BS EN 62026-3:2015

BSI Standards Publication

Low-voltage switchgear and controlgear — Controller-device interfaces (CDIs)

Part 3: DeviceNet

bsi.

...making excellence a habit.™

National foreword

This British Standard is the UK implementation of EN 62026-3:2015. It is identical to IEC 62026-3:2014, incorporating corrigendum March 2015. It supersedes BS EN 62026-3:2009 which is withdrawn.

The start and finish of text introduced or altered by corrigendum is indicated in the text by tags. Text altered by IEC corrigendum March 2015 is indicated in the text by [AC1) (AC1].

The UK participation in its preparation was entrusted by Technical Committee PEL/121, Switchgear and Controlgear and their assemblies for low voltage, to Subcommittee PEL/121/1, Low voltage switchgear and controlgear.

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

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

Published by BSI Standards Limited 2015

ISBN 978 0 580 78536 8

ICS 29.130.20; 35.200

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

This British Standard was published under the authority of the Standards Policy and Strategy Committee on 31 May 2015.

Amendments/corrigenda issued since publication

Date	Text affected

EUROPEAN STANDARD

NORME EUROPÉENNE

EUROPÄISCHE NORM

EN 62026-3

April 2015

ICS 29.130.20

Supersedes EN 62026-3:2009

English Version

Low-voltage switchgear and controlgear - Controller-device interfaces (CDIs) - Part 3: DeviceNet (IEC 62026-3:2014 + COR1:2015)

Appareillage à basse tension - Interfaces appareil de commande-appareil (CDI) - Partie 3: DeviceNet (IEC 62026-3:2014 + COR1:2015)

Niederspannungsschaltgeräte - Steuerung-Geräte-Netzwerke (CDIs) - Teil 3: DeviceNet (IEC 62026-3:2014 + COR1:2015)

This European Standard was approved by CENELEC on 2014-10-29. CENELEC members are bound to comply with the CEN/CENELEC Internal Regulations which stipulate the conditions for giving this European Standard the status of a national standard without any alteration.

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

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

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

CENELEC

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

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

Ref. No. EN 62026-3:2015 E

Foreword

The text of document 17B/1814/CDV, future edition 3 of IEC 62026-3 + Corrigendum March 2015, prepared by SC 121A "Low-voltage switchgear and controlgear" of IEC/TC 121 " Switchgear and controlgear and their assemblies for low voltage" was submitted to the IEC-CENELEC parallel vote and approved by CENELEC as EN 62026-3:2015.

The following dates are fixed:

• latest date by which the document has to be implemented at national level by publication of an identical national standard or by endorsement	(dop)	2015-10-03
• latest date by which the national standards conflicting with the document have to be withdrawn	(dow)	2017-09-26

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

This document has been prepared under a mandate given to CENELEC by the European Commission and the European Free Trade Association, and supports essential requirements of EU Directive.

For the relationship with EU Directive see informative Annex ZZ, which is an integral part of this document.

This standard covers the Principle Elements of the Safety Objectives for Electrical Equipment Designed for Use within Certain Voltage Limits (LVD - 2006/95/EC).

Endorsement notice

The text of the International Standard IEC 62026-3:2014 + Corrigendum March 2015 was approved by CENELEC as a European Standard without any modification.

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

IEC 61131-3:2013	NOTE	Harmonized as EN 61131-3:2013.
IEC 61158 (Series)	NOTE	Harmonized as EN 61158 (Series).
IEC 61508 (Series)	NOTE	Harmonized as EN 61508 (Series).
IEC 61784-1	NOTE	Harmonized as EN 61784-1.
IEC 61784-5-2	NOTE	Harmonized as EN 61784-5-2..
ISO/IEC 7498-1:1994	NOTE	Harmonized as EN ISO/IEC 7498-1:1995.

Annex ZA
(normative)

Normative references to international publications with their corresponding European publications

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

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

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

Publication	Year	Title	EN/HD	Year
IEC 60529	1989	Degrees of protection provided by enclosures (IP Code)	EN 60529	1991
			+EN 60529:1991/corrigendum May 1993	1993
IEC 60529:1989/A1	1999		EN 60529:1991/A1	2000
IEC 60529:1989/A2	2013		EN 60529:1991/A2	2013
IEC 60947-5-2	2007	Low-voltage switchgear and controlgear -- Part 5-2: Control circuit devices and switching elements - Proximity switches	EN 60947-5-2	2007
+A1	2012		+A1	2012
IEC 61000-4-2	2008	Electromagnetic compatibility (EMC) -- Part 4-2: Testing and measurement techniques - Electrostatic discharge immunity test	EN 61000-4-2	2009
IEC 61000-4-3	2006	Electromagnetic compatibility (EMC) -- Part 4-3: Testing and measurement techniques - Radiated, radio-frequency, electromagnetic field immunity test	EN 61000-4-3	2006
+A1	2007		+A1	2008
+A2	2010		+A2	2010
IEC 61000-4-4	2012	Electromagnetic compatibility (EMC) -- Part 4-4: Testing and measurement techniques - Electrical fast transient/burst immunity test	EN 61000-4-4	2012
IEC 61000-4-5	2005	Electromagnetic compatibility (EMC) -- Part 4-5: Testing and measurement techniques - Surge immunity test	EN 61000-4-5	2006
IEC 61000-4-6	2013	Electromagnetic compatibility (EMC) -- Part 4-6: Testing and measurement techniques - Immunity to conducted disturbances, induced by radio-frequency fields	EN 61000-4-6	2014
IEC 61158-4-2	2014	Industrial communication networks - Fieldbus specifications - Part 4-2: Data-link layer protocol specification - Type 2 elements	EN 61158-4-2	2014
IEC 61158-5-2	2014	Industrial communication networks - Fieldbus specifications -- Part 5-2: Application layer service definition - Type 2 elements	EN 61158-5-2	2014

IEC 61158-6-2	2014	Industrial communication networks - Fieldbus specifications - Part 6-2: Application layer protocol specification - Type 2 elements	EN 61158-6-2	2014
IEC 61784-3-2	-	Industrial communication networks - Profiles -- Part 3-2: Functional safety fieldbuses - Additional specifications for CPF 2	EN 61784-3-2	-
IEC 62026-1	2007	Low-voltage switchgear and controlgear - Controller-device interfaces (CDIs) -- Part 1: General rules	EN 62026-1	2007
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		Hydraulic Fluid Power Solenoid-piloted Industrial Valves - Interface Dimensions for Electrical Connectors	-	-
ASTM D 4566-94	-	Standard Test Methods for Electrical Performance Properties of Insulations and Jackets for Telecommunications Wire and Cable	-	-
CISPR 11 (mod)	2009	Industrial, scientific and medical equipment - Radio-frequency disturbance characteristics - Limits and methods of measurement	EN 55011	2009
			+prA +AA	
CISPR 11:2009/A1	2010		EN 55011:2009/A1	2010

Annex ZZ
(informative)

Coverage of Essential Requirements of EU Directives

This European Standard has been prepared under a mandate given to CENELEC by the European Commission and the European Free Trade Association and within its scope the standard covers protection requirements of Annex I Article 1 of the EU Directive 2004/108/EC.

Compliance with this standard provides presumption of conformity with the specified essential requirements of the Directives concerned.

NOTE: Other requirements and other EU Directives may be applicable to the products falling within the scope of this standard.

CONTENTS

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.

---

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 4566-94[2], *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.

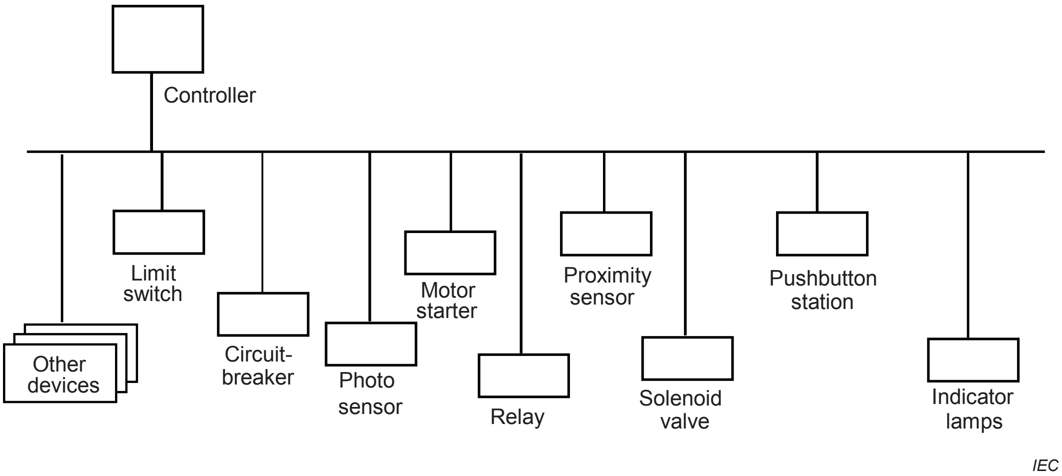

Figure 1 – Typical DeviceNet controller-device interfaces

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.

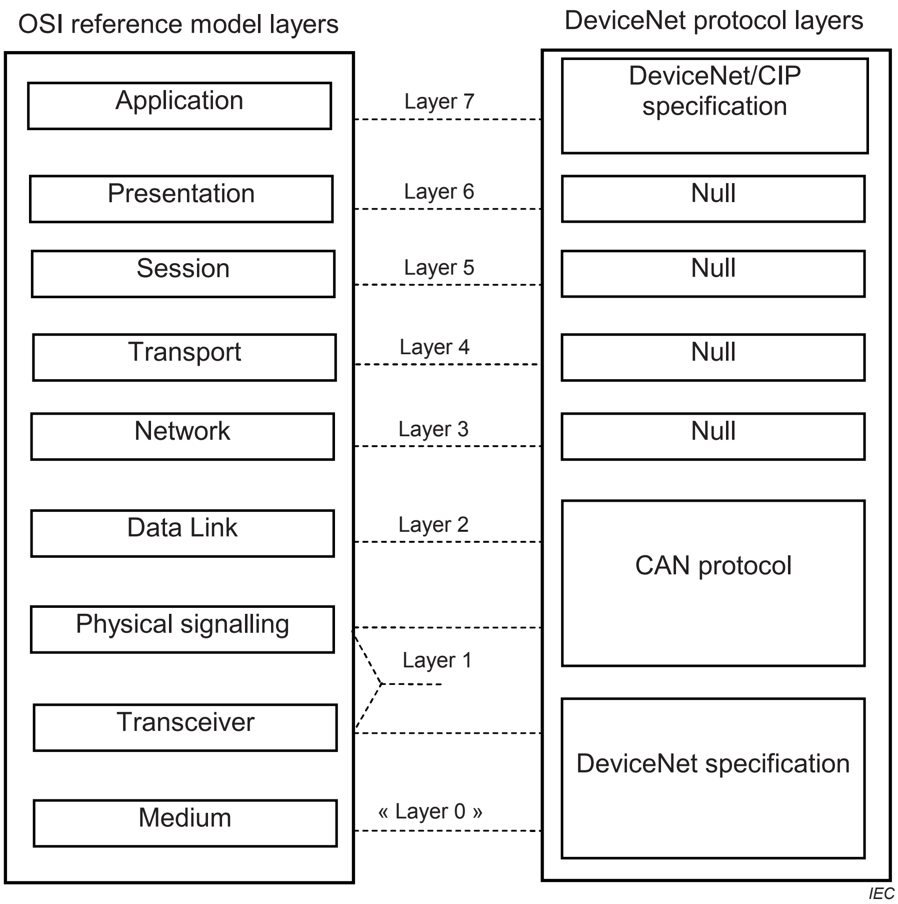

Figure 2 – DeviceNet protocol architecture compared with the OSI reference model

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).

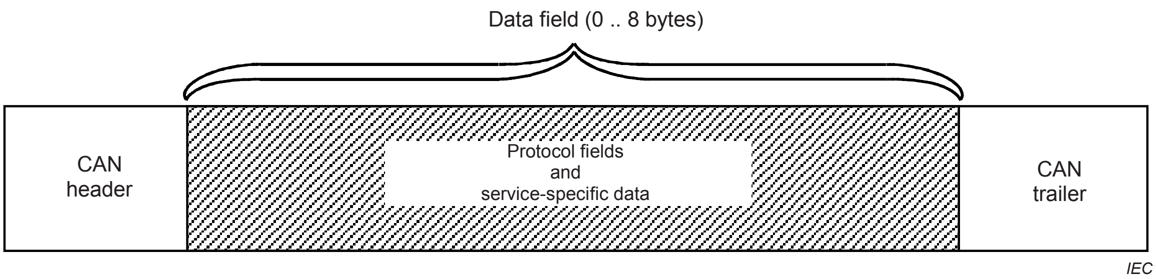

Figure 4 – Explicit message CAN data field use

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

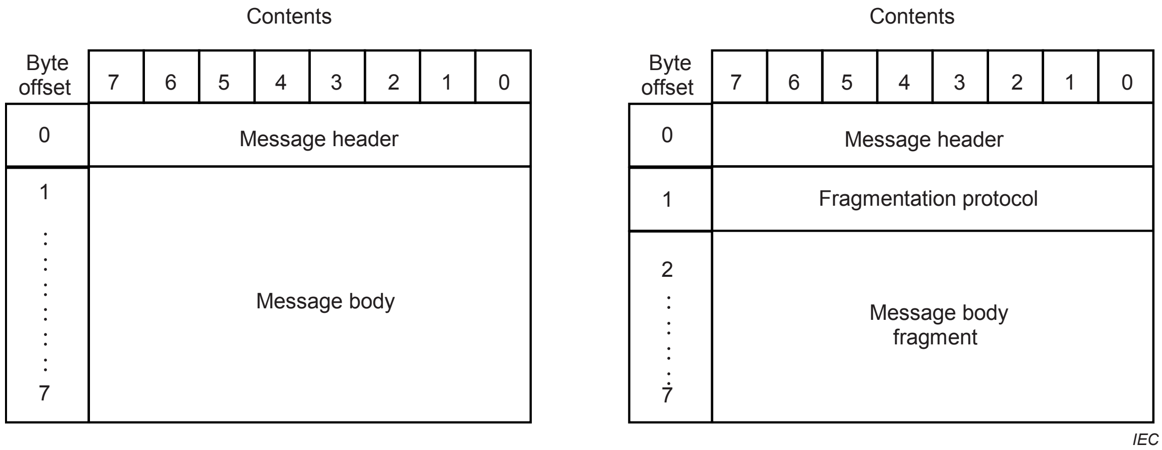

Figure 5 – Explicit message data field format

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:

Contents

Byte offset	7	6	5	4	3	2	1	0
0	Frag	XID	MAC ID					

IEC

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.

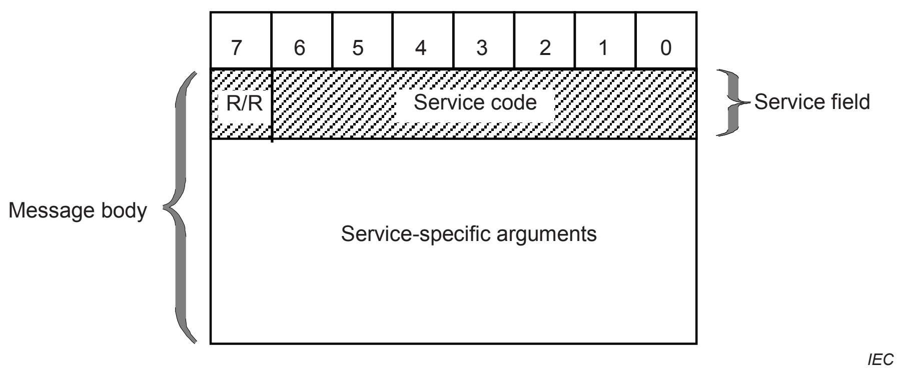

Figure 7 – Service field format

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).

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

IEC

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).

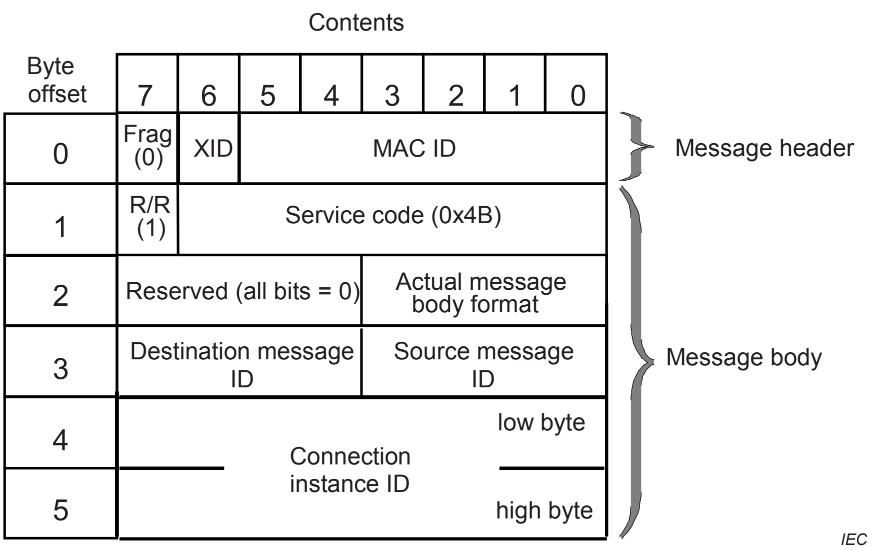

Figure 9 – Open explicit messaging connection response format

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.

Contents

Byte offset	7	6	5	4	3	2	1	0	
0	Frag (0)	XID	MAC ID						Message header
1	R/R (0)	Service code (0x4C)							Message body
2	Connection instance ID low byte								
3	Connection instance ID high byte								

IEC

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.

Contents

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

IEC

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:

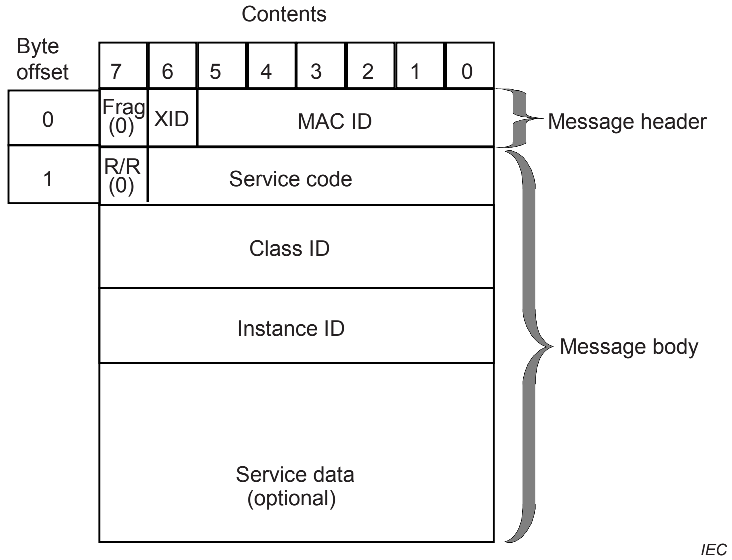

Figure 12 – Non-fragmented explicit request message format, values 0 – 3

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:

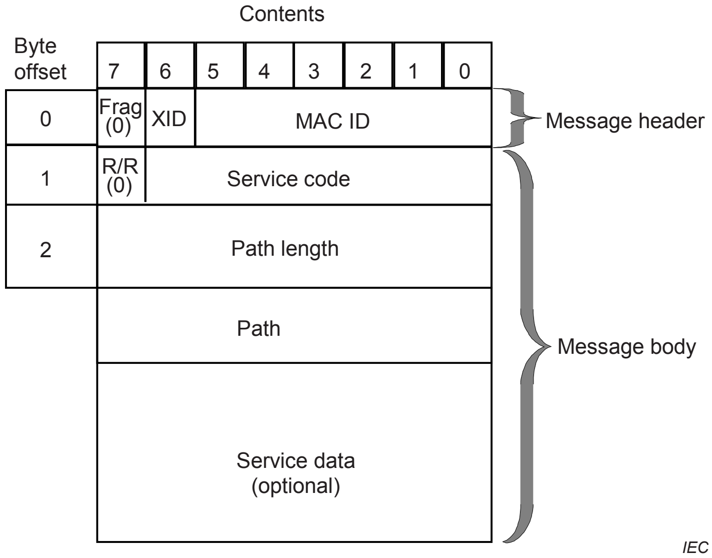

Figure 13 – Non-fragmented explicit request message format, value 4

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:

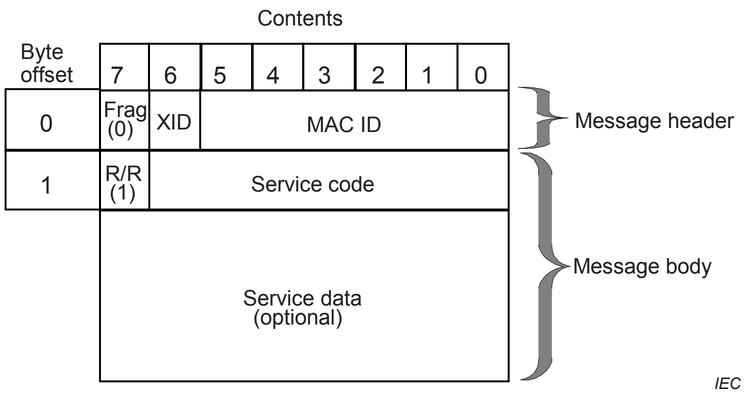

Figure 14 – Non-fragmented success response message format

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.

Byte offset	Contents 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).

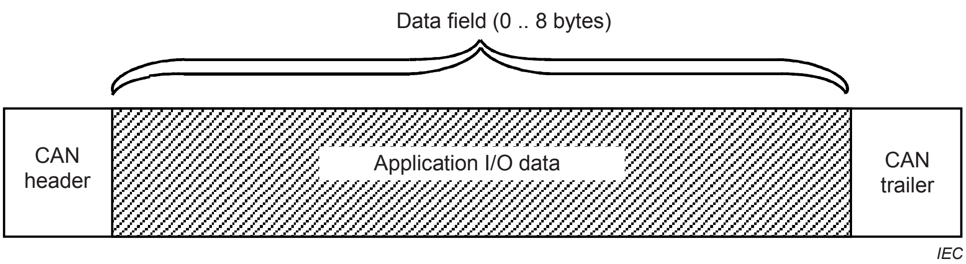

Figure 16 – Data field of an I/O message

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:

Contents

7	6	5	4	3	2	1	0
Fragment type		Fragment count					

IEC

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).

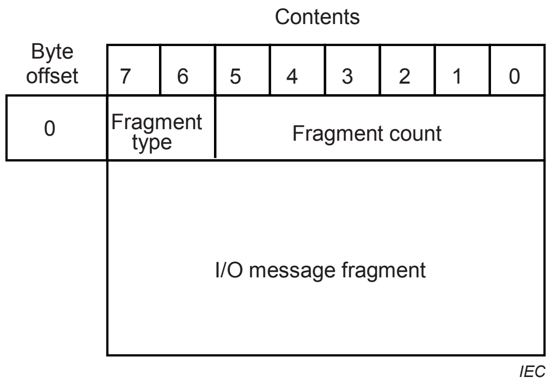

Figure 18 – I/O message fragment format

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

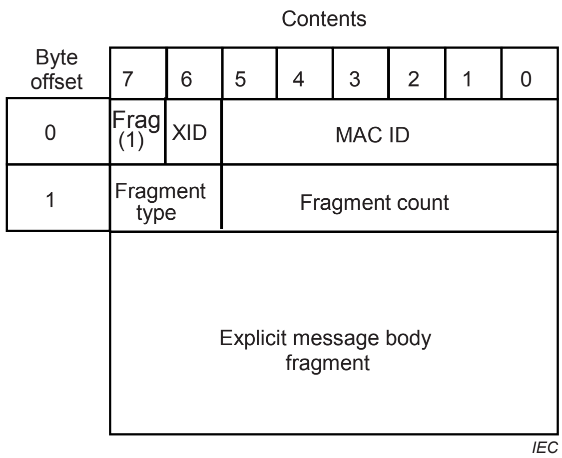

Figure 19 – Explicit message fragment format

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							

IEC

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 sublcause 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) Group ID					Message ID						Identity usage
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.

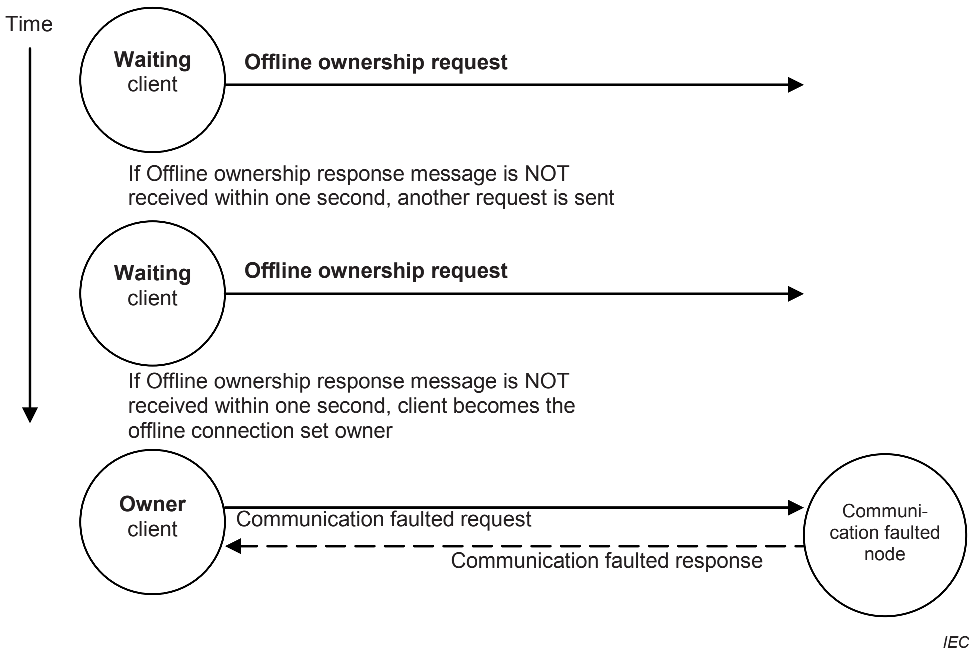

Figure 21 – Establishing the offline ownership

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*.

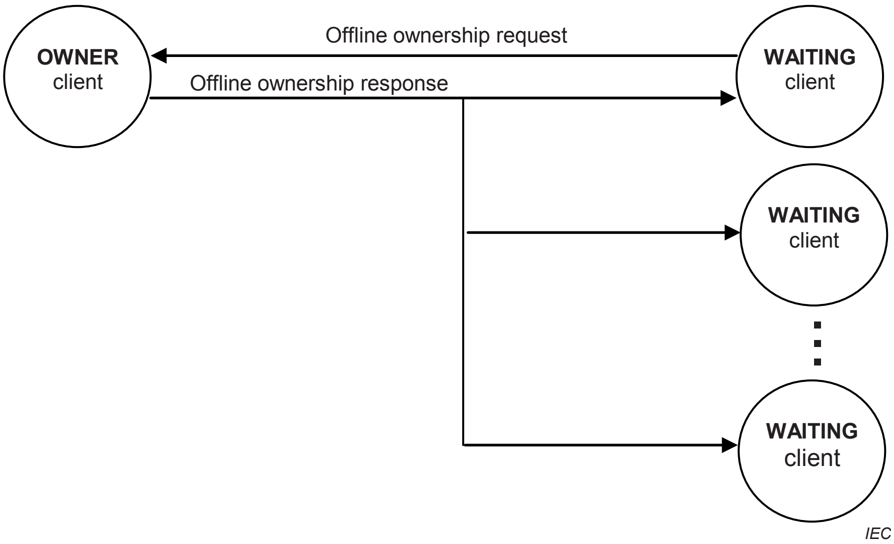

Figure 22 – Multicast nature of the offline ownership

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 [AC1) *text deleted* (AC1]) 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 AC1⟩ *text deleted* ⟨AC1 and the R/R bit is set (1). The response message shall be initiated within 100 ms, upon the receipt of the request message.

Byte offset	Contents 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.

Byte offset	Contents								
	7	6	5	4	3	2	1	0	
0	Frag [0]	XID	MAC ID						Message header
1	R/R [1]	Service code [0x4D]							Message body
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		

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.

<table>
<tr><th rowspan="2">Byte offset</th><th colspan="8">Contents</th><th></th></tr>
<tr><th>7</th><th>6</th><th>5</th><th>4</th><th>3</th><th>2</th><th>1</th><th>0</th><th></th></tr>
<tr><td>0</td><td>Frag [0]</td><td>XID</td><td colspan="6">MAC ID</td><td>Message header</td></tr>
<tr><td>1</td><td>R/R [1]</td><td colspan="7">Service code [0x4E]</td><td rowspan="7">Message body</td></tr>
<tr><td>2</td><td colspan="6" rowspan="2">Class ID</td><td colspan="2">Low byte</td></tr>
<tr><td>3</td><td colspan="2">High byte</td></tr>
<tr><td>4</td><td colspan="6" rowspan="2">Instance ID</td><td colspan="2">Low byte</td></tr>
<tr><td>5</td><td colspan="2">High byte</td></tr>
<tr><td>6</td><td colspan="6" rowspan="2">Shutdown code</td><td colspan="2">Low byte</td></tr>
<tr><td>7</td><td colspan="2">High byte</td></tr>
</table>

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.

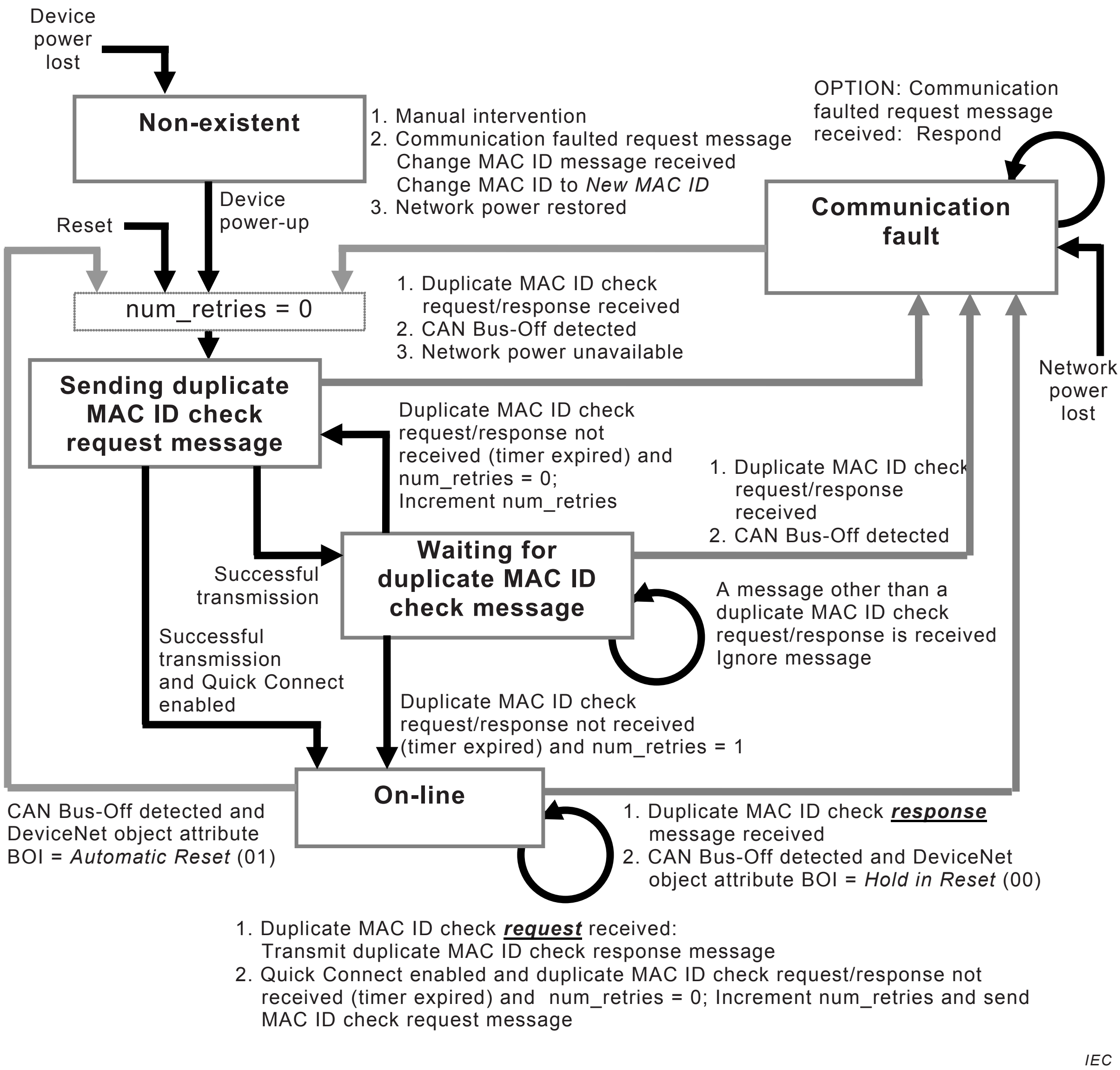

Figure 37 – Link access state transition diagram

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.

Byte offset	Contents 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.

Byte offset	Contents 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.

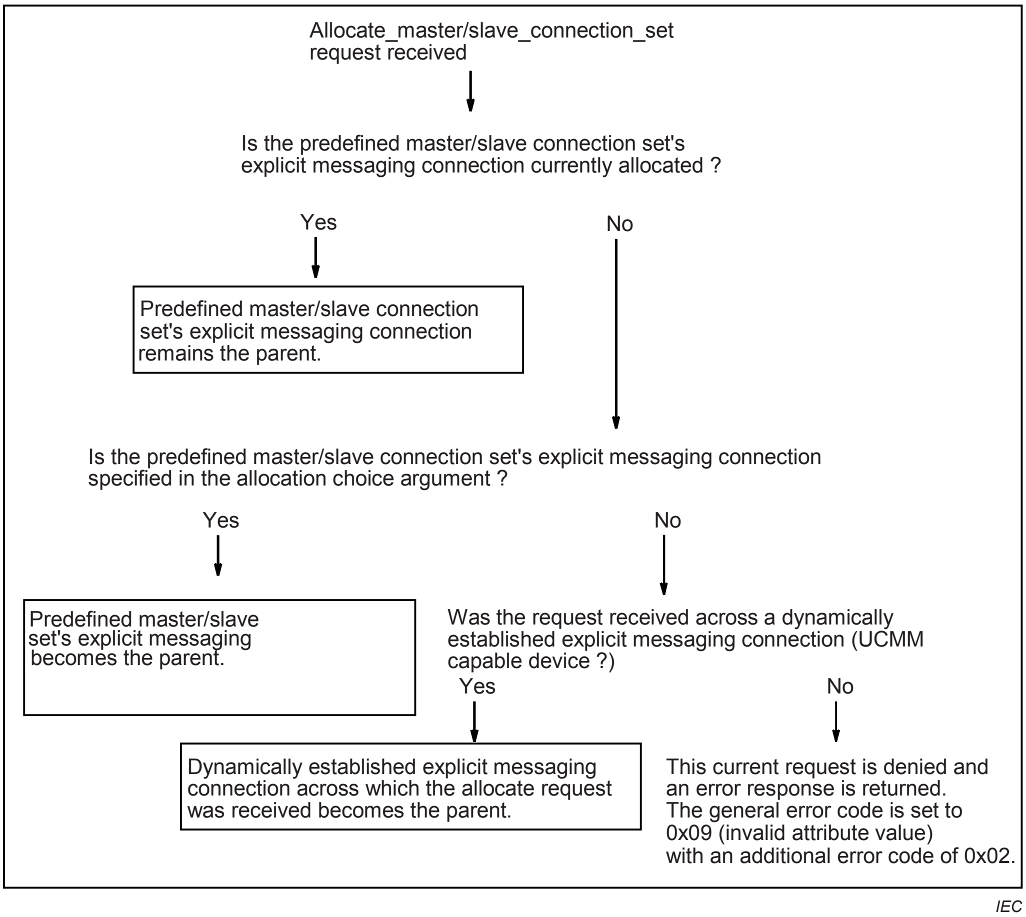

Figure 41 – Parent explicit messaging connection logic

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 [AC1⟩ *text deleted* ⟨AC1]). 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.

<table>
<tr><th></th><th colspan="8">Contents</th></tr>
<tr><th>Byte offset</th><th>7</th><th>6</th><th>5</th><th>4</th><th>3</th><th>2</th><th>1</th><th>0</th></tr>
<tr><td>0</td><td>Frag (0)</td><td>XID</td><td colspan="6">MAC ID</td></tr>
<tr><td>1</td><td>R/R (0)</td><td colspan="7">Service (0x4C)</td></tr>
<tr><td></td><td colspan="8">Class ID (3)</td></tr>
<tr><td></td><td colspan="8">Instance ID (1)</td></tr>
<tr><td></td><td colspan="8">Release choice</td></tr>
</table>

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.

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

IEC

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.

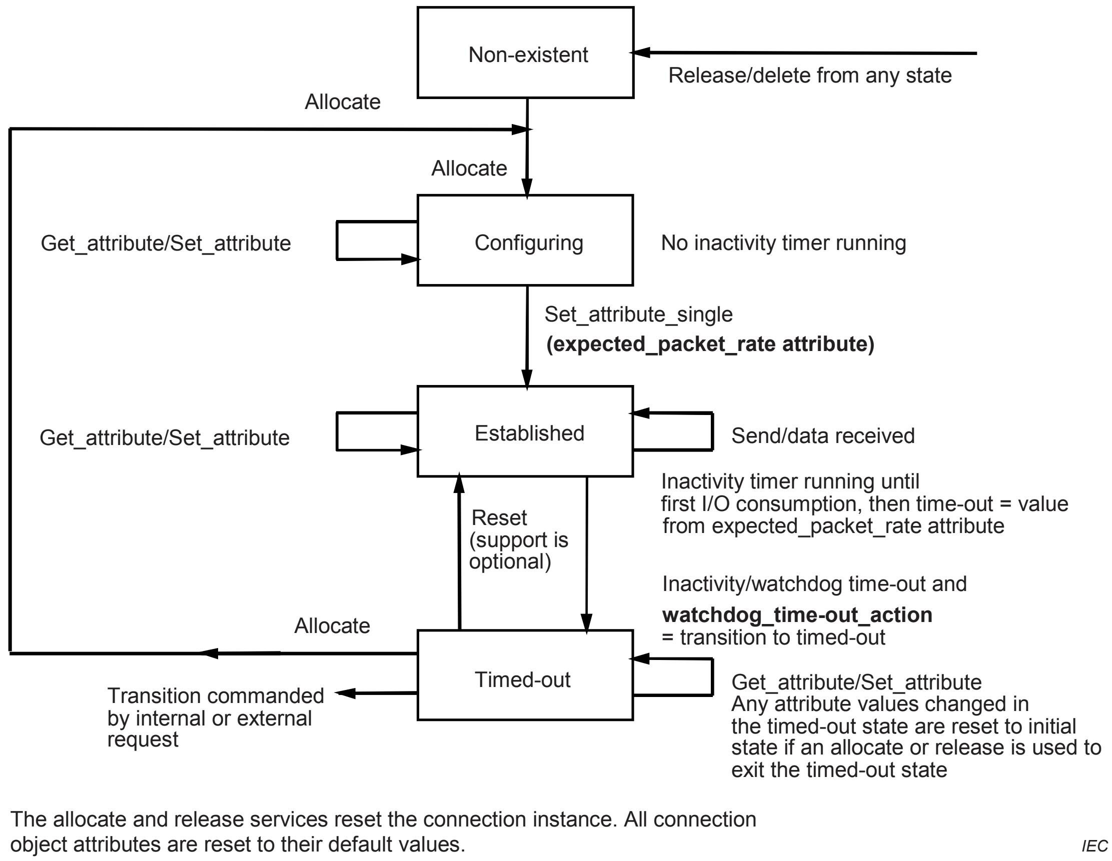

Figure 45 – Predefined master/slave I/O connection state transition diagram

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)*

<table>
<tr><th rowspan="2">Event</th><th colspan="4">I/O connection object state</th></tr>
<tr><th>Non-existent</th><th>Configuring</th><th>Established</th><th>Timed out</th></tr>
<tr><td>Get_attribute_single</td><td>As described in IEC 61158-6-2:2014, 7.2.1</td><td>Validate/service the request according to the access rules presented in 5.5.4.4.
Return appropriate response</td><td>Validate/service the request according to the access rules presented in 5.5.4.4.
Return appropriate response</td><td>Validate/service the request according to the access rules presented in 5.5.4.4.
Return appropriate response</td></tr>
<tr><td>Reset</td><td rowspan="5">As described in IEC 61158-6-2:2014, 7.2.1</td><td>As described in IEC 61158-6-2:2014, 7.2.1</td><td rowspan="5">As described in IEC 61158-6-2:2014, 7.2.1</td><td rowspan="5">As described in IEC 61158-6-2:2014, 7.2.1</td></tr>
<tr><td>Apply_attributes</td><td>As described in IEC 61158-6-2:2014, 7.2.1 [b]</td></tr>
<tr><td>Receive_data</td><td rowspan="3">As described in IEC 61158-6-2:2014, 7.2.1</td></tr>
<tr><td>Send_message</td></tr>
<tr><td>Inactivity/ watchdog timer expires</td></tr>
</table>

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.

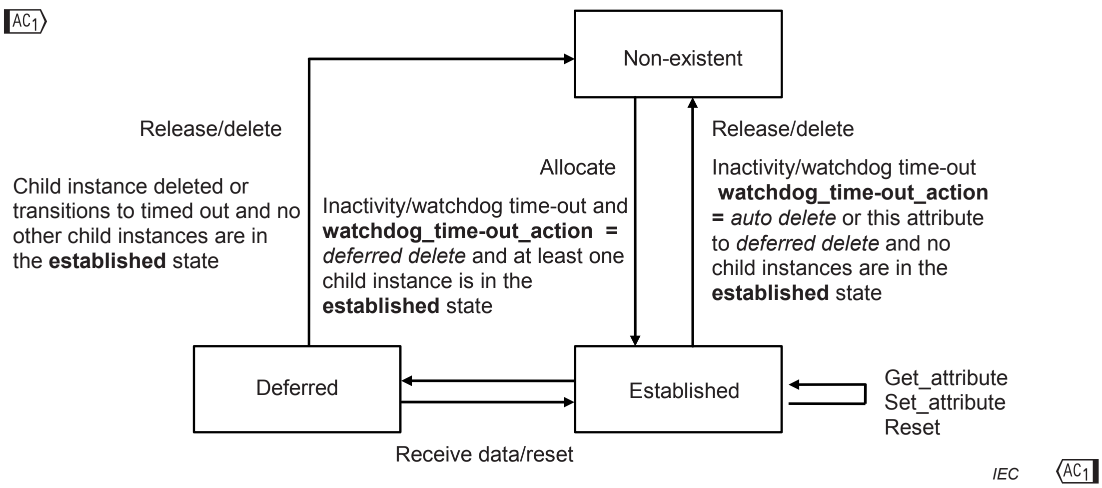

Figure 46 – Predefined master/slave explicit messaging connection state transition diagram

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				
Transportclass_trigger		Get/set [a]		
Produced_connection_id		Get only	Get only	Get only
Consumed_connection_id		Get only [c]	Get only [c]	Get only [c]
Initial_comm_characteristics		Get only	Get only	Get only
Produced_connection_size		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
Consumed_connection_size		Get/set		
Expected_packet_rate		As described in IEC 61158-5-2: 2014, 6.2.3.2.1.4		
Watchdog_timeout_action				
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 kΩ (power on)
Differential input impedance min. (recessive state)	Shunt C = 24 pF Shunt R = 20 kΩ (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.

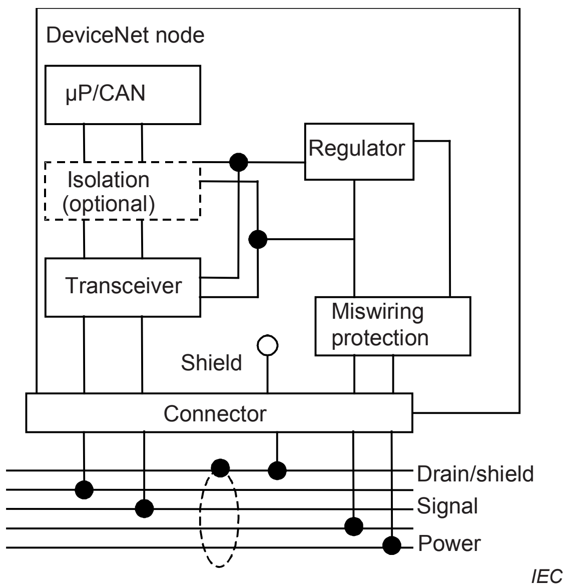

Figure 47 – Physical layer block diagram

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 MΩ, C = 0,01 μF (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.

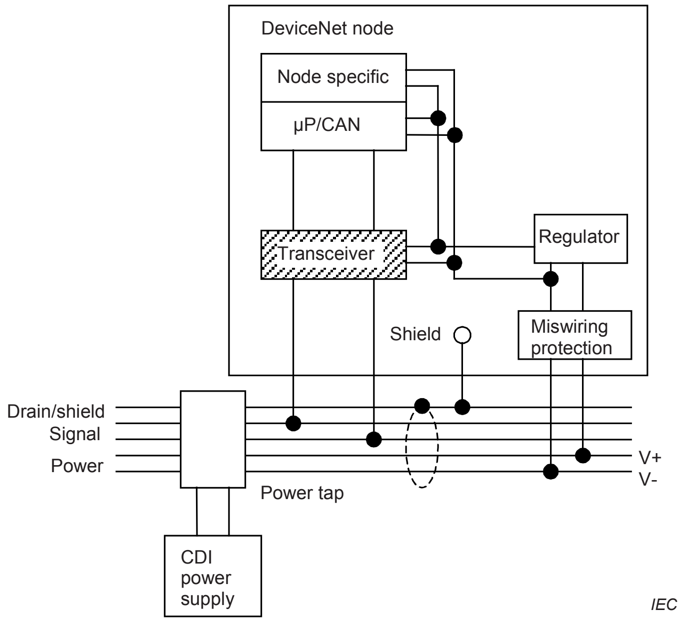

Figure 48 – Device containing a non-isolated physical layer

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.

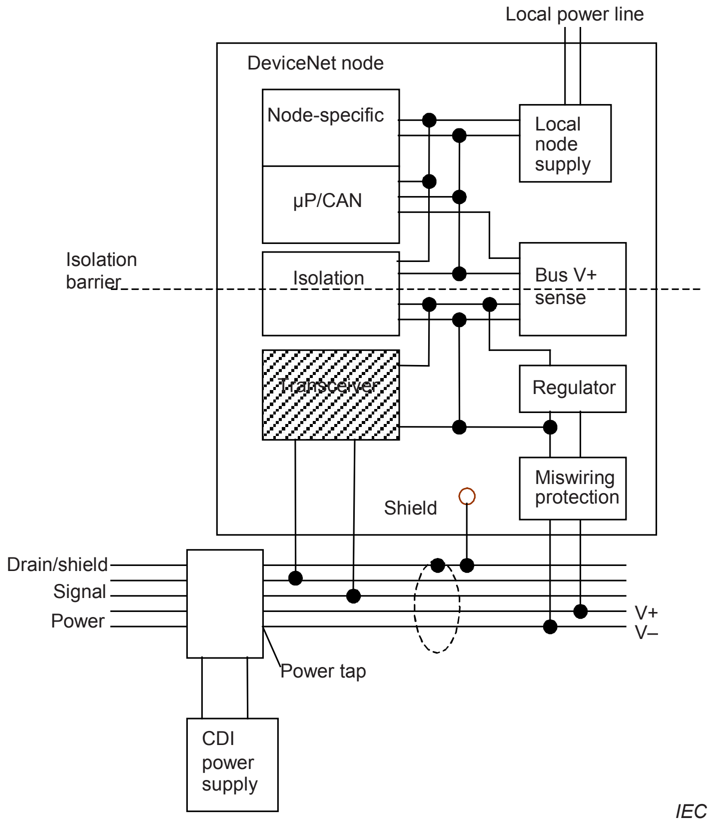

Figure 49 – Device containing an isolated physical layer

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.

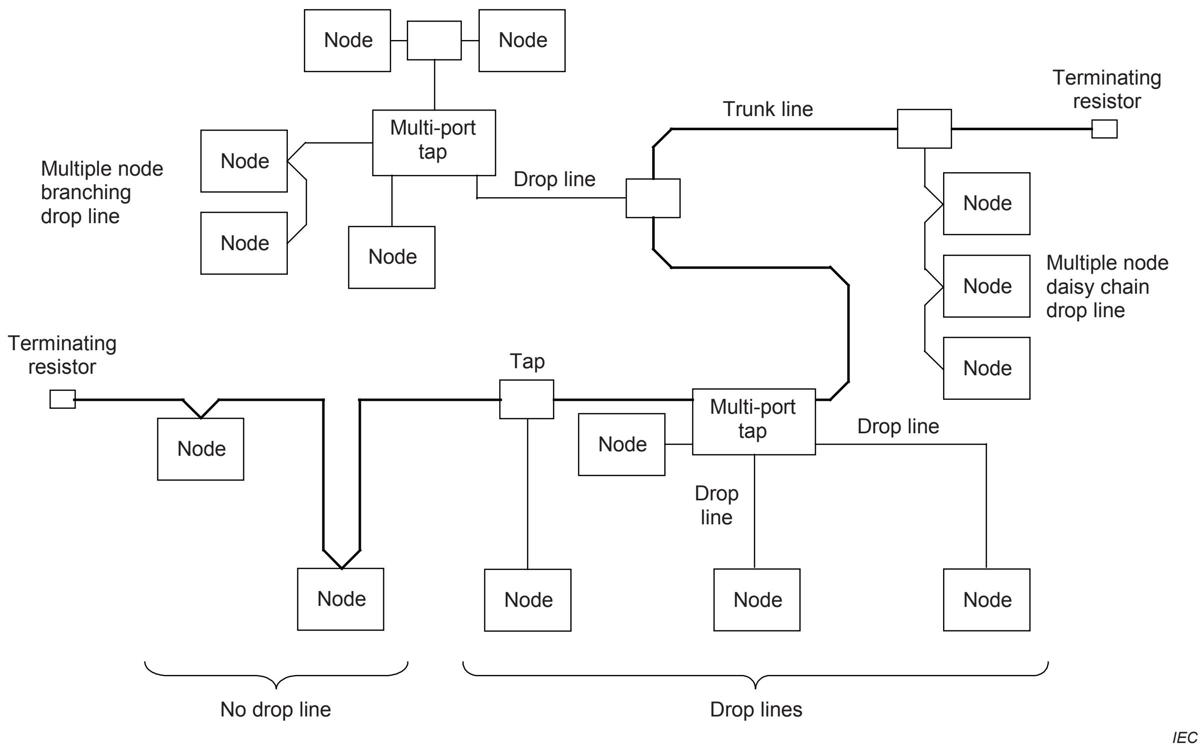

Figure 50 – DeviceNet medium topology

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 [AC1) 3 A (AC1]. 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 °C to +70 °C when carrying full current. This current rating shall be derated linearly to zero at 80 °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 °C to +70 °C when carrying full current. This current rating shall be derated linearly to zero at 80 °C.

7.1.2.6 Other CDI components

All other components of a DeviceNet CDI shall operate between the ambient temperatures of −25 °C to +70 °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 °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 °C;
– temperature: −40 °C to +85 °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 Ω ± 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	<#> Ω/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	<#> Ω/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 ±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

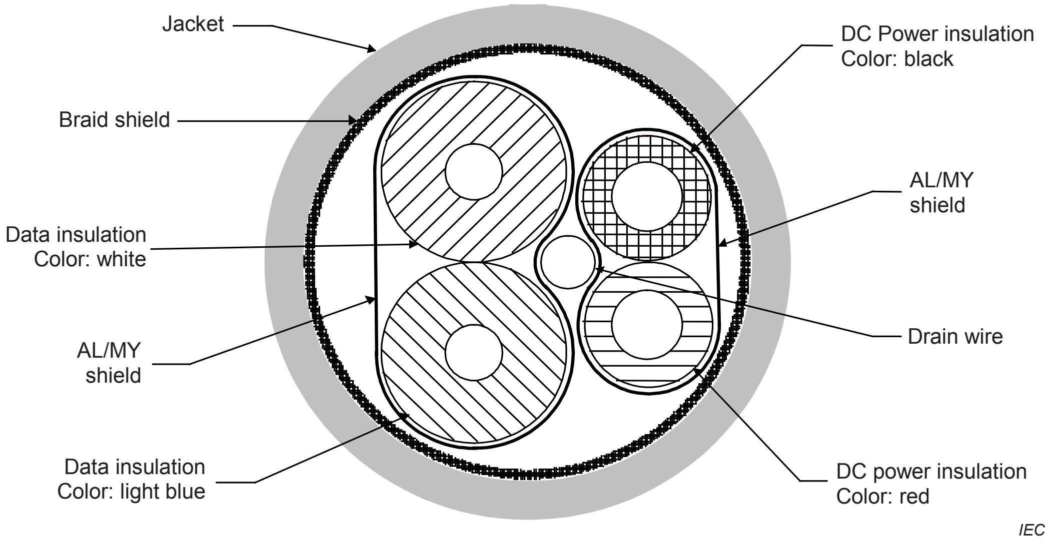

Figure 51 – Thick cable: physical configuration

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

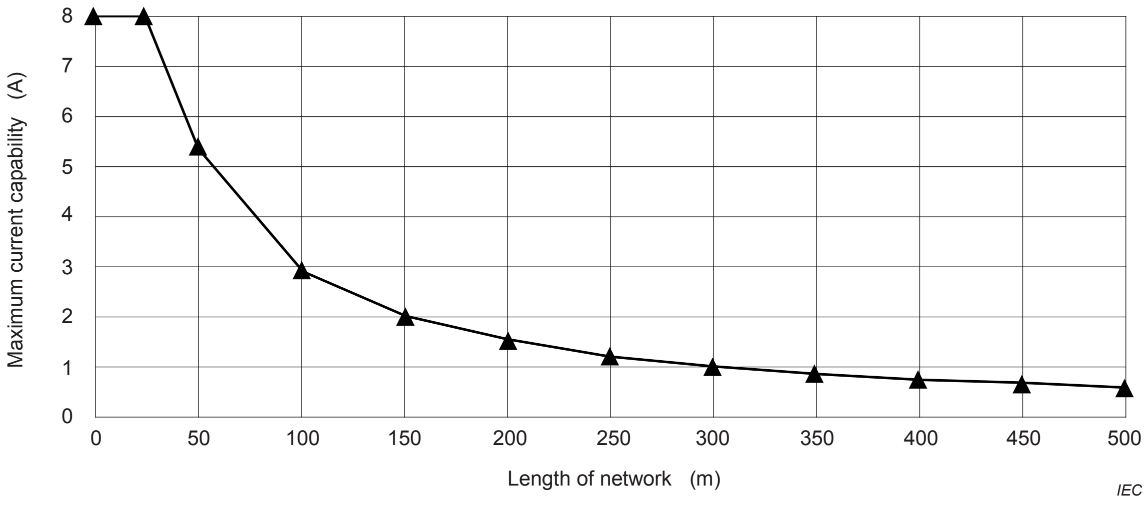

Figure 52 – Thick cable: current available on the DeviceNet power bus

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

I =4,65 V / ((cable DCR × length of network) + (contact DCR × 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 Ω ± 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

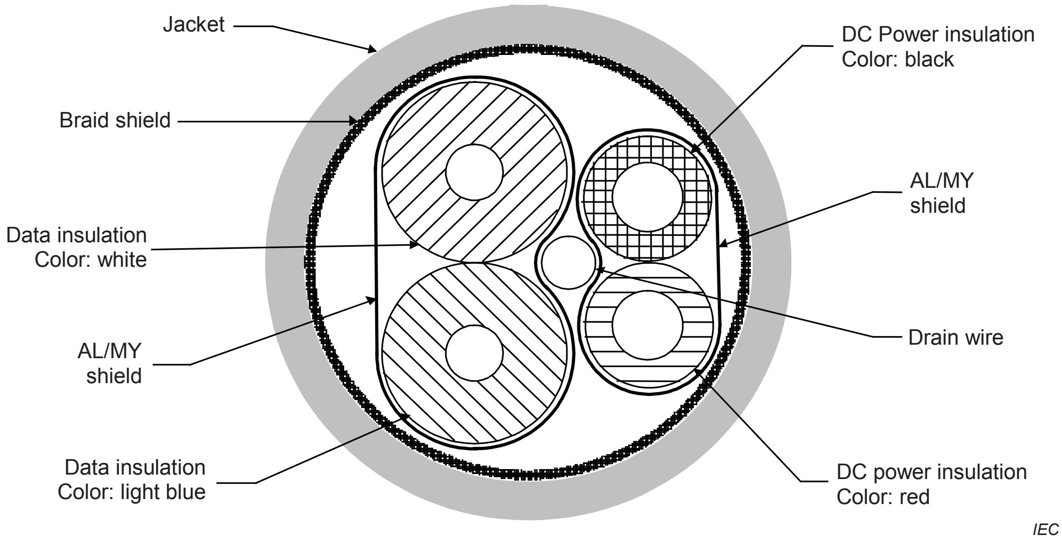

Figure 53 – Thin cable: physical configuration

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

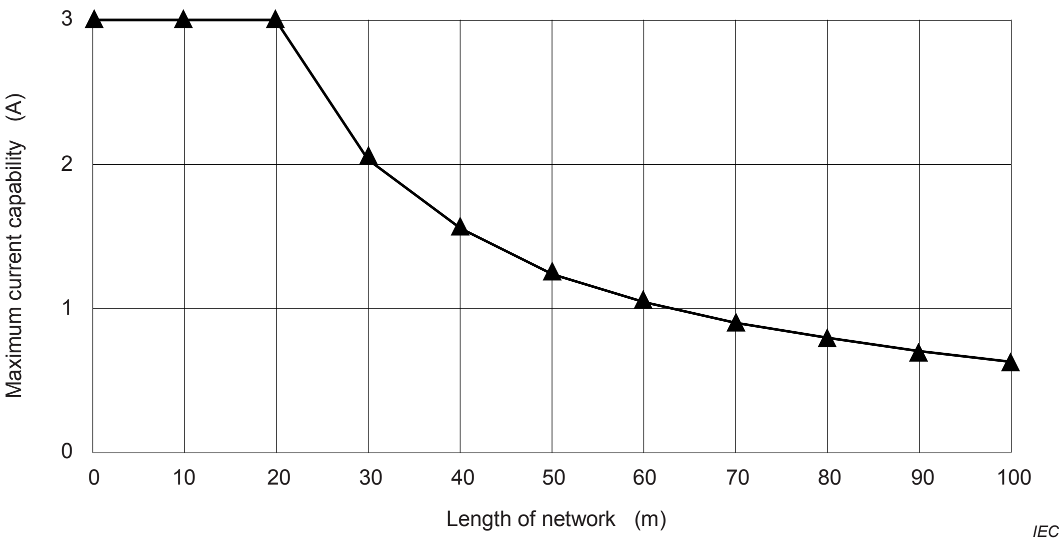

Figure 54 – Thin cable: current available on the DeviceNet power bus

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

I = 4,65 V / ((cable DCR × length of network) + (contact DCR × 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 Ω ± 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

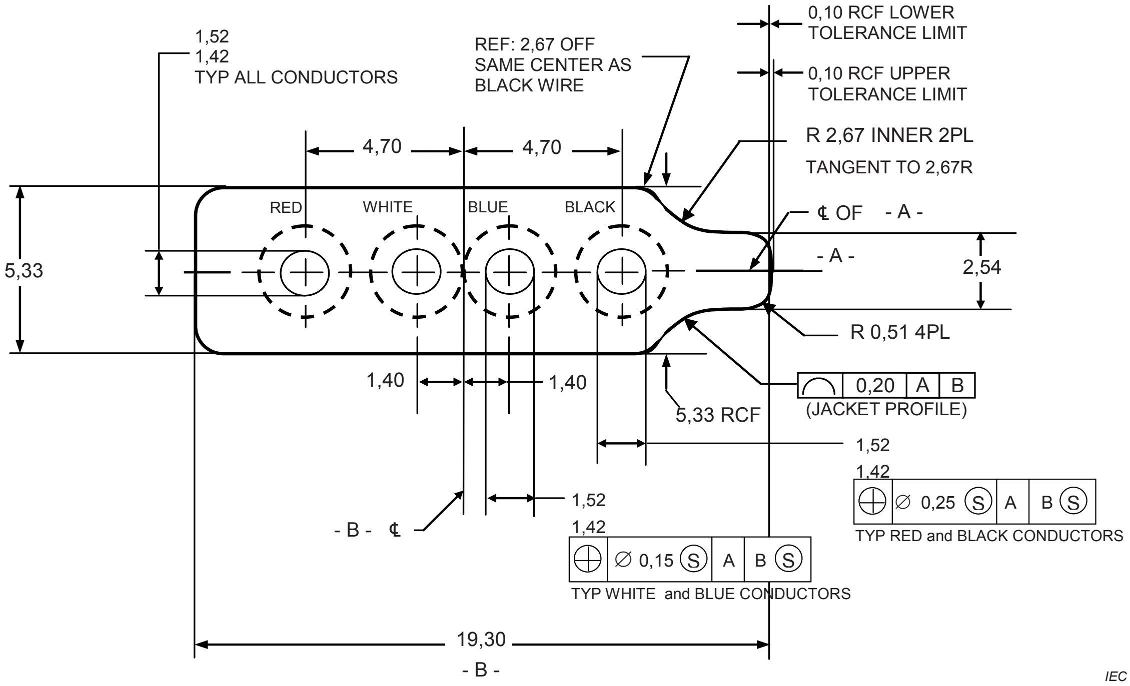

Figure 55 – Flat cable: physical configuration

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

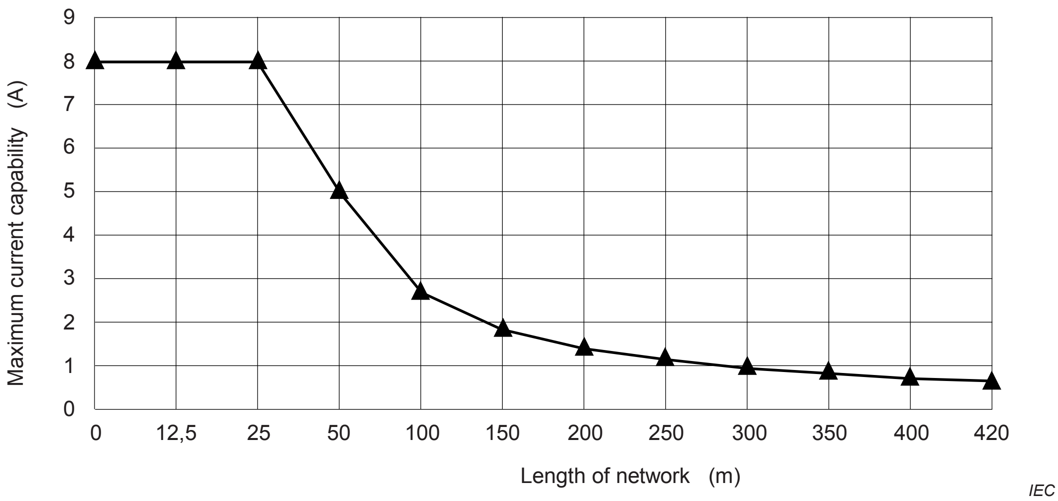

Figure 56 – Flat cable: current available on the DeviceNet power bus

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

I = 4,65 V / ((cable DCR × length of network) + (contact DCR × number of contacts))

where

– I is the allowable trunk line current;
– cable DCR = 0,016 1 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 mΩ and less than 5 mΩ 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

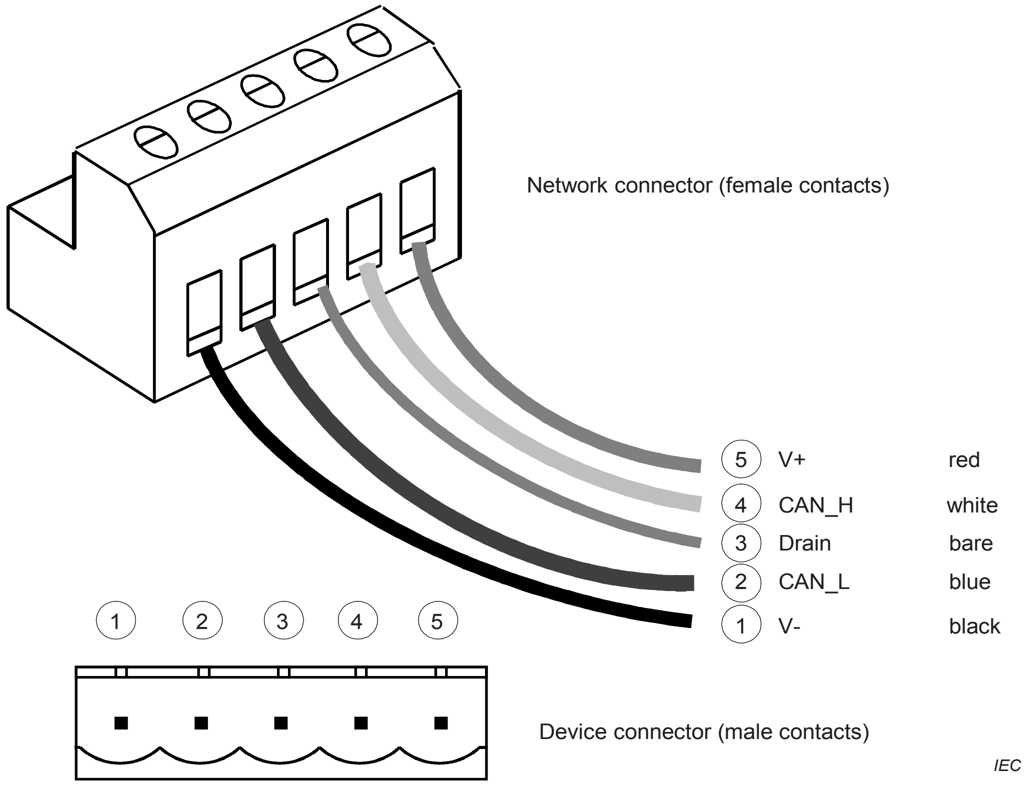

Figure 57 – Open connector pinout

Dimensions in millimetres

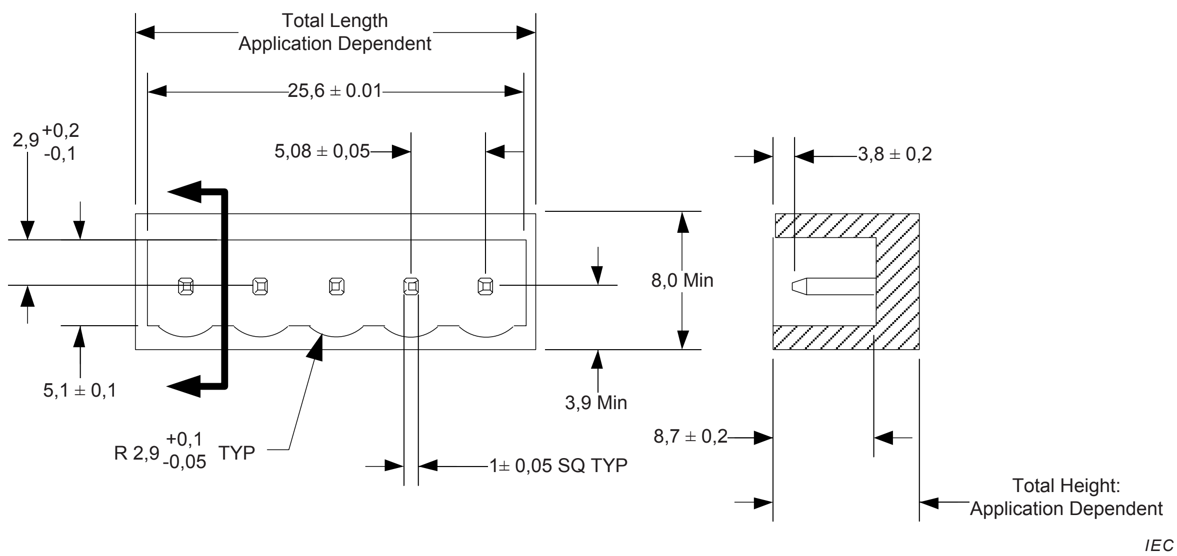

Figure 58 – Open connector geometry

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

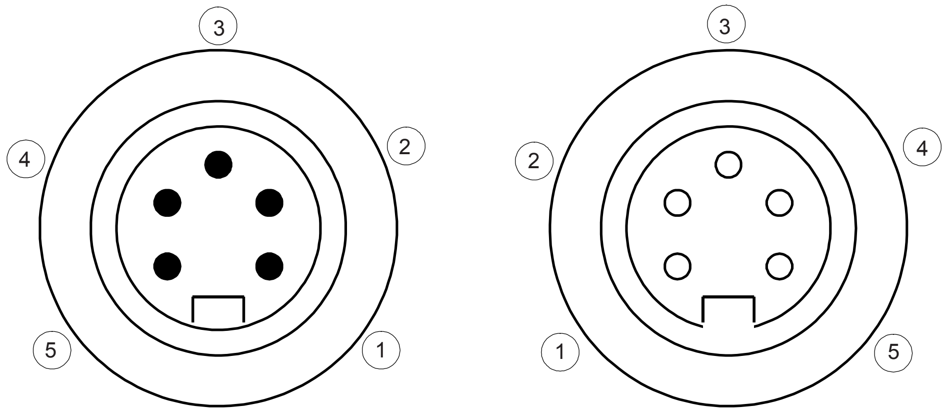

Figure 59 – Mini connector pinout

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

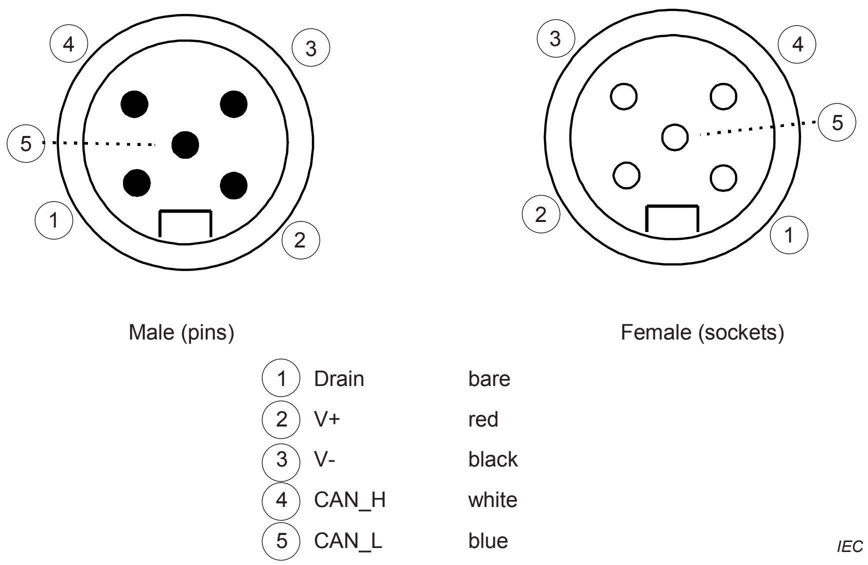

Figure 60 – Micro connector pinout

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 ¾ 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

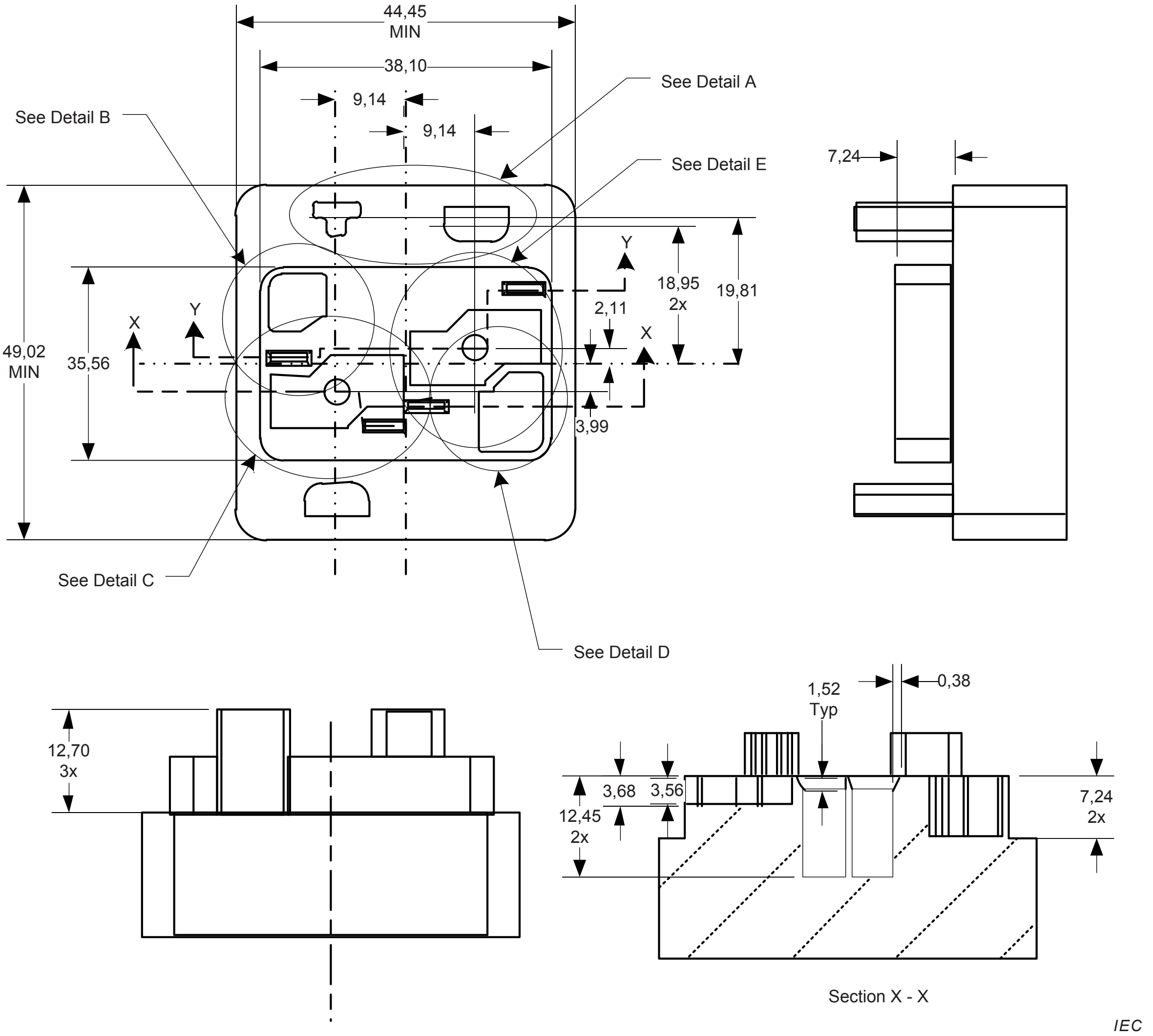

Figure 61 – Flat trunk connector layout – Part 1

Dimensions in millimetres

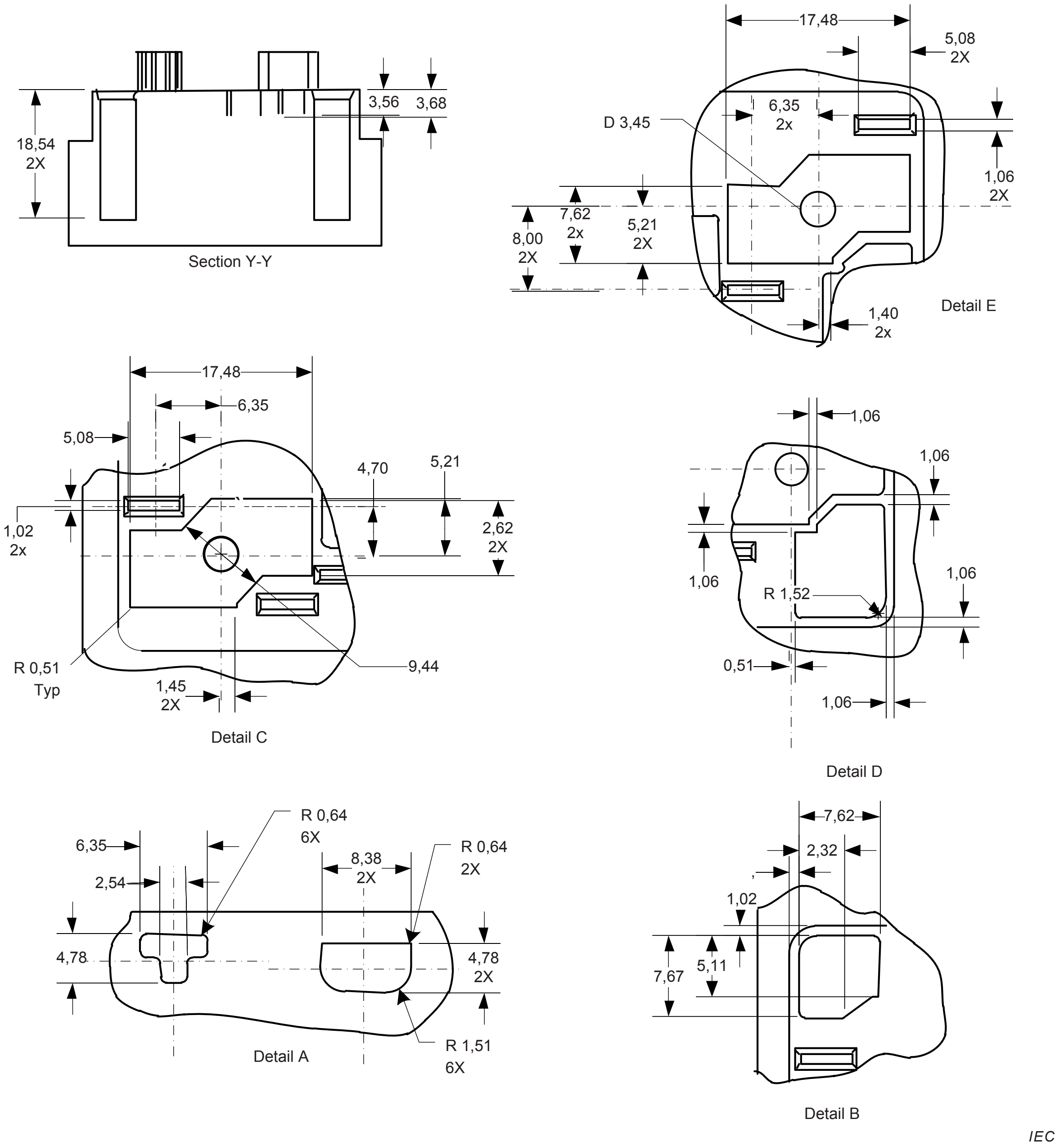

Figure 62 – Flat trunk connector layout – Part 2

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: < 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 μF 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	> 1 error bit set / 10 transfers (frames)	
- Fast transient		> 1 error bit set / 10 transfers (frames)
- ESD		Lockup
- Surge		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.

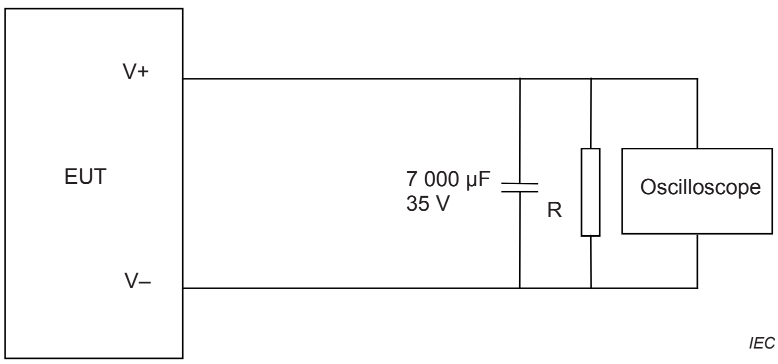

Figure 63 – Power supply rise time test circuit

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.

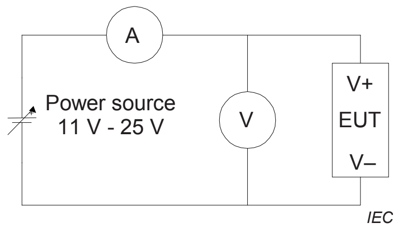

Figure 64 – Current consumption test circuit

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.

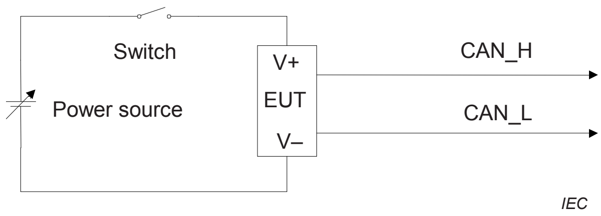

Figure 65 – Power ON test circuit

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.

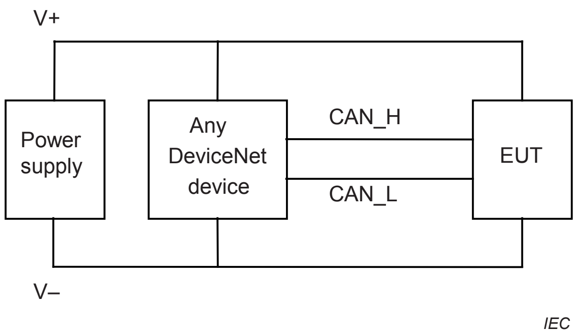

Figure 66 – Test circuit for reversal of V+ and V− and also V− interruption

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.

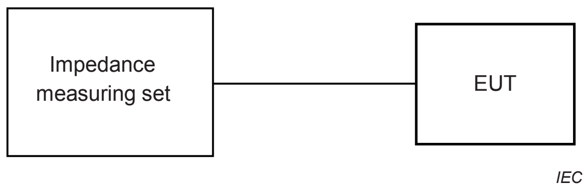

Figure 67 – Differential impedance test circuit

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 kHz ± 1 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.

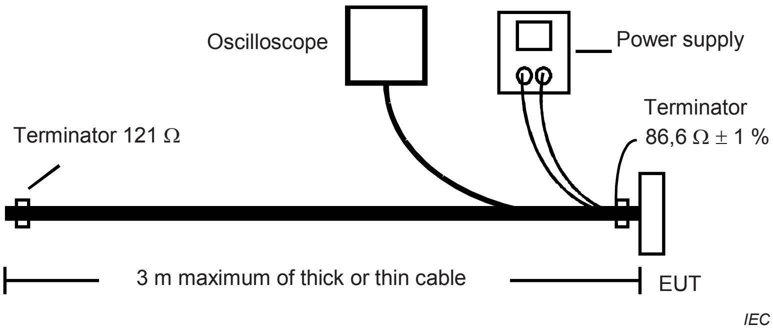

Figure 68 – Transmit level test setup

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.

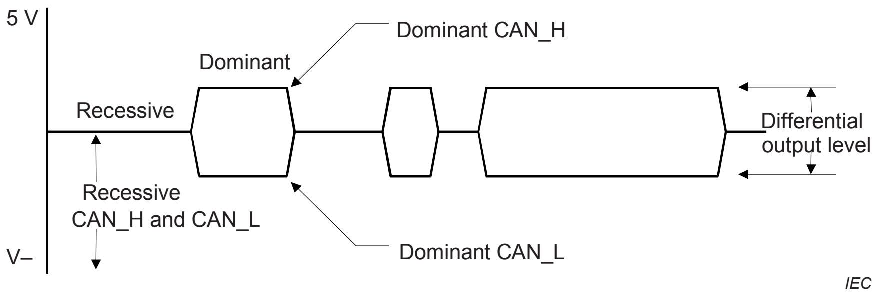

Figure 69 – Transmit levels

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.

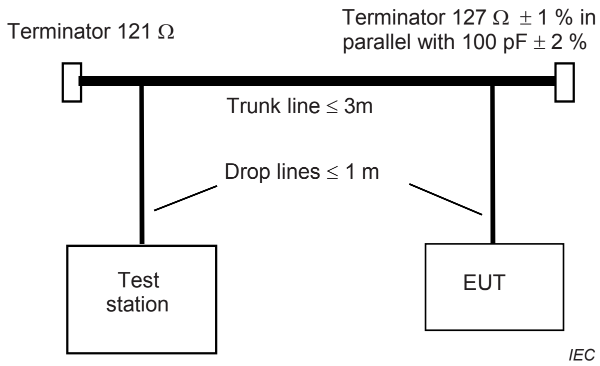

Figure 70 – Timing test setup

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).

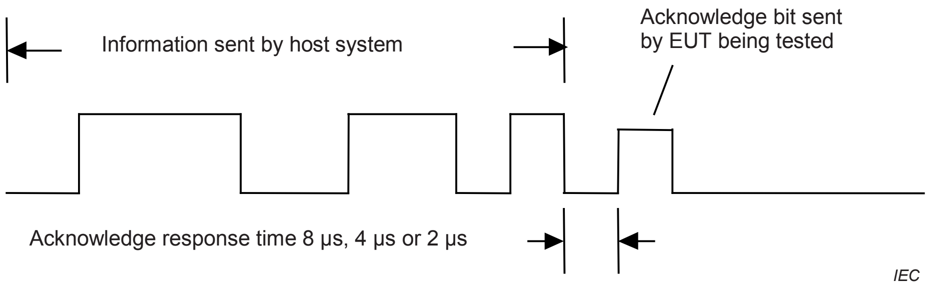

Figure 71 – Timing

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.

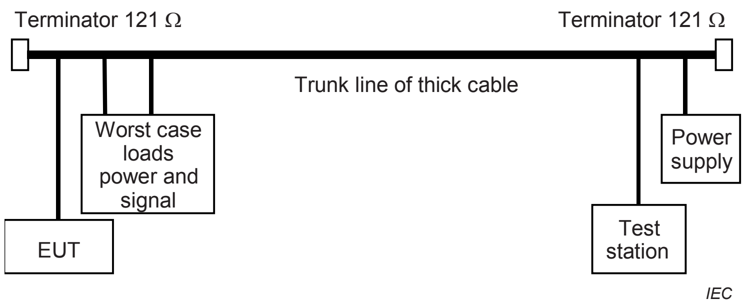

Figure 72 – CDI test configuration

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 °C ± 5 °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 be 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.

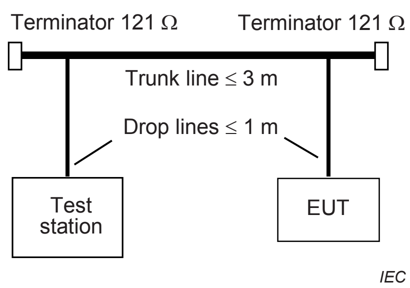

Figure 73 – Test circuit for logical tests

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.

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*

This page deliberately left blank

British Standards Institution (BSI)

BSI is the national body responsible for preparing British Standards and other standards-related publications, information and services.

BSI is incorporated by Royal Charter. British Standards and other standardization products are published by BSI Standards Limited.

About us

We bring together business, industry, government, consumers, innovators and others to shape their combined experience and expertise into standards -based solutions.

The knowledge embodied in our standards has been carefully assembled in a dependable format and refined through our open consultation process. Organizations of all sizes and across all sectors choose standards to help them achieve their goals.

Information on standards

We can provide you with the knowledge that your organization needs to succeed. Find out more about British Standards by visiting our website at bsigroup.com/standards or contacting our Customer Services team or Knowledge Centre.

Buying standards

You can buy and download PDF versions of BSI publications, including British and adopted European and international standards, through our website at bsigroup.com/shop, where hard copies can also be purchased.

If you need international and foreign standards from other Standards Development Organizations, hard copies can be ordered from our Customer Services team.

Subscriptions

Our range of subscription services are designed to make using standards easier for you. For further information on our subscription products go to bsigroup.com/subscriptions.

With **British Standards Online (BSOL)** you'll have instant access to over 55,000 British and adopted European and international standards from your desktop. It's available 24/7 and is refreshed daily so you'll always be up to date.

You can keep in touch with standards developments and receive substantial discounts on the purchase price of standards, both in single copy and subscription format, by becoming a **BSI Subscribing Member**.

PLUS is an updating service exclusive to BSI Subscribing Members. You will automatically receive the latest hard copy of your standards when they're revised or replaced.

To find out more about becoming a BSI Subscribing Member and the benefits of membership, please visit bsigroup.com/shop.

With a **Multi-User Network Licence (MUNL)** you are able to host standards publications on your intranet. Licences can cover as few or as many users as you wish. With updates supplied as soon as they're available, you can be sure your documentation is current. For further information, email bsmusales@bsigroup.com.

Revisions

Our British Standards and other publications are updated by amendment or revision.

We continually improve the quality of our products and services to benefit your business. If you find an inaccuracy or ambiguity within a British Standard or other BSI publication please inform the Knowledge Centre.

Copyright

Useful Contacts:

Customer Services
Tel: +44 845 086 9001
Email (orders): orders@bsigroup.com
Email (enquiries): cservices@bsigroup.com

Subscriptions
Tel: +44 845 086 9001
Email: subscriptions@bsigroup.com

Knowledge Centre
Tel: +44 20 8996 7004
Email: knowledgecentre@bsigroup.com

Copyright & Licensing
Tel: +44 20 8996 7070
Email: copyright@bsigroup.com

BSI Group Headquarters

389 Chiswick High Road London W4 4AL UK

bsi.

...making excellence a habit.™