note 1 to entry: Multiple DeviceNet nodes can be implemented in one device but they appear as logically distinct nodes on the DeviceNet link.

3.1.37

non-isolated device

device in which all components are referenced to the V- of the physical layer

Note 1 to entry: See isolated device (3.1.32).

3.1.38

object

(1) abstract representation of a device's capabilities;

Note 1 to entry: Objects can be composed of any or all of the following components:

- a) Data (information which changes with time);
- b) Configuration (parameters for behaviour);
- c) Procedures (actions that can be performed using data and configuration)

(2) collection of related data (in the form of variables) and procedures for operating on that data.

3.1.39

point-to-point connection

connection that exists between two objects only

Note 1 to entry: Explicit messaging connections are always point-to-point. I/O connections can be either point-to-point or multicast, see multicast connection (3.1.35).

3.1.40

predefined master/slave connection set

utilization of explicit messaging and I/O messaging connections to create and configure predefined connection objects within each connection end-point

Note 1 to entry: Uses the general rules as a basis for the definition of a set of connections which facilitate communications typically seen in a master/slave relationship.

3.1.41

producer

end point of a connection that is responsible for sending data

3.1.42

recessive

one of two complementary logic levels on the physical signal

Note 1 to entry: The recessive level is a logical "1".

3.1.43

serial number

unique 32-bit integer assigned by each manufacturer to every DeviceNet device

Note 1 to entry: The number is stored within the device as an attribute of the identity object and is unique with respect to the manufacturer.

3.1.44

server

object which provides services to another (client) object

Note 1 to entry: See client (3.1.11).

[SOURCE: IEC 61158-4-2:2014, 3.4.62, MODIFIED]

3.1.45

service

operation or function that an object performs upon request from another object

3.1.46

slave

node that receives data from and returns data to its master using the predefined master/slave connection set and a communication method set by the master

3.1.47

source MAC ID

MAC ID of a node that is transmitting a message

3.1.48

trigger

service used by an application to initiate the production of data

3.1.49

UCMM capable device

device that supports the UCMM

Note 1 to entry: See 3.1.52.

3.1.50

UCMM incapable device

device that does not support the UCMM

Note 1 to entry: See 3.1.52.

3.1.51

unconnected explicit message

explicit message between nodes that have not yet established a connection between each other

3.1.52

UCMM

unconnected message manager

function within a node that receives and processes unconnected explicit messages

3.1.53

USINT

8-bit integer

Note 1 to entry: USINT is the abbreviation of “unsigned short integer”.

3.1.54

UINT

16-bit integer

Note 1 to entry: UINT is the abbreviation of “unsigned integer”.

3.2 Symbols and abbreviated terms

BOI	bus-off interrupt
CAN	controller area network
CDI	controller-device interface, see IEC 62026-1
CID	connection identifier
CIP™	Common Industrial Protocol
CRC	cyclic redundancy check
DCR	DC resistance
EUT	equipment under test
LED	Light-emitting diode
MAC ID	medium access control identifier
UCMM	unconnected message manager
UINT	unsigned integer
USINT	unsigned short integer

4 Classification

4.1 General

DeviceNet interfaces controlling devices to control circuit devices or switching elements. DeviceNet uses two conductor pairs within one cable – one of these pairs provides a differential communication medium, and the other pair provides power to the devices. The maximum current supported is 8 A at 24 V d.c. Data is transmitted at bit rates of 125 kbit/s, 250 kbit/s or 500 kbit/s with maximum cable lengths of 500 m, 250 m and 100 m respectively. A maximum of 8 bytes of data may be transmitted without fragmentation. A maximum of 64 nodes may be connected using a linear topology with a trunk line and drop lines (see Figure 1). DeviceNet supports the transmission of I/O data, diagnostics, messaging and programming/configuration. Data exchange may be event driven (change of state), cyclic, polled, strobed, or multicast polled.

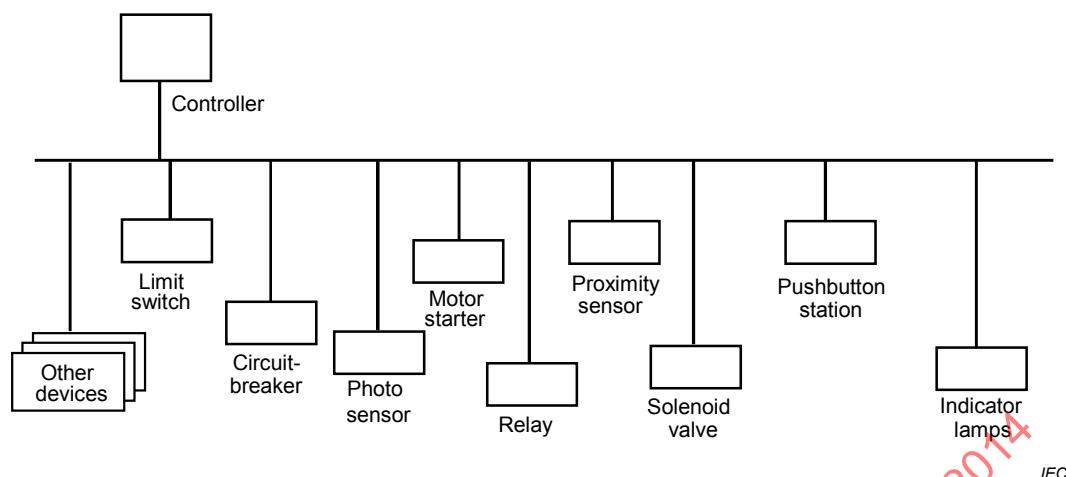


Figure 1 – Typical DeviceNet controller-device interfaces

This part of IEC 62026 defines a connection based scheme to facilitate all application communications. A DeviceNet connection provides a communication path between multiple end-points. The end-points of a connection are applications that need to share data. Transmissions associated with a particular connection are assigned an identification value when a connection is established. This identification value is called the connection ID (CID).

Connection objects model the communication characteristics of a particular application-to-application(s) relationship.

DeviceNet's connection based scheme defines a dynamic means by which the following two types of connections may be established:

- **I/O connections:** Provide dedicated, special purpose communication paths between a producing application and one or more consuming applications. Application specific I/O data moves through these paths.

I/O messages are exchanged across I/O connections. An I/O message consists of a CID and associated I/O data. The connection end-points shall have knowledge of the intended use or meaning of the I/O message.

This part of IEC 62026 does not define any particular use for I/O messaging. There are a wide variety of functions that may be accomplished using I/O messaging. The meaning and/or intended use of all I/O messages shall be made known to the system either by the particular type of product transmitting an I/O message, or based upon configuration performed using explicit messaging.

- **Explicit messaging connections:** Provide generic, multi-purpose communication paths between two devices. Explicit messages provide the typical request/response oriented communications.

Explicit messages are exchanged across explicit messaging connections. Explicit messages are used to command the performance of a particular task and to report the results of performing the task.

DeviceNet defines an explicit messaging protocol that states the meaning of the message. An explicit message consists of a CID and associated messaging protocol information.

The rules that govern the dynamic establishment of these connections are used as a foundation upon which a predefined set of connections is defined.

4.2 DeviceNet communication model

The abstract object oriented communication model of a DeviceNet node includes the following:

- **unconnected message manager (UCMM):** processes DeviceNet unconnected explicit messages;
- **identity object:** identifies and provides general information about the device;
- **connection class:** allocates and manages internal resources associated with both I/O and explicit messaging connections;
- **connection object:** manages the communication specific aspects associated with a particular application-to-application relationship;
- **DeviceNet object:** provides the configuration and status of a physical DeviceNet CDI;
- **message router:** forwards an explicit request message to the appropriate object;
- **application objects:** implement the intended purpose of the product.

4.3 DeviceNet, CAN and CIP™

DeviceNet lower layers are based on ISO 11898-1 and ISO 11898-2 and use the Controller Area Network (CAN) technology. DeviceNet upper layers use a subset of the Common Industrial Protocol (CIP™³) protocol and services specified in the IEC 61158 series.

The relationships between DeviceNet, CAN, CIP and the OSI reference model (ISO/IEC 7498-1:1994) are shown in Figure 2.

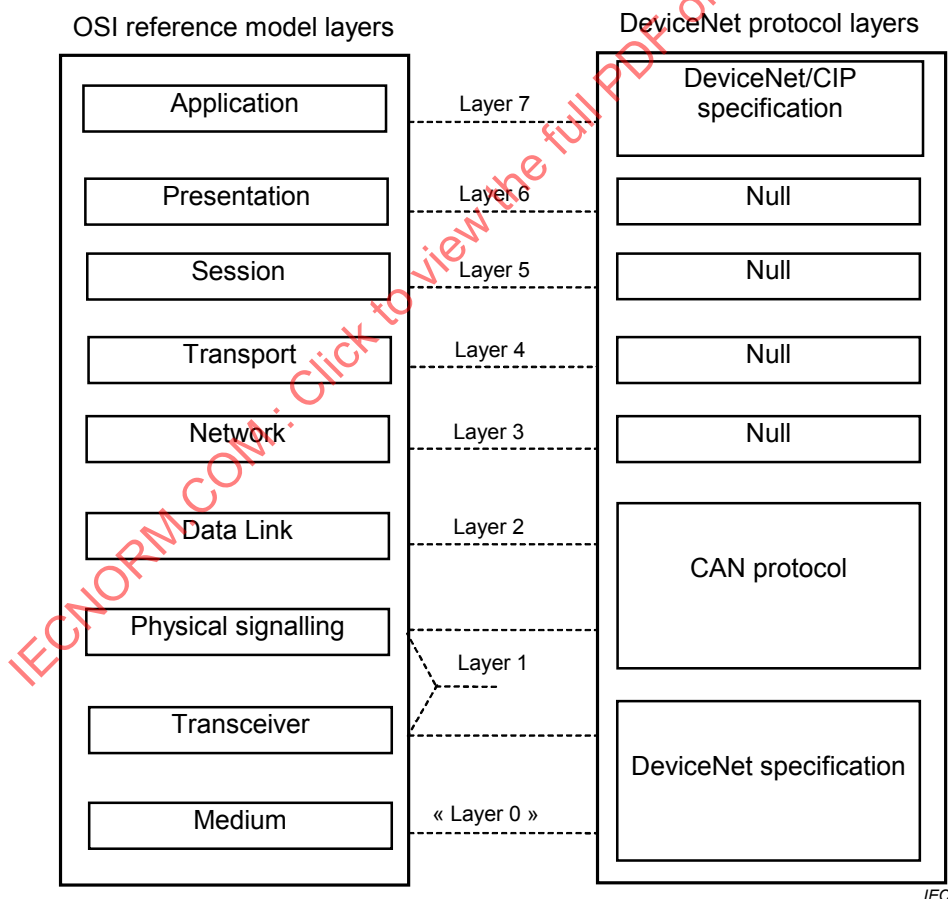


Figure 2 – DeviceNet protocol architecture compared with the OSI reference model

³ CIP™ is a trade name of ODVA, Inc. This information is given for the convenience of users of this International Standard and does not constitute an endorsement by IEC of the trademark holder or any of its products. Compliance to this profile does not require use of the trade name CIP™. Use of the trade name CIP™ requires permission of ODVA, Inc.

5 Characteristics

5.1 DeviceNet connections

5.1.1 General

DeviceNet is a connection based controller-device interface. DeviceNet connections are used to provide logical paths between multiple applications. When a connection is established, the transmissions associated with that connection are assigned a connection ID (CID). If the connection involves a bi-directional exchange, then two CID values shall be assigned to the connection.

DeviceNet uses the CAN identifier field and defines the steps involved in the dynamic establishment of I/O and explicit messaging connections.

5.1.2 DeviceNet's use of the CAN identifier field

DeviceNet uses the 11 bit CAN Identifier field ("Standard Frame Format") as described in ISO 11898-1.

The overall CAN identifiers available within DeviceNet are subdivided into four separate message groups: group 1, group 2, group 3 and group 4.

For connection based messages, the CID is placed within the CAN identifier field. Figure 3 also describes the components of a DeviceNet CID.

IDENTIFIER BITS											HEX RANGE	IDENTITY USAGE
10	9	8	7	6	5	4	3	2	1	0		
0	Group 1 message ID				Source MAC ID						0x000 – 0x3FF	Message group 1
1	0	MAC ID					Group 2 message ID				0x400 – 0x5FF	Message group 2
1	1	Group 3 message ID			Source MAC ID						0x600 – 0x7BF	Message group 3
1	1	1	1	1	Group 4 message ID (0 – 0x2F)						0x7C0 – 0x7EF	Message group 4
1	1	1	1	1	1	1	X	X	X	X	0x7F0 – 0x7FF	Invalid CAN identifiers
10	9	8	7	6	5	4	3	2	1	0		

IEC

Figure 3 – DeviceNet's use of the CAN identifier field

As shown in Figure 3 the CAN identifier field on DeviceNet contains the following:

- **message ID:** Identifies a message within a message group inside a particular node. When a connection is established, the nodes utilise a message ID in combination with a MAC ID to generate a CID. The resulting CID is specified in the CAN identifier field associated with related transmissions;
- **source MAC ID:** Groups 1 and 3 require the specification of a source MAC ID within the CAN identifier field;
- **MAC ID:** Message group 2 allows the specification of either source or destination within the MAC ID portion of the CAN identifier field.

Both explicit messaging and I/O connections may be established in message groups 1, 2 and 3.

Group 2 message ID values 6 and 7 are used as follows:

- group 2 message ID 6 is used for the configuration of the communications utilised in a master/slave application (see 5.5);
- group 2 message ID 7 is used in the detection of nodes that have been assigned identical MAC IDs (see 5.3.6).

Group 3 message ID values 5, 6 and 7 are used as follows:

- group 3 message ID 5 is used when sending responses associated with unconnected explicit messaging requests, as well as device heartbeat and device shutdown messages;
- group 3 message ID 6 is used when sending unconnected explicit messaging requests;
- group 3 message ID 7 is invalid and shall not be used.

Group 4 message ID values are used as follows:

- group 4 message IDs 0x00 through 0x2B are reserved for future use and shall not be used;
- group 4 message ID 0x2C is used for communication faulted response messages;
- group 4 message ID 0x2D is used for communication faulted request messages;
- group 4 message ID 0x2E is used for offline ownership response messages;
- group 4 message ID 0x2F is used for offline ownership request messages.

5.1.3 Connection establishment

5.1.3.1 Explicit messaging connections and the UCMM

Message group 3 defines the identifier codings to support unconnected explicit messaging. Unconnected explicit messages establish and manage explicit messaging connections. Unconnected request messages are specified by transmitting a group 3 message whose message ID component is set to 6. The only valid services that can be transmitted as unconnected explicit request messages are:

- open explicit messaging connection request;
- close connection request.

The messages listed above are never transmitted as connection-based messages (see connection-based explicit messaging in 5.2.1.6).

Responses to unconnected explicit requests are transmitted as unconnected response messages. Unconnected response messages are specified by transmitting a group 3 message whose message ID component is set to 5. The only valid services that can be transmitted as unconnected explicit response messages are:

- open explicit messaging connection response;
- close connection response;
- error response;
- device heartbeat message;
- device shutdown message.

5.1.3.2 I/O connections

I/O connections are dynamically established by interfacing with the connection object class across a previously established explicit messaging connection. A connection object instance shall be created and configured at each node.

5.2 DeviceNet messaging protocol

5.2.1 Explicit messaging

5.2.1.1 General

This subclause describes the explicit messaging protocol and presents details associated with the dynamic establishment of explicit messaging connections (see Figure 4).

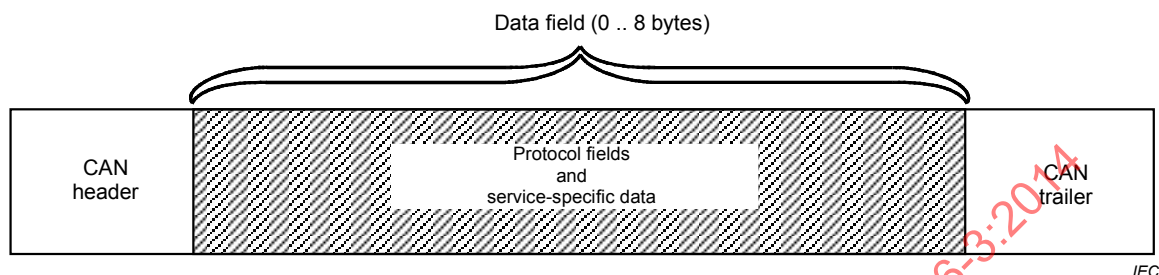


Figure 4 – Explicit message CAN data field use

Figure 5 provides the format of the CAN data field associated with explicit messages:

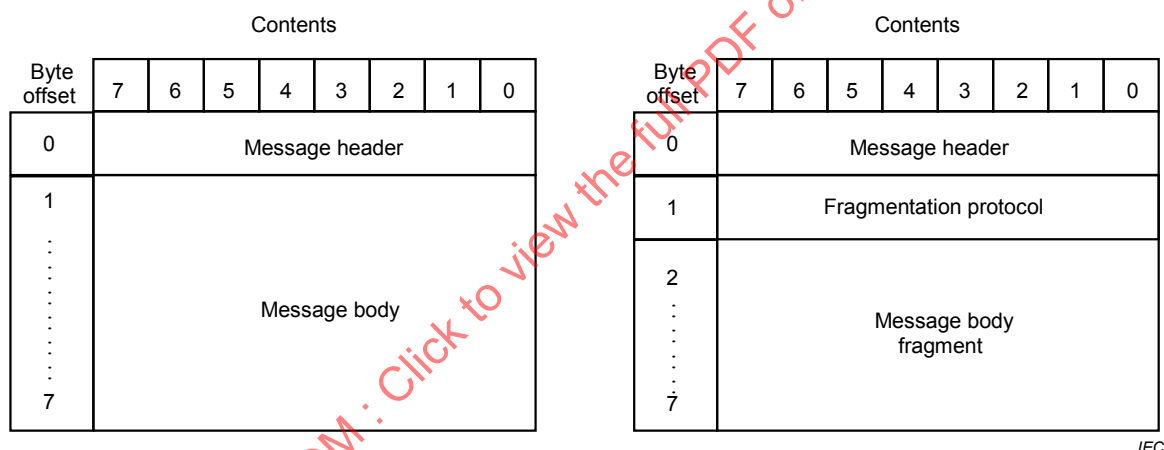


Figure 5 – Explicit message data field format

The data field of a transmission that contains the complete explicit message includes:

- a message header;
- the entire message body.

If the explicit message is greater than eight bytes in length, it shall be transmitted in a fragmented manner. The fragmentation/re-assembly function is provided by the connection object. A piece of a fragmented explicit message includes:

- a message header;
- the fragmentation protocol (see 5.2.3.2);
- a message body fragment.

5.2.1.2 Message header

The message header is located within byte offset zero in the CAN data field of an explicit message and shall be formatted as shown in Figure 6:

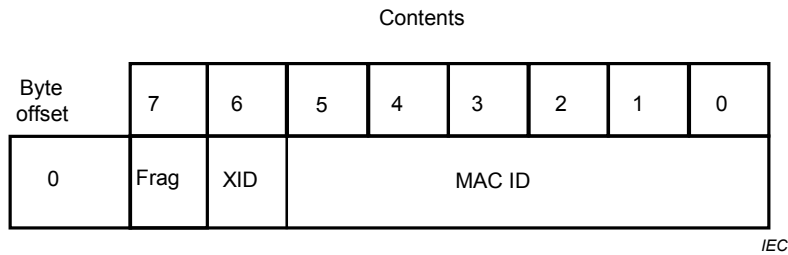


Figure 6 – Explicit message header format

Message header contents:

- **frag** (fragment bit): Indicates whether or not this transmission is a piece of a fragmented explicit message.
0 = non-fragmented
1 = fragmented
- **XID (transaction ID)**: This field is utilised by an application to match a response with the associated request. This field is simply echoed by the server in a response message. The server module does not utilise this field to perform any type of duplicate message detection logic;
- **MAC ID**: Contains either the source or destination MAC ID.

When an explicit message is received, the MAC ID field within the message header is examined. If the destination MAC ID is specified in the CID, then the source MAC ID of the other end-point shall be specified in the message header. If the source MAC ID is specified in the CID, then the receiving module's MAC ID shall be specified in the message header. If neither of these conditions are true, then the message shall be discarded.

5.2.1.3 Message body

The message body contains a service field and service specific arguments.

The first argument specified within the message body is the service field which identifies the particular request or response being delivered. Figure 7 illustrates the format of the service field.

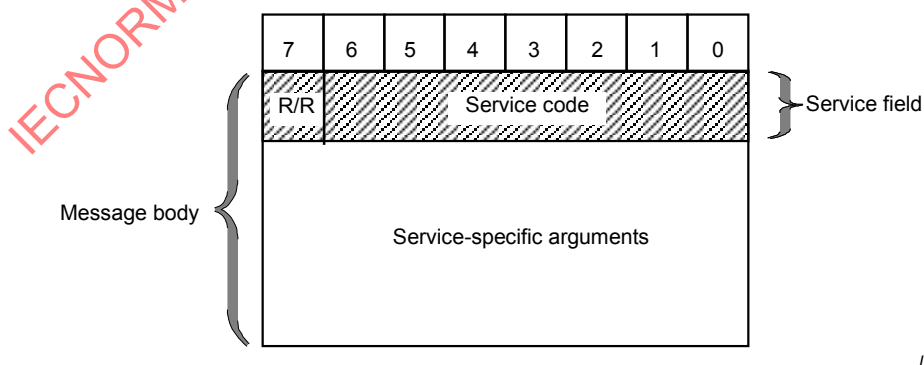


Figure 7 – Service field format

Service field contents:

- **service code**: the value specified within the least significant 7 bits of the service field byte that indicates the type of service being transmitted;

- **R/R:** the most significant bit in the service field. Its value determines whether a message is a request or a response.
 - 0 = request
 - 1 = response

DeviceNet defines a set of common services. DeviceNet common services are the open set whose parameters and required behaviours are defined in IEC 61158-5-2:2014, 6.2.1.3 and IEC 61158-6-2:2014, 4.1.8.

Information following the service field is specific to the particular type of service being transmitted.

5.2.1.4 Fragmentation protocol

If the transmission is a piece of a fragmented explicit message, then the data field shall contain the message header, the fragmentation protocol and a message body fragment. The fragmentation protocol facilitates the fragmentation and reassembly of explicit messages which have a message body greater than 8 bytes (see 5.2.3).

5.2.1.5 UCMM services

5.2.1.5.1 General

The unconnected message manager (UCMM) provides for the dynamic establishment of explicit messaging connections. This subclause presents a detailed description of the service specific arguments associated with the open explicit messaging connection and close connection services provided by the UCMM. See 9.3.3 for test specifications regarding establishment of explicit messaging connections.

The UCMM processes two services which manage the allocation and deallocation of explicit messaging connections:

- **open explicit messaging connection:** service code = 0x4B. Used to establish an explicit messaging connection;
- **close connection:** service code = 0x4C. Used to delete a connection object and deallocate all associated resources.

These services are accessed using the unconnected explicit request and response CAN identifier fields within message group 3 as specified in 5.1.2. When processing an unconnected explicit request the UCMM may need to return an error indication to the requestor and, as such, the error response explicit message may be transmitted with the unconnected explicit response CAN identifier.

5.2.1.5.2 Open explicit messaging connection request

This service requests the establishment of a logical connection between two nodes across which explicit messages will be transmitted. This service is transmitted as an unconnected request message (message group 3, message ID 6) (see Figure 8).

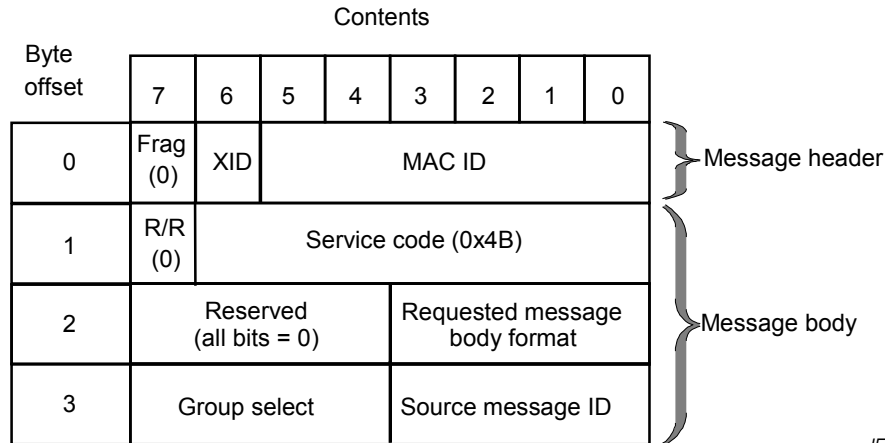


Figure 8 – Open explicit messaging connection request format

Open explicit messaging connection request contents:

- **frag (0)/transaction ID/MAC ID:** See 5.2.1.2. The destination MAC ID is always specified in the message header associated with an open explicit messaging connection request/response;
- **R/R bit (0):** Indicates this is a request message;
- **service code (0x4B):** Identifies this message as an open explicit messaging connection service;
- **reserved bits:** These bits shall be ignored by the receiver and shall be set to zero by the transmitter;
- **requested message body format:** The field used by the client to request a particular message body format for subsequent explicit messages transmitted over this connection. A server shall support at least one of the DeviceNet message body formats (values 0 – 3).
The server node responding to this open explicit messaging request defines the actual message body format to be used over this connection. See Table 1 for message body format values. Servers shall do one of the following:
 - refuse the request and return a supported format value in the open explicit messaging connection response; the returned format shall be one of the DeviceNet message body formats (values 0 – 3);
 - accept this request by echoing the same format value in the open explicit messaging connection response;

Table 1 – Message body format values

Value	Meaning
0	DeviceNet (8/8). Class ID = 8 bit integer, Instance ID = 8 bit integer
1	DeviceNet (8/16). Class ID = 8 bit integer, Instance ID = 16 bit integer
2	DeviceNet (16/16). Class ID = 16 bit integer, Instance ID = 16 bit integer
3	DeviceNet (16/8). Class ID = 16 bit integer, Instance ID = 8 bit integer
4	CIP path. The addressing size is variable and is provided as a packed EPATH on each request ^a
5 – 0xF	Reserved
NOTE Messages transmitted across this connection are formatted as described in 5.2.1.6.	
^a The CIP Path is specified as an 8-bit path length (USINT) followed by the path (packed EPATH). See IEC 61158-6-2:2014, 4.1.9 for information on EPATH. The server shall support, as a minimum, the most efficient encoding for any logical segment within the EPATH (for example 8-bit class encoding for class code 0x01).	

- **group select:** The field that indicates the message group across which messages associated with this connection are to be exchanged. The following values are shown in Table 2:

Table 2 – Group select values

Value	Meaning
0	Message group 1
1	Message group 2 ^a
2	Reserved
3	Message group 3
4 – 0xE	Reserved
0xF	Reserved for node ping ^b
^a The message group 2 identifier allows for specification of either the source or destination MAC ID. For explicit messaging connections established across message group 2, the client places the MAC ID of the server in the connection ID when transmitting messages across this connection. The server places its own MAC ID in the connection ID when transmitting messages across this connection. This process requires the server to allocate two separate message IDs from its group 2 pool.	
^b This group select value can be used to force a UCMM capable target node to respond without creating any resources within the node. When a request with this value is received, the target shall respond with a general error code of "invalid parameter" (0x20) and an additional error code of 0x01.	

The client selects the message group across which the transmissions associated with this explicit messaging connection will take place. If the server cannot satisfy the request, then it shall reject the request and return an error response.

- **source message ID:** The use of this field depends on the value within the group select field (see Table 3):

Table 3 – Source message ID in open explicit messaging connection request

If group select equals:	Then the source message ID:
0 or 3	Specifies the message ID the client has allocated from its group 1 or 3 message ID pool. The client shall use this message ID in combination with its own MAC ID to generate the connection ID specified when it transmits a message across this connection ^a
1	Is ignored/set to the value zero (0) ^b
^a The client places this value within the message ID component of the message group 1 or 3 Identifier. ^b Explicit messaging connections established across message group 2 require the server to allocate two group 2 message IDs and return them in the open explicit messaging connection response message. The client shall use one of these message IDs to generate the connection ID it specifies when transmitting a message across this connection. The other shall be used by the server to generate the connection ID it specifies when transmitting a message across this connection.	

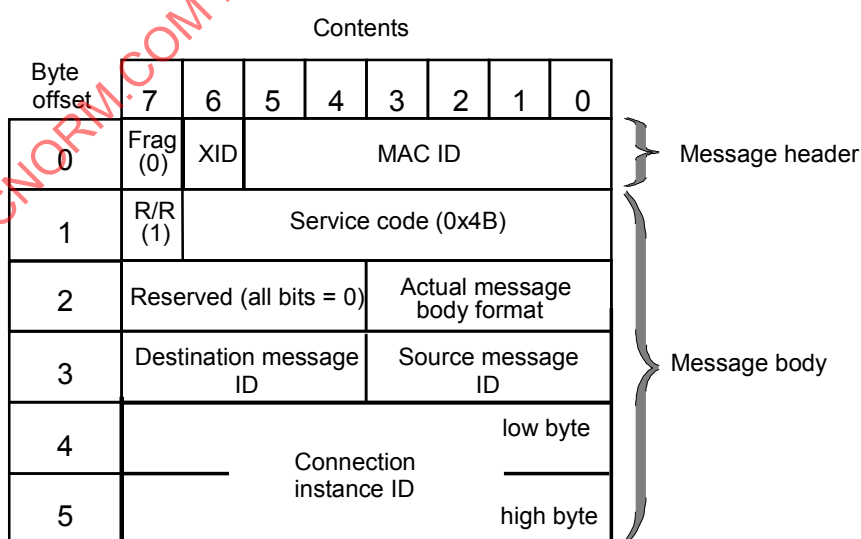
The UCMM within the server validates the open explicit messaging connection request arguments. If they are valid, the UCMM invokes the create service of the connection class to obtain a connection object instance (see IEC 61158-5-2:2014, 6.2.3). The resulting connection object is automatically configured to be an explicit messaging connection object.

If the server supports multiple message body formats and the client has requested one of those formats, then the server shall echo the requested message body format in the open explicit response message. If the server does not support multiple message body formats, then the server shall specify its default format in the open explicit messaging connection response.

If no errors are detected, then an open explicit messaging connection success response shall be returned. If an error is detected, then an error response message shall be returned.

5.2.1.5.3 Open explicit messaging connection success response

This service is used to respond to a successful open explicit messaging connection request message (see Figure 9).



IEC

Figure 9 – Open explicit messaging connection response format

Open explicit messaging connection response contents:

- **frag (0)/transaction ID/MAC ID:** See 5.2.1.2. The destination MAC ID is always specified in the message header associated with an open explicit messaging connection request/response;
- **R/R bit (1):** Indicates that this is a response message;
- **service code (0x4B):** Identifies this message as an open explicit messaging connection service;
- **reserved bits:** These shall be set to zero by the transmitter;
- **actual message body format:** The field used by the server to define the format of the message body associated with subsequent explicit messages transmitted over this connection (as described in Table 1);
- **destination message ID:** The use of this field depends on the message group across which the connection takes place (as described in Table 4);

Table 4 – Destination message ID in open explicit messaging connection response

If group select within the open request was set to:	Then the destination message ID in the open response:
0 or 3	Is ignored and shall be set to the value zero (0)
1	Is used by the client in combination with the server's MAC ID to generate the connection ID it specifies when transmitting across this connection

- **source message ID:** The message ID value that the server has allocated. The server allocates a message ID from its group 1, 2, or 3 message ID pool that is used in conjunction with its own MAC ID (source MAC ID) to generate the connection ID that is specified when it transmits a message across this connection;
- **connection instance ID:** The server creates an explicit messaging connection object when it successfully services an open request. This field holds the instance ID value (16 bit integer field) assigned to that explicit messaging connection object.

5.2.1.5.4 Close connection request

This service is used to terminate a connection (either I/O or messaging) within one of the end points. The reception of the close connection request message by the UCMM results in the invocation of the connection class's delete service (see IEC 61158-5-2:2014, 6.2.3). A close connection request message is transmitted as an unconnected request message.

The responder verifies that the specified connection instance exists. If the connection instance exists, and it can be deleted, then it shall be deleted. All resources associated with the connection instance are freed. If the request is successfully serviced, then a close connection response shall be returned. If the request is not successful, an error response shall be returned.

Close connection request contents (see Figure 10):

- **frag (0)/transaction ID/MAC ID:** See 5.2.1.2;
- **R/R bit (0):** Indicates that this is a request message;
- **service code (0x4C):** Identifies that this is a close connection service;
- **connection instance ID:** The field that specifies the connection instance to be deleted. The format for the connection instance ID within this message is always specified as a 16 bit integer.

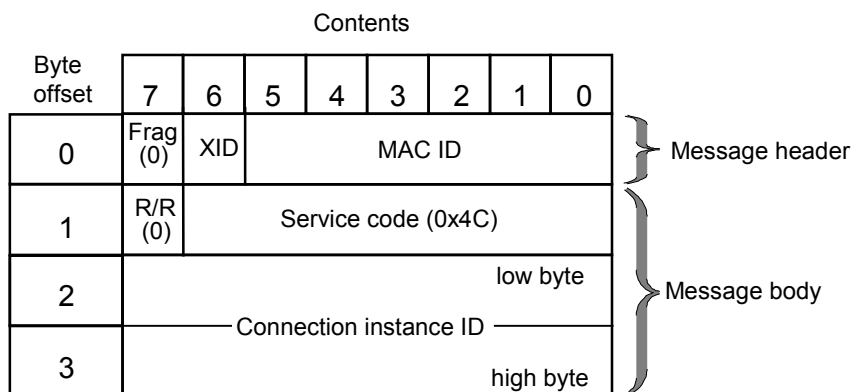


Figure 10 – Close connection request format

5.2.1.5.5 Close connection response

This service is used to respond to a successful close connection request message.

Close connection response format contents (see Figure 11):

- **frag (0)/transaction ID/MAC ID:** See 5.2.1.2;
- **R/R bit (1):** Indicates that this is a response message;
- **service code (0x4C):** Identifies this message as a close connection service.

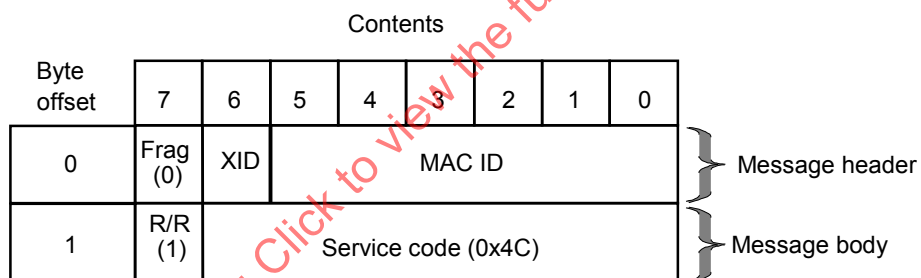


Figure 11 – Close connection response format

5.2.1.5.6 Error response

Table 5 shows a standard set of UCMM related error conditions and the error code (both general error code and additional error code) information to be used in an associated error response message. The format of an error response message is described in 5.2.1.6.5.

Table 5 – UCMM error conditions/codes

Error condition	General error name	General error code	Additional error code
Service code not “open” or “close”	Service not supported	0x08	0xFF
Group select resource error	Resource unavailable	0x02	0x01
Group select out of range	Invalid parameter	0x20	0x01
No server connections available	Resource unavailable	0x02	0x02
No server message IDs available	Resource unavailable	0x02	0x03
Client source message ID invalid	Invalid parameter	0x20	0x02
Duplicate client source message ID	Resource unavailable	0x02	0x04
Connection instance ID invalid	Object does not exist	0x16	0xFF

Error condition descriptions:

- **service not “open” or “close”:** A service received across the UCMM port is not of type “open” or “close” and is therefore not supported by UCMM;
- **group select resource error:** The group select argument indicates utilising a message group which is not supported by the device;
- **group select out of range:** The group select field contains an invalid value;
- **no server connections available:** The maximum number of connections supported by this server has already been reached;
- **no server message IDs available:** The server has allocated all available message IDs within the message group requested by the client;
- **client source message ID invalid:** The source message ID received with an open explicit message connection request is invalid for the specified message group;
- **duplicate client source message ID:** The source message ID and source MAC ID received within an open explicit messaging connection request are already in use for a group 1 or group 3 explicit messaging connection;
- **connection instance ID invalid:** The connection instance ID received with the close connection request does not exist.

5.2.1.6 Connection based explicit messaging

5.2.1.6.1 General

A connection based explicit message is a message transmitted over an explicit messaging connection. A connection based explicit message shall conform to the formats described in this subclause.

5.2.1.6.2 Explicit request message contents (message body format values 0 – 3)

Figure 12 shows the format for the message associated with a non-fragmented explicit request for message body format values 0 – 3:

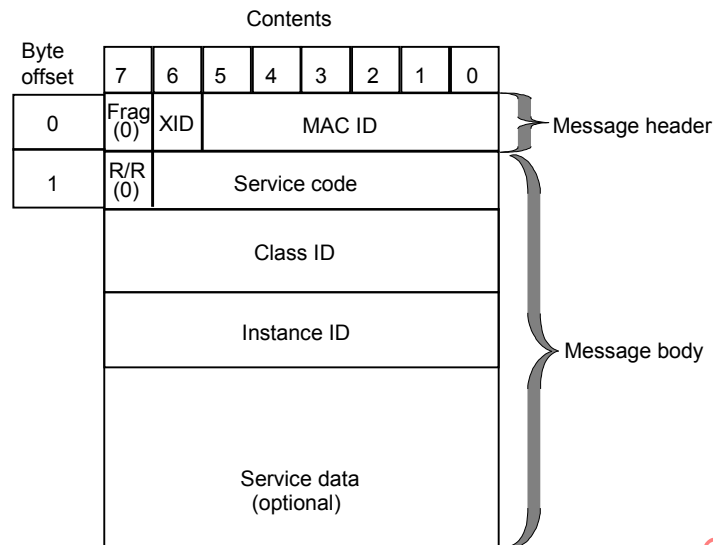


Figure 12 – Non-fragmented explicit request message format, values 0 – 3

Non-fragmented explicit request message contents:

- **frag (0)/transaction ID/MAC ID:** See 5.2.1.2;
- **R/R bit (0):** Indicates that this is a request message;
- **service code:** Defines the service being requested;
- **class ID:** Defines the object class towards which this request is directed. The class ID is specified within either an 8 bit or 16 bit integer field based on the actual message body format value returned in the open explicit messaging connection response;
- **instance ID:** Defines the particular instance within the object class towards which this request is directed. The instance ID is specified within either an 8 bit or 16 bit integer field based on the actual message body format value returned in the open explicit messaging connection response. DeviceNet uses the value zero to denote that the request is directed towards the class itself rather than a specific instance within the class;
- **service data:** Carries request specific data. Formats for the DeviceNet common services are described in IEC 61158-6-2:2014, 4.1.8. Class-specific and object-specific service definitions include the format of this field.

5.2.1.6.3 Explicit request message contents (message body format value 4)

Figure 13 shows the format for the message associated with a non-fragmented explicit request for message body format value 4:

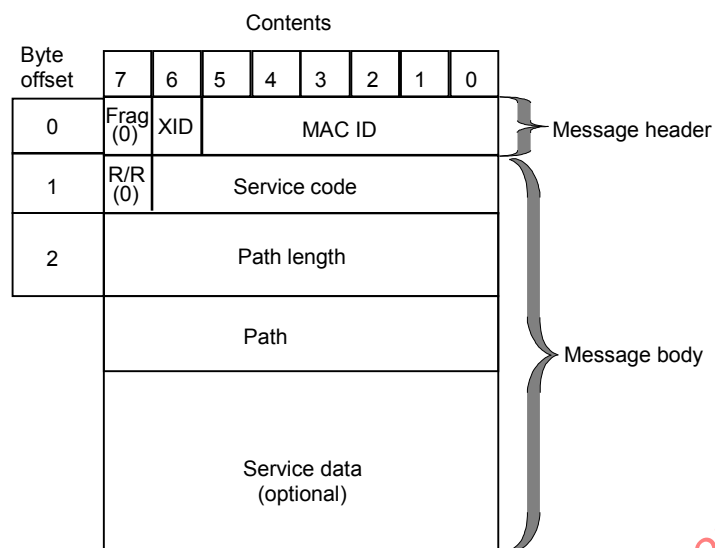


Figure 13 – Non-fragmented explicit request message format, value 4

Non-fragmented explicit request message contents:

- **frag (0)/transaction ID/MAC ID:** See 5.2.1.2;
- **R/R bit (0):** Indicates that this is a request message;
- **service code:** Defines the service being requested;
- **path length:** This 8-bit integer value (USINT) provides the length of the message request path in bytes;
- **path:** Defines the path of the message request (packed EPATH);
- **service data:** Carries request specific data. Formats for the DeviceNet common services are described in IEC 61158-6-2:2014, 4.1.8. Class-specific and object-specific service definitions include the format of this field.

5.2.1.6.4 Success response explicit message

Figure 14 shows the format for the message associated with a non-fragmented success response:

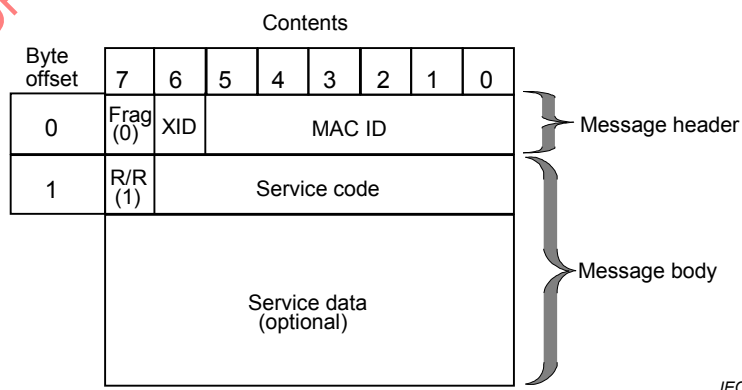


Figure 14 – Non-fragmented success response message format

Non-fragmented success response message contents:

- **frag (0)/transaction ID/MAC ID:** See 5.2.1.2;

- **R/R bit (1):** Indicates that this is a response message;
- **service code:** Contains the service code sent in the request message;
- **service data:** Carries request-specific data.

5.2.1.6.5 Error response message

The error response message is returned when an error is encountered while attempting to service a previously received explicit request message. The error response may be sent as either a connection based (request was received across an explicit messaging connection) or unconnected (request was an unconnected explicit request message) response message. Figure 15 shows the format of an error response message.

		Contents							
Byte offset	7	6	5	4	3	2	1	0	
0	Frag (0)	XID	MAC ID						
1	R/R (1)	Service code (0x14)							
2	General error code								
3	Additional code								

IEC

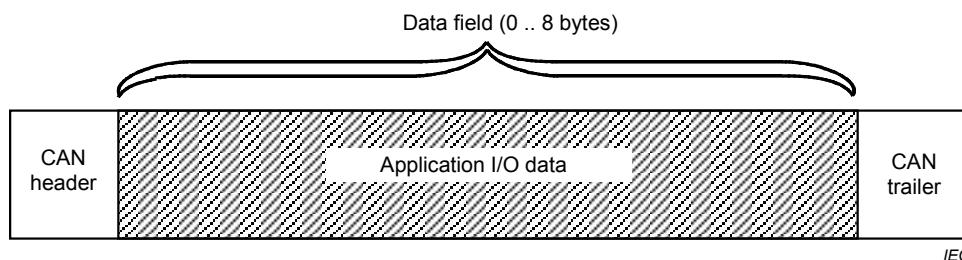
Figure 15 – Error response message

Error response message contents:

- **frag (0)/transaction ID/MAC ID:** See 5.2.1.2;
- **R/R bit (1):** Indicates that this is a response message;
- **service code (0x14):** Identifies this message as an error response;
- **general error code:** Identifies the encountered error. See IEC 61158-5-2:2014, 6.2.1.3.3 and IEC 61158-6-2:2014, 4.1.11 for a list of general error codes;
- **additional code:** Contains an object-specific or service-specific value that further describes the error condition. If the responding object has no additional information to specify, then the value 0xFF shall be placed within this field.

5.2.2 Input/output messaging

DeviceNet defines a fragmentation protocol for transfer of an I/O message greater than eight bytes in length. This is the only protocol information carried within the data field of an I/O message (see Figure 16).



IEC

Figure 16 – Data field of an I/O message

5.2.3 Fragmentation/reassembly

5.2.3.1 General

This subclause defines the means by which a message whose length is greater than eight bytes is fragmented and reassembled. The fragmentation/reassembly function is provided by the DeviceNet connection object.

The logic that triggers a fragmented transmission is:

- explicit messaging connection object instances examine the length of each message to be transmitted. If the message is greater than eight bytes in length, then the fragmentation protocol shall be used;
- I/O connection object instances examine the `produced_connection_size` attribute of the connection object (see 5.3.2). If the `produced_connection_size` attribute is greater than eight, then the fragmentation protocol shall be used.

Two types of fragmentation are defined:

- **acknowledged:** performed when fragmenting an explicit message;
- **unacknowledged:** performed when fragmenting an I/O message.

5.2.3.2 Fragmentation protocol

The fragmentation protocol is located within a single byte in the CAN data field, and is formatted as shown in Figure 17:

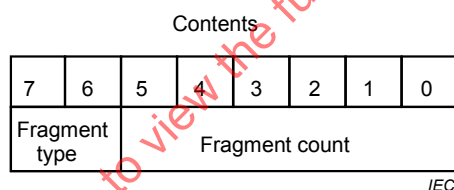


Figure 17 – Format of DeviceNet fragmentation protocol

Fragmentation protocol contents:

- **fragment type:** Indicates whether this is the first fragment, one of the middle fragments, or the last fragment (see Table 6).

Table 6 – Fragment type bit values

Value	Meaning
0	First fragment. The fragment count field shall contain the value 0 or 0x3F ^a
1	Middle fragment ^b
2	Last fragment ^c
3	Fragment acknowledge ^d
^a If the fragment count contains the value zero (0), then this is the first in a series of fragments. If the fragment count field contains the value 0x3F, then this is also the last transmission in the series. ^b This fragment is neither the first nor the last fragment in the series. ^c Marks this as the last fragment. ^d The value the receiver of a fragmented message uses to acknowledge the reception of a fragment.	

- **fragment count:** Marks each separate fragment such that the receiver may determine whether or not a fragment has been missed. If the fragment type is first fragment, then this field takes on a special meaning (as described in Table 6). The fragment count is increased by one for each successive fragment in a series and resets to zero at overflow (fragment count = (fragment count + 1) mod 64).

For I/O message fragmentation, the fragmentation protocol information is placed within byte offset 0 (see Figure 18).

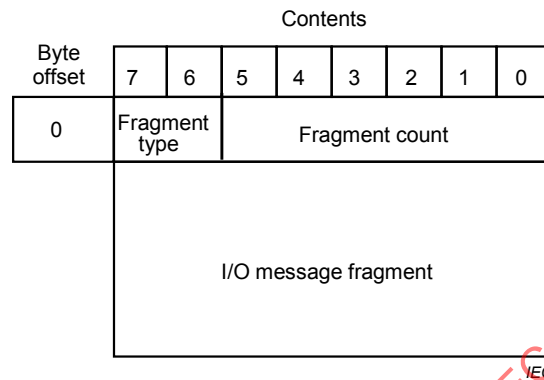


Figure 18 – I/O message fragment format

For explicit message fragmentation, the fragmentation protocol information is placed within byte offset 1 (see Figure 19).

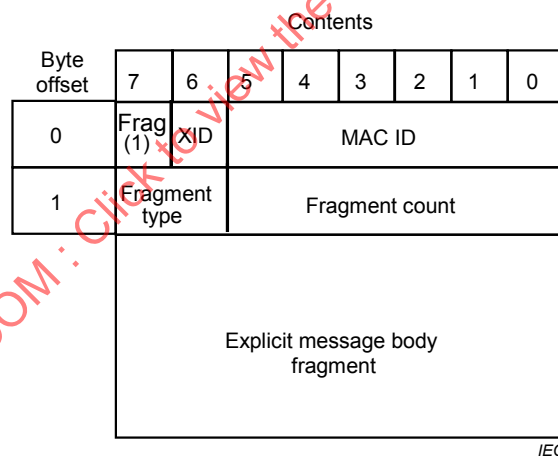


Figure 19 – Explicit message fragment format

The Frag bit within the message header in Figure 19 is set to 1 to indicate that this is a piece of the explicit message, not the entire message. The value 1 here also indicates that the next byte contains the fragmentation protocol.

The receiver of a fragmented series of transmissions parses the fragmented message as defined in this subclause. This procedure applies to both I/O and explicit message fragmentation.

If the first transmission is expected by the connection and the fragment type is equal to first fragment:

- if the fragment count is 0x3F, then this is the only transmission in the series, and the connection processes the message and awaits the beginning of a new series;

- if the fragment count is 0, then this is the first in a series of transmissions and the connection stores the fragment and saves the fragment count.

If the first transmission is expected by the connection and either the fragment type is not equal to first fragment or the fragment count is not equal to 0 or 0x3F, the connection discards the transmission and awaits the beginning of a new message.

If the first transmission is not expected by the connection, then the connection verifies:

- that the fragment type is not first fragment; and
- that the fragment count is numerically one (1) greater than the previous value received.

If one of these checks fails, then an error has been detected. If both checks pass, then the fragment is appended to the previously received fragment(s), and the fragment type is parsed to determine whether or not more fragments are to be expected.

If more fragments are forthcoming, then the connection saves the received fragment count and waits for the next fragment. If this is the last transmission in the series and an error has not been detected, then the connection processes the message and resets to looking for the beginning of a new series.

If an explicit message is being fragmented, then the receiver shall generate and transmit an acknowledgement after the reception of each fragment.

If an error is detected, then error recovery specific to whether this is a fragmented I/O or explicit message takes place.

If the detection of a missed fragment was triggered by the reception of the first fragment in the next series, then any processing associated with the current series is immediately discontinued, the fragments stored in memory are discarded, and processing immediately begins on the new series.

5.2.3.3 Unacknowledged fragmentation

Fragmentation of an I/O message is performed in an unacknowledged fashion. Unacknowledged fragmentation consists of the back-to-back transmission of the fragments from the transmitting module. The receiving module(s) returns no acknowledgements.

When an I/O connection's send_message service is invoked, it examines its connection_size attribute to determine whether or not a fragmented series of messages is to be transmitted. If the connection_size attribute is greater than eight (8), then the fragmentation protocol is placed within the I/O.

If the connection_size attribute is less than or equal to eight (8) bytes, then the raw data is transmitted without the presence of the fragmentation protocol.

If the application requests to transmit a piece of data whose length is greater than the connection_size attribute, then an internal error is indicated and the transmission does not occur. If the receiving I/O connection object detects a missed fragment by determining that the received fragment count is not equal to the previously received fragment count plus one (1), then the following error recovery steps are taken:

- all subsequent fragments in this series are dropped and the application is not informed of an I/O message reception;
- the connection object begins looking for the beginning of a new fragmented series of transmissions and discards the remaining fragments in this series.

5.2.3.4 Acknowledged fragmentation

Acknowledged fragmentation is used for fragmented explicit messages. Acknowledged fragmentation consists of the transmission of a fragment from the transmitting node followed by the transmission of an acknowledgement by the receiving node. The receiving node acknowledges the reception of each fragment.

Figure 20 shows the format for the acknowledgement message transmitted by the receiver after each explicit message fragment is received.

Byte offset	Contents							
	7	6	5	4	3	2	1	0
0	Frag (1)	XID	MAC ID					
1	Fragment type (3)		Fragment count					
2	Ack status							

Figure 20 – Acknowledgement message format

Acknowledgement message contents:

- **fragment type:** Indicates that this is a fragment acknowledge by placing the value 3 within this field;
- **fragment count:** Echoes the last fragment count value received;
- **ack status:** Indicates whether or not an error has been detected by the receiver of the fragmented message. The ack status bit values are defined in Table 7:

Table 7 – Ack status bit values

Value	Meaning
0	Success. No errors have been detected and the fragmented transmission shall continue
1	Too much data. The maximum amount of data the receiver can receive across this connection has been exceeded
2 – 255	Reserved

The transmitting node functions as specified below:

- the connection object formulates the message header by setting the Frag bit to one (1), the XID (transaction ID) field to the value specified by the application, and initializes the MAC ID field in the same manner it would a non-fragmented explicit message. The message headers associated with each separate fragment are identical;
- the connection object then places the appropriate fragmentation protocol information into the message. The connection object stores the fragment count that was inserted into the message;
- the connection object then takes the next piece of the message body and places it into the message;
- the message is transmitted and a wait for acknowledgement timer is started. The amount of time to wait is application specific;
- if the wait for acknowledgement timer expires, then the connection object automatically retries the last transmission. The connection object retries once. If the timer expires once again (two consecutive wait for acknowledgement time-outs), then the application is

informed that an error has been detected and the requested transmission cannot take place;

- if an acknowledgement is received, then the connection object determines whether or not the fragment count in the acknowledgement is equal to the last fragment count it transmitted. If they are equal, then the fragment was successfully delivered and acknowledged, and normal processing continues. If the values are not equal, then the connection object continues to wait for an acknowledgement with a matching fragment count.

The initial state associated with the receiving module entails waiting for either the first fragment in a fragmented transmission or waiting for the reception of a complete explicit message. The receiving side functions as specified below:

- If the message header indicates that this is a fragmented portion of an explicit message, then the connection object examines the fragmentation protocol to determine its validity. If the connection has yet to receive the first transmission in the series (in the initial state) and the fragment type field is not equal to the first fragment, then the fragment is dropped and no acknowledgement is returned;
- If the fragment type indicates that this is the first fragment, then the fragment count shall equal zero (0) or 0x3F. If the fragment count is 0x3F, then this is the only transmission in the series and the connection does the following:
 - processes the message;
 - resets to looking for the beginning of a new series

If the fragment count is zero (0), then this is the first in a series of transmissions and the connection stores the fragment and saves the fragment count. If the fragment type indicates first fragment and the fragment count is not zero (0) or 0x3F, then the fragment is dropped and no acknowledgement is returned;

- If the fragment count is one (1) greater than the previously received count and the fragment type does not indicate that this is the first fragment, then the next fragment has been received. The fragment is appended to the previously received fragment(s) and an acknowledgement is returned. The fragment count associated with this fragment is stored;
- If the fragment count is neither one (1) greater nor equal to the previously received count, then the fragment is discarded and no acknowledgement is returned. The receiver resets to the initial state;
- When the final fragment is received and the acknowledgement is transmitted, the connection object continues processing the message as if it were non-fragmented.

5.2.4 Offline connection set

5.2.4.1 General

This subclause describes the offline connection set messaging protocol and presents details associated with the establishment of offline connection set ownership. Support of the offline connection set is optional for all types of devices. The group 4 offline connection set messages are used by client tool(s) to recover nodes in the communication faulted state. Using the offline connection set messages, a client (tool) shall be able to:

- a) visually identify the faulted node(s) to which it is communicating with by flashing an LED;
- b) send fault recovery messages to the faulted node; and when possible
- c) recover the faulted node without having to unplug it from the subnet.

At any point in time, only one (1) client node shall communicate with nodes in the communication faulted state, connected to a single subnet. Ownership of communication faulted nodes is gained via a dialog between clients (tools) using the offline ownership request/response messages.

Table 8 shows the group 4 identifiers associated with the offline connection set.

Table 8 – Offline connection set

Identifier bits (Connection ID)											Identity usage	
Group ID					Message ID							
10	9	8	7	6	5	4	3	2	1	0	Group 4 messages	
1	1	1	1	1	0x2C							Communication faulted response message
1	1	1	1	1	0x2D							Communication faulted request message
1	1	1	1	1	0x2E							Offline ownership response message
1	1	1	1	1	0x2F							Offline ownership request message

Only clients wishing to support the offline connection set shall produce messages using group 4 message ID 0x2F and consume response messages with group 4 message ID 0x2E. Once ownership is gained, the client shall produce all messages destined for communication faulted node(s) using group 4 message ID 0x2D.

A client may NOT produce a “Communication faulted request message” until it has gained ownership of the “Offline connection set”.

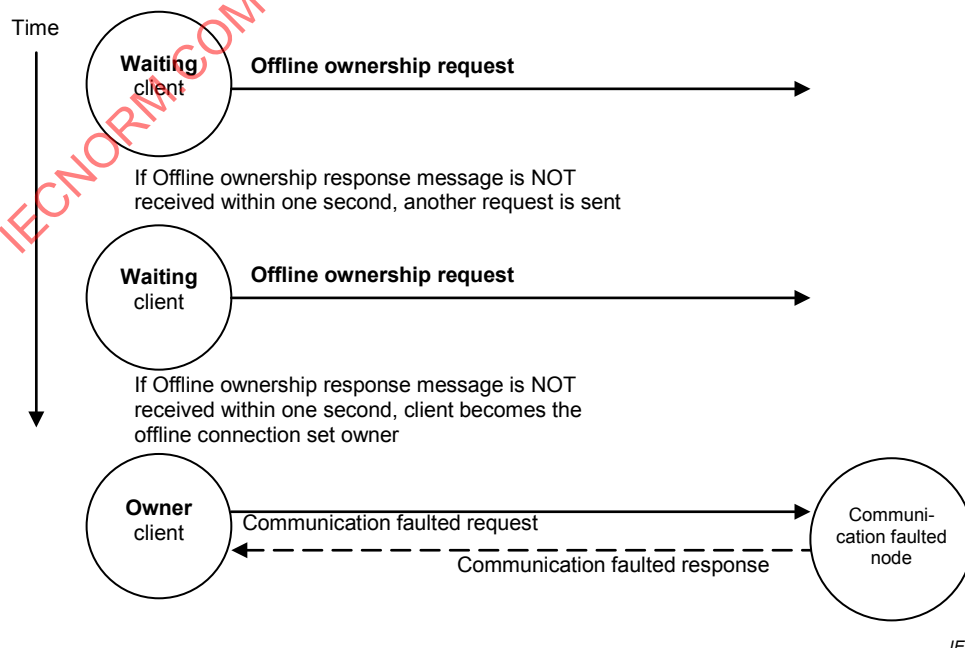
Once a client (tool) establishes offline connection set ownership, it is able to transmit communication faulted request messages with group 4 message ID 0x2D and receive communication faulted response messages with group 4 message ID 0x2C.

While in the communication faulted state, a node supporting this feature is only required to consume a single connection ID; group 4 message ID 0x2D. A faulted node shall produce its communication fault response messages on group 4 message ID 0x2C.

Offline connection set messages are low priority and may be subject to delays due to other network traffic.

5.2.4.2 Offline ownership

Figure 21 and Figure 22 illustrate the steps involved for a client (tool) to gain ownership of the offline connection set.



IEC

Figure 21 – Establishing the offline ownership

To gain control of the offline connection set a client (tool) shall produce its *offline ownership request message*. Upon successful transmission, the client (tool) shall wait for an *offline ownership response message* for a period of 1 s. If a response is not received, it shall produce a second *offline ownership request message* and shall wait another second. If a response is not received, it shall become the owner of the offline request message. If an *offline ownership response message* is received during either wait period, it shall not become the owner of the offline connection set and shall wait to become the owner.

Only one client tool shall have ownership of the offline connection set at any point in time.

The scenario where one client (tool) has previously claimed ownership of the offline connection set, and additional client nodes are arbitrating for ownership, the current owner shall reply to any *offline ownership request messages* with an *offline ownership response message*.

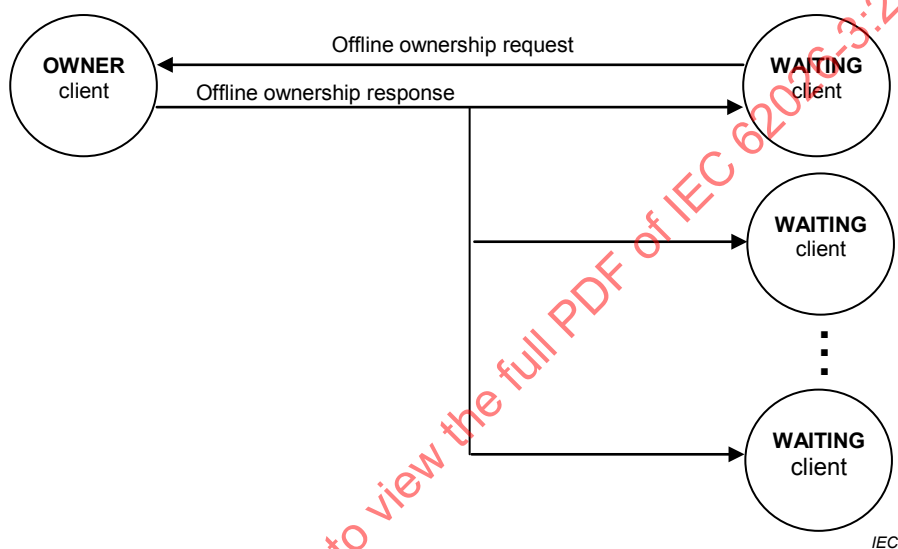


Figure 22 – Multicast nature of the offline ownership

A waiting client shall not attempt to send an offline ownership request message for a minimum of two seconds after receiving an offline ownership response message.

5.2.4.3 Offline ownership messages

5.2.4.3.1 General

The two messages used to manage the offline connection set are only produced and consumed by client nodes supporting this functionality. These messages are only serviced while a client is participating in a recovery activity, otherwise they are ignored by a client.

5.2.4.3.2 Offline ownership request message (client only)

To gain ownership of the offline connection set, a client tool shall produce an *offline ownership request message* with group 4 message ID 0x2F, see Figure 23.

Byte Offset	Contents							
	7	6	5	4	3	2	1	0
0	Reserved [0]		Client MAC ID					
1	R/R [0]	Allocate [0x4B]						
2	Vendor ID						Low byte	
3							High byte	
4	Serial number						Low byte	
5								
6								
7							High byte	

IEC

Figure 23 – Offline ownership request message

Upon successful transmission, the client (tool) shall wait for an *offline ownership response message* for at least 1 s. If a response is not received, it shall produce a second *offline ownership request message* and shall wait at least 1 s. If a response is not received, it shall become the owner of the offline connection set. If an *offline ownership response message* is received during either time period, it shall not become the owner of the offline connection set and shall wait to become the owner.

Once a client is the owner of the offline connection set, if it receives an *offline ownership request message*, it shall produce an *offline ownership response message* within 1 s.

If a client is waiting to become the owner of the offline connection set, it shall not produce an *offline ownership request message* at a rate faster than once every 2 s. This 2 s delay shall be reset upon the receipt of any message using an *offline ownership request or response connection ID*.

5.2.4.3.3 Offline ownership response message (client only)

The format of the *offline ownership response message* is the same as the *offline ownership request message*, except it is produced at group 4 message ID 0x2E, see Figure 24, and the R/R bit is set (1).

Byte offset	Contents							
	7	6	5	4	3	2	1	0
0	Reserved [0]		Client MAC ID					
1	R/R [1]	Allocate [0x4B]						
2	Vendor ID						Low byte	
3							High byte	
4	Serial number						Low byte	
5								
6								
7							High byte	

IEC

Figure 24 – Offline ownership response message protocol

Once a client (tool) is in the owner of the offline connection set, it may produce *communication faulted request messages* to all faulted nodes. Any node supporting the offline connection set shall service the *communication faulted request message* only while in the communication faulted state.

5.2.4.4 Communication faulted messages

5.2.4.4.1 General

All nodes in the communication faulted state, which support the faulted node recovery mechanism, shall receive the *communication faulted request messages* produced at group 4 message ID 0x2D. When appropriate, communication faulted nodes shall produce a *communication faulted response message* at group 4 message ID 0x2C.

Communication faulted request messages are consumed by all faulted nodes supporting this functionality. Depending upon the request message, any number of the faulted nodes may reply to a single request.

5.2.4.4.2 Communication faulted message protocols

To support the *communication faulted message* protocols, a node shall support the “Network Status” LED or “Combined Module/Network Status” LED, or have a suitable device-specific method of indicating the network and module status externally (such as a display screen or a text readout). It shall also have a network settable MAC ID.

The protocol of all the *communication faulted request messages* fall into two forms, multicast protocol (see Figure 25) and point-to-point protocol (see Figure 26).

Byte offset	Contents							
	7	6	5	4	3	2	1	0
0	Reserved [0]	Match	Value					
1	R/R [0]	Service						

IEC

Figure 25 – Communication faulted request message – Multicast protocol

The multicast protocol is used by a client to perform requests to all communication faulted node(s) to acquire their serial number(s) and vendor ID(s). Once this information is known by the client, a point-to-point protocol may be used.

When a communication faulted node receives a multicast *communication faulted request message*, it shall perform the following checks to determine whether the message should be accepted and serviced:

- Multicast match protocol:
 - IF match bit is set (1);
 - AND value equals its MAC ID;
 - AND the length of the message is two (2) bytes;
 - THEN accept and service the *communication faulted request message*.
- Multicast mask protocol:
 - IF match bit is reset (0);
 - AND value (logically AND'd with its MAC ID) equals its MAC ID;
 - AND the length of the message is two (2) bytes;

THEN accept and service the *communication faulted request message*.

The multicast mask protocol (logical AND) is used for dialogs where a range of MAC IDs are requested to respond. Table 9 indicates some, but not all, of the 64 possible mask values.

Table 9 – Addresses reporting based upon mask

5	4	3	2	1	0	Faulted nodes with
1	1	1	1	1	1	MAC ID < 64
0	1	1	1	1	1	MAC ID < 32
0	0	1	1	1	1	MAC ID < 16
0	0	0	1	1	1	MAC ID < 8
0	0	0	0	1	1	MAC ID < 4
0	0	0	0	0	1	MAC ID < 2

The multicast mask protocol is used to reduce the number of messages, which shall be sent on a subnet to determine if, and at what MAC ID, a communication faulted node resides.

The point-to-point protocol follows.

Byte offset	Contents							
	7	6	5	4	3	2	1	0
0	Reserved (service determines usage)							
1	R/R [0]	Service						
2	Vendor ID						Low byte	
3							High byte	
4	Serial number						Low byte	
5								
6								
7							High byte	

IEC

Figure 26 – Communication faulted request message – Point-to-point protocol

When a communication faulted node receives a point-to-point *communication faulted request message*, it shall perform the following checks to determine whether the message should be accepted and serviced:

Point-to-point match protocol:

- IF the length of the message is equal to eight (8) bytes;
- AND the serial number and vendor ID match its own;
- THEN accept and service the *communication faulted request message*.

The following services are currently defined for the *communication faulted request messages*:

Identify communication faulted request message – Multicast protocol:

Used when the client is attempting to identify the existence of any communication faulted nodes, which support the offline connection set feature, on a subnet.

Identify communication faulted request message – Point-to-point protocol:

Used when the client detects multiple communication faulted nodes on a subnet, and wishes to visually identify a specific communication faulted node.

Who communication faulted request message:

Used when the client has detected communication faulted node(s) on a subnet, and wishes to acquire their serial number and vendor ID.

Change MAC ID communication faulted request message:

Used when the client has detected a specific communication faulted node and wishes to change its MAC ID and have it attempt to come On-line at a specified MAC ID.

A device that implements the *communication faulted message* protocols shall implement support for all four of the above communication faulted message protocols.

5.2.4.5 Identify communication faulted messages**5.2.4.5.1 General**

The client may solicit an identify *communication faulted response message* from:

- ALL communication faulted nodes; or
- communication faulted nodes at one or more specific MAC IDs; or
- a communication faulted node with a specific vendor ID and serial number.

5.2.4.5.2 Identify request message – Multicast protocol

The *identify communication faulted request message* – multicast protocol shall be produced at group 4 message ID 0x2D, see Figure 27. The client produces this message when it is attempting to identify the existence of any nodes in the communications faulted state.

If the length of the *identify communication faulted request message* is two (2) bytes, any communication faulted node(s) servicing this message shall respond but not flash their LEDs.

Byte offset	Contents							
	7	6	5	4	3	2	1	0
0	Reserved [0]	Match	Value					
1	R/R [0]	Identify [0x4C]						

IEC

Figure 27 – Identify communication faulted request message – Multicast protocol

5.2.4.5.3 Identify response message – Multicast protocol

The format of the *identify communication faulted response message* is the same as the received *identify communication faulted request message* (with the match and value information the same as the information in the received request), except it is produced at group 4 message ID 0x2C, see Figure 28 and the R/R bit is set (1). The response message shall be initiated within 100 ms, upon the receipt of the request message.

Contents								
Byte offset	7	6	5	4	3	2	1	0
0	Reserved [0]	Match	Value					
1	R/R [1]	Identify [0x4C]						

IEC

Figure 28 – Communication faulted identify response message

If more than one node responds to the request, collisions will not occur on the subnet. This is a result of all communication faulted nodes producing the same message (CAN ID and data field). However, when more than one node replies, multiple messages may be received by the client.

5.2.4.5.4 Identify request message – Point-to-point protocol

The *identify communication faulted request message* – point-to-point protocol shall be produced with group 4 message ID 0x2D, see Figure 29. If a client (tool) detects communication faulted node(s) on a subnet, and the user wishes to visually identify a communication faulted node at a specific MAC ID, then the client (tool) shall produce the *identify communication faulted request message* – point-to-point protocol.

Byte offset	Contents							
	7	6	5	4	3	2	1	0
0	Reserved [0]	Match [0]	Value [0x3F]					
1	R/R [0]	Identify [0x4C]						
2	Vendor ID							Low byte
3								High byte
4	Serial number							Low byte
5								
6								
7								High byte

IEC

Figure 29 – Identify communication faulted request message – Point-to-point protocol

A client shall always set “match” to 0 and “value” to the 0x3F. A server may ignore the contents of byte 0 (checking vendor ID and serial number are sufficient for uniqueness).

IF a node accepts an *identify communication faulted request message* – point-to-point protocol AND is in the communication faulted state, it shall respond as follows:

- Flash its bi-colour network status LED or combined module/network status LED. The flash rate shall alternate the red LED ON for 250 ms, followed by the green LED turning ON for 250 ms. If the device does not have a bi-colour network status LED or combined module/network status LED, a suitable device-specific externally viewable indication shall be provided (e.g. via a display screen) to show that the *identify message* has been accepted;
- Produce an *identify communication faulted response message*, within 100 ms.

IF a node has accepted an *identify communication faulted request message* – point-to-point protocol AND is flashing its LED, it shall stop flashing its LED under the following conditions.

Similarly, if the device does not have a bi-colour network status LED or combined module/network status LED and the device-specific indication used to indicate that an *identify message* had been accepted is ON, it shall be turned OFF under these conditions:

- Another *identify communication faulted request message* – point-to-point protocol is not accepted within 500 ms.
- Another *identify communication faulted request message* – point-to-point protocol is received and NOT accepted (and another serial number or vendor ID).

If a node uses a network status LED or combined module/network status LED for the *identify message* indication, then, after it stops flashing the LED, the node shall turn off the LED for a minimum of 500 ms, but less than 1 s, prior to returning to its normal operation.

5.2.4.5.5 Identify response message – Point-to-point protocol

The *identify communication faulted response message* – point-to-point protocol is the same as the *identify communication faulted response message* – multicast protocol response.

5.2.4.5.6 Who communication faulted request message

The *who communication faulted request message* shall be produced at group 4 message ID 0x2D, see Figure 30. The *who communication faulted request message* is used to acquire the serial number and vendor ID of a communication faulted node.

Byte offset	Contents							
	7	6	5	4	3	2	1	0
0	Reserved [0]	Match	Value					
1	R/R [0]	Identify who [0x4B]						
2	Time delay byte offset [0-6]							

IEC

Figure 30 – Who communication faulted request message

IF the consuming node:

- is in the communication faulted state; and
- detects the *who communication faulted request message*; and
- meets the match and value criteria.

THEN the communication faulted node:

- shall wait a duration derived from the time delay byte offset; and
- produce a single *who response message* at the group 4 message ID 0x2C; and
- remain in the communication faulted state.

The client shall consume the duplicate MAC ID request message produced at the *communication faulted response message* ID 0x2C, thus acquiring the communication faulted node's serial number and vendor ID.

The accuracy of the wait period shall be dependent upon the timer resolution of the internal timer within the communication faulted node.

The interval the faulted node waits before responding is determined as follows:

- get the value at the time delay byte offset within the *who communication faulted request message* (0-6 are valid values);

- use this value as an offset into the *who response message* shown in Figure 31; then
- multiply the byte value at that offset times 50 ms; and
- delay the response for that period of time.

5.2.4.5.7 Who communication response

The *who response message*, produced following the specified delay, is the same protocol as a duplicate MAC ID request message, except it is produced at a group 4 message ID 0x2C, see Figure 31.

Byte offset	Contents							
	7	6	5	4	3	2	1	0
0	R/R [0]	Physical port number						
1	Vendor ID							Low byte
2								High byte
3	Serial number							Low byte
4								
5								
6								High byte

IEC

Figure 31 – Who response message

5.2.4.5.8 Change MAC ID communication faulted request message

The purpose of this message is to modify the MAC ID of a node in the communication faulted state. This message shall be produced using the group 4 message ID 0x2D, see Figure 32.

Byte offset	Contents							
	7	6	5	4	3	2	1	0
0	Reserved [0]		Value [New MAC ID]					
1	R/R [0]		Change MAC ID [0x4D]					
2	Vendor ID							Low byte
3								High byte
4	Serial number							Low byte
5								
6								
7								High byte

IEC

Figure 32 – Change MAC ID communication faulted request message

Nodes which support the *change_MAC ID communication faulted request message* feature shall not produce a *communication faulted response message*.

IF the consuming node:

- is in the communication faulted state; AND
- detects the *change_MAC ID communication faulted request message*; AND

- verifies that the serial number and the vendor ID matches its own,

THEN the faulted node:

- changes its MAC ID to the new MAC ID; AND
- enters the sending duplicate MAC ID state (see 5.4.3).

If, at the new MAC ID, the node fails the duplicate MAC ID check, the node shall retain the new MAC ID and re-enter the faulted state.

The contents of byte zero (0), bits 6 and 7 are ignored by the faulted node. Production of a value other than zero in this location is a client error, indicating non-conformance.

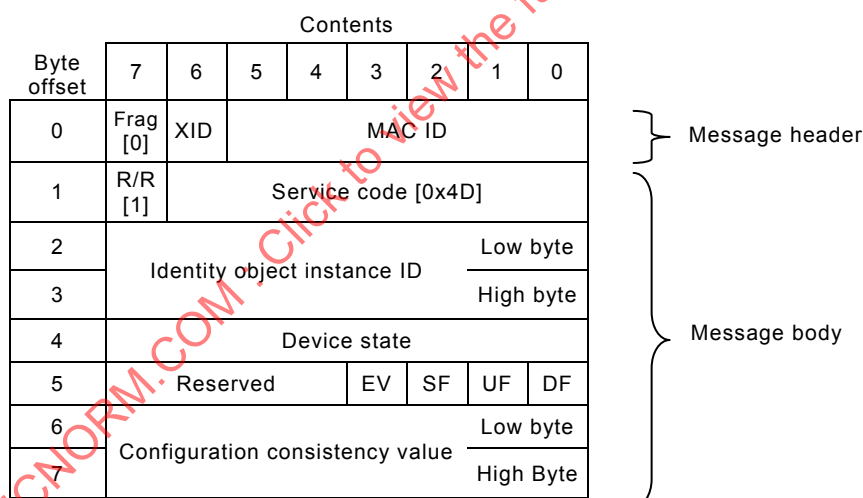
5.2.5 Device heartbeat

5.2.5.1 General

This subclause defines the protocol associated with the optional *device heartbeat message*. The *device heartbeat message* is triggered by the identity object, which is defined in 5.3.2.

5.2.5.2 Device heartbeat message

This message broadcasts the current state of the device. This message is transmitted by a UCMM capable device as an unconnected response message (message group 3, message ID 5) and by a group 2 only server as an unconnected response message (message group 2, message ID 3). The format of the *device heartbeat message* is shown in Figure 33.



IEC

Figure 33 – Device heartbeat message

5.2.5.3 Data frame contents, device heartbeat message

Frag (0)/Transaction ID/MAC ID – The heartbeat message is an unsolicited broadcast message; there is no destination MAC ID. The source MAC ID is specified in the message header. This is an exception to the general rule that the message header shall not contain the same MAC ID as the CAN identifier field.

R/R Bit (1) – Indicates this is a response message.

Service Code (0x4D) – Identifies this as a device heartbeat message.

Identity Object Instance ID – The instance ID of the identity object producing the device heartbeat message. This field shall be 2 bytes in length.

Device State – Attribute 8 of the associated identity object instance. If attribute 8 is not supported, device state shall be 3 (device operational).

EV – Event Flag, use is to be developed. This bit is ignored by the receiver and shall be set to zero by the transmitter.

SF – System Fault – A fault in the device caused by bus interaction (e.g. connection timeout). This flag is set when a system fault is present.

UF – User Fault – A fault in the device caused by user interaction. This flag is set when a user fault is present. The conditions under which this is set are vendor specific.

DF – Device Fault – An internal fault in the device not caused by user or bus interaction (e.g. hardware fault). This flag is set when a device fault is present.

Reserved bits – Use is to be developed. These bits currently are ignored by the receiver and shall be set to zero by the transmitter.

Configuration consistency value – Attribute 9 of the associated identity object instance. If attribute 9 is not supported configuration consistency value shall be zero.

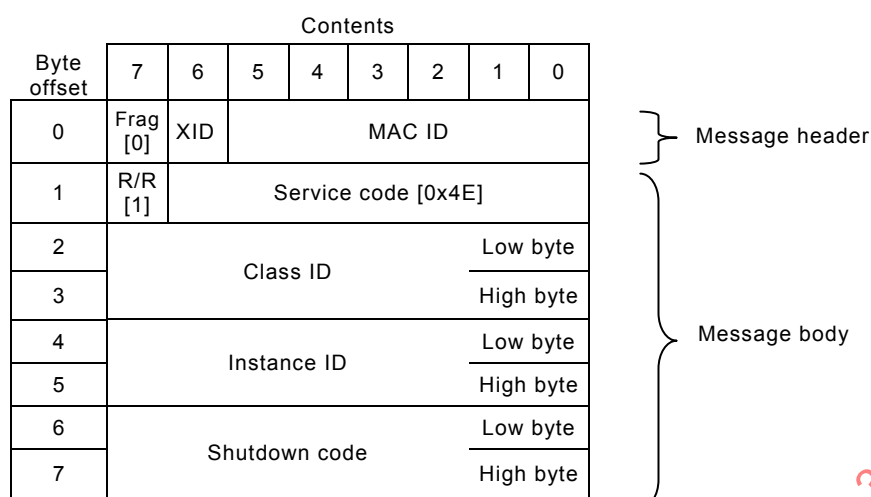
5.2.6 Device shutdown message

5.2.6.1 General

This subclause defines the protocol associated with the optional *device shutdown message*. The *device shutdown message* is produced by a device when it transitions to the offline state.

5.2.6.2 Device shutdown message

This message broadcasts the transition of a device to the offline or non-existent state. This message is transmitted by a UCMM capable device as an unconnected response message (message group 3, message ID 5) and by a group 2 only server as an unconnected response message (message group 2, message ID 3). The format of the *device shutdown message* is shown in Figure 34.



IEC

Figure 34 – Device shutdown message**5.2.6.3 Data frame contents, device shutdown message**

Frag (0)/Transaction ID/MAC ID – The shutdown message is an unsolicited broadcast message, there is no destination MAC ID. The source MAC ID is specified in the message header. This is an exception to the general rule that the message header shall not contain the same MAC ID as the CAN identifier field.

R/R Bit (1) – Indicates this is a response message.

Service Code (0x4E) – Identifies this as a device shutdown message.

Class ID / Instance ID – These two values identify the object class / instance responsible for the device's transition to the offline state. If the shutdown is not caused by a specific class, Class ID shall be 0. These fields shall each be 2 bytes in length.

Shutdown Code – This value which indicates the reason for the device's transition to the offline state is shown in Table 10 and Table 11.

Table 10 – Device shutdown message shutdown code ranges

Value	Meaning
0x0000 – 0x01FF	Open
0x0200 – 0x02FF	Vendor specific
0x0300 – 0x04FF	Object class specific
0x0500 – 0xFFFF	Reserved by DeviceNet for future use

Table 11 – Device shutdown message “Open” shutdown codes

Value	Meaning
0	Reserved
1	Operator shutdown
2	Operator reset
3	Remote shutdown
4	Remote reset
5	Internal diagnostic fault
6	Resource allocation fault
0x0007 – 0x01FF	Reserved

5.2.7 Duplicate MAC ID detection protocol

Each DeviceNet node shall be assigned a MAC ID and is required to participate in the duplicate MAC ID detection algorithm as defined in 5.4.

A special message ID value is defined within group 2 to identify the *duplicate MAC ID check message* (see Figure 35).

IDENTIFIER BITS											MESSAGE ID MEANING
10	9	8	7	6	5	4	3	2	1	0	
1	0	MAC ID					Group 2 message ID		Group 2 messages		
1	0	Destination MAC ID					1	1	1	Duplicate MAC ID check message	

IEC

Figure 35 – Duplicate MAC ID check CAN identifier field

The data field associated with the *duplicate MAC ID check message* has the format shown in Figure 36:

Byte offset	Contents							
	7	6	5	4	3	2	1	0
0	R/R		Physical port number					
1	Vendor ID						Low byte	
2							High byte	
3	Serial number						Low byte	
4								
5								
6							High byte	

IEC

Figure 36 – Duplicate MAC ID check message data field format

Duplicate MAC ID check message data field contents:

- **R/R bit:** Request/response bit. The value in this field indicates whether this is a duplicate MAC ID check request or a response message.
0 = Request
1 = Response

- **physical port number:** An identification value internally assigned to each physical DeviceNet port of a device. Devices that have a single port shall place the value zero (0) within this field;
- **vendor ID:** A 16 bit integer field (UINT) containing the identification code assigned to the manufacturer of the device that is transmitting the message;
- **serial number:** See 3.1.43.

5.2.8 Quick connect

The quick connect feature is an option enabled on a node-by-node basis. When enabled, a device transitions to the online state concurrently with sending the first duplicate MAC ID request message. The device is still required to execute the network State Transition Diagram, including going offline anytime a duplicate MAC ID response message is received.

Important: Although this feature allows a device to begin participating in network activity faster, it is at the expense of a delay in the duplicate node detection algorithm. It is left up to the user to guarantee that no nodes exist with the same MAC ID and that no more than one client device is configured to access the same device using the predefined master/slave connection set. Bus errors may occur if either of these conditions exists.

This feature is enabled within a device through a non-volatile attribute in the DeviceNet object. A device shall have this feature disabled (attribute set to "0") as the factory default.

5.3 DeviceNet communication object classes

5.3.1 General

DeviceNet communication in a node is modelled as a collection of objects: DeviceNet communication objects manage and provide the exchange of messages.

An object provides an abstract representation of a data structure within a node. An object class is a set of objects that all represent the same type of object. An object instance is the actual representation of a particular object within a class. Each instance of a class has the same set of attributes, but has its own particular set of attribute values.

An object instance and/or an object class have attributes, provide services and implement a behaviour.

Attributes are characteristics of an object and/or an object class. Attributes provide status information or govern the operation of an object. Services are invoked to trigger the object class or instance to perform a task. The behaviour of an object indicates how it responds to particular events.

The main communication objects used by DeviceNet are listed in this subclause (see 5.3.2.to 5.3.6).

Object class codes and their names are specified in IEC 61158-6-2:2014, 4.1.10.2.1. Data type specification and encoding is specified in IEC 61158-5-2:2014, Clause 5, and IEC 61158-6-2:2014, 4.2 and Clause 5.

5.3.2 Identity object class definition (class ID code: 0x01)

The identity object identifies and provides general information about the node.

The identity object is fully specified in IEC 61158-5-2:2014, 6.2.1.2.2 and IEC 61158-6-2:2014, 4.1.8.2.

5.3.3 Message router object class definition (class ID code: 0x02)

The message router object provides a messaging connection point through which a client may address a service to any object class or instance residing in the node.

The message router object is fully specified in IEC 61158-5-2:2014, 6.2.1.2.4 and IEC 61158-6-2:2014, 4.1.8.3.

5.3.4 DeviceNet object class definition (class ID code: 0x03)

The DeviceNet object provides the configuration and status of a physical attachment to DeviceNet (DeviceNet port). A product shall support one DeviceNet object per physical link attachment.

The DeviceNet object is specified in IEC 61158-4-2:2014, 7.7. This includes attributes and common services for the DeviceNet object class and instances.

DeviceNet object class specific services used to allocate and deallocate the predefined master/slave connection set are specified in 5.5.3.

5.3.5 Connection object class definition (class ID code: 0x05)

5.3.5.1 General

The connection object class allocates and manages the internal resources associated with both I/O and explicit messaging connections. The specific instance generated by the connection object class is referred to as a connection instance or a connection object.

The connection object class is fully specified in IEC 61158-5-2, IEC 61158-6-2 and IEC 61784-3-2. This includes:

- attributes and services for the connection object class and instances (see IEC 61158-5-2:2014, 6.2.3 and IEC 61158-6-2:2014, 4.1.8.8);
- specific services for functional safety (see IEC 61784-3-2);
- connection timing (see IEC 61158-5-2:2014, 6.2.3);
- connection instance behaviour (see IEC 61158-6-2:2014, 7.2).

5.3.5.2 Dynamic management of message IDs

Dynamically establishing both I/O and explicit messaging connections requires all end-points to implement internal message ID allocation procedures. These procedures shall also mark a previously allocated message ID as available when that message ID is no longer in use.

To reduce the likelihood of CID errors, the following rules shall be followed:

- a) the CID allocation process shall ensure that no two nodes on DeviceNet have configurations that allow transmission of an identical bit pattern within the connection identifier field;
- b) the CID deallocation process shall ensure that all end-points of a connection have timed out prior to reusing a message ID:
 - if the message ID is associated with a connection that activates an inactivity/watchdog timer, then a new inactivity/watchdog timer is activated. Upon expiration of this timer, the message ID is marked as available;
 - if the message ID is associated with a connection that does not activate an inactivity/watchdog timer, then the message ID can be immediately marked as available (it is assumed that this was taken into consideration when the connection was established).

Important: For a connection using transport class 0 or a connection whose `expected_packet_rate` attribute has been set to zero (0), rule b) cannot be achieved based on that connection object alone. For these types of connections, tasks that will result in the deallocation and possible re-use of the associated message ID(s) should be performed with caution (i.e., configuring the `watchdog_timeout_action` attribute to Auto Delete, manually deleting a connection object via the transmission of the delete service).

EXAMPLE 1

If the server end-point of a transport class 0 connection experiences an inactivity/watchdog timeout, it cannot know the state of the client based solely on this connection. The client could have just missed transmitting the message in a timely fashion and could still think the connection is operating normally.

EXAMPLE 2

If the client end-point of a connection whose `expected_packet_rate` has been set to zero (0) is deleted for some reason, the server end-point(s) could still be active.

5.3.6 Acknowledge handler object class definition (class ID code: 0x2B)

The acknowledge handler object manages the reception of message acknowledgements.

The acknowledge handler object is fully specified in IEC 61158-5-2:2014, 6.2.1.2.5 and IEC 61158-6-2:2014, 4.1.8.5.

5.4 Link access state machine

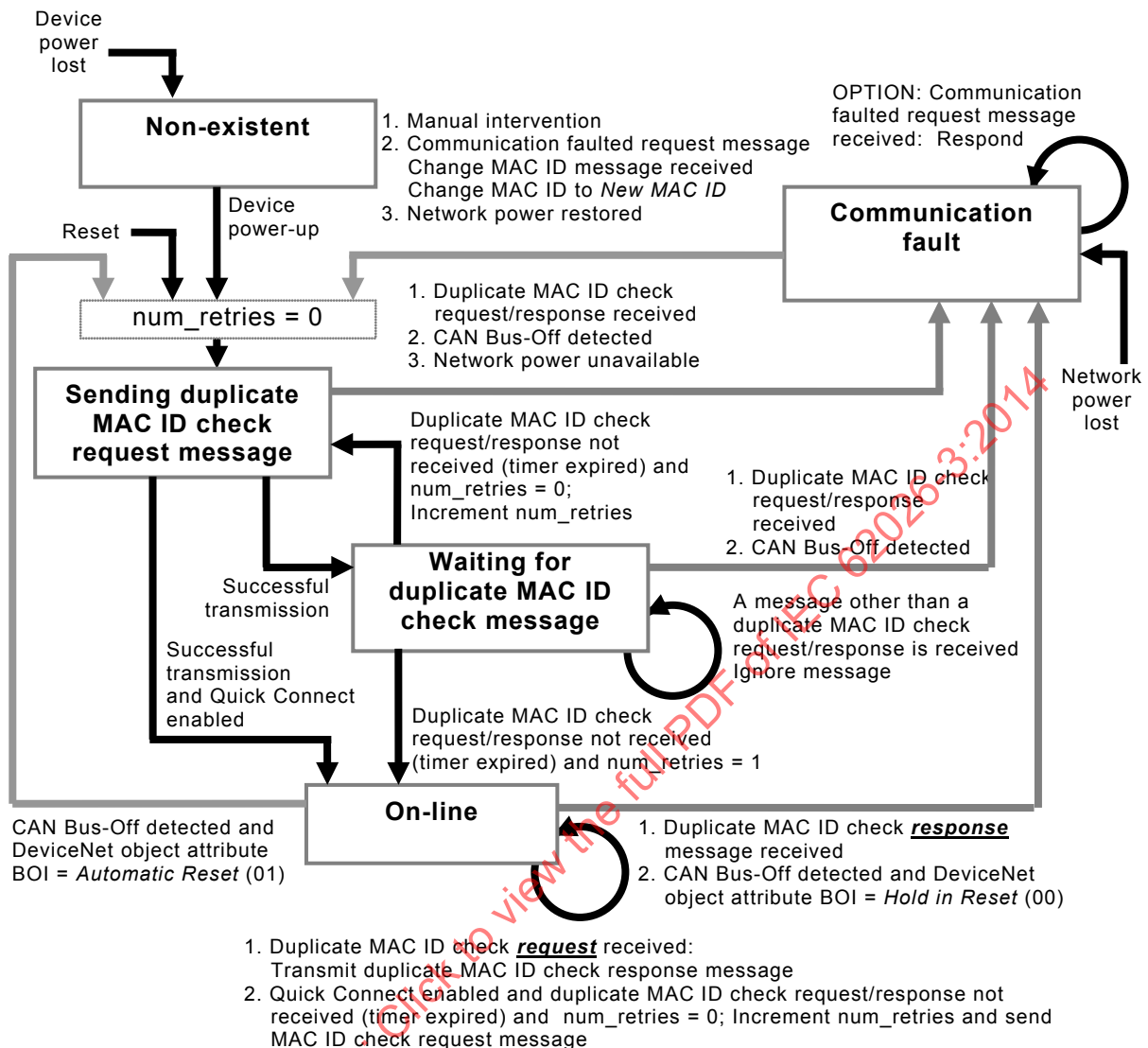
5.4.1 General

This subclause defines the link access state machine that every DeviceNet node shall implement. The link access state machine is described by the following:

- tasks that shall be performed prior to communicating through the CDI;
- link events that affect a node's ability to communicate through the CDI.

5.4.2 State transition diagram and event matrix

Figure 37 provides a general overview of the link access state machine.



IEC

Figure 37 – Link access state transition diagram

Table 12 provides a detailed state event matrix for the link access state machine.

Table 12 – Link access state event matrix (1 of 2)

Event	State			
	Sending duplicate MAC ID check request	Waiting for duplicate MAC ID check message	On-line	Communication fault
Successful transmission of the duplicate MAC ID check request message	Activate 1 s timer ^a . If quick connect is enabled transition to On-line, otherwise transition to waiting for duplicate MAC ID check message	Not applicable	Not applicable	Not applicable
Bus-off detected	Transceiver held	Transceiver held in	Access the DeviceNet	Not applicable

Event	State			
	Sending duplicate MAC ID check request	Waiting for duplicate MAC ID check message	On-line	Communication fault
	in reset. Transition to the communication fault state	reset. Transition to the communication fault state	object's BOI attribute. If the BOI attribute indicates that the transceiver shall be held in reset, then transition to communication fault. If the BOI attribute indicates that the MAC and transceiver shall be automatically reset, then: 1. reset, 2. request the transmission of the duplicate MAC ID check request message, and 3. transition to the sending duplicate MAC ID check request state	
Duplicate MAC ID check request message received	Duplicate MAC ID detected. Transition to the communication fault state	Duplicate MAC ID detected. Transition to the communication fault state	Transmit the duplicate MAC ID check response message	Discard message
Duplicate MAC ID check response message received	Duplicate MAC ID detected. Transition to the communication fault state	Duplicate MAC ID detected. Transition to the communication fault state	Duplicate MAC ID detected. Transition to the communication fault state	Discard message
1 s duplicate MAC ID check message timer expires ^a	Not applicable	If this is the first time-out, then request the transmission of the duplicate MAC ID check request message again and transition to sending duplicate MAC ID check request. If this is the second consecutive time-out, then transition to on-line	If quick connect is enabled request the transmission of the duplicate MAC ID check request message again	Not applicable
Internal message transmission request	Return internal error	Return internal error	Transmit message	Return internal error

Table 12 (2 of 2)

Event	State			
	Sending duplicate MAC ID check request	Waiting for duplicate MAC ID check message	On-line	Communication fault
A message other than duplicate MAC ID check request/response is received	Discard message	Discard message	Process the received message as appropriate	Discard message
A communication faulted request message is received	Discard message	Discard message	Discard message	Process the received message as appropriate
^a Valid timer range is 0,9 s to 1,5 s.				

The CAN fault confinement state machine considers the possibility that during system start-up /wake-up), only one node may be present on the link. If this node transmits a message, it will experience an acknowledgement error and will automatically repeat the message. In this situation the node will transition to error passive but not bus-off. For this reason, the only event that signifies an unsuccessful transmission of a duplicate MAC ID check message (request or response) is the bus-off event. If an error passive or error warning indication is received during the transmission of a duplicate MAC ID check message, then it shall be ignored, as it has no effect on the duplicate MAC ID detection state machine.

5.4.3 Duplicate MAC ID detection

The main step involved in the link access state machine is the execution of a duplicate MAC ID detection algorithm. Each node on DeviceNet is assigned a unique MAC ID, and to protect against errors, all DeviceNet nodes shall participate in the duplicate MAC ID detection algorithm.

NOTE The protocol associated with the duplicate MAC ID detection algorithm is described in 5.2.7.

As stated in 5.2.7, a special message within message group 2 is defined for performing duplicate MAC ID detection.

A DeviceNet node shall receive and process any duplicate MAC ID check message that contains its MAC ID in the message group 2 identifier field.

After transmitting a duplicate MAC ID check request message, a module shall wait 1 s before timing out and taking the appropriate action defined by the link access state machine.

The duplicate MAC ID check request message shall be transmitted twice without receiving a subsequent duplicate MAC ID check request or response message before transitioning to on-line.

See 9.2.3 and 9.3.2 for test specifications regarding the power ON behaviour and handling of the duplicate MAC ID mechanisms.

5.5 Predefined master/slave connection set

5.5.1 General

The preceding subclauses present the general rules for establishing connections between devices. The general rules call for the utilisation of an explicit messaging connection to create and configure connection objects within each connection end-point. This subclause uses the general rules as a basis for the definition of a set of connections, which facilitate

communications typically seen in a master/slave relationship. These connections are referred to collectively as the predefined master/slave connection set:

- **group 2 server:** An unconnected message manager (UCMM) capable device that has been configured to act as the server for the predefined master/slave identifier connections;
- **group 2 client:** A UCMM capable device that has gained ownership of the predefined master/slave connection set within a server such that it may act as the client on those connections;
- **UCMM capable device:** A device that supports the UCMM;
- **UCMM incapable device:** A device that does not support the UCMM;
- **group 2 only server:** A slave device that is UCMM incapable and uses the predefined master/slave connection set to establish communications. A group 2 only device can transmit and receive only those identifiers defined by the predefined master/slave connection set;
- **group 2 only client:** A device that is acting as a group 2 client to a group 2 only server. The group 2 only client provides the UCMM functionality for group 2 only servers that it has allocated;
- **DeviceNet master:** Refers to a type of application called master/slave. The DeviceNet master is the device that gathers and distributes I/O data for the process controller. A master scans its slave devices based on a scan list it contains. With respect to the network, the master is a group 2 client or a group 2 only client;
- **DeviceNet slave:** Refers to a type of application called master/slave. A slave returns I/O data to its master when it is scanned. With respect to the network, the slave is a group 2 server or a group 2 only server;
- **predefined master/slave connection set:** A set of connections that facilitates communications typically seen in a master/slave relationship. Many of the steps involved in the creation and configuration of an application to application connection have been removed within the predefined master/slave connection set definition. This, in turn, presents the means by which a communication environment can be established using less network and device resources.

5.5.2 Predefined master/slave connection set messages

The CAN identifier fields associated with the predefined master/slave connection set are shown in Table 13 together with the identifiers that shall be used with all connection based messaging used in the predefined master/slave connection set.

Table 13 – Predefined master/slave connection set identifier fields

Identifier bits											Usage	Range
10	9	8	7	6	5	4	3	2	1	0		
0	Group 1 message ID				Source MAC ID						Group 1 messages	0x000-0x3FF
0	1	1	0	0	Source MAC ID						Slave's I/O multicast poll response message	
0	1	1	0	1	Source MAC ID						Slave's I/O change of state or cyclic message	
0	1	1	1	0	Source MAC ID						Slave's I/O bit-strobe response message	
0	1	1	1	1	Source MAC ID						Slave's I/O poll response or change of state/cyclic acknowledge message	
1	0	MAC ID						Group 2 message ID			Group 2 messages	0x400-0x5FF
1	0	Source MAC ID						0	0	0	Master's I/O bit-strobe command message	
1	0	Multicast MAC ID						0	0	1	Master's I/O multicast poll command message	
1	0	Destination MAC ID						0	1	0	Master's change of state or cyclic acknowledge message	
1	0	Source MAC ID						0	1	1	Slave's explicit response messages	
1	0	Destination MAC ID						1	0	0	Master's explicit request messages	
1	0	Destination MAC ID						1	0	1	Master's I/O poll command/change of state/cyclic message	
1	0	Destination MAC ID						1	1	0	Group 2 only unconnected explicit request messages	
1	0	Destination MAC ID						1	1	1	Duplicate MAC ID check messages	

The following types of messages are included in Table 13:

- **I/O bit-strobe command/response messages:** The bit-strobe command is an I/O message that is transmitted by the master. Multiple slaves may receive and react to the same bit-strobe command. The bit-strobe response is an I/O message that a slave transmits back to the master when the bit-strobe command is received;
- **I/O poll command/response messages:** The poll command is an I/O message that is transmitted by the master. A poll command is directed to a specific slave. The poll response is an I/O message that a slave transmits back to the master when the poll command is received;
- **I/O change of state/cyclic messages:** The change of state/cyclic message is transmitted by either the master or the slave. A change of state/cyclic message is directed to a specific node. An acknowledge message shall be returned in response to this message unless the configuration includes suppression of acknowledge messages;
- **I/O multicast poll messages:** The multicast poll command is an I/O message that is transmitted by the master. A multicast poll is directed towards one or more slaves. The multicast poll response is an I/O message that a slave transmits back to the master when the multicast poll command is received;
- **explicit response/request messages:** See 5.2.1.6;
- **group 2 only unconnected explicit request messages:** The group 2 only unconnected explicit request port is used to allocate/release the predefined master/slave connection set;
- **group 2 only unconnected explicit response messages:** The group 2 only unconnected explicit response port is used to respond to group 2 only unconnected explicit request messages and to send device heartbeat / device shutdown messages. These messages are transmitted using the same identifier (group 2, message ID = 3) as explicit response messages;
- **duplicate MAC ID check message:** See 5.2.7.

5.5.3 DeviceNet object class specific services for the master/slave connection set

5.5.3.1 Allocate_master/slave_connection_set (service code: 0x4B)

5.5.3.1.1 General

This service allocates the predefined master/slave connection set. General error codes are defined in IEC 61158-5-2:2014, 6.2.1.3.3 and IEC 61158-6-2:2014, 4.1.11. DeviceNet object specific additional error code values are defined in 5.5.3.5. This service shall be transmitted using either the unconnected explicit request message of the master/slave connection set (see 5.5.2), or an explicit messaging connection.

The allocate_master/slave_connection_set service performs the following:

- connection object create;
- connection object configure.

See 9.3.4 and 9.3.5 for test specifications regarding the allocation of the master/slave connection set for both explicit and I/O messaging connections.

5.5.3.1.2 Request service data field parameters

The information in Table 14 is specified within the service data field of an allocate_master/slave_connection_set request.

Table 14 – Allocate_master/slave_connection_set request service data field parameters

Name	Data type	Description of parameter
Allocation choice	BYTE	Indicates which connections from the predefined master/slave connection set are to be allocated/configured for use by the master
Allocator's MAC ID	USINT	Contains the MAC ID associated with the node requesting the allocation

The allocation choice parameter is specified within a single byte (see Figure 38). Each bit denotes an explicit message and/or I/O connection(s) from the predefined master/slave connection set that are to be allocated, or in the case of acknowledge suppression, a command. If the bit is set to 1, then a request is being made to allocate the corresponding connection. If a bit is set to 0, then the requester does not want to allocate the corresponding connection.

7	6	5	4	3	2	1	0
Reserved	Acknowledge suppression	Cyclic	Change of state	Multicast Polled	Bit strobed	Polled	Explicit message

IEC

Figure 38 – Allocation choice byte contents

Bits 3 and 7 shall be set to 0 and the slave shall verify this requirement when the allocate_master/slave_connection_set request is received.

Figure 39 shows the format of this explicit message.

Contents								
Byte offset	7	6	5	4	3	2	1	0
0	Frag (0)	XID	MAC ID					
1	R/R (0)	Service (0x4B)						
	Class ID (3)							
	Instance ID (1)							
	Allocation choice							
	0	0	Allocator's MAC ID					

IEC

Figure 39 – Allocate_master/slave_connection_set request message

Allocate_master/slave_connection_set request message contents:

- **frag (0)/transaction ID/MAC ID:** As defined in 5.2.1.2;
- **R/R bit (0):** Indicates this is a request message;
- **service code (0x4B):** Identifies this message as an allocate_master/slave_connection_set service;
- **class ID:** Defines the object class to which this request is directed. When this message is transmitted across the unconnected explicit request message port of the master/slave connection set (see 5.5.2), the class ID value is specified within an 8 bit integer field. The class ID shall be set to 3;
- **instance ID:** Defines the particular instance within the object class to which this request is directed. When this message is transmitted across the unconnected explicit request message port of the master/slave connection set (see 5.5.2), the instance ID value is specified within an 8 bit integer field. The instance ID shall be set to 1;
- **allocation choice:** Specified by the byte following the instance ID field;
- **allocator's MAC ID:** Specified within the byte following the allocation choice field.

5.5.3.1.3 Success response service data field parameters

The information contained within the service data field of a successful allocate_master/slave_connection_set response is described in Table 15.

Table 15 – Allocate_master/slave_connection set response parameters

Name	Data type	Description of parameter
Message body format	Defined below	This is semantically equivalent to the actual message body format parameter returned with an open explicit message connection response message (as described in 5.2.1.5). This argument is significant when the allocate_master/slave_connection_set request was received across the unconnected explicit request message port of the master/slave connection set (see 5.5.2). It indicates the message body format associated with subsequent messages transmitted across the explicit messaging connection within the predefined master/slave connection set. If the allocate_master/slave_connection_set request was received across an explicit messaging connection within a UCMM capable device, then this parameter is set to the actual message body format associated with that explicit messaging connection

Figure 40 shows the format of a success response to the allocate_master/slave_connection_set request.

Contents								
Byte offset	7	6	5	4	3	2	1	0
0	Frag (0)	XID	MAC ID					
1	R/R (1)	Service code (0x4B)						
2	Reserved (all bits = 0)				Message body format			

IEC

Figure 40 – Success response to allocate_master/slave_connection_set request

Allocate_master/slave_connection_set request success response contents:

- **frag (0)/transaction ID/MAC ID:** As defined in 5.2.1.2;
- **R/R bit (1):** Indicates this is a response message;
- **service code (0x4B):** Identifies this message as an allocate_master/slave_connection_set service;
- **reserved bits:** These bits are not used by the receiver of the response and shall be set to 0 by the transmitter of the response message;
- **message body format:** As described in Table 15.

5.5.3.2 Allocate_master/slave_connection_set server behaviour

- a) If an error is encountered, then none of the requested connections shall be allocated. If this request cannot be fully serviced, then none of the requested allocations shall take place.
- b) If the receiving device does not support the predefined master/slave connection set, then an error response is returned. The general error code within the error response shall be set to 0x08 to indicate service not supported.
- c) The receiving device (slave) validates the allocator's MAC ID parameter within the request as follows:
 - if the predefined master/slave connection set is allocated and this request is not from the current master, then the slave returns an error. The general error code within the error response shall be set to 0x0C, with the additional error code set to an object specific value of 01;
 - if the predefined master/slave connection set is not allocated, then the slave validates the allocation choice parameter as specified in d);
 - if the predefined master/slave connection set is allocated and this request is from the current master, then the slave shall validate the allocation choice parameter as specified in d).
- d) The slave shall validate the allocation choice parameter within the request. If the slave does not support one of the connections specified in the allocation choice argument, then an error response shall be returned. The general error code within the error response shall be set to 02, with the additional error code set to an object specific value of 0x02.

If any of the connection(s) being requested are supported by this slave and have already been allocated to the master denoted by the allocator's MAC ID argument, the slave shall return an error response with the general error code set to 0x0B, with the additional error code set to an object specific value of 0x02. If the requested I/O connection is in the timed-out state the slave shall reallocate the I/O connection, setting it to the configuring state.

If the allocation choice byte has no bits set the slave shall return an error response with the general error code set to 0x09, with the additional error code set to an object specific value of 0x02.

If a resource that is required for use with the requested connections is not available, then an error response shall be returned with the general error code set to 0x02, and the additional error code set to an object specific value of 0x04.

The change of state and cyclic allocation choices are mutually exclusive. If an allocation request would result in both the cyclic and change of state bits being set, an error response shall be returned. The general error code within the error response shall be set to 0x09 (invalid attribute value) with an additional error code of 0x02.

If a master has allocated the change of state/cyclic connection set, and a subsequent allocation request is received with the polled allocation bit set, an error shall be returned. The general error code within the error response shall be set to 0x02 to indicate resource unavailable. If the allocation choice byte has the acknowledge suppression bit set and neither the change of state bit nor the cyclic bit are set, an error response shall be returned. The general error code within the error response shall be set to 0x09 with an additional error code of 0x02.

- e) The slave notes the fact that the predefined master/slave connection set has been allocated to the MAC ID within the allocator's MAC ID field by updating the DeviceNet object's allocation information attribute. If necessary, the produced_connection_id and/or consumed_connection_id attributes of the connection object(s) may now be initialised.

The allocation choice byte of the allocation information attribute indicates which connection objects from the predefined master/slave connection set are active (in the configuring, or established state). This byte is updated whenever a master/slave connection object changes state.

- f) Any allocated I/O connection(s) transition to the configuring state. With respect to the predefined master/slave connection objects, an implied apply_attributes service accompanies a set_attribute_single of the expected_packet_rate attribute while the connection is in the configuring state. A set_attribute_single of the expected_packet_rate attribute triggers the execution of the steps performed with an apply_attributes service and causes the predefined master/slave I/O connection object to transition to the established state. If an error is encountered when performing the apply_attributes portion, then the expected_packet_rate attribute shall be reset to its previous value, and an error shall be returned whose general error code is set to 0x09 and whose additional error code is set to the offending connection object's attribute ID.
- g) If allocated, the explicit messaging connection transitions to the established state. The inactivity/watchdog timer is started using the initial value specified in IEC 61158-5-2:2014, 6.2.3.2.1.7.

UCMM capable devices which support the predefined master/slave connection set shall support the allocation and use of the predefined master/slave connection set explicit messaging connection.

- h) Either the master/slave connection set's explicit messaging connection or a dynamically established explicit messaging connection acts as the allocated I/O connection(s) parent until the I/O connection(s) are in the established state. The term parent is used to indicate that if the explicit messaging connection is deleted and none of the allocated I/O connections exist in the established state, then the predefined master/slave connection set is automatically released by the slave. The logic describing which explicit messaging connection is to act as the parent is defined in Figure 41.

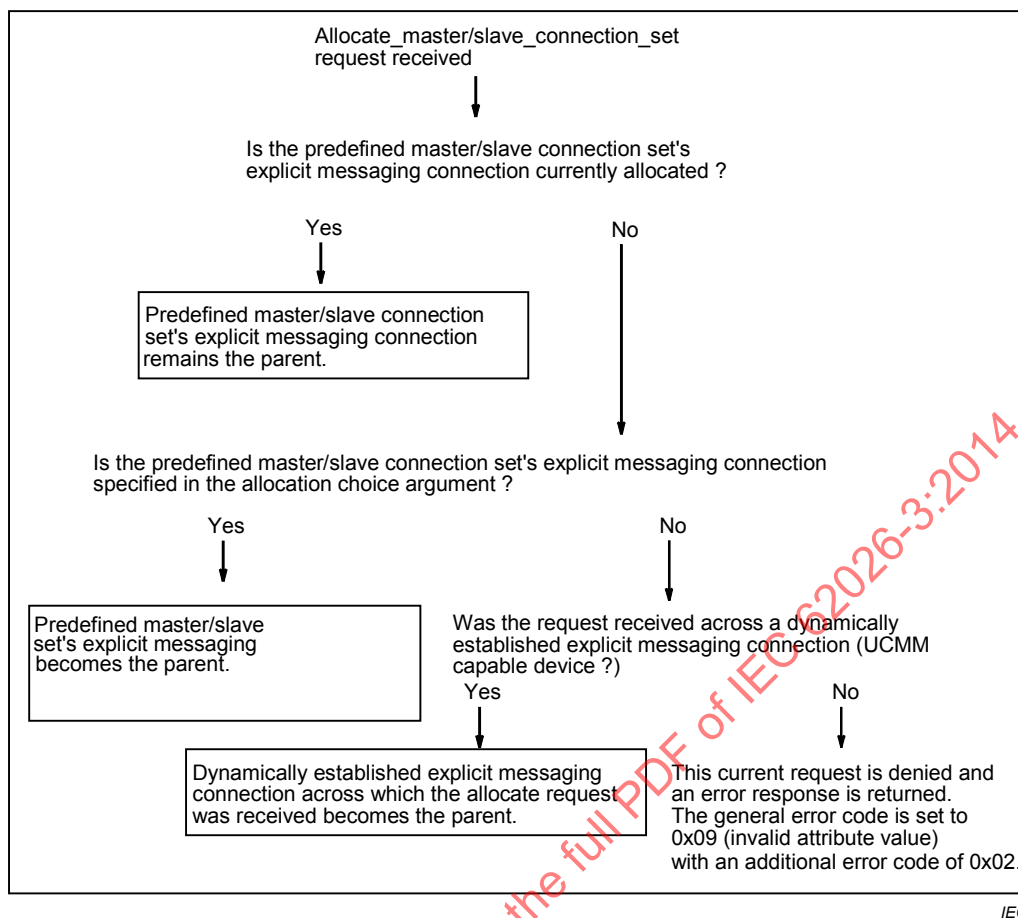


Figure 41 – Parent explicit messaging connection logic

The slave shall automatically release the predefined master/slave connection set and all connections shall return to the non-existent state if all of the following conditions are true:

- if none of the connection objects associated with the predefined master/slave connection set exist in the established state;
- if the parent explicit messaging connection object is not in the established state.

5.5.3.3 Release_master/slave_connection_set (service code: 0x4C)

5.5.3.3.1 General

This service is used to deallocate the predefined master/slave connection set within a slave. General error codes are defined in IEC 61158-5-2:2014, 6.2.1.3.3 and IEC 61158-6-2:2014, 4.1.11. DeviceNet object specific additional error code values are defined in 5.5.3.5. This service can be transmitted across the unconnected explicit request message port of the master/slave connection set (see 5.5.2) as well as an explicit messaging connection.

5.5.3.3.2 Request service data field parameters

The information in Table 16 is specified within the service data field of a release_master/slave_connection_set request.

Table 16 – Release_master/slave_connection_set request service data field parameters

Name	Data type	Description of parameter
Release choice	BYTE	Indicates which predefined master/slave connections are to be released

The release choice parameter is specified within a single byte (see Figure 42). Each bit denotes an explicit message and/or I/O connection(s) to be released. If the bit is set to 1, then a request is being made to release the corresponding connection. If a bit is set to 0, then the requester does not want to release the corresponding connection.

7	6	5	4	3	2	1	0
Reserved	Ignored	Cyclic	Change of state	Multicast polled	Bit strobed	Polled	Explicit message

IEC

Figure 42 – Release choice byte contents

Bits 3 and 7 shall be set to 0 and the slave shall verify this requirement when the release_master/slave_connection_set request is received.

A value of 00 is invalid.

Figure 43 shows the format of this explicit message.

Contents									
Byte offset	7	6	5	4	3	2	1	0	
0	Frag (0)	XID	MAC ID						
1	R/R (0)	Service (0x4C)							
	Class ID (3)								
	Instance ID (1)								
	Release choice								

IEC

Figure 43 – Release_master/slave_connection set request message

Release_master/slave_connection set request message contents:

- **frag (0)/transaction ID/MAC ID:** See 5.2.1.2;
- **R/R bit (0):** Indicates this is a request message;
- **service code (0x4C):** Identifies this message as a release_master/slave_connection_set service;
- **class ID:** Defines the object class to which this request is directed. When this message is transmitted across the unconnected explicit request message port of the master/slave connection set (see 5.5.2), the class ID value is specified within an 8 bit integer field. The class ID shall be set to 3;

- **instance ID:** Defines the particular instance within the object class to which this request is directed. When this message is transmitted across the unconnected explicit request message port of the master/slave connection set (see 5.5.2), the instance ID value is specified within an 8 bit integer field. The instance ID shall be set to 1;
- **release choice:** Specified within the byte following the instance ID field.

5.5.3.3.3 Success response message

Figure 44 shows the format of a success response to the release_master/slave_connection_set request.

		Contents							
Byte offset		7	6	5	4	3	2	1	0
0	Frag (0)	XID		MAC ID					
1	R/R (1)	Service (0x4C)							

Figure 44 – Success response to release_master/slave_connection_set request

Release_master/slave_connection_set request success response contents:

- **frag (0)/transaction ID/MAC ID:** See 5.2.1.2;
- **R/R bit (1):** Indicates this is a response message;
- **service code (0x4C):** Identifies this as a release_master/slave_connection_set service.

5.5.3.4 Release_master/slave_connection_set server behaviour

If an error is encountered, then none of the specified connections are released. If this request cannot be fully serviced, then none of the requested releases take place.

If the receiving device does not support the predefined master/slave connection set, then an error response shall be returned. The general error code within the error response shall be set to 0x08.

The receiving device (slave) shall validate the release choice parameter within the request. If the slave does not support one of the connections specified in the release choice argument, then an error response shall be returned. The general error code within the error response shall be set to 0x02, with the additional error code set to an object specific value of 0x02.

If the release choice byte has no bits set (all bits are 0) the slave shall return an error response with the general error code set to 0x09 (invalid attribute value), with the additional error code set to an object specific value of 0x02.

If one of the specified connections is in the non-existent state, then an error response shall be returned. The general error code within the error response shall be set to 0x0B.

The slave shall verify that it is in a state that allows it to discontinue use of the specified connection(s). If this is not the case, then an error response shall be returned. The general error code within the error response shall be set to 0x0C.

If the request is valid, then the slave shall release all resources associated with the specified connection(s). If this results in all predefined master/slave connections being released, the slave shall note the fact that the predefined master/slave connection set is no longer allocated by updating the allocation information attribute.

The slave shall not check to see that the release request came from its master.

If this request has resulted in none of the predefined master/slave connections existing in the established state, then the slave shall release the predefined master/slave connection set and all connections shall return to the non-existent state.

5.5.3.5 Error codes specific to the DeviceNet object

Table 17 lists additional error codes specific to the DeviceNet object.

Table 17 – DeviceNet object specific additional error codes

Value	Meaning
01	Predefined master/slave connection set allocation conflict. This is returned when an allocate_master/slave_connection_set request is received and the slave has already allocated the predefined master/slave connection set to another master
02	Invalid allocation/release choice parameter. This is returned when an allocate/release_master/slave_connection_set request is received and: 1) The slave does not support the choice specified in the choice parameter. 2) The slave was asked to allocate/release connection(s) already allocated/released. 3) The allocation choice/release byte contained all zeros, an invalid combination of bits, or did not contain the explicit message allocation choice when required
03	A server that does not support UCMM received a message that was not an allocate or release message on the unconnected explicit request message port of the master/slave connection set (see 5.5.2)
04	Resource required for use with the predefined master/slave connection set is unavailable

5.5.4 Slave connection object characteristics

5.5.4.1 General

This subclause presents the externally visible characteristics of the connection objects associated with the predefined master/slave connection set within slave devices. The predefined master/slave connection objects described for slave devices are:

- **the bit-strobe connection:** Responsible for receiving the master's bit-strobe command and returning the associated bit-strobe response;
- **the poll connection:** Responsible for receiving the master's poll command and returning the associated poll response;
- **the explicit messaging connection:** Responsible for the reception of explicit requests and returning associated responses;
- **the change of state/cyclic connection:** Responsible for sending the change of state/cyclic message and receiving the acknowledge response if not suppressed;
- **the multicast poll connection:** Responsible for receiving the master's multicast poll command and returning the associated multicast poll response.

This subclause gives additional information for connection objects within the predefined master/slave connection set. Except where noted, all information specified in 5.3 applies to the connection objects described in this subclause.

5.5.4.2 Connection instance IDs

Every existing connection object has an assigned connection instance ID which identifies the connection object within the connection class. The connection instance IDs that shall be used by a slave device to identify predefined master/slave connection objects are shown in Table 18.

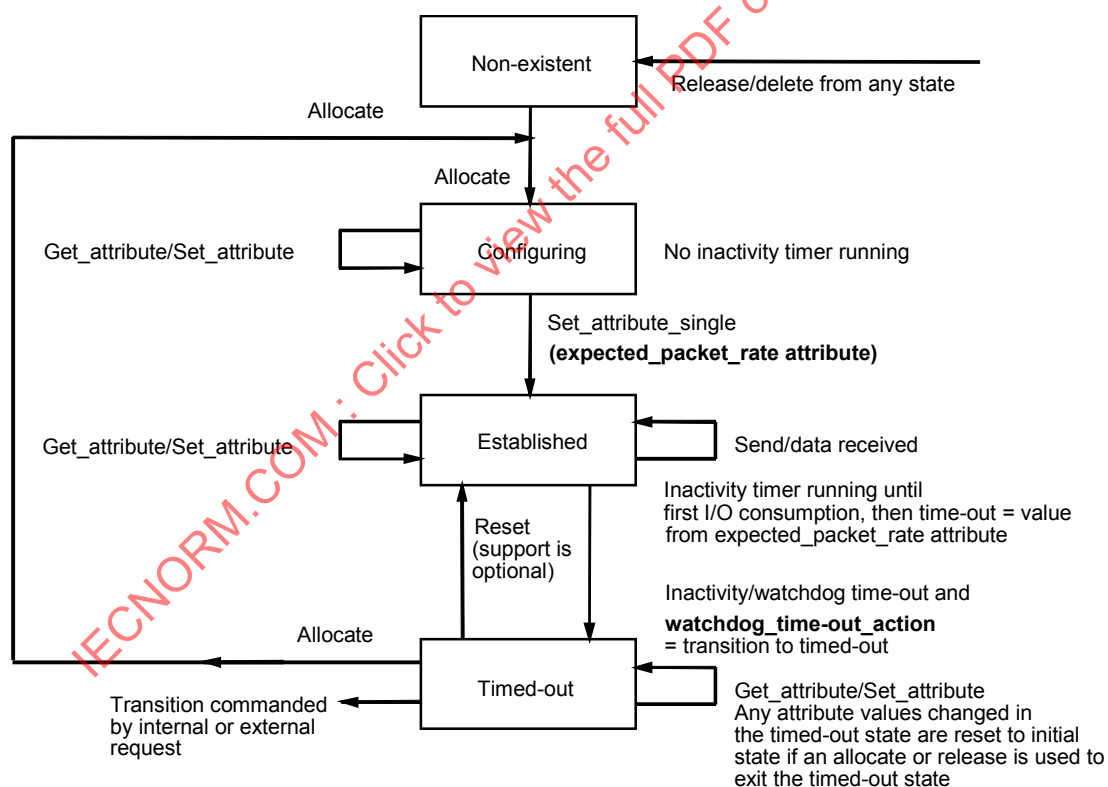
Table 18 – Connection instance IDs for predefined master/slave connections

Connection instance ID	Description
1	Designates the explicit messaging connection into the server
2	Designates the poll I/O connection
3	Designates the bit-strobe I/O connection
4	Designates the slave's change of state or cyclic I/O connection
5	Designates the multi-cast poll I/O connection

A slave shall reserve the Instance IDs from Table 18 for the predefined master/slave connections that it supports. For example, if a device supports the polled I/O connection, it shall reserve/utilize connection instance ID #2 to identify the polled I/O connection object. If a device does not support the poll connection, then it is free to allocate connection instance ID #2 to identify some other connection object.

5.5.4.3 Predefined master/slave connection instance behaviour

Figure 45 illustrates the predefined master/slave connection set I/O connection object state transition diagram.



The allocate and release services reset the connection instance. All connection object attributes are reset to their default values.

IEC

Figure 45 – Predefined master/slave I/O connection state transition diagram

For attribute modification, predefined master/slave I/O connections shall support at least the modification of the `expected_packet_rate` attribute.

The state event matrix presented in Table 19 provides a formal definition of the behaviour of I/O connections within the predefined master/slave connection set. This state event matrix

inherits from and/or overrides actions presented in the I/O connection object state event matrix in 5.3.

Table 19 – Predefined master/slave I/O connection state event matrix (1 of 2)

Event	I/O connection object state			
	Non-existent	Configuring	Established	Timed out
DeviceNet object receives an allocate master/slave connection set request that passes all error checks specified in 5.5.3.2. This request specifies one of the predefined master/slave I/O connections	Create a connection object for each requested I/O connection and set attributes to default values specified in 5.5.3.1. Transition to configuring	This is an error. See 5.5.3.2	This is an error. See 5.5.3.2	Set attributes to default values specified in 5.5.3.1. Transition to configuring
Connection class receives a delete request or the UCMM receives a close request and the request specifies a predefined master/slave I/O connection object	As described in Table 193 of IEC 61158-6-2:2014, 7.2.1	Release all associated resources. Transition to non-existent ^a	Release all associated resources. Transition to non-existent ^a	Release all associated resources. Transition to non-existent ^a
DeviceNet object receives a release master/slave connection set request that passes all error checks specified in 5.5.3.2. This request specifies one of the predefined master/slave I/O connections	This is an error. See 5.5.3.2	Release all associated resources. Transition to non-existent ^a	Release all associated resources. Transition to non-existent ^a	Release all associated resources. Transition to non-existent ^a
Set_attribute_single	As described in IEC 61158-6-2:2014, 7.2.1	Validate/service the request according to the access rules presented in 5.5.4.4. If this is a valid request to set the expected_packet_rate attribute, then perform the steps specified in IEC 61158-6-2:2014, 7.2.1 under the apply_attributes event in the configuring state and transition to established. Return appropriate response	Validate/service the request according to the access rules presented in 5.5.4.4. Return appropriate response	Validate/service the request according to the access rules presented in 5.5.4.4. Return appropriate response

Table 19 (2 of 2)

Event	I/O connection object state			
	Non-existent	Configuring	Established	Timed out
Get_attribute_single	As described in IEC 61158-6-2:2014, 7.2.1	Validate/service the request according to the access rules presented in 5.5.4.4. Return appropriate response	Validate/service the request according to the access rules presented in 5.5.4.4. Return appropriate response	Validate/service the request according to the access rules presented in 5.5.4.4. Return appropriate response
Reset	As described in IEC 61158-6-2:2014, 7.2.1	As described in IEC 61158-6-2:2014, 7.2.1	As described in IEC 61158-6-2:2014, 7.2.1	As described in IEC 61158-6-2:2014, 7.2.1
Apply_attributes		As described in IEC 61158-6-2:2014, 7.2.1 ^b		
Receive_data				
Send_message				
Inactivity/watchdog timer expires				

^a Whenever a connection object within the predefined master/slave connection set transitions out of the established state, the entire predefined master/slave connection set may need to be automatically released. When the predefined master/slave connection set is released, all associated connection objects return to the non-existent state. See 5.5.3.3 for more details concerning the automatic release of the predefined master/slave connection set.

^b Since an implied apply_attributes accompanies a set_attribute_single of the expected_packet_rate attribute of a predefined master/slave I/O connection object in the configuring state, the only time an apply_attributes explicit messaging request will succeed is when the expected_packet_rate has yet to be successfully modified via the set_attribute_single request and, thus, still contains the default value of 0.

The state transition diagram (see Figure 46) shows the behaviour of the predefined master/slave explicit messaging connection.

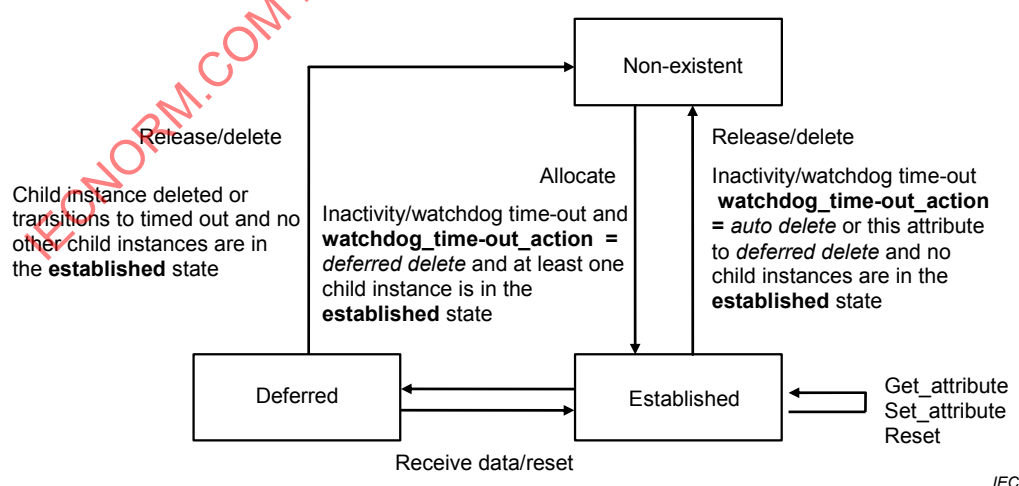


Figure 46 – Predefined master/slave explicit messaging connection state transition diagram

Table 20 provides a detailed state event matrix for the predefined master/slave explicit messaging connection object. This state event matrix inherits from and/or overrides actions presented in the I/O connection object state event matrix in 5.3.

Table 20 – Predefined master/slave explicit messaging connection state event matrix

Event	Explicit messaging connection object state		
	Non-existent	Established	Deferred
DeviceNet object receives an allocate master/slave connection set request that passes all error checks specified in 5.5.3.2. This request specifies the predefined master/slave explicit messaging connection	Create the predefined master/slave explicit messaging connection object for each requested I/O connection and set attributes to default values (see 5.5.3.1). Transition to established	This is an error. See 5.5.3.2	This is an error. See 5.5.3.2
UCMM receives a close request or the connection class receives a delete request and the request specifies the predefined master/slave explicit messaging connection object	As described in IEC 61158-6-2:2014, 7.2.3	As described in IEC 61158-6-2:2014, 7.2.3	As described in IEC 61158-6-2:2014, 7.2.3
DeviceNet object receives a release master/slave connection set request that passes all error checks specified in 5.5.3.2. This request specifies the predefined master/slave explicit messaging connection	This is an error. See 5.5.3.2	Release all associated resources. Transition to non-existent ^a	Release all associated resources. Transition to non-existent ^a
Set_attribute_single	As described in IEC 61158-6-2:2014, 7.2.3	As described in IEC 61158-6-2:2014, 7.2.3	As described in IEC 61158-6-2:2014, 7.2.3
Get_attribute_single			
Reset			
Apply_attributes			
Receive_data			
Send_message			
Inactivity/watchdog timer expires			
^a Whenever a connection object within the predefined master/slave connection set transitions out of the established state, the entire predefined master/slave connection set may need to be automatically released. When the predefined master/slave connection set is released, all associated connection objects return to the non-existent state. See 5.5.3.3 for more details concerning the automatic release of the predefined master/slave connection set.			

5.5.4.4 Connection instance attribute access rules

Subclause IEC 61158-5-2:2014, 6.2.3.2.1.4 gives a general description of the access rules associated with connection objects. Table 21 shows access rules specific to the predefined master/slave I/O connection objects and inherits from and/or overrides rules presented in IEC 61158-5-2:2014, 6.2.3.2.1.4.

Table 21 – Predefined master/slave I/O connection object attribute access

Attribute	I/O connection state			
	Non-existent	Configuring	Established	Timed out
State	As described in IEC 61158-5-2:2014, 6.2.3.2.1.4	As described in IEC 61158-5-2:2014, 6.2.3.2.1.4	As described in IEC 61158-5-2:2014, 6.2.3.2.1.4	As described in IEC 61158-5-2:2014, 6.2.3.2.1.4
Instance_type		Get/set ^a		
Transportclass_trigger		Get only	Get only	Get only
Produced_connection_id		Get only ^c	Get only ^c	Get only ^c
Consumed_connection_id		Get only	Get only	Get only
Initial_comm_characteristics		Get/set ^b	As described in IEC 61158-5-2:2014, 6.2.3.2.1.4	As described in IEC 61158-5-2:2014, 6.2.3.2.1.4
Produced_connection_size		Get/set		
Consumed_connection_size				
Expected_packet_rate				
Watchdog_timeout_action		As described in IEC 61158-5-2:2014, 6.2.3.2.1.4		
Produced_connection_path_length				
Produced_connection_path				
Consumed_connection_path_length				
Consumed_connection_path				
^a The transportclass_trigger attribute of a predefined master/slave I/O connection object within a slave device shall only be modifiable to one of the following values: 0x82 – server/transport class 2, 0x83 – server/transport class 3.				
^b The produced_connection_size attribute of the bit-strobe I/O connection shall not be settable to a value greater than 8.				
^c This attribute shall have Get/set access within the multicast poll connection.				

5.5.5 Master connection object characteristics

This part does not present characteristics with respect to connection objects within a master. The master shall exhibit the external behaviour necessary to interface with its slaves.

5.5.6 Bit-strobe command/response messages

5.5.6.1 General

Bit-strobe command and response messages move small packets of I/O data between a master and its bit-strobed slaves.

5.5.6.2 Bit-strobe command message

The bit-strobe command sends one bit of output data to each slave.

The bit-strobe command message contains a bit string of 64 bits (8 bytes) of output data, one output bit for each possible MAC ID on the link.

A slave device may be designed to do one or all of the following:

- ignore the bit-strobe command;
- consume the bit-strobe command and its output data;
- consume the bit-strobe command as a trigger and ignore the output data.

Slaves shall default to ignoring the bit-strobe command until the bit-strobe connection is allocated and established.

A bit-strobe command message transmitted with no data in the CAN data field is interpreted as a receive_idle event by an application object. A bit-strobe command message that contains data is interpreted as a run event by an application object. The behaviour of an application object upon detection of either the receive_idle or run event is application object specific. See the application object descriptions for more information concerning these events.

5.5.6.3 Bit-strobe response message

The bit-strobe response returns up to eight bytes of input data to the master from each slave.

A bit-strobe response message that contains no data and is configured to contain data indicates a master device no valid bit-strobe data event. The behaviour of the master upon detection of this event is implementation specific.

5.5.6.4 Bit-strobe message characteristics

The master implements the communication resources associated with the bit-strobe command as a client transport class 0 connection to transmit the command and a set of server transport class 0 connection objects to receive the responses.

The slave implements a single server transport class 2 or 3 connection to receive the bit-strobe command and send the associated response.

5.5.7 Poll command/response messages

5.5.7.1 General

The poll command and response messages transfer I/O data between a master and its polled slaves.

5.5.7.2 Poll command message

The poll command sends up to 65 535 bytes of output data (non-fragmented or fragmented) to the destination slave device.

A slave device is permitted to do one or all of the following:

- ignore the poll command if no poll connection is allocated;
- consume the poll command and its output data;
- consume the poll command as a trigger and ignore the output data.

Slaves shall default to ignoring the poll command until the poll connection is allocated and established.

A poll command message transmitted with no data in the CAN data field is interpreted as a receive_idle event by an application object. A poll command message that contains data is interpreted as a run event by an application object. The behaviour of an application object upon detection of the receive_idle or run event is application object specific. See the application object descriptions for more information concerning these events.

5.5.7.3 Poll response message

The poll response returns up to 65 535 bytes (non-fragmented or fragmented) of input data to the master from the slave.

A poll response message that contains no data and is configured to contain data indicates a master device no valid poll data event. The master behaviour upon detection of this event is implementation specific.

5.5.8 Multicast poll command/response messages

5.5.8.1 General

The multicast poll command and response messages transfer I/O data between a master and a group of multicast polled slaves.

5.5.8.2 Multicast poll command message

The multicast poll command sends up to 65 535 bytes of output data (non-fragmented or fragmented) to the destination slave device(s).

A slave device is permitted to do one or all of the following:

- ignore the multicast poll command if no multicast poll connection is allocated;
- consume the multicast poll command and its output data;
- consume the multicast poll command as a trigger and ignore the output data.

Slaves shall default to ignoring the multicast poll command until the multicast poll connection is allocated and established.

A multicast poll command message transmitted with no data in the CAN data field is interpreted as a receive_idle event by an application object. A multicast poll command message that contains data is interpreted as a run event by an application object. The behaviour of an application object upon detection of the receive_idle or run event is application object specific. See the application object descriptions for more information concerning these events.

The multicast poll command message uses a multicast MAC ID rather than a destination MAC ID in the MAC ID field of the CAN identifier. The multicast MAC ID shall be assigned by the master to a value corresponding to the MAC ID value of one of the slave members within the multicast group or itself. When the multicast MAC ID is the master's MAC ID, it may only be used for one multicast group and precludes the master from being a slave in a multicast group to another master.

5.5.8.3 Multicast poll response message

The multicast poll response returns up to 65 535 bytes (non-fragmented or fragmented) of input data to the master from the slave.

A multicast poll response message that contains no data and is configured to contain data indicates a master device no valid multicast poll data event. The master behaviour upon detection of this event is implementation specific.

5.5.9 Change of state/cyclic connections

5.5.9.1 General

The predefined master/slave connection set supports change of state or cyclic data transfer between the master and the slave. This data transfer may be either acknowledged or unacknowledged.

The change of state/cyclic connection sets use connection instance 2 for master to slave data transfer and slave to master acknowledgement. Connection instance 4 is used for slave to master data transfer and master to slave acknowledgement. If a device does not support the poll, and has no support for output data, connection instance 2 need not be created.

To keep the system behaviour consistent, the slave's change of state/cyclic connection (instance 4) shall have the consumed path set to the acknowledge handler object, and the instance of the acknowledge handler object shall be 1.

Because the polled and change of state/cyclic connection sets share connection instance 2, the slave shall follow certain procedures based on the allocation request to ensure the correct behaviour. As described above, when only the change of state/cyclic allocation bit is on, connection instances 2 and 4 are created. Connection instance 4 produces from the default input path and consumes acknowledgements to instance 1 of the acknowledge handler object. Connection instance 2 consumes to the default output path and produces a zero length acknowledgement. When the allocation request contains only the polled allocation bit, connection instance 2 consumes to the default output path and produces from the default input path.

If both the change of state/cyclic and the polled allocation bits are set, connection instance 2 shall behave as though only the polled allocation bit was set, and connection instance 4 shall continue to behave as described above. This is also the required behaviour if the master allocates the poll connection set in one message, and then the change of state/cyclic connection set in a following message.

To achieve unacknowledged data production, the acknowledge suppression bit is set with either the change of state or cyclic bit in the allocation choice byte. Connection instance 2 is configured as a transport class 0 connection for master to slave data production. Connection instance 4 is configured as a transport class 0 connection for slave to master data production. When the poll bit is set in the same allocation choice, this causes connection instance 2 to be configured as the poll, while connection instance 4 continues to be configured as an unacknowledged change of state or cyclic connection.

5.5.9.2 Change of state/cyclic messages trigger acknowledgement

5.5.9.2.1 General

The change of state/cyclic messages move up to 65 535 bytes of data between a master and slave using change of state or cyclic production triggers. Data production may be either acknowledged or unacknowledged.

The following subclauses describe:

- change of state/cyclic message (acknowledged);
- change of state/cyclic message (unacknowledged);
- change of state/cyclic message characteristics;
- change of state/cyclic example application.

5.5.9.2.2 Master's change of state/cyclic message

The master's change of state/cyclic message sends up to 65 535 bytes of data (non-fragmented or fragmented) to the destination slave device. Data production is triggered by either a change of state or transmission trigger time-out.

A change of state/cyclic message transmitted with no data in the CAN data field is interpreted as a receive_idle event by an application object. A change of state/cyclic message that contains data is interpreted as a run event by an application object. The behaviour of an application object upon detection of the run event is application object specific. See the application object descriptions for more information concerning these events.

Slaves shall default to ignoring the master's change of state/cyclic message until the change of state/cyclic connection is allocated and established.

5.5.9.2.3 Slave's change of state/cyclic acknowledge message

The slave's change of state/acknowledge message returns up to 65 535 bytes (non-fragmented or fragmented) of data to the master from the slave. By default, the acknowledge message is zero length.

5.5.9.2.4 Slave's change of state/cyclic message

The slave's change of state/cyclic message sends up to 65 535 bytes of data (non-fragmented or fragmented) to the master from the slave. Data production is triggered by either change of state or transmitted trigger time-out.

A change of state/cyclic message that contains no data and is configured to contain data indicates a master device no valid change of state/cyclic data event. The master behaviour upon detection of this event is implementation specific.

5.5.9.2.5 Master's change of state/cyclic acknowledge message

The master's change of state/cyclic acknowledge message returns up to 65 535 bytes (non-fragmented or fragmented) of data to the slave from the master. By default, the acknowledge message is zero length.

5.5.10 Group 2 only devices

To establish communications with a group 2 only device, a client shall allocate the predefined master/slave connection set. The request to allocate a group 2 only device is transmitted as a group 2 only unconnected explicit request. Instead of using the UCMM to establish explicit messaging connections, group 2 only devices receive and process group 2 only unconnected explicit request messages.

The only services that are valid when transmitted as group 2 only unconnected explicit request messages are:

- allocate_master/slave_connection_set message;
- release_master/slave_connection_set message.

These services are described in 5.5.3.

Responses to group 2 only unconnected explicit requests are returned by transmitting a group 2 message whose MAC ID component is set to the responding device's MAC ID (source MAC ID) and whose message ID is set to 3.

Group 2 only devices shall use the slave's explicit response message identifier field only for transmitting group 2 only unconnected response messages and predefined master/slave explicit messaging connection response messages.

If a group 2 only server receives a group 2 only unconnected request message that is not an allocate_master/slave_connection_set or release_master/slave_connection_set request, then an error response shall be returned whose general error code is set to 02, with the additional error code set to a DeviceNet object specific value of 0x03 (see 5.5.3.5).

5.6 CIP Safety™ on DeviceNet

5.6.1 General

CIP Safety™⁴ on DeviceNet uses a dedicated safety communication layer on top of the application layer of the base DeviceNet protocol: CIP Safety on DeviceNet utilizes the base DeviceNet communication objects and services in combination with specific safety related objects and services, defined as part of the functional safety communication profile CIP Safety. CIP Safety is fully specified in IEC 61784-3-2, including specific adaptations for DeviceNet.

When implemented as part of a safety-related system according to IEC 61508, CIP Safety on DeviceNet provides the necessary confidence in the transportation of information between two or more CDIs, or confidence in the safe behaviour in case of CDI failures. It can be used for applications requiring functional safety up to the Safety Integrity Level (SIL) specified in IEC 61784-3-2.

NOTE The resulting SIL claim of a system depends on the implementation of the selected functional safety communication profile within this system – implementation of a functional safety communication profile according to this part in a standard device is not sufficient to qualify it as a safety device.

5.6.2 Use of CAN identifiers for CIP Safety on DeviceNet

The allocation of CAN identifiers for CIP Safety on DeviceNet is defined in IEC 61784-3-2. The definition is done in such a way to not interfere with the predefined standard identifier usage.

5.7 Physical layer

5.7.1 General

Table 22 shows the general physical layer specifications which shall be met in DeviceNet systems (see 9.2.6 for test specifications).

⁴ CIP Safety™ is a trade name of ODVA, Inc. This information is given for the convenience of users of this International Standard and does not constitute an endorsement by IEC of the trademark holder or any of its products. Compliance to this standard does not require use of the trade name CIP Safety™. Use of the trade name CIP Safety™ requires permission of ODVA, Inc.

Table 22 – General physical layer characteristics

Characteristic	Specification
Bit rates	125 kbit/s, 250 kbit/s, 500 kbit/s
Maximum total length of trunk line and maximum cable length between any two devices	500 m at 125 kbit/s 250 m at 250 kbit/s 100 m at 500 kbit/s
Number of nodes supported by the transceiver	64 minimum
Signalling	In accordance with ISO 11898-1:2003 and ISO 11898-2:2003
Modulation	Baseband
Encoding	NRZ with bit stuffing
Coupling to transmission medium	DC coupled differential Tx/Rx
Isolation (between transceiver circuitry and CAN chip)	500 V d.c. Applies to isolated devices only
Differential input impedance typical (recessive state)	Shunt C = 5 pF Shunt R = 25 k Ω (power on)
Differential input impedance min. (recessive state)	Shunt C = 24 pF Shunt R = 20 k Ω (power on)
Voltage on the power conductors	11 V d.c. to 25 V d.c.
Maximum signal voltage range	-25 V d.c. to +18 V d.c. (CAN_H, CAN_L) ^a

^a Voltages at CAN_H and CAN_L shall be referenced to the transceiver ground. The potential at the transceiver ground may be higher than the V- terminal by an amount equal to the voltage drop across a Schottky diode or equivalent. This voltage shall be 0,6 V maximum with respect to V-.

The physical layer, comprising the transceiver, connector, mis-wiring protection circuitry, voltage regulator, transmission medium and optional isolation shall be as shown in Figure 47. Testing of the robustness of the physical layer implementation within a device is specified in 9.2.9.

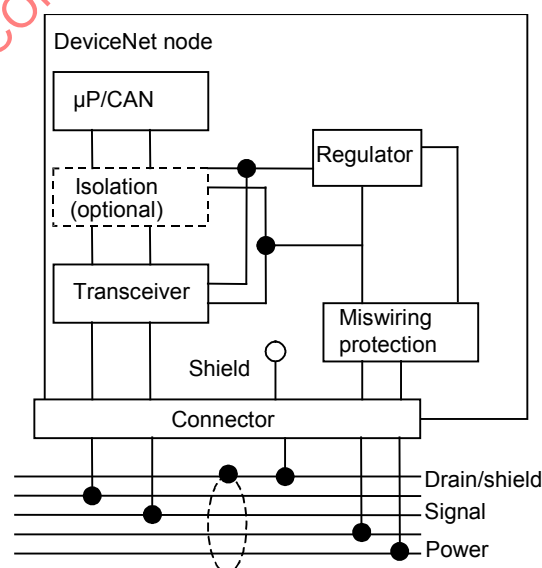


Figure 47 – Physical layer block diagram

5.7.2 Transceiver

A transceiver shall provide transmission and reception of CAN signals to and from the link. The transceiver shall receive differential signals from the link and forwards them to the CAN controller, and shall transmit differential signals from the CAN controller to the link.

To be compatible with the power system design, the transceiver CAN_H and CAN_L terminals shall support a minimum of ± 5 V common-mode operation, i.e. it shall tolerate ground differences of ± 5 V.

An unpowered transceiver may have a lower input impedance than one that is powered. This causes unnecessary network loading and signal attenuation. A powered or unpowered physical layer shall meet the differential input impedance as stated in Table 22.

The required characteristics for the transmitter and the receiver are given in Table 23 and Table 24 (see 9.2.7 for test specifications).

Table 23 – Transmitter characteristics

Transmitter characteristic	Specification
Differential output level (nominal)	2,0 V peak to peak
Differential output level (minimum) (measured at connector, 50 Ω load)	1,5 V peak to peak
Minimum dominant bus voltage at CAN_H	2,75 V ^a
Maximum dominant bus voltage at CAN_H	4,5 V ^a
Minimum dominant bus voltage at CAN_L	0,5 V ^a
Maximum dominant bus voltage at CAN_L	2,0 V ^a
Minimum recessive bus voltage at CAN_H and CAN_L	2,0 V ^a
Maximum recessive bus voltage at CAN_H and CAN_L	3,0 V ^a
Maximum transmitter delay (incl. Isolation)	120 ns
Output short-circuit protection	Internally limited
^a Voltages at CAN_H and CAN_L shall be referenced to the transceiver ground. The potential at the transceiver ground may be higher than the V-terminal by an amount equal to the voltage drop across a Schottky diode or equivalent. This voltage shall be 0,6 V maximum with respect to V-.	

Table 24 – Receiver characteristics

Receiver characteristic	Specification
Differential input voltage dominant	0,95 V min
Differential input voltage recessive	0,45 V max
Hysteresis	150 mV typical
Maximum receiver delay (include isolation)	130 ns
Operating voltage range (CAN_H and CAN_L)	–5 V to +10 V ^a
^a Voltages at CAN_H and CAN_L shall be referenced to the transceiver ground. The potential at the transceiver ground may be higher than the V– terminal by an amount equal to the voltage drop across a Schottky diode or equivalent. This voltage shall be 0,6 V maximum with respect to V–.	

A device shall have an acknowledge delay of less than or equal to 312 ns. The acknowledge delay comprises the propagation delays of the transmitter and receiver and CAN controller. Any combination of these delays is permitted provided that the total time is less than or equal to 312 ns and the maximum transmitter and receiver delays given in Table 23 and Table 24 respectively are not exceeded.

NOTE The maximum transmitter delay time in Table 23 is 120 ns and the maximum receiver delay time in Table 24 is 130 ns. A maximum CAN controller delay time of 62 ns ensures that the acknowledge delay does not exceed 312 ns.

See 9.2.8 for test specifications.

5.7.3 Grounding

To prevent ground loops, the link shall be grounded in only one location, preferably at the power supply. The physical layer circuitry in all devices shall be referenced to V–. A device shall not cause current to flow between V– and ground.

5.7.4 Isolation

5.7.4.1 General

Devices shall be either non-isolated devices or isolated devices.

Isolated devices and non-isolated devices may co-exist and communicate through CDIs.

5.7.4.2 Non-isolated devices

Within a device that contains a non-isolated physical layer (see Figure 48), components ground referenced to V– may connect to other external devices. These external devices shall be ground isolated, and this requirement shall be stated in the manufacturer's documentation.

The shield connection on the DeviceNet connector should be connected through a parallel RC circuit ($R = 1 \text{ M}\Omega$, $C = 0,01 \text{ }\mu\text{F}$ (500 V)) to the device enclosure.

NOTE Best electromagnetic conformance can be achieved with the conductor kept very short along this path and the enclosure being a closed structure of conductive material. If the device does not have such an enclosure, the shield pin of the connector can be left unconnected.

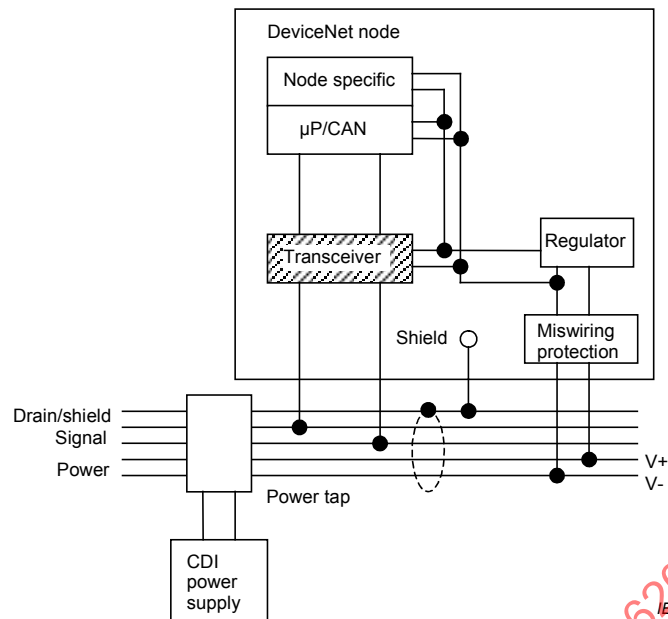


Figure 48 – Device containing a non-isolated physical layer

5.7.4.3 Isolated device

These devices shall obtain power from the link for the transceiver and isolation circuitry (see Figure 49). Link power may be used for other circuitry provided that this link powered circuitry is ground referenced to V– or is otherwise ground isolated.

For an isolated device, the CAN controller may still be active during a de-energized power line condition. Isolated devices shall be capable of sensing when their transceivers have become de-energized.

Within an isolated device, components ground referenced to V– may connect to other external devices through serial ports, parallel ports, or I/O connections. These external devices shall be ground isolated and this requirement shall be stated in the manufacturer's documentation.

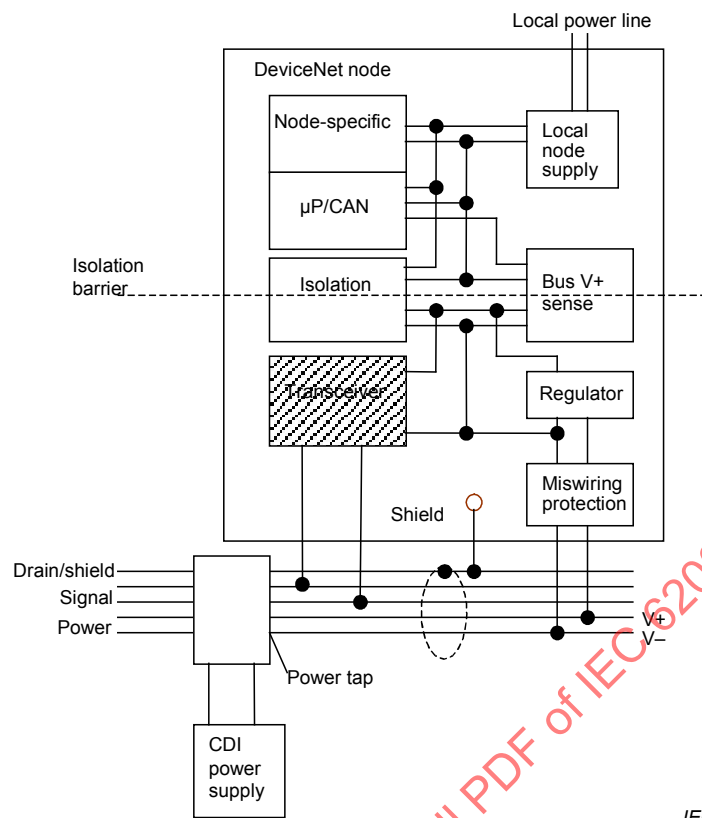


Figure 49 – Device containing an isolated physical layer

The shield connection of the link should be connected via a parallel RC circuit to the device enclosure.

5.7.5 Transmission medium

DeviceNet uses two conductor pairs within one cable. One of these pairs provides a differential communication medium, and the other pair provides power to the devices.

DeviceNet provides two major types of cable: round shielded (e.g. thick cable and thin cable) and flat unshielded (e.g. flat cable). Heavier cables allow longer trunk line distances and more sturdy trunk lines with higher current capability. Thinner cables provide easier routing and termination of either trunk lines or drop lines. Both thick and thin cable may be used for trunk lines and/or drop lines in any combination. Flat cable shall be used only for trunk lines.

5.7.6 Topology

5.7.6.1 General

The DeviceNet medium shall have a linear topology (see Figure 50). If used, drop lines may allow one or more nodes to be attached. Branching structures are only allowed on the drop line.

Terminating resistors shall be connected at each end of the trunk line.

The cable distance between any two points in the cable system shall not exceed the maximum cable distance allowed for the bit rate. The cable distance between two points includes both trunk line cable length and drop line cable length that exists between the two points.

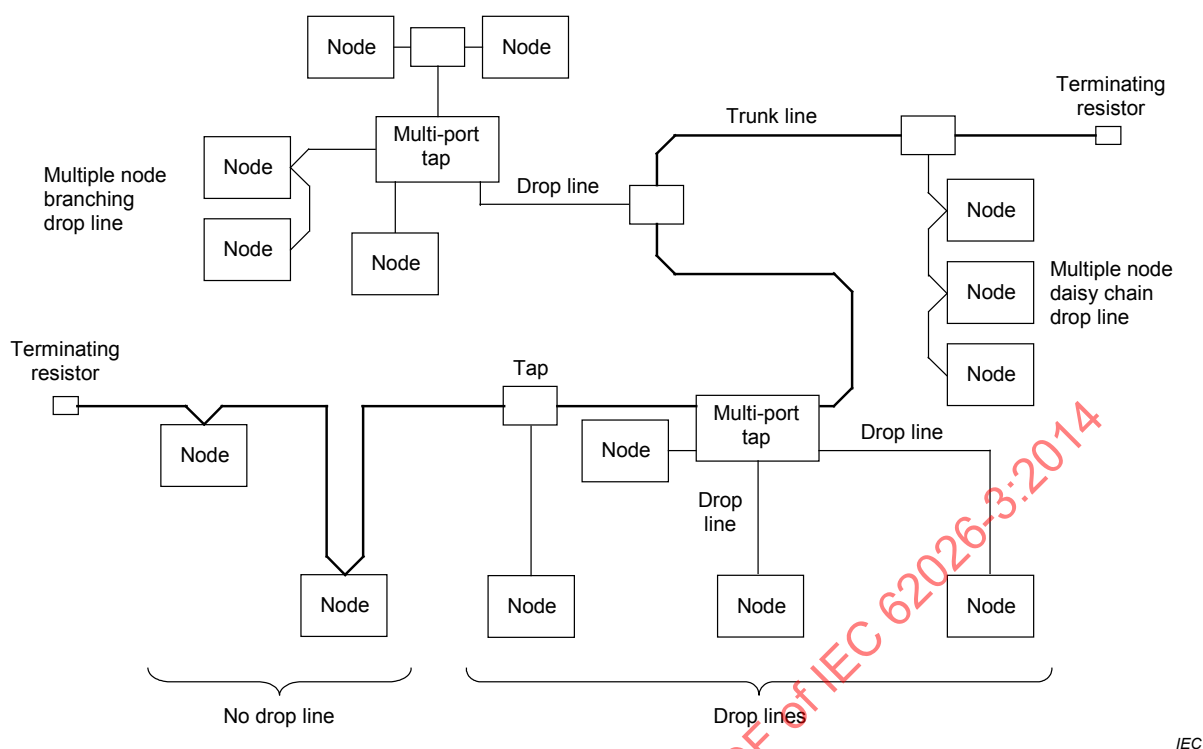


Figure 50 – DeviceNet medium topology

5.7.6.2 Trunk lines

The total length of trunk line allowable on the network depends upon the bit rate and the type of cable used.

For trunk lines constructed of only one type of cable, cable profiles in 8.2 specify the maximum cable distance based on the bit rate and the type of cable used.

DeviceNet allows mixing of different types of cables in a trunk system. The cable profiles for the respective cable types in 8.2 detail the equivalencies when mixing different types of cables in trunk lines.

5.7.6.3 Drop lines

The drop line length is the longest cable distance of those measured from the tap on the trunk line to each of the transceivers of the nodes on the drop line. This distance includes any drop line cable, which might be permanently attached to the device, and shall not exceed 6 m. The total length of drop line allowable on the link depends upon the bit rate and shall not exceed the values specified in the respective cable profiles in 8.2.

5.7.7 Link power

5.7.7.1 Power configuration

The link power shall be supplied by a rated 24 V d.c. source and can support up to 8 A on any section of thick cable trunk line or up to 3 A on any section of thin cable trunk line. Multiple power supplies can be used.

5.7.7.2 Load limits

The maximum link current shall be determined using the information shown in Table 25 (see 9.2.2 for test specifications).

Table 25 – Load limits

Characteristic	Specification
Maximum voltage drop on V– and V+	5 V each
Maximum thick cable trunk line current	8 A
Maximum thin cable trunk line current	3 A
Maximum drop line current range	0,75 A to 3,0 A ^a
Voltage range at each node	11 V to 25 V
Operating current of each device ^b	specified by manufacturer
^a Maximum drop line current depends upon the length of the drop (see below). ^b The operating current represents the average current drawn from the link. Peak operating current shall be specified if it exceeds the average current by more than 10 %.	

The maximum drop line current is also constrained by the following equation:

$$i = 4,57 / l$$

where

i is the maximum allowable drop line current (A);

l is the drop length (m).

6 Product information

In accordance with IEC 62026-1.

7 Normal service, mounting and transport conditions

7.1 Normal service conditions

7.1.1 General

Components of a DeviceNet CDI shall be capable of operating under the following conditions:

If the conditions for operation differ from those given in this part, the user should state the deviation from the standard conditions and consult the manufacturer on the suitability for use under such conditions.

7.1.2 Ambient air temperature

7.1.2.1 Thick cable

The cable shall operate normally within an ambient temperature range of –20 °C to +60 °C when carrying a current of 8 A. This current rating shall be derated linearly to zero at 80 °C.

7.1.2.2 Thin cable

The cable shall operate normally within an ambient temperature range of –20 °C to +70 °C when carrying a current of 3 A. This current rating shall be derated linearly to zero at 80 °C.

7.1.2.3 Flat cable

The cable shall operate normally within an ambient temperature range of –25 °C to +75 °C when carrying a current of 8 A. This current rating shall be derated linearly to zero at 80 °C.

7.1.2.4 Device taps

Device taps shall operate normally within an ambient temperature range of $-40\text{ }^{\circ}\text{C}$ to $+70\text{ }^{\circ}\text{C}$ when carrying full current. This current rating shall be derated linearly to zero at $80\text{ }^{\circ}\text{C}$. Maximum continuous current on power conductors is 3 A for micro connectors and 8 A for other connectors unless otherwise specified.

7.1.2.5 Power taps

Power taps shall operate normally within an ambient temperature range of $-40\text{ }^{\circ}\text{C}$ to $+70\text{ }^{\circ}\text{C}$ when carrying full current. This current rating shall be derated linearly to zero at $80\text{ }^{\circ}\text{C}$.

7.1.2.6 Other CDI components

All other components of a DeviceNet CDI shall operate between the ambient temperatures of $-25\text{ }^{\circ}\text{C}$ to $+70\text{ }^{\circ}\text{C}$ if not otherwise defined, for example in conjunction with a specific actuator or sensor type. The operating characteristics shall be maintained over the permissible range of ambient temperature.

7.1.3 Altitude

DeviceNet components shall be capable of operating at altitude in accordance with IEC 62026-1.

7.1.4 Climatic conditions

7.1.4.1 Humidity

DeviceNet components shall be capable of operating at $40\text{ }^{\circ}\text{C}$ with the relative humidity of the air not exceeding 95 %.

7.1.4.2 Pollution degree

DeviceNet components shall be capable of operating in polluted conditions in accordance with IEC 62026-1.

7.2 Conditions during transport and storage

A special agreement shall be made between the user and the manufacturer if the conditions during transport and storage differ from the following:

- humidity: relative humidity of the air not exceeding 95 % at $40\text{ }^{\circ}\text{C}$;
- temperature: $-40\text{ }^{\circ}\text{C}$ to $+85\text{ }^{\circ}\text{C}$.

7.3 Mounting

DeviceNet components shall be mounted in accordance with IEC 62026-1.

8 Constructional and performance requirements

8.1 Indicators and configuration switches

Indicators and configuration switches for DeviceNet are specified in IEC 61158-4-2:2014, Annex A. Both common and CDI specific requirements, option (3), apply.

DeviceNet Safety devices have additional requirements, see IEC 61784-3-2.

8.2 DeviceNet cable

8.2.1 Overview

This subclause includes specifications of the following cable profiles:

- thick cable;
- thin cable;
- flat cable.

8.2.2 Cable profile template

A cable profile defines the data pair specifications, d.c. power pair specifications, general specifications, topology and the physical configuration for the cable. The orientation of the data and power pairs is a requirement of the specification. Table 26, Table 27, Table 28 and Table 29 define the minimum fields that shall be defined for a DeviceNet cable profile.

Table 26 – Cable profile: data pair specifications

Physical characteristic	Specification
Conductor pair size	<size> <material>; <#> strands
Insulation diameter	<size>
Colours – (CAN_H, CAN_L)	
Pair twist	<#> / <distance>
Tape shield over pair	<material>
Electrical characteristic	Specification
Impedance	120 $\Omega \pm 10\%$ (at 500 kHz)
Propagation delay	<#> ns/m (maximum)
Capacitance between conductors	<#> pF/m at <#> kHz (maximum)
Capacitance between one conductor and other conductor connected to shield	<#> pF/m at <#> kHz (maximum)
Capacitive unbalance	<#> pF/m at <#> kHz (maximum) ASTM D4566-94
DCR – at 20 °C	<#> $\Omega/1\ 000\ m$ (maximum)
Attenuation	<#> dB/100 m at 125 kHz (maximum) <#> dB/100 m at 250 kHz (maximum) <#> dB/100 m at 500 kHz (maximum)

Table 27 – Cable profile: DC power pair specifications

Physical characteristic	Specification
Conductor pair size	<size> <material>; <#> strands
Insulation diameter	<size>
Colours – (V+, V-)	
Pair twist	<#>/ <distance>
Tape shield over pair	<material>
Electrical characteristic	Specification
DCR – at 20 °C	<#> $\Omega/1\ 000\ m$ (maximum)

Table 28 – Cable profile: general specifications

Physical characteristic	Specification
Geometry	
Overall braid shield	<#> % coverage, <#> <material>
Drain wire	<#> <material>; <#> strands
Outside diameter	<size> minimum to <size> maximum
Roundness	Radius delta to be <#> % of O.D.
Jacket marking	Vendor name, part # and additional markings
Electrical characteristic	Specification
DCR – (braid+tape+drain) at 20 °C	<#> Ω/1 000 m (maximum)
Applicable environmental characteristic	Specification
Agency certifications	
Flexure	<#> cycles at bend radius, <#> degrees, <#> pull force, <#> cycles/minute, <method>
Bend radius	<#> × diameter (installation) / <#> × diameter (fixed) <method>
Operating ambient temperature	<#> °C to <#> °C
Storage temperature	<#> °C to <#> °C
Pull tension	<#> N
Connector compatibility	<Open, Mini, Micro.....,>
Topology compatibility	<Trunk, Drop, ...>
Unique characteristic	Application specific

Table 29 – Cable profile: topology

Data rate	Maximum cable distance	Trunk exchange (thick cable)	Cumulative drop	Maximum drop
125 kbit/s	<#> m	<#>	<#> m	<#> m
250 kbit/s	<#> m	<#>	<#> m	<#> m
500 kbit/s	<#> m	<#>	<#> m	<#> m

Important: minimum and maximum lengths may be affected by connector DCR, therefore when defining maximum or minimum lengths of a new cable, the connector DCR shall be considered.

8.2.3 Thick cable profile

Included are the following specifications regarding thick cable:

- data pair specifications, see Table 30;
- power pair specifications, see Table 31;
- general specifications, see Table 32;
- topology, see Table 33;
- physical configuration, see Figure 51;
- available bus current, see Table 34 and Figure 52.

Table 30 – Thick cable: data pair specifications

Physical characteristic	Specification
Conductor pair size	0,823 mm ² copper (individually tinned)
Insulation diameter	3,81 mm (nominal)
Colours	Light blue, white
Pair twist	10/m (approximately)
Tape shield over pair	0,05 mm / 0,025 mm, Al / Mylar Al side out with shorting fold (pull-on applied)
Electrical characteristic	Specification
Impedance	120 Ω $\pm$ 10 % (at 1 MHz)
Propagation delay	4,46 ns/m (maximum)
Capacitance between conductors	39,37 pF/m at 1 kHz (nominal)
Capacitance between one conductor and other conductor connected to shield.	78,74 pF/m at 1 kHz (nominal)
Capacitive unbalance	3 937 pF / 1 000 m at 1 kHz (nominal)
DCR – at 20 °C	22,60 Ω / 1 000 m (maximum)
Attenuation:	0,43 dB / 100 m at 125 kHz (maximum) 0,82 dB / 100 m at 500 kHz (maximum) 1,31 dB / 100 m at 1 MHz (maximum)

Table 31 – Thick cable: DC power pair specifications

Physical characteristic	Specification
Conductor pair size	1,65 mm ² copper (individually tinned)
Insulation diameter	2,49 mm (nominal)
Colours	Red, black
Pair twist	10/m (approximately)
Tape shield over pair	0,025 mm / 0,025 mm, Al/Mylar Al side out with shorting fold (pull-on applied)
Electrical characteristic	Specification
DCR – at 20 °C	11,81 Ω / 1 000 m (maximum)

Table 32 – Thick cable: general specifications

Physical characteristic	Specification
Geometry	Two shielded pairs, common axis with drain wire in centre
Overall braid shield	65 % coverage, 0,012 7 mm ² Cu braid minimum (individually tinned)
Drain wire	0,823 mm ² copper; 19 strands minimum (individually tinned)
Outside diameter	10,41 mm (minimum) to 12,45 mm (maximum)
Roundness	Radius delta to be within 15 % of 0,5 O.D.
Jacket marking	Vendor name and part # and additional markings
Electrical characteristic	Specification
DCR – (braid+tape+drain) at 20 °C	5,74 Ω / 1 000 m (nominal)
Applicable environmental characteristic	Specification
Agency certifications (U.S. and Canada)	NEC (UL) type, CL2/CL3 (min.)
Flexure	2 000 cycles at bend radius, 90°, 8,90 N Pull force, 15 cycles per minute, Tic Toc or C track method
Bend radius	20 × diameter (installation) / 7 × diameter (fixed)
Operating ambient temperature	–20 °C to +60 °C at 8 A; derate current linearly to zero at 80 °C
Storage temperature	–40 °C to +85 °C
Pull tension	845 N (maximum)
Connector compatibility	Mini, Open
Topology compatibility	Trunk, Drop

Table 33 – Thick cable: topology

Data rate	Maximum cable distance	Trunk exchange (thick cable)	Cumulative drop	Maximum drop
125 kbit/s	500 m	1,0	156 m	6 m
250 kbit/s	250 m	1,0	78 m	6 m
500 kbit/s	100 m	1,0	39 m	6 m

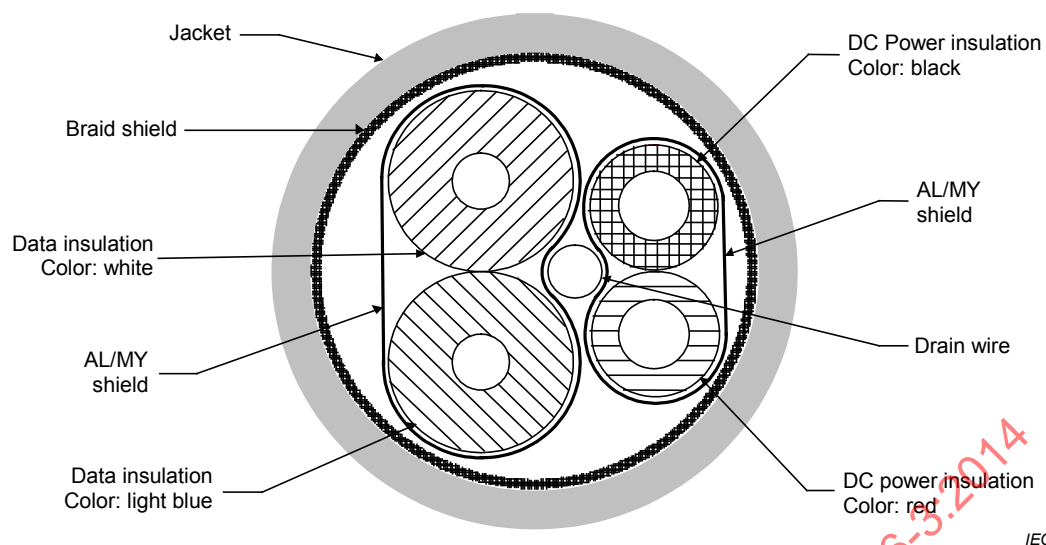


Figure 51 – Thick cable: physical configuration

Table 34 – Thick cable: maximum current available (A) based on network length

Network length m	0	25	50	100	150	200	250	300	350	400	450	500
Maximum current A	8,00	8,00	5,42	2,93	2,01	1,53	1,23	1,03	0,89	0,78	0,69	0,63

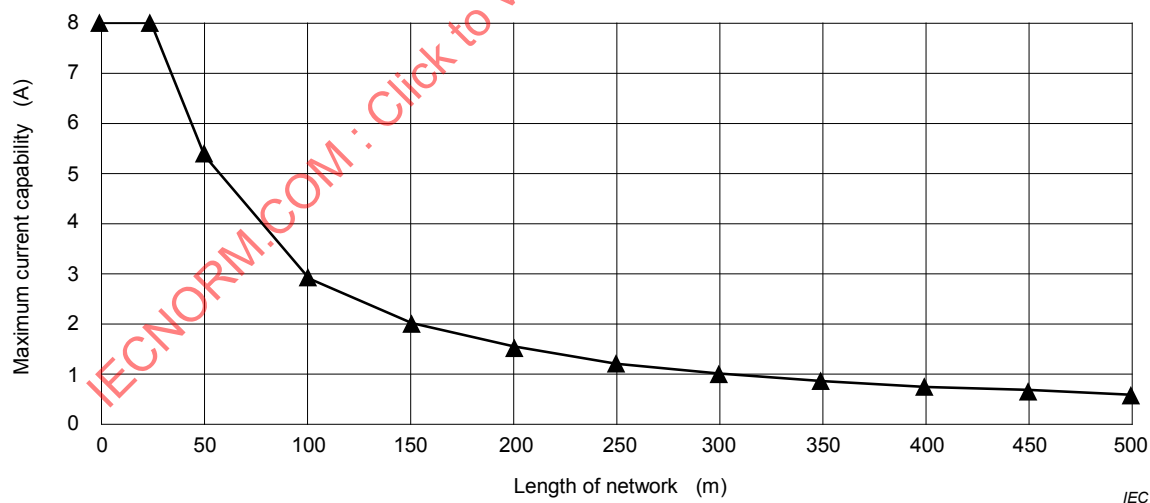


Figure 52 – Thick cable: current available on the DeviceNet power bus

Table 34 and Figure 52 are computed by using the formula:

$$I = 4,65 \text{ V} / ((\text{cable DCR} \times \text{length of network}) + (\text{contact DCR} \times \text{number of contacts}))$$

where

- I is the allowable trunk line current;
- cable DCR = 0,014 6 Ω/m ;

- contact DCR = 0,001 Ω ; and
- number of contacts = 128 (since each tap has two contacts in series).

The cable DCR is determined using an ambient of 80 °C, and temperature coefficient of 0,003 93 per °C.

8.2.4 Thin cable profile

Included are the following specifications regarding thin cable:

- data pair specifications, see Table 35;
- power pair specifications, see Table 36;
- general specifications, see Table 37;
- topology, see Table 38;
- physical configuration, see Figure 53;
- available bus current, see Table 39 and Figure 54.

Table 35 – Thin cable: data pair specifications

Physical characteristic	Specification
Conductor pair size	0,205 mm ² copper (minimum); 19 strands minimum (individually tinned)
Insulation diameter	1,96 mm (nominal)
Colours	Light blue, white
Pair twist	16,4/m (approximately)
Tape shield over pair	0,025 mm / 0,025 mm, Al / Mylar Al side out with shorting fold (pull-on applied)
Electrical characteristic	Specification
Impedance	120 $\Omega \pm 10$ % (at 1 MHz)
Propagation delay	4,46 ns/m (maximum)
Capacitance between conductors	39,37 pF/m at 1 kHz (nominal)
Capacitance between one conductor and other conductor connected to shield	78,74 pF/m at 1 kHz (nominal)
Capacitive unbalance	3 937 pF / 1 000 m at 1 kHz (maximum)
DCR – at 20 °C	91,9 Ω / 1 000 m (maximum)
Attenuation	0,95 dB / 100 m at 125 kHz (maximum) 1,6 dB / 100 m at 500 kHz (maximum) 2,3 dB / 100 m at 1 MHz (maximum)

Table 36 – Thin cable: DC power pair specifications

Physical characteristic	Specification
Conductor pair size	0,326 mm ² (minimum); 19 strands minimum (individually tinned)
Insulation diameter	1,4 mm (nominal)
Colours	Red, black
Pair twist	16,4/m (approximately)
Tape shield over pair	0,025 mm / 0,025 mm, Al/Mylar Al side out with shorting fold (pull-on applied)
Electrical characteristic	Specification
DCR – at 20 °C	57,4 Ω / 1 000 m (maximum)

Table 37 – Thin cable: general specifications

Physical characteristic	Specification
Geometry	Two shielded pairs, common axis with drain wire in centre
Overall braid shield	65 % coverage, 0,012 7 mm ² tinned Cu braid minimum (individually tinned)
Drain wire	0,326 mm ² copper; 19 strands minimum (individually tinned)
Outside diameter	6,1 mm (minimum) to 7,1 mm (maximum)
Roundness	Radius delta to be ±20 % of 0,5 O.D.
Jacket marking	Vendor name and part # and additional markings
Electrical characteristic	Specification
DCR – (braid+tape+drain) at 20 °C	10,5 Ω / 1 000 m (nominal)
Applicable environmental characteristic	Specification
Agency certifications (U.S. and Canada)	NEC (UL) type CL2 (min.)
Flexure	2 000 cycles at bend radius, 90°, 8,90 N Pull force, 15 cycles per minute, Tic Toc or C track method
Bend radius	20 × diameter (installation) / 7 × diameter (fixed)
Operating ambient temperature	–20 °C to +70 °C at 3 A; derate current linearly to zero at 80 °C
Storage temperature	–40 °C to +85 °C
Pull tension	289 N (maximum)
Connector compatibility	Mini, Micro, Open
Topology compatibility	Trunk, Drop

Table 38 – Thin cable: topology

Data rate	Maximum cable distance	Trunk exchange (thick cable)	Cumulative drop	Maximum drop
125 kbit/s	100 m	5,0	156 m	6 m
250 kbit/s	100 m	2,5	78 m	6 m
500 kbit/s	100 m	1,0	39 m	6 m

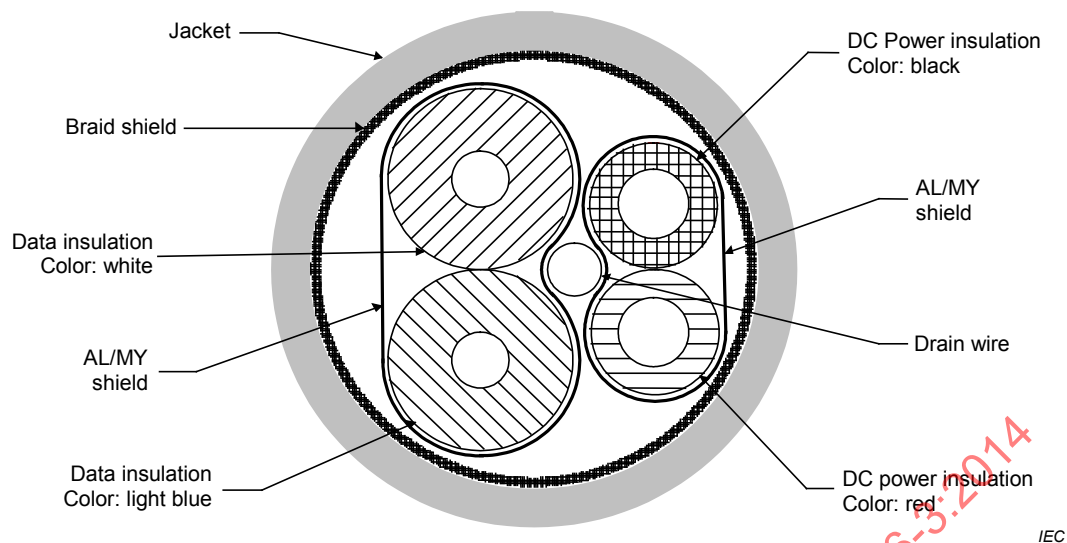


Figure 53 – Thin cable: physical configuration

Table 39 – Thin cable: maximum current available (A) based on network length

Network length m	0	10	20	30	40	50	60	70	80	90	100
Maximum current A	3,00	3,00	3,00	2,06	1,57	1,26	1,06	0,91	0,80	0,71	0,64

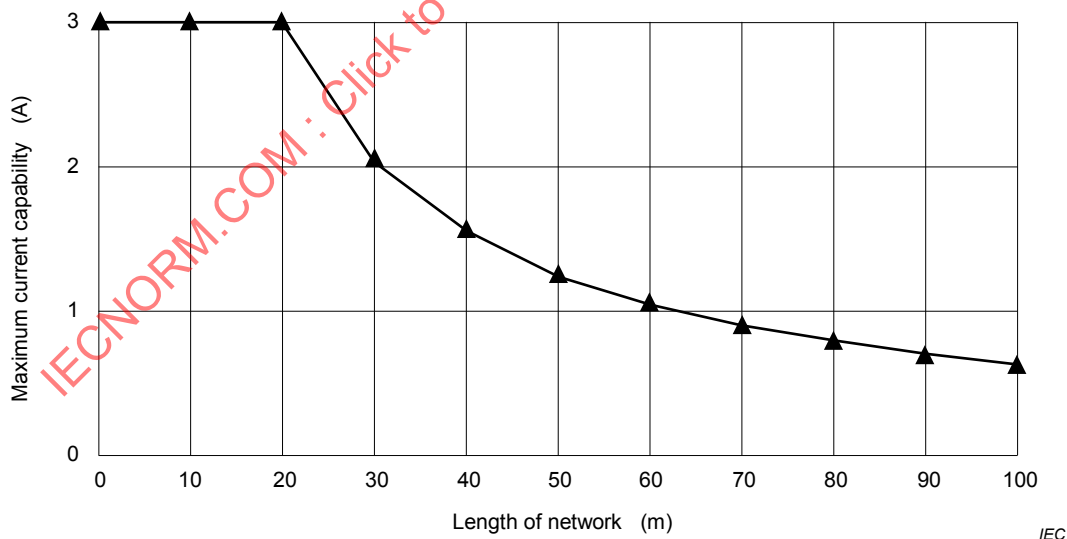


Figure 54 – Thin cable: current available on the DeviceNet power bus

Table 39 and Figure 54 are computed by using the formula:

$$I = 4,65 \text{ V} / ((\text{cable DCR} \times \text{length of network}) + (\text{contact DCR} \times \text{number of contacts}))$$

where

– I is the allowable trunk line current;

- cable DCR = 0,070 9 Ω /m;
- contact DCR = 0,001 Ω ; and
- number of contacts = 128 (since each taps has two contacts in series).

The cable DCR is determined using an ambient of 80 °C, and temperature coefficient of 0,003 93 per °C.

8.2.5 Flat cable profile

Included are the following specifications regarding flat cable:

- data pair specifications, see Table 40;
- power pair specifications, see Table 41;
- general specifications, see Table 42;
- topology, see Table 43;
- physical configuration, see Figure 55;
- available bus current, see Table 44 and Figure 56.

Table 40 – Flat cable: data pair specifications

Physical characteristic	Specification
Conductor pair size	1,31 mm ² copper (minimum); 19 strands minimum (individually tinned)
Insulation diameter	2,79 mm (nominal)
Colours	Light blue, white
Pair twist	N/A
Tape shield over pair	N/A
Electrical characteristic	Specification
Impedance	120 $\Omega \pm 10$ % (at 500 kHz)
Propagation delay	5,25 ns/m (maximum)
Capacitance between conductors	48,23 pF/m at 500 kHz (maximum)
Capacitance between one conductor and other conductor connected to shield	N/A
Capacitive unbalance	3 937 pF / 1 000 m at 500 kHz (maximum) ASTM D4566-94
DCR – at 20 °C	16,1 Ω / 1 000 m (maximum)
Attenuation	0,43 dB / 100 m at 125 kHz (maximum) 0,82 dB / 100 m at 250 kHz (maximum) 1,31 dB / 100 m at 500 kHz (maximum)

Table 41 – Flat cable: DC power pair specifications

Physical characteristic	Specification
Conductor pair size	1,31 mm ² copper (minimum); 19 strands minimum (individually tinned)
Insulation diameter	2,8 mm (nominal)
Colours	Red, black
Pair twist/m	N/A
Tape shield over pair	N/A
Electrical characteristic	Specification
DCR – at 20 °C	16,1 Ω / 1 000 m (maximum)

Table 42 – Flat cable: general specifications

Physical characteristic	Specification
Geometry	N/A
Overall braid shield	N/A
Drain wire	N/A
Outside diameter	See Figure 55
Roundness	N/A
Jacket marking	Vendor name and part # and additional markings
Electrical characteristic	Specification
DCR (braid+tape+drain)	N/A
Applicable environmental characteristic	Specification
Agency certifications (US and Canada)	NEC (UL) type CL2 (minimum)
Flexure	1,0 M cycles at bend radius, 1,83 m min length, 15 cycles per minute, C track method
Bend radius	10 × thickness (installation and fixed)
Operating ambient temperature	–25 °C to +75 °C at 8 A; derate current linearly to zero at 80 °C
Storage temperature	–40 °C to +85 °C
Pull tension	400 N (maximum)
Durometer	95 Shore A (maximum)
Connector compatibility	Flat
Topology compatibility	Trunk

Table 43 – Flat cable: topology

Data rate	Maximum cable distance	Trunk exchange (thick cable)	Cumulative drop	Maximum drop
125 kbit/s	420 m	N/A	156 m	6 m
250 kbit/s	200 m	N/A	78 m	6 m
500 kbit/s	75 m	N/A	39 m	6 m

Dimensions in millimetres

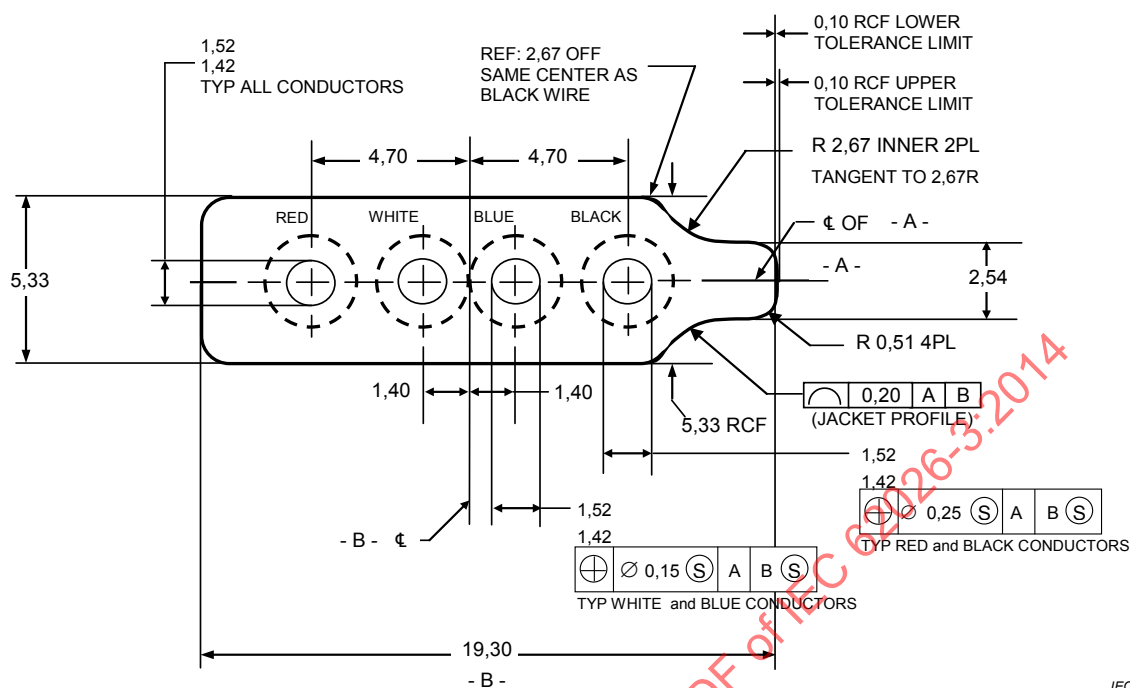


Figure 55 – Flat cable: physical configuration

Table 44 – Flat cable: maximum current available (A) based on network length

Network length m	0	12,5	25	50	100	150	200	250	300	350	400	420
Maximum current A	8,00	8,00	8,00	5,65	2,86	1,91	1,44	1,15	0,96	0,82	0,72	0,69

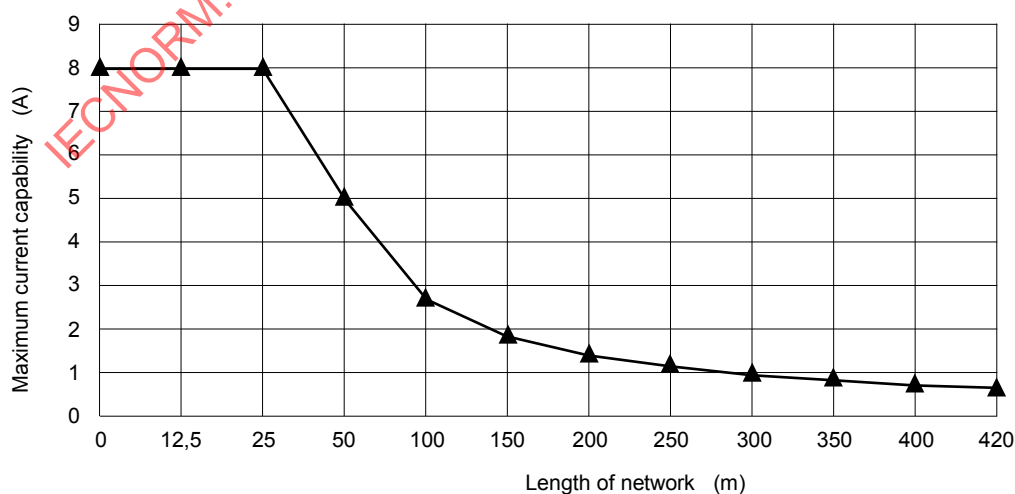


Figure 56 – Flat cable: current available on the DeviceNet power bus

Table 44 and Figure 56 are computed by using the formula:

$$I = 4,65 \text{ V} / ((\text{cable DCR} \times \text{length of network}) + (\text{contact DCR} \times \text{number of contacts}))$$

where

- I is the allowable trunk line current;
- cable DCR = 0,016 Ω /m;
- contact DCR = 0,010 Ω ; and
- number of contacts = 2 (since flat media taps installation does not put contact DCR in series).

The cable DCR is as specified at an ambient of 20 °C.

8.3 Terminating resistors

A 121 Ω , 1 %, 0,25 W metal film resistor shall be connected at each end of the trunk line.

Terminating resistors shall not be included in devices.

8.4 Connectors

8.4.1 General specifications

All connectors shall have five terminals: a signal pair, a power pair, and a shield. Connectors shall be either open or sealed type.

It is recommended that the connector be keyed such that a DeviceNet cable exits the instrument or device without interfering with any indicators, auxiliary connectors or anything else that may require access in the field. The mating DeviceNet receptacle on the instrument or device should be mounted such that the key orientation allows the cable to have no interference with indicators, auxiliary connectors, or anything else that may require access in the field.

Hard-wired connections, such as direct soldering, crimping, screw terminal blocks and barrier strips, are allowed. However, these methods shall support the requirement that the node is removable without severing the trunk.

Any DeviceNet connector shall meet the following minimum requirements:

- all wiping contacts shall be gold plated;
- minimal operating voltage of 25 V;
- nominal contact resistance of less than 1 m Ω and less than 5 m Ω over life.

8.4.2 Connector profile template

A connector profile defines the male and female general specifications, contact specifications, electrical specifications and environmental specifications. Table 45 defines the minimum fields that shall be defined for a DeviceNet connector profile.

Table 45 – Connector profile template

Male general characteristic	Specification
Number of pins	<#>
Coupling nut	<Male/Female/None>
Coupling nut thread	
Rotation	<Required/Optional>
Standard	
Pinout	Drain: Pin #, V+: Pin #, V-: Pin #, CAN_H: Pin #, CAN_L: Pin #
Female general characteristic	Specification
Number of sockets	<#>
Coupling nut	<Male/Female/None>
Coupling nut thread	
Rotation	<Required/Optional>
Standard	
Pinout	Drain: Pin #, V+: Pin #, V-: Pin #, CAN_H: Pin #, CAN_L: Pin #
Physical characteristic	Specification
Wiping contact Plating requirements	
Wiping contact life	<#> insertion-extractions.
Electrical characteristic	Specification
Operating voltage	<#> V minimum
Contact rating	<#> A minimum
Contact resistance	Nominal: less than <#> mΩ Maximum: <#> mΩ over life
Environmental characteristic	Specification
Water resistance	IPXX (according to IEC 60529:1989) and NEMA <#>
Oil resistance	
Operating ambient temperature	<#> °C to <#> °C
Storage temperature	<#> °C to <#> °C

8.4.3 Open connector profile

Table 46 defines the open connector profile; the pin out and the geometry are defined in Figure 57 and Figure 58.

Table 46 – Open connector

Male general characteristic	Specification
Number of pins	5
Coupling nut	None
Coupling nut thread	None
Rotation	None
Standard	See Figure 57 for interface geometry
Pinout	V–: Pin 1, CAN_L: Pin 2, Drain: Pin 3, CAN_H: Pin 4, V+: Pin 5
Female general characteristic	Specification
Number of sockets	5
Coupling nut	None
Coupling nut thread	None
Rotation	None
Standard	See Figure 58 for interface geometry
Pinout	V–: Pin 1, CAN_L: Pin 2, Drain: Pin 3, CAN_H: Pin 4, V+: Pin 5
Physical characteristic	Specification
Wiping contact Plating requirements	0,76 µm gold minimum over 1,3 µm nickel minimum or 0,13 µm gold minimum over 0,51 µm palladium-nickel minimum over 1,3 µm nickel. All gold shall be 24 carat
Wiping contact life	100 Insertion-extractions
Electrical characteristic	Specification
Operating voltage	25 V minimum
Contact rating	8 A minimum
Contact resistance	Nominal: less than 1 mΩ Maximum: 5 mΩ over life
Environmental characteristic	Specification
Water resistance	None
Oil resistance	None
Operating ambient temperature	–40 °C to +70 °C full power with linear derating to 0 A at 80 °C
Storage temperature	–40 °C to +85 °C

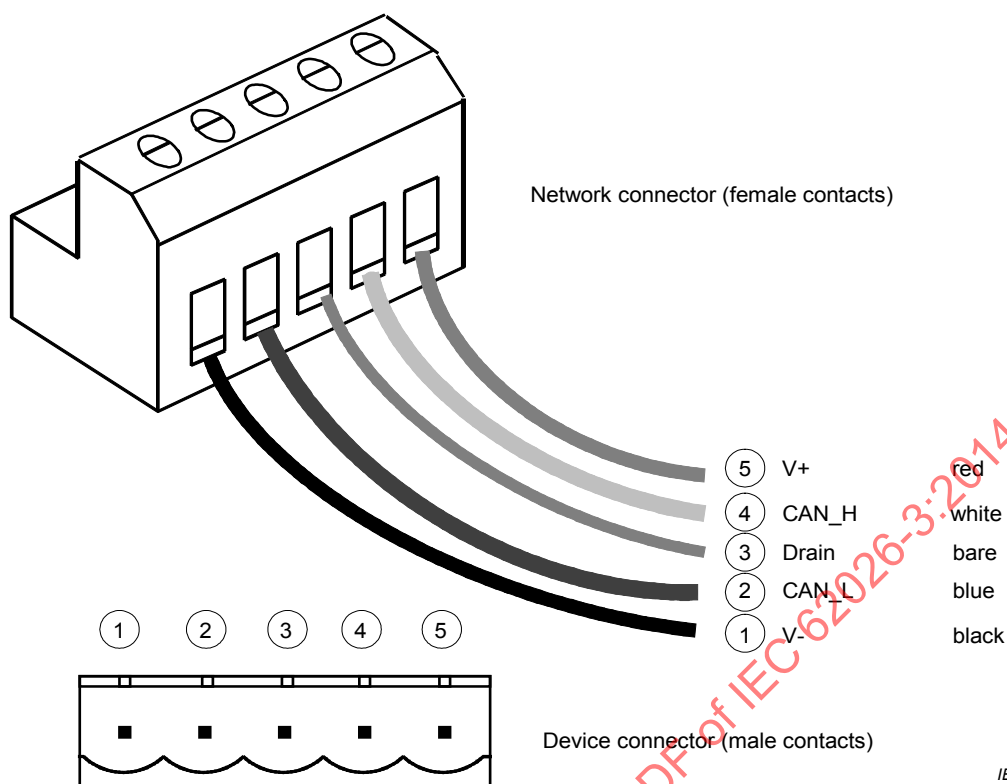


Figure 57 – Open connector pinout

Dimensions in millimetres

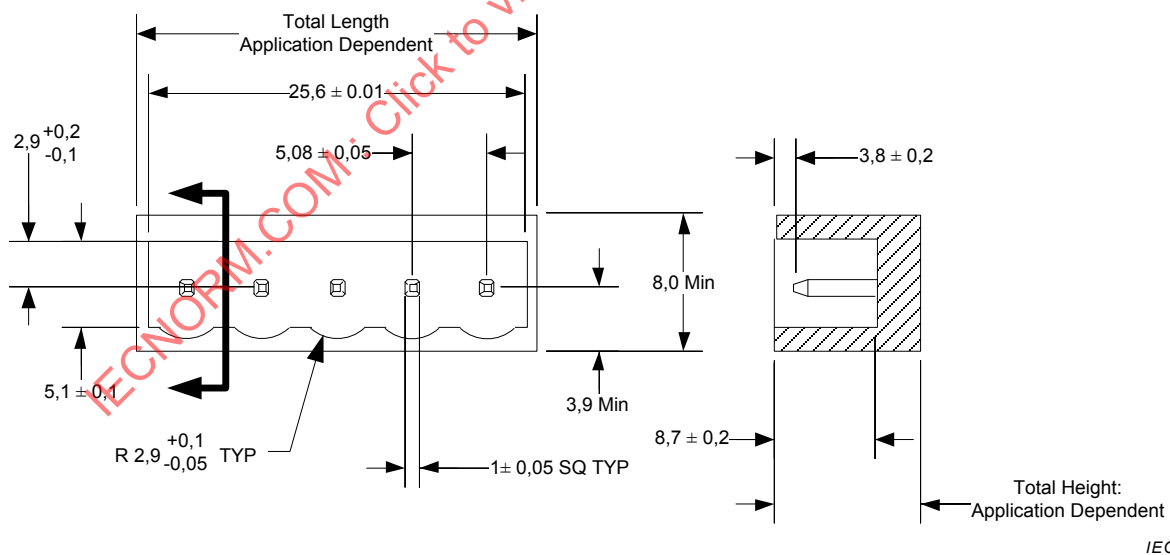


Figure 58 – Open connector geometry

8.4.4 Sealed mini connector profile

Table 47 defines the mini connector profile, the pin out is defined in Figure 59.

Table 47 – Sealed mini connector

Male general characteristic	Specification
Number of pins	5
Coupling nut	Male
Coupling nut thread	7/8-16 UN-2A THD
Rotation	Optional
Standard	ANSI/B93.55M-1981 (R 1988) intermateability requirements
Pinout	Drain: Pin 1, V+: Pin 2, V-: Pin 3, CAN_H: Pin 4, CAN_L: Pin 5
Female general characteristic	Specification
Number of sockets	5
Coupling nut	Female
Coupling nut thread	7/8-16 UN-2B THD
Rotation	Required
Standard	ANSI/B93.55M-1981 (R 1988) intermateability requirements
Pinout	Drain: Pin 1, V+: Pin 2, V-: Pin 3, CAN_H: Pin 4, CAN_L: Pin 5
Physical characteristic	Specification
Wiping contact plating requirements	0,76 µm gold minimum over 1,3 µm nickel minimum or 0,13 µm gold minimum over 0,51 µm palladium-nickel minimum over 1,3 µm nickel. All gold shall be 24 carat
Wiping contact life	1 000 insertion-extractions
Electrical characteristic	Specification
Operating voltage	25 V minimum
Contact rating	8 A minimum
Contact resistance	Nominal: less than 1 mΩ Maximum: 5 mΩ over life
Environmental characteristic	Specification
Water resistance	IP67 (according to IEC 60529:1989) and NEMA 4, 6, 6P, 13
Oil resistance	UL-1277, OIL RES II
Operating ambient temperature	-40 °C to +70 °C full power with linear derating to 0 A at 80 °C
Storage temperature	-40 °C to +85 °C

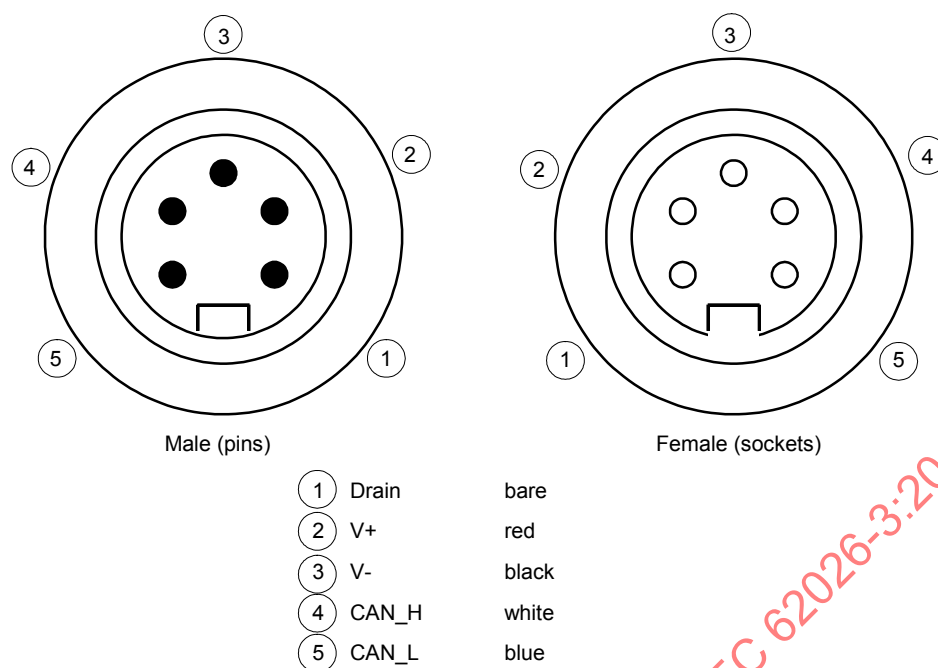
**Figure 59 – Mini connector pinout****8.4.5 Sealed micro connector profile**

Table 48 defines the micro connector profile; the pinout is defined in Figure 60.

Table 48 – Sealed micro connector (1 of 2)

Male general characteristic	Specification
Number of pins	5
Coupling nut	Male
Coupling nut thread	M12 × 1 6G fit
Rotation	Optional
Standard	See IEC 60947-5-2:2007, Amendment 1 (2012), Figure D.2 for intermateability requirements
Pinout	Drain: Pin 1, V+: Pin 2, V-: Pin 3, CAN_H: Pin 4, CAN_L: Pin 5
Female general characteristic	Specification
Number of sockets	5
Coupling nut	Female
Coupling nut thread	M12 × 1 6G fit
Rotation	Required
Standard	IEC 60947-5-2:2007, Amendment 1 (2012), Figure D.2 intermateability requirements
Pinout	Drain: Pin 1, V+: Pin 2, V-: Pin 3, CAN_H: Pin 4, CAN_L: Pin 5
Physical characteristic	Specification
Wiping contact plating requirements	0,76 µm gold minimum over 1,3 µm nickel minimum or 0,13 µm gold minimum over 0,51 µm palladium-nickel minimum over 1,3 µm nickel. All gold shall be 24 carat
Wiping contact life	1 000 insertion-extractions

Table 48 (2 of 2)

Electrical characteristic	Specification
Operating voltage	25 V minimum
Contact rating	3 A minimum
Contact resistance	Nominal: less than 1 mΩ. Maximum: 5 mΩ over life
Environmental characteristic	Specification
Water resistance	IP67 (according to IEC 60529:1989) and NEMA 4, 6, 6P, 13
Oil resistance	UL-1277, OIL RES II
Operating ambient temperature	–40 °C to +70 °C full power with linear derating to 0 A at 80 °C
Storage temperature	–40 °C to +85 °C

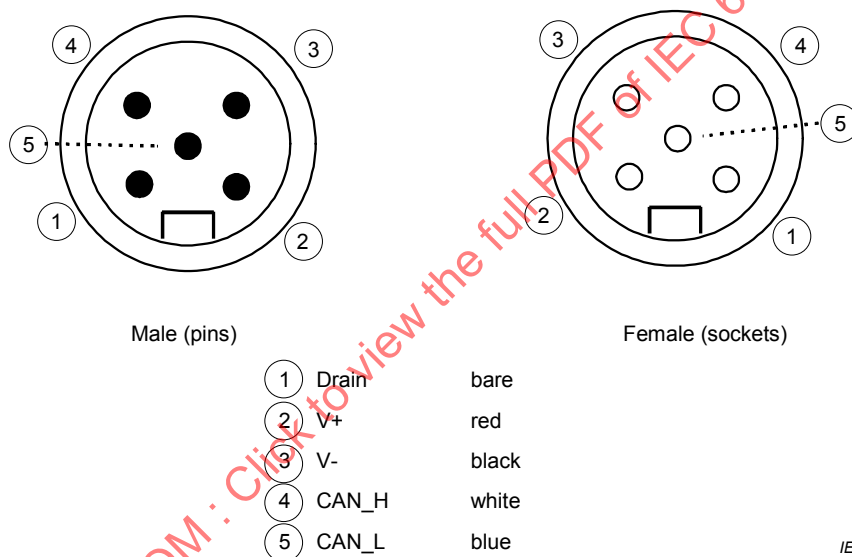


Figure 60 – Micro connector pinout

8.4.6 Flat trunk connector profile

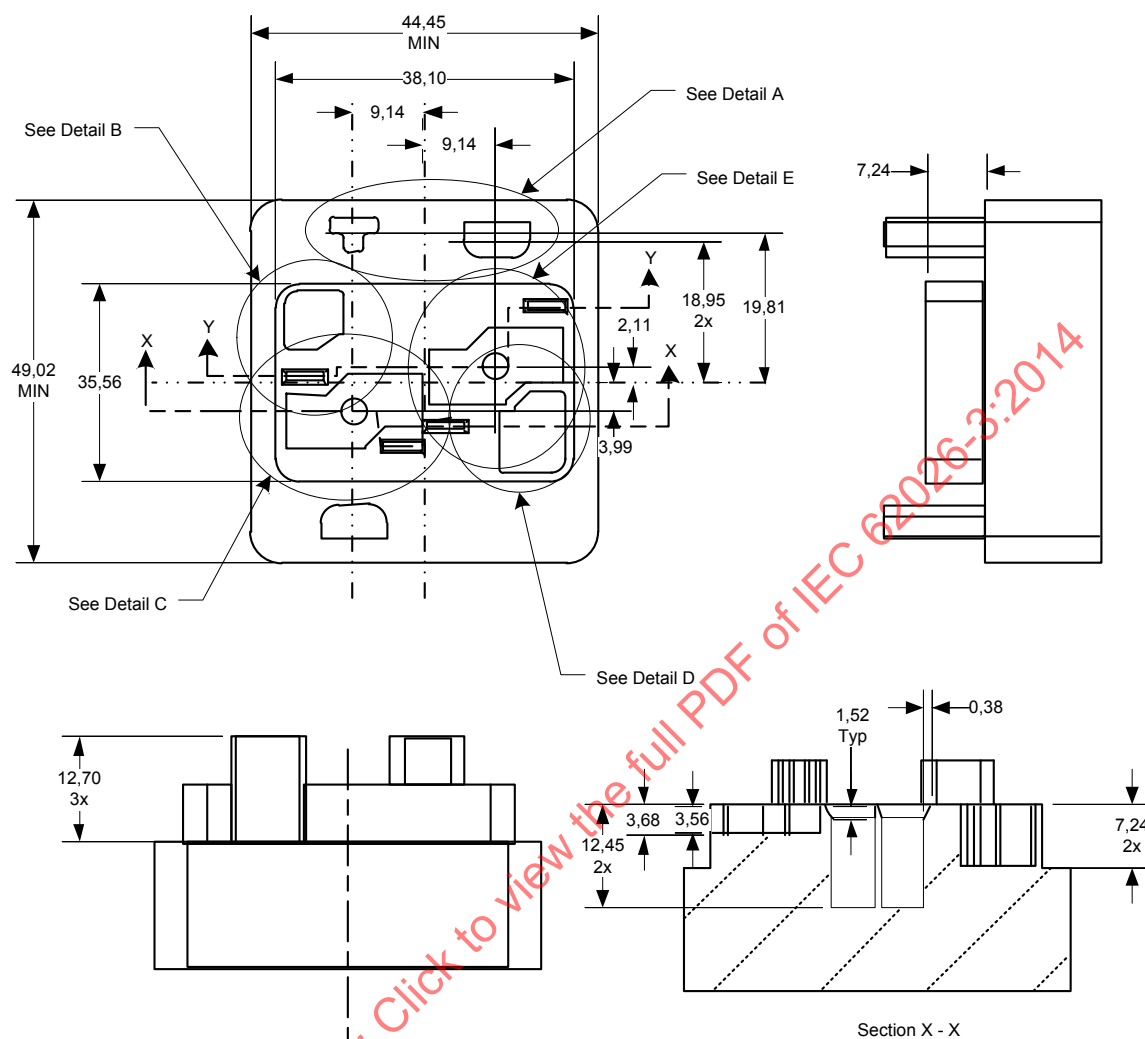
Table 49 defines the flat trunk connector profile; the pin out and the geometry are defined in Figure 61 and Figure 62.

Table 49 – Flat trunk connector

Male general characteristic	Specification
Number of pins	4
Coupling nut	(optional retention with #8 × 1 ¼ screw)
Coupling nut thread	(UNC 32)
Rotation	None
Standard	See Figure 61
Pinout	CAN_L: Pin 1, CAN_H: Pin 2, V+: Pin 3, V-: Pin 4
Female general characteristic	Specification
Number of sockets	4
Coupling nut	(optional retention with #8 insert or nut)
Coupling nut thread	(UNC 32)
Rotation	None
Standard	See Figure 61
Pinout	CAN_L: Pin 1, CAN_H: Pin 2, V+: Pin 3, V-: Pin 4
Physical characteristic	Specification
Wiping contact plating requirements	0,20 µm gold minimum flash over 1,27 µm nickel. All gold shall be 24 carat
Wiping contact life	100 insertion-extractions
Electrical characteristic	Specification
Operating voltage	25 V minimum
Contact rating	3 A minimum
Contact resistance	Nominal: less than 1 mΩ. Maximum: 5 mΩ over life
Capacitance between V+ and V-	0,22 µF ± 10 %
Environmental characteristic	Specification
Water resistance	IP67 (according to IEC 60529:1989) and NEMA 4, 6, 6P, 13
Oil resistance	UL-1277, OIL RES II
Operating ambient temperature	-40 °C to +70 °C full power with linear derating to 0 A at 80 °C
Storage temperature	-40 °C to +85 °C

If a device connects directly to the flat media without a drop cable, this interface can be used to connect to an existing two-piece connector that connects to the flat cable. This interface is not required when using a single piece design that directly connects to the flat cable.

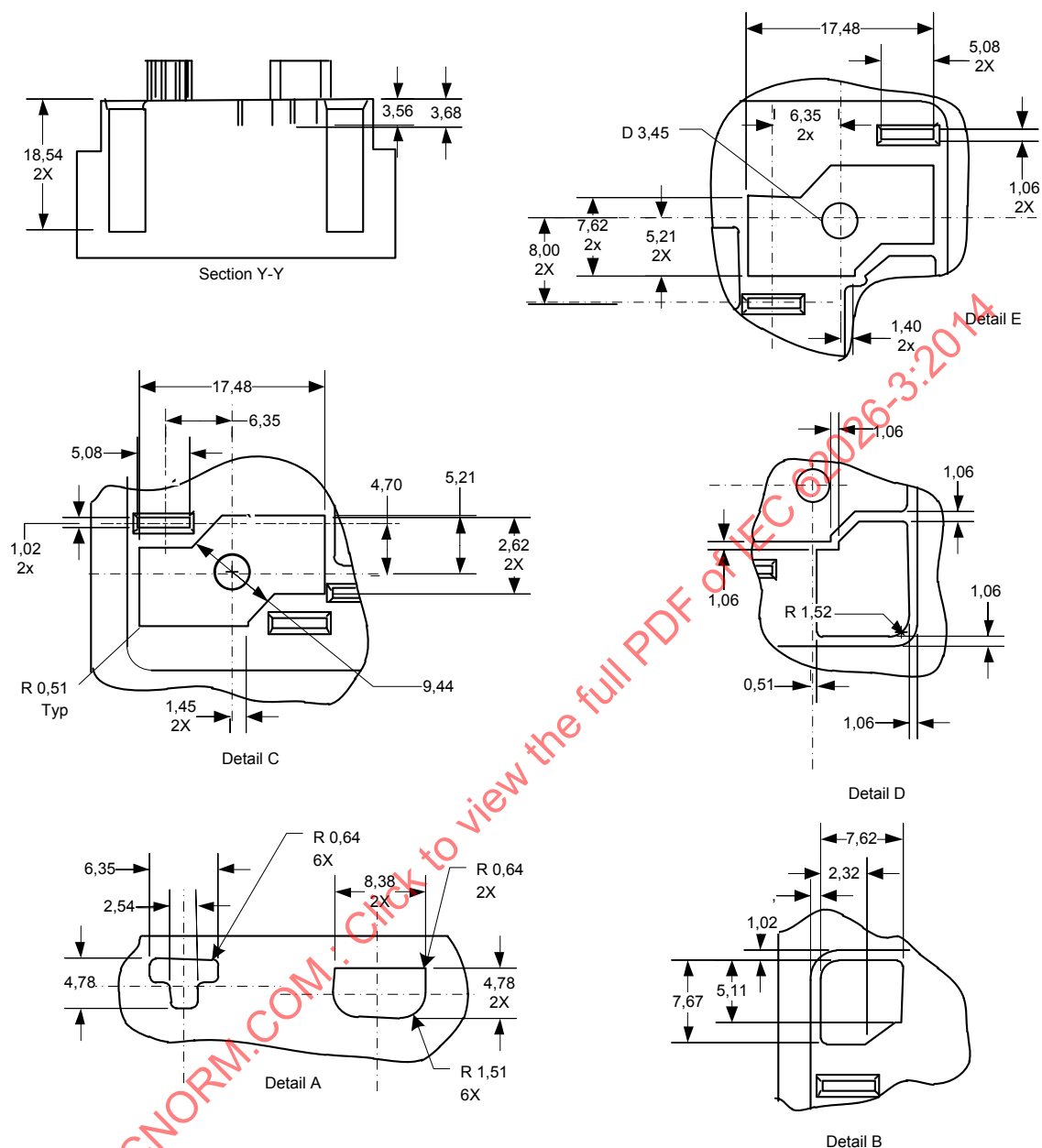
Dimensions in millimetres



IEC

Figure 61 – Flat trunk connector layout – Part 1

Dimensions in millimetres



IEC

Figure 62 – Flat trunk connector layout – Part 2

8.5 Device taps and power taps

8.5.1 Device taps

Device taps provide points of attachment onto the trunk line. Devices may be connected to the link either directly to the device tap or with a drop line. Device taps shall also allow removal of a device without disrupting CDI operation.

Device taps shall have a minimum voltage rating of 25 V d.c. and shall comply with the specifications given in Table 50 and Table 51.

Table 50 – Internal pass-through conductor specifications

Conductor description	Specification
Drain wire conductor	102 mm maximum length of conductor. Maximum resistance 2,3 mΩ
Power conductors V+ and V-	102 mm maximum length of conductor. Maximum resistance 1,2 mΩ. Rated current 8 A
Signal conductors CAN_H and CAN_L	102 mm maximum length of conductor. Maximum resistance 2,3 mΩ

Table 51 – Internal drop conductor specifications

Conductor description	Specification
Drain wire conductor	178 mm maximum length and 10 mΩ maximum resistance, both measured between the pass-through tap connector and any port connection
Power conductors V+ and V-	178 mm maximum length measured between the pass-through tap connector and any port connection. Maximum resistance 10 mΩ for conductors rated up to 3 A, 4 mΩ otherwise, both measured between the pass-through tap connector and any port connection
Signal conductors CAN_H and CAN_L	178 mm maximum length and 10 mΩ maximum resistance, both measured between the pass-through tap connector and any port connection

8.5.2 Power taps

A power tap connects a power supply to the trunk line. When connected to the link, a power tap shall provide continuous connection for the signal, drain and V – wires. A power tap may also provide:

- overcurrent protection in each direction from the tap;
- power supply protection when multiple power supplies are used;
- a connection to the shield/drain wire for grounding the link.

DeviceNet power taps shall have a minimum voltage rating of 25 V d.c. and shall comply with the specifications given in Table 52 and Table 53.

Table 52 – Internal pass-through conductor specifications

Conductor description	Specification
Drain wire conductor	178 mm maximum length of conductor. Maximum resistance 4,1 mΩ
V- power conductor	178 mm maximum length of conductor. Maximum resistance 2,1 mΩ. Rated current 8 A
Signal conductors CAN_H and CAN_L	178 mm maximum length of conductor. Maximum resistance 4,1 mΩ

Table 53 – Internal power drop conductor specifications

Conductor description	Specification
Drain wire conductor to ground terminal	178 mm maximum length and 2,1 mΩ maximum resistance, both measured between any tap connector and the ground terminal
Power conductors V+ and V-	305 mm maximum length and 3,6 mΩ maximum resistance, both measured between any tap connector and the power supply port connection. Wires shall be rated to carry the operating current

Power tap connectors (except the power supply connector) shall meet the requirements given in 8.4.

8.6 Link powered devices

Devices may draw power from the link. Such devices shall use a voltage regulator with an input voltage between 11 V d.c. and 25 V d.c. and incorporate transient protection and filtering.

The voltage regulator shall comply with the specifications given in Table 54.

Table 54 – Voltage regulator specifications

Parameter	Specification										
Input voltage range	11 V d.c. to 25 V d.c.										
Isolation (if required)	500 V										
Power ON delay	Linear regulators: none. Switch mode regulators: <table> <tr> <td>< 100 mA</td> <td>2 ms to 10 ms</td> </tr> <tr> <td>0,1 A – 0,5 A</td> <td>5 ms to 15 ms</td> </tr> <tr> <td>0,5 A – 1 A</td> <td>10 ms to 20 ms</td> </tr> <tr> <td>1 A – 2 A</td> <td>15 ms to 30 ms</td> </tr> <tr> <td>> 2 A</td> <td>20 ms to 40 ms</td> </tr> </table>	< 100 mA	2 ms to 10 ms	0,1 A – 0,5 A	5 ms to 15 ms	0,5 A – 1 A	10 ms to 20 ms	1 A – 2 A	15 ms to 30 ms	> 2 A	20 ms to 40 ms
< 100 mA	2 ms to 10 ms										
0,1 A – 0,5 A	5 ms to 15 ms										
0,5 A – 1 A	10 ms to 20 ms										
1 A – 2 A	15 ms to 30 ms										
> 2 A	20 ms to 40 ms										
Output short-circuit protection	Current limit										
Reverse polarity protection	Rectifier in ground path ^a										
^a If the rectifier is used to power the transceiver, it shall have a forward voltage of no greater than 0,6 V.											

8.7 Miswiring protection

Devices shall be capable of sustaining misconnection without suffering damage under the following conditions:

- V+ and V- conductors interchanged;
- connection of the drain wire to any other conductor;
- connection of CAN_H and/or CAN_L to a voltage not exceeding the range –25 V d.c. to +18 V d.c.

See 9.2.4 and 9.2.5 for test specifications.

8.8 Power supplies

Power supplies which are connected to the link shall comply with the specifications given in Table 55 (see 9.2.1 for test specifications).

Table 55 – DeviceNet power supply specifications

Parameter	Specification
Nominal voltage at 20 °C	24 V d.c. ± 1 % or adjustable to 0,2 %
Line regulation	0,3 % max. over the range specified by the manufacturer
Load regulation	0,3 % max.
Temperature coefficient	0,03 % per K max.
Output ripple	250 mV peak to peak
Load capacitance capability	7 000 μ F max.
Overcurrent protection	100 % to 125 % of rated current
Output voltage rise time (with full load)	250 ms max. to 95 % of final value
Power ON overshoot	0,2 % max.
Stability	0 % to 100 % load (all conditions)
Isolation	Output isolated from input voltage and chassis ground

8.9 Electromagnetic compatibility (EMC)

8.9.1 General

All immunity and emission tests are type tests and shall be carried out under representative conditions, both operational and environmental, using the recommended wiring practices and including all equipment necessary for communication and data transfer on the DeviceNet cable (see 9.2.10 for test specifications).

This requirement shall be met by the use of two devices with ongoing I/O communication and one power supply.

8.9.2 Immunity

8.9.2.1 Performance criteria

The test results are specified using the following performance criteria:

- **criterion A:** The apparatus shall continue to operate as intended during and after the test. Any loss of functionality as shown in Table 56 shall constitute a failure;
- **criterion B:** During the test, no change of actual operating state or stored data is allowed. The apparatus shall continue to operate as intended after the test. Any loss of functionality as shown in Table 56 shall constitute a failure.

Table 56 – Immunity performance criteria

Function type	Criterion A	Criterion B
Masters (processors and adapters)	Program loss Memory faults I/O reset Data table corruption	Program loss Memory faults I/O reset Data table corruption
Any module	Unexpected operation Lockup Operator intervention Damage	Unexpected operation Lockup Operator intervention Damage
External communications	Node off-line	Node off-line
Internal communications: - Radiated, conducted - Fast transient - ESD - Surge	> 1 error bit set / 10 transfers (frames)	> 1 error bit set / 10 transfers (frames) Lockup Lockup

8.9.2.2 Electrostatic discharge (ESD) immunity

DeviceNet CDIs shall meet the requirements specified in 8.2.1 of IEC 62026-1:2007.

8.9.2.3 Radiated radio-frequency electromagnetic field immunity

DeviceNet CDIs shall meet the requirements specified in 8.2.1 of IEC 62026-1:2007.

8.9.2.4 Electrical fast transient/burst immunity

DeviceNet CDIs shall meet the requirements specified in 8.2.1 of IEC 62026-1:2007, with the exception of DC I/O ports which shall be tested at ± 1 kV.

8.9.2.5 Surge immunity

DeviceNet CDIs shall meet the requirements specified in 8.2.1 of IEC 62026-1:2007.

8.9.2.6 Conducted radio-frequency disturbance immunity

DeviceNet CDIs shall meet the requirements specified in 8.2.1 of IEC 62026-1:2007.

8.9.2.7 Voltage dips and interruptions

Conformity of the power supply to the specified limits given in Table 55 eliminates the need for any further testing.

8.9.3 Emissions**8.9.3.1 Radiated emissions**

These shall be in accordance with CISPR 11:2009, Amendment 1 (2010), group 1, class A.

8.9.3.2 Conducted emissions

These shall be in accordance with CISPR 11:2009, Amendment 1 (2010), group 1, class A.

8.10 Additional functional safety requirements related to EMC

DeviceNet Safety CDIs shall meet the additional requirements specified in IEC 61784-3-2.

9 Tests

9.1 General

This clause specifies type tests for electrical and logical requirements. Tests are divided into two parts:

- electrical and EMC testing;
- logical testing (behaviour of a device on DeviceNet).

NOTE If DeviceNet specifications are implemented as an integral part of a product, some logical tests can depend on the behaviour of the product itself. Conformance to this part of the standard does not necessarily mean conformance to a specific standard of the whole product in which DeviceNet is integrated. A test of specific applications of DeviceNet functions is outside the scope of this part of the standard.

The tests applicable to a specific EUT shall be performed in the sequence indicated.

9.2 Electrical and EMC testing

9.2.1 Test of the DeviceNet power supply

9.2.1.1 Test purpose

The following tests verify the power supply requirements that are specific to DeviceNet.

9.2.1.2 Power supply output voltage rise time

9.2.1.2.1 Test circuit

The test circuit shall be as shown in Figure 63. Select the load resistor (R) to ensure that the power supply operates at rated current.

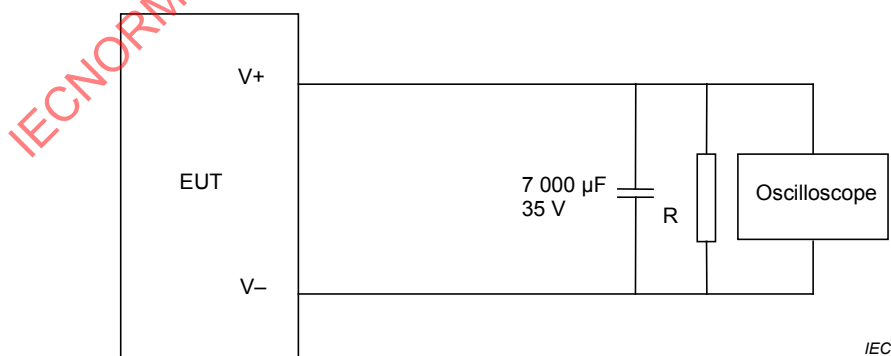


Figure 63 – Power supply rise time test circuit

9.2.1.2.2 Test procedure

The oscilloscope connected across the load resistor shall be set to 24 V, 500 ms full scale. The EUT shall be switched on and the rise time from the start of the voltage rise to 95 % of the power supply final voltage shall be recorded.

9.2.1.2.3 Criteria for compliance

The rise time shall be less than or equal to 250 ms (see Table 55).

9.2.1.3 Power supply ripple

9.2.1.3.1 Test circuit

The test circuit shall be as shown in Figure 63 with the capacitor disconnected.

9.2.1.3.2 Test procedure

The power supply shall be switched on and the a.c. peak to peak ripple of the power supply output voltage measured using the oscilloscope.

9.2.1.3.3 Criteria for compliance

In order to meet the power supply requirements, the peak to peak ripple shall be less than or equal to 250 mV (see Table 55).

9.2.2 Device peak current consumption

9.2.2.1 Test circuit

The test circuit shall be as shown in Figure 64.

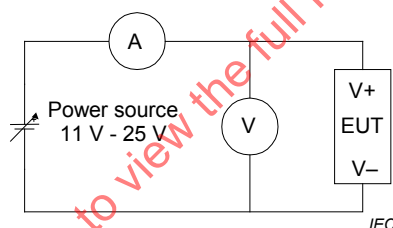


Figure 64 – Current consumption test circuit

9.2.2.2 Test procedure

To determine the operating state that results in maximum current consumption from the link. The EUT shall be operated in this state. If current fluctuations occur, the peak current drawn from the link shall be recorded.

The tests shall be performed at the upper and lower limits of the voltage range specified in 5.7.7.2.

9.2.2.3 Criteria for compliance

To determine the applicable current limit by inspection of the device and/or the manufacturer's documentation. The current readings obtained shall be in accordance with the manufacturer's specified value as outlined in 5.7.7.2 and shall be less than or equal to 8 A.

9.2.3 Power ON behaviour

9.2.3.1 Test purpose

The purpose of this test is to determine that the first transmitted signal is a valid duplicate MAC ID message under the following conditions:

- when added to an existing working CDI;

– when the power is cycled on the CDI.

9.2.3.2 Test circuit

The test circuit shall be as shown in Figure 65.

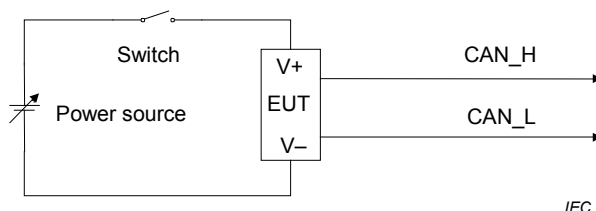


Figure 65 – Power ON test circuit

9.2.3.3 Test procedure

The switch shall be closed and the CAN_H and CAN_L signal lines shall be monitored.

9.2.3.4 Criteria for compliance

The first transmitted signal shall be a valid duplicate MAC ID request.

9.2.4 Reverse connection of V+ and V–

9.2.4.1 Test purpose

The purpose of this test is to verify compliance with requirements for the miswiring protection.

9.2.4.2 Test circuit

The test circuit shall be as shown in Figure 66.

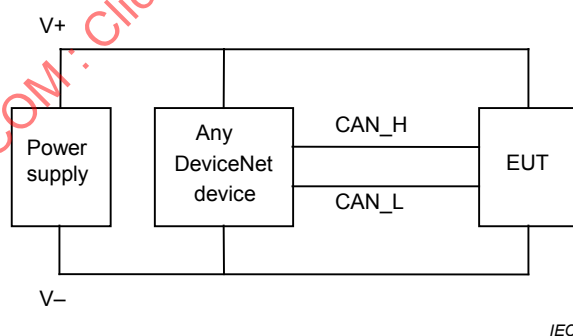


Figure 66 – Test circuit for reversal of V+ and V– and also V– interruption

9.2.4.3 Test procedure

The V+ terminal of the EUT shall be connected to the V– of the power supply and link, and the V– terminal of the EUT shall be connected to the V+ of the power supply and link. CAN_H and CAN_L shall be monitored.

9.2.4.4 Criteria for compliance

It shall be verified that the CAN_H and CAN_L signals remain within the limits specified in Table 23 and Table 24.

9.2.5 Disconnection of V–

9.2.5.1 Test purpose

The purpose of this test is to verify compliance with requirements for the miswiring protection.

9.2.5.2 Test circuit

The test circuit shall be as shown in Figure 66.

9.2.5.3 Test procedure

The V– connection shall be disconnected at the EUT and power shall be cycled. CAN_H and CAN_L shall be monitored.

9.2.5.4 Criteria for compliance

It shall be verified that the CAN_H and CAN_L signals remain within the limits specified in Table 23 and Table 24.

9.2.6 Differential input impedance test

9.2.6.1 Test purpose

The purpose of this test is to verify that the input impedance of the EUT meets the requirements of the physical layer.

9.2.6.2 Test circuit

The test circuit shall be as shown in Figure 67.

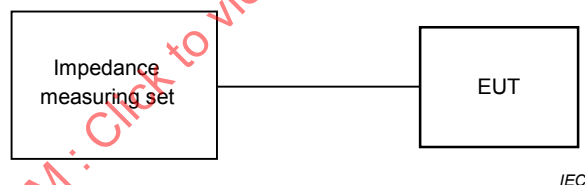


Figure 67 – Differential impedance test circuit

9.2.6.3 Test procedure

Power shall be applied to the EUT. The transceiver of the EUT shall be disabled in such a way that the impedance is not affected. The input impedance of the EUT between CAN_H and CAN_L shall then be measured at a frequency of $100\text{ kHz} \pm 1\text{ kHz}$.

9.2.6.4 Criteria for compliance

The input impedance of the EUT between CAN_H and CAN_L shall not exceed the values specified in Table 22.

9.2.7 Transmit levels

9.2.7.1 Test purpose

The purpose of this test is to determine whether the values of CAN_H and CAN_L with respect to V– under simulated worst case impedance conditions are within the limits specified as part of the physical layer requirements.

9.2.7.2 Test circuit

The test circuit shall be as shown in Figure 68.

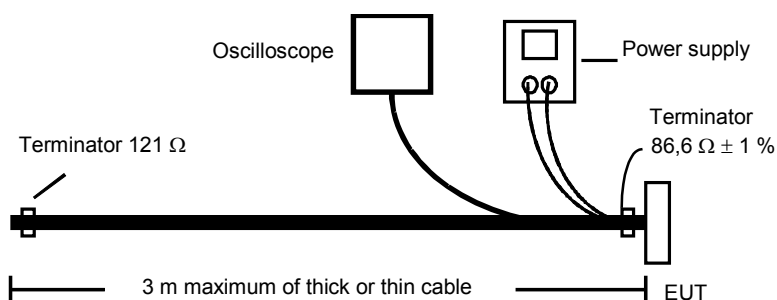


Figure 68 – Transmit level test setup

9.2.7.3 Test procedure

For this test, the EUT shall be transmitting. With no other device on the link, the EUT will automatically start sending duplicate MAC ID request messages after completing its power-up sequence. The actual message content from the EUT shall be observed on the oscilloscope. The following values shall be recorded (see Figure 69):

- recessive CAN_L voltage with respect to V-;
- recessive CAN_H voltage with respect to V-;
- dominant CAN_L voltage with respect to V-;
- dominant CAN_H voltage with respect to V-;
- differential output level.

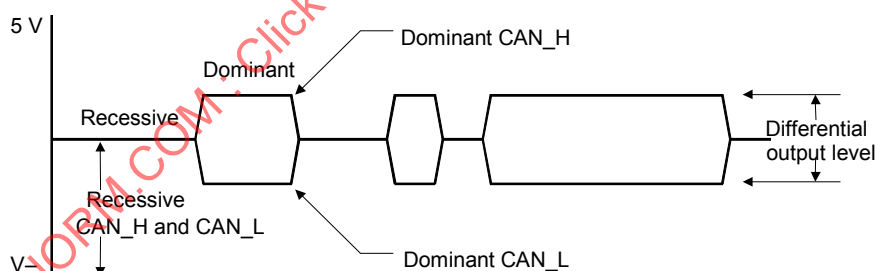


Figure 69 – Transmit levels

9.2.7.4 Criteria for compliance

The values shall be in accordance with those specified in Table 23.

9.2.8 Acknowledge delay

9.2.8.1 Test purpose

The purpose of this test is to check that the acknowledge delay is within the limits specified as part of the physical layer requirements.

9.2.8.2 Test circuit

The test circuit shall be as shown in Figure 70.

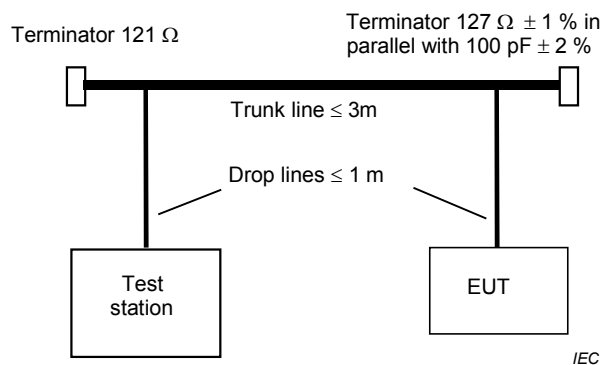


Figure 70 – Timing test setup

9.2.8.3 Test procedure

The EUT and the test station shall be set to the highest bit rate supported by the EUT. The test station shall repetitively send a `get_attribute_single` request to the EUT without opening a messaging connection. This ensures that the EUT sends no messages other than the acknowledge bit. An oscilloscope shall be used to measure the acknowledge response time at the EUT. This is the time from the falling edge of the last bit sent by the test station to the rising edge of the acknowledge bit sent by the EUT (see Figure 71).

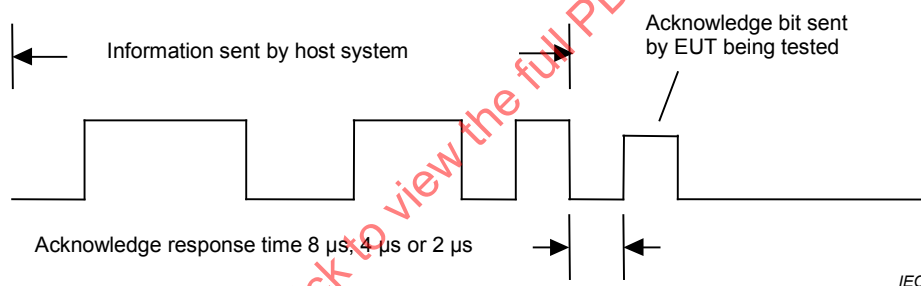


Figure 71 – Timing

9.2.8.4 Criteria for compliance

Depending on the bit rate used, the following values shall be subtracted from the response times measured:

- 125 kbit/s 8 μs;
- 250 kbit/s 4 μs;
- 500 kbit/s 2 μs.

The resultant device acknowledge delay shall be less than or equal to the value specified in 5.7.2.

9.2.9 CDI tests

9.2.9.1 Test purpose

The purpose of this test is to determine whether the missing acknowledgement bits and bus-off conditions under worst case conditions are within the limits specified within the manufacturer's documentation.

9.2.9.2 Test circuit

The test circuit shall be as shown in Figure 72 noting the following points:

- the system shall be run at the maximum bit rate supported by the EUT;
- the cable system shall be constructed of the maximum trunk line length of thick cable suitable for this bit rate;
- the maximum cumulative drop line shall be constructed of the maximum thin cable drop line length suitable for this bit rate;
- 62 devices or 62 device equivalents shall be used to load the CDI near the EUT;

NOTE A device equivalent is the minimum differential input impedance as defined in Table 22.

- a power supply shall be used that meets the specifications described in Table 55 (minimum current capacity 4 A);
- a worst case link power load shall be connected adjacent to the EUT. This load shall cause the V- line to develop a 5 V +0,0/-0,1 V drop from the power supply to the EUT;
- a test station shall be used that is capable of both producing valid communication traffic to the EUT and monitoring all communication traffic.

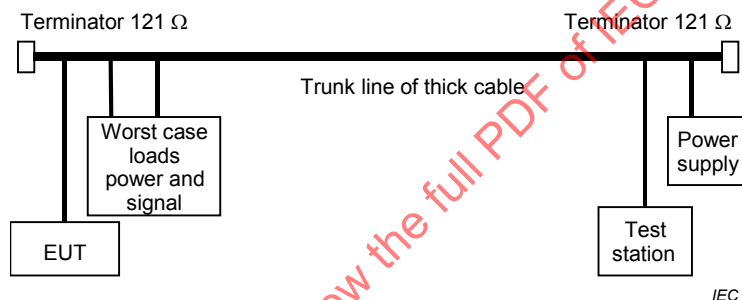


Figure 72 – CDI test configuration

9.2.9.3 Test procedure

Communications shall be established between the test station and the EUT. Any missing acknowledgement bits or bus-off conditions shall be recorded.

9.2.9.4 Criteria for compliance

The recorded missing acknowledgement bits and bus-off conditions shall be as specified by the manufacturer of the EUT.

9.2.10 Electromagnetic compatibility testing

9.2.10.1 General

The test circuit shall be as shown in Figure 72 noting the following points:

- the system shall be run at the maximum bit rate supported by the EUT;
- the trunk line shall be constructed of thick cable;
- the trunk line length shall be as required by the individual tests;
- a power supply shall be used that meets the specifications described in Table 55 (minimum current capacity 4 A);
- a test station shall be used that is capable of both producing valid communication traffic to the EUT and monitoring all communication traffic.

Unless otherwise noted, the tests shall be carried out at an ambient temperature of $+23\text{ °C} \pm 5\text{ °C}$.

The EUT shall be mounted in free air and shall be connected to the DeviceNet CDI according to the manufacturer's instructions and supplied with its rated voltage.

9.2.10.2 Immunity

9.2.10.2.1 Electrostatic discharge (ESD) immunity

This test shall be performed in accordance with IEC 61000-4-2:2008 and 8.9.2.2.

The trunk line length shall be the maximum allowable (see 5.7.6.2).

9.2.10.2.2 Radiated radio-frequency electromagnetic field immunity

This test shall be performed in accordance with IEC 61000-4-3:2006, Amendment 1 (2007), Amendment 2 (2010), and 8.9.2.3.

The trunk line length shall be the maximum allowable (see 5.7.6.2). The EUT shall be connected to the trunk line using the maximum length drop line allowable (see 5.7.6.3). The overall maximum distance between the EUT and the test station that is allowed for the given bit rate shall not be exceeded.

9.2.10.2.3 Electrical fast transient/burst immunity

This test shall be performed in accordance with IEC 61000-4-4:2012 and 8.9.2.4.

The trunk line length shall be the maximum allowable (see 5.7.6.2). The EUT shall be directly connected to the trunk line – if this is not possible, then the drop line shall not exceed 1 m. The overall maximum distance between the EUT and the test station that is allowed for the given bit rate shall not be exceeded.

9.2.10.2.4 Surge immunity

This test shall be performed in accordance with IEC 61000-4-5:2005 and 8.9.2.5.

The trunk line length shall be the maximum allowable (see 5.7.6.2). The EUT shall be directly connected to the trunk line – if this is not possible, then the drop line shall not exceed 1 m. The overall maximum distance between the EUT and the test station that is allowed for the given bit rate shall not be exceeded.

9.2.10.2.5 Conducted radio-frequency disturbance immunity

This test shall be performed in accordance with IEC 61000-4-6:2013 and 8.9.2.6.

The trunk line length shall be the maximum allowable (see 5.7.6.2). The EUT shall be directly connected to the trunk line – if this is not possible, then the drop line shall not exceed 1 m. The overall maximum distance between the EUT and the test station that is allowed for the given bit rate shall not be exceeded.

9.2.10.3 Emissions

9.2.10.3.1 Radiated emissions

This test shall be performed in accordance with CISPR 11:2009, Amendment 1 (2010), group 1, class A and 8.9.3.1.

The trunk line length shall be the maximum allowable (see 5.7.6.2). The EUT shall be connected to the trunk line using the maximum length drop line allowable (see 5.7.6.3). The overall maximum distance between the EUT and the test station that is allowed for the given bit rate shall be not be exceeded.

9.2.10.3.2 Conducted emissions

This test shall be performed in accordance with CISPR 11:2009, Amendment 1 (2010), group 1, class A and 8.9.3.2.

The trunk line length shall be the minimum required for the test fixture, and shall not exceed the maximum length specified in 5.7.6.2. The EUT shall be connected to the trunk line using the maximum length drop line allowable (see 5.7.6.3).

9.3 Logical testing

9.3.1 General

Tests shall only be performed on nodes that support the particular functionality being tested.

Logical tests require the presence of a test station on the controller-device interface, which can:

- send DeviceNet messages;
- receive and evaluate DeviceNet messages;
- record and time stamp traffic on the CDI.

9.3.2 Duplicate MAC ID check test

9.3.2.1 Test purpose

The purpose of this test is to verify that the EUT meets the requirements regarding the handling of duplicate MAC ID mechanisms.

9.3.2.2 Test circuit

The test circuit shall be as shown in Figure 73.

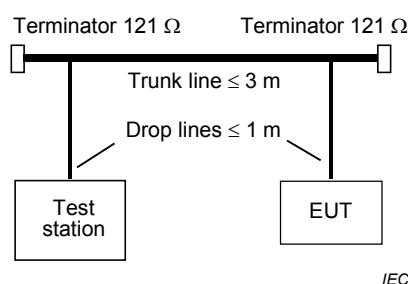


Figure 73 – Test circuit for logical tests

9.3.2.3 Test procedure

9.3.2.3.1 Successful duplicate MAC ID check

The EUT MAC ID shall be set to 0. Any necessary external power shall be applied to the EUT. The EUT shall then be physically connected to the link. The test station shall then be connected to the link and shall record the communication traffic.

9.3.2.3.2 Unsuccessful duplicate MAC ID check

The EUT MAC ID shall be set to 0. Any necessary external power shall be applied to the EUT. The EUT shall then be physically connected to the link. The test station shall respond to the first duplicate MAC ID check request message with a duplicate MAC ID check response message.

9.3.2.4 Criteria for compliance

9.3.2.4.1 Successful duplicate MAC ID check

It shall be verified that:

- the EUT transmits a duplicate MAC ID check request message;
- the EUT transmits a second duplicate MAC ID check request between 0,9 s and 1,5 s later;
- the EUT assumes the on-line state no earlier than 0,9 s after transmitting the second duplicate MAC ID check request message unless Quick Connect is enabled; in this case the EUT transitions to the on-line state immediately after successful transmission of the first duplicate MAC ID check request message while continuing with the remainder of the duplicate MAC ID check process;
- the EUT transmits no other messages except duplicate MAC ID check request and response messages before assuming the on-line state;
- the EUT does not react to any request messages except duplicate MAC ID check request and response messages before assuming the on-line state.

9.3.2.4.2 Unsuccessful duplicate MAC ID check

The EUT shall assume the communication fault state (see

Table 12) after receiving the duplicate MAC ID check response message.

9.3.3 UCMM

9.3.3.1 Test purpose

The purpose of this test is to determine whether a node that supports UCMM meets the requirements associated with the creation of explicit messaging connections.

9.3.3.2 Test circuit

The test circuit shall be as shown in Figure 73.

9.3.3.3 Test procedure

The test station shall transmit a UCMM open request message using message group 1. After opening this messaging connection, an explicit message shall be transmitted across this connection. The test station shall transmit a UCMM close request message. The test station shall transmit a second explicit message and wait 10 s for a response.

The above procedure shall be repeated for message groups 2 and 3.

9.3.3.4 Criteria for compliance

It shall be verified that:

- the first explicit message is transmitted and received successfully in at least one message group;

- the EUT does not transmit a response to the second explicit message.

9.3.4 Allocation of predefined master/slave connection set – Explicit messaging connection

9.3.4.1 Test purpose

The purpose of this test is to determine whether the EUT meets the requirements associated with the allocation of a predefined master/slave connection set for explicit messaging connections.

9.3.4.2 Test circuit

The test circuit shall be as shown in Figure 73.

9.3.4.3 Test procedure

The test station shall send a request to allocate the predefined master/slave connection set (allocation choice = 1). After allocating this messaging connection, an explicit message shall be transmitted across this connection. The test station shall de-allocate the predefined master/slave connection set (release choice = 1). The test station shall transmit a second explicit message and wait 10 s for a response.

9.3.4.4 Criteria for compliance

It shall be verified that:

- the first explicit message is transmitted and received as defined in the predefined master/slave connection set;
- the EUT does not transmit a response to the second explicit message.

9.3.5 Allocation of predefined master/slave connection set – I/O messaging connection

9.3.5.1 Test purpose

The purpose of this test is to determine whether the EUT meets the requirements associated with the allocation of a predefined master/slave connection set for I/O messaging connections.

9.3.5.2 Test circuit

The test circuit shall be as shown in Figure 73.

9.3.5.3 Test procedure

The test station shall establish an I/O connection from the predefined master/slave connection set with an EPR of 1 s or less. After establishing this I/O connection, the exchange of I/O messages across this connection shall be observed. The test station shall then de-allocate the predefined master/slave connection set. The test station shall transmit further I/O messages and wait 10 s for I/O messages from the EUT.

This procedure shall be performed for all the I/O allocation choices supported.

9.3.5.4 Criteria for compliance

It shall be verified that:

- the I/O messages are transmitted and received consistent with the EPR and the allocation choice;

- the EUT does not transmit any I/O messages after the connection has been released.

9.3.6 Logical testing of safety products

Additional requirements for logical testing of safety products are specified in IEC 61784-3-2.

IECNORM.COM : Click to view the full PDF of IEC 62026-3:2014

Bibliography

IEC 61131-3:2013, *Programmable controllers – Part 3: Programming languages*

IEC 61158 (all parts), *Industrial communication networks – Fieldbus specifications*

IEC 61508 (all parts), *Functional safety of electrical/electronic/programmable electronic safety-related systems*

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

IEC 61784-5-2, *Industrial communication networks – Profiles – Part 5-2: Installation of fieldbuses – Installation profiles for CPF 2*

ISO/IEC 7498-1:1994, *Information technology – Open Systems Interconnection – Basic Reference Model: The Basic Model*

ANSI/ASME B1.1, *Unified Inch Screw Threads (UN and UNR Thread Form)*

NEMA 250, *Enclosures for Electrical Equipment (1 000 V Maximum)*

UL 1277, *Electrical Power and Control Tray Cables with Optional Optical-Fiber Members*

IECNORM.COM : Click to view the full PDF of IEC 62026-3:2014

IECNORM.COM : Click to view the full PDF of IEC 62026-3:2014

SOMMAIRE

AVANT-PROPOS	132
INTRODUCTION.....	134
1 Domaine d'application	135
2 Références normatives	135
3 Termes, définitions, symboles et termes abrégés	136
3.1 Termes et définitions	136
3.2 Symboles et termes abrégés	142
4 Classification	142
4.1 Généralités	142
4.2 Modèle de communication DeviceNet.....	144
4.3 DeviceNet, CAN et CIP™	144
5 Caractéristiques	146
5.1 Connexions DeviceNet.....	146
5.1.1 Généralités	146
5.1.2 Utilisation du champ identificateur de CAN par DeviceNet	146
5.1.3 Établissement des connexions.....	147
5.2 Protocole de messagerie DeviceNet.....	148
5.2.1 Messagerie explicite	148
5.2.2 Messagerie d'entrée/sortie.....	162
5.2.3 Fragmentation/réassemblage.....	162
5.2.4 Jeu de connexions hors ligne.....	168
5.2.5 Cadenceur d'appareil.....	181
5.2.6 Message de déconnexion d'appareil	182
5.2.7 Protocole de détection de duplication d'adresse MAC ID	184
5.2.8 Connexion rapide	185
5.3 Classes d'objets de communication DeviceNet.....	186
5.3.1 Généralités	186
5.3.2 Définition de la classe d'objet identité (code d'ID de classe: 0x01).....	186
5.3.3 Définition de la classe d'objet routeur de messages (code d'ID de classe: 0x02)	186
5.3.4 Définition de la classe d'objet DeviceNet (code d'ID de classe: 0x03)	187
5.3.5 Définition de la classe d'objet connexion (code d'ID de classe: 0x05)	187
5.3.6 Définition de la classe d'objet gestionnaire d'acquittements (code d'ID de classe: 0x2B)	188
5.4 Diagramme d'états d'accès à la liaison	188
5.4.1 Généralités	188
5.4.2 Diagramme de transition d'états et table états-événements	188
5.4.3 Détection de duplication d'adresse MAC ID	192
5.5 Jeu de connexions maître/esclave prédéfini.....	193
5.5.1 Généralités	193
5.5.2 Messages du jeu de connexions maître/esclave prédéfini	193
5.5.3 Services spécifiques à la classe d'objet DeviceNet pour le jeu de connexions maître/esclave	195
5.5.4 Caractéristiques de l'objet connexion esclave.....	204
5.5.5 Caractéristiques de l'objet connexion maître.....	211
5.5.6 Messages de commande/réponse requête multiple	211
5.5.7 Messages de commande/réponse interrogation	212

5.5.8	Messages de commande/réponse interrogation multidestinataire	213
5.5.9	Connexions cyclique/changement d'état	214
5.5.10	Appareils du groupe 2 seulement	215
5.6	CIP Safety™ sur DeviceNet	216
5.6.1	Généralités	216
5.6.2	Utilisation d'identificateurs de CAN pour CIP Safety sur DeviceNet	216
5.7	Couche physique	216
5.7.1	Généralités	216
5.7.2	Transcepteur	218
5.7.3	Mise à la terre	219
5.7.4	Isolation	219
5.7.5	Support de transmission	221
5.7.6	Topologie	222
5.7.7	Alimentation de la liaison	223
6	Information sur le matériel	224
7	Conditions normales de service, de montage et de transport	224
7.1	Conditions normales de service	224
7.1.1	Généralités	224
7.1.2	Température de l'air ambiant	224
7.1.3	Altitude	225
7.1.4	Conditions climatiques	225
7.2	Conditions durant le transport et le stockage	225
7.3	Montage	225
8	Exigences relatives à la construction et au fonctionnement	226
8.1	Voyants et commutateurs mécaniques de configuration	226
8.2	Câble DeviceNet	226
8.2.1	Généralités	226
8.2.2	Modèle de profil de câble	226
8.2.3	Profil de câble épais	228
8.2.4	Profil de câble fin	231
8.2.5	Profil de câble plat	235
8.3	Résistances de terminaison	239
8.4	Connecteurs	240
8.4.1	Spécifications générales	240
8.4.2	Modèle de profil de connecteur	240
8.4.3	Profil de connecteur ouvert	241
8.4.4	Profil de mini-connecteur scellé	244
8.4.5	Profil de micro-connecteur scellé	246
8.4.6	Profil de connecteur principal plat	248
8.5	Dérivateurs et raccords d'alimentation	251
8.5.1	Dérivateurs d'appareil	251
8.5.2	Raccords d'alimentation	252
8.6	Appareils alimentés par la liaison	253
8.7	Protection contre les erreurs de câblage	253
8.8	Blocs d'alimentation	253
8.9	Compatibilité électromagnétique (CEM)	254
8.9.1	Généralités	254
8.9.2	Immunité	254
8.9.3	Emissions	255

8.10	Exigences supplémentaires de sécurité fonctionnelle relatives à la CEM	256
9	Essais	256
9.1	Généralités	256
9.2	Essais électriques et de compatibilité électromagnétique (CEM)	256
9.2.1	Essai du bloc d'alimentation DeviceNet	256
9.2.2	Consommation maximale de courant	257
9.2.3	Comportement à la mise sous tension	258
9.2.4	Inversion des connexions V+ et V-	258
9.2.5	Déconnexion de V-	259
9.2.6	Essai d'impédance d'entrée différentielle	259
9.2.7	Niveaux de transmission	260
9.2.8	Délai d'acquittement	261
9.2.9	Essais CDI	263
9.2.10	Essais de compatibilité électromagnétique	264
9.3	Essais logiques	265
9.3.1	Généralités	265
9.3.2	Essai de vérification de duplication d'adresse MAC ID	266
9.3.3	UCMM	267
9.3.4	Attribution d'un jeu de connexions maître/esclave prédéfini – Connexion de messagerie explicite	267
9.3.5	Attribution d'un jeu de connexions maître/esclave prédéfini – Connexion de messagerie d'E/S	268
9.3.6	Essais logiques des produits de sécurité	268
	Bibliographie	269
	Figure 1 – Interfaces appareil de commande-appareil DeviceNet types	143
	Figure 2 – Comparaison entre l'architecture du protocole DeviceNet et le modèle de référence OSI	145
	Figure 3 – Utilisation du champ identificateur de CAN par DeviceNet	146
	Figure 4 – Utilisation du champ de données CAN dans un message explicite	148
	Figure 5 – Format du champ de données des messages explicites	149
	Figure 6 – Format de l'en-tête d'un message explicite	150
	Figure 7 – Format du champ de service	151
	Figure 8 – Format de requête d'ouverture de connexion de messagerie explicite	152
	Figure 9 – Format de la réponse d'ouverture de connexion de messagerie explicite	155
	Figure 10 – Format de la requête de fermeture de connexion	157
	Figure 11 – Format de la réponse de fermeture de connexion	157
	Figure 12 – Format d'un message de requête explicite non fragmenté, valeurs 0 – 3	159
	Figure 13 – Format d'un message de requête explicite non fragmenté, valeur 4	160
	Figure 14 – Format d'un message de réponse positive non fragmentée	161
	Figure 15 – Message de réponse d'erreur	162
	Figure 16 – Champ de données d'un message d'E/S	162
	Figure 17 – Format du protocole de fragmentation DeviceNet	163
	Figure 18 – Format d'un fragment de message d'E/S	164
	Figure 19 – Format d'un fragment de message explicite	164
	Figure 20 – Format d'un message d'acquittement	166

Figure 21 – Établissement de la propriété hors ligne	169
Figure 22 – Nature multidestinataire de la propriété hors ligne	170
Figure 23 – Message de requête de propriété hors ligne	171
Figure 24 – Protocole de message de réponse de propriété hors ligne	172
Figure 25 – Message de requête de défaut de communication – Protocole multidestinataire	173
Figure 26 – Message de requête de défaut de communication – Protocole point à point	174
Figure 27 – Message de requête de défaut de communication "identification" – Protocole multidestinataire	176
Figure 28 – Message de réponse de défaut de communication "identification"	176
Figure 29 – Message de requête de défaut de communication "identification" – Protocole point à point	177
Figure 30 – Message de requête de défaut de communication "qui"	178
Figure 31 – Message de réponse "qui"	180
Figure 32 – Message de requête de défaut de communication "modification du MAC ID"	180
Figure 33 – Message de cadenceur d'appareil	182
Figure 34 – Message de déconnexion d'appareil	183
Figure 35 – Champ identificateur de CAN pour la vérification de duplication d'adresse MAC ID	184
Figure 36 – Format du champ de données d'un message de vérification de duplication d'adresse MAC ID	185
Figure 37 – Diagramme de transition d'états d'accès à la liaison	190
Figure 38 – Contenu de l'octet de choix d'attribution	196
Figure 39 – Message de requête allocate_master/slave_connection_set	196
Figure 40 – Réponse positive à une requête allocate_master/slave_connection_set	198
Figure 41 – Choix d'une connexion de messagerie explicite parent	200
Figure 42 – Contenu de l'octet de choix de libération	201
Figure 43 – Message de requête release_master/slave_connection_set	202
Figure 44 – Réponse positive à une requête release_master/slave_connection_set	203
Figure 45 – Diagramme de transition d'états des connexions d'E/S maître/esclave prédéfinies	206
Figure 46 – Diagramme de transition d'états pour la connexion de messagerie explicite maître/esclave prédéfinie	209
Figure 47 – Schéma fonctionnel de la couche physique	218
Figure 48 – Appareil contenant une couche physique non isolée	220
Figure 49 – Appareil contenant une couche physique isolée	221
Figure 50 – Topologie du support DeviceNet	223
Figure 51 – Câble épais: configuration physique	230
Figure 52 – Câble épais: courant disponible sur le bus d'alimentation DeviceNet	231
Figure 53 – Câble fin: configuration physique	234
Figure 54 – Câble fin: courant disponible sur le bus d'alimentation DeviceNet	235
Figure 55 – Câble plat: configuration physique	238
Figure 56 – Câble plat: courant disponible sur le bus d'alimentation DeviceNet	239
Figure 57 – Configuration des broches de connecteur ouvert	243
Figure 58 – Géométrie de connecteur ouvert	244

Figure 59 – Configuration des broches de mini-connecteur	246
Figure 60 – Configuration des broches de micro-connecteur	248
Figure 61 – Disposition du connecteur principal plat – Partie 1	250
Figure 62 – Disposition du connecteur principal plat – Partie 2	251
Figure 63 – Circuit d'essai du temps de montée du bloc d'alimentation	256
Figure 64 – Circuit d'essai de consommation de courant.....	257
Figure 65 – Circuit d'essai de mise sous tension.....	258
Figure 66 – Circuit d'essai pour l'inversion de V+ et V-, et aussi pour la coupure de V-	259
Figure 67 – Circuit d'essai d'impédance différentielle.....	260
Figure 68 – Installation pour l'essai des niveaux de transmission.....	260
Figure 69 – Niveaux de transmission	261
Figure 70 – Configuration de l'essai de temporisation	262
Figure 71 – Temporisation	262
Figure 72 – Configuration de l'essai CDI.....	264
Figure 73 – Circuit d'essai pour les essais logiques	266
Tableau 1 – Valeurs de format de corps de message	153
Tableau 2 – Valeurs de sélection des groupes	154
Tableau 3 – ID de message source dans une requête d'ouverture de connexion de messagerie explicite	154
Tableau 4 – ID de message de destination dans une réponse d'ouverture de connexion de messagerie explicite	156
Tableau 5 – Conditions/codes d'erreur UCMM	158
Tableau 6 – Valeurs des bits du type de fragment.....	163
Tableau 7 – Valeurs du bit d'état d'acquittement	167
Tableau 8 – Jeu de connexions hors ligne	168
Tableau 9 – Rapport d'adresses basé sur le masque	174
Tableau 10 – Plages des codes de déconnexion du message de déconnexion d'appareil.....	184
Tableau 11 – Codes de déconnexion "ouverts" du message de déconnexion d'appareil.....	184
Tableau 12 – Tableau états-événements d'accès à la liaison (1 de 3)	190
Tableau 13 – Champs identificateurs pour le jeu de connexions maître/esclave prédéfini	194
Tableau 14 – Paramètres du champ des données de service d'une requête allocate_master/slave_connection_set.....	195
Tableau 15 – Paramètres de la réponse allocate_master/slave_connection_set.....	197
Tableau 16 – Paramètres du champ des données de service d'une requête release_master/slave_connection_set.....	201
Tableau 17 – Codes d'erreur supplémentaires spécifiques à l'objet DeviceNet.....	204
Tableau 18 – ID d'instance de connexion des connexions maître/esclave prédéfinies	205
Tableau 19 – Table états-événements des connexions d'E/S maître/esclave prédéfinies (1 de 2)	207
Tableau 20 – Table états-événements des connexions de messagerie explicite maître/esclave prédéfinies	210
Tableau 21 – Accès aux attributs de l'objet connexion d'E/S maître/esclave prédéfinie.....	211

Tableau 22 – Caractéristiques générales de la couche physique.....	217
Tableau 23 – Caractéristiques du transmetteur.....	218
Tableau 24 – Caractéristiques du récepteur.....	219
Tableau 25 – Limites de charge.....	224
Tableau 26 – Profil de câble: spécifications des paires de données.....	226
Tableau 27 – Profil de câble: spécifications de la paire d'alimentation à courant continu	227
Tableau 28 – Profil de câble: spécifications générales.....	227
Tableau 29 – Profil de câble: topologie.....	227
Tableau 30 – Câble épais: spécifications des paires de données.....	228
Tableau 31 – Câble épais: spécifications de la paire d'alimentation à courant continu.....	228
Tableau 32 – Câble épais: spécifications générales.....	229
Tableau 33 – Câble épais: topologie.....	229
Tableau 34 – Câble épais: courant maximal disponible (A) d'après la longueur du réseau.....	230
Tableau 35 – Câble fin: spécifications des paires de données.....	232
Tableau 36 – Câble fin: spécifications de la paire d'alimentation à courant continu.....	232
Tableau 37 – Câble fin: spécifications générales.....	233
Tableau 38 – Câble fin: topologie.....	233
Tableau 39 – Câble fin: courant maximal disponible (A) d'après la longueur du réseau.....	234
Tableau 40 – Câble plat: spécifications des paires de données.....	236
Tableau 41 – Câble plat: spécifications de la paire d'alimentation à courant continu.....	236
Tableau 42 – Câble plat: spécifications générales.....	237
Tableau 43 – Câble plat: topologie.....	237
Tableau 44 – Câble plat: courant maximal disponible (A) d'après la longueur du réseau.....	239
Tableau 45 – Modèle de profil de connecteur.....	241
Tableau 46 – Connecteur ouvert.....	242
Tableau 47 – Mini-connecteur scellé.....	245
Tableau 48 – Micro-connecteur scellé.....	247
Tableau 49 – Connecteur principal plat.....	249
Tableau 50 – Spécifications des conducteurs principaux internes.....	252
Tableau 51 – Spécifications des conducteurs de dérivation internes.....	252
Tableau 52 – Spécifications des conducteurs principaux internes.....	252
Tableau 53 – Spécifications des conducteurs de dérivation internes.....	253
Tableau 54 – Spécifications du régulateur de tension.....	253
Tableau 55 – Spécifications des blocs d'alimentation DeviceNet.....	254
Tableau 56 – Critères de comportement pour les essais d'immunité.....	255

COMMISSION ÉLECTROTECHNIQUE INTERNATIONALE

APPAREILLAGE À BASSE TENSION – INTERFACES APPAREIL DE COMMANDE-APPAREIL (CDI) –

Partie 3: DeviceNet

AVANT-PROPOS

- 1) La Commission Electrotechnique Internationale (IEC) est une organisation mondiale de normalisation composée de l'ensemble des comités électrotechniques nationaux (Comités nationaux de l'IEC). L'IEC 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, l'IEC – 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 l'IEC"). 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 l'IEC, participent également aux travaux. L'IEC 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 l'IEC concernant les questions techniques représentent, dans la mesure du possible, un accord international sur les sujets étudiés, étant donné que les Comités nationaux de l'IEC intéressés sont représentés dans chaque comité d'études.
- 3) Les Publications de l'IEC se présentent sous la forme de recommandations internationales et sont agréées comme telles par les Comités nationaux de l'IEC. Tous les efforts raisonnables sont entrepris afin que l'IEC s'assure de l'exactitude du contenu technique de ses publications; l'IEC 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 l'IEC s'engagent, dans toute la mesure possible, à appliquer de façon transparente les Publications de l'IEC dans leurs publications nationales et régionales. Toutes divergences entre toutes Publications de l'IEC et toutes publications nationales ou régionales correspondantes doivent être indiquées en termes clairs dans ces dernières.
- 5) L'IEC 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 l'IEC. L'IEC 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 l'IEC, 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 l'IEC, 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 l'IEC ou de toute autre Publication de l'IEC, 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 l'IEC peuvent faire l'objet de droits de brevet. L'IEC 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 IEC 62026-3 a été établie par le sous-comité 121A: Appareillage à basse tension, du comité d'études 121 de l'IEC: Appareillages et ensembles d'appareillages basse tension.

Cette troisième édition de l'IEC 62026-3 annule et remplace la deuxième édition parue en 2008. Cette troisième édition constitue une révision technique.

Les principales modifications effectuées par rapport à l'édition antérieure sont les suivantes:

- spécification des messages du groupe 4 (5.1.2);
- clarification du protocole de messagerie (5.2);
- ajout de messages d'E/S interrogation multidestinataire (5.5.2 et 5.5.8);
- clarifications du comportement esclave (5.5.4 et 5.5.9);
- clarifications de la couche physique (transcepteur) en 5.7;
- corrections/clarifications diverses relatives au câble (8.2);
- clarification des essais de compatibilité électromagnétique (CEM) en 9.2.10 et des essais logiques (9.3).

Le texte de cette norme est issu des documents suivants:

FDIS	Rapport de vote
17B/1814/FDIS	121A/18/RVC

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/IEC, Partie 2

Une liste de toutes les parties de l'IEC 62026, regroupées sous le titre général *Appareillage à basse tension – Interfaces appareil de commande-appareil (CDI)*, peut être consultée sur le site web de l'IEC.

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 l'IEC 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.

Le contenu des corrigenda de mars 2015 et octobre 2019 a été pris en considération dans cet exemplaire.

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

INTRODUCTION

L'utilisation de DeviceNet™¹ est destinée, entre autres, à des applications industrielles d'automatisation. Ces applications peuvent inclure des appareils tels que les détecteurs de fin de course, les détecteurs de proximité, les vannes électropneumatiques, les relais, les démarreurs, les panneaux d'interface opérateur, les entrées analogiques, les sorties analogiques et les appareils de commande.

IECNORM.COM : Click to view the full PDF of IEC 62026-3:2014

¹ DeviceNet™ est une marque commerciale d'ODVA, Inc. Cette information est donnée à l'intention des utilisateurs de la présente Norme internationale et ne signifie nullement que l'IEC approuve ou recommande le détenteur de cette marque commerciale ou d'un quelconque de ses produits. La conformité à la présente norme ne nécessite pas l'utilisation de la marque commerciale DeviceNet™. L'utilisation de la marque commerciale DeviceNet™ nécessite l'autorisation d'ODVA, Inc.

APPAREILLAGE À BASSE TENSION – INTERFACES APPAREIL DE COMMANDE-APPAREIL (CDI) –

Partie 3: DeviceNet

1 Domaine d'application

La présente partie de l'IEC 62026 spécifie un système d'interfaces entre un ou plusieurs appareils de commande d'une part, et des appareils ou éléments de commutation pour circuits de commande, d'autre part. Le système d'interfaces utilise deux paires conductrices, réunies dans un câble – l'une des paires fournit un support de communication différentiel et l'autre alimente les appareils. Cette partie définit des exigences permettant l'interopérabilité des composants incluant de telles interfaces.

La présente partie de l'IEC 62026 spécifie, pour DeviceNet, les exigences particulières suivantes:

- exigences concernant les interfaces entre appareils de commande et éléments de commutation;
- conditions normales de service pour les appareils;
- exigences relatives à la construction et au fonctionnement;
- essais de vérification de la conformité aux exigences.

Ces exigences particulières s'ajoutent aux exigences générales de l'IEC 62026-1.

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).

IEC 60529:1989, *Degrés de protection procurés par les enveloppes (Code IP)*
IEC 60529:1989/AMD 1:1999
IEC 60529:1989/AMD 2:2013

IEC 60947-5-2:2007, *Appareillage à basse tension – Partie 5-2: Appareils et éléments de commutation pour circuits de commande – Détecteurs de proximité*
IEC 60947-5-2:2007/AMD 1:2012

IEC 61000-4-2:2008, *Compatibilité électromagnétique (CEM) – Partie 4-2: Techniques d'essai et de mesure – Essai d'immunité aux décharges électrostatiques*

IEC 61000-4-3:2006, *Compatibilité électromagnétique (CEM) – Partie 4-3: Techniques d'essai et de mesure – Essai d'immunité aux champs électromagnétiques rayonnés aux fréquences radioélectriques*
IEC 61000-4-3:2006/AMD 1:2007
IEC 61000-4-3:2006/AMD 2:2010

IEC 61000-4-4:2012, *Compatibilité électromagnétique (CEM) – Partie 4-4: Techniques d'essai et de mesure – Essai d'immunité aux transitoires électriques rapides en salves*

IEC 61000-4-5:2005, *Compatibilité électromagnétique (CEM) – Partie 4-5: Techniques d'essai et de mesure – Essai d'immunité aux ondes de choc*

IEC 61000-4-6:2013, *Compatibilité électromagnétique (CEM) – Partie 4-6: Techniques d'essai et de mesure – Immunité aux perturbations conduites, induites par les champs radioélectriques*

IEC 61158-4-2:2014 *Réseaux de communication industriels – Spécifications des bus de terrain – Partie 4-2: Spécification du protocole de la couche de liaison de données – Eléments de type 2*

IEC 61158-5-2:2014, *Réseaux de communication industriels – Spécification des bus de terrain – Partie 5-2: Définition des services de la couche application – Eléments de type 2*

IEC 61158-6-2:2014, *Réseaux de communication industriels – Spécifications des bus de terrain – Partie 6-2: Spécification de protocole de la couche application – Eléments de type 2*

IEC 61784-3-2, *Réseaux de communication industriels – Profils – Partie 3-2: Bus de terrain de sécurité fonctionnelle – Spécifications supplémentaires pour CPF 2*

IEC 62026-1:2007, *Appareillage à basse tension – Interfaces appareil de commande-appareil (CDI) – Partie 1: Règles générales*

CISPR 11:2009, *Appareils industriels, scientifiques et médicaux — Caractéristiques de perturbations radioélectriques – Limites et méthodes de mesure*
CISPR 11:2009/AMD 1:2010

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 haute vitesse*

ANSI B93.55M-1981 (R1988), *Hydraulic Fluid Power – Solenoid Piloted Industrial Valves – Interface Dimensions for Electrical Connectors* (disponible en anglais seulement)

ASTM D 4566-942, *Standard Test Methods for Electrical Performance Properties of Insulations and Jackets for Telecommunications Wire and Cable*

3 Termes, définitions, symboles et termes abrégés

Pour les besoins du présent document, les termes, définitions, symboles et abréviations de l'IEC 62026-1 ainsi que les suivants s'appliquent.

3.1 Termes et définitions

3.1.1

fragmentation acquittée

fragmentation effectuée sur un message explicite, dans laquelle la transmission d'un fragment par l'objet émetteur est suivie de la transmission d'un acquittement par l'objet chargé de la réception

Note 1 à l'article: La réception de chaque fragment est acquittée par l'objet effectuant la réception.

² Il existe une version plus récente de ce document (ASTM D4566-08e1), mais la révision citée s'applique pour la présente Norme.

3.1.2**état d'acquittement**

champ dans le format d'un message d'acquittement/réponse qui indique si une erreur a été trouvée ou non par le destinataire du message fragmenté

Note 1 à l'article: Cela s'applique spécifiquement au protocole de fragmentation de DeviceNet.

3.1.3**objets applicatifs**

ensemble de classes d'objets et de leurs instances présentes dans la station

Note 1 à l'article: Ces objets assurent la gestion et la production des données et messages échangés à travers des interfaces appareil de commande-appareil (CDI, controller-device interfaces) DeviceNet et à l'intérieur d'une station compatible DeviceNet.

3.1.4**attribut**

description d'une caractéristique ou fonction d'un objet accessible en externe

Note 1 à l'article: Typiquement, les attributs fournissent des informations d'état ou régissent le fonctionnement d'un objet.

3.1.5**requête multiple**

communication utilisant la procédure d'échange sur requête multiple

3.1.6**diffusion**

communication d'une station vers toutes les autres stations

3.1.7**attribut BOI****attribut bus-off interrupt**

attribut de l'objet DeviceNet qui définit le comportement d'un appareil après un événement bus-off dans le composant CAN

Note 1 à l'article: Voir l'IEC 61158-4-2:2014, 7.7.4.4 pour des informations plus détaillées.

3.1.8**CAN**

spécification ISO qui définit une couche physique générique et une procédure d'accès au support pour la couche liaison de données, utilisant un arbitrage sur bits non destructif

Note 1 à l'article: Voir ISO 11898-1 et ISO 11898-2.

Note 2 à l'article: CAN est l'abréviation de «controller area network», c'est-à-dire «gestionnaire de réseau de communication»

3.1.9**CAN_H**

moitié positive du signal physique différentiel CAN

3.1.10**CAN_L**

moitié négative du signal physique différentiel CAN

3.1.11**client**

(1) objet qui utilise les services d'un autre objet (serveur) pour accomplir une tâche;

(2) initiateur d'un message auquel un serveur réagit

Note 1 à l'article: Voir serveur (3.1.44).

[SOURCE: IEC 61158-4-2:2014, 3.4.9, MODIFIÉE]

3.1.12

service commun

service CIP utilisé par les objets DeviceNet

Note 1 à l'article: Voir l'IEC 61158-5-2:2014, 6.2.1.3 et l'IEC 61158-6-2:2014, 4.1.8.

3.1.13

objets de communication

objets qui assurent la gestion et la production des messages échangés sur DeviceNet en phase d'opération

3.1.14

connexion

lien logique entre deux ou plusieurs objets applicatifs

Note 1 à l'article: Ces objets applicatifs peuvent se trouver dans une même station ou dans des stations différentes.

3.1.15

CID

ID de connexion

identificateur de connexion attribué à toutes les transmissions associées à une connexion particulière entre plusieurs stations

3.1.16

objet connexion

gère les fonctions spécifiques à la communication associées aux connexions entre stations

3.1.17

consommateur

extrémité d'une connexion responsable de la réception de données

3.1.18

MAC ID de destination

MAC ID d'une station destinataire d'un message

3.1.19

dérivateur d'appareil

point physique de raccordement d'un appareil DeviceNet à un câble principal ou à un câble de dérivation

3.1.20

type d'appareil

identification d'une série d'informations dépendant de l'appareil et décrivant une combinaison appropriée d'options sélectionnées dans toutes les couches de la pile de communication

3.1.21

dominant

l'un des deux niveaux logiques complémentaires du signal physique

Note 1 à l'article: Le niveau dominant est un "0" logique.

3.1.22

détection de duplication d'adresse MAC ID

protocole défini par DeviceNet qui garantit qu'un même MAC ID ne sera pas attribué à deux stations sur la même liaison

3.1.23**messagerie explicite**

chaque message explicite commande l'exécution d'une tâche particulière et le renvoi des résultats correspondants au demandeur

3.1.24**fragmentation**

protocole DeviceNet fourni par l'objet connexion et qui définit une méthode grâce à laquelle les données d'une longueur supérieure à huit (8) octets peuvent être transmises

3.1.25**client du groupe 2**

appareil disposant de la fonction gestionnaire de messages non connecté (UCMM, unconnected message manager) et ayant obtenu la propriété du jeu de connexions maître/esclave prédéfini d'un serveur, de telle sorte qu'il peut agir en tant que client sur ces connexions

3.1.26**client du groupe 2 seulement**

appareil agissant comme un client du groupe 2 envers un serveur du groupe 2 seulement

Note 1 à l'article: Le client du groupe 2 seulement fournit la fonctionnalité UCMM aux serveurs du groupe 2 seulement qui lui sont alloués.

3.1.27**serveur du groupe 2**

appareil ayant la capacité de gestionnaire de messages non connecté (UCMM), configuré pour agir comme serveur des connexions maître/esclave prédéfinies

3.1.28**serveur du groupe 2 seulement**

appareil esclave privé de la fonction UCMM et qui utilise le jeu de connexions maître/esclave prédéfini pour établir des communications

Note 1 à l'article: Un appareil du groupe 2 seulement ne peut transmettre et recevoir qu'avec les identificateurs spécifiés par le jeu de connexions maître/esclave prédéfini.

3.1.29**connexion d'E/S**

connexion entre un producteur et un ou plusieurs consommateurs dans le but d'échanger des données d'E/S à temps critique spécifiques à une application

3.1.30**données d'E/S**

informations transférées entre des points d'E/S et les appareils de commande qui utilisent et définissent ces valeurs

3.1.31**messagerie d'E/S**

échange de données dans un format précédemment défini

3.1.32**appareil isolé**

appareil dans lequel certains composants n'ont pas comme référence le V- de la couche physique

Note 1 à l'article: Voir appareil non isolé (3.1.37).

3.1.33

maître

station qui rassemble et distribue les E/S en utilisant le jeu de connexions maître/esclave prédéfini

3.1.34

MAC ID

adresse d'une station DeviceNet pour la couche liaison de données

Note 1 à l'article: MAC ID est l'abréviation de «medium access control ID», c'est-à-dire «identificateur de contrôle d'accès au support».

3.1.35

connexion multidestinataire

connexion logique entre un objet et plusieurs autres objets

Note 1 à l'article: Une connexion multidestinataire permet le transfert des données en une seule transaction entre un producteur et de multiples consommateurs partageant la même connexion.

3.1.36

station

entité DeviceNet identifiée au niveau de la couche liaison de données par un MAC ID unique

Note 1 à l'article: Plusieurs stations DeviceNet peuvent être intégrées à un même appareil mais elles apparaissent sur la liaison DeviceNet comme des stations logiquement distinctes.

3.1.37

appareil non isolé

appareil dans lequel tous les composants ont pour référence le V- de la couche physique

Note 1 à l'article: Voir appareil isolé (3.1.32).

3.1.38

objet

(1) représentation abstraite des capacités d'un appareil;

Note 1 à l'article: Les objets peuvent être composés de certains ou de tous les composants suivants:

- a) Données (informations qui changent avec le temps);
 - b) Configuration (paramètres définissant le comportement);
 - c) Procédures (actions qui peuvent être effectuées à l'aide de données et d'une configuration).
- (2) ensemble cohérent de données (sous forme de variables) et de procédures opératoires sur ces données.

3.1.39

connexion point à point

connexion qui existe entre deux objets seulement

Note 1 à l'article: Les connexions de messagerie explicite sont toujours point à point. Les connexions d'E/S peuvent être point à point ou multidestinataire, voir connexion multidestinataire (3.1.35).

3.1.40

jeu de connexions maître/esclave prédéfini

utilisation d'une connexion de messagerie explicite et d'une connexion de messagerie d'E/S pour créer et configurer des objets connexion prédéfinie à chaque extrémité d'une connexion

Note 1 à l'article: Utilise les règles générales comme base pour définir un jeu de connexions qui facilite les communications types d'une relation maître/esclave.

3.1.41

producteur

extrémité d'une connexion responsable de l'envoi de données

3.1.42**récessif**

l'un des deux niveaux logiques complémentaires du signal physique

Note 1 à l'article: Le niveau récessif est un "1" logique.

3.1.43**numéro de série**

nombre entier unique de 32 bits attribué par chaque fabricant à chaque appareil DeviceNet

Note 1 à l'article: Le nombre est stocké dans l'appareil comme attribut de l'objet identité; il est unique pour le fabricant.

3.1.44**serveur**

objet qui fournit des services à un autre objet (client)

Note 1 à l'article: Voir client (3.1.11).

[SOURCE: IEC 61158-4-2:2014, 3.4.62, MODIFIEE]

3.1.45**service**

opération ou fonction qu'un objet effectue à la demande d'un autre objet

3.1.46**esclave**

station qui reçoit des données en provenance de son maître et lui retourne des données en utilisant le jeu de connexions maître/esclave prédéfini et une méthode de communication déterminée par le maître

3.1.47**MAC ID source**

MAC ID d'une station émettrice d'un message

3.1.48**déclenchement**

service utilisé par une application pour lancer la production de données

3.1.49**appareil avec capacité UCMM**

appareil disposant de la fonctionnalité UCMM

Note 1 à l'article: Voir 3.1.52.

3.1.50**appareil sans capacité UCMM**

appareil privé de la fonctionnalité UCMM

Note 1 à l'article: Voir 3.1.52.

3.1.51**message explicite non connecté**

message explicite entre stations n'ayant pas encore établi de connexion entre elles

3.1.52**UCMM****gestionnaire de messages non connecté**

fonction de réception et de traitement de messages explicites non connectés dans une station

3.1.53**USINT**

nombre entier de 8 bits

Note 1 à l'article: USINT est l'abréviation de «unsigned short integer», c'est-à-dire «nombre entier court non signé»

3.1.54**UINT**

nombre entier de 16 bits

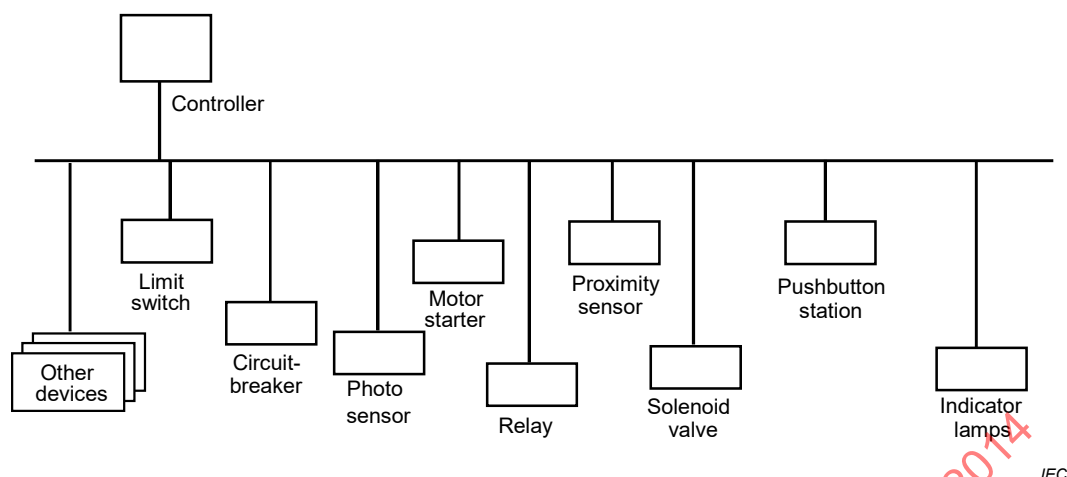
Note 1 à l'article: UINT est l'abréviation de «unsigned integer», c'est-à-dire «nombre entier non signé».

3.2 Symboles et termes abrégés

BOI	Bus-off interrupt
CAN	Gestionnaire de réseau de communication (controller area network)
CDI	Interface appareil de commande-appareil (controller-device interface, voir IEC 62026-1)
CID	Identificateur de connexion (connection identifier)
CIP™	Common Industrial Protocol
CRC	Contrôle de redondance cyclique (cyclic redundancy check)
DEL	Diode électroluminescente
DCR	Résistance en courant continu (DC resistance)
EUT	Appareil en essai (equipment under test)
MAC ID	Identificateur de contrôle d'accès au support (medium access control identifier)
UCMM	Gestionnaire de messages non connecté (unconnected message manager)
UINT	Nombre entier non signé (unsigned integer)
USINT	Nombre entier court non signé (unsigned short integer)

4 Classification**4.1 Généralités**

DeviceNet définit un système d'interface entre appareils de commande et appareils ou éléments de commutation pour circuits de commande. DeviceNet utilise deux paires conductrices réunies dans un câble – l'une des paires fournit un support de communication différentiel et l'autre alimente les appareils. Le courant maximal accepté est de 8 A à 24 V c.c. Les données sont transmises aux vitesses de 125 kbit/s, 250 kbit/s ou 500 kbit/s avec des câbles de longueurs maximales de 500 m, 250 m et 100 m respectivement. Un maximum de 8 octets de données peut être transmis sans fragmentation. Soixante-quatre stations maximum peuvent être connectées à l'aide d'une topologie linéaire comprenant une ligne principale et des lignes de dérivation (voir Figure 1). DeviceNet permet la transmission de données d'E/S, de diagnostics, de messages et d'informations pour la programmation et la configuration. Les échanges de données peuvent être de nature événementielle (changement d'état), cyclique, sur interrogation, multiples ou sur interrogation multidestinataire.



Légende

Anglais	Français
Controller	Appareil de commande
Limit switch	Détecteur de fin de course
Other devices	Autres appareils
Circuit-breaker	Disjoncteur
Photo sensor	Capteur photoélectrique
Motor starter	Démarrateur
Relay	Relais
Proximity sensor	Détecteur de proximité
Solenoid valve	Valve solénoïde
Pushbutton station	Panneau de boutons-poussoirs
Indicator lamps	Lampes témoins

Figure 1 – Interfaces appareil de commande-appareil DeviceNet types

La présente partie de l'IEC 62026 définit un système basé sur des connexions, afin de faciliter toutes les communications entre applications. Une connexion DeviceNet fournit une voie de communication entre ses extrémités multiples. Les extrémités d'une connexion sont des applications qui ont besoin de partager des données. Les transmissions associées à une connexion particulière reçoivent une valeur d'identification lorsque cette connexion est établie. Cette valeur d'identification est appelée ID de la connexion (CID).

Les objets connexion modélisent les caractéristiques de communication d'une relation particulière entre applications.

Le système de connexions DeviceNet définit les moyens dynamiques par lesquels les deux types suivants de connexions peuvent être établis:

- **Connexions d'E/S:** Fournissent des voies de communication dédiées, spécifiques, entre une application productrice et une ou plusieurs applications consommatrices. Les données d'E/S spécifiques à l'application sont transférées par ces voies.

Les messages d'E/S sont échangés via des connexions d'E/S. Un message d'E/S consiste en un CID et des données d'E/S associées. Les extrémités de la connexion doivent avoir connaissance de l'utilisation prévue ou de la signification du message d'E/S.

La présente partie de l'IEC 62026 ne définit aucune utilisation particulière de la messagerie d'E/S. De très nombreuses fonctions peuvent être accomplies avec la messagerie d'E/S. La signification et/ou l'utilisation prévue de tous les messages d'E/S

doivent être portées à la connaissance du système, soit par le type particulier de produit transmettant un message d'E/S, soit à partir de la configuration effectuée en utilisant la messagerie explicite.

- **Connexions de messagerie explicite:** Fournissent des voies de communication génériques tous usages entre deux appareils. Les messages explicites assurent les communications du type requête/réponse.

Les messages explicites sont échangés via des connexions de messagerie explicite. Ils sont utilisés pour commander l'exécution d'une tâche particulière et pour en rapporter les résultats.

DeviceNet définit un protocole de messagerie explicite qui indique la signification du message. Un message explicite consiste en un CID et des informations pour le protocole de messagerie associé.

Les règles qui régissent l'établissement dynamique de ces connexions servent de bases à partir desquelles un jeu prédéfini de connexions est déterminé.

4.2 Modèle de communication DeviceNet

Le modèle abstrait de communication orienté objet d'une station DeviceNet comprend les éléments suivants:

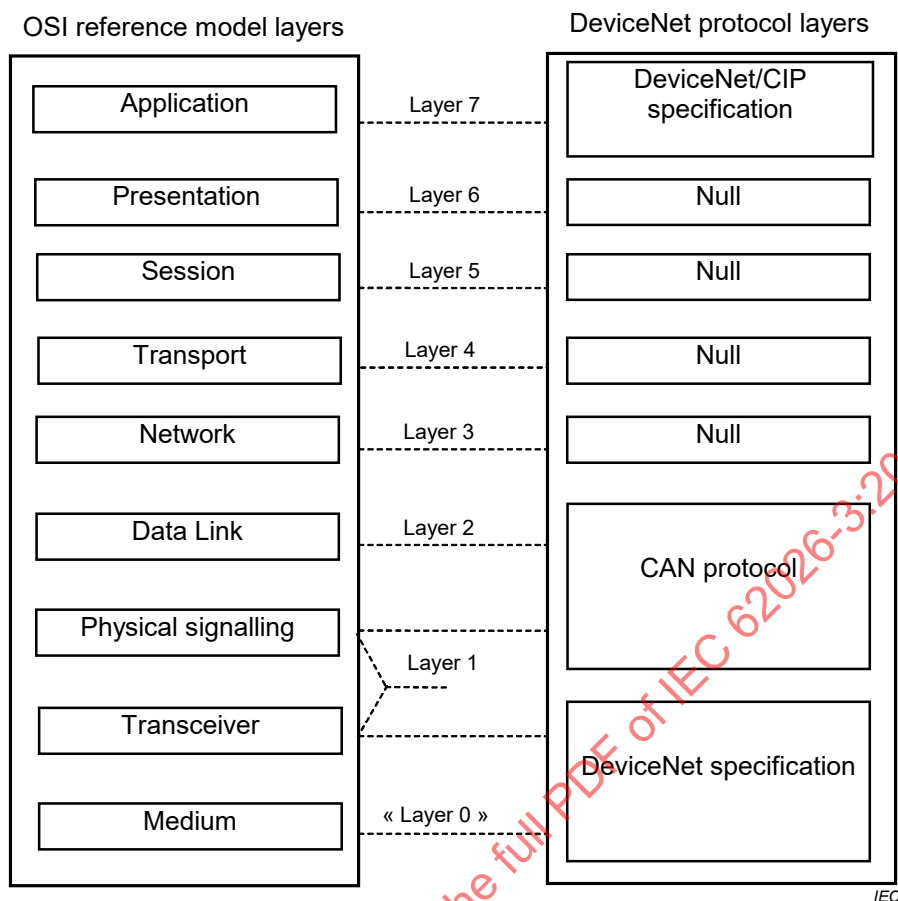
- **gestionnaire de messages non connecté (UCMM):** traite les messages DeviceNet explicites non connectés;
- **objet identité:** identifie et fournit des informations générales sur l'appareil;
- **classe connexion:** attribue et gère les ressources internes associées aux connexions d'E/S et de messagerie explicite;
- **objet connexion:** gère les aspects spécifiques de la communication associés à une relation particulière entre applications;
- **objet DeviceNet:** fournit la configuration et l'état d'un CDI DeviceNet physique;
- **routeur de messages:** transmet un message explicite de requête à l'objet approprié;
- **objets applicatifs:** réalisent les fonctions prévues pour le produit.

4.3 DeviceNet, CAN et CIP™

Les couches inférieures de DeviceNet sont basées sur l'ISO 11898-1 et l'ISO 11898-2, et utilisent la technologie CAN (Gestionnaire de réseau de communication, Controller Area Network). Les couches supérieures de DeviceNet utilisent un sous-ensemble du protocole industriel commun (CIP™³) et des services spécifiés dans la série IEC 61158.

Les correspondances entre DeviceNet, CAN, CIP et le modèle de référence OSI (ISO/IEC 7498-1:1994) sont illustrées sur la Figure 2.

³ CIP™ est une marque commerciale d'ODVA, Inc. Cette information est donnée à l'intention des utilisateurs de la présente Norme internationale et ne signifie nullement que l'IEC approuve ou recommande le détenteur de cette marque commerciale ou d'un quelconque de ses produits. La conformité à ce profil ne nécessite pas l'utilisation de la marque commerciale CIP™. L'utilisation de la marque commerciale CIP™ nécessite l'autorisation d'ODVA, Inc.



Légende

Anglais	Français
OSI reference model layers	Couches du modèle de référence OSI
DeviceNet protocol layers	Couches du protocole DeviceNet
Presentation	Présentation
Network	Réseau
Data link	Liaison de données
Physical signaling	Signalisation physique
Transceiver	Transcepteur
Medium	Support
Layer	Couche
DeviceNet/CIP specification	Spécification DeviceNet/CIP
Null	Vide
CAN protocol	Protocole CAN
DeviceNet specification	Spécification DeviceNet

Figure 2 – Comparaison entre l'architecture du protocole DeviceNet et le modèle de référence OSI

5 Caractéristiques

5.1 Connexions DeviceNet

5.1.1 Généralités

DeviceNet est une interface appareil de commande-appareil basée sur les connexions. Les connexions DeviceNet sont utilisées pour offrir des voies logiques entre plusieurs applications. Lorsqu'une connexion est établie, les transmissions associées reçoivent un ID de la connexion (CID). Si la connexion implique un échange bidirectionnel, deux valeurs CID doivent être attribuées à la connexion.

DeviceNet utilise le champ identificateur de CAN et définit les étapes à suivre dans l'établissement dynamique des connexions d'E/S et de messagerie explicite.

5.1.2 Utilisation du champ identificateur de CAN par DeviceNet

DeviceNet utilise le champ identificateur de CAN de 11 bits ("Format de trame standard") décrit dans l'ISO 11898-1.

Les identificateurs de CAN généraux disponibles dans DeviceNet sont répartis entre quatre groupes de messages séparés: groupe 1, groupe 2, groupe 3 et groupe 4.

Pour les messages de connexion, le CID est placé dans le champ identificateur de CAN. La Figure 3 décrit également les éléments d'un CID DeviceNet.

IDENTIFIER BITS											HEX RANGE	IDENTITY USAGE	
10	9	8	7	6	5	4	3	2	1	0			
0	Group 1 message ID				Source MAC ID						0x000 – 0x3FF	Message group 1	
1	0	MAC ID					Group 2 message ID				0x400 – 0x5FF	Message group 2	
1	1	Group 3 message ID			Source MAC ID						0x600 – 0x7BF	Message group 3	
1	1	1	1	1	Group 4 message ID (0 – 0x2F)						0x7C0 – 0x7EF	Message group 4	
1	1	1	1	1	1	1	1	X	X	X	X	0x7F0 – 0x7FF	Invalid CAN identifiers
10	9	8	7	6	5	4	3	2	1	0			

IEC

Légende

Anglais	Français
IDENTIFIER BITS	BITS DE L'IDENTIFICATEUR
Group 1 message ID	ID de message du groupe 1
Group 2 message ID	ID de message du groupe 2
Group 3 message ID	ID de message du groupe 3
Source MAC ID	MAC ID source
Group 4 message ID (0 – 0x2F)	ID de message du groupe 4 (0 – 0x2F)
HEX RANGE	PLAGE HEXA
IDENTITY USAGE	UTILISATION DE L'IDENTIFICATEUR
Message group	Groupe de messages
Invalid CAN identifiers	Identificateurs de CAN incorrects

Figure 3 – Utilisation du champ identificateur de CAN par DeviceNet

Comme le montre la Figure 3, le champ identificateur de CAN de DeviceNet contient les éléments suivants:

- **ID de message:** Identifie un message au sein d'un groupe de messages dans une station particulière. Lorsqu'une connexion est établie, les stations utilisent un ID de message combiné avec un MAC ID pour générer un CID. Le CID qui en résulte est placé dans le champ identificateur de CAN associé aux transmissions correspondantes;
- **MAC ID source:** Les groupes 1 et 3 imposent de spécifier un MAC ID source dans le champ identificateur de CAN;
- **MAC ID:** Le groupe 2 de message permet de spécifier la source ou la destination dans la partie MAC ID du champ identificateur de CAN.

Les deux types de connexions, de messagerie explicite et d'E/S, peuvent être établis en utilisant les groupes de messages 1, 2 et 3.

Dans le groupe 2, les valeurs 6 et 7 d'ID de message sont utilisées de la manière suivante:

- l'ID de message 6 du groupe 2 est utilisé pour la configuration des communications utilisées dans une application maître/esclave (voir 5.5);
- l'ID de message 7 du groupe 2 est utilisé pour la détection de stations ayant reçu des MAC ID identiques (voir 5.3.6).

Dans le groupe 3, les valeurs 5, 6 et 7 d'ID de message sont utilisées comme suit:

- l'ID de message 5 du groupe 3 est utilisé lors de l'envoi de réponses associées à des messages explicites non connectés de requête, ainsi que des messages de cadenceur et de déconnexion d'appareil;
- l'ID de message 6 du groupe 3 est utilisé lors de l'envoi de messages explicites non connectés de requête;
- l'ID de message 7 du groupe 3 n'est pas valide et ne doit pas être utilisé.

Dans le groupe 4, les valeurs d'ID de message sont utilisées de la manière suivante:

- les ID des messages du groupe 4 0x00 à 0x2B sont réservés à une utilisation ultérieure et ne doivent pas être utilisés;
- l'ID de message du groupe 4 0x2C est utilisé pour les messages de réponse de défaut de communication;
- l'ID de message du groupe 4 0x2D est utilisé pour les messages de requête de défaut de communication;
- l'ID de message du groupe 4 0x2E est utilisé pour les messages de réponse de propriété hors ligne;
- l'ID de message du groupe 4 0x2F est utilisé pour les messages de requête de propriété hors ligne.

5.1.3 Établissement des connexions

5.1.3.1 Connexions de messagerie explicite et UCMM

Le groupe de messages 3 définit le codage des identificateurs pour les messages explicites non connectés. Les messages explicites non connectés établissent et gèrent les connexions de la messagerie explicite. Les messages de requête non connectés sont spécifiés par la transmission d'un message du groupe 3 dont l'élément ID de message est fixé à 6. Les seuls services valides pouvant être transmis via des messages de requête explicites non connectés sont les suivants:

- requête d'ouverture de connexion de messagerie explicite;
- requête de fermeture de connexion.

Les messages énumérés ci-dessus ne sont jamais transmis comme des messages de connexion (voir messagerie explicite basée sur une connexion en 5.2.1.6).

Les réponses aux requêtes explicites non connectées sont transmises comme messages de réponse non connectés. Les messages de réponse non connectés sont spécifiés par la transmission d'un message du groupe 3 dont l'élément ID de message est fixé à 5. Les seuls services valides pouvant être transmis via des messages de réponse explicites non connectés sont les suivants:

- réponse à une ouverture de connexion de messagerie explicite;
- réponse à une fermeture de connexion;
- réponse d'erreur;
- message de cadenceur d'appareil;
- message de déconnexion d'appareil.

5.1.3.2 Connexions d'E/S

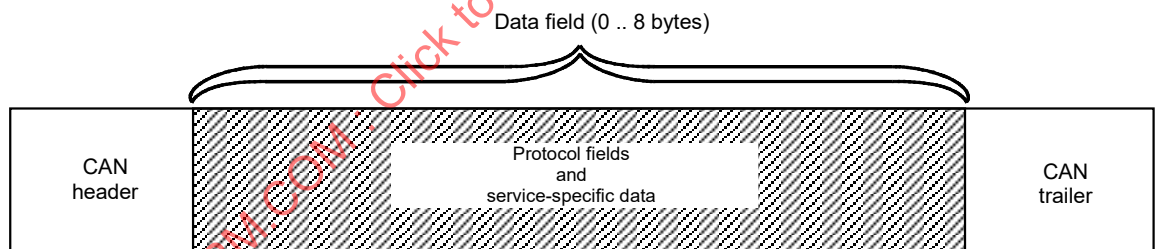
Les connexions d'E/S sont établies dynamiquement par l'intermédiaire de la classe d'objet connexion via une connexion de messagerie explicite préalablement établie. Une instance d'objet connexion doit être créée et configurée dans chaque station.

5.2 Protocole de messagerie DeviceNet

5.2.1 Messagerie explicite

5.2.1.1 Généralités

Le présent paragraphe décrit le protocole de messagerie explicite et présente les détails associés à l'établissement dynamique des connexions de la messagerie explicite (voir Figure 4).



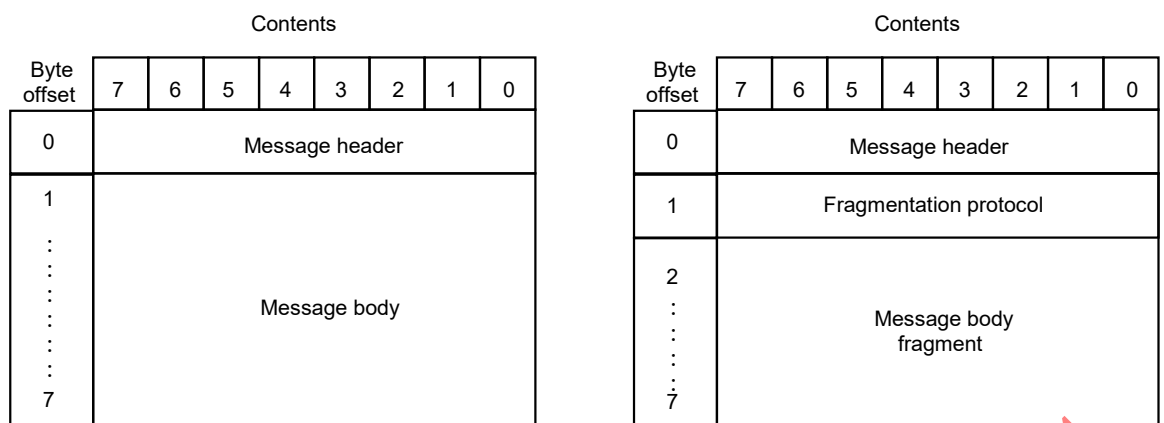
IEC

Légende

Anglais	Français
Data field (0 .. 8 bytes)	Champ de données (0 .. 8 octets)
CAN header	En-tête de CAN
Protocol fields and service-specific data	Champs de protocole et données spécifiques au service
CAN trailer	Amorce de fin de CAN

Figure 4 – Utilisation du champ de données CAN dans un message explicite

La Figure 5 indique le format du champ de données CAN associé aux messages explicites:



IEC

Légende

Anglais	Français
Byte offset	Rang de l'octet
Contents	Contenu
Message header	En-tête du message
Message body	Corps du message
Fragmentation protocol	Protocole de fragmentation
Message body fragment	Fragment du corps du message

Figure 5 – Format du champ de données des messages explicites

Le champ de données d'une transmission qui contient la totalité du message explicite comprend:

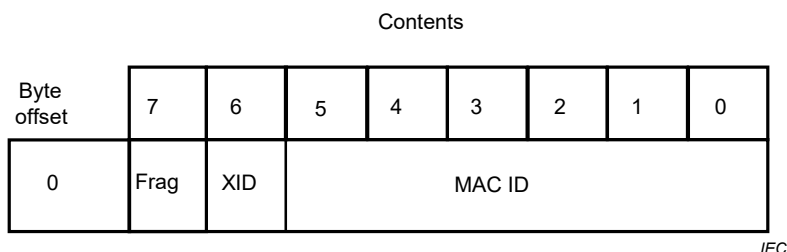
- un en-tête de message;
- la totalité du corps du message.

Si le message explicite est d'une longueur supérieure à huit octets, il doit être transmis par fragments. La fonction de fragmentation/réassemblage est fournie par l'objet connexion. Un tel fragment de message explicite comprend:

- un en-tête de message;
- le protocole de fragmentation (voir 5.2.3.2);
- un fragment du corps du message.

5.2.1.2 En-tête de message

L'en-tête de message se trouve dans l'octet de rang zéro du champ de données CAN d'un message explicite et doit être formaté comme indiqué sur la Figure 6:



Légende

Anglais	Français
Byte offset	Rang de l'octet
Contents	Contenu

Figure 6 – Format de l'en-tête d'un message explicite

Contenu de l'en-tête du message:

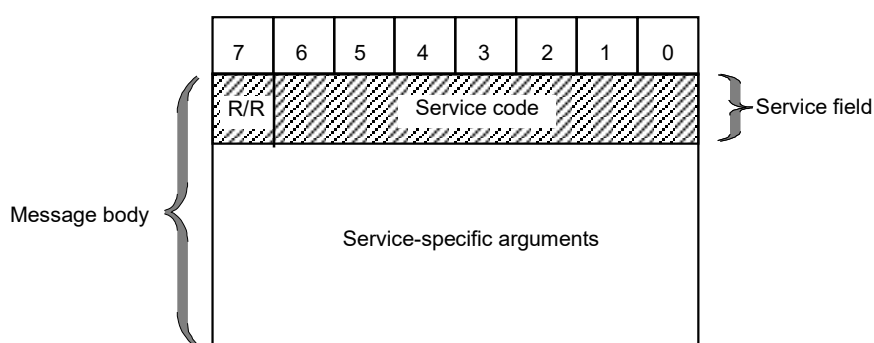
- **frag** (bit de fragment): Indique si cette transmission est ou n'est pas une partie d'un message explicite fragmenté.
0 = non fragmenté
1 = fragmenté
- **XID (ID de la transaction)**: Ce champ est utilisé par une application pour faire correspondre une réponse à la requête associée. Le serveur se contente de renvoyer le contenu de ce champ dans le message de réponse. Le serveur n'utilise pas ce champ pour effectuer une quelconque logique de détection de messages dupliqués;
- **MAC ID**: Contient le MAC ID source ou de destination.

Lorsqu'un message explicite est reçu, le champ MAC ID de l'en-tête du message est examiné. Si le MAC ID de destination est spécifié dans le CID, le MAC ID source de l'autre extrémité doit être spécifié dans l'en-tête du message. Si le MAC ID source est spécifié dans le CID, le MAC ID de la station de réception doit être spécifié dans l'en-tête du message. Si aucune de ces conditions n'est remplie, le message doit être rejeté.

5.2.1.3 Corps du message

Le corps du message se compose d'un champ de service et d'arguments spécifiques au service.

Le premier argument spécifié dans le corps du message est le champ de service, qui identifie la requête ou la réponse particulière délivrée par ce message. La Figure 7 illustre le format du champ de service.



IEC

Légende

Anglais	Français
Service code	Code de service
Message body	Corps du message
Service field	Champ de service
Service-specific arguments	Arguments spécifiques au service

Figure 7 – Format du champ de service

Contenu du champ de service:

- **code de service:** valeur spécifiée dans les 7 bits de poids faible de l'octet du champ de service et qui indique le type de service transmis;
- **R/R:** bit de poids fort du champ de service. Sa valeur détermine si le message est une requête ou une réponse.
 0 = requête
 1 = réponse

DeviceNet définit un ensemble de services communs. Ces services communs de DeviceNet sont l'ensemble des services publics dont les paramètres et comportements requis sont définis dans l'IEC 61158-5-2:2014, 6.2.1.3 et l'IEC 61158-6-2:2014, 4.1.8.

Les informations qui suivent le champ de service sont spécifiques au type particulier de service transmis.

5.2.1.4 Protocole de fragmentation

Si la transmission est une portion d'un message explicite fragmenté, le champ de données doit contenir l'en-tête du message, le protocole de fragmentation et un fragment du corps du message. Le protocole de fragmentation facilite la fragmentation et le réassemblage des messages explicites dont le contenu dépasse 8 octets (voir 5.2.3).

5.2.1.5 Services UCMM**5.2.1.5.1 Généralités**

Le gestionnaire de messages non connecté (UCMM) assure l'établissement dynamique des connexions de messagerie explicite. Le présent paragraphe présente une description détaillée des arguments spécifiques aux services d'ouverture et de fermeture de connexions de messagerie explicite fournis par l'UCMM. Voir 9.3.3 pour les spécifications d'essais concernant l'établissement des connexions de messagerie explicite.

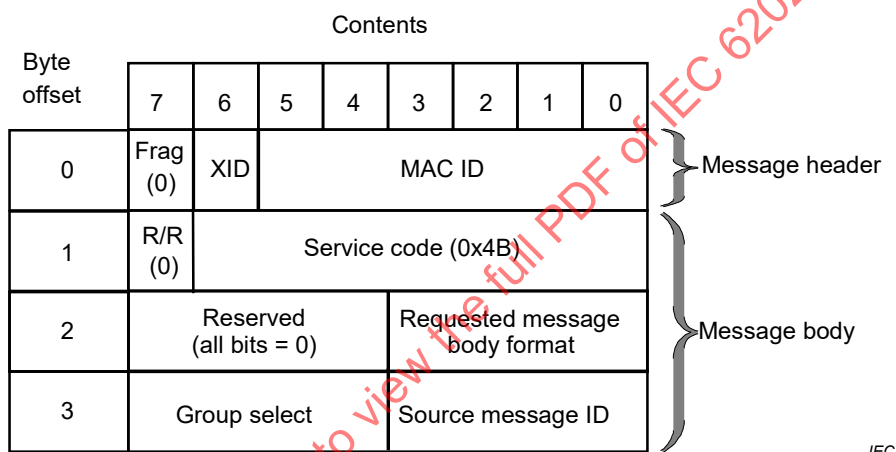
L'UCMM traite deux services qui gèrent l'affectation et la désaffectation des connexions de messagerie explicite:

- **ouverture d'une connexion de messagerie explicite:** code de service = 0x4B. Utilisé pour établir une connexion de messagerie explicite;
- **fermeture de connexion:** code de service = 0x4C. Utilisé pour supprimer un objet connexion et libérer toutes les ressources associées.

L'accès à ces services s'effectue à l'aide des champs identificateurs de CAN de requête et de réponse explicites non connectées au sein d'un groupe de messages 3, comme indiqué en 5.1.2. Lors de l'exécution d'une requête explicite non connectée, l'UCMM peut avoir besoin de retourner une indication d'erreur au demandeur et, par suite, le message de réponse explicite d'erreur peut être transmis à l'aide de l'identificateur de CAN de réponse explicite non connectée.

5.2.1.5.2 Requête d'ouverture de connexion de messagerie explicite

Ce service demande l'établissement d'une connexion logique entre deux stations par laquelle les messages explicites seront transmis. Ce service est transmis en tant que message de requête non connecté (groupe de messages 3, ID de message 6) (voir Figure 8).



Légende

Anglais	Français
Byte offset	Rang de l'octet
Contents	Contenu
Service code (0x4B)	Code de service (0x4B)
Message header	En-tête du message
Message body	Corps du message
Reserved (all bits = 0)	Réservé (tous les bits = 0)
Requested message body format	Format requis pour le corps du message
Group select	Sélection de groupe
Source message ID	ID de message source

Figure 8 – Format de requête d'ouverture de connexion de messagerie explicite

Contenu de la requête d'ouverture de connexion de messagerie explicite:

- **frag (0)/ID de transaction/MAC ID:** Voir 5.2.1.2. Le MAC ID de destination est toujours spécifié dans l'en-tête du message associé à une requête/réponse d'ouverture de connexion de messagerie explicite;
- **bit R/R (0):** Indique qu'il s'agit d'un message de requête;
- **code de service (0x4B):** Identifie ce message en tant que service d'ouverture de connexion de messagerie explicite;

- **bits réservés:** Ces bits ne doivent pas être pris en compte par le récepteur et doivent être mis à zéro par le transmetteur;
- **format requis pour le corps du message:** Champ utilisé par le client pour demander un format particulier de corps de message applicable aux messages explicites transmis ultérieurement via cette connexion. Un serveur doit accepter au moins un des formats de corps de message DeviceNet (valeurs 0 – 3).

La station serveur répondant à cette requête d'ouverture de connexion de messagerie explicite choisit le format réel de corps de message à utiliser sur cette connexion. Voir Tableau 1 pour les valeurs de format de corps de message. Les serveurs doivent agir de l'une ou l'autre des façons suivantes:

- refuser la requête et renvoyer une valeur de format acceptée dans la réponse d'ouverture de connexion de messagerie explicite; le format renvoyé doit être un des formats de corps de message DeviceNet (valeurs 0 – 3);
- accepter cette requête en renvoyant la même valeur de format dans la réponse d'ouverture de connexion de messagerie explicite;

Tableau 1 – Valeurs de format de corps de message

Valeur	Signification
0	DeviceNet (8/8). ID de classe = nombre entier de 8 bits, ID d'instance = nombre entier de 8 bits
1	DeviceNet (8/16). ID de classe = nombre entier de 8 bits, ID d'instance = nombre entier de 16 bits
2	DeviceNet (16/16). ID de classe = nombre entier de 16 bits, ID d'instance = nombre entier de 16 bits
3	DeviceNet (16/8). ID de classe = nombre entier de 16 bits, ID d'instance = nombre entier de 8 bits
4	Chemin d'accès CIP. La taille d'adressage est variable et est donnée comme un chemin électronique prêt à l'emploi à chaque requête ^a
5 – 0xF	Réservé
NOTE Les messages transmis via cette connexion sont formatés comme décrit en 5.2.1.6.	
^a Le chemin d'accès CIP est spécifié comme une longueur de chemin d'accès de 8 bits (USINT), suivie du chemin d'accès (chemin électronique prêt à l'emploi). Voir l'IEC 61158-6-2:2014, 4.1.9, qui donne des informations sur le chemin électronique. Le serveur doit accepter, au minimum, le codage le plus efficace pour tout segment logique dans le chemin électronique (par exemple, codage de classe de 8 bits pour le code de classe 0x01).	

- **sélection de groupe:** Champ qui indique le groupe de messages à utiliser pour échanger les messages associés à cette connexion. Les valeurs suivantes sont présentées dans le Tableau 2:

Tableau 2 – Valeurs de sélection des groupes

Valeur	Signification
0	Groupe de messages 1
1	Groupe de messages 2 ^a
2	Réservé
3	Groupe de messages 3
4 – 0xE	Réservé
0xF	Réservé pour message ping de station ^b
^a L'identificateur du groupe de messages 2 permet la spécification du MAC ID source ou de destination. Pour les connexions de messagerie explicite établies via le groupe de messages 2, le client place le MAC ID du serveur dans l'ID de connexion lors de la transmission de messages via cette connexion. Le serveur place son propre MAC ID dans l'ID de connexion lors de la transmission de messages via cette connexion. Cette procédure impose au serveur d'allouer deux ID de message distincts à partir de son pool de groupe 2. ^b Cette valeur de sélection des groupes peut être utilisée pour forcer une station cible à capacité UCMM à répondre sans créer de ressources dans la station. Lorsqu'une requête comportant cette valeur est reçue, la cible doit répondre par un code d'erreur général de "paramètre incorrect" (0x20) et un code d'erreur supplémentaire de 0x01.	

Le client sélectionne le groupe de messages par lequel les transmissions associées à cette connexion de messagerie explicite s'effectueront. Si le serveur ne peut pas satisfaire à cette requête, il doit alors la rejeter et renvoyer une réponse d'erreur.

- **ID de message source:** L'utilisation de ce champ dépend de la valeur dans le champ de sélection de groupe (voir Tableau 3):

Tableau 3 – ID de message source dans une requête d'ouverture de connexion de messagerie explicite

Si la sélection de groupe est égale à:	Alors l'ID de message source:
0 ou 3	Spécifie l'ID de message alloué par le client à partir du pool d'ID de messages du groupe 1 ou 3. Le client doit utiliser cet ID de message en combinaison avec son propre MAC ID pour générer l'ID de connexion spécifié lorsqu'il transmet un message via cette connexion ^a
1	Est ignoré/fixé à la valeur zéro (0) ^b
^a Le client place cette valeur dans l'élément ID de message de l'identificateur de message du groupe 1 ou 3. ^b Les connexions de messagerie explicite établies via le groupe de messages 2 imposent au serveur d'allouer deux ID de message du groupe 2 et de les renvoyer dans le message de réponse d'ouverture de connexion de messagerie explicite. Le client doit utiliser l'un de ces ID de message pour générer l'ID de connexion qu'il spécifie lors de la transmission d'un message via cette connexion. L'autre ID doit être utilisé par le serveur pour générer l'ID de connexion qu'il spécifie lors de la transmission d'un message via cette connexion.	

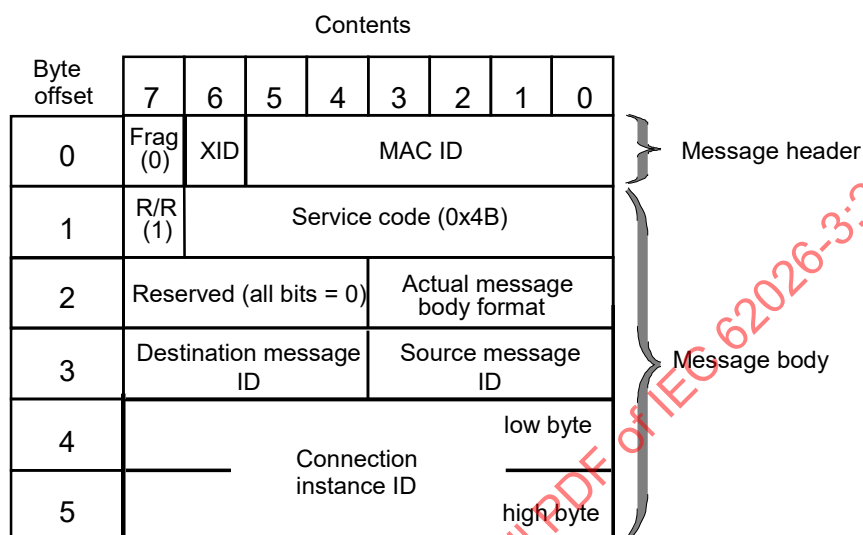
L'UCMM du serveur valide les arguments de requête d'ouverture de connexion de messagerie explicite. S'ils sont valides, l'UCMM fait appel au service création («create») de la classe connexion pour obtenir une instance d'objet connexion (voir l'IEC 61158-5-2:2014, 6.2.3). L'objet connexion qui en résulte est automatiquement configuré pour être un objet connexion de messagerie explicite.

Si le serveur accepte plusieurs formats de corps de message et si le client a demandé l'un de ces formats, le serveur doit renvoyer le format de corps de message demandé dans la réponse d'ouverture de connexion explicite. Si le serveur n'accepte pas plusieurs formats de corps de message, il doit préciser son format par défaut dans la réponse d'ouverture de connexion de messagerie explicite.

Si aucune erreur n'est détectée, une réponse positive d'ouverture de connexion de messagerie explicite doit être renvoyée. Si une erreur est détectée, un message de réponse d'erreur doit être renvoyé.

5.2.1.5.3 Réponse positive d'ouverture de connexion de messagerie explicite

Ce service est utilisé pour répondre à un message couronné de succès de requête d'ouverture de connexion de messagerie explicite (voir Figure 9).



Légende

Anglais	Français
Byte offset	Rang de l'octet
Contents	Contenu
Message header	En-tête du message
Service code (0x4B)	Code de service (0x4B)
Reserved (all bits = 0)	Réservé (tous les bits = 0)
Actual message body format	Format réel du corps du message
Message body	Corps du message
Destination message ID	ID de message de destination
Source message ID	ID de message source
Connection instance ID	ID d'instance de connexion
Low byte	Octet de poids faible
High byte	Octet de poids fort

Figure 9 – Format de la réponse d'ouverture de connexion de messagerie explicite

Contenu de la réponse d'ouverture de connexion de messagerie explicite:

- **frag (0)/ID de transaction/MAC ID:** Voir 5.2.1.2. Le MAC ID de destination est toujours spécifié dans l'en-tête du message associé à une requête/réponse d'ouverture de connexion de messagerie explicite;
- **bit R/R (1):** Indique qu'il s'agit d'un message de réponse;
- **code de service (0x4B):** Identifie ce message en tant que service d'ouverture de connexion de messagerie explicite;
- **bits réservés:** Ils doivent être mis à zéro par le transmetteur;

- **format réel du corps du message:** Champ utilisé par le serveur pour définir le format de corps de message applicable aux messages explicites transmis ultérieurement via cette connexion (comme décrit dans le Tableau 1);
- **ID de message de destination:** L'utilisation de ce champ dépend du groupe de messages via lequel la connexion s'effectue (comme décrit dans le Tableau 4);

Tableau 4 – ID de message de destination dans une réponse d'ouverture de connexion de messagerie explicite

Si la sélection de groupe dans la requête d'ouverture était:	L'ID de message de destination de la réponse ouverte:
0 ou 3	Est ignoré et doit être mis à la valeur zéro (0)
1	Est utilisé par le client en combinaison avec le MAC ID du serveur pour générer l'ID de connexion qu'il spécifie lors de transmissions via cette connexion

- **ID de message source:** Valeur de l'ID de message alloué par le serveur. Le serveur alloue un ID de message à partir de son pool d'ID de messages du groupe 1, 2 ou 3. Cet ID de message est utilisé conjointement avec son propre MAC ID (MAC ID source) pour générer l'ID de connexion spécifié lorsqu'il transmet un message via cette connexion;
- **ID d'instance de connexion:** Le serveur crée un objet connexion de messagerie explicite lorsqu'il traite avec succès une requête d'ouverture. Ce champ contient la valeur d'ID d'instance (nombre entier de 16 bits) attribuée à cet objet connexion de messagerie explicite.

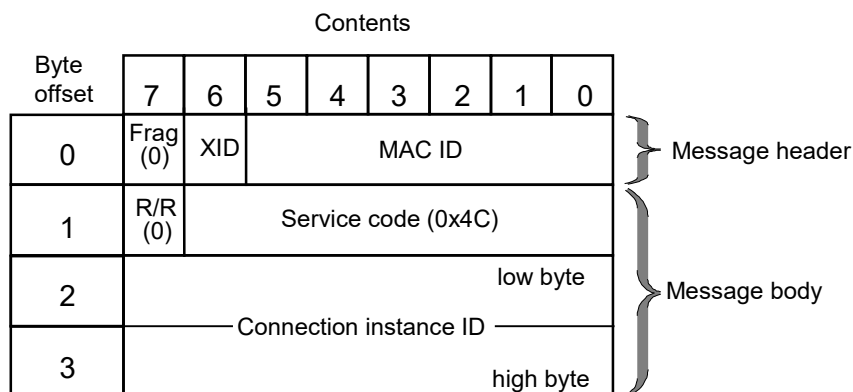
5.2.1.5.4 Requête de fermeture de connexion

Ce service est utilisé pour mettre fin à une connexion (d'E/S ou de messagerie) à l'une de ses extrémités. La réception du message de requête de fermeture de connexion par l'UCMM provoque l'exécution du service suppression («delete») de la classe de connexion (voir l'IEC 61158-5-2:2014, 6.2.3). Un message de requête de fermeture de connexion est transmis en tant que message de requête non connecté.

Le répondeur vérifie que l'instance de connexion spécifiée existe. Si elle existe et si elle peut être supprimée, elle doit alors être supprimée. Toutes les ressources associées à l'instance de connexion sont libérées. Si la requête est exécutée avec succès, une réponse de fermeture de connexion doit être renvoyée. Si la requête échoue, une réponse d'erreur doit être renvoyée.

Contenu de la requête de fermeture de connexion (voir Figure 10):

- **frag (0)/ID de transaction/MAC ID:** Voir 5.2.1.2;
- **bit R/R (0):** Indique qu'il s'agit d'un message de requête;
- **code de service (0x4C):** Identifie qu'il s'agit du service de fermeture de connexion;
- **ID d'instance de connexion:** Champ qui spécifie l'instance de connexion à supprimer. Le format de l'ID d'instance de connexion est toujours spécifié dans ce message en tant que nombre entier de 16 bits.



IEC

Légende

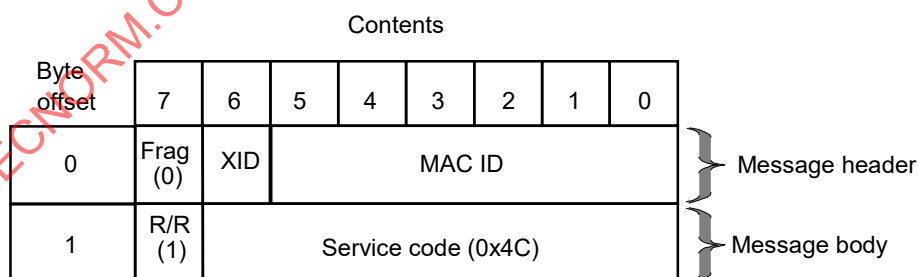
Anglais	Français
Byte offset	Rang de l'octet
Contents	Contenu
Service code (0x4C)	Code de service (0x4C)
Message header	En-tête du message
Message body	Corps du message
Low byte	Octet de poids faible
Connection instance ID	ID d'instance de connexion
High byte	Octet de poids fort

Figure 10 – Format de la requête de fermeture de connexion**5.2.1.5.5 Réponse de fermeture de connexion**

Ce service est utilisé pour répondre à un message couronné de succès de requête de fermeture de connexion.

Contenu du format de la réponse de fermeture de connexion (voir Figure 11):

- **frag (0)/ID de transaction/MAC ID:** Voir 5.2.1.2;
- **bit R/R (1):** Indique qu'il s'agit d'un message de réponse;
- **code de service (0x4C):** Identifie qu'il s'agit du service de fermeture de connexion.



IEC

Légende

Anglais	Français
Byte offset	Rang de l'octet
Contents	Contenu
Service code (0x4C)	Code de service (0x4C)
Message header	En-tête du message
Message body	Corps du message

Figure 11 – Format de la réponse de fermeture de connexion

5.2.1.5.6 Réponse d'erreur

Le Tableau 5 représente un ensemble standard de conditions d'erreur liées à l'UCMM et les codes d'erreur correspondants à utiliser dans les messages de réponse d'erreur (codes d'erreur généraux et supplémentaires). Le format d'un message de réponse d'erreur est décrit en 5.2.1.6.5.

Tableau 5 – Conditions/codes d'erreur UCMM

Condition de l'erreur	Nom d'erreur général	Code d'erreur général	Code d'erreurs supplémentaire
Code de service ni ouverture ni fermeture	Service non accepté	0x08	0xFF
Erreur de ressource (sélection de groupe)	Ressource non disponible	0x02	0x01
Sélection de groupe hors limites	Paramètre incorrect	0x20	0x01
Aucune connexion de serveur disponible	Ressource non disponible	0x02	0x02
Aucun ID de message serveur disponible	Ressource non disponible	0x02	0x03
ID de message source client incorrect	Paramètre incorrect	0x20	0x02
ID de message source client dupliqué	Ressource non disponible	0x02	0x04
ID d'instance de connexion incorrect	L'objet n'existe pas	0x16	0xFF

Descriptions des conditions d'erreur:

- **service ni ouverture ni fermeture:** Un service reçu via l'accès UCMM n'est ni de type ouverture ni de type fermeture et n'est donc pas accepté par l'UCMM;
- **erreur de ressource (sélection de groupe):** L'argument de sélection de groupe spécifie un groupe de messages qui n'est pas accepté par l'appareil;
- **sélection de groupe hors limites:** Le champ de sélection de groupe contient une valeur incorrecte;
- **aucune connexion de serveur disponible:** Le nombre maximal de connexions acceptées par ce serveur a déjà été atteint;
- **aucun ID de message serveur disponible:** Le serveur a alloué tous les ID de message disponibles dans le groupe de messages demandé par le client;
- **ID de message source client incorrect:** L'ID de message source reçu dans une requête d'ouverture de connexion de messagerie explicite est incorrect pour le groupe de messages spécifié;
- **ID de message source client dupliqué:** L'ID de message source et le MAC ID source reçus dans une requête d'ouverture de connexion de messagerie explicite sont déjà utilisés pour une connexion de messagerie explicite du groupe 1 ou 3;
- **ID d'instance de connexion incorrect:** L'ID d'instance de connexion reçu dans la requête de fermeture de connexion n'existe pas.

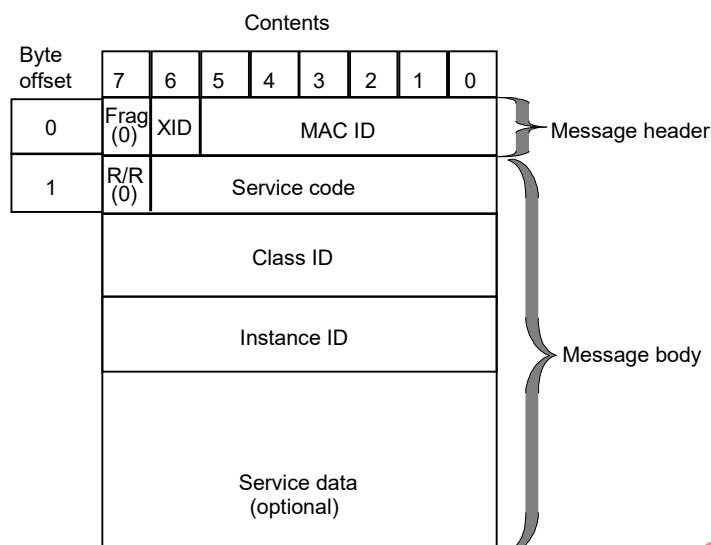
5.2.1.6 Messagerie explicite basée sur une connexion

5.2.1.6.1 Généralités

Un message explicite basé sur une connexion est un message transmis via une connexion de messagerie explicite. Un message explicite basé sur une connexion doit se conformer aux formats décrits dans le présent paragraphe.

5.2.1.6.2 Contenu du message de requête explicite (valeurs de format de corps de message 0 – 3)

La Figure 12 indique le format du message associé à une requête explicite non fragmentée pour les valeurs de format de corps de message 0 – 3:



Légende

Anglais	Français
Byte offset	Rang de l'octet
Contents	Contenu
Message header	En-tête du message
Service code	Code de service
Class ID	ID de classe
Instance ID	ID d'instance
Message body	Corps du message
Service data (optional)	Données du service (en option)

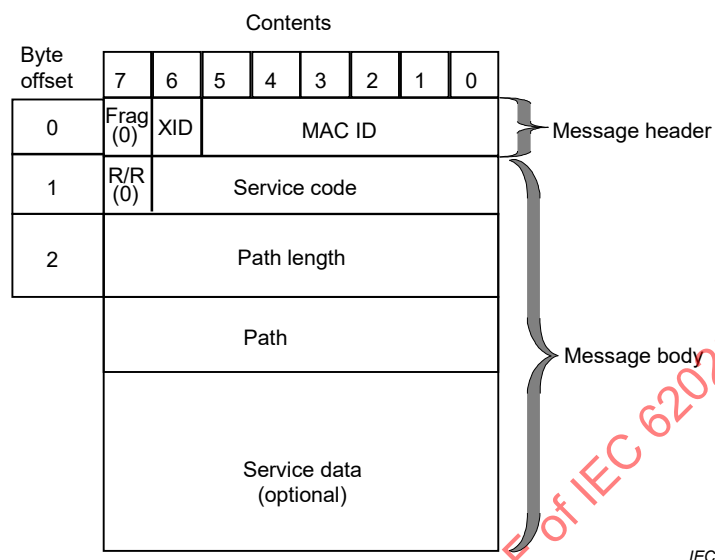
Figure 12 – Format d'un message de requête explicite non fragmenté, valeurs 0 – 3

Contenu d'un message de requête explicite non fragmenté:

- **frag (0)/ID de transaction/MAC ID:** Voir 5.2.1.2;
- **bit R/R (0):** Indique qu'il s'agit d'un message de requête;
- **code de service:** Définit le service demandé;
- **ID de classe:** Définit la classe d'objet vers laquelle cette requête est dirigée. L'ID de classe est spécifié par un nombre entier de 8 bits ou 16 bits en fonction de la valeur effective du format de corps de message renvoyée dans la réponse d'ouverture de connexion de messagerie explicite;
- **ID d'instance:** Définit l'instance particulière dans la classe d'objet vers laquelle cette requête est dirigée. L'ID d'instance est spécifié par un nombre entier de 8 bits ou 16 bits en fonction de la valeur effective du format de corps de message renvoyée dans la réponse d'ouverture de connexion de messagerie explicite. DeviceNet utilise la valeur zéro pour indiquer que la requête est dirigée vers la classe elle-même plutôt que vers une instance particulière de la classe;
- **données du service:** Ce champ contient les données spécifiques à la requête. Pour les services communs de DeviceNet, les formats sont décrits dans l'IEC 61158-6-2:2014, 4.1.8. Les définitions des services spécifiques à une classe et à un objet comprennent le format particulier de ce champ.

5.2.1.6.3 Contenu du message de requête explicite (valeur de format de corps de message 4)

La Figure 13 indique le format du message associé à une requête explicite non fragmentée pour la valeur de format de corps de message 4:



Légende

Anglais	Français
Byte offset	Rang de l'octet
Contents	Contenu
Message header	En-tête du message
Message body	Corps du message
Service code	Code de service
Path length	Longueur du chemin d'accès
Path	Chemin d'accès
Service data (optional)	Données du service (en option)

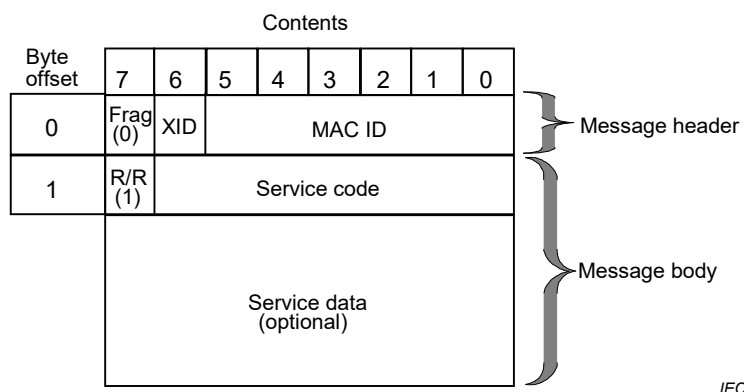
Figure 13 – Format d'un message de requête explicite non fragmenté, valeur 4

Contenu d'un message de requête explicite non fragmenté:

- **frag (0)**/ID de transaction/MAC ID: Voir 5.2.1.2;
- **bit R/R (0)**: Indique qu'il s'agit d'un message de requête;
- **code de service**: Définit le service demandé;
- **longueur du chemin d'accès**: Cette valeur de nombre entier de 8 bits (USINT) donne la longueur en octets du chemin d'accès du message de requête;
- **chemin d'accès**: Définit le chemin d'accès du message de requête (chemin électronique prêt à l'emploi);
- **données du service**: Ce champ contient les données spécifiques à la requête. Pour les services communs de DeviceNet, les formats sont décrits dans l'IEC 61158-6-2:2014, 4.1.8. Les définitions des services spécifiques à une classe et à un objet comprennent le format particulier de ce champ.

5.2.1.6.4 Message de réponse explicite positive

La Figure 14 indique le format du message associé à une réponse positive non fragmentée:

**Légende**

Anglais	Français
Byte offset	Rang de l'octet
Contents	Contenu
Message header	En-tête du message
Message body	Corps du message
Service code	Code de service
Service data (optional)	Données du service (en option)

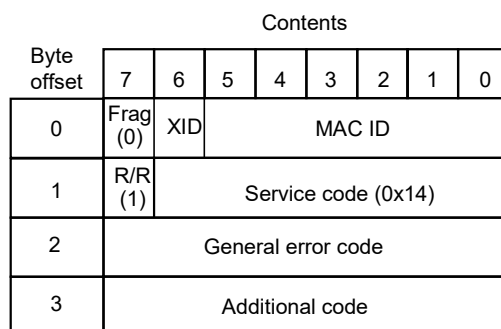
Figure 14 – Format d'un message de réponse positive non fragmentée

Contenu d'un message de réponse positive non fragmentée:

- **frag (0)/ID de transaction/MAC ID:** Voir 5.2.1.2;
- **bit R/R (1):** Indique qu'il s'agit d'un message de réponse;
- **code de service:** Contient le code de service envoyé dans le message de requête;
- **données du service:** Ce champ contient les données spécifiques à la requête.

5.2.1.6.5 Message de réponse d'erreur

Le message de réponse d'erreur est renvoyé lors de la survenue d'une erreur au cours de l'exécution d'un service demandé via un message de requête explicite reçu précédemment. La réponse d'erreur peut être envoyée soit comme message de réponse basé sur une connexion (requête reçue via une connexion de messagerie explicite), soit comme message de réponse non connecté (la requête était un message de requête explicite non connecté). La Figure 15 indique le format d'un message de réponse d'erreur.

**Légende**

Anglais	Français
Byte offset	Rang de l'octet

Anglais	Français
Contents	Contenu
Service code (0x14)	Code de service (0x14)
General error code	Code d'erreurs général
Additional code	Code supplémentaire

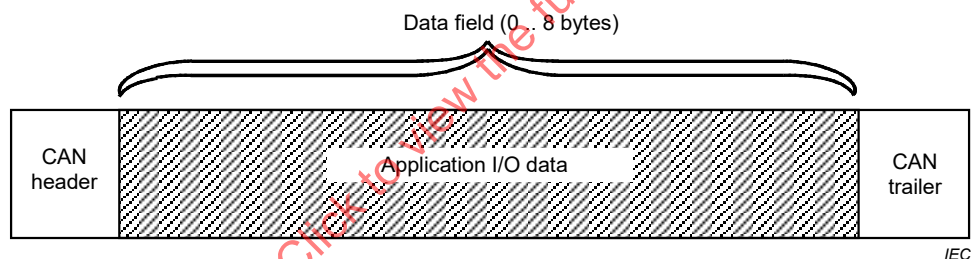
Figure 15 – Message de réponse d'erreur

Contenu d'un message de réponse d'erreur:

- **frag (0)/ID de transaction/MAC ID:** Voir 5.2.1.2;
- **bit R/R (1):** Indique qu'il s'agit d'un message de réponse;
- **code de service (0x14):** Identifie ce message en tant que réponse d'erreur;
- **code d'erreur général:** Identifie l'erreur rencontrée. Voir l'IEC 61158-5-2:2014, 6.2.1.3.3 et l'IEC 61158-6-2:2014, 4.1.11, pour la liste des codes d'erreur généraux;
- **code supplémentaire:** Contient une valeur spécifique à l'objet ou au service qui précise la condition d'erreur. Si l'objet qui répond n'a pas d'informations supplémentaires à spécifier, la valeur 0xFF doit être placée dans ce champ.

5.2.2 Messagerie d'entrée/sortie

DeviceNet définit un protocole de fragmentation pour le transfert de messages d'E/S d'une longueur supérieure à huit octets. Ce sont les seules informations de protocole contenues dans le champ de données d'un message d'E/S (voir Figure 16).



Légende

Anglais	Français
Data field (0 .. 8 bytes)	Champ de données (0 .. 8 octets)
CAN header	En-tête de CAN
Application I/O data	Données d'E/S d'application
CAN trailer	Amorce de fin de CAN

Figure 16 – Champ de données d'un message d'E/S

5.2.3 Fragmentation/réassemblage

5.2.3.1 Généralités

Le présent paragraphe définit le moyen par lequel un message dont la longueur est supérieure à huit octets est fragmenté et réassemblé. La fonction de fragmentation / réassemblage est fournie par l'objet connexion de DeviceNet.

La logique qui déclenche une transmission fragmentée est la suivante:

- les instances de l'objet connexion de messagerie explicite examinent la longueur de chaque message à transmettre. Si le message est d'une longueur supérieure à huit octets, le protocole de fragmentation doit être utilisé;

- les instances de l'objet connexion d'E/S examinent l'attribut `produced_connection_size` (taille de connexion produite) de l'objet connexion (voir 5.3.2). Si cet attribut est supérieur à huit, le protocole de fragmentation doit être utilisé.

Deux types de fragmentation sont définis:

- **acquitté**: utilisé lors de la fragmentation d'un message explicite;
- **non acquitté**: utilisé lors de la fragmentation d'un message d'E/S.

5.2.3.2 Protocole de fragmentation

Le protocole de fragmentation est situé dans un seul octet du champ de données de CAN; il est formaté comme indiqué à la Figure 17:

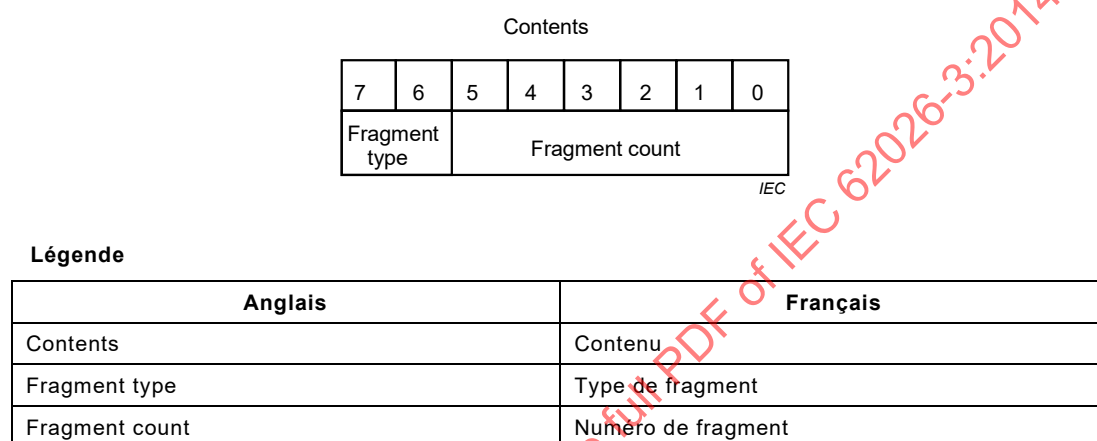


Figure 17 – Format du protocole de fragmentation DeviceNet

Contenu du protocole de fragmentation:

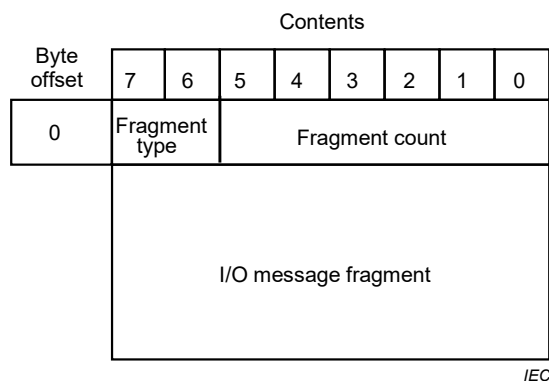
- **type de fragment**: Indique s'il s'agit du premier fragment, d'un fragment du milieu ou du dernier fragment (voir Tableau 6).

Tableau 6 – Valeurs des bits du type de fragment

Valeur	Signification
0	Premier fragment. Le champ numéro de fragment doit contenir la valeur 0 ou 0x3F ^a
1	Fragment du milieu ^b
2	Dernier fragment ^c
3	Acquittement de fragment ^d
^a Si le numéro de fragment contient la valeur zéro (0), il s'agit du premier d'une série de fragments. Si le champ numéro de fragment contient la valeur 0x3F, il s'agit également de la dernière transmission de la série. ^b Ce fragment n'est ni le premier ni le dernier de la série. ^c Indique que ce fragment est le dernier. ^d Valeur utilisée par le récepteur d'un message fragmenté pour acquitter la réception d'un fragment.	

- **numéro de fragment**: Marque chaque fragment individuel de sorte que le récepteur peut déterminer si un fragment manque. Si le type de fragment indique qu'il s'agit du premier fragment, ce champ a une signification spéciale (comme décrit au Tableau 6). Le numéro de fragment est augmenté de un à chaque fragment successif d'une série; il se remet à zéro en cas de dépassement (numéro de fragment = (numéro de fragment + 1) mod 64).

Pour la fragmentation d'un message d'E/S, les informations de protocole de fragmentation sont placées dans l'octet de rang 0 (voir Figure 18).

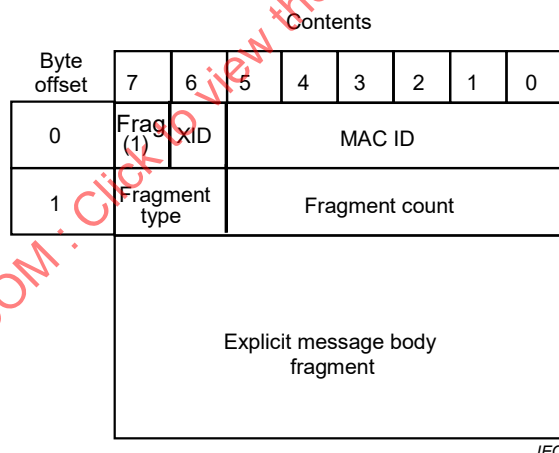


Légende

Anglais	Français
Contents	Contenu
Fragment type	Type de fragment
Byte offset	Rang de l'octet
Fragment count	Numéro de fragment
I/O message fragment	Fragment de message d'E/S

Figure 18 – Format d'un fragment de message d'E/S

Pour une fragmentation de message explicite, les informations de protocole de fragmentation sont placées dans l'octet de rang 1 (voir Figure 19).



Légende

Anglais	Français
Contents	Contenu
Byte offset	Rang de l'octet
Fragment type	Type de fragment
Fragment count	Numéro de fragment
Explicit message body fragment	Fragment du corps d'un message explicite

Figure 19 – Format d'un fragment de message explicite

Le bit de fragmentation se trouvant dans l'en-tête de message de la Figure 19 est mis à 1 pour indiquer qu'il s'agit d'un fragment du message explicite et non de tout le message. Dans ce cas, la valeur 1 indique également que l'octet suivant contient le protocole de fragmentation.

Le récepteur d'une transmission fragmentée analyse le message fragmenté comme défini dans le présent paragraphe. Cette procédure s'applique aux deux types de fragmentation: messages d'E/S et messages explicites.

Si la connexion est en attente de la première transmission et si le type de fragment reçu est égal au premier fragment:

- si le numéro de fragment est 0x3F, alors il s'agit de la seule transmission de la série; la connexion traite le message et attend le début d'une nouvelle série;
- si le numéro de fragment est 0, alors il s'agit de la première d'une série de transmissions; la connexion stocke le fragment et enregistre le numéro de fragment.

Si la connexion est en attente de la première transmission et si le type de fragment n'est pas égal au premier fragment ou si le numéro de fragment est différent de 0 ou de 0x3F, la connexion ignore la transmission et attend le début d'un nouveau message.

Si la connexion n'est pas en attente de la première transmission, la connexion vérifie alors:

- que le type de fragment n'est pas le premier fragment; et
- que le numéro de fragment est numériquement supérieur d'une unité (1) à la précédente valeur reçue.

Si l'une de ces vérifications échoue, alors une erreur a été détectée. Si les deux vérifications réussissent, le fragment est ajouté à celui ou ceux reçus antérieurement, et le type de fragment est analysé pour déterminer s'il faut attendre ou non d'autres fragments.

Si d'autres fragments sont sur le point d'arriver, alors la connexion enregistre le numéro de fragment reçu et attend le suivant. S'il s'agit de la dernière transmission de la série et qu'il ne s'est pas produit d'erreur, la connexion traite le message et se réinitialise en attente du début d'une nouvelle série.

Si un message explicite est fragmenté, le récepteur doit générer et transmettre un acquittement à la réception de chaque fragment.

Si une erreur est détectée, il y a lieu d'effectuer une correction d'erreur en fonction du type de message (d'E/S fragmenté ou explicite).

Si la détection d'un fragment manquant a été déclenchée par la réception du premier fragment de la série suivante, tout traitement associé à la série en cours est alors immédiatement arrêté, les fragments stockés en mémoire sont ignorés et le traitement commence immédiatement sur la nouvelle série.

5.2.3.3 Fragmentation non acquittée

La fragmentation d'un message d'E/S s'effectue de façon non acquittée. La fragmentation non acquittée consiste en une transmission des fragments l'un après l'autre depuis l'unité de transmission. La ou les unités de réception ne renvoient pas d'acquittement.

Lorsque le service `send_message` (envoyer message) d'une connexion d'E/S est invoqué, celle-ci examine son attribut `connection_size` (taille de la connexion), afin de déterminer s'il faut transmettre une série de messages fragmentés. Si l'attribut `connection_size` est supérieur à huit (8), le protocole de fragmentation est alors placé dans le message d'E/S.

Si l'attribut `connection_size` est inférieur ou égal à huit (8) octets, les données brutes sont transmises sans la présence du protocole de fragmentation.

Si l'application demande la transmission d'un bloc de données dont la longueur est supérieure à l'attribut `connection_size`, une erreur est indiquée en interne et la transmission n'a pas lieu.

Si l'objet connexion d'E/S récepteur détecte un fragment manquant, c'est-à-dire si le numéro de fragment reçu n'est pas égal à celui reçu antérieurement plus un (1), les mesures de correction d'erreur suivantes sont entreprises:

- tous les fragments ultérieurs de cette suite sont ignorés et l'application n'est pas informée de la réception d'un message d'E/S;
- l'objet connexion attend le début d'une nouvelle série de transmissions fragmentées et ignore les fragments restants.

5.2.3.4 Fragmentation acquittée

La fragmentation acquittée est utilisée pour les messages explicites fragmentés. La fragmentation acquittée comprend la transmission d'un fragment provenant de la station émettrice, suivie de la transmission d'un acquittement par la station réceptrice. La station réceptrice acquitte la réception de chaque fragment.

La Figure 20 indique le format du message d'acquittement transmis par le récepteur après la réception de chaque fragment de message explicite.

Byte offset	Contents							
	7	6	5	4	3	2	1	0
0	Frag (1)	XID	MAC ID					
1	Fragment type (3)		Fragment count					
2	Ack status							

IEC

Légende

Anglais	Français
Contents	Contenu
Byte offset	Rang de l'octet
Fragment type (3)	Type (3) de fragment
Fragment count	Numéro de fragment
Ack status	État d'acquittement

Figure 20 – Format d'un message d'acquittement

Contenu d'un message d'acquittement:

- **type de fragment:** Indique qu'il s'agit d'un acquittement de fragment en plaçant la valeur 3 dans ce champ;
- **numéro de fragment:** Retourne la dernière valeur du numéro de fragment reçu;
- **état d'acquittement:** Indique si une erreur a été détectée ou non par le récepteur du message fragmenté. Les valeurs du bit d'état d'acquittement sont définies dans le Tableau 7:

Tableau 7 – Valeurs du bit d'état d'acquittement

Valeur	Signification
0	Réussite. Aucune erreur n'a été détectée et la transmission fragmentée doit continuer
1	Trop de données. La quantité maximale de données pouvant être reçues par le récepteur via cette connexion a été dépassée
2 – 255	Réservé

La station de transmission fonctionne de la manière suivante:

- l'objet connexion construit l'en-tête du message en activant le bit de fragmentation (mise à 1), en plaçant dans le champ XID (ID de transaction) la valeur spécifiée par l'application, et initialise le champ MAC ID de la même manière que pour un message explicite non fragmenté. Les en-têtes de message associés à chaque fragment individuel sont identiques;
- l'objet connexion place ensuite les informations de protocole de fragmentation appropriées dans le message. Il stocke le numéro de fragment inséré dans le message;
- puis l'objet connexion prend la partie de corps de message suivante et la place dans le message;
- le message est transmis et attend que le temporisateur d'acquittement démarre. Le délai d'attente dépend de l'application;
- si le délai d'attente du temporisateur d'acquittement expire, l'objet connexion relance automatiquement la dernière transmission. Il essaie encore une fois. Si le temporisateur expire une nouvelle fois (délai d'attente expiré par deux fois consécutives), l'application est informée de la survenue d'une erreur et de l'impossibilité d'effectuer la transmission demandée;
- si un acquittement est reçu, l'objet connexion détermine si le numéro de fragment de l'acquittement est égal ou non au dernier numéro de fragment transmis. S'il s'agit du même numéro, le fragment a bien été transmis et acquitté, et le traitement normal continue. Si les valeurs sont différentes, l'objet connexion continue à attendre l'acquittement du numéro de fragment correspondant.

L'état initial associé à l'unité de réception lui impose d'attendre soit le premier fragment d'une transmission fragmentée, soit la réception d'un message explicite complet. Le côté réception fonctionne de la manière suivante:

- Si l'en-tête de message indique qu'il s'agit d'une partie fragmentée d'un message explicite, l'objet connexion examine le protocole de fragmentation afin de déterminer sa validité. Si la connexion n'a pas encore reçu la première transmission de la série (état initial) et si le champ type de fragment n'indique pas premier fragment, le fragment est alors ignoré et aucun acquittement n'est renvoyé;
- Si le type de fragment indique premier fragment, le numéro de fragment doit être égal à zéro (0) ou 0x3F. Si le numéro de fragment est 0x3F, alors il s'agit de la seule transmission de la série et la connexion:
 - traite le message;
 - se réinitialise en attente du début d'une nouvelle série.

Si le numéro de fragment est zéro (0), alors il s'agit de la première d'une série de transmissions; la connexion stocke le fragment et enregistre le numéro de fragment. S'il s'agit du premier fragment et si le numéro de fragment n'est pas égal à zéro (0) ou 0x3F, le fragment est alors ignoré et aucun acquittement n'est renvoyé;

- Si le numéro de fragment est supérieur de un (1) au numéro précédemment reçu et si le type de fragment n'indique pas premier fragment, alors le fragment suivant a été reçu. Ce fragment est ajouté à celui ou ceux reçus antérieurement et un acquittement est renvoyé. Le numéro de fragment associé à ce fragment est enregistré;

- Si le numéro de fragment n'est ni supérieur de un (1), ni égal au numéro reçu antérieurement, le fragment est ignoré et aucun acquittement n'est renvoyé. Le récepteur se remet à l'état initial;
- Lorsque le dernier fragment est reçu et l'acquiescement transmis, l'objet connexion continue le traitement du message comme s'il n'était pas fragmenté.

5.2.4 Jeu de connexions hors ligne

5.2.4.1 Généralités

Le présent paragraphe décrit le protocole de messagerie du jeu de connexions hors ligne et présente les détails associés à l'établissement de la propriété du jeu de connexions hors ligne. Le support du jeu de connexions hors ligne est facultatif pour tous les types d'appareils. Les messages du jeu de connexions hors ligne du groupe 4 sont utilisés par l' (les) outil(s) client(s) pour corriger les stations dans l'état de défaut de communication. En utilisant les messages du jeu de connexions hors ligne, un (outil) client doit être capable:

- a) d'identifier visuellement la (les) station(s) défectueuse(s) avec laquelle (lesquelles) il communique en faisant clignoter une DEL;
- b) d'envoyer des messages de correction des défauts à la station défectueuse; et, si possible
- c) de corriger la station défectueuse sans avoir à la débrancher du sous-réseau.

À tout instant, seule une (1) station client doit communiquer avec les stations dans l'état de défaut de communication, reliée à un seul sous-réseau. La propriété des stations de défaut de communication est acquise par l'intermédiaire d'un dialogue entre les (outils) clients utilisant les messages de requête/réponse de propriété hors ligne.

Le Tableau 8 présente les identificateurs du groupe 4 associés au jeu de connexions hors ligne.

Tableau 8 – Jeu de connexions hors ligne

Bits d'identificateur (ID de connexion)											Utilisation d'identité
ID de groupe					ID de message						
10	9	8	7	6	5	4	3	2	1	0	Messages du groupe 4
1	1	1	1	1			0x2C				Message de réponse de défaut de communication
1	1	1	1	1			0x2D				Message de requête de défaut de communication
1	1	1	1	1			0x2E				Message de réponse de propriété hors ligne
1	1	1	1	1			0x2F				Message de requête de propriété hors ligne

Seuls les clients qui désirent accepter le jeu de connexions hors ligne doivent produire des messages en utilisant l'ID de message 0x2F du groupe 4, et consommer les messages de réponse avec l'ID de message 0x2E du groupe 4. Une fois que la propriété est acquise, le client doit produire tous les messages destinés à la (aux) station(s) de défaut de communication à l'aide de l'ID de message 0x2D du groupe 4.

Un client peut NE PAS produire de "Message de requête de défaut de communication" jusqu'à ce qu'il ait obtenu la propriété du "Jeu de connexions hors ligne".

Une fois qu'un (outil) client établit la propriété du jeu de connexions hors ligne, il est à même de transmettre les messages de requête de défaut de communication avec l'ID de message 0x2D du groupe 4 et de recevoir les messages de réponse de défaut de communication avec l'ID de message 0x2C du groupe 4.

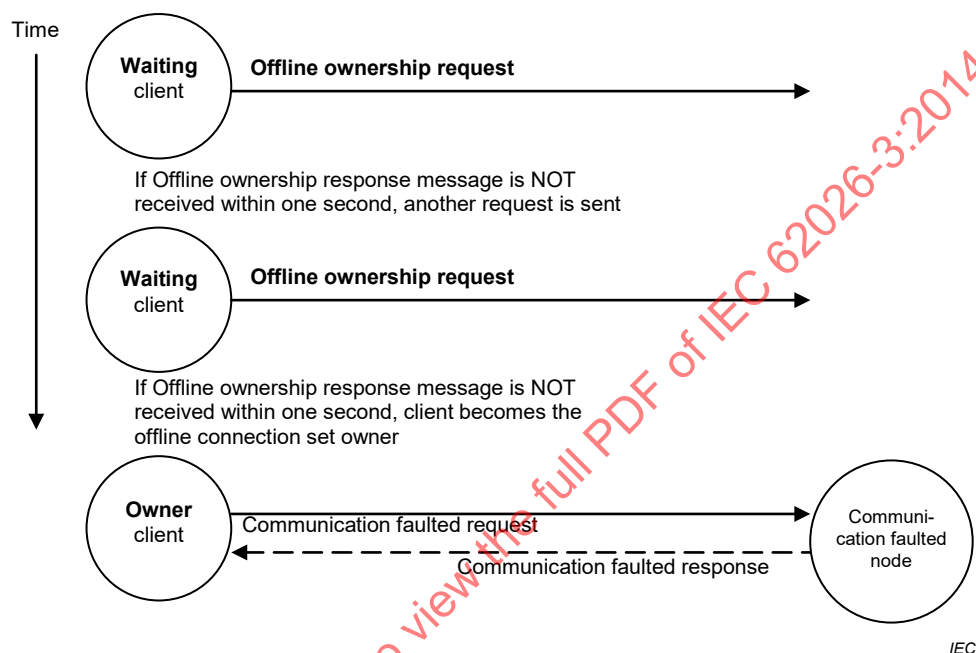
Alors qu'elle est en état de défaut de communication, il est uniquement exigé qu'une station acceptant cette caractéristique consomme un seul ID de connexion; l'ID de message 0x2D du

groupe 4. Une station défectueuse doit produire ses messages de réponse de défaut de communication sur l'ID de message 0x2C du groupe 4.

Les messages du jeu de connexions hors ligne sont de priorité faible et peuvent être soumis à des délais dus à d'autres trafics sur le réseau.

5.2.4.2 Propriété hors ligne

La Figure 21 et la Figure 22 illustrent les étapes à suivre pour un (outil) client pour obtenir la propriété du jeu de connexions hors ligne.



Légende

Anglais	Français
Time	Temps
Waiting client	Client en attente
Offline ownership request	Requête de propriété hors ligne
If Offline ownership response message is NOT received within one second, another request is sent	Si le message de réponse de propriété hors ligne n'est PAS reçu en l'espace d'une seconde, une autre requête est envoyée
If Offline ownership response message is NOT received within one second, client becomes the offline connection set owner	Si le message de réponse de propriété hors ligne n'est PAS reçu en l'espace d'une seconde, le client devient le propriétaire du jeu de connexions hors ligne
Owner client	Client propriétaire
Communication faulted request	Requête de défaut de communication
Communication faulted response	Réponse de défaut de communication
Communication faulted node	Station de communication défectueuse

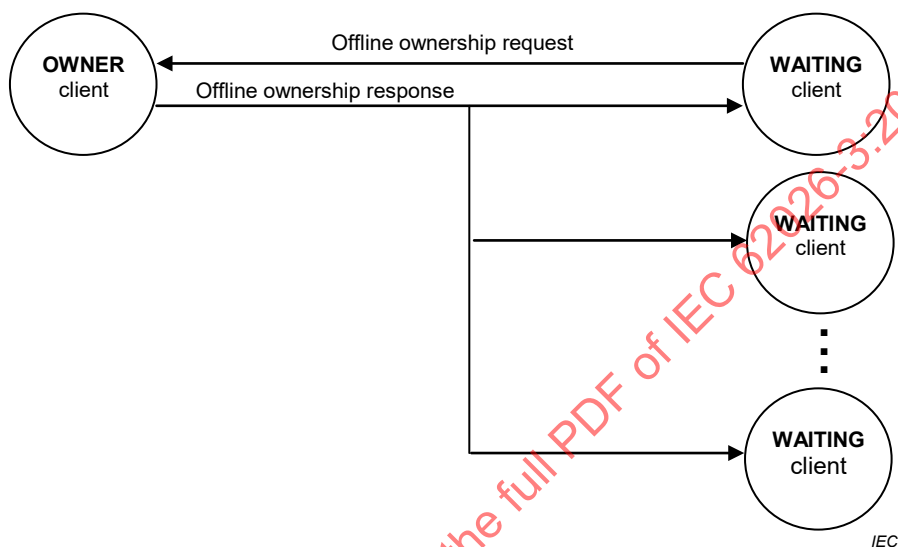
Figure 21 – Établissement de la propriété hors ligne

Pour accéder à la commande du jeu de connexions hors ligne, un (outil) client doit produire son *message de requête de propriété hors ligne*. À la suite d'une transmission réussie, l'(outil) client doit attendre un *message de réponse de propriété hors ligne* pendant 1 s. S'il ne reçoit pas de réponse, il doit produire un deuxième *message de requête de propriété hors ligne* et doit attendre encore une seconde. S'il ne reçoit pas de réponse, il doit devenir le

propriétaire du message de requête hors ligne. Si un *message de réponse de propriété hors ligne* est reçu au cours de l'une ou l'autre des périodes d'attente, il ne doit pas devenir le propriétaire du jeu de connexions hors ligne et doit attendre pour devenir le propriétaire.

Seul l'outil client doit être propriétaire du jeu de connexions hors ligne à tout instant.

Le scénario au cours duquel un (outil) client a précédemment déclaré la propriété du jeu de connexions hors ligne, et où les stations clients supplémentaires sont en train d'arbitrer pour la propriété, le propriétaire actuel doit répondre à tout *message de requête de propriété hors ligne* avec un *message de réponse de propriété hors ligne*.



Légende

Anglais	Français
OWNER client	Client PROPRIÉTAIRE
Offline ownership request	Requête de propriété hors ligne
Offline ownership response	Réponse de propriété hors ligne
WAITING client	Client EN ATTENTE

Figure 22 – Nature multidestinataire de la propriété hors ligne

Un client en attente ne doit pas tenter d'envoyer de message de requête de propriété hors ligne pendant au moins deux secondes après la réception d'un message de réponse de propriété hors ligne.

5.2.4.3 Messages de propriété hors ligne

5.2.4.3.1 Généralités

Les deux messages utilisés pour gérer le jeu de connexions hors ligne sont uniquement produits et consommés par les stations clients acceptant cette fonctionnalité. Ces messages sont uniquement traités alors qu'un client participe à une activité de correction, sinon ils sont ignorés par le client.

5.2.4.3.2 Message de requête de propriété hors ligne (client uniquement)

Pour accéder à la propriété du jeu de connexions hors ligne, un outil client doit produire un *message de requête de propriété hors ligne* avec l'ID de message 0x2F du groupe 4, voir Figure 23.

Byte Offset	Contents							
	7	6	5	4	3	2	1	0
0	Reserved [0]		Client MAC ID					
1	R/R [0]	Allocate [0x4B]						
2	Vendor ID						Low byte	
3							High byte	
4	Serial number						Low byte	
5								
6								
7							High byte	

IEC

Légende

Anglais	Français
Byte offset	Rang de l'octet
Contents	Contenu
Reserved	Réservé
Client MAC ID	MAC ID du client
Allocate [0x4B]	Attribution [0x4B]
Vendor ID	ID du fournisseur
Low byte	Octet de poids faible
High byte	Octet de poids fort
Serial number	Numéro de série

Figure 23 – Message de requête de propriété hors ligne

À la suite d'une transmission réussie, l'(outil) client doit attendre un *message de réponse de propriété hors ligne* pendant au moins 1 s. S'il ne reçoit pas de réponse, il doit produire un deuxième *message de requête de propriété hors ligne* et doit attendre au moins 1 s. S'il ne reçoit pas de réponse, il doit devenir le propriétaire du jeu de connexions hors ligne. Si un *message de réponse de propriété hors ligne* est reçu au cours de l'une ou l'autre des périodes, il ne doit pas devenir le propriétaire du jeu de connexions hors ligne et doit attendre pour devenir le propriétaire.

Une fois qu'un client est le propriétaire du jeu de connexions hors ligne, s'il reçoit un *message de requête de propriété hors ligne*, il doit produire un *message de réponse de propriété hors ligne* en l'espace de 1 s.

Si un client attend pour devenir le propriétaire du jeu de connexions hors ligne, il ne doit pas produire de *message de requête de propriété hors ligne*, à un rythme plus rapide qu'une fois toutes les 2 s. Ce délai de 2 s doit être réinitialisé à la réception de tout message à l'aide d'un *ID de connexion de requête ou de réponse de propriété hors ligne*.

5.2.4.3.3 Message de réponse de propriété hors ligne (client uniquement)

Le format du *message de réponse de propriété hors ligne* est le même que le *message de requête de propriété hors ligne*, à l'exception qu'il est produit à l'ID de message 0x2E du groupe 4, voir Figure 24, et le bit R/R est initialisé (1).

Byte offset	Contents							
	7	6	5	4	3	2	1	0
0	Reserved [0]		Client MAC ID					
1	R/R [1]	Allocate [0x4B]						
2	Vendor ID						Low byte	
3							High byte	
4	Serial number						Low byte	
5								
6								
7							High byte	

IEC

Légende

Anglais	Français
Byte offset	Rang de l'octet
Contents	Contenu
Reserved	Réservé
Client MAC ID	MAC ID du client
Allocate [0x4B]	Attribution [0x4B]
Vendor ID	ID du fournisseur
Low byte	Octet de poids faible
High byte	Octet de poids fort
Serial number	Numéro de série

Figure 24 – Protocole de message de réponse de propriété hors ligne

Une fois qu'un (outil) client est le propriétaire du jeu de connexions hors ligne, il peut produire des *messages de requête de défaut de communication* à toutes les stations défectueuses. Toute station acceptant le jeu de connexions hors ligne doit traiter le *message de requête de défaut de communication* uniquement dans l'état de défaut de communication.

5.2.4.4 Messages de défaut de communication

5.2.4.4.1 Généralités

Toutes les stations dans l'état de défaut de communication, qui acceptent le mécanisme de correction de station défectueuse, doivent recevoir les *messages de requête de défaut de communication* produits à l'ID de message 0x2D du groupe 4. Si approprié, les stations de défaut de communication doivent produire un *message de réponse de défaut de communication* à l'ID de message 0x2C du groupe 4.

Les *messages de requête de défaut de communication* sont consommés par toutes les stations défectueuses acceptant cette fonctionnalité. En fonction du message de requête, un nombre quelconque de stations défectueuses peut répondre à une seule requête.

5.2.4.4.2 Protocoles de messages de défaut de communication

Pour accepter les protocoles de *messages de défaut de communication*, une station doit accepter la DEL "état du réseau" ou la DEL "état combiné du module/réseau", ou avoir une méthode adaptée spécifique à l'appareil indiquant l'état du réseau et du module extérieurement (comme par exemple un écran d'affichage ou un affichage de texte). Elle doit également avoir un MAC ID ajustable sur le réseau.

Le protocole de tous les *messages de requête de défaut de communication* se présente de deux façons: le protocole multidestinataire (voir Figure 25) et le protocole point à point (voir Figure 26).

Byte offset	Contents							
	7	6	5	4	3	2	1	0
0	Reserved [0]	Match	Value					
1	R/R [0]	Service						

IEC

Légende

Anglais	Français
Byte offset	Rang de l'octet
Contents	Contenu
Reserved	Réservé
Match	Correspondance
Value	Valeur

Figure 25 – Message de requête de défaut de communication – Protocole multidestinataire

Le protocole multidestinataire est utilisé par un client pour effectuer des requêtes à toutes les stations de défaut de communication pour obtenir leur(s) numéro(s) de série et leur(s) ID(s) de fournisseur. Une fois que ces informations sont connues par le client, un protocole point à point peut être utilisé.

Lorsqu'une station de défaut de communication reçoit un *message de requête de défaut de communication* multidestinataire, elle doit réaliser les vérifications suivantes pour déterminer s'il convient ou non que le message soit accepté et traité:

- Protocole de correspondance multidestinataire:
 - SI le bit de correspondance est initialisé (1);
 - ET SI sa valeur est égale à son MAC ID;
 - ET SI la longueur du message est de deux (2) octets;
 - ALORS accepter et traiter le *message de requête de défaut de communication*.
- Protocole de masque multidestinataire:
 - SI le bit de correspondance est réinitialisé (0);
 - ET SI sa valeur (ET logique avec son MAC ID) est égale à son MAC ID;
 - ET SI la longueur du message est de deux (2) octets;
 - ALORS accepter et traiter le *message de requête de défaut de communication*

Le protocole de masque multidestinataire (ET logique) est utilisé pour les dialogues dans lesquels il est exigé qu'une plage de MAC ID réponde. Le Tableau 9 indique certaines, mais pas l'ensemble, des 64 valeurs de masque possibles.

Tableau 9 – Rapport d'adresses basé sur le masque

5	4	3	2	1	0	Stations défectueuses avec
1	1	1	1	1	1	MAC ID < 64
0	1	1	1	1	1	MAC ID < 32
0	0	1	1	1	1	MAC ID < 16
0	0	0	1	1	1	MAC ID < 8
0	0	0	0	1	1	MAC ID < 4
0	0	0	0	0	1	MAC ID < 2

Le protocole de masque multidestinataire est utilisé pour réduire le nombre de messages, qui doivent être envoyés sur un sous-réseau pour déterminer s'il y a une station de défaut de communication et à quel MAC ID elle se produit.

Le protocole point à point suit.

Byte offset	Contents							
	7	6	5	4	3	2	1	0
0	Reserved (service determines usage)							
1	R/R [0]	Service						
2	Vendor ID						Low byte	
3							High byte	
4	Serial number						Low byte	
5								
6								
7							High byte	

IEC

Légende

Anglais	Français
Byte offset	Rang de l'octet
Contents	Contenu
Reserved (service determines usage)	Réservé (le service détermine l'utilisation)
Vendor ID	ID du fournisseur
Low byte	Octets de poids faible
High byte	Octet de poids fort
Serial number	Numéro de série

Figure 26 – Message de requête de défaut de communication – Protocole point à point

Lorsqu'une station de défaut de communication reçoit un *message de requête de défaut de communication* point à point, elle doit réaliser les vérifications suivantes pour déterminer s'il convient ou non que le message soit accepté et traité:

Protocole de correspondance point à point:

SI la longueur du message est égale à huit (8) octets;

ET SI le numéro de série et l'ID du fournisseur correspondent à sa propre valeur;

ALORS accepter et traiter le *message de requête de défaut de communication*.

Les services suivants sont actuellement définis pour les *messages de requête de défaut de communication*.

Message de requête de défaut de communication "identification" – Protocole multidestinataire:

Utilisé lorsque le client tente d'identifier l'existence de toutes les stations de défaut de communication, qui acceptent la caractéristique du jeu de connexions hors ligne, sur un sous-réseau.

Message de requête de défaut de communication "identification" – Protocole point à point:

Utilisé lorsque le client détecte de multiples stations de défaut de communication sur un sous-réseau, et souhaite identifier visuellement une station de défaut de communication spécifique.

Message de requête de défaut de communication "qui":

Utilisé lorsque le client a détecté une (des) station(s) de défaut de communication sur un sous-réseau, et souhaite obtenir leur numéro de série et leur ID de fournisseur.

Message de requête de défaut de communication "modification du MAC ID":

Utilisé lorsque le client a détecté une station de défaut de communication spécifique et souhaite modifier son MAC ID, et a tenté de se connecter à un MAC ID spécifié.

Un appareil qui met en œuvre les protocoles de *messages de défaut de communication* doit être capable de gérer les quatre protocoles de messages de défaut de communication ci-dessus.

5.2.4.5 Messages de défaut de communication "identification"

5.2.4.5.1 Généralités

Le client peut solliciter un *message de réponse de défaut de communication "identification"* à partir des éléments suivants:

- TOUTES les stations de défaut de communication; ou
- les stations de défaut de communication à un ou plusieurs MAC ID spécifiques; ou
- une station de défaut de communication avec un ID de fournisseur et un numéro de série spécifiques.

5.2.4.5.2 Message de requête "identification" – Protocole multidestinataire

Le *message de requête de défaut de communication "identification"* – protocole multidestinataire, doit être produit à l'ID de message 0x2D du groupe 4, voir Figure 27. Le client produit ce message lorsqu'il tente d'identifier l'existence de toute station dans l'état de défaut de communication.

Si la longueur du *message de requête de défaut de communication "identification"* est de deux (2) octets, toutes les stations de défaut de communication traitant ce message doivent répondre mais ne pas faire clignoter leurs DEL.

Byte offset	Contents							
	7	6	5	4	3	2	1	0
0	Reserved [0]	Match	Value					
1	R/R [0]	Identify [0x4C]						

IEC

Légende

Anglais	Français
Byte offset	Rang de l'octet
Contents	Contenu
Reserved	Réservé
Match	Correspondance
Value	Valeur
Identify [0x4C]	Identification [0x4C]

Figure 27 – Message de requête de défaut de communication "identification" – Protocole multidestinataire

5.2.4.5.3 Message de réponse "identification" – Protocole multidestinataire

Le format du *message de réponse de défaut de communication "identification"* est le même que le *message de requête de défaut de communication "identification"* reçu (avec les informations de correspondance et de valeur identiques aux informations dans la requête reçue), à l'exception qu'il est produit à l'ID de message 0x2C du groupe 4, voir Figure 28, et le bit R/R est initialisé (1). Ce message de réponse doit être initié en l'espace de 100 ms, à la réception du message de requête.

Byte offset	Contents							
	7	6	5	4	3	2	1	0
0	Reserved [0]	Match	Value					
1	R/R [1]	Identify [0x4C]						

IEC

Légende

Anglais	Français
Byte offset	Rang de l'octet
Contents	Contenu
Reserved	Réservé
Match	Correspondance
Value	Valeur
Identify [0x4C]	Identification [0x4C]

Figure 28 – Message de réponse de défaut de communication "identification"

Si plus d'une station répond à la requête, des collisions ne se produiront pas sur le sous-réseau. Ceci est le résultat de toutes les stations de défaut de communication produisant le même message (ID de CAN et champ de données). Néanmoins, lorsque plus d'une station répond, des messages multiples peuvent être reçus par le client.

5.2.4.5.4 Message de requête "identification" – Protocole point à point

Le *message de requête de défaut de communication "identification"* – protocole point à point, doit être produit à l'ID de message 0x2D du groupe 4, voir Figure 29. Si un (outil) client détecte la (les) station(s) de défaut de communication sur un sous-réseau, et si l'utilisateur souhaite identifier visuellement une station de défaut de communication à un MAC ID spécifique, alors l'(outil) client doit produire le *message de requête de défaut de communication "identification"* – protocole point à point.

Byte offset	Contents							
	7	6	5	4	3	2	1	0
0	Reserved [0]	Match [0]	Value [0x3F]					
1	R/R [0]	Identify [0x4C]						
2	Vendor ID							Low byte
3								High byte
4	Serial number							Low byte
5								
6								
7								High byte

IEC

Légende

Anglais	Français
Byte offset	Rang de l'octet
Contents	Contenu
Reserved	Réservé
Match	Correspondance
Value [0x3F]	Valeur [0x3F]
Identify [0x4C]	Identification [0x4C]
Vendor ID	ID du fournisseur
Low byte	Octet de poids faible
High byte	Octet de poids fort
Serial number	Numéro de série

Figure 29 – Message de requête de défaut de communication "identification" – Protocole point à point

Un client doit toujours fixer la "correspondance" sur 0 et la "valeur" sur 0x3F. Un serveur peut ignorer le contenu de l'octet 0 (la vérification de l'ID du fournisseur et du numéro de série est suffisante pour l'unicité).

Si une station accepte un *message de requête de défaut de communication "identification"* – protocole point à point, ET se trouve dans l'état de défaut de communication, elle doit répondre comme suit:

- Faire clignoter sa DEL bicolore d'état du réseau ou sa DEL d'état combiné du module/réseau. La vitesse de clignotement doit faire alterner le fonctionnement de la DEL rouge pendant 250 ms, puis de la DEL verte pendant 250 ms. Si l'appareil n'a pas de DEL bicolore d'état du réseau ou de DEL d'état combiné du module/réseau, une indication appropriée visible depuis l'extérieur spécifique à l'appareil doit être prévue

(par exemple, par l'intermédiaire d'un écran d'affichage), afin de montrer que le *message "identification"* a été accepté;

- Produire un *message de réponse de défaut de communication "identification"*, en l'espace de 100 ms.

Si une station a accepté un *message de requête de défaut de communication "identification"* – protocole point à point, ET fait clignoter sa DEL, elle doit arrêter de faire clignoter sa DEL dans les conditions suivantes. De même, si l'appareil n'a pas de DEL bicolore d'état du réseau ou de DEL d'état combiné du module/réseau, et si l'indication spécifique à l'appareil utilisée pour indiquer qu'un *message "identification"* avait été accepté est ALLUMÉE, elle doit être ARRETEE dans ces conditions:

- Un autre *message de requête de défaut de communication "identification"* – protocole point à point, n'est pas accepté en l'espace de 500 ms;
- Un autre *message de requête de défaut de communication "identification"* – protocole point à point, est reçu et n'est PAS accepté (et un autre numéro de série ou ID de fournisseur).

Si une station utilise une DEL d'état du réseau ou une DEL d'état combiné du module/réseau pour l'indication du *message "identification"*, alors, après avoir arrêté de faire clignoter la DEL, la station doit arrêter la DEL pendant une durée comprise entre 500 ms et 1 s, avant de revenir en fonctionnement normal.

5.2.4.5.5 Message de réponse "identification" – Protocole point à point

Le *message de réponse de défaut de communication "identification"* – protocole point à point, est le même que le *message de réponse de défaut de communication "identification"* pour le protocole multidestinataire.

5.2.4.5.6 Message de requête de défaut de communication «qui»

Le *message de requête de défaut de communication "qui"* doit être produit à l'ID de message 0x2D du groupe 4, voir Figure 30. Le *message de requête de défaut de communication "qui"* est utilisé pour obtenir le numéro de série et l'ID du fournisseur d'une station de défaut de communication.

		Contents						
Byte offset	7	6	5	4	3	2	1	0
0	Reserved [0]	Match	Value					
1	R/R [0]	Identify who [0x4B]						
2	Time delay byte offset [0-6]							

IEC

Légende

Anglais	Français
Byte offset	Rang de l'octet
Contents	Contenu
Reserved	Réservé
Match	Correspondance
Value	Valeur
Identify who [0x4B]	Identifieur qui [0x4B]
Time delay byte offset [0 – 6]	Rang de l'octet de temporisation [0 – 6]

Figure 30 – Message de requête de défaut de communication "qui"

SI la station qui consomme:

- est dans l'état de défaut de communication; ET
- détecte le *message de requête de défaut de communication "qui"*; ET
- satisfait aux critères de correspondance et de valeur.

ALORS la station de défaut de communication:

- doit attendre pendant une durée déduite du rang de l'octet de temporisation; ET
- produire un *message de réponse "qui"* unique à l'ID de message 0x2C du groupe 4; ET
- rester dans l'état de défaut de communication.

Le client doit consommer le message de requête de duplication d'adresse MAC ID produit lors de l'ID 0x2C du *message de réponse de défaut de communication*, obtenant ainsi le numéro de série et l'ID du fournisseur de la station de défaut de communication.

La précision de la période d'attente doit dépendre de la résolution du temporisateur interne dans la station de défaut de communication.

L'intervalle pendant lequel la station défectueuse attend avant de répondre est déterminé de la façon suivante:

- obtenir la valeur au rang de l'octet de temporisation dans le *message de requête de défaut de communication "qui"* (les valeurs comprises entre 0 et 6 sont valables);
- utiliser cette valeur comme un rang dans le *message de réponse "qui"* présenté sur la Figure 31; puis
- multiplier la valeur de l'octet à ce rang fois 50 ms; et
- retarder la réponse pendant cette période de temps.

5.2.4.5.7 Réponse de communication "qui"

Le *message de réponse "qui"*, produit suivant le délai spécifié, suit le même protocole qu'un message de requête de duplication d'adresse MAC ID, à l'exception qu'il est produit à l'ID de message 0x2C du groupe 4, voir Figure 31.

Byte offset	Contents							
	7	6	5	4	3	2	1	0
0	R/R [0]	Physical port number						
1	Vendor ID							Low byte
2								High byte
3	Serial number							Low byte
4								
5								
6								High byte

IEC

Légende

Anglais	Français
Byte offset	Rang de l'octet
Contents	Contenu
Physical port number	Numéro de l'accès physique
Vendor ID	ID du fournisseur

Anglais	Français
Low byte	Octet de poids faible
High byte	Octet de poids fort
Serial number	Numéro de série

Figure 31 – Message de réponse "qui"

5.2.4.5.8 Message de requête de défaut de communication "modification du MAC ID"

L'objet de ce message est de modifier le MAC ID d'une station dans l'état de défaut de communication. Ce message doit être produit à l'aide de l'ID de message 0x2D du groupe 4, voir Figure 32.

Byte offset	Contents							
	7	6	5	4	3	2	1	0
0	Reserved [0]		Value [New MAC ID]					
1	R/R [0]	Change MAC ID [0x4D]						
2	Vendor ID							Low byte
3								High byte
4	Serial number							Low byte
5								
6								
7								High byte

IEC

Légende

Anglais	Français
Byte offset	Rang de l'octet
Contents	Contenu
Reserved	Réservé
Value [New MAC ID]	Valeur [Nouveau MAC ID]
Change MAC ID [0x4D]	Modification du MAC ID [0x4D]
Vendor ID	ID du fournisseur
Low byte	Octet de poids faible
High byte	Octet de poids fort
Serial number	Numéro de série

Figure 32 – Message de requête de défaut de communication "modification du MAC ID"

Les stations qui acceptent la caractéristique du *message de requête de défaut de communication "modification du MAC ID"* ne doivent pas produire de *message de réponse de défaut de communication*.

Si la station qui consomme:

- est dans l'état de défaut de communication; ET
- détecte le *message de requête de défaut de communication "modification du MAC ID"*; ET
- vérifie que le numéro de série et l'ID du fournisseur correspondent à sa propre valeur,

ALORS la station défectueuse:

- modifie son MAC ID en nouveau MAC ID; ET
- passe à l'état envoi d'une duplication d'adresse MAC ID (voir 5.4.3).

Si, au nouveau MAC ID, la station échoue lors de la vérification de duplication d'adresse MAC ID, la station doit conserver le nouveau MAC ID et repasser à l'état défectueux.

Le contenu de l'octet zéro (0), bits 6 et 7, est ignoré par la station défectueuse. La production d'une valeur différente de zéro à cet emplacement est une erreur du client, indiquant une non-conformité.

5.2.5 Cadenceur d'appareil

5.2.5.1 Généralités

Le présent paragraphe définit le protocole associé au *message de cadenceur d'appareil* facultatif. Le *message de cadenceur d'appareil* est déclenché par l'objet identité, qui est défini en 5.3.2.

5.2.5.2 Message de cadenceur d'appareil

Ce message diffuse l'état actuel de l'appareil. Ce message est transmis par un appareil avec capacité UCMM comme un message de réponse non connecté (message du groupe 3, ID de message 5) et par un serveur du groupe 2 seulement comme un message de réponse non connecté (message du groupe 2, ID de message 3). Le format du *message de cadenceur d'appareil* est présenté sur la Figure 33.

Byte offset		Contents							
		7	6	5	4	3	2	1	0
0		Frag [0]	XID	MAC ID					
1		R/R [1]	Service code [0x4D]						
2		Identity object instance ID						Low byte	
3								High byte	
4		Device state							
5		Reserved			EV	SF	UF	DF	
6		Configuration consistency value						Low byte	
7								High Byte	

Message header

Message body

Message header

Message body

IEC

Légende

Anglais	Français
Byte offset	Rang de l'octet
Contents	Contenu
Service code [0x4D]	Code de service [0x4D]
Message header	En-tête du message
Low byte	Octet de poids faible
High byte	Octet de poids fort
Identity object instance ID	ID d'instance de l'objet identité

Anglais	Français
Device state	État de l'appareil
Message body	Corps du message
Reserved	Réservé
Configuration consistency value	Valeur de cohérence de configuration

Figure 33 – Message de cadenceur d'appareil

5.2.5.3 Contenu de la trame de données, message de cadenceur d'appareil

Frag (0)/ID de transaction/MAC ID – Le message de cadenceur est un message de diffusion non sollicité; il n'y a pas de MAC ID de destination. Le MAC ID source est spécifié dans l'en-tête du message. Le fait que l'en-tête du message ne doit pas contenir le même MAC ID que le champ identificateur de CAN constitue une exception à la règle générale.

Bit R/R (1) – Indique qu'il s'agit d'un message de réponse.

Code de service (0x4D) – Indique qu'il s'agit d'un message de cadenceur d'appareil.

ID d'instance de l'objet identité – L'ID d'instance de l'objet identité produisant le message de cadenceur d'appareil. Ce champ doit avoir une longueur de 2 octets.

État de l'appareil – Attribut 8 de l'instance de l'objet identité associée. Si l'attribut 8 n'est pas accepté, l'état de l'appareil doit être 3 (appareil opérationnel).

EV – Drapeau d'événement (Event Flag), son utilisation est à développer. Ce bit n'est pas pris en compte par le récepteur et doit être mis à zéro par le transmetteur.

SF – Défaut du système (System Fault) – défaut dans le système causé par interaction de bus (par exemple expiration du délai de connexion). Ce drapeau est initialisé lorsqu'un défaut est présent dans le système.

UF – Défaut utilisateur (User Fault) – défaut dans l'appareil causé par interaction avec l'utilisateur. Ce drapeau est initialisé lorsqu'un défaut utilisateur est présent. Les conditions dans lesquelles il est initialisé sont spécifiques au fournisseur.

DF – Défaut de l'appareil (Device Fault) – défaut interne dans l'appareil non causé par interaction avec l'utilisateur ou de bus (par exemple défaut matériel). Ce drapeau est initialisé lorsqu'un défaut est présent dans l'appareil.

Bits réservés – Utilisation à développer. Ces bits ne sont actuellement pas pris en compte par le récepteur et doivent être mis à zéro par le transmetteur.

Valeur de cohérence de configuration – Attribut 9 de l'instance de l'objet identité associée. Si l'attribut 9 n'est pas accepté, la valeur de cohérence de configuration doit être zéro.

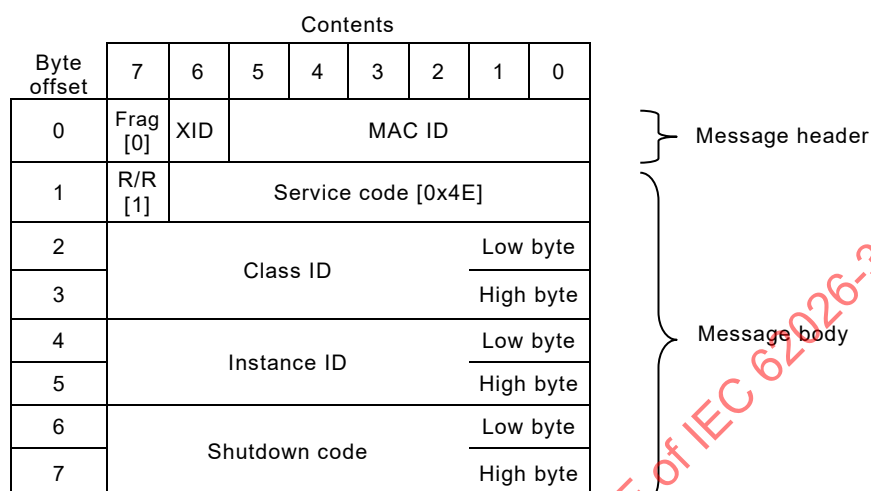
5.2.6 Message de déconnexion d'appareil

5.2.6.1 Généralités

Le présent paragraphe définit le protocole associé au *message de déconnexion d'appareil* facultatif. Le *message de déconnexion d'appareil* est produit par un appareil lorsqu'il passe à l'état hors ligne.

5.2.6.2 Message de déconnexion d'appareil

Ce message diffuse la transition d'un appareil vers l'état hors ligne ou inexistant. Ce message est transmis par un appareil avec capacité UCMM comme un message de réponse non connecté (message du groupe 3, ID de message 5) et par un serveur du groupe 2 seulement comme un message de réponse non connecté (message du groupe 2, ID de message 3). Le format du *message de déconnexion d'appareil* est présenté sur la Figure 34.



IEC

Légende

Anglais	Français
Byte offset	Rang de l'octet
Contents	Contenu
Message header	En-tête du message
Service code [0x4E]	Code de service [0x4E]
Class ID	ID de classe
Low byte	Octet de poids faible
High byte	Octet de poids fort
Message body	Corps du message
Instance ID	ID d'instance
Shutdown code	Code de déconnexion

Figure 34 – Message de déconnexion d'appareil

5.2.6.3 Contenu de la trame de données, message de déconnexion d'appareil

Frag (0)/ID de transaction/MAC ID – Le message de déconnexion est un message de diffusion non sollicité, il n'y a pas de MAC ID de destination. Le MAC ID source est spécifié dans l'en-tête du message. Le fait que l'en-tête du message ne doit pas contenir le même MAC ID que le champ identificateur de CAN constitue une exception à la règle générale.

Bit R/R (1) – Indique qu'il s'agit d'un message de réponse.

Code de service (0x4E) – Indique qu'il s'agit d'un message de déconnexion d'appareil.

ID de classe / ID d'instance – Ces deux valeurs identifient la classe / l'instance d'objet responsable de la transition de l'appareil vers l'état hors ligne. Si la déconnexion n'est pas

causée par une classe spécifique, l'ID de classe doit être 0. Ces champs doivent avoir chacun une longueur de 2 octets.

Code de déconnexion – Cette valeur qui indique la raison de la transition de l'appareil vers l'état hors ligne est présentée dans le Tableau 10 et le Tableau 11.

Tableau 10 – Plages des codes de déconnexion du message de déconnexion d'appareil

Valeur	Signification
0x0000 – 0x01FF	Ouvert
0x0200 – 0x02FF	Spécifique au fournisseur
0x0300 – 0x04FF	Spécifique à la classe de l'objet
0x0500 – 0xFFFF	Réservé par DeviceNet à une utilisation ultérieure

Tableau 11 – Codes de déconnexion "ouverts" du message de déconnexion d'appareil

Valeur	Signification
0	Réservé
1	Déconnexion par l'opérateur
2	Réinitialisation par l'opérateur
3	Déconnexion à distance
4	Réinitialisation à distance
5	Défaut de diagnostic interne
6	Défaut d'attribution des ressources
0x0007 – 0x01FF	Réserve

5.2.7 Protocole de détection de duplication d'adresse MAC ID

Chaque station DeviceNet doit recevoir un MAC ID et est tenue de participer à l'algorithme de détection de duplication d'adresse MAC ID, comme indiqué en 5.4.

Une valeur spéciale d'ID de message est définie dans le groupe 2 pour identifier le *message de vérification de duplication d'adresse MAC ID* (voir Figure 35).

IDENTIFIER BITS										MESSAGE ID MEANING
10	9	8	7	6	5	4	3	2	1	
1	0	MAC ID					Group 2 message ID			Group 2 messages
1	0	Destination MAC ID					1	1	1	Duplicate MAC ID check message

IEC

Légende

Anglais	Français
IDENTIFIER BITS	BITS DE L'IDENTIFICATEUR
MESSAGE ID MEANING	SIGNIFICATION DE L'ID DE MESSAGE
Destination MAC ID	MAC ID de destination
Group 2 message ID	ID message groupe 2
Group 2 messages	Messages du groupe 2
Duplicate MAC ID check message	Message de vérification de duplication de MAC ID

Figure 35 – Champ identificateur de CAN pour la vérification de duplication d'adresse MAC ID

Le champ de données associé au *message de vérification de duplication d'adresse MAC ID* est au format indiqué à la Figure 36:

Byte offset	Contents							
	7	6	5	4	3	2	1	0
0	R/R	Physical port number						
1	Vendor ID							Low byte
2								High byte
3	Serial number							Low byte
4								
5								
6								High byte

IEC

Légende

Anglais	Français
Byte offset	Rang de l'octet
Contents	Contenu
Physical port number	Numéro d'accès physique
Vendor ID	ID du fournisseur
Low byte	Octet de poids faible
High byte	Octet de poids fort
Serial number	Numéro de série

Figure 36 – Format du champ de données d'un message de vérification de duplication d'adresse MAC ID

Contenu du champ de données d'un message de vérification de duplication d'adresse MAC ID:

- **bit R/R:** bit de requête/réponse. La valeur de ce champ indique s'il s'agit d'un message de requête de vérification de duplication d'adresse MAC ID ou d'un message de réponse.
0 = Requête
1 = Réponse
- **numéro de l'accès physique:** Valeur d'identification attribuée en interne à chaque accès physique DeviceNet d'un appareil. Les appareils munis d'un seul accès doivent placer la valeur zéro (0) dans ce champ;
- **ID du fournisseur:** Nombre entier de 16 bits (UINT) contenant le code d'identification attribué au fabricant de l'appareil qui transmet le message;
- **numéro de série:** Voir 3.1.43.

5.2.8 Connexion rapide

La caractéristique de connexion rapide est une option activée sur une base station par station. Lorsqu'il est activé, un appareil passe à l'état en ligne tout en envoyant le premier message de requête de duplication d'adresse MAC ID. Il est toujours exigé que l'appareil

exécute la norme réseau, y compris le fait de se mettre hors ligne à chaque fois qu'un message de réponse de duplication d'adresse MAC ID est reçu.

Important: Bien que cette caractéristique permette à un appareil de commencer à participer plus rapidement à l'activité du réseau, ceci est au détriment d'un délai dans l'algorithme de détection de duplication de stations. Il appartient à l'utilisateur de garantir qu'aucune station n'existe avec le même MAC ID et que pas plus d'un appareil client est configuré pour accéder au même appareil en utilisant le jeu de connexions maître/esclave prédéfini. Cependant, des erreurs peuvent se produire si l'une ou l'autre de ces conditions existe.

Cette caractéristique est activée dans un appareil par l'intermédiaire d'un attribut permanent dans l'objet DeviceNet. Cette caractéristique doit être désactivée dans un appareil (attribut mis à "0") comme réglage usine par défaut.

5.3 Classes d'objets de communication DeviceNet

5.3.1 Généralités

La communication DeviceNet est modélisée dans une station par un ensemble d'objets: les objets de communication DeviceNet assurent la gestion et la production des messages échangés.

Un objet donne une représentation abstraite d'une structure de données dans une station. Une classe d'objets est un ensemble d'objets qui sont tous d'un même type. Une instance d'objet est l'occurrence réelle d'un objet particulier dans la classe. Chaque instance d'une classe possède le même jeu d'attributs, mais a des valeurs d'attributs qui lui sont propres.

Une instance d'objet et/ou une classe d'objets possèdent des attributs, offrent des services et réagissent suivant un comportement.

Les attributs sont les caractéristiques d'un objet et/ou d'une classe d'objets. Ils donnent des informations d'état ou dirigent le fonctionnement d'un objet. Les services sont invoqués pour déclencher l'exécution d'une tâche par la classe ou l'instance d'objets. Le comportement d'un objet indique la façon dont il répond à des événements particuliers.

Les principaux objets de communication utilisés par DeviceNet sont énumérés dans le présent paragraphe (voir 5.3.2 à 5.3.6).

Les codes des classes d'objets et leurs noms sont spécifiés dans l'IEC 61158-6-2:2014, 4.1.10.2.1. La spécification et le codage des types de données sont spécifiés dans l'IEC 61158-5-2:2014, Article 5, et l'IEC 61158-6-2:2014, 4.2 et Article 5.

5.3.2 Définition de la classe d'objet identité (code d'ID de classe: 0x01)

L'objet identité identifie et fournit des informations générales sur la station.

L'objet identité est entièrement spécifié dans l'IEC 61158-5-2:2014, 6.2.1.2.2 et l'IEC 61158-6-2:2014, 4.1.8.2.

5.3.3 Définition de la classe d'objet routeur de messages (code d'ID de classe: 0x02)

L'objet routeur de messages fournit un point de connexion de messagerie par lequel un client peut adresser un service à n'importe quelle classe ou instance d'objet de la station.

L'objet routeur de messages est entièrement spécifié dans l'IEC 61158-5-2:2014, 6.2.1.2.4 et l'IEC 61158-6-2:2014, 4.1.8.3.

5.3.4 Définition de la classe d'objet DeviceNet (code d'ID de classe: 0x03)

L'objet DeviceNet fournit la configuration et l'état d'une liaison physique à DeviceNet (accès DeviceNet). Un appareil doit mettre en œuvre un objet DeviceNet par liaison physique.

L'objet DeviceNet est spécifié dans l'IEC 61158-4-2:2014, 7.7. Ceci inclut des attributs et des services communs pour la classe et les instances d'objets DeviceNet.

Les services spécifiques à la classe d'objet DeviceNet utilisés pour accorder et retirer l'attribution du jeu de connexions maître/esclave prédéfini sont spécifiés en 5.5.3.

5.3.5 Définition de la classe d'objet connexion (code d'ID de classe: 0x05)

5.3.5.1 Généralités

La classe d'objet connexion attribue et gère les ressources internes associées aux connexions d'E/S et de messagerie explicite. L'instance spécifique générée par la classe d'objet connexion est appelée instance de connexion ou objet connexion.

La classe d'objet connexion est entièrement spécifiée dans l'IEC 61158-5-2, l'IEC 61158-6-2 et l'IEC 61784-3-2. Cela comprend:

- des attributs et des services pour la classe et les instances d'objets connexion (voir l'IEC 61158-5-2:2014, 6.2.3 et l'IEC 61158-6-2:2014, 4.1.8.8);
- des services spécifiques pour la sécurité fonctionnelle (voir l'IEC 61784-3-2);
- la temporisation de connexion (voir l'IEC 61158-5-2:2014, 6.2.3);
- le comportement d'une instance de connexion (voir l'IEC 61158-6-2:2014, 7.2).

5.3.5.2 Gestion dynamique des ID de messages

L'établissement dynamique des connexions d'E/S et de messagerie explicite nécessite que toutes les extrémités mettent en œuvre des procédures internes d'attribution d'ID de message. Ces procédures doivent aussi noter qu'un ID de message précédemment alloué est de nouveau disponible lorsque cet ID de message n'est plus utilisé.

Pour réduire la possibilité d'erreurs de CID, les règles qui suivent doivent être observées:

- a) la procédure d'attribution d'un CID doit s'assurer qu'il n'y a pas deux stations DeviceNet avec des configurations conduisant à la transmission de séquences binaires identiques dans le champ identificateur de connexion;
- b) la procédure de libération d'un CID doit s'assurer que toutes les extrémités d'une connexion sont à l'état "délai expiré" avant de réutiliser un ID de message:
 - si l'ID de message est associé à une connexion qui active un temporisateur d'inactivité/surveillance, un nouveau temporisateur d'inactivité/surveillance est alors activé. À l'expiration du temporisateur, l'ID de message est marqué comme étant disponible;
 - si l'ID de message est associé à une connexion qui n'active pas un temporisateur d'inactivité/surveillance, il peut être immédiatement noté comme disponible (on suppose que ceci a été pris en compte lorsque la connexion a été établie).

Important: Pour une connexion utilisant la classe de transport 0 ou une connexion dont l'attribut `expected_packet_rate` a été mis à zéro (0), la règle b) ne peut pas être obtenue d'après cet objet connexion seul. Pour ces types de connexions, il convient de réaliser avec précaution les tâches qui entraîneront la libération et la réutilisation possible de l' (des) ID(s) de message(s) associé(s) (c'est-à-dire en configurant l'attribut `watchdog_timeout_action` en suppression automatique («Auto Delete»), en supprimant manuellement un objet connexion via la transmission du service suppression («delete»)).

EXEMPLE 1

Si l'extrémité serveur d'une connexion de classe de transport 0 subit une expiration du délai d'inactivité/de surveillance, elle ne peut pas connaître l'état du client en se basant uniquement sur cette connexion. Le client pourrait avoir simplement oublié de transmettre le message ponctuellement et pourrait encore penser que la connexion fonctionne normalement.

EXEMPLE 2

Si l'extrémité client d'une connexion dont l'attribut `expected_packet_rate` a été mis à zéro (0) est supprimée pour une raison quelconque, l'(les) extrémité(s) serveur pourrai(en)t toujours être active(s).

5.3.6 Définition de la classe d'objet gestionnaire d'acquittements (code d'ID de classe: 0x2B)

L'objet gestionnaire d'acquittements gère la réception des acquittements de messages.

L'objet gestionnaire d'acquittements est entièrement spécifié dans l'IEC 61158-5-2:2014, 6.2.1.2.5 et l'IEC 61158-6-2:2014, 4.1.8.5.

5.4 Diagramme d'états d'accès à la liaison

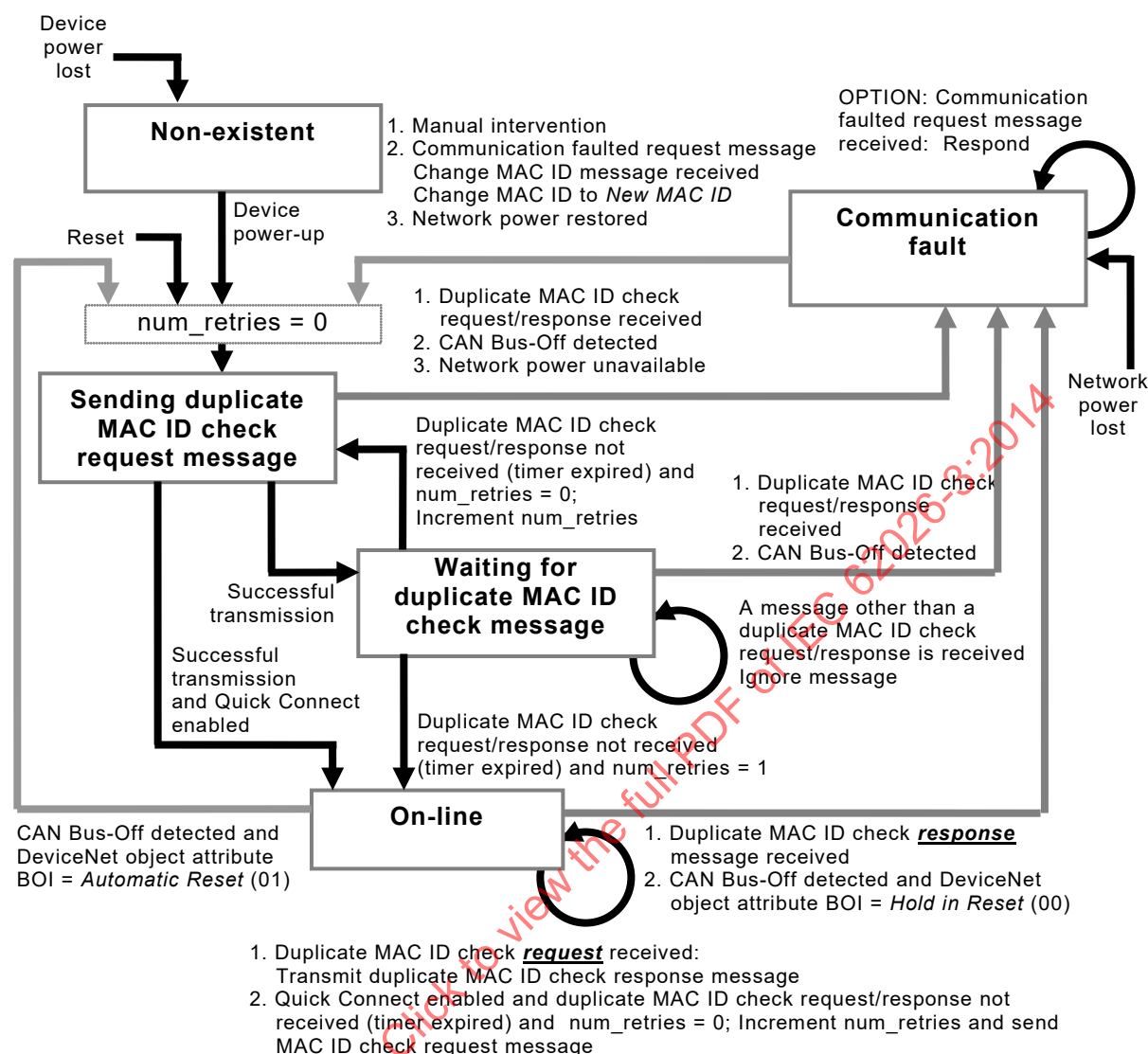
5.4.1 Généralités

Le présent paragraphe définit le diagramme d'états d'accès à la liaison que chaque station DeviceNet doit mettre en œuvre. Le diagramme d'états d'accès à la liaison est décrit par les critères suivants:

- tâches qui doivent être effectuées avant de communiquer à travers le CDI;
- événements de la liaison affectant les possibilités de communication d'une station à travers le CDI.

5.4.2 Diagramme de transition d'états et table états-événements

La Figure 37 fournit un aperçu général du diagramme d'états d'accès à la liaison.



IEG

Légende

Anglais	Français
Device power lost	Perte d'alimentation de l'appareil
Non-existent	Non-existant
Device power-up	Mise sous tension de l'appareil
Reset	Réinitialisation
Sending duplicate MAC ID check request message	Envoi d'un message de requête de vérification de duplication d'adresse MAC ID
Successful transmission	Transmission réussie
Successful transmission and Quick Connect enabled	Transmission réussie et connexion rapide activée
CAN Bus-Off detected and DeviceNet object attribute BOI = <i>Automatic Reset</i> (01)	Détection d'un bus-off CAN et l'attribut BOI de l'objet DeviceNet = Remise à zéro (RAZ) automatique (01)
1. Manual intervention 2. Communication faulted request message Change MAC ID message received Change MAC ID to New MAC ID 3. Network power restored	1. Intervention manuelle 2. Message de requête de défaut de communication Message "modification du MAC ID" reçu Modifier le MAC ID en Nouveau MAC ID 3. Rétablissement de l'alimentation réseau
1. Duplicate MAC ID check	1. Requête/réponse de vérification de

Anglais	Français
request/response received 2. CAN Bus-Off detected 3. Network power unavailable	duplication d'adresse MAC ID reçues 2. Détection d'un bus-off CAN 3. Alimentation réseau indisponible
Duplicate MAC ID check request/response not received (timer expired) and num_retries = 0; Increment num_retries	Requête/réponse de vérification de duplication d'adresse MAC ID pas reçues (temporisateur expiré) et num_retries = 0; Incrémenter num_retries
Waiting for duplicate MAC ID check message	Attente du message de vérification de duplication d'adresse MAC ID
Duplicate MAC ID check request/response not received (timer expired) and num_retries =	Requête/réponse de vérification de duplication d'adresse MAC ID pas reçues (temporisateur expiré) et num_retries = 1
On-line	En ligne
1. Duplicate MAC ID check request received: Transmit duplicate MAC ID check response message 2. Quick Connect enabled and duplicate MAC ID check request/response not received (timer expired) and num_retries = 0; Increment num_retries and send MAC ID check request message	1. Message de requête de vérification de duplication d'adresse MAC ID reçu: Transmettre un message de réponse de vérification de duplication d'adresse MAC ID 2. Connexion rapide activée et requête/réponse de vérification de duplication d'adresse MAC ID pas reçues (temporisateur expiré) et num_retries = 0; Incrémenter num_retries et envoyer le message de requête de vérification de duplication d'adresse MAC ID
OPTION: Communication faulted request message received: Respond	OPTION: Message de requête de défaut de communication reçu: Répondre
Communication fault	Défaut de communication
Network power lost	Perte d'alimentation réseau
1. Duplicate MAC ID check request/response received 2. CAN Bus-Off detected	1. Requête/réponse de vérification de duplication d'adresse MAC ID reçues 2. Détection d'un bus-off CAN
A message other than a duplicate MAC ID check request/response is received Ignore message	Un message autre qu'une requête/réponse de vérification de duplication d'adresse MAC ID est reçu Ignorer le message
1. Duplicate MAC ID check response message received 2. CAN Bus-Off detected and DeviceNet object attribute BOI = Hold in Reset (00)	1. Message de réponse de vérification de duplication d'adresse MAC ID reçu 2. Détection d'un bus-off CAN et l'attribut BOI de l'objet DeviceNet = Maintien en RAZ (00)

Figure 37 – Diagramme de transition d'états d'accès à la liaison

Le Tableau 12 détaille la table états-événements pour le diagramme d'états d'accès à la liaison.

Tableau 12 – Tableau états-événements d'accès à la liaison (1 de 3)

Événement	Etat			
	Envoi d'une requête de vérification de duplication d'adresse MAC ID	Attente du message de vérification de duplication d'adresse MAC ID	En ligne	Défaut de communication
Transmission réussie du message de requête de vérification de duplication d'adresse MAC ID	Activer le temporisateur de 1 s ^a . Si la connexion rapide est activée, passer à l'état en ligne, sinon passer à l'état attente du message de vérification de duplication d'adresse MAC ID	Non applicable	Non applicable	Non applicable

Tableau 12 (2 de 3)

Événement	Etat			
	Envoi d'une requête de vérification de duplication d'adresse MAC ID	Attente du message de vérification de duplication d'adresse MAC ID	En ligne	Défaut de communication
Détection bus-off	Transcepteur maintenu en remise à zéro (RAZ). Transition vers l'état défaut de communication	Transcepteur maintenu en remise à zéro. Transition vers l'état défaut de communication	Accéder à l'attribut BOI de l'objet DeviceNet. Si l'attribut BOI indique que le transcepteur doit être maintenu en RAZ, passer en défaut de communication. Si l'attribut BOI indique que le MAC et le transcepteur doivent être automatiquement remis à zéro, procéder comme suit: 1. remettre à zéro, 2. demander la transmission du message de requête de vérification de duplication d'adresse MAC ID, et 3. passer à l'état envoi d'une requête de vérification de duplication d'adresse MAC ID	Non applicable
Message de requête de vérification de duplication d'adresse MAC ID reçu	Duplication d'adresse MAC ID détectée. Transition vers l'état défaut de communication	Duplication d'adresse MAC ID détectée. Transition vers l'état défaut de communication	Transmettre le message de réponse de vérification de duplication d'adresse MAC ID	Ne pas tenir compte du message
Message de réponse de vérification de duplication d'adresse MAC ID reçu	Duplication d'adresse MAC ID détectée. Transition vers l'état défaut de communication	Duplication d'adresse MAC ID détectée. Transition vers l'état défaut de communication	Duplication d'adresse MAC ID détectée. Transition vers l'état défaut de communication	Ne pas tenir compte du message
Le temporisateur de 1 s du message de vérification de duplication d'adresse MAC ID expire ^a	Non applicable	S'il s'agit du premier délai expiré, demander de nouveau la transmission d'un message de requête de vérification de duplication d'adresse MAC ID et passer à l'envoi de la requête de vérification de duplication d'adresse MAC ID. S'il s'agit du deuxième délai expiré consécutif, passer en ligne	Si la connexion rapide est activée, demander de nouveau la transmission du message de requête de vérification de duplication d'adresse MAC ID	Non applicable

Tableau 12 (3 de 3)

Événement	Etat			
	Envoi d'une requête de vérification de duplication d'adresse MAC ID	Attente du message de vérification de duplication d'adresse MAC ID	En ligne	Défaut de communication
Requête de transmission de message interne	Renvoyer une erreur interne	Renvoyer une erreur interne	Transmettre le message	Renvoyer une erreur interne
Un message autre qu'une requête/ réponse de vérification de duplication d'adresse MAC ID est reçu	Ne pas tenir compte du message	Ne pas tenir compte du message	Traiter le message reçu de la façon appropriée	Ne pas tenir compte du message
Un message de requête de défaut de communication est reçu	Ne pas tenir compte du message	Ne pas tenir compte du message	Ne pas tenir compte du message	Traiter le message reçu de la façon appropriée
^a Plage de validité pour le temporisateur: de 0,9 s à 1,5 s.				

Le diagramme d'états de gestion des erreurs du protocole CAN prévoit que, pendant la mise en route/remise en route du système, une seule station peut être présente sur la liaison. Si cette station transmet un message, elle va subir une erreur d'acquiescement et répéter automatiquement le message. Dans cette situation, la station va passer dans l'état "error passive" mais non "bus-off". Pour cette raison, le seul événement qui signifie l'échec de transmission d'un message de vérification de duplication d'adresse MAC ID (requête ou réponse) est l'événement bus-off. Si une indication "error passive" ou "error warning" est reçue pendant la transmission d'un message de vérification de duplication d'adresse MAC ID, elle doit être ignorée, car elle est sans effet sur le diagramme d'états de détection de duplication d'adresse MAC ID.

5.4.3 Détection de duplication d'adresse MAC ID

La principale étape du diagramme d'états d'accès à la liaison est l'exécution de l'algorithme de détection de duplication d'adresse MAC ID. Chaque station DeviceNet reçoit un MAC ID unique; par mesure de protection contre les erreurs, toutes les stations DeviceNet doivent participer à l'algorithme de détection de duplication d'adresse MAC ID.

NOTE Le protocole associé à l'algorithme de détection de duplication d'adresse MAC ID est décrit en 5.2.7.

Comme indiqué en 5.2.7, un message spécial du groupe 2 est réservé pour l'exécution de la détection de duplication d'adresse MAC ID.

Une station DeviceNet doit recevoir et traiter tout message de vérification de duplication d'adresse MAC ID qui contient son propre MAC ID dans le champ identificateur du groupe de messages 2.

Après la transmission d'un message de requête de vérification de duplication d'adresse MAC ID, un appareil doit attendre 1 s avant de décréter l'expiration du délai et d'entreprendre l'action appropriée définie par le diagramme d'états d'accès à la liaison.

Le message de requête de vérification de duplication d'adresse MAC ID doit être transmis deux fois sans recevoir en retour aucun message de requête ou réponse de vérification de duplication d'adresse MAC ID avant de pouvoir passer en ligne.

Voir 9.2.3 et 9.3.2 pour les spécifications d'essais concernant le comportement à la mise en route et la gestion des mécanismes liés à la duplication d'adresse MAC ID.

5.5 Jeu de connexions maître/esclave prédéfini

5.5.1 Généralités

Les paragraphes précédents décrivent les règles générales d'établissement de connexions entre appareils. Les règles générales prescrivent l'utilisation d'une connexion de messagerie explicite pour créer et configurer les objets connexion à chaque extrémité de la connexion. Le présent paragraphe utilise les règles générales comme base de la définition d'un jeu de connexions qui facilite les communications typiques d'une relation maître/esclave. Ces connexions sont collectivement appelées jeu de connexions maître/esclave prédéfini:

- **serveur du groupe 2:** Appareil ayant la capacité de gestionnaire de messages non connecté (UCMM), configuré pour agir comme serveur des connexions d'identificateur maître/esclave prédéfinies;
- **client du groupe 2:** Appareil ayant la capacité UCMM et ayant obtenu la propriété du jeu de connexions maître/esclave prédéfini dans un serveur de sorte qu'il peut agir en tant que client sur ces connexions;
- **appareil à capacité UCMM:** Appareil qui dispose de la fonctionnalité UCMM;
- **appareil sans capacité UCMM:** Appareil qui ne dispose pas de la fonctionnalité UCMM;
- **serveur du groupe 2 seulement:** Appareil esclave sans capacité UCMM et qui utilise le jeu de connexions maître/esclave prédéfini pour établir des communications. Un appareil du groupe 2 seulement ne peut transmettre et recevoir que les identificateurs spécifiés par le jeu de connexions maître/esclave prédéfini;
- **client du groupe 2 seulement:** Appareil agissant comme client du groupe 2 envers un serveur du groupe 2 seulement. Le client du groupe 2 seulement fournit la fonctionnalité UCMM aux serveurs du groupe 2 seulement qui lui sont alloués;
- **maître DeviceNet:** Fait référence à un type d'application appelé maître/esclave. Le maître DeviceNet est l'appareil qui rassemble et distribue les données d'E/S pour le régulateur de processus. Un maître contrôle ses appareils esclaves d'après une liste de contrôle qu'il contient. Concernant le réseau, le maître est un client du groupe 2 ou un client du groupe 2 seulement;
- **esclave DeviceNet:** Fait référence à un type d'application appelé maître/esclave. Un esclave renvoie les données d'E/S lorsqu'il est contrôlé. Concernant le réseau, l'esclave est un serveur du groupe 2 ou un serveur du groupe 2 seulement;
- **jeu de connexions maître/esclave prédéfini:** Jeu de connexions qui facilite les communications typiques d'une relation maître/esclave. De nombreuses étapes impliquées dans la création et la configuration d'une connexion entre applications ont été retirées dans la définition du jeu de connexions maître/esclave prédéfini. Ceci présente successivement les moyens par lesquels un environnement de communication peut être établi en utilisant moins de ressources réseau et appareil.

5.5.2 Messages du jeu de connexions maître/esclave prédéfini

Les champs identificateurs de CAN associés au jeu de connexions maître/esclave prédéfini sont indiqués dans le Tableau 13, en même temps que les identificateurs qui doivent être utilisés pour toute messagerie de connexion utilisée dans le cadre du jeu de connexions maître/esclave prédéfini.

Tableau 13 – Champs identificateurs pour le jeu de connexions maître/esclave prédéfini

Bits d'identificateur											Utilisation	Plage
10	9	8	7	6	5	4	3	2	1	0		
0	ID des messages du groupe 1				MAC ID source						Messages du groupe 1	0x000-0x3FF
0	1	1	0	0	MAC ID source						Message d'E/S réponse interrogation multidestinataire de l'esclave	
0	1	1	0	1	MAC ID source						Message d'E/S changement d'état ou cyclique de l'esclave	
0	1	1	1	0	MAC ID source						Message d'E/S réponse requête multiple de l'esclave	
0	1	1	1	1	MAC ID source						Message d'E/S réponse interrogation ou acquittement ou changement d'état/cyclique de l'esclave	
1	0	MAC ID						ID des messages du groupe 2			Messages du groupe 2	0x400-0x5FF
1	0	MAC ID source						0	0	0	Message d'E/S de commande requête multiple du maître	
1	0	MAC ID multidestinataire						0	0	1	Message d'E/S de commande interrogation multidestinataire du maître	
1	0	MAC ID destination						0	1	0	Message d'acquiescement ou changement d'état/cyclique du maître	
1	0	MAC ID source						0	1	1	Messages de réponse explicite de l'esclave	
1	0	MAC ID destination						1	0	0	Messages de requête explicite du maître	
1	0	MAC ID destination						1	0	1	Message d'E/S de commande interrogation/changement d'état/cyclique du maître	
1	0	MAC ID destination						1	1	0	Messages de requête explicite non connectés du groupe 2 seulement	
1	0	MAC ID destination						1	1	1	Messages de vérification de duplication d'adresse MAC ID	

Les types de messages suivants sont inclus dans le Tableau 13:

- **messages d'E/S de commande/réponse requête multiple:** La commande requête multiple est un message d'E/S transmis par le maître. De nombreux esclaves peuvent recevoir et réagir à la même commande requête multiple. La réponse requête multiple est un message d'E/S renvoyé par un esclave au maître à la réception d'une commande de requête multiple;
- **messages d'E/S de commande/réponse interrogation:** La commande interrogation est un message d'E/S transmis par le maître. Une commande interrogation est dirigée vers un esclave spécifique. La réponse interrogation est un message d'E/S renvoyé par un esclave au maître à la réception d'une commande interrogation;
- **messages d'E/S changement d'état/cyclique:** Il s'agit de messages changement d'état/cyclique transmis par le maître ou l'esclave. Un message changement d'état/cyclique est dirigé vers une station spécifique. Un message d'acquiescement doit être renvoyé en réponse à ce message sauf si la configuration prévoit la suppression des messages d'acquiescement;
- **messages d'E/S interrogation multidestinataire** La commande interrogation multidestinataire est un message d'E/S transmis par le maître. Une interrogation multidestinataire est dirigée vers un ou plusieurs esclaves. La réponse interrogation multidestinataire est un message d'E/S renvoyé par un esclave au maître à la réception d'une commande interrogation multidestinataire;

- **messages de requête/réponse explicite:** Voir 5.2.1.6;
- **messages de requête explicite non connectés du groupe 2 seulement:** L'accès de requête explicite non connectée du groupe 2 seulement est utilisé pour attribuer/libérer le jeu de connexions maître/esclave prédéfini;
- **messages de réponse explicite non connectés du groupe 2 seulement:** L'accès de réponse explicite non connectée du groupe 2 seulement est utilisé pour répondre aux messages de requête explicite non connectés du groupe 2 seulement et pour envoyer des messages de cadenceur / déconnexion d'appareil. Ces messages sont transmis en utilisant le même identificateur (groupe 2, ID de message = 3) que les messages de réponse explicite;
- **message de vérification de duplication d'adresse MAC ID:** Voir 5.2.7.

5.5.3 Services spécifiques à la classe d'objet DeviceNet pour le jeu de connexions maître/esclave

5.5.3.1 Allocate_master/slave_connection_set (code de service: 0x4B)

5.5.3.1.1 Généralités

Ce service attribue le jeu de connexions maître/esclave prédéfini. Les codes d'erreur généraux sont définis dans l'IEC 61158-5-2:2014, 6.2.1.3.3 et l'IEC 61158-6-2:2014, 4.1.11. Les valeurs des codes d'erreur supplémentaires spécifiques à l'objet DeviceNet sont définies en 5.5.3.5. Ce service doit être transmis au moyen du message de requête explicite non connecté du jeu de connexions maître/esclave (voir 5.5.2) ou bien via une connexion de messagerie explicite.

Le service allocate_master/slave_connection_set effectue les tâches suivantes:

- création d'un objet connexion;
- configuration d'un objet connexion.

Voir 9.3.4 et 9.3.5 pour les spécifications d'essais concernant l'attribution du jeu de connexions maître/esclave pour les connexions d'E/S et de messagerie explicite.

5.5.3.1.2 Paramètres du champ des données de service d'une requête

Les informations du Tableau 14 sont placées dans le champ des données de service d'une requête allocate_master/slave_connection_set.

Tableau 14 – Paramètres du champ des données de service d'une requête allocate_master/slave_connection_set

Nom	Type de données	Description du paramètre
Choix d'attribution	BYTE	Indique quelles connexions du jeu de connexions maître/esclave prédéfini sont à attribuer/configurer pour une utilisation par le maître
MAC ID de l'appareil demandeur	USINT	Contient le MAC ID associé à la station demandant l'attribution

Le paramètre choix d'attribution est spécifié dans un seul octet (voir Figure 38). Chaque bit correspond à une ou à des connexions de messagerie explicite et/ou d'E/S du jeu de connexions maître/esclave prédéfini qui sont à attribuer, ou bien, dans le cas de la suppression d'acquittement, à une commande. Si le bit est mis à 1, une requête d'attribution est faite pour la connexion correspondante. Si le bit est mis à 0, le demandeur ne veut pas attribuer la connexion correspondante.

7	6	5	4	3	2	1	0
Reserved	Acknowledge suppression	Cyclic	Change of state	Multicast Polled	Bit strobed	Polled	Explicit message

IEC

Légende

Anglais	Français
Reserved	Réservé
Acknowledge suppression	Suppression d'acquittement
Cyclic	Cyclique
Change of state	Changement d'état
Multicast polled	Réservé
Bit strobed	Requête multiple
Polled	Interrogation
Explicit message	Message explicite

Figure 38 – Contenu de l'octet de choix d'attribution

Les bits 3 et 7 doivent être mis à 0 et l'esclave doit vérifier cette exigence lorsqu'il reçoit la requête allocate_master/slave_connection_set.

La Figure 39 indique le format de ce message explicite.

		Contents							
Byte offset		7	6	5	4	3	2	1	0
0	Frag (0)	XID MAC ID							
1	R/R (0)	Service (0x4B)							
		Class ID (3)							
		Instance ID (1)							
		Allocation choice							
		0	0	Allocator's MAC ID					

IEC

Légende

Anglais	Français
Byte offset	Rang de l'octet
Contents	Contenu
Class ID (3)	ID de classe (3)
Instance ID (1)	ID d'instance (1)
Allocation choice	Choix d'attribution
Allocator's MAC ID	MAC ID de l'appareil demandeur

Figure 39 – Message de requête allocate_master/slave_connection_set

Contenu du message de requête allocate_master/slave_connection_set:

- **frag (0)/ID de transaction/MAC ID:** Comme indiqué en 5.2.1.2;
- **bit R/R (0):** Indique qu'il s'agit d'un message de requête;
- **code de service (0x4B):** Identifie ce message comme étant le service allocate_master/slave_connection_set;
- **ID de classe:** Définit la classe d'objet vers laquelle cette requête est dirigée. Lorsque ce message est transmis via l'accès du message de requête explicite non connecté du jeu de connexions maître/esclave (voir 5.5.2), la valeur de l'ID de classe est spécifiée dans un nombre entier de 8 bits. L'ID de classe doit être sur 3;
- **ID d'instance:** Définit l'instance particulière dans la classe d'objet vers laquelle cette requête est dirigée. Lorsque ce message est transmis via l'accès du message de requête explicite non connecté du jeu de connexions maître/esclave (voir 5.5.2), la valeur de l'ID d'instance est spécifiée dans un nombre entier de 8 bits. L'ID d'instance doit être mis à 1;
- **choix d'attribution:** Spécifié dans l'octet qui suit le champ de l'ID d'instance;
- **MAC ID de l'appareil demandeur:** Spécifié dans l'octet qui suit le champ choix d'attribution.

5.5.3.1.3 Paramètres du champ des données de service d'une réponse positive

Les informations contenues dans le champ des données de service d'une réponse positive allocate_master/slave_connection_set sont décrites dans le Tableau 15.

Tableau 15 – Paramètres de la réponse allocate_master/slave_connection_set

Nom	Type de données	Description du paramètre
Format du corps du message	Défini ci-dessous	Ceci équivaut sémantiquement au paramètre de format réel de corps du message renvoyé avec une réponse d'ouverture de connexion de messagerie explicite (comme décrit en 5.2.1.5). Cet argument est significatif lorsque la requête allocate_master/slave_connection_set a été reçue par l'accès du message de requête explicite non connecté du jeu de connexions maître/esclave (voir 5.5.2). Il indique le format du corps du message associé aux messages ultérieurs transmis via la connexion de messagerie explicite du jeu de connexions maître/esclave prédéfini. Si la requête allocate_master/slave_connection_set a été reçue via une connexion de messagerie explicite d'un appareil ayant la capacité UCMM, ce paramètre est fixé au format réel de corps du message associé à cette connexion de messagerie explicite

La Figure 40 indique le format d'une réponse positive à la requête allocate_master/slave_connection_set.

		Contents						
Byte offset	7	6	5	4	3	2	1	0
0	Frag (0)	XID	MAC ID					
1	R/R (1)	Service code (0x4B)						
2	Reserved (all bits = 0)				Message body format			

IEC

Légende

Anglais	Français
Byte offset	Rang de l'octet
Contents	Contenu
Reserved (all bits = 0)	Réservé (tous les bits = 0)

Anglais	Français
Message body format	Format du corps du message

Figure 40 – Réponse positive à une requête `allocate_master/slave_connection_set`

Contenu de la réponse positive à la requête `allocate_master/slave_connection_set`:

- **frag (0)/ID de transaction/MAC ID**: Comme indiqué en 5.2.1.2;
- **bit R/R (1)**: Indique qu'il s'agit d'un message de réponse;
- **code de service (0x4B)**: Identifie ce message comme étant le service `allocate_master/slave_connection_set`;
- **bits réservés**: Ces bits ne sont pas utilisés par le récepteur de la réponse et doivent être mis à 0 par le transmetteur du message de réponse;
- **format du corps du message**: Comme décrit dans le Tableau 15.

5.5.3.2 Comportement du serveur `allocate_master/slave_connection_set`

- a) Si une erreur se produit, aucune des connexions demandées ne doit être attribuée. Si cette requête ne peut être entièrement satisfaite, aucune des attributions demandées ne doit être effectuée.
- b) Si l'appareil de réception ne dispose pas d'un jeu de connexions maître/esclave prédéfini, une réponse d'erreur est renvoyée. Le code d'erreur général de la réponse d'erreur doit être mis à 0x08 pour indiquer un service non accepté.
- c) L'appareil de réception (esclave) valide le paramètre MAC ID de l'appareil demandeur dans la requête, de la façon suivante:
 - si le jeu de connexions maître/esclave prédéfini est déjà attribué et si cette requête ne provient pas du maître en cours, l'esclave retourne une erreur. Le code d'erreur général de la réponse d'erreur doit être mis à 0x0C, et le code d'erreur supplémentaire à la valeur 01, spécifique à l'objet;
 - si le jeu de connexions maître/esclave prédéfini n'est pas attribué, l'esclave valide le paramètre choix d'attribution, comme indiqué en d);
 - si le jeu de connexions maître/esclave prédéfini est déjà attribué et si cette requête provient du maître en cours, l'esclave doit valider le paramètre choix d'attribution, comme indiqué en d).
- d) L'esclave doit valider le paramètre choix d'attribution contenu dans la requête. Si l'esclave ne dispose pas de l'une des connexions spécifiées dans l'argument choix d'attribution, une réponse d'erreur doit être renvoyée. Le code d'erreur général de la réponse d'erreur doit être mis à 02, et le code d'erreur supplémentaire à la valeur 0x02, spécifique à l'objet.

Si une ou plusieurs des connexions demandées sont disponibles dans cet esclave mais ont déjà été attribuées au maître indiqué par l'argument MAC ID de l'appareil demandeur, l'esclave doit renvoyer une réponse d'erreur avec le code d'erreur général mis à 0x0B, et le code d'erreur supplémentaire à la valeur 0x02, spécifique à l'objet. Si la connexion d'E/S demandée est dans l'état délai expiré, l'esclave doit réattribuer la connexion d'E/S, en la mettant dans l'état configuration.

Si l'octet choix d'attribution n'a aucun bit mis à 1, l'esclave doit renvoyer une réponse d'erreur avec le code d'erreur général mis à 0x09, et le code d'erreur supplémentaire à la valeur 0x02, spécifique à l'objet.

Si une ressource dont l'utilisation est requise pour les connexions demandées n'est pas disponible, une réponse d'erreur doit être renvoyée avec le code d'erreur général mis à 0x02, et le code d'erreur supplémentaire à la valeur 0x04, spécifique à l'objet.

Les choix d'attribution cyclique et de changement d'état s'excluent l'un l'autre. Si une requête d'attribution implique la mise à 1 simultanée des bits d'affectation cyclique et de changement d'état, une réponse d'erreur doit être renvoyée. Le code d'erreur général de la réponse d'erreur doit être mis à 0x09 (valeur d'attribut incorrecte) et le code d'erreur supplémentaire à 0x02.

Si un maître a attribué le jeu de connexions cyclique/changement d'état et si une requête d'attribution ultérieure est reçue avec le bit d'attribution interrogation mis à 1, une erreur doit être renvoyée. Le code d'erreur général de la réponse d'erreur doit être mis à 0x02 pour indiquer que la ressource n'est pas disponible. Si le bit de suppression d'acquittement de l'octet choix d'attribution est mis à 1 mais pas le bit changement d'état ni le bit cyclique, une réponse d'erreur doit être renvoyée. Le code d'erreur général de la réponse d'erreur doit être mis à 0x09 et le code d'erreur supplémentaire à 0x02.

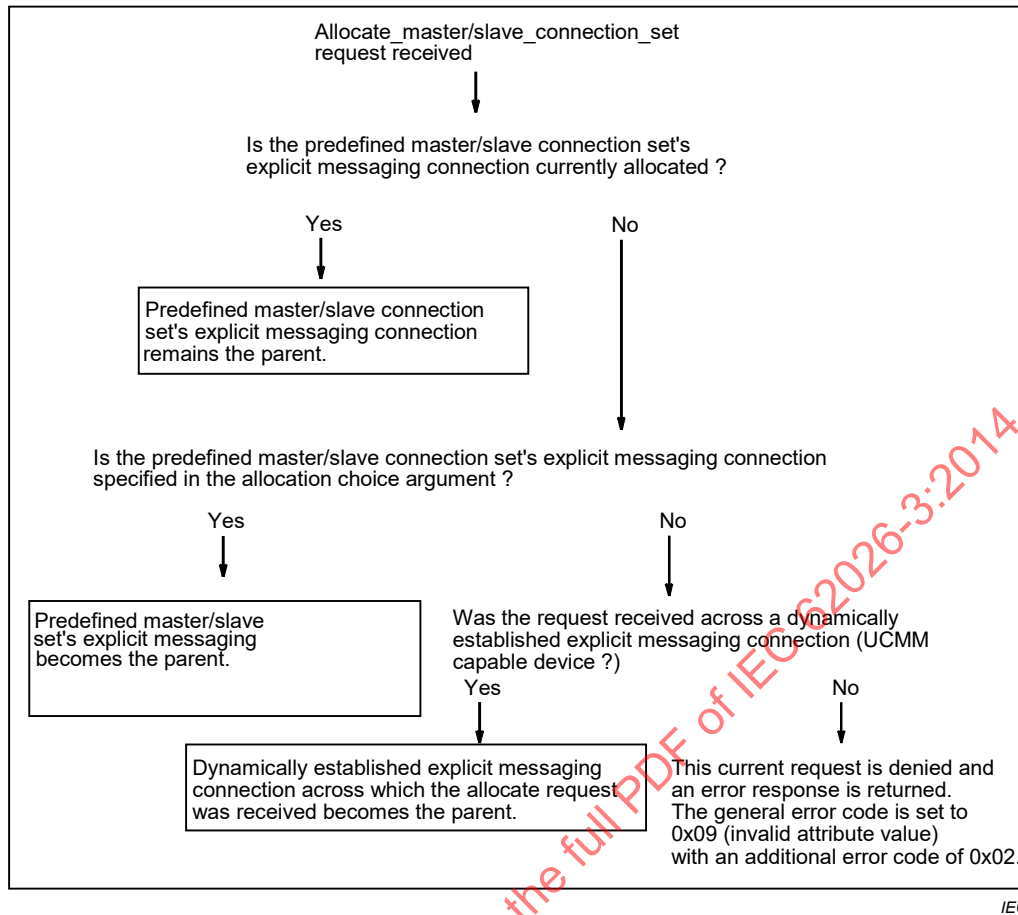
- e) L'esclave prend note du fait que le jeu de connexions maître/esclave prédéfini a été attribué au MAC ID indiqué dans le champ MAC ID de l'appareil demandeur, en mettant à jour l'attribut informations d'affectation de l'objet DeviceNet. Si nécessaire, les attributs `produced_connection_id` et/ou `consumed_connection_id` de l'objet ou des objets connexion peuvent maintenant être initialisés.

L'octet choix d'attribution de l'attribut informations d'attribution indique quels objets connexion du jeu de connexions maître/esclave prédéfini sont actifs (dans l'état configuration ou établi). Cet octet est mis à jour chaque fois qu'un objet connexion maître/esclave change d'état.

- f) Les connexions d'E/S attribuées passent à l'état configuration. Pour ce qui est des objets connexion maître/esclave prédéfinis, un service `apply_attributes` implicite accompagne un `set_attribute_single` pour l'attribut `expected_packet_rate` effectué pendant que la connexion est à l'état configuration. Un `set_attribute_single` pour l'attribut `expected_packet_rate` déclenche l'exécution des étapes prévues pour un service `apply_attributes` et entraîne la transition de l'objet connexion d'E/S maître/esclave prédéfini vers l'état établi. Si une erreur se produit lors de l'exécution de `apply_attributes`, l'attribut `expected_packet_rate` doit être remis à sa valeur précédente et une erreur doit être renvoyée, avec le code d'erreur général mis à 0x09 et le code d'erreur supplémentaire chargé avec l'ID de l'attribut erroné de l'objet connexion.
- g) Si elle est attribuée, la connexion de messagerie explicite passe à l'état établi. Le temporisateur d'inactivité/surveillance est lancé à l'aide de la valeur initiale spécifiée dans l'IEC 61158-5-2:2014, 6.2.3.2.1.7.

Les appareils à capacité UCMM qui disposent du jeu de connexions maître/esclave prédéfini doivent accepter l'attribution et l'utilisation de la connexion de messagerie explicite du jeu de connexions maître/esclave prédéfini.

- h) La connexion de messagerie explicite du jeu de connexions maître/esclave, ou bien une connexion de messagerie explicite dynamiquement établie, agit comme parent de la ou des connexions d'E/S attribuées jusqu'à ce que ces connexions d'E/S passent à l'état établi. Le terme parent est utilisé pour indiquer qu'en cas de suppression de la connexion de messagerie explicite alors qu'aucune des connexions d'E/S attribuées n'est à l'état établi, le jeu de connexions maître/esclave prédéfini est automatiquement libéré par l'esclave. La logique qui définit quelle connexion de messagerie explicite est tenue d'agir comme parent est définie à la Figure 41.



Légende

Anglais	Français
Allocate_master/slave_connection_set request received	Requête Allocate_master/slave_connection_set reçue
Is the predefined master/slave connection set's explicit messaging connection currently allocated?	Est-ce que la connexion de messagerie explicite du jeu de connexions maître/esclave prédéfini est déjà attribuée?
Yes	Oui
No	Non
Predefined master/slave connection set's explicit messaging connection remains the parent.	La connexion de messagerie explicite du jeu de connexions maître/esclave prédéfini reste le parent.
Is the predefined master/slave connection set's explicit messaging connection specified in the allocation choice argument?	Est-ce que la connexion de messagerie explicite du jeu de connexions maître/esclave prédéfini est spécifiée dans l'argument choix d'attribution?
Predefined master/slave set's explicit messaging becomes the parent.	La connexion de messagerie explicite du jeu de connexions maître/esclave prédéfini devient le parent.
Was the request received across a dynamically established explicit messaging connection (UCMM capable device?)	Est-ce que la requête a été reçue via une connexion de messagerie explicite établie dynamiquement (appareil à capacité UCMM)?
Dynamically established explicit messaging connection across which the allocate request was received becomes the parent	La connexion de messagerie explicite établie dynamiquement via laquelle la demande d'attribution a été reçue devient le parent.
This current request is denied and an error response is returned. The general error code is set to 0x09 (invalid attribute value) with an additional error code of 0x02.	Cette requête est rejetée et une réponse d'erreur est renvoyée. Le code d'erreur général est mis à 0x09 (valeur d'attribut incorrecte) avec un code d'erreur supplémentaire à 0x02.

Figure 41 – Choix d'une connexion de messagerie explicite parent

L'esclave doit libérer automatiquement le jeu de connexions maître/esclave prédéfini et toutes les connexions doivent repasser à l'état non existant si toutes les conditions suivantes sont vraies:

- aucun des objets connexion associés au jeu de connexions maître/esclave prédéfini n'existe à l'état établi;
- l'objet connexion de messagerie explicite parent n'est pas à l'état établi.

5.5.3.3 Release_master/slave_connection_set (code de service: 0x4C)

5.5.3.3.1 Généralités

Ce service est utilisé pour libérer le jeu de connexions maître/esclave prédéfini dans un esclave. Les codes d'erreur généraux sont définis dans l'IEC 61158-5-2:2014, 6.2.1.3.3 et l'IEC 61158-6-2:2014, 4.1.11. Les valeurs des codes d'erreur supplémentaires spécifiques à l'objet DeviceNet sont définies en 5.5.3.5. Ce service peut être transmis via l'accès du message de requête explicite non connecté du jeu de connexions maître/esclave (voir 5.5.2), de même que via une connexion de messagerie explicite.

5.5.3.3.2 Paramètres du champ des données de service d'une requête

Les informations du Tableau 16 sont spécifiées dans le champ des données de service d'une requête release_master/slave_connection_set.

Tableau 16 – Paramètres du champ des données de service d'une requête release_master/slave_connection_set

Nom	Type de données	Description du paramètre
Choix de libération	BYTE	Indique quelles connexions maître/esclave prédéfinies sont à libérer

Le paramètre choix de libération est spécifié dans un seul octet (voir Figure 42). Chaque bit indique une ou des connexions de messagerie explicite et/ou d'E/S à libérer. Si le bit est mis à 1, une requête de libération est faite pour la connexion correspondante. Si le bit est mis à 0, le demandeur ne veut pas libérer la connexion correspondante.

7	6	5	4	3	2	1	0
Reserved	Ignored	Cyclic	Change of state	Multicast polled	Bit strobed	Polled	Explicit message

IEC

Légende

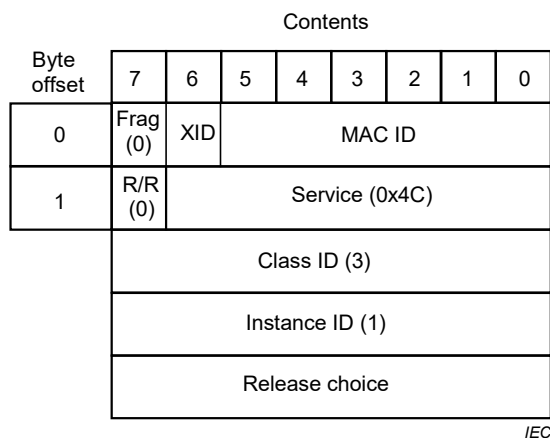
Anglais	Français
Reserved	Réservé
Ignored	Ignoré
Cyclic	Cyclique
Change of state	Changement d'état
Multicast polled	Interrogation multidestinataire
Bit strobed	Requête multiple
Polled	Interrogation
Explicit message	Message explicite

Figure 42 – Contenu de l'octet de choix de libération

Les bits 3 et 7 doivent être mis à 0 et l'esclave doit vérifier cette exigence lorsqu'il reçoit la requête release_master/slave_connection_set.

La valeur 00 n'est pas valide.

La Figure 43 indique le format de ce message explicite.



Légende

Anglais	Français
Byte offset	Rang de l'octet
Contents	Contenu
Class ID (3)	ID de classe (3)
Instance ID (1)	ID d'instance (1)
Release choice	Choix de libération

Figure 43 – Message de requête release_master/slave_connection_set

Contenu du message de requête release_master/slave_connection_set:

- **frag (0)/ID de transaction/MAC ID:** Voir 5.2.1.2;
- **bit R/R (0):** Indique qu'il s'agit d'un message de requête;
- **code de service (0x4C):** Identifie ce message comme étant le service release_master/slave_connection_set;
- **ID de classe:** Définit la classe de l'objet vers laquelle cette requête est dirigée. Lorsque ce message est transmis via l'accès du message de requête explicite non connecté du jeu de connexions maître/esclave (voir 5.5.2), la valeur de l'ID de classe est spécifiée dans un nombre entier de 8 bits. L'ID de classe doit être sur 3;
- **ID d'instance:** Définit l'instance particulière dans la classe d'objet vers laquelle cette requête est dirigée. Lorsque ce message est transmis via l'accès du message de requête explicite non connecté du jeu de connexions maître/esclave (voir 5.5.2), la valeur de l'ID d'instance est spécifiée dans un nombre entier de 8 bits. L'ID d'instance doit être mis à 1;
- **choix de libération:** Spécifié dans l'octet qui suit le champ de l'ID d'instance.

5.5.3.3.3 Message de réponse positive

La Figure 44 indique le format d'une réponse positive à la requête release_master/slave_connection_set.

Contenu								
Rang de l'octet	7	6	5	4	3	2	1	0
0	Frag (0)	XID	MAC ID					
1	R/R (1)	Service (0x4C)						

IEC

Légende

Anglais	Français
Byte offset	Rang de l'octet
Contents	Contenu

Figure 44 – Réponse positive à une requête `release_master/slave_connection_set`

Contenu de la réponse positive à la requête `release_master/slave_connection_set`:

- **frag (0)/ID de transaction/MAC ID**: Voir 5.2.1.2;
- **bit R/R (1)**: Indique qu'il s'agit d'un message de réponse;
- **code de service (0x4C)**: Identifie ce message comme étant le service `release_master/slave_connection_set`.

5.5.3.4 Comportement du serveur `release_master/slave_connection_set`

Si une erreur se produit, aucune des connexions spécifiées n'est libérée. Si cette requête ne peut être entièrement satisfaite, aucune des libérations demandées n'est effectuée.

Si l'appareil de réception ne dispose pas du jeu de connexions maître/esclave prédéfini, une réponse d'erreur doit être renvoyée. Le code d'erreur général de la réponse d'erreur doit être à 0x08.

Le dispositif de réception (esclave) doit valider le paramètre choix de libération dans la requête. Si l'esclave ne dispose pas de l'une des connexions spécifiées dans l'argument choix de libération, une réponse d'erreur doit être renvoyée. Le code d'erreur général de la réponse d'erreur doit être mis à 0x02, et le code d'erreur supplémentaire à la valeur 0x02, spécifique à l'objet.

Si l'octet choix de libération n'a aucun bit mis à 1 (tous les bits sont à 0), l'esclave doit renvoyer une réponse d'erreur avec le code d'erreur général mis à 0x09 (valeur d'attribut incorrecte), et le code d'erreur supplémentaire à la valeur 0x02, spécifique à l'objet.

Si l'une des connexions spécifiées est dans l'état non existant, une réponse d'erreur doit être renvoyée. Le code d'erreur général de la réponse d'erreur doit être à 0x0B.

L'esclave doit vérifier qu'il est dans un état qui lui permet de ne plus utiliser la ou les connexions spécifiées. Si ce n'est pas le cas, une réponse d'erreur doit être renvoyée. Le code d'erreur général de la réponse d'erreur doit être à 0x0C.

Si la requête est valide, l'esclave doit libérer toutes les ressources associées à la ou aux connexions spécifiées. Si cela a pour conséquence la libération de toutes les connexions maître/esclave prédéfinies, l'esclave doit prendre note du fait que le jeu de connexions maître/esclave prédéfini n'est plus attribué en mettant à jour l'attribut informations d'attribution.

L'esclave ne doit pas vérifier si la requête de libération vient de son maître.

Si, à la suite de cette requête, aucune des connexions maître/esclave prédéfinies n'existe à l'état établi, l'esclave doit libérer le jeu de connexions maître/esclave prédéfini et toutes les connexions doivent retourner à l'état non existant.

5.5.3.5 Codes d'erreur spécifiques à l'objet DeviceNet

Le Tableau 17 donne la liste des codes d'erreur supplémentaires spécifiques à l'objet DeviceNet.

Tableau 17 – Codes d'erreur supplémentaires spécifiques à l'objet DeviceNet

Valeur	Signification
01	Conflit d'attribution pour le jeu de connexions maître/esclave prédéfini. Renvoyé lorsqu'une requête <code>allocate_master/slave_connection_set</code> est reçue et que l'esclave a déjà attribué le jeu de connexions maître/esclave prédéfini à un autre maître
02	Paramètre choix d'attribution/libération incorrect. Renvoyé lorsqu'une requête <code>allocate/release_master/slave_connection_set</code> est reçue et que: 1) L'esclave n'accepte pas le choix spécifié dans le paramètre choix. 2) L'esclave a reçu la demande d'attribuer/de libérer une ou des connexions déjà attribuées/libérées. 3) L'octet choix d'attribution/libération ne contenait que des zéros, une combinaison de bits incorrecte, ou ne spécifiait pas d'attribution de message explicite lorsque c'était nécessaire
03	Un serveur qui n'a pas la capacité UCMM a reçu un message qui n'était pas un message d'attribution ou de libération sur l'accès du message de requête explicite non connecté du jeu de connexions maître/esclave (voir 5.5.2)
04	La ressource dont l'utilisation est requise pour le jeu de connexions maître/esclave prédéfini n'est pas disponible

5.5.4 Caractéristiques de l'objet connexion esclave

5.5.4.1 Généralités

Le présent paragraphe présente les caractéristiques, visibles en externe, des objets connexion associés au jeu de connexions maître/esclave prédéfini dans les appareils esclaves. Les objets des connexions maître/esclave prédéfinies décrits pour les appareils esclaves comprennent:

- **la connexion requête multiple:** Responsable de la réception de la commande requête multiple du maître et de l'envoi de la réponse requête multiple associée;
- **la connexion interrogation:** Responsable de la réception de la commande interrogation du maître et de l'envoi de la réponse interrogation associée;
- **la connexion de messagerie explicite:** Responsable de la réception des requêtes explicites et de l'envoi des réponses associées;
- **la connexion changement d'état/cyclique:** Responsable de l'envoi du message changement d'état/cyclique et de la réception de la réponse d'acquittement si cette fonction n'a pas été supprimée;
- **la connexion interrogation multidestinataire:** Responsable de la réception de la commande interrogation multidestinataire du maître et de l'envoi de la réponse interrogation multidestinataire associée.

Le présent paragraphe donne des informations supplémentaires sur les objets connexion au sein d'un jeu de connexions maître/esclave prédéfini. Sauf indication contraire, toutes les informations spécifiées en 5.3 s'appliquent aux objets connexion décrits dans le présent paragraphe.

5.5.4.2 ID d'instance de connexion

Chaque objet connexion existant possède un ID d'instance de connexion qui identifie l'objet connexion dans la classe connexion. Les ID d'instance de connexion qui doivent être utilisés

par un appareil esclave afin d'identifier les objets des connexions maître/esclave prédéfinies sont indiqués au Tableau 18.

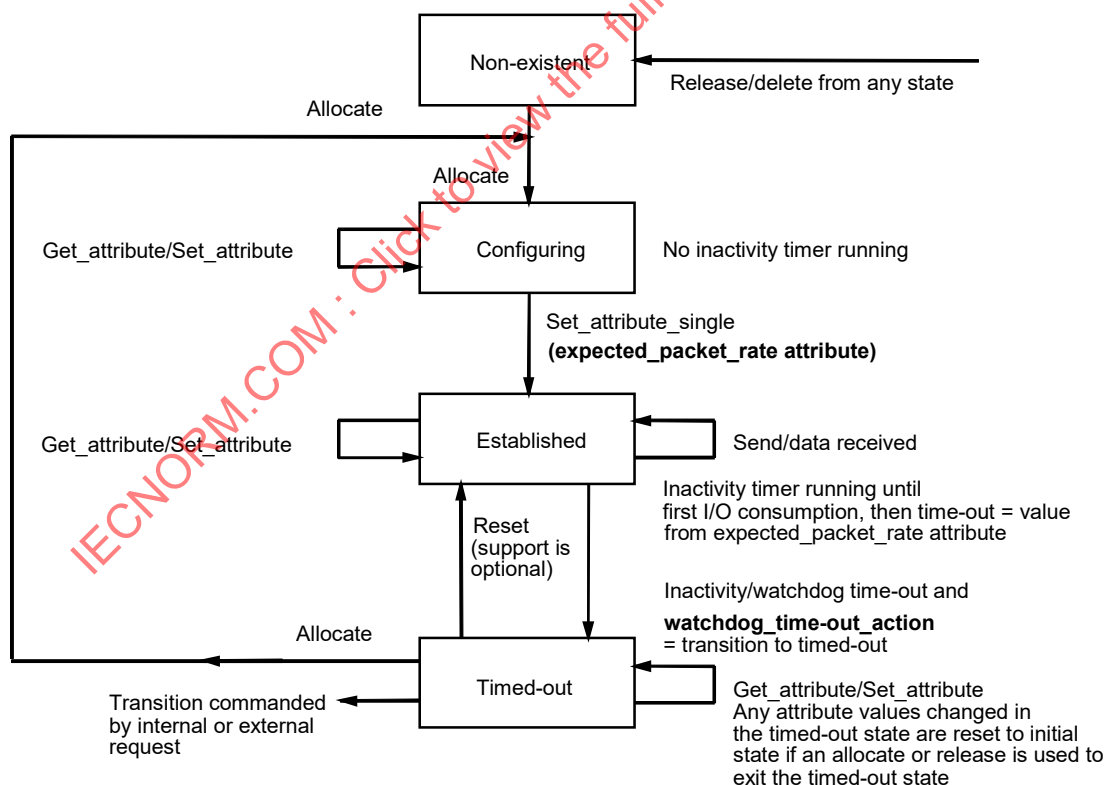
Tableau 18 – ID d'instance de connexion des connexions maître/esclave prédéfinies

ID d'instance de connexion	Description
1	Désigne la connexion de messagerie explicite dans le serveur
2	Désigne la connexion d'E/S sur interrogation
3	Désigne la connexion d'E/S sur requête multiple
4	Désigne la connexion d'E/S sur changement d'état ou cyclique de l'esclave
5	Désigne la connexion d'E/S sur interrogation multidestinataire

Un esclave doit réserver les ID d'instance du Tableau 18 pour les connexions maître/esclave prédéfinies qu'il gère. Par exemple, si un appareil gère la connexion d'E/S sur interrogation, il doit réserver/utiliser l'ID d'instance de connexion n° 2 pour identifier l'objet connexion d'E/S sur interrogation. Si un appareil ne gère pas la connexion sur interrogation, il est alors libre d'attribuer l'ID d'instance de connexion n° 2 pour identifier un autre objet connexion.

5.5.4.3 Comportement d'une instance de connexion maître/esclave prédéfinie

La Figure 45 représente le diagramme de transition d'états des objets des connexions d'E/S du jeu de connexions maître/esclave prédéfini.



The allocate and release services reset the connection instance. All connection object attributes are reset to their default values.

IEC

Légende

Anglais	Français
Non-existent	Non existant
Allocate	Attribution

Anglais	Français
Release/delete from any state	Libération/suppression depuis un état quelconque
Configuring	Configuration
No inactivity timer running	Temporisateur d'inactivité arrêté
Established	Établi
(expected_packet_rate attribute)	(attribut expected_packet-rate)
Send/data received	Send/data reçu
Inactivity timer running until first I/O consumption, then time-out = value from expected_packet_rate attribute	Temporisateur d'inactivité en route jusqu'à la première consommation d'E/S, ensuite délai expiré = valeur de l'attribut expected_packet_rate attribute
Reset (support is optional)	Réinitialisation (mise en œuvre en option)
Timed-out	Délai expiré
Transition commanded by internal or external request	Transition commandée par une requête en interne ou en externe
Any attribute values changed in the timed-out state are reset to initial state if an allocate or release is used to exit the timed-out state	Tous les attributs modifiés pendant l'état délai expiré sont remis à leur valeur initiale si on utilise un service d'attribution ou de suppression pour sortir de l'état délai expiré
The allocate and release services reset the connection instance. All connection object attributes are reset to their default values.	Les services d'attribution et de suppression réinitialisent l'instance de connexion. Tous les attributs de l'objet connexion sont remis à leurs valeurs par défaut.
Inactivity/watchdog time-out and watchdog_time-out_action = transition to timed-out	Expiration du temporisateur d'inactivité/surveillance et watchdog_time-out_action = transition vers délai expiré

Figure 45 – Diagramme de transition d'états des connexions d'E/S maître/esclave prédéfinies

Pour la modification des attributs, les connexions d'E/S maître/esclave prédéfinies doivent au moins accepter la modification de l'attribut expected_packet_rate.

La table états-événements présentée dans le Tableau 19 donne une définition formelle du comportement des connexions d'E/S dans un jeu de connexions maître/esclave prédéfini. Cette table états-événements conserve et/ou remplace les actions présentées dans la table états-événements de l'objet connexion d'E/S de 5.3.

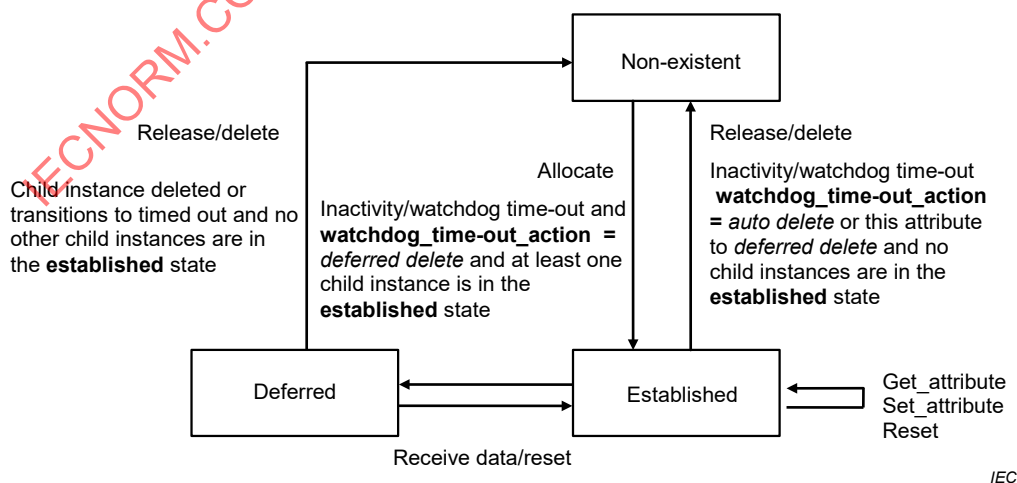
Tableau 19 – Table états-événements des connexions d'E/S maître/esclave prédéfinies
(1 de 2)

Événement	État de l'objet connexion d'E/S			
	Non existant	Configuration	Etabli	Délai expiré
L'objet DeviceNet reçoit une requête d'attribution du jeu de connexions maître/ esclave qui satisfait à toutes les vérifications d'erreur indiquées en 5.5.3.2. Cette requête spécifie l'une des connexions d'E/S maître/esclave prédéfinies	Créer un objet connexion pour chaque connexion d'E/S demandée et fixer les attributs aux valeurs par défaut spécifiées en 5.5.3.1. Transition vers configuration	Ceci est une erreur. Voir 5.5.3.2	Ceci est une erreur. Voir 5.5.3.2	Fixer les attributs aux valeurs par défaut spécifiées en 5.5.3.1. Transition vers configuration
La classe connexion reçoit une requête de suppression ou l'UCMM reçoit une requête de fermeture et la requête spécifie un objet connexion d'E/S maître/esclave prédéfinie	Comme décrit dans le Tableau 193 de l'IEC 61158-6-2:2014, 7.2.1	Libérer toutes les ressources associées. Transition vers non existant ^a	Libérer toutes les ressources associées. Transition vers non existant ^a	Libérer toutes les ressources associées. Transition vers non existant ^a
L'objet DeviceNet reçoit une requête de libération du jeu de connexions maître/ esclave qui satisfait à toutes les vérifications d'erreur indiquées en 5.5.3.2. Cette requête spécifie l'une des connexions d'E/S maître/esclave prédéfinies	Ceci est une erreur. Voir 5.5.3.2	Libérer toutes les ressources associées. Transition vers non existant ^a	Libérer toutes les ressources associées. Transition vers non existant ^a	Libérer toutes les ressources associées. Transition vers non existant ^a
Set_attribute_single	Comme décrit dans l'IEC 61158-6-2:2014, 7.2.1	Valider/exécuter la requête en fonction des règles d'accès présentées en 5.5.4.4. S'il s'agit d'une requête valide d'écriture de l'attribut expected_packet_rate, accomplir les étapes spécifiées dans l'IEC 61158-6-2:2014, 7.2.1 pour l'événement apply_attributes dans l'état configuration et passer à l'état établi. Renvoyer la réponse appropriée	Valider/exécuter la requête en fonction des règles d'accès présentées en 5.5.4.4. Renvoyer la réponse appropriée	Valider/exécuter la requête en fonction des règles d'accès présentées en 5.5.4.4. Renvoyer la réponse appropriée

Tableau 19 (2 de 2)

Événement	État de l'objet connexion d'E/S			
	Non existant	Configuration	Établi	Délai expiré
Get_attribute_single	Comme décrit dans l'IEC 61158-6-2:2014, 7.2.1	Valider/exécuter la requête en fonction des règles d'accès présentées en 5.5.4.4. Renvoyer la réponse appropriée	Valider/exécuter la requête en fonction des règles d'accès présentées en 5.5.4.4. Renvoyer la réponse appropriée	Valider/exécuter la requête en fonction des règles d'accès présentées en 5.5.4.4. Renvoyer la réponse appropriée
Remise à zéro	Comme décrit dans l'IEC 61158-6-2:2014, 7.2.1	Comme décrit dans l'IEC 61158-6-2:2014, 7.2.1	Comme décrit dans l'IEC 61158-6-2:2014, 7.2.1	Comme décrit dans l'IEC 61158-6-2:2014, 7.2.1
Apply_attributes		Comme décrit dans l'IEC 61158-6-2:2014, 7.2.1 ^b		
Receive_data				
Send_message				
Expiration du temporisateur d'inactivité/surveillance				
^a Chaque fois qu'un objet connexion d'un jeu de connexions maître/esclave prédéfini quitte l'état établi, il peut être nécessaire de libérer automatiquement le jeu complet de connexions maître/esclave prédéfini. Lorsque le jeu de connexions maître/esclave prédéfini est libéré, tous les objets connexion associés repassent à l'état non existant. Voir 5.5.3.3 pour plus de détails concernant la libération automatique du jeu de connexions maître/esclave prédéfini. ^b Du fait qu'un service apply_attributes implicite accompagne un set_attribute_single pour l'attribut expected_packet_rate d'un objet connexion d'E/S maître/esclave prédéfinie, dans l'état configuration, une requête de messagerie explicite apply_attributes réussira uniquement si expected_packet_rate attend encore d'être modifié avec succès via la requête set_attribute_single et, par conséquent, contient toujours la valeur par défaut 0.				

Le diagramme de transition d'états (voir Figure 46) montre le comportement d'une connexion de messagerie explicite maître/esclave prédéfinie.



Légende

Anglais	Français
Non-existent	Non-existant
Release/delete	Libération/suppression

Anglais	Français
Allocate	Attribution
Inactivity/Watchdog Timeout & watchdog_timeout_action = <i>Auto Delete</i> or this attribute is set to <i>Deferred Delete</i> and no child instances are in the Established state.	Expiration du délai d'inactivité/surveillance watchdog_time-out_action = suppression automatique ou cet attribut est défini sur suppression différée et aucune instance enfant n'est dans l'état établi
Reset	Remise à zéro
Established	Établi
Receive Data / Reset	Recevoir des données/remettre à zéro
Deferred	Différé
Inactivity/Watchdog Timeout & watchdog_timeout_action = <i>Deferred Delete</i> at least one child instance is in the Established state.	Expiration du délai d'inactivité/surveillance et watchdog_time-out_action = suppression différée et au moins une instance enfant est dans l'état établi
Child instance Deleted or Transitions to Timed Out and no other child instances are in the Established state	Instance enfant supprimée ou transitions vers l'état de délai expiré et aucune autre instance enfant n'est dans l'état établi

Figure 46 – Diagramme de transition d'états pour la connexion de messagerie explicite maître/esclave prédéfinie

Le Tableau 20 détaille la table états-événements pour l'objet connexion de messagerie explicite maître/esclave prédéfinie. Cette table états-événements conserve et/ou remplace les actions présentées dans la table états-événements de l'objet connexion d'E/S de 5.3.

Tableau 20 – Table états-événements des connexions de messagerie explicite maître/esclave prédéfinies

Événement	État d'un objet connexion de messagerie explicite		
	Non existant	Établi	Différé
L'objet DeviceNet reçoit une requête d'attribution du jeu de connexions maître/esclave qui satisfait à toutes les vérifications d'erreur indiquées en 5.5.3.2. Cette requête spécifie la connexion de messagerie explicite maître/esclave prédéfinie	Créer l'objet connexion de messagerie explicite maître/esclave prédéfinie pour chaque connexion d'E/S demandée et fixer les attributs aux valeurs par défaut (voir 5.5.3.1). Passer à établi	Ceci est une erreur. Voir 5.5.3.2	Ceci est une erreur. Voir 5.5.3.2
L'UCMM reçoit une requête de fermeture ou la classe connexion reçoit une requête de suppression, et la requête spécifie l'objet connexion de messagerie explicite maître/esclave prédéfinie	Comme décrit dans l'IEC 61158-6-2:2014, 7.2.3	Comme décrit dans l'IEC 61158-6-2:2014, 7.2.3	Comme décrit dans l'IEC 61158-6-2:2014, 7.2.3
L'objet DeviceNet reçoit une requête de libération du jeu de connexions maître/esclave qui satisfait à toutes les vérifications d'erreur indiquées en 5.5.3.2. Cette requête spécifie la connexion de messagerie explicite maître/esclave prédéfinie	Ceci est une erreur. Voir 5.5.3.2	Libérer toutes les ressources associées. Transition vers non existant ^a	Libérer toutes les ressources associées. Transition vers non existant ^a
Set_attribute_single	Comme décrit dans l'IEC 61158-6-2:2014, 7.2.3	Comme décrit dans l'IEC 61158-6-2:2014, 7.2.3	Comme décrit dans l'IEC 61158-6-2:2014, 7.2.3
Get_attribute_single			
Remise à zéro			
Apply_attributes			
Receive_data			
Send_message			
Expiration du temporisateur d'inactivité/surveillance			
^a Chaque fois qu'un objet connexion d'un jeu de connexions maître/esclave prédéfini quitte l'état établi, il peut être nécessaire de libérer automatiquement le jeu complet de connexions maître/esclave prédéfini. Lorsque le jeu de connexions maître/esclave prédéfini est libéré, tous les objets connexion associés repassent à l'état non existant. Voir 5.5.3.3 pour plus de détails concernant la libération automatique du jeu de connexions maître/esclave prédéfini.			

5.5.4.4 Règles d'accès aux attributs d'instance de connexion

Le paragraphe 6.2.3.2.1.4 de l'IEC 61158-5-2:2014 donne une description générale des règles d'accès associées aux objets connexion. Le Tableau 21 indique les règles d'accès spécifiques aux objets connexion d'E/S maître/esclave prédéfinie et conserve et/ou remplace les règles présentées en 6.2.3.2.1.4 de l'IEC 61158-5-2:2014.

Tableau 21 – Accès aux attributs de l'objet connexion d'E/S maître/esclave prédéfinie

Attribut	État de la connexion d'E/S			
	Non existant	Configuration	Établi	Délai expiré
État	Comme décrit dans l'IEC 61158-5-2:2014, 6.2.3.2.1.4	Comme décrit dans l'IEC 61158-5-2:2014, 6.2.3.2.1.4	Comme décrit dans l'IEC 61158-5-2:2014, 6.2.3.2.1.4	Comme décrit dans l'IEC 61158-5-2:2014, 6.2.3.2.1.4
Instance_type		Get/set ^a		
Transportclass_trigger		Get only	Get only	Get only
Produced_connection_id		Get only ^c	Get only ^c	Get only ^c
Consumed_connection_id		Get only	Get only	Get only
Initial_comm_characteristics		Get/set ^b	Comme décrit dans l'IEC 61158-5-2:2014, 6.2.3.2.1.4	Comme décrit dans l'IEC 61158-5-2:2014, 6.2.3.2.1.4
Produced_connection_size		Get/set		
Consumed_connection_size				
Expected_packet_rate				
Watchdog_timeout_action				
Produced_connection_path_length				
Produced_connection_path				
Consumed_connection_path_length				
Consumed_connection_path				

^a L'attribut transportclass_trigger de l'objet connexion d'E/S maître/esclave prédéfinie d'un appareil esclave ne doit être modifiable qu'avec l'une des valeurs suivantes: 0x82 – serveur/classe de transport 2, 0x83 – serveur/classe de transport 3.

^b L'attribut produced_connection_size de la connexion d'E/S requête multiple ne doit pas pouvoir être fixé à une valeur supérieure à 8.

^c Cet attribut doit avoir un accès Get/set dans la connexion interrogation multidestinataire.

5.5.5 Caractéristiques de l'objet connexion maître

Cette partie ne décrit pas de caractéristiques pour les objets connexion d'un maître. Le maître doit adopter un comportement externe qui lui permette d'interagir avec ses esclaves.

5.5.6 Messages de commande/réponse requête multiple

5.5.6.1 Généralités

Les messages de commande et réponse requête multiple transfèrent des trames de données d'E/S courtes entre un maître et ses esclaves en mode requête multiple.

5.5.6.2 Message de commande requête multiple

La commande requête multiple envoie un bit de données de sortie à chaque esclave.

Le message de commande requête multiple contient une chaîne de 64 bits (8 octets) de données de sortie, un bit de sortie pour chaque MAC ID possible sur la liaison.

Un appareil esclave peut être conçu pour accomplir l'une des tâches ou toutes les tâches suivantes:

- ignorer la commande requête multiple;
- consommer la commande requête multiple ainsi que ses données de sortie;
- consommer la commande requête multiple en tant que déclencheur et ignorer les données de sortie.

Par défaut, les esclaves doivent ignorer la commande requête multiple tant que la connexion requête multiple n'a pas été attribuée et établie.

Un message de commande requête multiple transmis sans données dans le champ de données CAN est interprété comme étant un événement `receive_idle` par un objet applicatif. Un message de commande requête multiple qui contient des données est interprété comme étant un événement d'exécution par un objet applicatif. Le comportement d'un objet applicatif sur détection de `receive_idle` ou d'un événement d'exécution est spécifique à l'objet applicatif. Voir la description des objets applicatifs pour obtenir plus d'informations sur ces événements.

5.5.6.3 Message de réponse requête multiple

La réponse requête multiple renvoie jusqu'à huit octets de données d'entrée au maître en provenance de chaque esclave.

Un message de réponse requête multiple qui ne contient aucune donnée alors qu'il est configuré pour en contenir indique comme événement "pas de données requête multiple valides pour le maître". Le comportement du maître, sur détection de cet événement, est spécifique à une réalisation.

5.5.6.4 Caractéristiques d'un message requête multiple

Le maître met en œuvre les ressources de communication associées à la commande requête multiple en définissant une connexion client de classe de transport 0 pour transmettre la commande et un ensemble d'objets connexion serveur de classe de transport 0 pour recevoir les réponses.

L'esclave met en œuvre une seule connexion serveur de classe de transport 2 ou 3 pour recevoir la commande requête multiple et renvoyer la réponse associée.

5.5.7 Messages de commande/réponse interrogation

5.5.7.1 Généralités

Les messages de commande et réponse interrogation transfèrent des données d'E/S entre un maître et ses esclaves interrogés.

5.5.7.2 Message de commande interrogation

La commande interrogation envoie jusqu'à 65 535 octets de données de sortie (non fragmentées ou fragmentées) à l'appareil esclave de destination.

Un appareil esclave a la possibilité d'accomplir l'une ou l'ensemble des tâches suivantes:

- ignorer la commande interrogation si aucune connexion interrogation n'est attribuée;
- consommer la commande interrogation ainsi que ses données de sortie;
- consommer la commande interrogation en tant que déclencheur et ignorer les données de sortie.

Par défaut, les esclaves doivent ignorer la commande interrogation tant que la connexion interrogation n'a pas été attribuée et établie.

Un message de commande interrogation transmis sans données dans le champ de données CAN est interprété comme étant un événement `receive_idle` par un objet applicatif. Un message de commande interrogation qui contient des données est interprété comme étant un événement d'exécution par un objet applicatif. Le comportement d'un objet applicatif sur détection de `receive_idle` ou d'un événement d'exécution est spécifique à l'objet applicatif. Voir la description des objets applicatifs pour obtenir plus d'informations sur ces événements.

5.5.7.3 Message de réponse interrogation

La réponse interrogation renvoie jusqu'à 65 535 octets (non fragmentés ou fragmentés) de données d'entrée au maître en provenance de l'esclave.

Un message de réponse interrogation qui ne contient aucune donnée alors qu'il est configuré pour en contenir indique comme événement "pas de données interrogation valides pour l'appareil maître". Le comportement du maître, sur détection de cet événement, dépend de la réalisation.

5.5.8 Messages de commande/réponse interrogation multidestinataire

5.5.8.1 Généralités

Les messages de commande et réponse interrogation multidestinataire transfèrent des données d'E/S entre un maître et des esclaves sur interrogation multidestinataire.

5.5.8.2 Messages de commande interrogation multidestinataire

La commande interrogation multidestinataire envoie jusqu'à 65 535 octets de données de sortie (non fragmentés ou fragmentés) à l'appareil esclave de destination.

Un appareil esclave a la possibilité d'accomplir l'une ou l'ensemble des tâches suivantes:

- ignorer la commande interrogation multidestinataire si aucune connexion interrogation multidestinataire n'est attribuée;
- consommer la commande interrogation multidestinataire ainsi que ses données de sortie;
- consommer la commande interrogation multidestinataire en tant que déclencheur et ignorer les données de sortie.

Par défaut, les esclaves doivent ignorer la commande interrogation multidestinataire tant que la connexion interrogation multidestinataire n'a pas été attribuée et établie.

Un message de commande interrogation multidestinataire transmis sans données dans le champ de données CAN est interprété comme étant un événement `receive_idle` par un objet applicatif. Un message de commande interrogation multidestinataire qui contient des données est interprété comme étant un événement d'exécution par un objet applicatif. Le comportement d'un objet applicatif sur détection de `receive_idle` ou d'un événement d'exécution est spécifique à l'objet applicatif. Voir la description des objets applicatifs pour obtenir plus d'informations sur ces événements.

Le message de commande interrogation multidestinataire utilise un MAC ID multidestinataire au lieu d'un MAC ID destination dans le champ MAC ID de l'identificateur de CAN. Le maître doit attribuer au MAC ID multidestinataire une valeur correspondant à la valeur du MAC ID de l'un des membres esclaves dans le groupe multidestinataire ou à sa propre valeur. Lorsque le MAC ID multidestinataire est le MAC ID du maître, il ne peut être utilisé que pour un groupe multidestinataire et empêche le maître d'être l'esclave d'un autre maître dans un groupe multidestinataire.

5.5.8.3 Message de réponse interrogation multidestinataire

La réponse interrogation multidestinataire renvoie jusqu'à 65 535 octets (non fragmentés ou fragmentés) de données d'entrée au maître en provenance de l'esclave.

Un message de réponse interrogation multidestinataire qui ne contient aucune donnée alors qu'il est configuré pour en contenir indique comme événement "pas de données interrogation multidestinataire valides pour l'appareil maître". Le comportement du maître, sur détection de cet événement, dépend de la réalisation.

5.5.9 Connexions cyclique/changement d'état

5.5.9.1 Généralités

Le jeu de connexions maître/esclave prédéfini permet le transfert de données sur changement d'état ou cyclique entre le maître et l'esclave. Ce transfert de données peut être acquitté ou non acquitté.

Les jeux de connexions cyclique/changement d'état utilisent l'instance de connexion 2 pour le transfert du maître à l'esclave et l'acquiescement de l'esclave envers le maître. L'instance de connexion 4 est utilisée pour le transfert de données de l'esclave au maître et l'acquiescement du maître envers l'esclave. Si un appareil n'accepte pas le mode interrogation et ne prévoit aucune donnée de sortie, l'instance de connexion 2 n'a pas besoin d'être créée.

Pour garantir un fonctionnement cohérent du système, le chemin de consommation de la connexion changement d'état/cyclique de l'esclave (instance 4) doit spécifier l'objet gestionnaire d'acquiescement; l'instance de l'objet gestionnaire d'acquiescement doit être 1.

Du fait que les jeux de connexion interrogation et changement d'état/cyclique partagent l'instance de connexion 2, l'esclave doit suivre certaines procédures basées sur la requête d'attribution afin d'assurer un comportement correct. Comme décrit ci-dessus, les instances de connexion 2 et 4 ne sont créées que lorsque le bit d'attribution changement d'état/cyclique est mis à 1. L'instance de connexion 4 produit des données d'entrée à partir du chemin par défaut et consomme les acquiescements dirigés vers l'instance 1 de l'objet gestionnaire d'acquiescement. L'instance de connexion 2 consomme des données de sortie dirigées vers le chemin par défaut et produit un acquiescement de longueur nulle. Lorsqu'une requête d'attribution ne contient que le bit d'attribution interrogation, l'instance de connexion 2 consomme des données de sortie dirigées vers le chemin de sortie par défaut et produit des données d'entrée à partir du chemin d'entrée par défaut.

Si les deux bits d'attribution changement d'état/cyclique et interrogation sont mis à 1, l'instance de connexion 2 doit se comporter comme si seul le bit d'attribution "interrogation" était à 1, et l'instance de connexion 4 doit continuer à se comporter comme décrit ci-dessus. C'est aussi le comportement requis si le maître attribue le jeu de connexions "interrogation" dans un seul message, puis le jeu de connexions changement d'état/cyclique dans un message ultérieur.

Pour accomplir une production de données non acquittées, le bit de suppression d'acquiescement est mis à 1 avec le bit changement d'état ou le bit cyclique de l'octet choix d'attribution. L'instance de connexion 2 est configurée comme connexion de classe de transport 0 pour la production de données du maître vers l'esclave. L'instance de connexion 4 est configurée comme connexion de classe de transport 0 pour la production de données de l'esclave vers le maître. Lorsque le bit interrogation est mis à 1 dans le même choix d'attribution, cela entraîne la configuration de l'instance de connexion 2 pour le mode interrogation, alors que l'instance de connexion 4 continue à être configurée comme une connexion de changement d'état ou cyclique non acquittée.

5.5.9.2 Acquiescement du déclencheur de messages changement d'état/cyclique

5.5.9.2.1 Généralités

Les messages changement d'état/cyclique transfèrent jusqu'à 65 535 octets de données entre un maître et un esclave en utilisant des déclencheurs de production sur changement d'état ou bien cycliques. La production des données peut être acquittée ou non acquittée.

Les paragraphes suivants décrivent:

- le message changement d'état/cyclique (acquitté);
- le message changement d'état/cyclique (non acquitté);

- les caractéristiques du message changement d'état/cyclique;
- un exemple d'application changement d'état/cyclique.

5.5.9.2.2 Message changement d'état/cyclique du maître

Le message changement d'état/cyclique du maître envoie jusqu'à 65 535 octets de données (non fragmentés ou fragmentés) à l'appareil esclave de destination. La production des données est déclenchée par un changement d'état ou par l'expiration du délai du déclencheur de transmission.

Un message changement d'état/cyclique transmis sans données dans le champ de données CAN est interprété comme étant un événement `receive_idle` par un objet applicatif. Un message changement d'état/cyclique qui contient des données est interprété comme étant un événement d'exécution par un objet applicatif. Le comportement d'un objet applicatif sur détection de l'événement d'exécution est spécifique à l'objet applicatif. Voir la description des objets applicatifs pour obtenir plus d'informations sur ces événements.

Par défaut, les esclaves doivent ignorer le message changement d'état/cyclique du maître tant que la connexion changement d'état/cyclique est attribuée et établie.

5.5.9.2.3 Message d'acquittement changement d'état/cyclique de l'esclave

Le message d'acquittement/changement d'état de l'esclave renvoie jusqu'à 65 535 octets de données (non fragmentés ou fragmentés) au maître en provenance de l'esclave. Par défaut, le message d'acquittement a une longueur nulle.

5.5.9.2.4 Message changement d'état/cyclique de l'esclave

Le message changement d'état/cyclique de l'esclave envoie jusqu'à 65 535 octets de données (non fragmentés ou fragmentés) au maître en provenance de l'esclave. La production de données est déclenchée par un changement d'état ou par l'expiration du délai du déclencheur de transmission.

Un message changement d'état/cyclique qui ne contient aucune données alors qu'il est configuré pour en contenir indique comme événement "pas de données changement d'état/cyclique valides pour l'appareil maître". Le comportement du maître, sur détection de cet événement, dépend de la réalisation.

5.5.9.2.5 Message d'acquittement changement d'état/cyclique du maître

Le message d'acquittement changement d'état/cyclique du maître renvoie jusqu'à 65 535 octets de données (non fragmentés ou fragmentés) à l'esclave en provenance du maître. Par défaut, le message d'acquittement a une longueur nulle.

5.5.10 Appareils du groupe 2 seulement

Pour établir des communications avec un appareil du groupe 2 seulement, un client doit attribuer le jeu de connexions maître/esclave prédéfini. La requête d'attribution d'un appareil du groupe 2 seulement est transmise en tant que requête explicite non connectée du groupe 2 seulement. Au lieu d'utiliser l'UCMM pour établir les connexions de messagerie explicite, les appareils du groupe 2 seulement reçoivent et traitent les messages de requête explicite non connectés du groupe 2 seulement.

Les seuls services valides qui peuvent être transmis en tant que messages de requête explicite non connectés du groupe 2 seulement, sont les suivants:

- message `allocate_master/slave_connection_set`;
- message `release_master/slave_connection_set`.

Ces services sont décrits en 5.5.3.

Les réponses aux requêtes explicites non connectées du groupe 2 seulement sont renvoyées par transmission d'un message du groupe 2 dont l'élément MAC ID est chargé avec le MAC ID de l'appareil qui répond (MAC ID source), sachant que l'ID de message est fixé à 3.

Les appareils du groupe 2 seulement ne doivent utiliser le champ identificateur de message de réponse explicite de l'esclave que pour transmettre des messages de réponse non connectés du groupe 2 seulement et des messages de réponse pour la connexion de messagerie explicite maître/esclave prédéfinie.

Si un serveur du groupe 2 seulement reçoit un message de requête non connecté du groupe 2 seulement qui n'est pas une requête `allocate_master/slave_connection_set` ou `release_master/slave_connection_set`, une réponse d'erreur doit alors être renvoyée, avec le code d'erreur général fixé à 02 et le code d'erreur supplémentaire fixé à la valeur 0x03, spécifique à l'objet DeviceNet (voir 5.5.3.5).

5.6 CIP Safety™ sur DeviceNet

5.6.1 Généralités

CIP Safety™⁴ sur DeviceNet utilise une couche de communication de sécurité dédiée au-dessus de la couche d'application du protocole DeviceNet de base: CIP Safety sur DeviceNet utilise les objets et services de communication DeviceNet de base en combinaison avec les objets et services spécifiques relatifs à la sécurité, définis comme partie du profil de communication de sécurité fonctionnelle CIP Safety. CIP Safety est entièrement spécifié dans l'IEC 61784-3-2, y compris les adaptations spécifiques pour DeviceNet.

Lorsqu'il est mis en application comme partie d'un système relatif à la sécurité conformément à l'IEC 61508, CIP Safety sur DeviceNet fournit le degré de confiance nécessaire dans le transport des informations entre deux ou plusieurs CDI, ou dans le fonctionnement sans danger en cas de défaillances du CDI. Il peut être utilisé pour des applications exigeant une sécurité fonctionnelle jusqu'au niveau d'intégrité de sécurité (SIL, Safety Integrity Level) spécifié dans l'IEC 61784-3-2.

NOTE La revendication de SIL résultante d'un système dépend de la mise en application du profil de communication de sécurité fonctionnelle choisi dans ce système; la mise en application d'un profil de communication de sécurité fonctionnelle conformément à la présente partie dans un dispositif normalisé n'est pas suffisante pour le qualifier de dispositif de sécurité.

5.6.2 Utilisation d'identificateurs de CAN pour CIP Safety sur DeviceNet

L'attribution d'identificateurs de CAN pour CIP Safety sur DeviceNet est définie dans l'IEC 61784-3-2. La définition est présentée de telle sorte qu'elle n'interfère pas avec l'utilisation d'un identificateur standard prédéfini.

5.7 Couche physique

5.7.1 Généralités

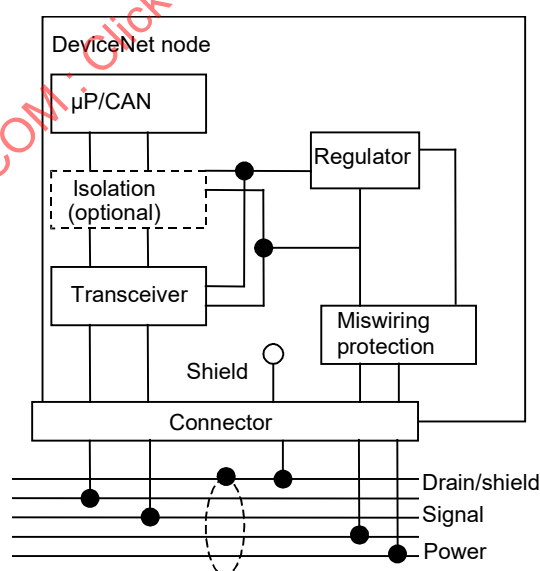
Le Tableau 22 donne les spécifications générales de la couche physique qui doivent être respectées dans les systèmes DeviceNet (voir 9.2.6 pour les spécifications d'essais).

⁴ CIP Safety™ est une marque commerciale d'ODVA, Inc. Cette information est donnée à l'intention des utilisateurs de la présente Norme internationale et ne signifie nullement que l'IEC approuve ou recommande le détenteur de cette marque commerciale ou d'un quelconque de ses produits. La conformité à la présente norme ne nécessite pas l'utilisation de la marque commerciale CIP Safety™. L'utilisation de la marque commerciale CIP Safety™ nécessite l'autorisation de ODVA, Inc.

Tableau 22 – Caractéristiques générales de la couche physique

Caractéristique	Spécification
Vitesses de transmission	125 kbit/s, 250 kbit/s, 500 kbit/s
Longueur totale maximale de la ligne principale et longueur maximale de câble entre deux appareils quelconques	500 m à 125 kbit/s 250 m à 250 kbit/s 100 m à 500 kbit/s
Nombre de stations accepté par le transcepteur	64 minimum
Signalisation	Conformément à l'ISO 11898-1:2003 et à l'ISO 11898-2:2003
Modulation	Bande de base
Codage	NRZ avec bits de remplissage
Couplage au support de transmission	Tx/Rx à couplage différentiel c.c.
Isolation (entre le circuit transcepteur et le composant CAN)	500 V c.c. Ne concerne que les appareils isolés
Impédance d'entrée différentielle typique (état récessif)	Shunt C = 5 pF Shunt R = 25 kΩ (sous tension)
Impédance d'entrée différentielle minimale (état récessif)	Shunt C = 24 pF Shunt R = 20 kΩ (sous tension)
Tension sur les conducteurs d'alimentation	11 V c.c. à 25 V c.c.
Plage maximale de tension des signaux	-25 V c.c. à +18 V c.c. (CAN_H, CAN_L) ^a
^a Les tensions à CAN_H et à CAN_L doivent avoir pour référence la terre du transcepteur. Le potentiel à la terre du transcepteur peut être supérieur à celui de la borne V- d'une quantité égale à la chute de tension au travers d'une diode Schottky ou équivalent. Cette tension doit être de 0,6 V maximum relativement à V-.	

La couche physique, comprenant le transcepteur, le connecteur, le circuit de protection contre les erreurs de câblage, le régulateur de tension, le support de transmission et l'isolation (en option), doit être réalisée comme indiqué par la Figure 47. Les essais permettant de vérifier la robustesse de la réalisation de la couche physique dans un appareil sont spécifiés en 9.2.9.



IEC

Légende

Anglais	Français
DeviceNet node	Station DeviceNet
Regulator	Régulateur
Isolation (optional)	Isolation (en option)