

INTERNATIONAL STANDARD

NORME INTERNATIONALE



**Digital addressable lighting interface –
Part 103: General requirements – Control devices**

**Interface d'éclairage adressable numérique –
Partie 103: Exigences générales – Dispositifs de commande**

IECNORM.COM : Click to view the full PDF of IEC 62386-103:2022



THIS PUBLICATION IS COPYRIGHT PROTECTED

Copyright © 2022 IEC, Geneva, Switzerland

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

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

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

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

About the IEC

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

About IEC publications

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

IEC publications search - webstore.iec.ch/advsearchform

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 It also gives information on projects, replaced and withdrawn publications.

IEC Just Published - webstore.iec.ch/justpublished

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

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

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

IEC Products & Services Portal - products.iec.ch

Discover our powerful search engine and read freely all the publications previews. With a subscription you will always have access to up to date content tailored to your needs.

Electropedia - www.electropedia.org

The world's leading online dictionary on electrotechnology, containing more than 22 300 terminological entries in English and French, with equivalent terms in 19 additional languages. Also known as the International Electrotechnical Vocabulary (IEV) online.

A propos de l'IEC

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

A propos des publications IEC

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

Recherche de publications IEC -

webstore.iec.ch/advsearchform

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

IEC Just Published - webstore.iec.ch/justpublished

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

Service Clients - webstore.iec.ch/csc

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

IEC Products & Services Portal - products.iec.ch

Découvrez notre puissant moteur de recherche et consultez gratuitement tous les aperçus des publications. Avec un abonnement, vous aurez toujours accès à un contenu à jour adapté à vos besoins.

Electropedia - www.electropedia.org

Le premier dictionnaire d'électrotechnologie en ligne au monde, avec plus de 22 300 articles terminologiques en anglais et en français, ainsi que les termes équivalents dans 19 langues additionnelles. Egalement appelé Vocabulaire Electrotechnique International (IEV) en ligne.

INTERNATIONAL STANDARD

NORME INTERNATIONALE



**Digital addressable lighting interface –
Part 103: General requirements – Control devices**

**Interface d'éclairage adressable numérique –
Partie 103: Exigences générales – Dispositifs de commande**

INTERNATIONAL
ELECTROTECHNICAL
COMMISSION

COMMISSION
ELECTROTECHNIQUE
INTERNATIONALE

ICS 29.140.50; 29.140.99

ISBN 978-2-8322-5966-5

**Warning! Make sure that you obtained this publication from an authorized distributor.
Attention! Veuillez vous assurer que vous avez obtenu cette publication via un distributeur agréé.**

CONTENTS

FOREWORD	7
INTRODUCTION	9
1 Scope	11
2 Normative references	11
3 Terms and definitions	11
4 General	14
4.1 General	14
4.2 Version number	14
5 Electrical specification	15
6 Bus power supply	15
7 Transmission protocol structure	15
7.1 General	15
7.2 24-bit forward frame encoding	15
7.2.1 Frame format for instructions and queries	15
7.2.2 Frame format for event messages	17
8 Timing	18
9 Method of operation	18
9.1 General	18
9.2 Device features	18
9.3 Application controller	18
9.3.1 General	18
9.3.2 Single-master application controller	19
9.3.3 Multi-master application controller	19
9.4 Input device	20
9.5 Instances of input devices	20
9.5.1 General	20
9.5.2 Instance number	20
9.5.3 Instance type	20
9.5.4 Instance features	20
9.5.5 Instance groups	21
9.6 Commands excluding event messages	21
9.6.1 General	21
9.6.2 Device commands	22
9.6.3 Instance commands	22
9.6.4 Feature commands	22
9.7 Event messages	23
9.7.1 Response to event messages	23
9.7.2 Device power cycle event	23
9.7.3 Input notification event	23
9.7.4 Event message filter	24
9.8 Input signal, measured value and “ <i>inputValue</i> ”	24
9.8.1 General	24
9.8.2 Input resolution	24
9.8.3 Getting the input value	25
9.8.4 Notification of changes	26

9.9	System failure	26
9.10	Operating a control device	26
9.10.1	Enable/disable the application controller	26
9.10.2	Application controller always active	26
9.10.3	Enable/disable event messages	27
9.10.4	Quiescent mode	27
9.10.5	Modes of operation	27
9.11	Memory banks	28
9.11.1	General	28
9.11.2	Memory map	29
9.11.3	Selecting a memory bank location	30
9.11.4	Protectable memory locations	30
9.11.5	Memory bank reading	30
9.11.6	Memory bank writing	32
9.11.7	Memory bank 0	33
9.11.8	Memory bank 1 (optional)	36
9.11.9	Manufacturer-specific memory banks	37
9.11.10	Reserved memory banks	37
9.12	Reset	38
9.12.1	Reset operation	38
9.12.2	Reset memory bank operation	38
9.13	Power on behaviour	38
9.13.1	Power on	38
9.13.2	Power cycle notification	39
9.14	Priority use	39
9.14.1	General	39
9.14.2	Priority of input notifications	39
9.15	Assigning short addresses	40
9.15.1	General	40
9.15.2	Random address allocation	40
9.15.3	Identification of a device	40
9.16	Exception handling	41
9.17	Device capabilities and status information	41
9.17.1	Device capabilities	41
9.17.2	Device status	41
9.17.3	Instance status	42
9.18	Non-volatile memory	42
9.19	Instance types and configuration	42
9.20	Current bus unit configuration	43
10	Declaration of variables	43
11	Definition of commands	45
11.1	General	45
11.2	Overview sheets	45
11.3	Event messages	52
11.3.1	INPUT NOTIFICATION (<i>device/instance, event</i>)	52
11.3.2	POWER NOTIFICATION (<i>device</i>)	52
11.4	Device control instructions	52
11.4.1	General	52
11.4.2	IDENTIFY DEVICE	52

11.4.3	RESET POWER CYCLE SEEN	53
11.5	Device configuration instructions.....	53
11.5.1	General	53
11.5.2	RESET	53
11.5.3	RESET MEMORY BANK (<i>DTR0</i>)	54
11.5.4	SET SHORT ADDRESS (<i>DTR0</i>)	54
11.5.5	ENABLE WRITE MEMORY	54
11.5.6	ENABLE APPLICATION CONTROLLER	54
11.5.7	DISABLE APPLICATION CONTROLLER	54
11.5.8	SET OPERATING MODE (<i>DTR0</i>)	54
11.5.9	ADD TO DEVICE GROUPS 0-15 (<i>DTR2:DTR1</i>)	55
11.5.10	ADD TO DEVICE GROUPS 16-31 (<i>DTR2:DTR1</i>)	55
11.5.11	REMOVE FROM DEVICE GROUPS 0-15 (<i>DTR2:DTR1</i>).....	55
11.5.12	REMOVE FROM DEVICE GROUPS 16-31 (<i>DTR2:DTR1</i>).....	55
11.5.13	START QUIESCENT MODE	55
11.5.14	STOP QUIESCENT MODE	55
11.5.15	ENABLE POWER CYCLE NOTIFICATION	55
11.5.16	DISABLE POWER CYCLE NOTIFICATION	55
11.5.17	SET EVENT PRIORITY (<i>DTR0</i>).....	55
11.6	Device queries	56
11.6.1	General	56
11.6.2	QUERY DEVICE CAPABILITIES	56
11.6.3	QUERY DEVICE STATUS	56
11.6.4	QUERY APPLICATION CONTROLLER ERROR	56
11.6.5	QUERY INPUT DEVICE ERROR	56
11.6.6	QUERY MISSING SHORT ADDRESS.....	57
11.6.7	QUERY VERSION NUMBER.....	57
11.6.8	QUERY CONTENT <i>DTR0</i>	57
11.6.9	QUERY NUMBER OF INSTANCES.....	57
11.6.10	QUERY CONTENT <i>DTR1</i>	57
11.6.11	QUERY CONTENT <i>DTR2</i>	57
11.6.12	QUERY RANDOM ADDRESS (H)	57
11.6.13	QUERY RANDOM ADDRESS (M).....	57
11.6.14	QUERY RANDOM ADDRESS (L).....	57
11.6.15	READ MEMORY LOCATION (<i>DTR1</i> , <i>DTR0</i>).....	57
11.6.16	QUERY APPLICATION CONTROLLER ENABLED	58
11.6.17	QUERY OPERATING MODE	58
11.6.18	QUERY MANUFACTURER SPECIFIC MODE	58
11.6.19	QUERY QUIESCENT MODE.....	58
11.6.20	QUERY DEVICE GROUPS 0-7	58
11.6.21	QUERY DEVICE GROUPS 8-15	58
11.6.22	QUERY DEVICE GROUPS 16-23	58
11.6.23	QUERY DEVICE GROUPS 24-31	58
11.6.24	QUERY POWER CYCLE NOTIFICATION	58
11.6.25	QUERY EXTENDED VERSION NUMBER(<i>DTR0</i>)	58
11.6.26	QUERY RESET STATE	59
11.6.27	QUERY APPLICATION CONTROLLER ALWAYS ACTIVE	59
11.6.28	QUERY FEATURE TYPE.....	59
11.6.29	QUERY NEXT FEATURE TYPE.....	59

11.6.30	QUERY EVENT PRIORITY	59
11.7	Instance control instructions	59
11.8	Instance configuration instructions	59
11.8.1	General	59
11.8.2	ENABLE INSTANCE	60
11.8.3	DISABLE INSTANCE	60
11.8.4	SET PRIMARY INSTANCE GROUP (<i>DTR0</i>)	60
11.8.5	SET INSTANCE GROUP 1 (<i>DTR0</i>)	60
11.8.6	SET INSTANCE GROUP 2 (<i>DTR0</i>)	60
11.8.7	SET EVENT SCHEME (<i>DTR0</i>)	60
11.8.8	SET EVENT PRIORITY (<i>DTR0</i>)	61
11.8.9	SET EVENT FILTER (<i>DTR2:DTR1:DTR0</i>)	61
11.8.10	SET INSTANCE TYPE (<i>DTR0</i>)	61
11.8.11	SET INSTANCE CONFIGURATION (<i>DTR0, DTR2:DTR1</i>)	61
11.9	Instance queries	62
11.9.1	General	62
11.9.2	QUERY INSTANCE TYPE	62
11.9.3	QUERY RESOLUTION	62
11.9.4	QUERY INSTANCE ERROR	62
11.9.5	QUERY INSTANCE STATUS	62
11.9.6	QUERY INSTANCE ENABLED	62
11.9.7	QUERY PRIMARY INSTANCE GROUP	62
11.9.8	QUERY INSTANCE GROUP 1	63
11.9.9	QUERY INSTANCE GROUP 2	63
11.9.10	QUERY EVENT SCHEME	63
11.9.11	QUERY INPUT VALUE	63
11.9.12	QUERY INPUT VALUE LATCH	63
11.9.13	QUERY EVENT PRIORITY	63
11.9.14	QUERY FEATURE TYPE	63
11.9.15	QUERY NEXT FEATURE TYPE	64
11.9.16	QUERY EVENT FILTER 0-7	64
11.9.17	QUERY EVENT FILTER 8-15	64
11.9.18	QUERY EVENT FILTER 16-23	64
11.9.19	QUERY INSTANCE CONFIGURATION (<i>DTR0</i>)	64
11.9.20	QUERY AVAILABLE INSTANCE TYPES	65
11.10	Special commands	65
11.10.1	General	65
11.10.2	TERMINATE	65
11.10.3	INITIALISE (<i>device</i>)	65
11.10.4	RANDOMISE	65
11.10.5	COMPARE	66
11.10.6	WITHDRAW	66
11.10.7	SEARCHADDRH (<i>data</i>)	66
11.10.8	SEARCHADDRM (<i>data</i>)	66
11.10.9	SEARCHADDRRL (<i>data</i>)	67
11.10.10	PROGRAM SHORT ADDRESS (<i>data</i>)	67
11.10.11	VERIFY SHORT ADDRESS (<i>data</i>)	67
11.10.12	QUERY SHORT ADDRESS	67
11.10.13	WRITE MEMORY LOCATION (<i>DTR1, DTR0, data</i>)	67

11.10.14	WRITE MEMORY LOCATION – NO REPLY (<i>DTR1</i> , <i>DTR0</i> , <i>data</i>)	68
11.10.15	DTR0 (<i>data</i>)	68
11.10.16	DTR1 (<i>data</i>)	68
11.10.17	DTR2 (<i>data</i>)	68
11.10.18	DIRECT WRITE MEMORY (<i>DTR1</i> , <i>offset</i> , <i>data</i>)	68
11.10.19	DTR1:DTR0 (<i>data1</i> , <i>data0</i>).....	68
11.10.20	DTR2:DTR1 (<i>data2</i> , <i>data1</i>).....	69
11.10.21	SEND TESTFRAME (<i>data</i>)	69
Bibliography.....		70
Figure 1 – IEC 62386 graphical overview		9
Table 1 – 24-bit command frame encoding.....		16
Table 2 – Instance byte in a command frame		16
Table 3 – 24-bit event message frame encoding		17
Table 4 – Instance types		20
Table 5 – Feature types		21
Table 6 – Instance group variables		21
Table 7 – Device address information in power cycle event		23
Table 8 – Event addressing schemes.....		23
Table 9 – Measured value ($\approx 50\%$) versus resolution and “ <i>inputValue</i> ”.....		25
Table 10 – Example of querying sequence to read a 4-byte input value		25
Table 11 – Memory types.....		29
Table 12 – Basic memory map of memory banks		29
Table 13 – Memory map of memory bank 0.....		34
Table 14 – Memory map of memory bank 1.....		36
Table 15 – Control device capabilities.....		41
Table 16 – Control device status.....		42
Table 17 – Instance status.....		42
Table 18 – Current bus unit configuration		43
Table 19 – Declaration of device variables.....		44
Table 20 – Declaration of instance variables.....		45
Table 21 – Instance event messages		45
Table 22 – Device event messages.....		46
Table 23 – Standard commands.....		47
Table 24 – Special commands (implemented by both application controller and input device).....		51
Table 25 – Device addressing with “INITIALISE (<i>device</i>)”		65

INTERNATIONAL ELECTROTECHNICAL COMMISSION

DIGITAL ADDRESSABLE LIGHTING INTERFACE –**Part 103: General requirements –
Control devices****FOREWORD**

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

IEC 62386-103 has been prepared by IEC technical committee 34: Lighting. It is an International Standard.

This second edition cancels and replaces the first edition published in 2014 and Amendment 1:2018. This edition constitutes a technical revision.

This edition includes the following significant technical changes with respect to the previous edition:

- a) the scope has been updated;
- b) quiescent mode has been updated;
- c) non-volatile memory (NVM) save time has been added, and SAVE PERSISTENT VARIABLES command removed;
- d) memory bank 0 has been modified, and common memory bank requirements have been added;

- e) IDENTIFY DEVICE has been updated;
- f) version number has been changed;
- g) bus unit configuration has been added; and
- h) instance types and configuration have been added.

The text of this International Standard is based on the following documents:

Draft	Report on voting
34/946/FDIS	34/990/RVD

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

The language used for the development of this International Standard is English.

This document was drafted in accordance with ISO/IEC Directives, Part 2, and developed in accordance with ISO/IEC Directives, Part 1 and ISO/IEC Directives, IEC Supplement, available at www.iec.ch/members_experts/refdocs. The main document types developed by IEC are described in greater detail at www.iec.ch/standardsdev/publications.

This Part 103 of IEC 62386 is intended to be used in conjunction with Part 101, which contains general requirements for the relevant product type (system), and with the appropriate Parts 3xx (particular requirements for control devices) containing clauses to supplement or modify the corresponding clauses in Part 101 and Part 103 in order to provide the relevant requirements for each type of product.

A list of all parts in the IEC 62386 series, published under the general title *Digital addressable lighting interface*, can be found on the IEC website.

The committee has decided that the contents of this document will remain unchanged until the stability date indicated on the IEC website under webstore.iec.ch in the data related to the specific document. At this date, the document will be

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

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

INTRODUCTION

IEC 62386 contains several parts, referred to as series. The IEC 62386 series specifies a bus system for control by digital signals of electronic lighting equipment. The IEC 62386-1xx series includes the basic specifications. Part 101 contains general requirements for system components, Part 102 extends this information with general requirements for control gear and Part 103 extends it further with general requirements for control devices. Part 104 and Part 105 can be applied to control gear or control devices. Part 104 gives requirements for wireless and alternative wired system components. Part 105 describes firmware transfer. Part 150 gives requirements for an auxiliary power supply which can be stand-alone, or built into control gear or control devices.

The IEC 62386-2xx series extends the general requirements for control gear with lamp specific extensions (mainly for backward compatibility with Edition 1 of IEC 62386) and with control gear specific features.

The IEC 62386-3xx series extends the general requirements for control devices with input device specific extensions describing the instance types as well as some common features that can be combined with multiple instance types.

This second edition of IEC 62386-103 is intended to be used in conjunction with IEC 62386-101 and with the various parts that make up the IEC 62386-3xx series of particular requirements for control devices, and can be used together with IEC 62386-102 and with the various parts that make up the IEC 62386-2xx series for control gear. The division into separately published parts provides for ease of future amendments and revisions. Additional requirements will be added as and when a need for them is recognised.

The setup of the standards is graphically represented in Figure 1 below.

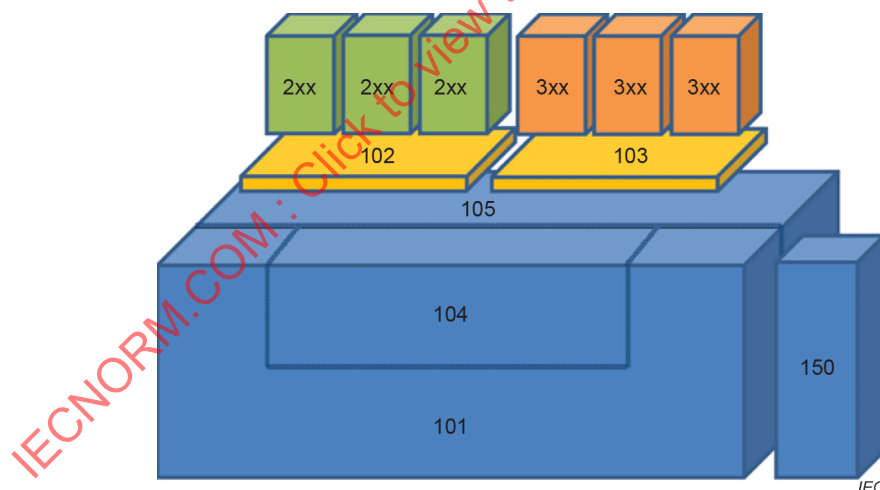


Figure 1 – IEC 62386 graphical overview

When this part of IEC 62386 refers to any of the clauses of the other parts of the IEC 62386-1xx series, the extent to which such a clause is applicable is specified. The other parts also include additional requirements, as necessary.

All numbers used in this document are decimal numbers unless otherwise noted. Hexadecimal numbers are given in the format 0xVV, where VV is the value. Binary numbers are given in the format XXXXXXXXXb or in the format XXXX XXXX, where X is 0 or 1, "x" in binary numbers means "don't care".

The following typographic expressions are used:

Variables: *variableName* or *variableName[3:0]*, giving only bits 3 to 0 of *variableName*;

Range of values: [lowest, highest];

Command: "COMMAND NAME".

IECNORM.COM : Click to view the full PDF of IEC 62386-103:2022

DIGITAL ADDRESSABLE LIGHTING INTERFACE –

Part 103: General requirements – Control devices

1 Scope

This part of IEC 62386 is applicable to control devices for control by digital signals of electronic lighting equipment.

2 Normative references

The following documents are referred to in the text in such a way that some or all of their content constitutes requirements of this document. For dated references, only the edition cited applies. For undated references, the latest edition of the referenced document (including any amendments) applies.

IEC 62386-101:2022, *Digital addressable lighting interface – Part 101: General requirements – System components*

IEC 62386-102:2022, *Digital addressable lighting interface – Part 102: General requirements – Control gear*

IEC 62386-3xx (all parts), *Digital addressable lighting interface – Part 3xx: Particular requirements for control devices*

3 Terms and definitions

For the purposes of this document, the terms and definitions given in IEC 62386-101 and the following apply.

ISO and IEC maintain terminology databases for use in standardization at the following addresses:

- IEC Electropedia: available at <http://www.electropedia.org/>
- ISO Online browsing platform: available at <http://www.iso.org/obp>

3.1

broadcast

type of address used to simultaneously address all control devices in the system

3.2

broadcast unaddressed

type of address used to simultaneously address all control devices in the system that have no short address

3.3

device command

command which addresses the control device and has a value of 0xFE in the instance byte of the command frame but is not an event message

3.4

device group

type of address used to address a group of control devices in the system at once

3.5

DTR

data transfer register

multipurpose register used to exchange data

3.6

event

instance report, characterized by its event number, of a change or a defined sequence of changes of its input value

Note 1 to entry: The event number is specific to the type of instance that sends the report.

3.7

event scheme

characterisation of the information, as provided by an instance when producing an event message, that identifies the source of the event

3.8

feature

optional extension at instance and/or device level

3.9

feature command

command which addresses one or more features of an input device or device instance and has a value different from 0xFE in the instance byte of the command frame but is not an instance command or an event message

3.10

GTIN

global trade item number

number used for the unique identification of trade items worldwide

Note 1 to entry: For further information see <http://en.wikipedia.org/wiki/GTIN>.

Note 2 to entry: The global trade item number is comprised of a GS1 or U.P.C. company prefix followed by an item reference number and a check digit. It is described in the "GS1 General Specifications" (see [1]).

3.11

input signal

physical value that an instance of an input device is designed to detect and process

Note 1 to entry: Examples for physical values are "illuminance" and "button state".

3.12

identification

temporary state used during commissioning that allows the installer to identify particular control devices

3.13

input value

encoded data, representing the input signal

Note 1 to entry: The way in which the input signal is encoded depends on the instance type.

3.14**instance command**

command which addresses one or more instances of an input device and has a value different from 0xFE in the instance byte of the command frame but is not a feature command or an event message

3.15**MASK**

value with all binary digits set to 1

Note 1 to entry: This means that an 8-bit backward frame of MASK is a value of 0xFF, and a multi-byte memory location of 24 bits containing MASK is a value of 0xFFFFF.

3.16**NO**

answer to a query where no backward frame is sent

Note 1 to entry: If a query is asked where the answer is NO, there will be no response, such that the sender of the query will conclude "no backward frame" following IEC 62386-101:2022, 8.2.5.

Note 2 to entry: The answer NO could also be triggered by a missed query.

3.17**NVM****non-volatile memory**

non-volatile read/write memory, the content of which can be changed and will not be lost due to a power cycle

3.18**NVM-RO**

NVM that cannot be written using any command

3.19**NVM-RW**

NVM that can be modified using one or more commands

3.20**opcode****operation code**

that part of a command frame that identifies the command to be executed

3.21**operating mode**

set of states identified by a number in the range [0,255], characterised by a collection of variables and memory settings, and used to select a set of functionalities to be exhibited by a control device, including its required reaction to commands

Note 1 to entry: Control devices can support more than one operating mode.

3.22**PING**

16-bit forward frame with bits [15:0] equal to 0xAD00

Note 1 to entry: As specified in IEC 62386-102, PING has no meaning for control gear.

3.23**quiescent mode**

temporary mode in which the device does not send forward frames

3.24

RAM

volatile read/write memory, the content of which can be changed and will be lost due to a power cycle

3.25

RAM-RO

RAM that cannot be written using any command

3.26

RAM-RW

RAM that can be modified using one or more commands

3.27

random address

random 24-bit number generated by the control device on request during system initialisation

3.28

reset state

state in which all NVM variables of the control device have their reset value, except those that are marked "no change" or are otherwise explicitly excluded

3.29

ROM

non-volatile read-only memory, the content of which is fixed

Note 1 to entry: In this document read-only is meant from a system perspective. A ROM variable may actually be implemented in NVM, but this document does not provide any mechanism to change its value.

3.30

search address

24-bit number used to identify an individual control device in the system during initialisation

3.31

short address

type of address used to address an individual control device in the system

3.32

TMASK

value with the least significant bit set to 0 and all other binary digits set to 1

Note 1 to entry: This means that an 8-bit backward frame of TMASK is a value of 0xFE, and a multi-byte memory location of 24 bits containing TMASK is a value of 0xFFFFFE.

3.33

YES

answer to a query where a backward frame of MASK is sent

4 General

4.1 General

The requirements of IEC 62386-101:2022, Clause 4 apply, with the restrictions, changes and additions identified below.

4.2 Version number

This subclause replaces IEC 62386-101:2022, 4.2.

The version shall be in the format "x.y", where the major version number x is in the range of 0 to 62 and the minor version number y is in the range of 0 to 2. When the version number is encoded into a byte, the major version number x shall be placed in bits 7 to 2 and the minor version number y shall be placed in bits 1 to 0.

At each amendment to an edition of IEC 62386-103 the minor version number shall be incremented by one.

At a new edition of IEC 62386-103 the major version number shall be incremented by one and the minor version number shall be set to 0.

The current version number is "*versionNumber*" as defined in Table 19.

NOTE IEC documents are generally subject to two amendments before a new edition is prepared.

5 Electrical specification

The requirements of IEC 62386-101:2022, Clause 5 apply.

6 Bus power supply

If a bus power supply is integrated into a control device, the requirements of IEC 62386-101:2022, Clause 6 apply.

7 Transmission protocol structure

7.1 General

The requirements of IEC 62386-101:2022, Clause 7 apply with the following additions.

7.2 24-bit forward frame encoding

7.2.1 Frame format for instructions and queries

7.2.1.1 General

For 7.2.1, commands shall be interpreted as instructions and queries, and exclude event messages. The 24-bit forward frame shall be encoded as shown in Table 1 and Table 2.

Table 1 – 24-bit command frame encoding

Bytes/Bits										Device addressing
Address byte								Instance byte	Opcode byte	
23	22	21	20	19	18	17	16	15...8	7...0	
0	64 short addresses						1	Device or instance or feature, see Table 2		Short addressing
1	0	32 device group addresses					1			Device group addressing
1	1	1	1	1	1	0	1			Broadcast unaddressed
1	1	1	1	1	1	1	1			Broadcast
1	1	0	16 special command spaces				1	Command specific		Special command
1	1	1	0	x	x	x	1	Reserved		Reserved
1	1	1	1	0	x	x	1			
1	1	1	1	1	0	x	1			

Table 2 – Instance byte in a command frame

Instance byte								Addressing
15	14	13	12	11	10	09	08	
0	0	0	32 Instance numbers					Instance number
1	0	0	32 Instance groups					Instance group
1	1	0	32 Instance types					Instance type
0	0	1	32 Instance numbers					Feature on instance number level
1	0	1	32 Instance groups					Feature on instance group level
0	1	1	32 Instance types					Feature on instance type level
1	1	1	1	1	0	0	1	Feature broadcast
1	1	1	1	1	1	0	1	Feature on instance broadcast level
1	1	1	1	1	1	1	1	Instance broadcast
1	1	1	1	1	1	0	0	Feature on device level
1	1	1	1	1	1	1	0	Device
0	1	0	x	x	x	x	x	Reserved
1	1	1	0	x	x	x	x	
1	1	1	1	0	x	x	x	
1	1	1	1	1	0	1	x	
1	1	1	1	1	0	0	0	

7.2.1.2 Address byte

The address byte provides

- the method of device addressing used by the transmitter;
- the indication that a command, not an event message, is being transmitted: bit 16 is set for commands;
- 16 special command spaces;
- reserved device addresses. Reserved addresses shall not be used by the transmitter.

7.2.1.3 Instance byte

The instance byte provides

- for standard commands, the indication of whether a device command, feature command or an instance command is being transmitted;
- for standard instance commands, the method of instance addressing used by the transmitter;
- command-specific information for special commands;
- for standard commands, reserved instance addresses. Reserved instance addresses shall not be used by the transmitter;
- for standard feature commands, the feature that is being addressed;
- reserved information for reserved commands.

7.2.1.4 Opcode byte

The opcode byte provides

- for standard commands, the opcode;
- command-specific information for special commands;
- reserved information for reserved commands.

7.2.2 Frame format for event messages

7.2.2.1 General

For event messages, the 24-bit forward frame shall be encoded as shown in Table 3.

Table 3 – 24-bit event message frame encoding

Bits														Event scheme ^a / Source				
Event source information													Event information					
23	22	21	20	19	18	17	16	15	14	13	12	11	10			9...0		
0	64 short addresses						0	0	32 instance types						Event	1	Device	
0	64 short addresses						0	1	32 instance numbers							2	Device and instance	
1	0	32 device groups						0	0	32 instance types						3	Device group	
1	0	32 instance types						0	1	32 instance numbers						0	Instance	
1	1	32 instance groups						0	0	32 instance types						4	Instance group	
1	1	0	x	x	x	x	0	1	x	x	x	x	x	Reserved	Reserved			
1	1	1	0	x	x	x	0	1	x	x	x	x	x					
1	1	1	1	0	x	x	0	1	x	x	x	x	x					
1	1	1	1	1	0	x	0	1	x	x	x	x	x					
1	1	1	1	!	1	0	0	1	x	x	x	x	x					
1	1	1	!	1	1	1	0	1	0	x	x	x	x					
1	1	1	1	1	1	1	0	1	1	0	x	x	x					
1	1	1	1	1	1	1	0	1	1	1	Short address and device group information, refer to 9.7.2.			Device power cycle				
^a Refer to 9.7.3 for further information on event schemes.																		

^a Refer to 9.7.3 for further information on event schemes.

7.2.2.2 Event source information

The event source information in Table 3 provides:

- the indication that an event message, not an instruction or query, is being transmitted: bit 16 shall be clear for event messages;
- relevant event instance type information, such that the receiver of an event message will be able to understand the meaning of the event;
- relevant event source information, such that the receiver of an event message is able to understand where the message is coming from;
- reserved values.

Events are instance type specific. Application controllers can derive, either explicitly or implicitly, the instance type of the transmitting instance, from the event source information.

NOTE The event source schemes are not equally valuable in terms of telling the receiver where the event message originated.

7.2.2.3 Event information

The event information provides the 10-bit event number and/or event data. Event information is instance type specific and is defined in the applicable parts of the IEC 62386-3xx series that describe the instance type.

8 Timing

The requirements of IEC 62386-101:2022, Clause 8 apply.

9 Method of operation

9.1 General

The requirements of IEC 62386-101:2022, Clause 9 apply with the following additions.

9.2 Device features

The IEC 62386 series allows for the future publication of feature extensions that extend the requirements in this document, or exempt particular requirements.

The features for a device can be queried by "QUERY FEATURE TYPE" and "QUERY NEXT FEATURE TYPE" while the instance byte is set to "feature on device level" (see Table 2).

Table 5 shows the feature type encoding. For further information on the different feature types see parts of the IEC 62386-3xx series.

9.3 Application controller

9.3.1 General

An application controller is that part of a control system that makes the system "work":

- an application controller can commission and configure the system (including available control gear);
- an application controller can make the system react to changes in the environment (based on information coming from input devices);

- an application controller can change the system response of control gear in the system (possibly using any command defined in IEC 62386-102:2022).

9.3.2 Single-master application controller

A single-master application controller is not intended to share the bus with other control devices.

A single-master application controller can try to configure other control devices on the bus, and/or change the system response of control gear in the system, thereby using any command defined in IEC 62386-102 and/or instructions and queries defined in this document.

NOTE Especially if the single-master application controller does not handle collisions appropriately, any such attempt can fail and affect the system negatively.

On the other hand, a single-master application controller is not required to have a receiver on board. For this reason, the following holds:

From 9.3.3 onwards, this document assumes a control device to be a multi-master control device.

In order to make itself known as a possibly anonymous transmitting bus unit, a single-master application controller shall transmit a PING message at regular intervals of $10 \text{ min} \pm 1 \text{ min}$. The first such PING message shall appear at a random time between 5 min and 10 min after completion of the power-on procedure.

9.3.3 Multi-master application controller

From 9.3.3 onwards, this document assumes a control device to be a multi-master control device.

A control device that includes an application controller shall have "*applicationControllerPresent*" set to TRUE. "*applicationControllerPresent*" shall be set to FALSE otherwise.

NOTE 1 "*applicationControllerPresent*" can be observed through "QUERY DEVICE CAPABILITIES".

In most cases, a system will have only one application controller active (refer to 9.10.1), but multiple application controllers can be operational in a single system.

An application controller shall accept commands (from other application controllers) according to Table 23 and Table 24. It is part of the system integration to ensure that the application controllers will accept commands in such a way that it results in a correctly functioning system.

NOTE 2 System integrity is easiest to achieve by allowing only a single application controller to do commissioning and configuration.

NOTE 3 An application controller can be commissioned through alternative interfaces.

An application controller shall not transmit event messages other than for the device power cycle event.

NOTE 4 If an application controller is active, it can send 24-bit forward frames for purposes other than transmitting events.

An application controller shall not transmit PING messages.

9.4 Input device

Input devices make a system sensitive to changes in its environment, by transmitting event messages.

Input devices shall be multi-master control devices and shall allow commissioning and configuration by an application controller.

Input devices shall use forward frames only to transmit event messages.

9.5 Instances of input devices

9.5.1 General

An input device shall have at least one instance and a maximum of 32 instances, as shall be indicated by "*numberOfInstances*", which can be queried using "QUERY NUMBER OF INSTANCES".

A control device that is only an application controller shall have a "*numberOfInstances*" equal to 0.

9.5.2 Instance number

Each instance shall have a unique "*instanceNumber*" in the range $[0, \text{"numberOfInstances"} - 1]$.

9.5.3 Instance type

The instance type for each of the instances of an input device can be different. It can be queried by "QUERY INSTANCE TYPE" while the instance byte is set to either "instance type" or "instance number" scheme (see Table 2). The meaning of event information transmitted by means of "INPUT NOTIFICATION (*device/instance, event*)" depends on the instance type.

Table 4 shows the instance type encoding. For further information on the different instance types see the relevant parts of the IEC 62386-3xx series.

Table 4 – Instance types

Instance type	IEC 62386	Used for
0	Part 103	Generic purpose, input devices that are not defined. Another method of identifying the device shall be implemented, to allow application controller to interpret the events.
1 to 31	Part 301 to Part 331	These IEC 62386-3xx parts describe instance types, where xx ranges from 01 to 31.

9.5.4 Instance features

The IEC 62386 series allows for the future publication of feature extensions that extend the requirements in this document, or exempt particular requirements.

The features for each of the instances of an input device can be different. They can be queried by "QUERY FEATURE TYPE" and "QUERY NEXT FEATURE TYPE" while the instance byte is set to either "feature on instance type level" or "feature on instance number level" scheme (see Table 2).

Table 5 shows the feature type encoding. For further information on the different feature types see the relevant parts of the IEC 62386-3xx series.

Table 5 – Feature types

Feature type	IEC 62386	Used for
32 to 96	Part 332 to Part 396	These IEC 62386-3xx parts describe feature extensions, where xx ranges from 32 to 96.

9.5.5 Instance groups

Instance groups are a means for an application controller to put instances into logical groups, across input devices. Consequently, these logical groups can be used to configure multiple instances at once.

An application controller can use up to 32 such groups, numbered in the range [0,31]. Each instance can be declared to be a member of up to 3 instance groups and shall expose instance group variables as given in Table 6.

Table 6 – Instance group variables

Variable	Description
"instanceGroup0"	Primary instance group number, MASK if no membership defined.
"instanceGroup1"	Additional instance group number, MASK if no membership defined.
"instanceGroup2"	Additional instance group number, MASK if no membership defined.

Instance groups are assigned and queried by using the following instance operations:

- "SET PRIMARY INSTANCE GROUP (*DTR0*)", "QUERY PRIMARY INSTANCE GROUP";
- "SET INSTANCE GROUP 1 (*DTR0*)", "QUERY INSTANCE GROUP 1";
- "SET INSTANCE GROUP 2 (*DTR0*)", "QUERY INSTANCE GROUP 2".

The primary group is special in the sense that only this number shall be used when reporting events (if instance group event reporting is used). Additional groups are a means of configuring multiple instances at once.

9.6 Commands excluding event messages

9.6.1 General

A control device shall check the device addressing scheme to see if it is addressed by a command. The control device shall accept the command, unless any of the following conditions hold:

- the command is sent using short addressing, and the given short address is not equal to "*shortAddress*";
- the command is sent using device group addressing, and the given device group does not match any of the groups identified by "*deviceGroups*";
- the command is sent using broadcast unaddressed addressing and "*shortAddress*" is not MASK;
- the command is sent using reserved addressing;
- the command is not defined;
- the command is sent using feature addressing, and the given feature is not implemented.

NOTE For instance and feature commands, additional conditions for command acceptance hold. These are given in 9.6.3 and 9.6.4.

9.6.2 Device commands

The instance byte shall be 0xFE for device commands. If the instance byte is not equal to 0xFE, the control device shall not accept these commands.

NOTE This addressing mechanism allows the opcode values for device commands and instance commands to overlap.

9.6.3 Instance commands

For instance commands that are accepted by an input device (refer to 9.5), the instance addressing scheme determines the intended (set of) receiving instances within that device. An instance shall accept the instance command, unless any of the following additional conditions hold:

- the command is sent using instance number addressing and the given instance number is not equal to "*instanceNumber*";
- the command is sent using instance group addressing, and the given instance group does not match any of the groups identified by "*instanceGroup0*", "*instanceGroup1*" and "*instanceGroup2*" (see Table 6);
- the command is sent using instance type addressing and the given instance type is not equal to "*instanceType*";
- the command is sent using reserved addressing.

9.6.4 Feature commands

For feature commands that are accepted by a device or instance (refer to 9.5), the feature addressing scheme determines the intended (set of) receiving features within that device (see Table 2).

A feature on device level shall accept the feature command unless any of the following additional conditions hold:

- the command is sent using feature addressing other than feature addressing on the device level;
- the command is sent with instance byte reserved addressing.

A feature on instance level shall accept the feature command unless any of the following additional conditions hold:

- the command is sent using feature on instance number level addressing and the given instance number is not equal to "*instanceNumber*";
- the command is sent using feature on instance group level addressing and the given instance group does not match any of the groups identified by "*instanceGroup0*", "*instanceGroup1*" and "*instanceGroup2*";
- the command is sent using feature on instance type level addressing and the given instance type is not equal to "*instanceType*";
- the command is sent using feature on device level addressing;
- the command is sent with instance addressing;
- the command is sent with device addressing;
- the command is sent with instance byte reserved addressing.

A feature broadcast command shall be used to address a given feature both on device level and instance level. A feature broadcast shall be accepted unless any of the following additional conditions hold:

- the command is sent using feature addressing other than feature broadcast;
- the command is sent with instance byte reserved addressing.

If the feature command is accepted, the opcode byte determines which feature is addressed.

9.7 Event messages

9.7.1 Response to event messages

An application controller is free to act upon reception of any event message or to ignore the message.

NOTE A disabled application controller does not send any forward frames, but can still update its internal state based on messages received.

9.7.2 Device power cycle event

Since the power cycle event (see 9.13.2) is a device event, it does not adhere to the default event frame format. Bits 12 through 0 shall carry device address information as indicated in Table 7.

Table 7 – Device address information in power cycle event

Bits												
12	11	10	09	08	07	06	05	04	03	02	01	00
1 = device group valid	Lowest device group					1 = short address valid	Short address					

Bit 12 shall be set if and only if the transmitting control device is member of at least one device group. Bits [11:7] shall indicate the lowest device group number of membership in that case. If bit 12 is not set, bits [11:7] shall be clear.

Bit 6 shall be set if and only if the transmitting control device has a "*shortAddress*" different from MASK. Bits [5:0] shall indicate the device short address in that case. If bit 6 is not set, bits [5:0] shall be clear.

9.7.3 Input notification event

An instance of an input device shall, when transmitting an event message, use the selected event source addressing scheme as defined in Table 8.

Table 8 – Event addressing schemes

"eventScheme"	Description
0 (default)	Instance addressing, using instance type and number.
1	Device addressing, using short address and instance type.
2	Device and instance addressing, using short address and instance number.
3	Device group addressing, using device group and instance type.
4	Instance group addressing, using instance group and type.

An application controller can set and query the "*eventScheme*" by means of "SET EVENT SCHEME (*DTR0*)" and "QUERY EVENT SCHEME" respectively.

NOTE 1 An instance can only implement an event scheme while certain conditions have been satisfied by the application controller as well. Instance addressing is the only addressing scheme that will work under all circumstances.

In the following situations, the instance shall immediately revert "*eventScheme*" to the default instance addressing scheme shown in Table 8:

- "*eventScheme*" has been set to 1 or 2 whereas the containing logical unit has no short address;
- "*eventScheme*" has been set to 3 whereas the containing logical unit is not a member of a device group;
- "*eventScheme*" has been set to 4 whereas the instance has no primary instance group membership (see 9.5.5).

NOTE 2 The above situations can occur because of a new "SET EVENT SCHEME (*DTR0*)" command and/or because of a change of conditions.

NOTE 3 This implies that the command "SET EVENT SCHEME (*DTR0*)" can "fail", rather than setting a preference that is granted sometime later. To avoid this failure, application controllers can set the desired event scheme after completing those configuration aspects that influence event scheme operation.

Furthermore, and given a viable addressing scheme, the instance shall

- only refer to "*instanceNumber*" as the instance number;
- only refer to "*instanceType*" as the instance type;
- only refer to "*instanceGroup0*" as the instance group;
- only refer to "*shortAddress*" as the short address of the containing logical unit;
- only refer to the lowest device group number that the containing logical unit is a member of.

9.7.4 Event message filter

The event message filter can be used to enable and disable specific events. While the event filter of a specific event is disabled, this specific event shall not be generated. To enable or disable all events see 9.10.3.

An application controller can set the "*eventFilter*" by means of SET EVENT FILTER (*DTR2:DTR1:DTR0*) and can query the variable by means of QUERY EVENT FILTER 0-7, QUERY EVENT FILTER 8-15 and QUERY EVENT FILTER 16-23 respectively.

The relevant parts of the IEC 62386-3xx series shall define the meaning of the bits in "*eventFilter*", and can reduce the width of the variable "*eventFilter*" if needed. If the width is reduced to 2 bytes, *DTR2* shall be ignored for SET EVENT FILTER (*DTR2:DTR1:DTR0*) and QUERY EVENT FILTER 16-23 shall answer NO. Similarly, if the width is reduced to 1 byte, *DTR1* shall additionally be ignored for SET EVENT FILTER (*DTR2:DTR1:DTR0*) and QUERY EVENT FILTER 8-15 shall also answer NO.

9.8 Input signal, measured value and "*inputValue*"

9.8.1 General

An instance shall process its input signal into a measured value and expose this value to the system, as described in 9.8.2 to 9.8.4.

9.8.2 Input resolution

The processing shall be done with a precision which is indicated by "*resolution*". The actual resolution used for a particular instance (type) can be subject to requirements in relevant parts of the IEC 62386-3xx series and/or manufacturer choice.

The measured value shall be contained in the N -byte variable "*inputValue*", where N is the minimum number of bytes needed to contain at least "*resolution*" bits.

NOTE N is computed as ($\text{"resolution"}/8$) rounded up to the nearest integer. With "*resolution*" in the range [1,255], "*inputValue*" can span up to 32 bytes.

The measured value shall be most significant bit (msb)-aligned in "*inputValue*". Unused bits in "*inputValue*" shall contain a repeating pattern of the most significant bit(s) of the measured value.

Table 9 provides an example, which shows a measured value of almost 50 % latched into a 1-byte "*inputValue*" after being processed with a "*resolution*" of 3, 4 and 5 bits respectively.

Table 9 – Measured value ($\approx 50\%$) versus resolution and "*inputValue*"

Resolution	Measured value	Bits								" <i>inputValue</i> "
		7	6	5	4	3	2	1	0	
3-bits	3 of [0, 7]	0	1	1	0	1	1	0	1	109
4-bits	7 of [0, 15]	0	1	1	1	0	1	1	1	119
5-bits	15 of [0, 31]	0	1	1	1	1	0	1	1	123

NOTE The grey shaded bits are (part of) the (first) repetition of the significant bits.

This method allows an application controller to interpret the input value correctly as an 8-bit value, regardless of the actual instance resolution or sensor precision. The minimum value of all bytes in "*inputValue*" is always 0, the maximum value 0xFF, for all resolutions. The relative measured value corresponds (albeit with variable accuracy) to the relative input value.

9.8.3 Getting the input value

An instance shall support a latching mechanism that allows an application controller to obtain a consistent multi-byte input value. An example of such latching scenario is given in Table 10.

Application controllers can start reading a multi-byte value by sending the command "QUERY INPUT VALUE". This command shall trigger a latch that contains a copy of "*inputValue*" such that the remaining bytes can be read using a sequence of "QUERY INPUT VALUE LATCH" queries. After having returned the last byte of the latch, the instance shall not answer further "QUERY INPUT VALUE LATCH" queries until after the next "QUERY INPUT VALUE".

Table 10 – Example of querying sequence to read a 4-byte input value

Input signal	" <i>inputValue</i> "	Command	Answer	Latched " <i>inputValue</i> "
"12340000"	0x12340000	...	...	unspecified
"12345678"	0x12345678	"QUERY INPUT VALUE"	0x12	0x12345678
"852"	0x00000852	"QUERY INPUT VALUE LATCH"	0x34	0x12345678
"124852"	0x00124852	"QUERY INPUT VALUE LATCH"	0x56	0x12345678
"124852"	0x00124852	"QUERY INPUT VALUE LATCH"	0x78	0x12345678
"124852"	0x00124852	"QUERY INPUT VALUE LATCH"	NO	0x12345678

The "*inputValue*" that is latched is the "*inputValue*" at the moment "QUERY INPUT VALUE" is executed.

NOTE 1 This implies that if an application controller queries the "*inputValue*" because of an event message it has just received, the value obtained is not necessarily the same value that triggered the event.

The latched value shall be updated only when the next "QUERY INPUT VALUE" is executed. If the application controller uses "QUERY INPUT VALUE LATCH" without having used "QUERY INPUT VALUE" as the command before this one, the answer can contain old or invalid data.

NOTE 2 To prevent concurrent access to the latched data, an application controller can transmit the necessary queries for this scenario within a transaction, exiting the scenario at any point.

NOTE 3 If an application controller can work sufficiently accurately with 16-bit input for the given instance type, it can stop after having received the most significant 16 bits of input value, and handle those bits as if they were delivered by an instance with "resolution" equal to 16. This allows straightforward resolution-independent algorithm implementation.

9.8.4 Notification of changes

A change or a sequence of changes in the input signal of an instance shall result in an event message as required by this document or that part of the IEC 62386-3xx series that describes the "instanceType" (see 9.5.3) of that instance.

The event message shall be sent using "INPUT NOTIFICATION (*device*/*instance*, *event*)", as described in 11.3.1.

NOTE The manufacturer of the input device can ensure that no event is lost by providing a queue. Parts of the IEC 62386-3xx series can impose additional restrictions, e.g. to avoid event flooding.

9.9 System failure

An application controller should detect system failure and recovery. Preferably, it should act upon any bus power failure with a duration longer than 40 ms, thus anticipating a power cycle of bus powered devices.

NOTE Bus powered devices can shut down due to a power outage of more than 40 ms.

Next, when the system failure is resolved, the application controller should ensure that the system resumes normal operation.

9.10 Operating a control device

9.10.1 Enable/disable the application controller

If present, the application controller is either active or not-active, as shall be reflected by "applicationActive". While deactivated, the application controller shall not send any forward frames, except possibly a power cycle notification (see 9.13.2).

"applicationActive" shall have no influence on the response to incoming forward transmissions, including the transmission of backward frames following queries.

NOTE This allows the application controller to monitor the bus, but the application controller cannot use forward frames to react.

"applicationActive" shall be stored in the NVM of the application controller. The default value shall be TRUE in case there is an application controller present, which can be changed by another application controller using the commands ENABLE APPLICATION CONTROLLER and DISABLE APPLICATION CONTROLLER. "applicationActive" can be queried using "QUERY APPLICATION CONTROLLER ENABLED".

9.10.2 Application controller always active

If an application controller is present it can be always active. This shall be reflected by "applicationControllerAlwaysActive" being TRUE.

When "applicationControllerAlwaysActive" is TRUE, "applicationControllerPresent" and "applicationActive" shall always be TRUE.

"*applicationControllerAlwaysActive*" can be observed through "QUERY DEVICE CAPABILITIES" and "QUERY APPLICATION CONTROLLER ALWAYS ACTIVE".

9.10.3 Enable/disable event messages

Event messages are either enabled or disabled, as shall be reflected by "*instanceActive*". While deactivated, the instance shall not send any forward frames. That is, the instance will not produce any event messages.

"*instanceActive*" shall have no influence on the response to incoming forward transmissions, including the transmission of backward frames following queries.

"*instanceActive*" shall be stored in the persistent memory of the input device. The default value shall be TRUE, which can be changed by an application controller using the commands "ENABLE INSTANCE" and "DISABLE INSTANCE". "*instanceActive*" can be queried using "QUERY INSTANCE ENABLED".

To limit the event messages when enabled, filtering is also available, see 9.7.4.

NOTE Queries are the only way to get information from an instance when event messages are disabled.

9.10.4 Quiescent mode

In quiescent mode, the control device shall not produce any forward frames except as a possible result of execution of SEND TESTFRAME. No commands (see also 9.10.1), and no event messages (see also 9.10.3) shall be transmitted, regardless of "*applicationActive*" or any "*instanceActive*". Event messages shall not be queued and shall not be set as pending. This means that such events are discarded.

Quiescent mode is a temporary mode which is started or restarted with the command "START QUIESCENT MODE". It shall end automatically 15 min $\pm$ 1,5 min after the last "START QUIESCENT MODE" command was executed. Additionally, the command "STOP QUIESCENT MODE" shall terminate quiescent mode immediately.

In quiescent mode, a control device shall still respond to commands. "QUERY QUIESCENT MODE" can be used to determine whether or not a control device is in quiescent mode.

At power on of the control device, quiescent mode shall be DISABLED.

NOTE 1 Quiescent mode can be used by the application controller during initialisation (see 9.15) to ensure that random address comparisons are not frustrated by forward frames from other devices on the bus.

NOTE 2 Quiescent mode works independently from "*applicationActive*" and "*instanceActive*". This implies that ending quiescent mode does not necessarily enable forward frame transmissions.

9.10.5 Modes of operation

9.10.5.1 General

Different operating modes can be selected at device level by means of command "SET OPERATING MODE (*DTR0*)". The currently selected "*operatingMode*" can be queried by means of "QUERY OPERATING MODE".

Operating modes 0x00 to 0x7F are defined in this document. At least operating mode 0x00 shall be available. Operating modes 0x80 to 0xFF are manufacturer specific. The query "QUERY MANUFACTURER SPECIFIC MODE" can be used to determine whether the control device is in an IEC 62386 standard operating mode, or in a manufacturer-specific mode.

9.10.5.2 Operating mode 0x00: standard mode

If a device is in "*operatingMode*" 0x00, its behaviour shall be as is required by this document, until it is set in an operating mode different from 0x00.

9.10.5.3 Operating mode 0x01 to 0x7F: reserved

Operating modes 0x01 to 0x7F are reserved and shall not be used.

9.10.5.4 Operating mode 0x80 to 0xFF: manufacturer-specific modes

Manufacturer-specific modes should only be used if the features required by the application are not covered by the IEC 62386 series. If a control device is in a manufacturer-specific operating mode, the behaviour of the control device may be manufacturer specific as well, with the following exceptions:

- with regard to bus access, the control device shall adhere to IEC 62386-101:2022.
- the control device shall adhere to this document at least as far as the following commands are concerned:
 - "SET OPERATING MODE (*DTR0*)", and "QUERY OPERATING MODE" and "QUERY MANUFACTURER SPECIFIC MODE".
 - All special commands (see 11.10) except WRITE MEMORY LOCATION (*DTR1*, *DTR0*, *data*), WRITE MEMORY LOCATION – NO REPLY (*DTR1*, *DTR0*, *data*) and DIRECT WRITE MEMORY (*DTR1*, *offset*, *data*).

For the above commands, the requirements of 7.2.1 shall apply.

It is recommended that even in manufacturer-specific modes, the commands as specified in all implemented parts of the IEC 62386 series still be obeyed.

9.11 Memory banks

9.11.1 General

Memory banks are freely accessible memory spaces containing information or configuration settings of the control device. Not all consecutive memory banks need to be implemented. Also within a memory bank not all consecutive locations need to be implemented. All implemented memory bank locations of all implemented memory banks are readable using memory access commands. Part of the memory is read-only and either programmed by the manufacturer of the control device or updated by the control device itself. For all other parts, write access using memory access commands can be enabled by the manufacturer. Except for location 0x02, all writable locations of a memory bank are lockable unless specified in the corresponding memory bank table. Locations within the memory banks shall be implemented using the memory types shown in Table 11.

Table 11 – Memory types

Memory type	Accessibility via the bus ^a	Volatility ^b	May be changed autonomously by the control device during run time	Description
ROM	RO	NV	No	For all fixed values that will not change during run time of the control device. NOTE ROM is read-only by its nature, but can be changed during production programming.
RAM-RO	RO	V	Yes	For values that will not be retained through a power-cycle. Cannot be changed over the interface.
RAM-RW	RW	V	Yes	For values that will not be retained through a power-cycle. Can be changed over the interface.
NVM-RO	RO	NV	Yes	For values that will be retained through a power-cycle. Cannot be changed over the interface.
NVM-RW	RW	NV	Yes	For values that will be retained through a power-cycle. Can be changed over the interface.
^a RO: Read-only. RW: Read-write.				
^b V: volatile (not retained through a power-cycle). NV: non-volatile (retained through a power-cycle).				

The addressable memory space is limited to a maximum of 256 memory banks of maximum 255 bytes each (approximately 64 kBytes). As this document specifies how to implement memory bank 0 and 1 (if present), and reserves memory banks 200 to 255, this leaves room for 198 memory banks for manufacturer-specific purposes in the range of [2,199].

9.11.2 Memory map

If a manufacturer-specific memory bank in the range of [2,199] is implemented, allocation of its content shall comply with the memory map provided in Table 12.

Table 12 – Basic memory map of memory banks

Address	Description	Default value (factory)	Reset value ^b	Memory type
0x00	Address of last accessible memory location	factory burn-in, range [0x03,0xFE]	no change	ROM
0x01	Indicator byte ^a	a	a	any ^a
0x02	Memory bank lock byte. Lockable bytes in the memory bank shall be read-only while the lock byte has a value different from 0x55.	0xFF	0xFF ^c	RAM-RW
[0x03,0xFE]	Memory bank content ^a	a	a	any ^a
0xFF	Reserved – not implemented	answer NO	no change	n.a.
^a The purpose, default values, power on values, reset values, and memory types of these bytes shall be defined by the manufacturer.				
^b Reset value after "RESET MEMORY BANK".				
^c Also used as power on value unless explicitly stated otherwise.				

The byte in location 0x00 of each bank contains the address of the last accessible memory location of the bank. The value shall be in the range [0x03,0xFE].

The byte in location 0x01 is manufacturer specific. If implemented, the usage of this byte should be described by the manufacturer (as well as the entire content of the memory bank).

NOTE 1 It could be used for example to store a checksum in case of a memory bank with static content. Using a checksum on a memory bank where the content is changed by the control device is not useful.

The byte in location 0x02 shall be used to lock write access. Memory location 0x02 itself shall never be locked for writing. While this memory location contains any value different from 0x55, all memory locations marked "(lockable)" of the corresponding memory bank shall be read only. The control device shall not change the value of the lock byte other than as a consequence of a power cycle or of a "RESET MEMORY BANK (*DTR0*)" command or other command affecting the lock byte.

Location 0xFF is a reserved location in every memory bank, and is not accessible. This location shall not be implemented as a normal memory bank location. When addressed, the control device shall respond as if this location is not implemented, and it shall not increment "*DTR0*".

NOTE 2 This location is reserved in order to stop the auto increment of "*DTR0*".

9.11.3 Selecting a memory bank location

In order to select a memory bank location a combination of memory bank number and location inside the memory bank is required.

The memory bank shall be selected by setting the memory bank number in "*DTR1*". The location in the memory bank shall be selected by the value in "*DTR0*".

9.11.4 Protectable memory locations

Memory bank locations marked as "protectable" can have read or write protection applied or removed using a manufacturer-specific method.

Protectable locations with read protection enabled shall reply MASK as a result of "READ MEMORY LOCATION (*DTR1*, *DTR0*)". In this case, the implementation or provision of the related feature is optional.

NOTE 1 Read protection can be enabled by default. If read protection is enabled, this means that the related feature is possibly not implemented and there is no obligation for the manufacturer to provide a mechanism to deactivate read protection.

Protectable locations with write protection enabled cannot be overwritten.

NOTE 2 This means no reply to the WRITE MEMORY LOCATION command when attempting to write to a write-protected location.

9.11.5 Memory bank reading

9.11.5.1 General

A selected memory bank location can be read using command "READ MEMORY LOCATION (*DTR1*, *DTR0*)". The answer shall be the value of the byte at the addressed memory bank location.

If the selected memory bank is not implemented, the command shall be discarded.

If the selected memory bank location is below location 0xFF, "*DTR0*" shall be incremented by one, even if the memory location is not implemented. Otherwise, "*DTR0*" shall not change. This mechanism allows for easy consecutive reading of memory bank locations.

After reading a number of bytes from a memory bank, the application controller should check the value of "*DTR0*" to verify it is at the expected or desired location. Any mismatch indicates an error while reading.

9.11.5.2 Reading multi-byte values

To ensure consistent data when reading a multi-byte value from a memory bank, a mechanism shall be implemented that latches all bytes of the multi-byte value when the first byte of the multi-byte value is read, and holds them latched until the first byte of any multi-byte or single byte value in any memory bank of the same logical unit is read.

Reading from a multi-byte location shall reply with the stored value, and not values from the write-buffer that is used to buffer values before the complete value is written to memory (see 9.11.6.3).

9.11.5.3 Unimplemented locations

If a memory bank exists, and the selected memory bank location is

- not implemented and MASK is not shown in the allowed range of values, or
- above the last accessible memory location,

the answer to "READ MEMORY LOCATION (*DTR1*, *DTR0*)" shall be NO.

If a memory bank exists, and the selected memory bank location is

- not implemented, and
- the allowed range for the value stored in the memory bank location includes MASK,

the answer to "READ MEMORY LOCATION (*DTR1*, *DTR0*)" shall be MASK.

NOTE The use of MASK in the range of allowed values shown in a memory bank table, effectively allows the implementation of the location to be optional.

9.11.5.4 Temporarily unavailable locations

For memory locations which include TMASK in the range of allowed values in a memory bank table, the answer to "READ MEMORY LOCATION (*DTR1*, *DTR0*)" shall be TMASK if the value is temporarily unavailable. Except in fault conditions, the control device shall provide a valid value within 30 s.

EXAMPLE After an external supply power cycle, memory bank values such as measurements can be temporarily unavailable due to the settling time of digital filters, and is indicated by the use of TMASK.

NOTE If TMASK is returned repeatedly over a period of time, application controllers can deduce that there is a fault in the control device.

9.11.5.5 Latching a complete memory bank for reading

Memory bank tables that show the lock byte value 0xAA latches the complete bank, shall operate as follows:

If the lock byte contains a value other than 0xAA, writing the following values to the lock byte shall cause the stated result:

- 0xAA: All locations in the memory bank shall be latched and shall not change until the lock byte is written, or a power cycle occurs.
- Other values: The memory bank latch shall not be affected.

If the lock byte contains the value 0xAA, writing the following values to the lock byte shall cause the stated result:

- 0xAA: All locations in the memory bank shall be re-latched (updated) and shall not change again until the lock byte is written, or a power cycle occurs.
- Other values: The memory bank latch shall be removed. Memory reads shall result in the latest values being returned.

An attempt to write to any location other than the lock byte of a latched memory bank shall result in the same behaviour as if the memory location is not implemented.

Latching of a full memory bank shall not affect reading or writing of other memory banks.

EXAMPLE The following example demonstrates two application controllers accessing a latched memory bank: If application controller A latches a full memory bank 202, then application controller B needs to be aware that memory bank 202 is latched. When application controller B reads from memory bank 202, application controller B will get the latched data. Application controller B can determine this by either monitoring the bus activity related to writing the lock byte, or by reading the lock byte before reading other locations.

9.11.6 Memory bank writing

9.11.6.1 General

Write commands are special commands and therefore not addressable. In order to select the correct control device(s) the addressable command "ENABLE WRITE MEMORY" shall be used. Upon execution of "ENABLE WRITE MEMORY", the addressed control device(s) shall set "*writeEnableState*" to ENABLED.

Only while "*writeEnableState*" is ENABLED, and the addressed memory bank is implemented, the control device shall execute the following commands to write to a selected memory bank location:

- "WRITE MEMORY LOCATION (*DTR1*, *DTR0*, *data*)": The control device shall confirm writing a memory location with an answer equal to the value *data*.

NOTE 1 The value that can be read from the memory bank location is not necessarily *data*.

- "WRITE MEMORY LOCATION – NO REPLY (*DTR1*, *DTR0*, *data*)": Writing a memory location shall not cause the control device to reply.
- "DIRECT WRITE MEMORY (*DTR1*, *offset*, *data*)": The address of the memory location inside the selected bank is given by the content of the instance byte. *offset* is copied to "*DTR0*", after which the command is treated as "WRITE MEMORY LOCATION (*DTR1*, *DTR0*, *data*)". The control device shall confirm writing a memory location by replying with an answer equal to *data*.

A control device shall set "*writeEnableState*" to DISABLED if any command other than one of the following commands is accepted:

- "WRITE MEMORY LOCATION (*DTR1*, *DTR0*, *data*)",
"WRITE MEMORY LOCATION – NO REPLY (*DTR1*, *DTR0*, *data*)",
"DIRECT WRITE MEMORY (*DTR1*, *offset*, *data*)";
- "*DTR0* (*data*)", "*DTR1* (*data*)", "*DTR1:DTR0* (*data1*, *data0*)", *DTR2* (*data*),
DTR2:DTR1 (*data2*, *data1*);
- "QUERY CONTENT *DTR0*", "QUERY CONTENT *DTR1*", "QUERY CONTENT *DTR2*".

NOTE 2 This means that commands other than those shown above, which are accepted and then discarded, still set "*writeEnableState*" to DISABLED.

9.11.6.2 Write failures

If the selected memory bank location is

- not implemented, or
- above the last accessible memory location, or
- locked (see 9.11.2), or
- not writeable, or
- is a protectable location that is currently write protected (see 9.11.4), or
- there is an attempt to write a value outside of the permitted range,

the answer to "WRITE MEMORY LOCATION (*DTR1*, *DTR0*, *data*)" and "DIRECT WRITE MEMORY (*DTR1*, *offset*, *data*)" shall be NO and no memory location shall be written to. If this applies when writing to a multi-byte location, the reply of NO shall occur after attempting to write to the least significant byte (LSB).

If the selected memory bank location is below 0xFF, "*DTR0*" shall be incremented by one. Otherwise, "*DTR0*" shall not change. This mechanism allows for easy consecutive writing to memory bank locations.

9.11.6.3 Writing multi-byte values

To ensure consistent data when writing a multi-byte value into a memory bank, a RAM buffer shall be used such that the buffer stores the temporary bytes being written until the LSB of the multi-byte value is written, at which point the complete value is written to the memory bank locations.

NOTE 1 The contents of buffers used to store temporary bytes for multi-byte writing can be lost when a write operation is started to any other multi-byte value in the same memory bank, or during a power-cycle, or as a result of execution of RESET MEMORY BANK (*DTR0*). This means that it is sufficient to include only one such RAM buffer for writing, per memory bank.

After writing a number of bytes to a memory bank, the application controller should check the value of "*DTR0*" to verify it is at the expected or desired location. Any mismatch indicates an error while writing.

NOTE 2 "*DTR0*" is also incremented if a non-implemented memory bank location is addressed before 0xFF is reached.

9.11.7 Memory bank 0

Memory bank 0 contains information about the control device. Memory bank 0 shall be implemented in all multi-master control devices.

Memory bank 0 shall be implemented using the memory map shown in Table 13, with at least the memory locations up to address 0x7F implemented, excluding reserved locations.

Table 13 – Memory map of memory bank 0

Address	Description	Default value (factory)	Memory type
0x00	Address of last accessible memory location	factory burn-in	ROM
0x01	Reserved – not implemented	answer NO	n.a.
0x02	Number of last accessible memory bank	factory burn-in, range [0,0xFF]	ROM
0x03	GTIN byte 0 (MSB) ^a	factory burn-in	ROM
0x04	GTIN byte 1	factory burn-in	ROM
0x05	GTIN byte 2	factory burn-in	ROM
0x06	GTIN byte 3	factory burn-in	ROM
0x07	GTIN byte 4	factory burn-in	ROM
0x08	GTIN byte 5 (LSB)	factory burn-in	ROM
0x09	Firmware version (major)	factory burn-in	ROM
0x0A	Firmware version (minor)	factory burn-in	ROM
0x0B	Identification number byte 0 (MSB)	factory burn-in	ROM
0x0C	Identification number byte 1	factory burn-in	ROM
0x0D	Identification number byte 2	factory burn-in	ROM
0x0E	Identification number byte 3	factory burn-in	ROM
0x0F	Identification number byte 4	factory burn-in	ROM
0x10	Identification number byte 5	factory burn-in	ROM
0x11	Identification number byte 6	factory burn-in	ROM
0x12	Identification number byte 7 (LSB)	factory burn-in	ROM
0x13	Hardware version (major)	factory burn-in	ROM
0x14	Hardware version (minor)	factory burn-in	ROM
0x15	101 version number ^b	factory burn-in, according to implemented version number	ROM
0x16	102 version number of all integrated control gear ^c	factory burn-in, according to implemented version number	ROM
0x17	103 version number of all integrated control devices ^d	factory burn-in, according to implemented version number	ROM
0x18	Number of logical control device units in the bus unit	factory burn-in, range [1,64]	ROM
0x19	Number of logical control gear units in the bus unit	factory burn-in, range [0,64]	ROM
0x1A	Index number of this logical control device unit	factory burn-in, range [0,(location 0x18) – 1]	ROM
0x1B ^f	Current bus unit configuration ^f	factory burn-in ^g	ROM
[0x1C,0x7F]	Reserved – not implemented	answer NO	n.a.
[0x80,0xFE]	Additional control device information ^e	^e	ROM
0xFF	Reserved – not implemented	answer NO	n.a.

Key

GTIN global trade item number

LSB least significant byte

MSB most significant byte

- ^a It is recommended that the product GTIN is not re-used within the expected lifetime of the product after installation.
- ^b Format of the version number is defined in IEC 62386-101:2022, 4.2.
- ^c Format of the version number is defined in IEC 62386-102:2022, 4.2. If not implemented, this is indicated by 0xFF.
- ^d Format of the version number is defined in 4.2.
- ^e Purpose and (default) value of these bytes shall be defined by the manufacturer.
- ^f See 9.20. If this location is not implemented, the answer shall be NO.
- ^g The current bus unit configuration can be changed by a manufacturer-specific method.

If there is more than one logical unit built into one bus unit, all logical units shall have the same values in memory bank locations 0x03 up to and including 0x19. All control device logical units shall have the same values in location 0x1B.

A bus unit can contain both control gear and control devices. They share various numbers (e.g. GTIN, identification number). To avoid problems when reading, and getting different answers depending on the addressing scheme used, the memory bank layout is the same for control gear and for control devices up to and including location 0x19. The data shall be the same as well. The application controller can use either the commands specified in IEC 62386-102:— or the commands specified in this document to identify the basic data, provided both are implemented.

The bytes in locations 0x03 to 0x08 ("GTIN 0" to "GTIN 5") shall contain the global trade item number (GTIN) in binary (see [1]¹). The bytes shall be stored most significant first and filled with leading zeroes.

The bytes in locations 0x09 and 0x0A ("firmware version") shall contain the firmware version of the bus unit.

The bytes in locations 0x0B to 0x12 ("identification number byte 0" to "identification number byte 7") shall contain 64 bits of an identification number of the bus unit. It is recommended that this identification number is the serial number. The identification number shall be stored with the least significant byte in "identification number byte 7" and unused bits shall be filled with 0.

The combination of the identification number and the GTIN number shall be unique.

The byte in location 0x13 and 0x14 ("hardware version") shall contain the hardware version of the bus unit.

The byte in location 0x15 shall contain the implemented IEC 62386-101 version number of the bus unit.

The byte in location 0x16 shall contain the implemented IEC 62386-102 version number of the bus unit. If no control gear is implemented, the version number shall be 0xFF.

¹ Numbers in square brackets refer to the Bibliography.

The byte in location 0x17 shall contain the implemented IEC 62386-103 version number of the bus unit.

The byte in location 0x18 shall contain the number of logical control device units integrated into the bus unit. The number of logical units shall be in the range of 1 to 64.

The byte in location 0x19 shall contain the number of logical control gear units integrated into the bus unit. The number of logical units shall be in the range of 0 to 64.

The byte in location 0x1A shall represent the unique index number of the logical control device unit that implements that memory bank. The valid range of this index number is 0 to the total number of logical control device units in the bus unit minus one.

EXAMPLE A product can contain three control device logical units with three different short addresses. Each of these logical units has the same GTIN and identification number, each reports the number of logical control device units with the value 3 and the index of the three control device logical units is reported as 0, 1 or 2 respectively. Reading location 0x1A using broadcast yields a backward frame according to IEC62386-101:2022, 9.6.2 (corrupted backward frame).

The byte in location 0x1B, if present, shall contain the current bus unit configuration, which shall indicate the currently selected implementation of application controllers and logical units. See 9.20.

9.11.8 Memory bank 1 (optional)

Memory bank 1 is reserved for use by an original equipment manufacturer (OEM), (e.g. a luminaire manufacturer) to store additional information, which has no impact on the functionality of the control device. Implementation of memory bank 1 is optional.

If implemented, memory bank 1 shall at least implement the memory locations up to and including address 0x10. The fixed usage for location 0x00 to 0x02 and the recommended memory map usage for locations 0x03 to 0x10 is shown in Table 14.

Table 14 – Memory map of memory bank 1

Address	Description	Default value (factory)	Reset value ^b	Memory type
0x00	Address of last accessible memory location	factory burn-in, range [0x10,0xFE]	no change	ROM
0x01	Indicator byte ^a	a	a	any ^a
0x02	Memory bank 1 lock byte. Lockable bytes in the memory bank shall be read-only while the lock byte has a value different from 0x55.	0xFF	0xFF ^c	RAM-RW
0x03	OEM GTIN byte 0 (MSB)	0xFF	no change	NVM-RW (lockable)
0x04	OEM GTIN byte 1	0xFF	no change	NVM-RW (lockable)
0x05	OEM GTIN byte 2	0xFF	no change	NVM-RW (lockable)
0x06	OEM GTIN byte 3	0xFF	no change	NVM-RW (lockable)
0x07	OEM GTIN byte 4	0xFF	no change	NVM-RW (lockable)
0x08	OEM GTIN byte 5 (LSB)	0xFF	no change	NVM-RW (lockable)
0x09	OEM identification number byte 0 (MSB)	0xFF	no change	NVM-RW (lockable)

Address	Description	Default value (factory)	Reset value ^b	Memory type
0x0A	OEM identification number byte 1	0xFF	no change	NVM-RW (lockable)
0x0B	OEM identification number byte 2	0xFF	no change	NVM-RW (lockable)
0x0C	OEM identification number byte 3	0xFF	no change	NVM-RW (lockable)
0x0D	OEM identification number byte 4	0xFF	no change	NVM-RW (lockable)
0x0E	OEM identification number byte 5	0xFF	no change	NVM-RW (lockable)
0x0F	OEM identification number byte 6	0xFF	no change	NVM-RW (lockable)
0x10	OEM identification number byte 7 (LSB)	0xFF	no change	NVM-RW (lockable)
≥ 0x11	Additional control device information ^a	a	a	a
0xFF	Reserved – not implemented	answer NO	no change	n.a.
Key GTIN global trade item number LSB least significant byte MSB most significant byte OEM original equipment manufacturer ^a The purpose, default value, power on value, reset value and memory type of these bytes shall be defined by the manufacturer. ^b Reset value after "RESET MEMORY BANK". ^c Also used as power on value.				

The bytes in locations 0x03 to 0x08 ("OEM GTIN 0" to "OEM GTIN 5") should be used to identify the product containing the control device. If the bytes are used for the GTIN, the bytes shall be stored most significant bit first and filled with leading zeroes. These bytes should be programmed by the OEM.

The bytes in locations 0x09 to 0x10 ("OEM identification number byte 0" to "OEM identification number byte 7") should contain 64 bits of an identification number of the OEM product. If the bytes are used for the identification number, it shall be stored with the least significant byte in "identification number byte 7" and unused bits shall be filled with 0. These bytes should be programmed by the OEM.

The combination of OEM GTIN and OEM identification number should be unique.

9.11.9 Manufacturer-specific memory banks

The manufacturer may use additional memory banks in the range of 2 to 199 to store additional information. The memory map of additional banks shall comply with Table 12.

9.11.10 Reserved memory banks

Memory banks 200 to 255 are reserved for future use, or are described in the IEC 62386-3xx series. Implementation or use other than described in the IEC 62386-3xx series is not permitted.

9.12 Reset

9.12.1 Reset operation

A control device shall implement a reset operation to set all device variables and instance variables (see Table 19 and Table 20) to their reset values.

NOTE For some variables this operation could have no effect at all.

The reset operation shall take at most 300 ms to complete. While the reset operation is in progress, the control device can respond to commands or not. However, until the reset operation is complete, none of the affected variables needs to have a defined value.

An application controller can trigger the reset operation using the "RESET" instruction and should wait at least 350 ms to ensure all control devices have finished the reset operation.

9.12.2 Reset memory bank operation

A control device shall implement a reset operation to set the content of all unlocked memory banks to their reset values (see 9.11), followed by locking the memory banks.

NOTE For some memory bank locations this operation could have no effect at all.

The reset memory bank operation shall take at most 10 s to complete. While this reset operation is in progress, the control device can respond to commands or not. However, until this reset memory bank operation is complete, none of the affected memory locations have a defined value.

An application controller can trigger the reset operation for a specific memory bank, or for all implemented memory banks, using the "RESET MEMORY BANK (*DTR0*)" instruction and it should then wait for at least 10,1 s to ensure all control devices have enough time to finish the reset memory bank operation.

9.13 Power on behaviour

9.13.1 Power on

After an external power cycle (see IEC 62386-101:2022, 4.11.1 and 4.11.5), the device shall maintain its most recent configuration of NVM variables saved according to 9.18, with the following exceptions:

- all variables mentioned in Table 19 and Table 20 shall be set to the value indicated in the power on value column. The variables that are marked with 'no change' in the power on value column shall not be considered. The variables defined in implemented parts of the IEC 62386-3xx series shall be included;
- the memory bank write enable state shall be disabled for all memory banks and the lock byte shall be set to 0xFF;
- quiescent mode shall be cancelled (see 9.10.4);
- all running timers shall be stopped and cancelled (reset);
- "*powerCycleSeen*" shall be set to TRUE.

"*powerCycleSeen*" can be observed through "QUERY DEVICE STATUS".

In order to observe a subsequent power cycle, the application controller should clear "*powerCycleSeen*", using the command "RESET POWER CYCLE SEEN".

NOTE In a system with multiple application controllers that make use of power cycle information of other control devices in the system, application controllers can take consideration when clearing "*powerCycleSeen*".

9.13.2 Power cycle notification

After completing its power on behaviour according to 9.13.1, a bus unit shall generate one power cycle event message per device if the "*powerCycleNotification*" is ENABLED for at least one of its logical units.

An application controller can use "ENABLE POWER CYCLE NOTIFICATION" and "DISABLE POWER CYCLE NOTIFICATION" to enable/disable power cycle events for specific control devices. "*powerCycleNotification*" shall be DISABLED by default.

NOTE 1 The power cycle notification is not inhibited by "*applicationActive*" nor by any "*instanceActive*".

The event shall be generated using the "POWER NOTIFICATION (*device*)" message as described in 11.2. The event message shall be sent once using priority 2 and with a uniformly distributed delay between 1,3 s and 5,0 s after completion of the power-on procedure.

NOTE 2 Applying a random delay helps to avoid collisions of power cycle notifications.

9.14 Priority use

9.14.1 General

The purpose of forward frame priorities is to facilitate appropriate system behaviour within a multi-master system. Priorities ensure that transmissions for time critical system reaction will have precedence over transmissions for non-time critical system operation.

- The first forward frame in a transaction (see IEC62386-101:2022, 9.3) shall be sent with priority 2 to 5. All other forward frames in a transaction shall be sent with priority 1.
- Forward frames that are not part of a transaction shall be sent with priority 2 to 5.
- Priority 2 should be used to execute user instigated actions for switching or dimming the lights. This implies appropriate event messages and level instructions. Priority 2 can also be used during commissioning (e.g. addressing).

EXAMPLE 1 Switching or dimming actions triggered via push-button or presence detector.

- Priority 3 should be used for configuration of a bus unit and for those event messages that are not covered by Priorities 2 and 4.

EXAMPLE 2 Writing to memory banks or feedback events.

- Priority 4 should be used to execute automatic actions for switching or dimming the lights. This means sending appropriate event messages and level instructions.

EXAMPLE 3 Switching or dimming actions triggered by a light sensor.

- Priority 5 should be used for periodic query commands.

9.14.2 Priority of input notifications

An instance shall use a default "*eventPriority*" equal to Priority 4 when transmitting an event message to produce an "INPUT NOTIFICATION (*device/instance, event*)". For particular instance types, this default priority is subject to change by the relevant parts of the IEC 62386-3xx series.

In a system, the default "*eventPriority*" can be overruled by the application controller using "SET EVENT PRIORITY (*DTR0*)". "QUERY EVENT PRIORITY" can be used to observe the currently active "*eventPriority*".

9.15 Assigning short addresses

9.15.1 General

"shortAddress" shall be derived from *data* or "DTR0" depending on the command used. It shall be set on execution of "PROGRAM SHORT ADDRESS (*data*)" or "SET SHORT ADDRESS (*DTR0*)" as follows:

- if *data* or "DTR0" = MASK: MASK (effectively deleting the short address);
- if *data* or "DTR0" < 0x40: *data* or "DTR0";
- in all other cases: no change.

9.15.2 Random address allocation

A control device shall implement an initialisation state, only in which, apart from the other operations identified in this document, a set of commands are enabled that allow an application controller to detect and uniquely identify control devices available on the bus and assign short addresses to these devices.

The initialisation state is a temporary state which is entered with the command "INITIALISE (*device*)". It shall end automatically 15 min ± 1,5 min after the last "INITIALISE (*device*)" command was executed. Additionally, a power cycle or the command "TERMINATE" shall cause the control device to leave the initialisation state immediately.

The control device shall have three possible values for "*initialisationState*":

- DISABLED, not in initialisation state;
- ENABLED, in initialisation state;
- WITHDRAWN, in initialisation state, yet identified and withdrawn.

The following (special) commands are initialisation commands:

- "RANDOMISE", "COMPARE" and "WITHDRAW";
- "SEARCHADDRH (*data*)", "SEARCHADDRM (*data*)" and "SEARCHADDRL (*data*)";
- "PROGRAM SHORT ADDRESS (*data*)", "VERIFY SHORT ADDRESS (*data*)" and "QUERY SHORT ADDRESS";
- "IDENTIFY DEVICE".

NOTE "IDENTIFY DEVICE" is by itself not an initialisation command, but typically used during initialisation.

9.15.3 Identification of a device

No variables shall be affected by the identification procedure. Normal processing shall continue, i.e. events can be generated on a change of the input value; this shall not stop identification. Where appropriate, variables can be temporarily ignored, so that after the identification has ended, there are no side effects.

Identification shall be stopped upon execution of any instruction other than INITIALISE (*device*) or "IDENTIFY DEVICE".

Identification can be started by sending the instruction "IDENTIFY DEVICE". This shall start or restart a 10 s ± 1 s timer. While the timer is running, a procedure enabling an observer to identify the selected control device shall run. If the timer expires, identification shall stop.

NOTE The actual procedure is manufacturer specific.

When identification is stopped by an application controller, the corresponding timer shall be cancelled immediately.

9.16 Exception handling

Control devices and instances shall expose whether an error has occurred by setting (in case of error) and resetting (in case of no error) the following flags.

- An application controller shall change "*applicationControllerError*". This status can be queried through "QUERY DEVICE STATUS" (see 9.17.2). Detailed error information can be obtained from "QUERY APPLICATION CONTROLLER ERROR" (see 11.6.4).
- A control device that is not an application controller shall have "*applicationControllerError*" set to FALSE.
- An input device shall change "*inputDeviceError*" to indicate an error in any of the input device instances of the logical unit. This status can be queried through "QUERY DEVICE STATUS" (see 9.17.2). Detailed error information can be obtained from "QUERY INPUT DEVICE ERROR" (see 11.6.5).
- A control device that is not an input device shall have "*inputDeviceError*" set to FALSE.
- An instance shall change "*instanceError*". This status can be queried through "QUERY INSTANCE STATUS" (see 9.17.3). Detailed error information can be obtained from "QUERY INSTANCE ERROR" (see 11.9.4).

9.17 Device capabilities and status information

9.17.1 Device capabilities

Each control device shall expose its features as a combination of device capabilities as given in Table 15:

Table 15 – Control device capabilities

Bit	Description	Value	See Subclause
0	" <i>applicationControllerPresent</i> " is TRUE?	"1" = "Yes"	9.3.3
1	" <i>numberOfInstances</i> " is greater than 0?	"1" = "Yes"	9.5.2
2	" <i>applicationControllerAlwaysActive</i> " is TRUE?	"1" = "Yes"	9.10.2
3 to 4	Reserved for IEC 62386-104 (see [2])	"0" = default value	
5	At least one instance supports configuration of " <i>instanceType</i> " or " <i>instanceConfiguration</i> []"	"1" = "Yes"	9.19
6 to 7	unused	"0" = default value	

The device capabilities can be queried using "QUERY DEVICE CAPABILITIES".

9.17.2 Device status

Each control device shall expose its status as a combination of device properties as given in Table 16:

Table 16 – Control device status

Bit	Description	Value	See Subclause
0	"inputDeviceError" is TRUE?	"1" = "Yes"	9.16
1	"quiescentMode" is ENABLED?	"1" = "Yes"	9.10.4
2	"shortAddress" is MASK?	"1" = "Yes"	9.15.1
3	"applicationActive" is TRUE?	"1" = "Yes"	9.10.1
4	"applicationControllerError" is TRUE?	"1" = "Yes"	9.16
5	"powerCycleSeen" is TRUE?	"1" = "Yes"	9.13
6	"resetState" is TRUE?	"1" = "Yes"	See below
7	Reserved for IEC 62386-104 (see [2])	"0" = default value	

The device status can be queried using "QUERY DEVICE STATUS".

Bit 6: "resetState" shall be set to TRUE if all the NVM variables mentioned in Table 19 and Table 20 are at their reset value. The NVM variables that are marked with 'no change' in the reset value column shall not be considered. NVM variables defined in implemented parts of the IEC 62386-3xx series shall be included. In all other cases the bit shall be set to FALSE.

9.17.3 Instance status

Each instance shall expose its status as a combination of instance properties as given in Table 17:

Table 17 – Instance status

Bit	Description	Value	See subclause
0	"instanceError" is TRUE?	"1" = "Yes"	9.16
1	"instanceActive" is TRUE?	"1" = "Yes"	9.10.3
2 to 7	unused	"0" = default value	

The instance status can be observed using "QUERY INSTANCE STATUS".

9.18 Non-volatile memory

After any change to an NVM variable, the new value shall be restored after a power cycle, provided a period of at least 30 s followed the NVM variable change, before the power cycle.

In case the above condition is not met, it is possible that the changed NVM variable will not be restored with the new value.

NOTE The method used to ensure NVM variables are saved is manufacturer specific. One possible method is to save some NVM variables within 30 s of them being changed, with others being saved at the time the external power failure occurs. The method chosen is likely to influence the lifetime of the non-volatile memory.

9.19 Instance types and configuration

Instances may optionally support a change of "instanceType" with the instruction "SET INSTANCE TYPE (DTR0)".

The current instance type can be determined by using "QUERY INSTANCE TYPE", and all available instance types can be discovered by use of "QUERY AVAILABLE INSTANCE TYPES".

Instances may optionally support a change of *"instanceConfiguration[]"* with the instruction "SET INSTANCE CONFIGURATION (*DTR0*, *DTR2:DTR1*)".

The current instance configuration can be discovered by using "QUERY INSTANCE CONFIGURATION (*DTR0*)".

Any standardised values of *"instanceConfiguration[]"* are published in the appropriate parts of the IEC 62386-3xx series. A manufacturer-specific range of *"instanceConfiguration[]"* may be used, but use of such values shall not remove any requirements in this document.

If instance configuration is supported, the manual shall state which locations within *"instanceConfiguration[]"* are supported, shall describe the use of such locations in the case that these are in the manufacturer-specific range, and shall state their memory type (readable and/or writeable).

EXAMPLE 1 Changing the instance type: An instance with volt-free inputs can support a change of instance type between type 1 (push-button) and type 2 (absolute/switch-input device).

EXAMPLE 2 Changing the instance configuration: An occupancy sensor instance with multiple sensor technologies such as PIR, microwave/doppler and ultra-sonic can allow configuration of which sensor technologies are currently active.

9.20 Current bus unit configuration

If present, the value of current bus unit configuration from memory bank 0 shall indicate the currently active implementation of:

- the value of *"applicationControllerPresent"*.

For the given value of current bus unit configuration, the implementation shall be as shown in Table 18.

Table 18 – Current bus unit configuration

Current bus unit configuration	<i>"applicationControllerPresent"</i>	Description
0 to 191 ^a		Reserved
192 to 255		Manufacturer-specific
^a Values reserved for future updates to this document.		

A manufacturer-specific method is permitted for selection of the current configuration. After changing, the bus unit can require a certain time, power-cycle or other requirement before the new configuration becomes active.

After changing the configuration, the value of "Current bus unit configuration" in memory bank 0 shall indicate the active configuration, and the bus unit operation shall be updated according to the new configuration.

10 Declaration of variables

Table 19 shows the default values, the reset values, the range of validity and the type of memory of the defined instance independent variables.

Table 20 shows the default values, the reset values, the range of validity and the type of memory of the defined variables of each of the instances.

The variables that are declared in this Clause 10 shall not be made available for writing through a memory bank.

Table 19 – Declaration of device variables

Variable	Default value (factory)	Reset value	Power on value	Range of validity	Memory type
"shortAddress"	MASK (no address)	no change	no change	[0,63], MASK	NVM
"deviceGroups"	0x0000 0000	0x0000 0000	no change	[0,0xFFFF FFFF]	NVM
"searchAddress"	^a	0xFF FF FF	0xFF FF FF	[0,0xFF FF FF]	RAM
"randomAddress"	0xFF FF FF	0xFF FF FF	no change	[0,0xFF FF FF]	NVM
"DTR0"	^a	no change	0x00	[0,0xFF]	RAM
"DTR1"	^a	no change	0x00	[0,0xFF]	RAM
"DTR2"	^a	no change	0x00	[0,0xFF]	RAM
"numberOfInstances"	factory burn-in	no change	no change	[0,32]	ROM
"operatingMode"	factory burn-in	no change	no change	0, [0x80,0xFF]	NVM
"quiescentMode"	^a	DISABLED	DISABLED	[ENABLED, DISABLED]	RAM
"applicationActive"	applicationControllerPresent	no change	no change	[TRUE, FALSE]	NVM
"writeEnableState"	^a	DISABLED	DISABLED	[ENABLED, DISABLED]	RAM
"applicationControllerPresent"	factory burn-in	no change	no change	[TRUE, FALSE]	ROM
"applicationControllerAlwaysActive"	factory burn-in	no change	no change	[TRUE, FALSE]	ROM
"powerCycleSeen"	^a	FALSE	TRUE	[TRUE, FALSE]	RAM
"powerCycleNotification"	DISABLED	no change	no change	[ENABLED, DISABLED]	NVM
"initialisationState"	^a	no change	DISABLED	[ENABLED, DISABLED, WITHDRAWN]	RAM
"applicationControllerError"	^a	^c	FALSE ^b	[TRUE, FALSE]	RAM
"inputDeviceError"	^a	^c	FALSE ^b	[TRUE, FALSE]	RAM
"resetState"	TRUE	TRUE	TRUE ^b	[TRUE, FALSE]	RAM
"eventPriority" ^d	4	no change	no change	[2,5]	NVM
"versionNumber"	3.0	no change	no change	00001100b	ROM
^a Not applicable. ^b The value should reflect the actual situation as soon as possible. ^c The value could change as a consequence of the RESET command execution. ^d This variable is independent of the instance variable(s) "eventPriority".					

Table 20 – Declaration of instance variables

Variable	Default value (factory)	Reset value	Power on value	Range of validity	Memory type
"instanceGroup0"	MASK	MASK	no change	[0,31], MASK	NVM
"instanceGroup1"	MASK	MASK	no change	[0,31], MASK	NVM
"instanceGroup2"	MASK	MASK	no change	[0,31], MASK	NVM
"instanceActive"	TRUE	no change	no change	[TRUE, FALSE]	NVM
"instanceType"	factory burn-in	no change	no change	[0,31]	ROM or NVM
"resolution"	factory burn-in	no change	no change	[1,255]	ROM
"inputValue"	^a	no change	no change ^b	$[0, 2^{N*8} - 1]^c$	RAM
"instanceNumber"	factory burn-in	no change	no change	[0, "numberOfInstances" - 1]	ROM
"eventFilter" ^d	0xFF FF FF	0xFF FF FF	no change	[0, 0xFF FF FF]	NVM
"eventScheme"	0	0	no change	[0,4]	NVM
"eventPriority" ^{d f}	4	no change	no change	[2,5]	NVM
"instanceError"	^a	^e	FALSE ^b	[TRUE, FALSE]	RAM
"instanceConfiguration[]" ^g	factory burn-in	no change	no change	[0, 0xFFFF] ^g	NVM or ROM

^a Not applicable.
^b The value should reflect the actual situation as soon as possible.
^c N computed as ("resolution"/8) rounded up to the nearest integer.
^d For particular instance types, the values belonging to this variable can be changed by the relevant part of the IEC 62386-3xx series.
^e The value could change as a consequence of the RESET command execution.
^f Instance variable(s) "eventPriority" are independent of the device variable "eventPriority".
^g The size of this array is manufacturer-specific, from 0 to 256 locations. Implemented locations are up to 16 bits in size.

11 Definition of commands

11.1 General

Unused opcodes are reserved for future needs.

11.2 Overview sheets

Table 21 to Table 24 give an overview of the control device's event messages, standard commands and special commands.

Table 21 – Instance event messages

Event message name	Event source information	Event information	References	Command subclause
INPUT NOTIFICATION (<i>device/instance, event</i>)	<i>device/instance</i>	<i>event</i>	9.7.3 and 9.8.4	11.3.1

Table 22 – Device event messages

Event message name	Bits [23,13]	Bits [12,0]	References	Command subclause
POWER NOTIFICATION (<i>device</i>)	0x7F7	<i>device</i>	9.7.2 and 9.13.2	11.3.2

IECNORM.COM : Click to view the full PDF of IEC 62386-103:2022

Table 23 – Standard commands

Command name	Address byte	Instance byte			Opcode byte	App Ctrl	Input device	DTR0	DTR1	DTR2	Answer	Send twice	References	Command subclause
		Device	Instance	Feature										
IDENTIFY DEVICE	Device	✓			0x00	✓	✓					✓	9.15.3	11.4.2
RESET POWER CYCLE SEEN	Device	✓			0x01	✓	✓					✓	9.13.1	11.4.3
RESET	Device	✓			0x10	✓	✓					✓	9.12.1	11.5.2
RESET MEMORY BANK (DTR0)	Device	✓			0x11		✓					✓	9.12.2	11.5.3
SET SHORT ADDRESS (DTR0)	Device	✓			0x14	✓	✓					✓	9.15.1	11.5.4
ENABLE WRITE MEMORY	Device	✓			0x15	✓	✓					✓	9.11.6	11.5.5
ENABLE APPLICATION CONTROLLER	Device	✓			0x16	✓						✓	9.10.1	11.5.6
DISABLE APPLICATION CONTROLLER	Device	✓			0x17	✓						✓	9.10.1	11.5.7
SET OPERATING MODE (DTR0)	Device	✓			0x18	✓	✓					✓	9.10.5	11.5.8
ADD TO DEVICE GROUPS 0-15 (DTR2:DTR1)	Device	✓			0x19	✓	✓		✓	✓		✓		11.5.9
ADD TO DEVICE GROUPS 16-31 (DTR2:DTR1)	Device	✓			0x1A	✓	✓		✓	✓		✓		11.5.10
REMOVE FROM DEVICE GROUPS 0-15 (DTR2:DTR1)	Device	✓			0x1B	✓	✓		✓	✓		✓		11.5.11
REMOVE FROM DEVICE GROUPS 16-31 (DTR2:DTR1)	Device	✓			0x1C	✓	✓		✓	✓		✓		11.5.12
START QUIESCENT MODE	Device	✓			0x1D	✓	✓					✓	9.10.4	11.5.13
STOP QUIESCENT MODE	Device	✓			0x1E	✓	✓					✓	9.10.4	11.5.14
ENABLE POWER CYCLE NOTIFICATION	Device	✓			0x1F	✓	✓					✓	9.13.2	11.5.15
DISABLE POWER CYCLE NOTIFICATION	Device	✓			0x20	✓	✓					✓	9.13.2	11.5.16
Reserved ^a	Device	✓			0x21 ^a	✓	✓					✓		
Reserved for IEC 62386-104 (see [2])	Device	✓			0x22	✓		✓						

Command name	Address byte	Instance byte			Opcode byte	App Ctrl	Input device	DTR0	DTR1	DTR2	Answer	Send twice	References	Command subclause
		Device	Instance	Feature										
Reserved for IEC 62386-104 (see [2])	Device	✓			0x23	✓								
Reserved for IEC 62386-104 (see [2])	Device	✓			0x24	✓								
QUERY DEVICE STATUS	Device	✓			0x30	✓	✓				✓		9.17.2	11.6.3
QUERY APPLICATION CONTROLLER ERROR	Device	✓			0x31	✓					✓		9.16	11.6.4
QUERY INPUT DEVICE ERROR	Device	✓			0x32		✓				✓		9.16	11.6.5
QUERY MISSING SHORT ADDRESS	Device	✓			0x33		✓				✓			11.6.6
QUERY VERSION NUMBER	Device	✓			0x34	✓	✓				✓		4.2	11.6.7
QUERY NUMBER OF INSTANCES	Device	✓			0x35	✓	✓				✓		9.5	11.6.9
QUERY CONTENT DTR0	Device	✓			0x36	✓	✓	✓			✓			11.6.8
QUERY CONTENT DTR1	Device	✓			0x37	✓	✓		✓		✓			11.6.10
QUERY CONTENT DTR2	Device	✓			0x38	✓	✓			✓	✓			11.6.11
QUERY RANDOM ADDRESS (H)	Device	✓			0x39	✓	✓				✓			11.6.12
QUERY RANDOM ADDRESS (M)	Device	✓			0x3A	✓	✓				✓			11.6.13
QUERY RANDOM ADDRESS (L)	Device	✓			0x3B	✓	✓				✓			11.6.14
READ MEMORY LOCATION (DTRL, DTR0)	Device	✓			0x3C	✓	✓	✓	✓		✓		9.11.5	11.6.15
QUERY APPLICATION CONTROLLER ENABLED	Device	✓			0x3D	✓					✓		9.10.1	11.6.16
QUERY OPERATING MODE	Device	✓			0x3E	✓	✓				✓		9.10.5	11.6.17
QUERY MANUFACTURER SPECIFIC MODE	Device	✓			0x3F	✓	✓				✓		9.10.5	11.6.18
QUERY QUIESCENT MODE	Device	✓			0x40	✓	✓				✓		9.10.4	11.6.19
QUERY DEVICE GROUPS 0-7	Device	✓			0x41	✓	✓				✓			11.6.20
QUERY DEVICE GROUPS 8-15	Device	✓			0x42	✓	✓				✓			11.6.21

Command name	Address byte	Instance byte			Opcode byte	App Ctrl	Input device	DTR0	DTR1	DTR2	Answer	Send twice	References	Command subclause
		Device	Instance	Feature										
QUERY DEVICE GROUPS 16-23	Device	✓			0x43	✓	✓				✓			11.6.22
QUERY DEVICE GROUPS 24-31	Device	✓			0x44	✓	✓				✓			11.6.23
QUERY POWER CYCLE NOTIFICATION	Device	✓			0x45	✓	✓				✓		9.13.2	11.6.24
QUERY DEVICE CAPABILITIES	Device	✓			0x46	✓	✓				✓		9.17.1	11.6.2
QUERY EXTENDED VERSION NUMBER(<i>DTR0</i>)	Device	✓			0x47	✓	✓	✓			✓			11.6.25
QUERY RESET STATE	Device	✓			0x48	✓	✓				✓		9.17.2	11.6.26
QUERY APPLICATION CONTROLLER ALWAYS ACTIVE	Device	✓			0x49						✓		9.10.2	11.6.27
SET EVENT PRIORITY (<i>DTR0</i>)	Device	✓	✓		0x61		✓	✓				✓	9.14.2	11.8.8, 11.5.17
ENABLE INSTANCE	Device		✓		0x62		✓					✓	9.10.3	11.8.2
DISABLE INSTANCE	Device		✓		0x63		✓					✓	9.10.3	11.8.3
SET PRIMARY INSTANCE GROUP (<i>DTR0</i>)	Device				0x64		✓	✓				✓	9.5.5	11.8.4
SET INSTANCE GROUP 1 (<i>DTR0</i>)	Device		✓		0x65		✓	✓				✓	9.5.5	11.8.5
SET INSTANCE GROUP 2 (<i>DTR0</i>)	Device		✓		0x66		✓	✓				✓	9.5.5	11.8.6
SET EVENT SCHEME (<i>DTR0</i>)	Device		✓		0x67		✓	✓				✓	9.7.3	11.8.7
SET EVENT FILTER (<i>DTR2, DTR1, DTR0</i>)	Device		✓		0x68		✓	✓	✓	✓		✓	9.7.4	11.8.9
SET INSTANCE TYPE (<i>DTR0</i>)	Device		✓		0x69		✓	✓				✓	9.19	11.8.10
SET INSTANCE CONFIGURATION (<i>DTR0, DTR2:DTR1</i>)	Device		✓		0x6A		✓	✓	✓	✓		✓	9.19	11.8.11
QUERY INSTANCE TYPE	Device		✓		0x80		✓				✓		9.5.3	11.9.2
QUERY RESOLUTION	Device		✓		0x81		✓				✓		9.8.2	11.9.3
QUERY INSTANCE ERROR	Device		✓		0x82		✓				✓		9.16	11.9.4

Command name	Address byte	Instance byte			Opcode byte	App Ctrl	Input device	DTR0	DTR1	DTR2	Answer	Send twice	References	Command subclause
		Device	Instance	Feature										
QUERY INSTANCE STATUS	Device	✓	✓		0x83		✓				✓		9.17.3	11.9.5
QUERY EVENT PRIORITY	Device	✓	✓		0x84		✓				✓		9.14.2	11.9.13, 11.6.30
QUERY INSTANCE ENABLED	Device		✓		0x86		✓				✓		9.10.3	11.9.6
QUERY PRIMARY INSTANCE GROUP	Device	✓	✓		0x88		✓				✓		9.5.5	11.9.7
QUERY INSTANCE GROUP 1	Device		✓		0x89		✓				✓		9.5.5	11.9.8
QUERY INSTANCE GROUP 2	Device		✓		0x8A		✓				✓		9.5.5	11.9.9
QUERY EVENT SCHEME	Device		✓		0x8B		✓				✓		9.7.3	11.9.10
QUERY INPUT VALUE	Device		✓		0x8C		✓				✓		9.8.3	11.9.11
QUERY INPUT VALUE LATCH	Device		✓		0x8D		✓				✓		9.8.3	11.9.12
QUERY FEATURE TYPE	Device	✓			0x8E	✓	✓				✓		9.2, 9.5.4	11.9.14, 11.6.28
QUERY NEXT FEATURE TYPE	Device		✓		0x8F	✓	✓				✓		9.2, 9.5.4	11.9.15, 11.6.29
QUERY EVENT FILTER 0-7	Device		✓		0x90		✓				✓		9.7.4	11.9.16
QUERY EVENT FILTER 8-15	Device		✓		0x91		✓				✓		9.7.4	11.9.17
QUERY EVENT FILTER 16-23	Device		✓		0x92		✓				✓		9.7.4	11.9.18
QUERY INSTANCE CONFIGURATION (<i>DTR0</i>)	Device		✓		0x93		✓	✓		✓	✓		9.19	11.9.19
QUERY AVAILABLE INSTANCE TYPES	Device		✓		0x94		✓	✓		✓	✓		9.19	11.9.20

^a Reserved to maintain backward compatibility due to use in Edition 1 of IEC 62386-103:2014 (see [3]).

Table 24 – Special commands (implemented by both application controller and input device)

Command name	Address byte	Instance byte	Opcode byte	DTR0	DTR1	DTR2	Answer	Send twice	References	Command subclause
TERMINATE	0xC1	0x00	0x00							11.10.2
INITIALISE (<i>device</i>)	0xC1	0x01	<i>device</i>					✓	9.15	11.10.3
RANDOMISE	0xC1	0x02	0x00					✓	9.15	11.10.4
COMPARE	0xC1	0x03	0x00				✓		9.15	11.10.5
WITHDRAW	0xC1	0x04	0x00						9.15	11.10.6
SEARCHADDRH (<i>data</i>)	0xC1	0x05	<i>data</i>						9.15	11.10.7
SEARCHADDRM (<i>data</i>)	0xC1	0x06	<i>data</i>						9.15	11.10.8
SEARCHADDRL (<i>data</i>)	0xC1	0x07	<i>data</i>						9.15	11.10.9
PROGRAM SHORT ADDRESS (<i>data</i>)	0xC1	0x08	<i>data</i>						9.15	11.10.10
VERIFY SHORT ADDRESS (<i>data</i>)	0xC1	0x09	<i>data</i>				✓		9.15	11.10.11
QUERY SHORT ADDRESS	0xC1	0x0A	0x00				✓		9.15	11.10.12
Reserved for IEC 62386-104 (see [2])	0xC1	0x0B	<i>data</i>	✓			✓			
Reserved for IEC 62386-104 (see [2])	0xC1	0x0C	<i>data</i>							
Reserved for IEC 62386-104 (see [2])	0xC1	0x0D	<i>data</i>							
WRITE MEMORY LOCATION (<i>DTRL</i> , <i>DTR0</i> , <i>data</i>)	0xC1	0x20	<i>data</i>	✓	✓		✓		9.11.6	11.10.13
WRITE MEMORY LOCATION – NO REPLY (<i>DTRL</i> , <i>DTR0</i> , <i>data</i>)	0xC1	0x21	<i>data</i>	✓	✓				9.11.6	11.10.14
DTR0 (<i>data</i>)	0xC1	0x30	<i>data</i>	✓						11.10.15
DTR1 (<i>data</i>)	0xC1	0x31	<i>data</i>		✓					11.10.16
DTR2 (<i>data</i>)	0xC1	0x32	<i>data</i>			✓				11.10.17
SEND TESTFRAME (<i>data</i>)	0xC1	0x33	<i>data</i>	✓	✓	✓				11.10.21
DIRECT WRITE MEMORY (<i>DTRL</i> , <i>offset</i> , <i>data</i>)	0xC5	<i>offset</i>	<i>data</i>	✓	✓		✓		9.11.6	11.10.18
DTR1:DTR0 (<i>data1</i> , <i>data0</i>)	0xC7	<i>data1</i>	<i>data0</i>	✓	✓					11.10.19
DTR2:DTR1 (<i>data2</i> , <i>data1</i>)	0xC9	<i>data2</i>	<i>data1</i>		✓	✓				11.10.20

11.3 Event messages

11.3.1 INPUT NOTIFICATION (*device/instance, event*)

The event message notifies of a change or a series of changes of "*inputValue*" at the instance of an input device as required by this document or by the relevant part of the IEC 62386-3xx series corresponding to the "*instanceType*" of the instance.

The transmitting instance shall

- generate the event message only while "*instanceActive*" is TRUE;
- generate the event message only while it is not in an error condition that prevents operation (see 9.16);
- use the currently active "*eventScheme*";
- use the requested "*eventPriority*".

Refer to 9.7 and 9.8 for further information.

11.3.2 POWER NOTIFICATION (*device*)

The event notifies of a control device power cycle completion and shall be generated following the requirements as stated in 9.7.2 and 9.13.2.

11.4 Device control instructions

11.4.1 General

Device control instructions are used to modify property values of a control device. For this reason a device control instruction shall be discarded, unless it is accepted twice according to the requirements specified in IEC 62386-101:2022, 9.4.

Unless explicitly stated otherwise in the description of the particular device control instruction, the following holds:

- the instruction shall be ignored if so required by the provisions of 9.6;
- the control device shall not reply to the instruction;
- the instruction shall apply to device variables.

11.4.2 IDENTIFY DEVICE

The control device shall start or restart a $10\text{ s} \pm 1\text{ s}$ identification procedure which shall enable an observer to distinguish any control device(s) running this process from any devices (of the same type) which are not running it. On expiry of this timer, the identification procedure shall stop.

The identification shall be stopped immediately upon execution of any instruction other than "INITIALISE (*device*)" or "IDENTIFY DEVICE".

NOTE 1 Identification can be used during commissioning, allowing an installer to locate devices and allocate the particular identified device to a particular device group.

The indication can be done in various ways, such as by flashing an LED, by producing a sound, other visual or audible means, or alternative methods such as a wireless transmission to a smart device or tool. The exact process used to identify is manufacturer specific and should be described in the manual. In choosing the method, consideration should be given for the availability of the required tools for the intended lifetime of the product.

Support for IDENTIFY DEVICE is optional provided all of the following conditions are met:

- there is no controllable emitter that could be used for identification purposes (such as an LED, buzzer or wireless transmitter), and
- the device can issue an INPUT NOTIFICATION in response to a user generated trigger, such as a push-button or light sensor, and
- a statement is included in the product documents, "This control device does not support identification by means of an LED, buzzer or other emitter."

If any of these conditions are not met, support for IDENTIFY DEVICE shall be implemented.

For the case when IDENTIFY DEVICE is not implemented, it is recommended that the GTIN and serial (identification) number, as used in memory bank 0, are shown on the product label in decimal format. If serial numbers are formatted in hexadecimal, then it is recommended to prefix with "0x".

NOTE 2 Application controllers can support event messages from input devices, as a method of identifying a device during commissioning.

NOTE 3 The application controller can also stop the identification process using a "RESET" command.

Refer to 9.15.3 for further information.

11.4.3 RESET POWER CYCLE SEEN

"powerCycleSeen" shall be set to FALSE.

Refer to 9.13.1 for further information.

11.5 Device configuration instructions

11.5.1 General

Device configuration instructions are used to change the configuration and/or the mode of operation of the control device. For this reason a device configuration instruction shall be discarded, unless it is accepted twice according to the requirements specified in IEC 62386-101:2022, 9.4.

Unless explicitly stated otherwise in the description of the particular device configuration instruction, the following holds:

- the instruction shall be ignored if so required by the provisions of 9.6;
- the control device shall not reply to the instruction;
- the instruction shall apply to device variables.

11.5.2 RESET

All variables shall be changed to their reset values. Control devices shall start to react properly to commands no later than 300 ms after the execution of the instruction has started.

If during a reset mains power fails, it is not guaranteed that "RESET" is completed.

Refer to 9.12.1, Table 19 and Table 20 for further information.

11.5.3 RESET MEMORY BANK (*DTR0*)

The command shall trigger the process to change the memory bank content to its reset values as follows:

- if "*DTR0*" = 0: all implemented and unlocked memory banks except memory bank 0 shall be reset;
- in all other cases: the memory bank identified by "*DTR0*" shall be reset, provided it is implemented and unlocked.

A memory bank needs to be unlocked to allow both lockable and non-lockable locations to be reset.

The control device shall start to react properly to commands no later than 10 s after the execution of the instruction has started.

Refer to 9.12.2 for further information.

11.5.4 SET SHORT ADDRESS (*DTR0*)

The "*shortAddress*" shall be set to "*DTR0*".

The command shall be discarded if "*DTR0*" does not contain a valid "*shortAddress*" value.

Refer to 9.15.1 for further information.

11.5.5 ENABLE WRITE MEMORY

"*writeEnableState*" shall be set to ENABLED.

NOTE There is no command to explicitly disable memory write access, since any command that is not directly involved with writing into memory banks will reset "*writeEnableState*" back to DISABLED.

Refer to 9.11.6 for further information.

11.5.6 ENABLE APPLICATION CONTROLLER

If "*applicationControllerPresent*" is TRUE, "*applicationActive*" shall be set to TRUE, otherwise this command shall be discarded.

Refer to 9.10.1 for further information.

11.5.7 DISABLE APPLICATION CONTROLLER

If "*applicationControllerAlwaysActive*" is TRUE, this command shall be discarded.

If "*applicationControllerPresent*" is TRUE, "*applicationActive*" shall be set to FALSE, otherwise this command shall be discarded.

Refer to 9.10.1 and 9.10.2 for further information.

11.5.8 SET OPERATING MODE (*DTR0*)

"*operatingMode*" shall be set to "*DTR0*".

If "*DTR0*" does not correspond to an implemented operating mode, the command shall be discarded.

Refer to 9.10.5 for further information.

11.5.9 ADD TO DEVICE GROUPS 0-15 (*DTR2:DTR1*)

The control device shall set those bits in "*deviceGroups[15:0]*" that are set in [*"DTR2:DTR1"*]. The other bits shall not change.

11.5.10 ADD TO DEVICE GROUPS 16-31 (*DTR2:DTR1*)

The control device shall set those bits in "*deviceGroups[31:16]*" that are set in [*"DTR2:DTR1"*]. The other bits shall not change.

11.5.11 REMOVE FROM DEVICE GROUPS 0-15 (*DTR2:DTR1*)

The control device shall clear those bits in "*deviceGroups[15:0]*" that are set in [*"DTR2:DTR1"*]. The other bits shall not change.

11.5.12 REMOVE FROM DEVICE GROUPS 16-31 (*DTR2:DTR1*)

The control device shall clear those bits in "*deviceGroups[31:16]*" that are set in [*"DTR2:DTR1"*]. The other bits shall not change.

11.5.13 START QUIESCENT MODE

The control device shall start or restart quiescent mode by setting "*quiescentMode*" to ENABLED and (re-)triggering the timer.

Refer to 9.10.4 for further information.

11.5.14 STOP QUIESCENT MODE

"*quiescentMode*" shall be set to DISABLED.

Refer to 9.10.4 for further information.

11.5.15 ENABLE POWER CYCLE NOTIFICATION

"*powerCycleNotification*" shall be set to ENABLED.

Refer to 9.13.2 for further information.

11.5.16 DISABLE POWER CYCLE NOTIFICATION

"*powerCycleNotification*" shall be set to DISABLED.

Refer to 9.13.2 for further information.

11.5.17 SET EVENT PRIORITY (*DTR0*)

The command shall be discarded if "*DTR0*" is not in the range [2,5].

"*eventPriority*" shall be set to "*DTR0*".

11.6 Device queries

11.6.1 General

Device queries are used to retrieve device property values from a control device. The addressed control device returns the queried property value in a backward frame.

Unless explicitly stated otherwise in the description of the particular device query, the following holds:

- the query shall be ignored if so required by the provisions of 9.6;
- the query shall apply to device variables.

When applicable, the query shall be discarded if any of the parameter values (in "*DTR0*", "*DTR1*" and "*DTR2*") are outside the range of validity of the addressed device variables, as given in Table 19.

11.6.2 QUERY DEVICE CAPABILITIES

The answer shall be a combination of control device capabilities.

Refer to 9.17.1 for further information.

11.6.3 QUERY DEVICE STATUS

The answer shall be the status, which is formed by a combination of control device properties.

Refer to 9.17.2 for further information.

11.6.4 QUERY APPLICATION CONTROLLER ERROR

The answer shall be the detailed error information regarding an application controller:

- if an error in the application controller has occurred (as indicated by "*applicationControllerError*"), but the device is not able to give detailed error information: MASK;
- if an error in the application controller has occurred (as indicated by "*applicationControllerError*"), and the device is able to give detailed error information: error number [0,254];
- if no application controller error has occurred: NO.

Detailed error information is manufacturer specific and should be described in product documentation.

Refer to 9.16 for further information.

11.6.5 QUERY INPUT DEVICE ERROR

The answer shall be the detailed error information regarding an input device:

- if an error in the input device has occurred (as indicated by "*inputDeviceError*"), but the device is not able to give detailed error information: MASK;
- if an error in the input device has occurred (as indicated by "*inputDeviceError*"), and the device is able to give detailed error information: error number [0,254];
- if no input device error has occurred: NO.

Detailed error information is manufacturer specific and should be described in product documentation.

Refer to 9.16 for further information.

11.6.6 QUERY MISSING SHORT ADDRESS

The answer shall be YES if "*shortAddress*" is equal to MASK and NO otherwise.

NOTE Since the control device answers only if no short address is stored, the use of the command is useful only in broadcast mode or when device group addressing is used.

11.6.7 QUERY VERSION NUMBER

The answer shall be "*versionNumber*".

See 4.2 and Table 19 for more information.

11.6.8 QUERY CONTENT DTR0

The answer shall be "*DTR0*".

11.6.9 QUERY NUMBER OF INSTANCES

The answer shall be "*numberOfInstances*".

Refer to 9.5 for further information.

11.6.10 QUERY CONTENT DTR1

The answer shall be "*DTR1*".

11.6.11 QUERY CONTENT DTR2

The answer shall be "*DTR2*".

11.6.12 QUERY RANDOM ADDRESS (H)

The answer shall be "*randomAddress*[23:16]".

11.6.13 QUERY RANDOM ADDRESS (M)

The answer shall be "*randomAddress*[15:8]".

11.6.14 QUERY RANDOM ADDRESS (L)

The answer shall be "*randomAddress*[7:0]".

11.6.15 READ MEMORY LOCATION (*DTR1*, *DTR0*)

The query shall be discarded if the memory bank identified by "*DTR1*" is not implemented.

If executed, the answer shall be the content of the memory location identified by offset "*DTR0*" within memory bank "*DTR1*".

The control device shall answer NO if the addressed memory location is not implemented.

NOTE 1 This allows gaps in the memory bank implementation.

If the addressed offset is below location 0xFF in the bank, the control device shall increment "*DTR0*" by one.

NOTE 2 This allows efficient multi-byte reading within a transaction.

Refer to 9.11.5 for further information.

11.6.16 QUERY APPLICATION CONTROLLER ENABLED

The answer shall be YES if "*applicationActive*" is TRUE, NO otherwise.

Refer to 9.10.1 for further information.

11.6.17 QUERY OPERATING MODE

The answer shall be "*operatingMode*".

Refer to 9.10.5 for further information.

11.6.18 QUERY MANUFACTURER SPECIFIC MODE

The answer shall be YES when "*operatingMode*" is in the range [0x80,0xFF] and NO otherwise.

Refer to 9.10.5 for further information.

11.6.19 QUERY QUIESCENT MODE

The answer shall be YES if "*quiescentMode*" is ENABLED, and NO otherwise.

Refer to 9.10.4 for further information.

11.6.20 QUERY DEVICE GROUPS 0-7

The answer shall be "*deviceGroups*[7:0]".

11.6.21 QUERY DEVICE GROUPS 8-15

The answer shall be "*deviceGroups*[15:8]".

11.6.22 QUERY DEVICE GROUPS 16-23

The answer shall be "*deviceGroups*[23:16]".

11.6.23 QUERY DEVICE GROUPS 24-31

The answer shall be "*deviceGroups*[31:24]".

11.6.24 QUERY POWER CYCLE NOTIFICATION

The answer shall be YES if "*powerCycleNotification*" is ENABLED, and NO otherwise.

Refer to 9.13.2 for further information.

11.6.25 QUERY EXTENDED VERSION NUMBER(*DTR0*)

The answer shall be the version number of the relevant part of the IEC 62386-3xx series where xx is given by *DTR0*.

The answer shall be:

- if the Part 3xx given by *DTR0* is not implemented: NO;
- if the Part 3xx given by *DTR0* is implemented: the version number of the Part 3xx.

Refer to the relevant part of the IEC 62386-3xx series for further information.

11.6.26 QUERY RESET STATE

The answer shall be YES if "*resetState*" is TRUE, and NO otherwise.

Refer to 9.17.2 for further information.

11.6.27 QUERY APPLICATION CONTROLLER ALWAYS ACTIVE

The answer shall be YES if "*applicationControllerAlwaysActive*" is TRUE, and NO otherwise.

Refer to 9.10.2 for further information.

11.6.28 QUERY FEATURE TYPE

See 11.9.14 for command execution and 9.2 for information on device features.

11.6.29 QUERY NEXT FEATURE TYPE

See 11.9.15 for command execution and 9.2 for information on device features.

11.6.30 QUERY EVENT PRIORITY

See 11.9.13 for command execution.

11.7 Instance control instructions

Instance control instructions are used to modify property values of an instance of an input device.

Unless explicitly stated otherwise in the description of the particular instance control instruction, the following holds:

- the instruction shall be ignored if so required by the provisions of 9.6;
- the input device shall not reply to the instruction;
- the instruction shall apply to instance variables.

NOTE This document does not describe any instance control instructions. However, the above requirements do apply to instance instructions described in the relevant parts of the IEC 62386-3xx series.

11.8 Instance configuration instructions

11.8.1 General

Instance configuration commands are used to change the configuration and/or the mode of operation of an instance within the input device. For this reason an instance configuration instruction shall be discarded, unless it is accepted twice according to the requirements specified in IEC 62386-101:2022, 9.4.

Unless explicitly stated otherwise in the description of the particular instance configuration instruction, the following holds:

- the instruction shall be ignored if so required by the provisions of 9.6;
- the input device shall not reply to the instruction;
- the instruction shall apply to instance variables.

11.8.2 ENABLE INSTANCE

"instanceActive" shall be set to TRUE.

Refer to 9.10.3 for further information.

11.8.3 DISABLE INSTANCE

"instanceActive" shall be set to FALSE.

Refer to 9.10.3 for further information.

11.8.4 SET PRIMARY INSTANCE GROUP (*DTR0*)

The instance shall have the primary membership to an instance group assigned or removed, by setting *"instanceGroup0"* to *"DTR0"*.

The command shall be discarded if *"DTR0"* is not in the range [0,31] and different from MASK.

Refer to 9.5.5 for further information.

11.8.5 SET INSTANCE GROUP 1 (*DTR0*)

The instance shall have an additional membership to an instance group assigned or removed, by setting *"instanceGroup1"* to *"DTR0"*.

The command shall be discarded if *"DTR0"* is not in the range [0,31] and different from MASK.

Refer to 9.5.5 for further information.

11.8.6 SET INSTANCE GROUP 2 (*DTR0*)

The instance shall have an additional membership to an instance group assigned or removed, by setting *"instanceGroup2"* to *"DTR0"*.

The command shall be discarded if *"DTR0"* is not in the range [0,31] and different from MASK.

Refer to 9.5.5 for further information.

11.8.7 SET EVENT SCHEME (*DTR0*)

The instance shall, if the provisions identified in 9.7.3 allow so, apply a new event addressing scheme for subsequent "INPUT NOTIFICATION (*device/instance, event*)" events by setting *"eventScheme"* to *"DTR0"*.

NOTE Circumstances can dictate that the operation cannot be granted by the receiving instance, meaning that the answer to the corresponding "QUERY EVENT SCHEME" can be different from the event scheme requested here.

The command shall be discarded if *"DTR0"* is not in the range [0,4].

Refer to 9.7.3 for further information.

11.8.8 SET EVENT PRIORITY (*DTR0*)

The instance shall apply a new message priority for subsequent "INPUT NOTIFICATION (*device/instance, event*)" events by setting "*eventPriority*" to "*DTR0*".

The command shall be discarded if "*DTR0*" is not in the range [2,5].

Refer to 9.14 for further information.

11.8.9 SET EVENT FILTER (*DTR2:DTR1:DTR0*)

The instance shall set "*eventFilter*[23:0]" to ["*DTR2:DTR1:DTR0*"].

Refer to 9.7.4 for further information.

11.8.10 SET INSTANCE TYPE (*DTR0*)

The instance shall discard this command if any of the following conditions are true:

- configuration of the instance type is not supported by the instance, or
- "*DTR0*[7:5]" is non-zero, or
- "*DTR0*[4:0]" is not a supported instance type.

If executed, "*instanceType*" shall be set to "*DTR0*[4:0]".

NOTE Application controllers can query the instance configuration after execution of this command, to check which instance configuration was activated.

A change of "*instanceType*" shall cause the instance variables of the addressed instance to be set to their reset values, and can cause the device to re-start with all other variables being set to the power-on values. In the case the device re-starts, the system start-up timing shall be met (see IEC 62386-101:2022, 4.11.6).

11.8.11 SET INSTANCE CONFIGURATION (*DTR0, DTR2:DTR1*)

The instance shall discard this command if any of the following conditions are true:

- configuration of "*instanceConfiguration*[*DTR0*]" is not supported by the instance, or
- "*DTR0*" < 191, and points to a value of "*instanceConfiguration*[]" that is not defined in the corresponding part of the IEC 62386-3xx series, or
- "*DTR0*" is 191, and "*DTR2:DTR1*" is not 0x55CC.

NOTE 1 "*DTR0*" values in the range [192, 255] allow manufacturer-specific instance configuration.

If executed and "*DTR0*" is not 191, then "*DTR2:DTR1*" shall be written to "*instanceConfiguration*[*DTR0*]", with any unused bits of the 16-bit value discarded.

If "*DTR0*" is 191, and "*DTR2:DTR1*" is 0x55CC, then all implemented locations of "*instanceConfiguration*[]" shall be set to their factory default values.

A change of "*instanceConfiguration*[*DTR0*]" can cause the device to re-start with all variables being set to the power-on values. In the case the device re-starts, all further frames may be discarded until the re-start has completed. The system start-up timing shall be met (see IEC 62386-101:2022, 4.11.6).

NOTE 2 Application controllers can send query commands to determine if the device is ready to receive further frames, so allowing for the possibility of the device re-starting.

11.9 Instance queries

11.9.1 General

Instance queries are used to retrieve instance property values from an instance of an input device. The addressed input device returns the property value queried in a backward frame.

Unless explicitly stated otherwise in the description of the particular instance query, the following holds:

- the query shall be ignored if so required by the provisions in 9.6;
- if the query addresses multiple instances within the input device, the query shall be answered as if each instance is a logical device (see IEC 62386-101:2022, 9.6.2);
- the query shall apply to instance variables.

11.9.2 QUERY INSTANCE TYPE

The answer shall be *"instanceType"*.

Refer to 9.5.3 for further information.

11.9.3 QUERY RESOLUTION

The answer shall be *"resolution"*.

Refer to 9.8.2 for further information.

11.9.4 QUERY INSTANCE ERROR

The answer shall be detailed error information:

- if an error has occurred (as indicated by *"instanceError"*), but the instance is not able to give detailed error information: 0;
- if an error has occurred (as indicated by *"instanceError"*), and the instance is able to give detailed error information: error number [1,255];
- if no error has occurred: NO.

NOTE Detailed error information is instance type specific and is described in the IEC 62386-3xx series.

Refer to 9.16 for further information.

11.9.5 QUERY INSTANCE STATUS

The command queries the status of a combination of instance properties.

Refer to 9.17.3 for further information.

11.9.6 QUERY INSTANCE ENABLED

The answer shall be YES, if *"instanceActive"* is TRUE in at least one of the addressed instances, and NO otherwise.

Refer to 9.10.3 for further information.

11.9.7 QUERY PRIMARY INSTANCE GROUP

The answer shall be *"instanceGroup0"*.

Refer to 9.5.5 for further information.

11.9.8 QUERY INSTANCE GROUP 1

The answer shall be "*instanceGroup1*".

Refer to 9.5.5 for further information.

11.9.9 QUERY INSTANCE GROUP 2

The answer shall be "*instanceGroup2*".

Refer to 9.5.5 for further information.

11.9.10 QUERY EVENT SCHEME

The answer shall be "*eventScheme*".

Refer to 9.7.3 for further information.

11.9.11 QUERY INPUT VALUE

The instance shall latch a new "*inputValue*" and reply with the most significant byte of the latched input value.

If the query addresses multiple instances within the input device, it shall be discarded.

Refer to 9.8.3 for further information.

11.9.12 QUERY INPUT VALUE LATCH

The instance shall reply with the next byte from a latched "*inputValue*".

Following the least significant byte of the latched input value, the answer shall be NO, until a new "QUERY INPUT VALUE" has been executed.

If the query addresses multiple instances within the input device, it shall be discarded.

Refer to 9.8.3 for further information.

11.9.13 QUERY EVENT PRIORITY

The answer shall be "*eventPriority*".

Refer to 9.14 for further information.

11.9.14 QUERY FEATURE TYPE

The answer shall be:

- if no feature 3xx is implemented: 254;
- if one device/instance feature is supported: the device/instance feature number;
- if more than one device/instance feature is supported: MASK.

Refer to 9.5.4 for further information.

11.9.15 QUERY NEXT FEATURE TYPE

The answer shall be:

- if directly preceded by "QUERY FEATURE TYPE", and more than one device/instance feature is supported: the first and lowest device/instance feature number;
- if directly preceded by "QUERY NEXT FEATURE TYPE", and not all device/instance features have been reported: the next lowest device/instance feature number;
- if directly preceded by "QUERY NEXT FEATURE TYPE", and all feature types have been reported: 254;
- in all other cases: NO.

The sequence of commands shall only be accepted as long as they use the same combination of address byte and instance byte. Multi-master transmitters should send such sequences as a transaction.

Refer to 9.5.4 for further information.

11.9.16 QUERY EVENT FILTER 0-7

The answer shall be *"eventFilter[7:0]"*.

Refer to 9.7.4 for further information.

11.9.17 QUERY EVENT FILTER 8-15

The answer shall be *"eventFilter[15:8]"*.

Refer to 9.7.4 for further information.

11.9.18 QUERY EVENT FILTER 16-23

The answer shall be *"eventFilter[23:16]"*.

Refer to 9.7.4 for further information.

11.9.19 QUERY INSTANCE CONFIGURATION (*DTR0*)

If *"instanceConfiguration[DTR0]"* is not implemented, there shall be no reply, otherwise the answer shall be the least significant byte of *"instanceConfiguration[DTR0]"*, and *"DTR2:DTR1"* shall be set to *"instanceConfiguration[DTR0]"*, padded with zeros for unused bits.

If *"DTR0"* is 191, the reply shall be MASK and *"DTR2:DTR1"* shall be set to 0xFFFF if all other implemented values of *"instanceConfiguration[]"* are at the factory default values, otherwise the reply shall be 0xAA and *"DTR2:DTR1"* shall be set to 0AAAA.

EXAMPLE If *DTR0* = 200, and *instanceConfiguration[200]* only has possible values in the range [0, 15], then this query will result in *DTR1* and the answer being set to the corresponding value in the range [0, 15], and *DTR2* being set to 0.

Refer to 9.19 for further information.

11.9.20 QUERY AVAILABLE INSTANCE TYPES

The answer shall be a bit-field containing one bit for each instance type 0 to 31. For each available instance type, the corresponding bit shall be set to 1, otherwise the corresponding bit shall be 0. The 32 bits are encoded in the answer and in "DTR0" through "DTR2" as follows:

- answer: bits [7:0] representing instance types 7 to 0,
- "DTR0": bits [15:8] representing instance types 15 to 8,
- "DTR1": bits [23:16] representing instance types 23 to 16,
- "DTR2": bits [31:24] representing instance types 31 to 24.

Refer to 9.19 for further information.

11.10 Special commands

11.10.1 General

All special mode commands shall be interpreted as instructions unless explicitly stated otherwise.

11.10.2 TERMINATE

The following processes shall be terminated immediately upon execution of this instruction:

- Initialisation, "*initialisationState*" shall be set to DISABLED;
- Identification, as started by "IDENTIFY DEVICE"

The instruction could also terminate other processes as identified in the relevant parts of the IEC 62386-3xx series.

11.10.3 INITIALISE (*device*)

This instruction shall be discarded, unless it is accepted twice according to the requirements specified in IEC 62386-101:2022, 9.4.

Only devices matching the given *device* shall respond to the instruction, as indicated in Table 25:

Table 25 – Device addressing with "INITIALISE (*device*)"

<i>device</i>	Responsive device(s)
00AAAAAAb	Device(s) with " <i>shortAddress</i> " equal to 00AAAAAAb
01111111b	Devices with " <i>shortAddress</i> " equal to MASK
MASK	All devices
Other	None

The instruction shall start or prolong the initialisation state, by setting "*initialisationState*" to ENABLED if it was DISABLED and (re-)trigger the timer. There shall be no answer.

Refer to 9.15 for further information.

11.10.4 RANDOMISE

This instruction shall be discarded, unless it is accepted twice according to the requirements specified in IEC 62386-101:2022, 9.4.

The instruction shall be discarded if "*initialisationState*" is equal to DISABLED.

If executed, the instruction shall generate a random value for "*randomAddress*", in the range of [0x000000,0xFFFFFE] which shall be available within 100 ms for use.

If there are multiple logical units present and the instruction is executed using broadcast addressing, the generated random addresses within the bus unit shall be unique, i.e. every logical unit shall have a value of "*randomAddress*" that is not found in any of the other logical units contained in the bus unit.

There shall be no reply to this instruction.

Refer to 9.15 for further information.

11.10.5 COMPARE

The query shall be discarded unless "*initialisationState*" is ENABLED.

If executed, the control device shall answer:

- if "*randomAddress*" ≤ "*searchAddress*": YES;
- in all other cases: NO.

Refer to 9.15 for further information.

11.10.6 WITHDRAW

The instruction shall be discarded unless the following conditions hold:

- "*initialisationState*" is equal to ENABLED, and
- "*randomAddress*" is equal to "*searchAddress*".

If the instruction is executed, the control device shall change "*initialisationState*" to WITHDRAWN.

NOTE 1 Before withdrawing a control device, the application controller can assign it a short address using "PROGRAM SHORT ADDRESS (*data*)".

NOTE 2 The effect is that the control device is excluded from subsequent "COMPARE" operations, thus allowing the application controller to conduct a (binary) search operation across all devices until the "COMPARE" query leads to no answer (from any control device) on the bus.

Refer to 9.15 for further information.

11.10.7 SEARCHADDRH (*data*)

The instruction shall be discarded if "*initialisationState*" is equal to DISABLED.

If the instruction is executed, "*searchAddress*[23:16]" shall be set to the given *data*.

Refer to 9.15 for further information.

11.10.8 SEARCHADDRM (*data*)

The instruction shall be discarded if "*initialisationState*" is equal to DISABLED.

If the instruction is executed, "*searchAddress*[15:8]" shall be set to the given *data*.

Refer to 9.15 for further information.

11.10.9 SEARCHADDRL (*data*)

The instruction shall be discarded if "*initialisationState*" is equal to DISABLED.

If the instruction is executed, "*searchAddress*[7:0]" shall be set to the given *data*.

Refer to 9.15 for further information.

11.10.10 PROGRAM SHORT ADDRESS (*data*)

The instruction shall be discarded unless the following conditions hold:

- "*initialisationState*" is equal to ENABLED or WITHDRAWN;
- "*randomAddress*" is equal to "*searchAddress*";
- *data* contains a valid "*shortAddress*" value.

NOTE The format for *data* for PROGRAM SHORT ADDRESS (*data*) is different from the format used for the equivalent command in IEC 62386-102.

If the instruction is executed, the "*shortAddress*" shall be set to *data*.

Refer to 9.15 for further information.

11.10.11 VERIFY SHORT ADDRESS (*data*)

The query shall be discarded if "*initialisationState*" is equal to DISABLED.

If the query is executed, the answer shall be YES if "*shortAddress*" is equal to the given *data* (*data* given in 00AAAAAAb format), and NO otherwise.

Refer to 9.15 for further information.

11.10.12 QUERY SHORT ADDRESS

The query shall be discarded if:

- "*initialisationState*" is equal to DISABLED, or
- "*randomAddress*" is not equal to "*searchAddress*".

If the query is executed, the answer shall be "*shortAddress*".

Refer to 9.15 for further information.

11.10.13 WRITE MEMORY LOCATION (*DTR1*, *DTR0*, *data*)

The instruction shall be discarded if any of the following conditions hold:

- the addressed memory bank is not implemented, or
- "*writeEnableState*" is DISABLED.

NOTE 1 This operation is a broadcast operation. Selective control device addressing can be achieved by setting the write enable condition selectively.

If the instruction is executed, the control device shall write *data* into the memory location identified by offset "*DTR0*" within memory bank "*DTR1*" and return *data* as an answer.

NOTE 2 Simultaneous writing to multiple control devices will probably lead to framing errors because of colliding answers.

NOTE 3 The value that can be read from the memory bank location is not necessarily *data*.

If the selected memory bank location is

- not implemented, or
- above the last accessible memory location, or
- locked (see 9.11.2), or
- not writeable,

the answer to WRITE MEMORY LOCATION (*DTR1*, *DTR0*, *data*) shall be NO and no memory location shall be written to.

If the addressed location is below location 0xFF in the bank, the control device shall increment "*DTR0*" by one.

NOTE 4 This allows efficient multi-byte writing within a transaction.

Refer to 9.11.6 for further information.

11.10.14 WRITE MEMORY LOCATION – NO REPLY (*DTR1*, *DTR0*, *data*)

This instruction is identical to the "WRITE MEMORY LOCATION (*DTR1*, *DTR0*, *data*)" command (see 11.10.13), except that the receiving control device shall not reply to the command.

Refer to 9.11.6 for further information.

11.10.15 DTR0 (*data*)

"*DTR0*" shall be set to the given *data*.

11.10.16 DTR1 (*data*)

"*DTR1*" shall be set to the given *data*.

11.10.17 DTR2 (*data*)

"*DTR2*" shall be set to the given *data*.

11.10.18 DIRECT WRITE MEMORY (*DTR1*, *offset*, *data*)

The instruction shall be discarded if any of the following conditions hold:

- the memory bank identified by "*DTR1*" is not implemented, or
- "*writeEnableState*" is DISABLED.

NOTE This operation is a broadcast operation. Selective control device addressing can be achieved by setting the write enable condition selectively.

If the command is executed, the control device shall copy the value of *offset* to "*DTR0*", then execute WRITE MEMORY LOCATION (*DTR1*, *DTR0*, *data*).

Refer to 9.11.6 for further information.

11.10.19 DTR1:DTR0 (*data1*, *data0*)

"*DTR1*" shall be set to the given *data1*; "*DTR0*" shall be set to the given *data0*.

11.10.20 DTR2:DTR1 (*data2*, *data1*)

"DTR2" shall be set to the given *data2*; "DTR1" shall be set to the given *data1*.

11.10.21 SEND TESTFRAME (*data*)

data shall be interpreted as *data*(CTARRPPPb).

The instruction shall be executed unless any of the following conditions hold:

- Cb ≠ 0;
- PPPb > 5;
- PPPb < 1;
- Ab = 1 and "applicationControllerPresent" = FALSE.

If executed, and Ab = 0: a 24-bit forward frame shall be sent with the following content:

- Address byte: "DTR0";
- Instance byte: "DTR1";
- Opcode byte: "DTR2".

If executed, and Ab = 1: a 16-bit forward frame shall be sent with the following content:

- Address byte: "DTR0";
- Opcode byte: "DTR1".

As the command is not addressable, the command shall be executed once per bus unit independent of its number of instances and/or logical devices. This also implies that if at least one application controller is present, the 16-bit forward frames shall be sent.

The forward frame shall be sent using priority PPPb and shall then be repeated RRB times. If Tb = 1, the frames shall be sent in a transaction. If Tb=0, the repeated frames shall be sent using the priority set by PPPb.

This command is used to test collision detection for multi-master application controller and input devices. If a collision is detected while the test frame is transmitted, the collision shall be dealt with. In the case of a transaction, the entire transaction shall be retransmitted. In the case of repeating frames, only the frame(s) containing the collision(s) shall be repeated.

Bibliography

- [1] *GS1 General Specification*, latest version, available at: www.gs1.org
- [2] IEC 62386-104:2019, *Digital addressable lighting interface – Part 104: General requirements – Wireless and alternative wired system components*
- [3] IEC 62386-103:2014², *Digital addressable lighting interface – Part 103: General requirements – Control devices*

IECNORM.COM : Click to view the full PDF of IEC 62386-103:2022

² The first edition will be withdrawn and replaced upon publication of this second edition.

IECNORM.COM : Click to view the full PDF of IEC 62386-103:2022

SOMMAIRE

AVANT-PROPOS	77
INTRODUCTION	79
1 Domaine d'application	81
2 Références normatives	81
3 Termes et définitions	81
4 Généralités	85
4.1 Généralités	85
4.2 Numéro de version	85
5 Spécifications électriques	85
6 Alimentation électrique du bus	85
7 Structure du protocole de transmission	85
7.1 Généralités	85
7.2 Codage de trame en avant à 24 bits	86
7.2.1 Format de trames pour les instructions et requêtes	86
7.2.2 Format de trames pour les messages d'événement	88
8 Cadencement	89
9 Mode de fonctionnement	89
9.1 Généralités	89
9.2 Caractéristiques des dispositifs	89
9.3 Contrôleur d'application	89
9.3.1 Généralités	89
9.3.2 Contrôleur d'application à un seul maître	89
9.3.3 Contrôleur d'application à plusieurs maîtres	90
9.4 Dispositif d'entrée	90
9.5 Instances de dispositifs d'entrée	90
9.5.1 Généralités	90
9.5.2 Numéro d'instance	91
9.5.3 Type d'instance	91
9.5.4 Caractéristiques d'instance	91
9.5.5 Groupes d'instances	91
9.6 Commandes qui excluent les messages d'événement	92
9.6.1 Généralités	92
9.6.2 Commandes de dispositif	93
9.6.3 Commandes d'instance	93
9.6.4 Commandes de caractéristique	93
9.7 Messages d'événement	94
9.7.1 Réponse aux messages d'événement	94
9.7.2 Événement de cycle de mise sous tension de dispositif	94
9.7.3 Événement de notification d'entrée	94
9.7.4 Filtre de message d'événement	95
9.8 Signal d'entrée, valeur mesurée et "inputValue"	96
9.8.1 Généralités	96
9.8.2 Résolution d'entrée	96
9.8.3 Obtention de la valeur d'entrée	96
9.8.4 Notification des modifications	97

9.9	Défaillance système.....	97
9.10	Fonctionnement d'un dispositif de commande	98
9.10.1	Activer/désactiver le contrôleur d'application	98
9.10.2	Contrôleur d'application toujours actif	98
9.10.3	Activer/désactiver les messages d'événement	98
9.10.4	Mode repos	98
9.10.5	Modes de fonctionnement.....	99
9.11	Blocs de mémoire	100
9.11.1	Généralités.....	100
9.11.2	Carte de mémoire	101
9.11.3	Sélection d'un emplacement de bloc de mémoire.....	102
9.11.4	Emplacements de mémoire protégéables.....	102
9.11.5	Lecture dans le bloc de mémoire	102
9.11.6	Écriture dans le bloc de mémoire.....	104
9.11.7	Bloc de mémoire 0.....	105
9.11.8	Bloc de mémoire 1 (facultatif)	108
9.11.9	Blocs de mémoire spécifiques au fabricant	110
9.11.10	Blocs de mémoire réservés.....	110
9.12	Réinitialisation	110
9.12.1	Opération de réinitialisation	110
9.12.2	Opération de réinitialisation des blocs de mémoire	110
9.13	Comportement lors de la mise sous tension.....	110
9.13.1	Mise sous tension.....	110
9.13.2	Notification du cycle de mise sous tension.....	111
9.14	Utilisation prioritaire.....	111
9.14.1	Généralités.....	111
9.14.2	Priorité des notifications d'entrée.....	112
9.15	Attribution d'adresses courtes.....	112
9.15.1	Généralités.....	112
9.15.2	Affectation d'adresses aléatoires	112
9.15.3	Identification d'un dispositif.....	113
9.16	Traitement des exceptions	113
9.17	Informations de capacités et d'état du dispositif	113
9.17.1	Capacités du dispositif.....	113
9.17.2	État du dispositif.....	114
9.17.3	État d'instance.....	115
9.18	Mémoire non volatile.....	115
9.19	Types et configuration d'instances	115
9.20	Configuration actuelle de l'unité de bus.....	116
10	Déclaration des variables	116
11	Définition des commandes.....	118
11.1	Généralités	118
11.2	Fiches de vue d'ensemble.....	118
11.3	Messages d'événement.....	126
11.3.1	INPUT NOTIFICATION (<i>device/instance, event</i>).....	126
11.3.2	POWER NOTIFICATION (<i>device</i>)	126
11.4	Instructions relatives à la commande de dispositif.....	126
11.4.1	Généralités.....	126
11.4.2	IDENTIFY DEVICE	126

11.4.3	RESET POWER CYCLE SEEN	127
11.5	Instructions relatives à la configuration du dispositif	127
11.5.1	Généralités	127
11.5.2	RESET	127
11.5.3	RESET MEMORY BANK (<i>DTR0</i>)	128
11.5.4	SET SHORT ADDRESS (<i>DTR0</i>)	128
11.5.5	ENABLE WRITE MEMORY	128
11.5.6	ENABLE APPLICATION CONTROLLER	128
11.5.7	DISABLE APPLICATION CONTROLLER	128
11.5.8	SET OPERATING MODE (<i>DTR0</i>)	129
11.5.9	ADD TO DEVICE GROUPS 0-15 (<i>DTR2:DTR1</i>)	129
11.5.10	ADD TO DEVICE GROUPS 16-31 (<i>DTR2:DTR1</i>)	129
11.5.11	REMOVE FROM DEVICE GROUPS 0-15 (<i>DTR2:DTR1</i>)	129
11.5.12	REMOVE FROM DEVICE GROUPS 16-31 (<i>DTR2:DTR1</i>)	129
11.5.13	START QUIESCENT MODE	129
11.5.14	STOP QUIESCENT MODE	129
11.5.15	ENABLE POWER CYCLE NOTIFICATION	129
11.5.16	DISABLE POWER CYCLE NOTIFICATION	129
11.5.17	SET EVENT PRIORITY (<i>DTR0</i>)	129
11.6	Requêtes propres au dispositif	130
11.6.1	Généralités	130
11.6.2	QUERY DEVICE CAPABILITIES	130
11.6.3	QUERY DEVICE STATUS	130
11.6.4	QUERY APPLICATION CONTROLLER ERROR	130
11.6.5	QUERY INPUT DEVICE ERROR	130
11.6.6	QUERY MISSING SHORT ADDRESS	131
11.6.7	QUERY VERSION NUMBER	131
11.6.8	QUERY CONTENT <i>DTR0</i>	131
11.6.9	QUERY NUMBER OF INSTANCES	131
11.6.10	QUERY CONTENT <i>DTR1</i>	131
11.6.11	QUERY CONTENT <i>DTR2</i>	131
11.6.12	QUERY RANDOM ADDRESS (H)	131
11.6.13	QUERY RANDOM ADDRESS (M)	131
11.6.14	QUERY RANDOM ADDRESS (L)	131
11.6.15	READ MEMORY LOCATION (<i>DTR1</i> , <i>DTR0</i>)	131
11.6.16	QUERY APPLICATION CONTROLLER ENABLED	132
11.6.17	QUERY OPERATING MODE	132
11.6.18	QUERY MANUFACTURER SPECIFIC MODE	132
11.6.19	QUERY QUIESCENT MODE	132
11.6.20	QUERY DEVICE GROUPS 0-7	132
11.6.21	QUERY DEVICE GROUPS 8-15	132
11.6.22	QUERY DEVICE GROUPS 16-23	132
11.6.23	QUERY DEVICE GROUPS 24-31	132
11.6.24	QUERY POWER CYCLE NOTIFICATION	133
11.6.25	QUERY EXTENDED VERSION NUMBER(<i>DTR0</i>)	133
11.6.26	QUERY RESET STATE	133
11.6.27	QUERY APPLICATION CONTROLLER ALWAYS ACTIVE	133
11.6.28	QUERY FEATURE TYPE	133
11.6.29	QUERY NEXT FEATURE TYPE	133

11.6.30	QUERY EVENT PRIORITY	133
11.7	Instructions relatives à la commande d'instance.....	133
11.8	Instructions relatives à la configuration d'instance	134
11.8.1	Généralités	134
11.8.2	ENABLE INSTANCE	134
11.8.3	DISABLE INSTANCE	134
11.8.4	SET PRIMARY INSTANCE GROUP (<i>DTR0</i>)	134
11.8.5	SET INSTANCE GROUP 1 (<i>DTR0</i>)	134
11.8.6	SET INSTANCE GROUP 2 (<i>DTR0</i>)	134
11.8.7	SET EVENT SCHEME (<i>DTR0</i>)	135
11.8.8	SET EVENT PRIORITY (<i>DTR0</i>)	135
11.8.9	SET EVENT FILTER (<i>DTR2:DTR1:DTR0</i>)	135
11.8.10	SET INSTANCE TYPE (<i>DTR0</i>)	135
11.8.11	SET INSTANCE CONFIGURATION (<i>DTR0, DTR2:DTR1</i>)	135
11.9	Requêtes d'instance	136
11.9.1	Généralités	136
11.9.2	QUERY INSTANCE TYPE	136
11.9.3	QUERY RESOLUTION	136
11.9.4	QUERY INSTANCE ERROR	136
11.9.5	QUERY INSTANCE STATUS	137
11.9.6	QUERY INSTANCE ENABLED	137
11.9.7	QUERY PRIMARY INSTANCE GROUP	137
11.9.8	QUERY INSTANCE GROUP 1	137
11.9.9	QUERY INSTANCE GROUP 2	137
11.9.10	QUERY EVENT SCHEME	137
11.9.11	QUERY INPUT VALUE	137
11.9.12	QUERY INPUT VALUE LATCH	137
11.9.13	QUERY EVENT PRIORITY	138
11.9.14	QUERY FEATURE TYPE	138
11.9.15	QUERY NEXT FEATURE TYPE	138
11.9.16	QUERY EVENT FILTER 0-7	138
11.9.17	QUERY EVENT FILTER 8-15	138
11.9.18	QUERY EVENT FILTER 16-23	139
11.9.19	QUERY INSTANCE CONFIGURATION (<i>DTR0</i>)	139
11.9.20	QUERY AVAILABLE INSTANCE TYPES	139
11.10	Commandes spéciales	139
11.10.1	Généralités	139
11.10.2	TERMINATE	139
11.10.3	INITIALISE (<i>device</i>)	140
11.10.4	RANDOMISE	140
11.10.5	COMPARE	140
11.10.6	WITHDRAW	141
11.10.7	SEARCHADDRH (<i>data</i>)	141
11.10.8	SEARCHADDRM (<i>data</i>)	141
11.10.9	SEARCHADDRL (<i>data</i>)	141
11.10.10	PROGRAM SHORT ADDRESS (<i>data</i>)	141
11.10.11	VERIFY SHORT ADDRESS (<i>data</i>)	142
11.10.12	QUERY SHORT ADDRESS	142
11.10.13	WRITE MEMORY LOCATION (<i>DTR1, DTR0, data</i>)	142

11.10.14	WRITE MEMORY LOCATION – NO REPLY (<i>DTR1</i> , <i>DTR0</i> , <i>data</i>)	143
11.10.15	DTR0 (<i>data</i>)	143
11.10.16	DTR1 (<i>data</i>)	143
11.10.17	DTR2 (<i>data</i>)	143
11.10.18	DIRECT WRITE MEMORY (<i>DTR1</i> , <i>offset</i> , <i>data</i>)	143
11.10.19	DTR1:DTR0 (<i>data1</i> , <i>data0</i>).....	143
11.10.20	DTR2:DTR1 (<i>data2</i> , <i>data1</i>).....	143
11.10.21	SEND TESTFRAME (<i>data</i>)	143
Bibliographie.....		145

Figure 1 – Représentation graphique générale de l'IEC 62386	79
---	----

Tableau 1 – Codage de la trame de commande à 24 bits	86
Tableau 2 – Octet d'instance dans une trame de commande.....	86
Tableau 3 – Codage de la trame de message d'événement à 24 bits	88
Tableau 4 – Types d'instances.....	91
Tableau 5 – Types de caractéristiques.....	91
Tableau 6 – Variables de groupes d'instances	92
Tableau 7 – Information d'adresse de dispositif dans le cadre d'un événement de cycle de mise sous tension	94
Tableau 8 – Schémas d'adressage d'événements.....	95
Tableau 9 – Valeur mesurée ($\approx 50\%$) par rapport à la résolution et à " <i>inputValue</i> "	96
Tableau 10 – Exemple de séquence de requête pour lire une valeur d'entrée à 4 octets	97
Tableau 11 – Types de mémoires	100
Tableau 12 – Carte de mémoire de base des blocs de mémoire.....	101
Tableau 13 – Carte de la mémoire du bloc de mémoire 0.....	106
Tableau 14 – Carte de la mémoire du bloc de mémoire 1.....	108
Tableau 15 – Capacités du dispositif de commande.....	114
Tableau 16 – État du dispositif de commande.....	114
Tableau 17 – État d'instance.....	115
Tableau 18 – Configuration actuelle de l'unité de bus	116
Tableau 19 – Déclaration des variables de dispositif.....	117
Tableau 20 – Déclaration des variables d'instance.....	118
Tableau 21 – Messages d'événement d'instances	119
Tableau 22 – Messages d'événement de dispositif.....	119
Tableau 23 – Commandes normalisées.....	120
Tableau 24 – Commandes spéciales (mises en œuvre par le contrôleur d'application et le dispositif d'entrée).....	125
Tableau 25 – Adressage de dispositif avec "INITIALISE (<i>device</i>)"	140

COMMISSION ÉLECTROTECHNIQUE INTERNATIONALE

INTERFACE D'ÉCLAIRAGE ADRESSABLE NUMÉRIQUE –

**Partie 103: Exigences générales –
Dispositifs de commande**

AVANT-PROPOS

- 1) La Commission Électrotechnique Internationale (IEC) est une organisation mondiale de normalisation composée de l'ensemble des comités électrotechniques nationaux (Comités nationaux de l'IEC). L'IEC a pour objet de favoriser la coopération internationale pour toutes les questions de normalisation dans les domaines de l'électricité et de l'électronique. A cet effet, l'IEC – entre autres activités – publie des Normes internationales, des Spécifications techniques, des Rapports techniques, des Spécifications accessibles au public (PAS) et des Guides (ci-après dénommés "Publication(s) de l'IEC"). Leur élaboration est confiée à des comités d'études, aux travaux desquels tout Comité national intéressé par le sujet traité peut participer. Les organisations internationales, gouvernementales et non gouvernementales, en liaison avec l'IEC, participent également aux travaux. L'IEC collabore étroitement avec l'Organisation Internationale de Normalisation (ISO), selon des conditions fixées par accord entre les deux organisations.
- 2) Les décisions ou accords officiels de l'IEC concernant les questions techniques représentent, dans la mesure du possible, un accord international sur les sujets étudiés, étant donné que les Comités nationaux de l'IEC intéressés sont représentés dans chaque comité d'études.
- 3) Les Publications de l'IEC se présentent sous la forme de recommandations internationales et sont agréées comme telles par les Comités nationaux de l'IEC. Tous les efforts raisonnables sont entrepris afin que l'IEC s'assure de l'exactitude du contenu technique de ses publications; l'IEC ne peut pas être tenue responsable de l'éventuelle mauvaise utilisation ou interprétation qui en est faite par un quelconque utilisateur final.
- 4) Dans le but d'encourager l'uniformité internationale, les Comités nationaux de l'IEC s'engagent, dans toute la mesure possible, à appliquer de façon transparente les Publications de l'IEC dans leurs publications nationales et régionales. Toutes divergences entre toutes Publications de l'IEC et toutes publications nationales ou régionales correspondantes doivent être indiquées en termes clairs dans ces dernières.
- 5) L'IEC elle-même ne fournit aucune attestation de conformité. Des organismes de certification indépendants fournissent des services d'évaluation de conformité et, dans certains secteurs, accèdent aux marques de conformité de l'IEC. L'IEC n'est responsable d'aucun des services effectués par les organismes de certification indépendants.
- 6) Tous les utilisateurs doivent s'assurer qu'ils sont en possession de la dernière édition de cette publication.
- 7) Aucune responsabilité ne doit être imputée à l'IEC, à ses administrateurs, employés, auxiliaires ou mandataires, y compris ses experts particuliers et les membres de ses comités d'études et des Comités nationaux de l'IEC, pour tout préjudice causé en cas de dommages corporels et matériels, ou de tout autre dommage de quelque nature que ce soit, directe ou indirecte, ou pour supporter les coûts (y compris les frais de justice) et les dépenses découlant de la publication ou de l'utilisation de cette Publication de l'IEC ou de toute autre Publication de l'IEC, ou au crédit qui lui est accordé.
- 8) L'attention est attirée sur les références normatives citées dans cette publication. L'utilisation de publications référencées est obligatoire pour une application correcte de la présente publication.
- 9) L'attention est attirée sur le fait que certains des éléments de la présente Publication de l'IEC peuvent faire l'objet de droits de brevet. L'IEC ne saurait être tenue pour responsable de ne pas avoir identifié de tels droits de brevets.

L'IEC 62386-103 a été établie par le comité d'études 34 de l'IEC: Éclairage. Il s'agit d'une Norme internationale.

Cette deuxième édition annule et remplace la première édition parue en 2014 et l'Amendement 1:2018. Cette édition constitue une révision technique.

Cette édition inclut les modifications techniques majeures suivantes par rapport à l'édition précédente:

- a) le domaine d'application a été mis à jour;
- b) le mode repos a été mis à jour;
- c) la durée de sauvegarde de la mémoire non volatile (NVM) a été ajoutée, et la commande SAVE PERSISTENT VARIABLES a été supprimée;

- d) le bloc de mémoire 0 a été modifié, et les exigences communes pour les blocs de mémoire ont été ajoutées;
- e) IDENTIFY DEVICE a été mise à jour;
- f) le numéro de version a été modifié;
- g) la configuration de l'unité de bus a été ajoutée; et
- h) les types d'instances et leur configuration ont été ajoutés.

Le texte de cette Norme internationale est issu des documents suivants:

Projet	Rapport de vote
34/946/FDIS	34/990/RVD

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

La langue employée pour l'élaboration de cette Norme internationale est l'anglais.

Ce document a été rédigé selon les Directives ISO/IEC, Partie 2, il a été développé selon les Directives ISO/IEC, Partie 1 et les Directives ISO/IEC, Supplément IEC, disponibles sous www.iec.ch/members_experts/refdocs. Les principaux types de documents développés par l'IEC sont décrits plus en détail sous www.iec.ch/standardsdev/publications.

La présente Partie 103 de l'IEC 62386 est destinée à être utilisée avec la Partie 101, qui comporte les exigences générales relatives au type de produit adapté (système), et avec les Parties 3xx applicables (exigences particulières pour les dispositifs de commande) qui comportent des articles destinés à compléter ou modifier les articles correspondants de la Partie 101 et de la Partie 103, afin de spécifier les exigences applicables pour chaque type de produit.

Une liste de toutes les parties de la série IEC 62386, publiées sous le titre général *Interface d'éclairage adressable numérique*, se trouve sur le site web de l'IEC.

Le comité a décidé que le contenu de ce document ne sera pas modifié avant la date de stabilité indiquée sur le site web de l'IEC sous webstore.iec.ch dans les données relatives au document recherché. A cette date, le document sera

- reconduit,
- supprimé,
- remplacé par une édition révisée, ou
- amendé.

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

INTRODUCTION

L'IEC 62386 est composée de plusieurs parties, appelées séries. La série IEC 62386 spécifie un réseau de bus pour la commande par des signaux numériques des appareils d'éclairage électroniques. La série IEC 62386-1xx inclut les spécifications de base. La Partie 101 contient les exigences générales relatives aux composants de système, la Partie 102 complète ces informations avec les exigences générales relatives aux appareillages de commande et la Partie 103 complète ces informations avec les exigences générales relatives aux dispositifs de commande. La Partie 104 et la Partie 105 peuvent s'appliquer à l'appareillage de commande ou aux dispositifs de commande. La Partie 104 fournit les exigences relatives aux composants de système à connexion alternative ou sans fil. La Partie 105 décrit le transfert du microprogramme. La Partie 150 fournit les exigences concernant une alimentation électrique auxiliaire qui peut être autonome ou intégrée aux appareillages de commande ou aux dispositifs de commande.

La série IEC 62386-2xx étend les exigences générales relatives aux appareillages de commande aux extensions spécifiques aux lampes (principalement pour la rétrocompatibilité avec l'Edition 1 de l'IEC 62386) et aux caractéristiques spécifiques aux appareillages de commande.

La série IEC 62386-3xx étend les exigences générales relatives aux dispositifs de commande aux extensions spécifiques aux dispositifs d'entrée qui décrivent les types d'instances ainsi que certaines caractéristiques communes qui peuvent être combinées à plusieurs types d'instances.

Cette deuxième édition de l'IEC 62386-103 est destinée à être utilisée conjointement avec l'IEC 62386-101 et avec les différentes parties qui composent la série IEC 62386-3xx qui spécifie les exigences particulières relatives aux dispositifs de commande, et peut être utilisée conjointement avec l'IEC 62386-102 et avec les différentes parties qui composent la série IEC 62386-2xx relative aux appareillages de commande. La présentation en parties publiées séparément facilitera les futurs amendements et révisions. Des exigences supplémentaires seront ajoutées en fonction des besoins identifiés.

La structure des normes est représentée sous forme de graphique à la Figure 1 ci-dessous.

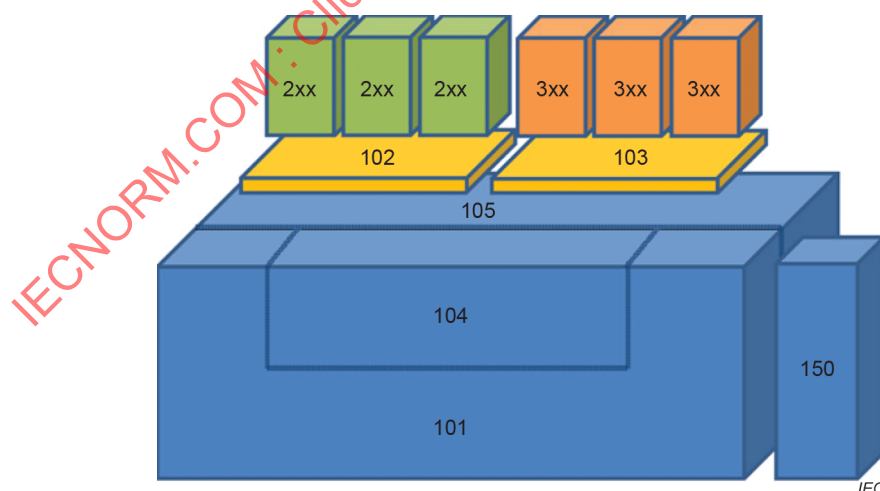


Figure 1 – Représentation graphique générale de l'IEC 62386

La présente partie de l'IEC 62386, tout en faisant référence à un article quelconque des autres parties de la série IEC 62386-1xx, spécifie la mesure dans laquelle un article s'applique. Les autres parties contiennent également des exigences supplémentaires, s'il y a lieu.

Tous les nombres utilisés dans le présent document sont des nombres décimaux, sauf indication contraire. Les nombres hexadécimaux sont donnés dans le format 0xVV, où VV est la valeur. Les nombres binaires sont donnés dans le format XXXXXXb ou dans le format XXXX XXXX, où X est 0 ou 1; "x" dans les nombres binaires signifie que "la valeur n'a pas d'influence".

Les expressions typographiques suivantes sont utilisées:

Variables: *variableName* ou *variableName[3:0]*, qui donne uniquement les bits 3 à 0 de *variableName*;

Plage de valeurs: [valeur minimale, valeur maximale];

Commande: "NOM DE LA COMMANDE".

IECNORM.COM : Click to view the full PDF of IEC 62386-103:2022

INTERFACE D'ÉCLAIRAGE ADRESSABLE NUMÉRIQUE –

Partie 103: Exigences générales – Dispositifs de commande

1 Domaine d'application

La présente partie de l'IEC 62386 s'applique aux dispositifs de commande par signaux numériques des équipements d'éclairage électroniques.

2 Références normatives

Les documents suivants sont cités dans le texte de sorte qu'ils constituent, pour tout ou partie de leur contenu, des exigences du présent document. Pour les références datées, seule l'édition citée s'applique. Pour les références non datées, la dernière édition du document de référence s'applique (y compris les éventuels amendements).

IEC 62386-101:2022, *Interface d'éclairage adressable numérique – Partie 101: Exigences générales – Composants de système*

IEC 62386-102:2022, *Interface d'éclairage adressable numérique – Partie 102: Exigences générales – Appareillages de commande*

IEC 62386-3xx (toutes les parties), *Interface d'éclairage adressable numérique – Partie 3xx: Exigences particulières pour les dispositifs de commande*

3 Termes et définitions

Pour les besoins du présent document, les termes et définitions de l'IEC 62386-101 ainsi que les suivants s'appliquent.

L'ISO et l'IEC tiennent à jour des bases de données terminologiques destinées à être utilisées en normalisation, consultables aux adresses suivantes:

- IEC Electropedia: disponible à l'adresse <http://www.electropedia.org/>
- ISO Online browsing platform: disponible à l'adresse <http://www.iso.org/obp>

3.1

diffusion

type d'adresse utilisé pour adresser simultanément l'ensemble des dispositifs de commande dans le système

3.2

diffusion non adressée

type d'adresse utilisé pour adresser simultanément l'ensemble des dispositifs de commande dans le système qui n'ont pas d'adresse courte

3.3

commande de dispositif

commande d'adressage du dispositif de commande qui comporte une valeur de 0xFE dans l'octet d'instance de la trame de commande, mais qui n'est pas un message d'événement

3.4

groupe de dispositifs

type d'adresse utilisé pour adresser simultanément un groupe de dispositifs de commande dans le système

3.5

DTR

registre de transfert des données

registre polyvalent utilisé pour l'échange de données

Note 1 à l'article: L'abréviation "DTR" est dérivée du terme anglais développé correspondant "data transfer register".

3.6

événement

rapport d'instance, caractérisé par son nombre d'événements, d'une modification ou d'une séquence définie de modifications de sa valeur d'entrée

Note 1 à l'article: Le nombre d'événements est spécifique au type de l'instance qui envoie le rapport.

3.7

schéma d'événement

caractérisation de l'information, telle que fournie par une instance lors de la production d'un message d'événement, qui identifie la source de l'événement

3.8

caractéristique

extension facultative au niveau de l'instance et/ou du dispositif

3.9

commande de caractéristique

commande d'adressage d'une ou de plusieurs caractéristiques d'un dispositif d'entrée ou d'une instance de dispositif qui comporte une valeur différente de 0xFE dans l'octet d'instance de la trame de commande, mais qui n'est pas une commande d'instance ou un message d'événement

3.10

GTIN

code article international

numéro utilisé pour identifier de façon unique, partout dans le monde, les articles commercialisés

Note 1 à l'article: Pour plus d'informations, voir <http://en.wikipedia.org/wiki/GTIN>

Note 2 à l'article: Le code article international est composé d'un préfixe de société GS1 ou CUP, suivi d'un numéro de référence d'article et d'un chiffre de contrôle. Il est décrit dans les "GS1 General Specifications" ("Spécifications générales GS1") (voir [1]).

Note 3 à l'article: L'abréviation "GTIN" est dérivée du terme anglais développé correspondant "global trade item number".

3.11

signal d'entrée

valeur physique qu'une instance d'un dispositif d'entrée est conçue pour détecter et traiter

Note 1 à l'article: Les exemples de valeurs physiques sont "illuminance" (éclairage lumineux) et "button state" (état du bouton).

3.12

identification

état provisoire utilisé pendant la mise en service qui permet à l'installateur d'identifier des dispositifs de commande particuliers

3.13**valeur d'entrée**

donnée codée, qui représente le signal d'entrée

Note 1 à l'article: La méthode de codage du signal d'entrée dépend du type d'instance.

3.14**commande d'instance**

commande d'adressage d'une ou de plusieurs instances d'un dispositif d'entrée qui comporte une valeur différente de 0xFE dans l'octet d'instance de la trame de commande, mais qui n'est pas une commande de caractéristique ou un message d'événement

3.15**MASK**

valeur dont tous les chiffres binaires sont définis sur 1

Note 1 à l'article: Cela signifie qu'une trame en arrière de 8 bits de MASK correspond à une valeur de 0xFF, et qu'un emplacement de mémoire à plusieurs octets de 24 bits qui contient MASK correspond à une valeur de 0xFFFFF.

3.16**NO**

réponse à une requête dans laquelle aucune trame en arrière n'est envoyée

Note 1 à l'article: Si une requête est formulée pour laquelle la réponse est NO, il n'y a pas de réponse, de sorte que l'émetteur de la requête conclue "pas de trame en arrière" suivant 8.2.5 de l'IEC 62386-101:2022.

Note 2 à l'article: Une absence de requête peut également déclencher la réponse NO.

3.17**NVM****mémoire non volatile**

mémoire en lecture/écriture non volatile dont le contenu peut être modifié et n'est pas perdu en raison d'un cycle de mise sous tension

Note 1 à l'article: L'abréviation "NMV" est dérivée du terme anglais développé correspondant "non-volatile memory".

3.18**NVM-RO**

NVM qui ne peut faire l'objet d'une saisie à l'aide d'aucune commande

3.19**NVM-RW**

NVM qui peut être modifiée à l'aide d'une ou de plusieurs commandes

3.20**opcode****code de fonctionnement**

partie d'une trame de commande qui identifie la commande à exécuter

Note 1 à l'article: L'abréviation "opcode" est dérivée du terme anglais développé correspondant "operation code".

3.21

mode de fonctionnement

ensemble d'états identifiés par un nombre compris dans la plage [0,255], caractérisé par un groupe de variables et de paramètres de mémoire, et utilisé pour sélectionner un ensemble de fonctionnalités à présenter par un dispositif de commande, y compris sa réaction nécessaire aux commandes

Note 1 à l'article: Les dispositifs de commande peuvent prendre en charge plusieurs modes de fonctionnement.

3.22

PING

trame en avant à 16 bits dont les bits [15:0] équivalent à 0xAD00

Note 1 à l'article: Conformément à l'IEC 62386-102, PING n'a aucune signification pour les appareillages de commande.

3.23

mode repos

mode provisoire dans lequel le dispositif n'envoie pas de trames en avant

3.24

RAM

mémoire en lecture/écriture volatile dont le contenu peut être modifié et est perdu en raison d'un cycle de mise sous tension

3.25

RAM-RO

RAM qui ne peut faire l'objet d'une saisie à l'aide d'aucune commande

3.26

RAM-RW

RAM qui peut être modifiée à l'aide d'une ou de plusieurs commandes

3.27

adresse aléatoire

numéro à 24 bits aléatoire généré par le dispositif de commande sur demande au cours de l'initialisation du système

3.28

état réinitialisé

état dans lequel toutes les variables NVM du dispositif de commande présentent des valeurs réinitialisées, sauf celles qui portent le marquage "pas de modification" ou qui sont explicitement exclues de toute autre manière

3.29

ROM

mémoire en lecture seule non volatile dont le contenu est fixe

Note 1 à l'article: Dans le présent document, le terme "en lecture seule" est défini par rapport au système. Une variable ROM peut effectivement être mise en œuvre dans la NVM, mais le présent document ne prévoit aucun mécanisme de modification de sa valeur.

3.30

adresse de recherche

nombre à 24 bits utilisé pour identifier un dispositif de commande particulier dans le système au cours de l'initialisation

3.31

adresse courte

type d'adresse utilisé pour adresser un dispositif de commande particulier dans le système

3.32

TMASK

valeur dont le bit de poids faible est défini sur 0 et dont tous les autres chiffres binaires sont définis sur 1

Note 1 à l'article: Cela signifie qu'une trame en arrière de 8 bits de TMASK correspond à une valeur de 0xFE, et qu'un emplacement de mémoire à plusieurs octets de 24 bits qui contient TMASK correspond à une valeur de 0xFFFFFE.

3.33

YES

réponse à une requête dans laquelle une trame en arrière de MASK est envoyée

4 Généralités

4.1 Généralités

Les exigences l'Article 4 de l'IEC 62386-101:2022 s'appliquent, avec les restrictions, modifications et ajouts identifiés ci-dessous.

4.2 Numéro de version

Le présent paragraphe remplace le 4.2 de l'IEC 62386-101:2022.

La version doit se présenter sous le format "x.y", où le numéro de version principale x se situe dans la plage comprise entre 0 et 62 et le numéro de version secondaire y se situe dans la plage comprise entre 0 et 2. Lorsque le numéro de version est codé par octet, le numéro de version principale x doit se situer dans les bits 7 à 2 et le numéro de version secondaire y doit se situer dans les bits 1 à 0.

A chaque amendement d'une édition de l'IEC 62386-103, le numéro de version secondaire doit être augmenté de un.

Lors d'une nouvelle édition de l'IEC 62386-103, le numéro de version principale doit être augmenté de un et le numéro de version secondaire doit être égal à 0.

Le numéro de version actuel est "*versionNumber*" comme cela est défini dans le Tableau 19.

NOTE Les documents IEC font généralement l'objet de deux amendements avant toute nouvelle édition.

5 Spécifications électriques

Les exigences de l'Article 5 de l'IEC 62386-101:2022 s'appliquent.

6 Alimentation électrique du bus

Lorsqu'une alimentation de bus est intégrée à un dispositif de commande, les exigences de l'Article 6 de l'IEC 62386-101:2022 s'appliquent.

7 Structure du protocole de transmission

7.1 Généralités

Les exigences de l'Article 7 de l'IEC 62386-101:2022 s'appliquent avec les ajouts suivants.

7.2 Codage de trame en avant à 24 bits

7.2.1 Format de trames pour les instructions et requêtes

7.2.1.1 Généralités

Pour le 7.2.1, les commandes doivent être interprétées comme des instructions et des requêtes, et excluent les messages d'événement. La trame en avant à 24 bits doit être codée comme cela est indiqué dans le Tableau 1 et le Tableau 2.

Tableau 1 – Codage de la trame de commande à 24 bits

Octets/Bits								Adressage des dispositifs		
Octet d'adresse									Octet d'instance	Octet de code de fonctionnement
23	22	21	20	19	18	17	16	15...8	7...0	
0	64 adresses courtes						1	Dispositif, instance ou caractéristique, voir Tableau 2		Adressage court
1	0	32 adresses de groupes de dispositifs					1			Adressage de groupes de dispositifs
1	1	1	1	1	1	0	1			Diffusion non adressée
1	1	1	1	1	1	1	1			Diffusion
1	1	0	16 espaces de commandes spéciales				1	Spécifique à une commande		Commande spéciale
1	1	1	0	x	x	x	1	Réservé		Réservé
1	1	1	1	0	x	x	1			
1	1	1	1	1	0	x	1			

Tableau 2 – Octet d'instance dans une trame de commande

Octet d'instance								Adressage
15	14	13	12	11	10	09	08	
0	0	0	32 numéros d'instances					Numéro d'instance
1	0	0	32 groupes d'instances					Groupe d'instances
1	1	0	32 types d'instances					Type d'instance
0	0	1	32 numéros d'instances					Caractéristique au niveau du numéro d'instance
1	0	1	32 groupes d'instances					Caractéristique au niveau du groupe d'instances
0	1	1	32 types d'instances					Caractéristique au niveau du type d'instance
1	1	1	1	1	0	0	1	Diffusion de caractéristique
1	1	1	1	1	1	0	1	Caractéristique au niveau de la diffusion d'instance
1	1	1	1	1	1	1	1	Diffusion d'instance
1	1	1	1	1	1	0	0	Caractéristique au niveau du dispositif
1	1	1	1	1	1	1	0	Dispositif
0	1	0	x	x	x	x	x	Réservé
1	1	1	0	x	x	x	x	
1	1	1	1	0	x	x	x	
1	1	1	1	1	0	1	x	
1	1	1	1	1	0	0	0	

7.2.1.2 Octet d'adresse

L'octet d'adresse fournit

- la méthode d'adressage des dispositifs utilisée par l'émetteur;
- l'indication selon laquelle une commande, et non un message d'événement, est en cours de transmission: le bit 16 est défini pour les commandes;
- 16 espaces de commandes spéciales;
- les adresses de dispositif réservées. L'émetteur ne doit pas utiliser les adresses réservées.

7.2.1.3 Octet d'instance

L'octet d'instance fournit

- pour les commandes normalisées, l'indication selon laquelle une commande de dispositif, de caractéristique ou d'instance est en cours de transmission;
- pour les commandes d'instance normalisées, la méthode d'adressage d'instance utilisée par l'émetteur;
- l'information spécifique à la commande pour les commandes spéciales;
- pour les commandes normalisées, les adresses d'instance réservées. L'émetteur ne doit pas utiliser les adresses d'instance réservées;
- pour les commandes de caractéristique normalisées, la caractéristique qui est adressée;
- l'information réservée pour les commandes réservées.

7.2.1.4 Octet de code de fonctionnement

L'octet de code de fonctionnement fournit

- pour les commandes normalisées, le code de fonctionnement;
- l'information spécifique à la commande pour les commandes spéciales;
- l'information réservée pour les commandes réservées.

7.2.2 Format de trames pour les messages d'événement

7.2.2.1 Généralités

Pour les messages d'événement, la trame en avant à 24 bits doit être codée comme cela est indiqué dans le Tableau 3.

Tableau 3 – Codage de la trame de message d'événement à 24 bits

Bits															Schéma ^a / source d'événement		
Information de source d'événement														Information d'événement			
23	22	21	20	19	18	17	16	15	14	13	12	11	10	9...0			
0	64 adresses courtes						0	0	32 types d'instances						Événement	1	Dispositif
0	64 adresses courtes						0	1	32 numéros d'instances							2	Dispositif et instance
1	0	32 groupes de dispositifs					0	0	32 types d'instances							3	Groupe de dispositifs
1	0	32 types d'instances					0	1	32 numéros d'instances							0	Instance
1	1	32 groupes d'instances					0	0	32 types d'instances							4	Groupe d'instances
1	1	0	x	x	x	x	0	1	x	x	x	x	x	Réservé	Réservé		
1	1	1	0	x	x	x	0	1	x	x	x	x	x				
1	1	1	1	0	x	x	0	1	x	x	x	x	x				
1	1	1	1	1	0	x	0	1	x	x	x	x	x				
1	1	1	1	!	1	0	0	1	x	x	x	x	x				
1	1	1	!	1	1	1	0	1	0	x	x	x	x				
1	1	1	1	1	1	1	0	1	1	0	x	x	x				
1	1	1	1	1	1	1	0	1	1	1	Information d'adresse courte et de groupe de dispositifs, se référer au 9.7.2.			Cycle de mise sous tension de dispositif			
^a Se référer au 9.7.3 pour plus informations sur les schémas d'événements.																	

^a Se référer au 9.7.3 pour plus informations sur les schémas d'événements.

7.2.2.2 Information de source d'événement

L'information de source d'événement du Tableau 3 fournit:

- l'indication selon laquelle un message d'événement, et non une instruction ou une requête, est en cours de transmission: le bit 16 doit être supprimé pour les messages d'événement;
- l'information pertinente propre au type d'instance d'événement, de sorte que le récepteur d'un message d'événement soit capable de comprendre la signification de l'événement;
- l'information pertinente propre à la source d'événement, de sorte que le récepteur d'un message d'événement est capable de comprendre d'où vient le message;
- les valeurs réservées.

Les événements sont spécifiques au type d'instance. Les contrôleurs d'application peuvent déduire, de manière explicite ou implicite, le type d'instance de l'instance de transmission, à partir de l'information de source d'événement.

NOTE Les schémas de source d'événement n'ont pas la même valeur en ce qui concerne l'indication au récepteur de la provenance du message d'événement.

7.2.2.3 Information d'événement

L'information d'événement fournit le nombre d'événements et/ou les données d'événement à 10 bits. L'information d'événement est spécifique au type d'instance et est définie dans les parties applicables de la série IEC 62386-3xx qui décrivent le type d'instance.

8 Cadencement

Les exigences de l'Article 8 de l'IEC 62386-101:2022 s'appliquent.

9 Mode de fonctionnement

9.1 Généralités

Les exigences de l'Article 9 de l'IEC 62386-101:2022 s'appliquent avec les ajouts suivants.

9.2 Caractéristiques des dispositifs

La série IEC 62386 autorise la publication future d'extensions de caractéristiques qui élargissent les exigences du présent document, ou exonèrent des exigences particulières.

Les caractéristiques d'un dispositif peuvent faire l'objet d'une requête en utilisant "QUERY FEATURE TYPE" et "QUERY NEXT FEATURE TYPE" avec l'octet d'instance réglé sur "caractéristique au niveau du dispositif" (voir Tableau 2).

Le Tableau 5 montre le codage de types de caractéristiques. Pour plus d'informations sur les différents types de caractéristiques, voir les parties de la série IEC 62386-3xx.

9.3 Contrôleur d'application

9.3.1 Généralités

Un contrôleur d'application est la partie d'un système de commande qui fait "fonctionner" le système:

- un contrôleur d'application peut mettre en service et configurer le système (y compris l'appareillage de commande disponible);
- un contrôleur d'application peut faire réagir le système aux changements qui interviennent dans l'environnement (d'après les informations en provenance des dispositifs d'entrée);
- un contrôleur d'application peut modifier la réponse du système de l'appareillage de commande dans le système (éventuellement en utilisant une commande définie dans l'IEC 62386-102:2022).

9.3.2 Contrôleur d'application à un seul maître

Un contrôleur d'application à un seul maître n'est pas destiné à partager le bus avec d'autres dispositifs de commande.

Un contrôleur d'application à un seul maître peut tenter de configurer d'autres dispositifs de commande sur le bus, et/ou de modifier la réponse du système de l'appareillage de commande dans le système, en utilisant ainsi toute commande définie dans l'IEC 62386-102 et/ou les instructions et requêtes définies dans le présent document.

NOTE Ce type de tentative peut échouer et affecter le système de manière négative, notamment si le contrôleur d'application à un seul maître ne traite pas les collisions de façon appropriée.

D'autre part, il n'est pas nécessaire qu'un contrôleur d'application à un seul maître comporte un récepteur embarqué. Pour cette raison, le principe suivant s'applique:

À partir du 9.3.3, le présent document prend pour hypothèse qu'un dispositif de commande est un dispositif de commande à plusieurs maîtres.

Afin de se faire reconnaître comme unité de bus de transmission éventuellement anonyme, un contrôleur d'application à un seul maître doit émettre un message PING à intervalles réguliers d'une durée de $10 \text{ min} \pm 1 \text{ min}$. Le premier message PING de ce type doit apparaître à un moment aléatoire compris entre 5 min et 10 min après réalisation de la procédure de mise sous tension.

9.3.3 Contrôleur d'application à plusieurs maîtres

À partir du 9.3.3, le présent document prend pour hypothèse qu'un dispositif de commande est un dispositif de commande à plusieurs maîtres.

Un dispositif de commande qui comprend un contrôleur d'application doit avoir *"applicationControllerPresent"* réglé sur TRUE. *"applicationControllerPresent"* doit être réglé sur FALSE dans le cas contraire.

NOTE 1 *"applicationControllerPresent"* peut être observé par "QUERY DEVICE CAPABILITIES".

Dans la plupart des cas, un système ne comporte qu'un seul contrôleur d'application actif (voir 9.10.1), mais plusieurs contrôleurs d'application peuvent être opérationnels dans un même système.

Un contrôleur d'application doit accepter des commandes (d'autres contrôleurs d'application) selon le Tableau 23 et le Tableau 24. Le fait de s'assurer que les contrôleurs d'application acceptent les commandes de sorte qu'un système fonctionne effectivement correctement fait partie de l'intégration du système.

NOTE 2 L'intégrité du système est plus facile à réaliser en permettant uniquement à un seul contrôleur d'application d'effectuer la mise en service et la configuration.

NOTE 3 Un contrôleur d'application peut être mis en service par des interfaces alternatives.

Un contrôleur d'application ne doit pas transmettre de messages d'événement autres que pour l'événement de cycle de mise sous tension de dispositif.

NOTE 4 Lorsqu'un contrôleur d'application est actif, il peut envoyer des trames en avant à 24 bits à des fins autres que la transmission d'événements.

Un contrôleur d'application ne doit pas transmettre de messages PING.

9.4 Dispositif d'entrée

Les dispositifs d'entrée rendent un système sensible aux changements qui interviennent dans son environnement, par la transmission de messages d'événement.

Les dispositifs d'entrée doivent être des dispositifs de commande à plusieurs maîtres et doivent permettre la mise en service et la configuration par un contrôleur d'application.

Les dispositifs d'entrée doivent utiliser les trames en avant uniquement pour transmettre des messages d'événement.

9.5 Instances de dispositifs d'entrée

9.5.1 Généralités

Un dispositif d'entrée doit comporter au moins une instance et 32 instances au maximum, comme cela doit être indiqué par *"numberOfInstances"*, qui peut faire l'objet d'une requête en utilisant "QUERY NUMBER OF INSTANCES".

Un dispositif de commande qui n'est qu'un contrôleur d'application doit avoir un *"numberOfInstances"* égal à 0.

9.5.2 Numéro d'instance

Chaque instance doit avoir un *"instanceNumber"* unique compris dans la plage $[0, \text{"numberOfInstances"} - 1]$.

9.5.3 Type d'instance

Le type d'instance pour chacune des instances d'un dispositif d'entrée peut être différent. Il peut faire l'objet d'une requête en utilisant "QUERY INSTANCE TYPE" avec l'octet d'instance réglé sur le schéma "type d'instance" ou "numéro d'instance" (voir Tableau 2). La signification de l'information d'événement transmise par "INPUT NOTIFICATION (*device/instance, event*)" dépend du type d'instance.

Le Tableau 4 montre le codage de type d'instance. Pour plus d'informations sur les différents types d'instances, voir les parties applicables de la série IEC 62386-3xx.

Tableau 4 – Types d'instances

Type d'instance	IEC 62386	Utilisé pour
0	Partie 103	Objet générique, dispositifs d'entrée non définis. Une autre méthode d'identification du dispositif doit être mise en œuvre, pour permettre au contrôleur d'application d'interpréter les événements.
1 à 31	Partie 301 à Partie 331	Ces parties IEC 62386-3xx décrivent les types d'instances, xx allant de 01 à 31.

9.5.4 Caractéristiques d'instance

La série IEC 62386 autorise la publication future d'extensions de caractéristiques qui élargissent les exigences du présent document, ou exonèrent des exigences particulières.

Les caractéristiques pour chacune des instances d'un dispositif d'entrée peuvent être différentes. Elles peuvent faire l'objet d'une requête en utilisant "QUERY FEATURE TYPE" et "QUERY NEXT FEATURE TYPE" avec l'octet d'instance réglé sur le schéma "caractéristique au niveau du type d'instance" ou "caractéristique au niveau du numéro d'instance" (voir Tableau 2).

Le Tableau 5 montre le codage de types de caractéristiques. Pour plus d'informations sur les différents types de caractéristiques, voir les parties applicables de la série IEC 62386-3xx.

Tableau 5 – Types de caractéristiques

Type de caractéristique	IEC 62386	Utilisé pour
32 à 96	Partie 332 à Partie 396	Ces parties IEC 62386-3xx décrivent les extensions de caractéristiques, xx allant de 32 à 96.

9.5.5 Groupes d'instances

Les groupes d'instances constituent un moyen pour un contrôleur d'application de placer des instances dans des groupes logiques, à travers des dispositifs d'entrée. Par conséquent, ces groupes logiques peuvent être utilisés pour configurer simultanément plusieurs instances.

Un contrôleur d'application peut utiliser jusqu'à 32 groupes de ce type, numérotés dans la plage [0,31]. Chaque instance peut être déclarée comme étant un membre de 3 groupes d'instances au maximum et doit exposer les variables de groupes d'instances comme cela est indiqué dans le Tableau 6.

Tableau 6 – Variables de groupes d'instances

Variable	Description
"instanceGroup0"	Numéro de groupe d'instances principal, MASK si aucune appartenance à un groupe n'est définie.
"instanceGroup1"	Numéro de groupe d'instances supplémentaire, MASK si aucune appartenance à un groupe n'est définie.
"instanceGroup2"	Numéro de groupe d'instances supplémentaire, MASK si aucune appartenance à un groupe n'est définie.

Les groupes d'instances sont attribués et font l'objet d'une requête en appliquant les opérations d'instance suivantes:

- "SET PRIMARY INSTANCE GROUP (*DTR0*)", "QUERY PRIMARY INSTANCE GROUP";
- "SET INSTANCE GROUP 1 (*DTR0*)", "QUERY INSTANCE GROUP 1";
- "SET INSTANCE GROUP 2 (*DTR0*)", "QUERY INSTANCE GROUP 2".

Le groupe principal est spécial dans le sens où seul ce numéro doit être utilisé lors du compte-rendu des événements (lorsque le compte-rendu d'événements de groupes d'instances s'applique). Les groupes supplémentaires constituent un moyen de configurer simultanément plusieurs instances.

9.6 Commandes qui excluent les messages d'événement

9.6.1 Généralités

Un dispositif de commande doit vérifier le schéma d'adressage du dispositif afin de déterminer si ce dernier est adressé par une commande. Le dispositif de commande doit accepter la commande, à moins que l'une ou plusieurs des conditions suivantes s'appliquent:

- la commande est envoyée par adressage court, et l'adresse courte attribuée n'est pas égale à "*shortAddress*";
- la commande est envoyée par adressage de groupes de dispositifs, et le groupe de dispositifs attribué ne correspond à aucun des groupes identifiés par "*deviceGroups*";
- la commande est envoyée par adressage de diffusion non adressée, et "*shortAddress*" n'est pas la valeur MASK;
- la commande est envoyée par adressage réservé;
- la commande n'est pas définie;
- la commande est envoyée par adressage de caractéristique, et la caractéristique donnée n'est pas mise en œuvre.

NOTE Pour les commandes d'instance et de caractéristique, les conditions supplémentaires d'acceptation des commandes s'appliquent. Ces dernières sont indiquées en 9.6.3 et 9.6.4.

9.6.2 Commandes de dispositif

L'octet d'instance doit être 0xFE pour les commandes de dispositif. Lorsque l'octet d'instance n'est pas égal à 0xFE, le dispositif de commande ne doit pas accepter ces commandes.

NOTE Ce mécanisme d'adressage permet le chevauchement des valeurs de code de fonctionnement pour les commandes de dispositif et les commandes d'instance.

9.6.3 Commandes d'instance

Pour les commandes d'instance acceptées par un dispositif d'entrée (voir 9.5), le schéma d'adressage d'instances détermine l'ensemble prévu d'instances de réception dans ce dispositif. Une instance doit accepter la commande d'instance, à moins que l'une ou plusieurs des conditions suivantes s'appliquent:

- la commande est envoyée par adressage de numéro d'instance, et le numéro d'instance attribué n'est pas égal à *"instanceNumber"*;
- la commande est envoyée par adressage de groupes d'instances, et le groupe d'instances attribué ne correspond à aucun des groupes identifiés par *"instanceGroup0"*, *"instanceGroup1"* et *"instanceGroup2"* (voir Tableau 6);
- la commande est envoyée par adressage de type d'instance, et le type d'instance attribué n'est pas égal à *"instanceType"*;
- la commande est envoyée par adressage réservé.

9.6.4 Commandes de caractéristique

Pour les commandes de caractéristique acceptées par un dispositif ou une instance (voir 9.5), le schéma d'adressage de caractéristique détermine l'ensemble prévu de caractéristiques de réception dans ce dispositif (voir Tableau 2).

Une caractéristique au niveau du dispositif doit accepter la commande de caractéristique, à moins que l'une ou plusieurs des conditions suivantes s'appliquent:

- la commande est envoyée par adressage de caractéristique différent de l'adressage de caractéristique au niveau du dispositif;
- la commande est envoyée par adressage réservé d'octet d'instance.

Une caractéristique au niveau de l'instance doit accepter la commande de caractéristique, à moins que l'une ou plusieurs des conditions suivantes s'appliquent:

- la commande est envoyée par adressage de caractéristique au niveau du numéro d'instance, et le numéro d'instance attribué n'est pas égal à *"instanceNumber"*;
- la commande est envoyée par adressage de caractéristique au niveau du groupe d'instances, et le groupe d'instances attribué ne correspond à aucun des groupes identifiés par *"instanceGroup0"*, *"instanceGroup1"* et *"instanceGroup2"*;
- la commande est envoyée par adressage de caractéristique au niveau du type d'instance, et le type d'instance attribué n'est pas égal à *"instanceType"*;
- la commande est envoyée par adressage de caractéristique au niveau du dispositif;
- la commande est envoyée par adressage d'instance;
- la commande est envoyée par adressage de dispositif;
- la commande est envoyée par adressage réservé d'octet d'instance.

Une commande diffusion de caractéristique doit être utilisée pour adresser une caractéristique attribuée à la fois au niveau du dispositif et au niveau de l'instance. Une diffusion de caractéristique doit être acceptée, à moins que l'une ou plusieurs des conditions supplémentaires suivantes s'appliquent:

- la commande est envoyée par adressage de caractéristique différent de la diffusion de caractéristique;
- la commande est envoyée par adressage réservé d'octet d'instance.

Si la commande de caractéristique est acceptée, l'octet de code de fonctionnement détermine quelle caractéristique est adressée.

9.7 Messages d'événement

9.7.1 Réponse aux messages d'événement

Un contrôleur d'application est libre de réagir à la réception de tout message d'événement ou d'ignorer le message.

NOTE Un contrôleur d'application désactivé n'envoie pas de trames en avant, mais il peut toujours actualiser son état interne en fonction des messages reçus.

9.7.2 Événement de cycle de mise sous tension de dispositif

Dans la mesure où l'événement de cycle de mise sous tension (voir 9.13.2) est un événement de dispositif, il ne respecte pas le format de trame d'événement par défaut. Les bits 12 à 0 doivent acheminer l'information d'adresse de dispositif comme- cela est indiqué dans le Tableau 7.

Tableau 7 – Information d'adresse de dispositif dans le cadre d'un événement de cycle de mise sous tension

Bits											
12	11	10	09	08	07	06	05	04	03	02	01 00
1 = groupe de dispositifs valide	Groupe de dispositifs le plus bas					1 = adresse courte valide	Adresse courte				

Le bit 12 doit être réglé si et seulement si le dispositif de commande de transmission est membre d'au moins un groupe de dispositifs. Les bits [11:7] doivent indiquer le numéro le plus bas d'appartenance à un groupe de dispositifs dans ce cas. Lorsque le bit 12 n'est pas réglé, les bits [11:7] doivent être supprimés.

Le bit 6 doit être réglé si et seulement si le dispositif de commande de transmission a une "shortAddress" différente de MASK. Les bits [5:0] doivent indiquer l'adresse courte du dispositif dans ce cas. Lorsque le bit 6 n'est pas réglé, les bits [5:0] doivent être supprimés.

9.7.3 Événement de notification d'entrée

Une instance d'un dispositif d'entrée doit, lorsqu'elle transmet un message d'événement, utiliser le schéma d'adressage de source d'événement sélectionné comme cela est défini dans le Tableau 8.

Tableau 8 – Schémas d'adressage d'événements

"eventScheme"	Description
0 (valeur par défaut)	Adressage d'instance, qui utilise le type et le numéro d'instance.
1	Adressage de dispositif, qui utilise l'adresse courte et le type d'instance.
2	Adressage de dispositif et d'instance, qui utilise l'adresse courte et le numéro d'instance.
3	Adressage de groupes de dispositifs, qui utilise le groupe de dispositifs et le type d'instance.
4	Adressage de groupes d'instances, qui utilise le groupe et le type d'instances.

Un contrôleur d'application peut définir et solliciter par le biais de requêtes l'"eventScheme" au moyen de "SET EVENT SCHEME (DTR0)" et de "QUERY EVENT SCHEME", respectivement.

NOTE 1 Une instance peut mettre en œuvre un schéma d'événement uniquement lorsque certaines conditions ont été remplies également par le contrôleur d'application. L'adressage d'instances est le seul schéma d'adressage qui fonctionne dans toutes les situations.

Dans les situations suivantes, l'instance doit immédiatement faire revenir "eventScheme" sur le schéma d'adressage d'instances par défaut décrit dans le Tableau 8:

- "eventScheme" a été réglé sur 1 ou 2 alors que l'unité logique contenante n'a pas d'adresse courte;
- "eventScheme" a été réglé sur 3 alors que l'unité logique contenante n'est pas membre d'un groupe de dispositifs;
- "eventScheme" a été réglé sur 4 alors que l'instance n'est pas membre d'un groupe d'instances principal (voir 9.5.5).

NOTE 2 Les situations ci-dessus peuvent se produire en raison d'une nouvelle commande "SET EVENT SCHEME (DTR0)" et/ou d'un changement de conditions.

NOTE 3 Ceci implique que la commande "SET EVENT SCHEME (DTR0)" peut "échouer", plutôt que de définir une préférence qui est accordée quelque temps plus tard. Pour éviter cet échec, les contrôleurs d'application peuvent définir le schéma d'événement souhaité après réalisation des aspects de configuration qui influencent le fonctionnement de ce même schéma.

De plus, et compte tenu d'un schéma d'adressage viable, l'instance doit

- se rapporter à "instanceNumber" uniquement comme numéro d'instance;
- se rapporter à "instanceType" uniquement comme type d'instance;
- se rapporter à "instanceGroup0" uniquement comme groupe d'instances;
- se rapporter à "shortAddress" uniquement comme adresse courte de l'unité logique contenante;
- se rapporter uniquement au numéro de groupe de dispositifs le plus bas, auquel appartient l'unité logique contenante.

9.7.4 Filtre de message d'événement

Le filtre de message d'événement peut être utilisé pour activer et désactiver des événements spécifiques. Lorsque le filtre d'un événement spécifique est désactivé, ce dernier ne doit pas être généré. Pour activer ou désactiver tous les événements, voir 9.10.3.

Un contrôleur d'application peut définir l'"eventFilter" au moyen de SET EVENT FILTER (DTR2:DTR1:DTR0) et peut solliciter par le biais de requêtes la variable au moyen de QUERY EVENT FILTER 0-7, de QUERY EVENT FILTER 8-15 et de QUERY EVENT FILTER 16-23, respectivement.

Les parties applicables de la série IEC 62386-3xx doivent définir la signification des bits dans *"eventFilter"*, et peuvent réduire la largeur de la variable *"eventFilter"* si nécessaire. Si la largeur est réduite à 2 octets, DTR2 doit être ignoré pour SET EVENT FILTER (*DTR2:DTR1:DTR0*) et QUERY EVENT FILTER 16-23 doit répondre NO. De façon similaire, si la largeur est réduite à 1 octet, DTR1 doit également être ignoré pour SET EVENT FILTER (*DTR2:DTR1:DTR0*) et QUERY EVENT FILTER 8-15 doit aussi répondre NO.

9.8 Signal d'entrée, valeur mesurée et *"inputValue"*

9.8.1 Généralités

Une instance doit transformer son signal d'entrée en valeur mesurée et exposer cette valeur au système, comme cela est décrit du 9.8.2 au 9.8.4.

9.8.2 Résolution d'entrée

La transformation doit être effectuée avec une précision indiquée par *"resolution"*. La résolution réelle utilisée pour un type d'instance particulier peut être soumise aux exigences des parties applicables de la série IEC 62386-3xx et/ou au choix du fabricant.

La valeur mesurée doit figurer dans la variable à *N* octets *"inputValue"*, où *N* est le nombre minimal d'octets nécessaire pour contenir au moins les bits *"resolution"*.

NOTE *N* est calculé comme (*"resolution"/8*) arrondi à l'entier le plus proche. Avec *"resolution"* comprise dans la plage [1,255], *"inputValue"* peut s'étendre jusqu'à 32 octets au maximum.

La valeur mesurée doit être alignée sur le bit de poids fort (msb, *most significant bit*) d'*"inputValue"*. Les bits non utilisés d'*"inputValue"* doivent contenir un schéma répétitif du ou des bits de poids fort de la valeur mesurée.

Le Tableau 9 fournit un exemple, qui montre une valeur mesurée de près de 50 % associée par verrouillage à une *"inputValue"* de 1 octet après transformation avec une *"resolution"* de 3, 4 et 5 bits, respectivement.

Tableau 9 – Valeur mesurée (≈ 50 %) par rapport à la résolution et à *"inputValue"*

Résolution	Valeur mesurée	Bits								<i>"inputValue"</i>
		7	6	5	4	3	2	1	0	
3 bits	3 de [0, 7]	0	1	1	0	1	1	0	1	109
4 bits	7 de [0, 15]	0	1	1	1	0	1	1	1	119
5 bits	15 de [0, 31]	0	1	1	1	1	0	1	1	123
NOTE Les bits grisés font partie intégrante de la (première) répétition des bits de poids fort.										

Cette méthode permet à un contrôleur d'application d'interpréter correctement la valeur d'entrée comme une valeur à 8 bits, indépendamment de la résolution d'instance réelle ou de la précision du capteur. La valeur minimale de tous les octets dans *"inputValue"* est toujours égale à 0, la valeur maximale 0xFF, pour toutes les résolutions. La valeur mesurée relative correspond (nonobstant une précision variable) à la valeur d'entrée relative.

9.8.3 Obtention de la valeur d'entrée

Une instance doit prendre en charge un mécanisme de verrouillage qui permet à un contrôleur d'application d'obtenir une valeur d'entrée à plusieurs octets cohérente. Un exemple de ce type de scénario de verrouillage est donné dans le Tableau 10.

Les contrôleurs d'application peuvent commencer à lire une valeur à plusieurs octets par l'envoi de la commande "QUERY INPUT VALUE". Cette commande doit déclencher un verrou qui contient une reproduction d'"inputValue" de sorte que les octets restants puissent être lus en utilisant une séquence de requêtes "QUERY INPUT VALUE LATCH". Après avoir retourné le dernier octet du verrou, l'instance ne doit répondre à aucune autre requête "QUERY INPUT VALUE LATCH" jusqu'après la "QUERY INPUT VALUE" suivante.

Tableau 10 – Exemple de séquence de requête pour lire une valeur d'entrée à 4 octets

Signal d'entrée	"inputValue"	Commande	Réponse	"inputValue" verrouillée
"12340000"	0x12340000	...	...	non spécifiée
"12345678"	0x12345678	"QUERY INPUT VALUE"	0x12	0x12345678
"852"	0x00000852	"QUERY INPUT VALUE LATCH"	0x34	0x12345678
"124852"	0x00124852	"QUERY INPUT VALUE LATCH"	0x56	0x12345678
"124852"	0x00124852	"QUERY INPUT VALUE LATCH"	0x78	0x12345678
"124852"	0x00124852	"QUERY INPUT VALUE LATCH"	NO	0x12345678

L'"inputValue" verrouillée est l'"inputValue" au moment de l'exécution de "QUERY INPUT VALUE".

NOTE 1 Ceci implique que lorsqu'un contrôleur d'application sollicite par le biais de requêtes l'"inputValue" en raison d'un message d'événement qu'il vient de recevoir, la valeur obtenue n'est pas nécessairement la même valeur que celle qui a déclenché l'événement.

La valeur verrouillée doit être actualisée uniquement lorsque la "QUERY INPUT VALUE" suivante est exécutée. Lorsque le contrôleur d'application utilise "QUERY INPUT VALUE LATCH" sans avoir utilisé "QUERY INPUT VALUE" comme commande précédente, la réponse peut contenir des données anciennes ou non valides.

NOTE 2 Pour empêcher tout accès simultané aux données verrouillées, un contrôleur d'application peut transmettre les requêtes nécessaires pour ce scénario dans une transaction, en sortant du scénario à tout moment.

NOTE 3 Lorsqu'un contrôleur d'application peut fonctionner avec suffisamment de précision avec une entrée à 16 bits pour le type d'instance donné, il peut s'interrompre après avoir reçu les 16 bits de poids fort de valeur d'entrée, et traiter ces bits comme s'ils étaient fournis par une instance avec "resolution" égale à 16. Ceci permet une mise en œuvre directe d'un algorithme indépendant de la résolution.

9.8.4 Notification des modifications

Une modification ou une séquence de modifications dans le signal d'entrée d'une instance doit générer un message d'événement, comme cela est exigé dans le présent document ou la présente partie de la série IEC 62386-3xx, qui décrit l'"instanceType" (voir 9.5.3) de cette instance.

Le message d'événement doit être envoyé en utilisant "INPUT NOTIFICATION (device/instance, event)", comme cela est décrit en 11.3.1.

NOTE Le fabricant du dispositif d'entrée peut s'assurer qu'aucun événement n'est perdu en fournissant une file d'attente. Les parties de la série IEC 62386-3xx peuvent imposer des restrictions supplémentaires, par exemple pour éviter l'inondation d'événements.

9.9 Défaillance système

Il convient qu'un contrôleur d'application détecte toute défaillance système et toute récupération du système. Il convient de préférence qu'il réagisse à toute panne d'alimentation du bus d'une durée supérieure à 40 ms, anticipant de ce fait un cycle de mise sous tension des dispositifs alimentés par le bus.

NOTE Les dispositifs alimentés par le bus peuvent s'éteindre en raison d'une panne d'alimentation de plus de 40 ms.

Ensuite, lorsque la défaillance système est résolue, il convient que le contrôleur d'application s'assure que le système reprend son fonctionnement normal.

9.10 Fonctionnement d'un dispositif de commande

9.10.1 Activer/désactiver le contrôleur d'application

S'il y a un contrôleur d'application, il est actif ou non actif, comme cela doit être reflété par *"applicationActive"*. Désactivé, le contrôleur d'application ne doit envoyer aucune trame en avant, sauf éventuellement une notification de cycle de mise sous tension (voir 9.13.2).

"applicationActive" ne doit pas influencer sur la réponse aux transmissions en avant à l'arrivée, y compris la transmission des trames en arrière à la suite de requêtes.

NOTE Ceci permet au contrôleur d'application de surveiller le bus, mais ce même contrôleur ne peut pas utiliser des trames en avant pour réagir.

"applicationActive" doit être mémorisé dans la NVM du contrôleur d'application. S'il y a un contrôleur d'application, la valeur par défaut doit être TRUE, qui peut être modifiée par un autre contrôleur d'application au moyen des commandes ENABLE APPLICATION CONTROLLER et DISABLE APPLICATION CONTROLLER. *"applicationActive"* peut faire l'objet d'une requête en utilisant "QUERY APPLICATION CONTROLLER ENABLED".

9.10.2 Contrôleur d'application toujours actif

Si un contrôleur d'application est présent, il peut être toujours actif. Cela doit se traduire par *"applicationControllerAlwaysActive"* étant TRUE.

Lorsqu'*"applicationControllerAlwaysActive"* est TRUE, *"applicationControllerPresent"* et *"applicationActive"* doivent toujours être TRUE.

"applicationControllerAlwaysActive" peut être observé par "QUERY DEVICE CAPABILITIES" et "QUERY APPLICATION CONTROLLER ALWAYS ACTIVE".

9.10.3 Activer/désactiver les messages d'événement

Les messages d'événement sont activés ou désactivés, comme cela doit être reflété par *"instanceActive"*. Désactivée, l'instance ne doit envoyer aucune trame en avant. Autrement dit, l'instance ne produit aucun message d'événement.

"instanceActive" ne doit pas influencer sur la réponse aux transmissions en avant à l'arrivée, y compris la transmission des trames en arrière à la suite de requêtes.

"instanceActive" doit être mémorisé dans la mémoire rémanente du dispositif d'entrée. La valeur par défaut doit être TRUE, qui peut être modifiée par un contrôleur d'application au moyen des commandes "ENABLE INSTANCE" et "DISABLE INSTANCE". *"instanceActive"* peut faire l'objet d'une requête en utilisant "QUERY INSTANCE ENABLED".

Pour limiter le nombre de messages d'événement lorsqu'ils sont activés, le filtrage est aussi possible, voir 9.7.4.

NOTE Les requêtes constituent le seul moyen d'obtenir des informations d'une instance lorsque les messages d'événement sont désactivés.

9.10.4 Mode repos

En mode repos, le dispositif de commande ne doit générer aucune trame en avant, sauf comme résultat possible de l'exécution de SEND TESTFRAME. Aucune commande (voir également 9.10.1) et aucun message d'événement (voir également 9.10.3) ne doivent être transmis, indépendamment d'*"applicationActive"* ou de tout *"instanceActive"*. Les messages

d'événements ne doivent pas être mis en file d'attente et ne doivent pas être définis comme étant en attente. Cela signifie que de tels événements sont rejetés.

Le mode repos est un mode provisoire lancé ou relancé à l'aide de la commande "START QUIESCENT MODE". Il doit s'interrompre automatiquement dans un délai de $15 \text{ min} \pm 1,5 \text{ min}$ après exécution de la dernière commande "START QUIESCENT MODE". De plus, la commande "STOP QUIESCENT MODE" doit interrompre le mode repos immédiatement.

En mode repos, un dispositif de commande doit continuer à répondre aux commandes. "QUERY QUIESCENT MODE" peut être utilisée pour déterminer si un dispositif de commande est ou non en mode repos.

A la mise sous tension du dispositif de commande, le mode repos doit être DISABLED.

NOTE 1 Le mode repos peut être utilisé par le contrôleur d'application lors de l'initialisation (voir 9.15) afin de s'assurer que les comparaisons d'adresses aléatoires ne sont pas rendues inexécutables par les trames en avant en provenance d'autres dispositifs du bus.

NOTE 2 Le mode repos fonctionne indépendamment d'"*applicationActive*" et d'"*instanceActive*". Ceci implique que l'arrêt du mode repos n'active pas nécessairement les transmissions de trames en avant.

9.10.5 Modes de fonctionnement

9.10.5.1 Généralités

Différents modes de fonctionnement peuvent être sélectionnés au niveau du dispositif au moyen de la commande "SET OPERATING MODE (*DTR0*)". L'"*operatingMode*" actuellement sélectionné peut faire l'objet d'une requête au moyen de "QUERY OPERATING MODE".

Les modes de fonctionnement 0x00 à 0x7F sont définis dans le présent document. Le mode de fonctionnement 0x00 au moins doit être disponible. Les modes de fonctionnement 0x80 à 0xFF sont spécifiques au fabricant. La requête "QUERY MANUFACTURER SPECIFIC MODE" peut être utilisée pour déterminer si le mode de fonctionnement du dispositif de commande est un mode normal défini dans l'IEC 62386 ou un mode spécifique au fabricant.

9.10.5.2 Mode de fonctionnement 0x00: mode normal

Lorsqu'un dispositif est en "*operatingMode*" 0x00, son comportement doit être tel qu'exigé par le présent document, jusqu'à ce qu'il soit réglé en mode de fonctionnement différent de 0x00.

9.10.5.3 Mode de fonctionnement 0x01 à 0x7F: réservé

Les modes de fonctionnement 0x01 à 0x7F sont réservés et ne doivent pas être utilisés.

9.10.5.4 Mode de fonctionnement 0x80 à 0xFF: modes spécifiques au fabricant

Il convient d'utiliser les modes spécifiques au fabricant uniquement si les caractéristiques exigées par l'application ne sont pas couvertes par la série IEC 62386. Lorsque le mode de fonctionnement d'un dispositif de commande est spécifique au fabricant, le comportement dudit dispositif peut également être spécifique au fabricant, avec les exceptions suivantes:

- en ce qui concerne l'accès au bus, le dispositif de commande doit être conforme à l'IEC 62386-101:2022;
- le dispositif de commande doit être conforme au présent document tant que les commandes suivantes sont concernées:
 - "SET OPERATING MODE (*DTR0*)", "QUERY OPERATING MODE" et "QUERY MANUFACTURER SPECIFIC MODE";

- toutes les commandes spéciales (voir 11.10) sauf
WRITE MEMORY LOCATION (*DTR1, DTR0, data*), WRITE MEMORY LOCATION –
NO REPLY (*DTR1, DTR0, data*) et DIRECT WRITE MEMORY (*DTR1, offset, data*).

Pour les commandes ci-dessus, les exigences du 7.2.1 s'appliquent.

Il est recommandé de continuer à suivre les commandes spécifiées dans toutes les parties mises en œuvre de la série IEC 62386 même dans le cas de modes spécifiques au fabricant.

9.11 Blocs de mémoire

9.11.1 Généralités

Les blocs de mémoire sont des espaces mémoire librement accessibles qui contiennent les informations ou les paramètres de configuration du dispositif de commande. Il n'est pas nécessaire de mettre en œuvre tous les blocs de mémoire consécutifs. De même, au sein d'un bloc de mémoire, il n'est pas nécessaire de mettre en œuvre tous les emplacements consécutifs. Tous les emplacements de blocs de mémoire mis en œuvre de tous les blocs de mémoire également mis en œuvre peuvent être lus au moyen de commandes d'accès de la mémoire. Une partie de la mémoire est en lecture seule et programmée par le fabricant du dispositif de commande ou mise à jour par le dispositif de commande lui-même. Pour toutes les autres parties, l'accès en écriture au moyen de commandes d'accès de la mémoire peut être activé par le fabricant. À l'exception de l'emplacement 0x02, tous les emplacements inscriptibles d'un bloc de mémoire sont verrouillables, sauf spécification contraire dans la table des blocs de mémoire correspondante. Les emplacements dans les blocs de mémoire doivent être mis en œuvre en utilisant les types de mémoires indiqués dans le Tableau 11.

Tableau 11 – Types de mémoires

Type de mémoire	Accessibilité par le bus ^a	Volatilité ^b	Peut être modifiée de manière autonome par le dispositif de commande pendant l'exécution	Description
ROM	RO	NV	Non	Concerne l'ensemble des valeurs fixes qui ne varient pas pendant l'exécution du dispositif de commande. NOTE La mémoire ROM est par nature en lecture seule, mais peut être modifiée lors de la programmation au cours de la production.
RAM-RO	RO	V	Oui	Pour les valeurs qui ne sont pas retenues sur un cycle de mise sous tension. Le contenu ne peut pas être modifié à partir de l'interface.
RAM-RW	RW	V	Oui	Pour les valeurs qui ne sont pas retenues sur un cycle de mise sous tension. Le contenu peut être modifié à partir de l'interface.
NVM-RO	RO	NV	Oui	Pour les valeurs qui sont retenues sur un cycle de mise sous tension. Le contenu ne peut pas être modifié à partir de l'interface.
NVM-RW	RW	NV	Oui	Pour les valeurs qui sont retenues sur un cycle de mise sous tension. Le contenu peut être modifié à partir de l'interface.
^a RO: lecture seule (<i>Read-Only</i>). RW: lecture-écriture (<i>Read-Write</i>). ^b V: volatile (non retenue sur un cycle de mise sous tension). NV: non volatile (retenue sur un cycle de mise sous tension).				

L'espace mémoire adressable est limité à un espace maximal de 256 blocs de mémoire, chacun comportant au maximum 255 octets (environ 64 ko). Dans la mesure où le présent document spécifie comment mettre en œuvre les blocs de mémoire 0 et 1 (lorsqu'ils existent), et réserve les blocs de mémoire 200 à 255, ceci laisse de l'espace pour 198 blocs de mémoire pour des besoins spécifiques au fabricant dans la plage de [2,199].

9.11.2 Carte de mémoire

Lorsqu'un bloc de mémoire spécifique au fabricant dans la plage de [2,199] est mis en œuvre, l'affectation de son contenu doit être conforme à la carte de mémoire fournie dans le Tableau 12.

Tableau 12 – Carte de mémoire de base des blocs de mémoire

Adresse	Description	Valeur par défaut (valeur d'origine)	Valeur réinitialisée ^b	Type de mémoire
0x00	Adresse du dernier emplacement de mémoire accessible	rodage en usine, plage [0x03,0xFE]	pas de modification	ROM
0x01	Octet indicateur ^a	a	a	tout type ^a
0x02	Octet de verrouillage du bloc de mémoire. Les octets verrouillables dans le bloc de mémoire doivent être en lecture seule, tandis que l'octet de verrouillage a une valeur différente de 0x55.	0xFF	0xFF ^c	RAM-RW
[0x03,0xFE]	Contenu de bloc de mémoire ^a	a	a	tout type ^a
0xFF	Réservée – non mise en œuvre	réponse NO	pas de modification	n.a.
^a L'objet, les valeurs par défaut, les valeurs de mise sous tension, les valeurs réinitialisées et les types de mémoires de ces octets doivent être définis par le fabricant. ^b Réinitialisation de la valeur après une commande "RESET MEMORY BANK". ^c Également utilisée comme valeur de mise sous tension, sauf indication contraire explicite.				

L'octet qui se trouve dans l'emplacement 0x00 de chaque bloc contient l'adresse du dernier emplacement de mémoire accessible du bloc. La valeur doit être comprise dans la plage [0x03,0xFE].

L'octet dans l'emplacement 0x01 est spécifique au fabricant. Lorsqu'il est mis en œuvre, il convient que le fabricant décrive l'utilisation de cet octet (ainsi que l'ensemble du contenu du bloc de mémoire).

NOTE 1 Il peut être utilisé, par exemple, pour mémoriser une somme de contrôle dans le cas d'un bloc de mémoire à contenu statique. Il n'est pas utile d'utiliser une somme de contrôle dans un bloc de mémoire dont le contenu est modifié par le dispositif de commande.

L'octet de l'emplacement 0x02 doit être utilisé pour verrouiller l'accès en écriture. L'emplacement de mémoire 0x02 proprement dit ne doit jamais être verrouillé pour écriture. Alors que cet emplacement de mémoire contient toute valeur différente de 0x55, tous les emplacements de mémoire marqués "(verrouillable)" du bloc de mémoire correspondant doivent être en lecture seule. Le dispositif de commande ne doit pas modifier la valeur de l'octet de verrouillage autrement qu'à la suite d'un cycle de puissance, d'une commande "RESET MEMORY BANK (DTR0)" ou de toute autre commande qui affecte l'octet de verrouillage.

L'emplacement 0xFF est un emplacement réservé dans chaque bloc de mémoire et n'est pas accessible. Cet emplacement ne doit pas être mis en œuvre comme emplacement de bloc de mémoire normal. Lorsqu'il est adressé, le dispositif de commande doit répondre comme si cet emplacement n'était pas mis en œuvre, et il ne doit pas augmenter "DTR0".

NOTE 2 Cet emplacement est réservé pour interrompre l'augmentation automatique de "DTR0".

9.11.3 Sélection d'un emplacement de bloc de mémoire

Pour sélectionner un emplacement de bloc de mémoire, une combinaison de numéro et d'emplacement au sein du bloc de mémoire est exigée.

Le bloc de mémoire doit être sélectionné par un réglage du numéro de bloc de mémoire en "DTR1". L'emplacement dans le bloc de mémoire doit être sélectionné par la valeur en "DTR0".

9.11.4 Emplacements de mémoire protégéables

Les emplacements de blocs de mémoire marqués comme "protégéables" peuvent être protégés contre la lecture ou l'écriture ou être supprimés à l'aide d'une méthode spécifique au fabricant.

Les emplacements protégéables dont la protection contre la lecture est activée doivent répondre MASK à la commande "READ MEMORY LOCATION (DTR1, DTR0)". Dans ce cas, la mise en œuvre ou la mise à disposition de la caractéristique associée est facultative.

NOTE 1 La protection contre la lecture peut être activée par défaut. Si la protection contre la lecture est activée, cela signifie que la fonction associée peut ne pas être mise en œuvre et que le fabricant n'est pas obligé de fournir un mécanisme pour désactiver la protection contre la lecture.

Le contenu des emplacements protégéables dont la protection contre l'écriture est activée ne peut pas être écrasé.

NOTE 2 Cela signifie que la commande WRITE MEMORY LOCATION ne renvoie aucune réponse lors d'une tentative d'écriture dans un emplacement protégé contre l'écriture.

9.11.5 Lecture dans le bloc de mémoire

9.11.5.1 Généralités

Un emplacement de bloc de mémoire sélectionné peut être lu au moyen de la commande "READ MEMORY LOCATION (DTR1, DTR0)". La réponse doit être la valeur de l'octet à l'emplacement de bloc de mémoire adressé.

Lorsque le bloc de mémoire sélectionné n'est pas mis en œuvre, la commande doit être rejetée.

Lorsque l'emplacement de bloc de mémoire sélectionné se situe au-dessous de l'emplacement 0xFF, "DTR0" doit être augmenté de un, même si l'emplacement de mémoire n'est pas mis en œuvre. Dans le cas contraire, "DTR0" ne doit pas varier. Ce mécanisme permet une lecture consécutive facile des emplacements de blocs de mémoire.

Après lecture de plusieurs octets issus d'un bloc de mémoire, il convient que le contrôleur d'application vérifie la valeur de "DTR0" afin de s'assurer que son emplacement est celui attendu ou souhaité. Tout défaut d'adaptation indique une erreur de lecture.

9.11.5.2 Lecture de valeurs à plusieurs octets

Pour assurer la cohérence des données lors de la lecture d'une valeur à plusieurs octets dans un bloc de mémoire, un mécanisme doit être mis en œuvre afin de verrouiller l'ensemble des octets de la valeur à plusieurs octets lorsque le premier octet de cette valeur est lu, et de les déverrouiller lorsque le premier octet d'une autre valeur à un octet ou à plusieurs octets de tout bloc de mémoire de la même unité logique est lu.

La lecture d'un emplacement à plusieurs octets doit restituer la valeur stockée, mais pas les valeurs qui résident dans le tampon d'écriture utilisé pour la mise en tampon tant que la valeur complète n'a pas été écrite en mémoire (voir 9.11.6.3).

9.11.5.3 Emplacements non mis en œuvre

Lorsqu'un bloc de mémoire existe, que l'emplacement de bloc de mémoire sélectionné

- n'est pas mis en œuvre, et que MASK n'apparaît pas dans la plage de valeurs admise, ou
- se situe au-dessus du dernier emplacement de mémoire accessible,

la réponse à "READ MEMORY LOCATION (*DTR1*, *DTR0*)" doit être NO.

Lorsqu'un bloc de mémoire existe, que l'emplacement de bloc de mémoire sélectionné

- n'est pas mis en œuvre, et
- que la plage admise pour la valeur stockée dans l'emplacement du bloc de mémoire inclut MASK,

la réponse à "READ MEMORY LOCATION (*DTR1*, *DTR0*)" doit être MASK.

NOTE L'utilisation de MASK dans la plage de valeurs admises indiquée dans une table des blocs de mémoire rend effectivement la mise en œuvre de l'emplacement facultative.

9.11.5.4 Emplacements temporairement indisponibles

Pour les emplacements de mémoire qui incluent TMASK dans la plage de valeurs admises d'une table des blocs de mémoire, la réponse à "READ MEMORY LOCATION (*DTR1*, *DTR0*)" doit être TMASK si la valeur est temporairement indisponible. Sauf en conditions de défaut, le dispositif de commande doit fournir une valeur valide dans un délai de 30 s.

EXEMPLE A la suite d'un cycle de mise sous tension externe, les valeurs de bloc de mémoire telles que les mesures peuvent être temporairement indisponibles en raison du temps d'établissement des filtres numériques et sont indiquées par l'utilisation de TMASK.

NOTE Si plusieurs valeurs TMASK sont renvoyées sur une période donnée, les contrôleurs d'application peuvent en déduire qu'une condition de défaut s'est produite sur le dispositif de commande.

9.11.5.5 Verrouillage d'un bloc de mémoire complet pour la lecture

Les tables des blocs de mémoire qui indiquent que la valeur 0xAA de l'octet de verrouillage verrouille le bloc complet doivent fonctionner comme suit:

Si l'octet de verrouillage contient une valeur autre que 0xAA, l'écriture des valeurs suivantes dans l'octet de verrouillage doit produire le résultat indiqué:

- 0xAA: tous les emplacements du bloc de mémoire doivent être verrouillés et ne doivent pas changer tant que l'octet de verrouillage n'a pas été écrit ou qu'un cycle de mise sous tension ne s'est pas produit;
- autres valeurs: le verrou du bloc de mémoire ne doit pas être affecté.

Si l'octet de verrouillage contient la valeur 0xAA, l'écriture des valeurs suivantes dans l'octet de verrouillage doit produire le résultat indiqué:

- 0xAA: tous les emplacements du bloc de mémoire doivent être reverrouillés (mis à jour) et ne doivent pas changer à nouveau tant que l'octet de verrouillage n'a pas été écrit ou qu'un cycle de mise sous tension ne s'est pas produit;
- autres valeurs: le verrou du bloc de mémoire doit être désactivé. Les lectures à partir de la mémoire doivent restituer les dernières valeurs renvoyées.

Une tentative d'écriture dans un autre emplacement que l'octet de verrouillage d'un bloc de mémoire verrouillé doit produire le même comportement que si l'emplacement de mémoire n'était pas mis en œuvre.

Le verrouillage d'un bloc de mémoire complet ne doit pas compromettre la lecture ou l'écriture dans les autres blocs de mémoire.

EXEMPLE L'exemple suivant montre deux contrôleurs d'application qui accèdent à un bloc de mémoire verrouillé: Si le contrôleur d'application A verrouille le bloc de mémoire complet 202, le contrôleur d'application B a besoin de savoir que le bloc de mémoire 202 est verrouillé. Lorsque le contrôleur d'application B lit le bloc de mémoire 202, il obtient les données verrouillées. Le contrôleur d'application B peut déterminer cela en surveillant l'activité du bus liée à l'écriture de l'octet de verrouillage, ou en lisant l'octet de verrouillage avant de lire d'autres emplacements.

9.11.6 Écriture dans le bloc de mémoire

9.11.6.1 Généralités

Les commandes en écriture sont des commandes spéciales et ne sont par conséquent pas adressables. Afin de sélectionner le ou les dispositifs de commande corrects, la commande adressable "ENABLE WRITE MEMORY" doit être utilisée. A l'exécution de "ENABLE WRITE MEMORY", le ou les dispositifs de commande adressés doivent régler *"writeEnableState"* sur ENABLED.

Le dispositif de commande doit exécuter les commandes suivantes d'écriture dans un emplacement de bloc de mémoire sélectionné uniquement lorsque *"writeEnableState"* est ENABLED et que le bloc de mémoire adressé est mis en œuvre:

- "WRITE MEMORY LOCATION (*DTR1*, *DTR0*, *data*)": le dispositif de commande doit confirmer l'écriture dans un emplacement de mémoire par une réponse qui équivaut à la valeur *data*;

NOTE 1 La valeur qui peut être lue à partir de l'emplacement de bloc de mémoire n'est pas nécessairement *data*.

- "WRITE MEMORY LOCATION – NO REPLY (*DTR1*, *DTR0*, *data*)": l'écriture dans un emplacement de mémoire ne doit pas entraîner de réponse du dispositif de commande;
- "DIRECT WRITE MEMORY (*DTR1*, *offset*, *data*)": l'adresse de l'emplacement de mémoire à l'intérieur du bloc sélectionné est indiquée par le contenu de l'octet d'instance. *offset* est reproduit dans "*DTR0*", après quoi la commande est traitée comme "WRITE MEMORY LOCATION (*DTR1*, *DTR0*, *data*)". Le dispositif de commande doit confirmer l'écriture dans un emplacement de mémoire par une réponse qui équivaut à *data*.

Un dispositif de commande doit régler *"writeEnableState"* sur DISABLED en cas d'acceptation de toute commande autre que l'une des commandes suivantes:

- "WRITE MEMORY LOCATION (*DTR1*, *DTR0*, *data*)",
"WRITE MEMORY LOCATION – NO REPLY (*DTR1*, *DTR0*, *data*)",
"DIRECT WRITE MEMORY (*DTR1*, *offset*, *data*)";
- "DTR0 (*data*)", "DTR1 (*data*)", "DTR1:DTR0 (*data1*, *data0*)", "DTR2 (*data*)",
"DTR2:DTR1 (*data2*, *data1*)";
- "QUERY CONTENT DTR0", "QUERY CONTENT DTR1", "QUERY CONTENT DTR2".

NOTE 2 Cela signifie que les commandes autres que celles indiquées ci-dessus, qui sont acceptées puis rejetées, continuent de définir *"writeEnableState"* sur DISABLED.

9.11.6.2 Échecs d'écriture

Lorsque l'emplacement de bloc de mémoire sélectionné

- n'est pas mis en œuvre, ou
- se situe au-dessus du dernier emplacement de mémoire accessible, ou
- est verrouillé (voir 9.11.2), ou
- n'est pas inscriptible, ou
- est un emplacement protégeable actuellement protégé contre l'écriture (voir 9.11.4), ou
- en cas de tentative d'écriture d'une valeur située en dehors de la plage admise,

la réponse à "WRITE MEMORY LOCATION (*DTR1*, *DTR0*, *data*)" et "DIRECT WRITE MEMORY (*DTR1*, *offset*, *data*)" doit être NO et aucun emplacement de mémoire ne doit être en mode écriture. Si cela s'applique lors de l'écriture dans un emplacement à plusieurs octets, la réponse NO doit se produire après la tentative d'écriture dans l'octet de poids faible (LSB, *Least Significant Byte*).

Lorsque l'emplacement de bloc de mémoire sélectionné se situe au-dessous de 0xFF, "*DTR0*" doit être augmenté de un. Dans le cas contraire, "*DTR0*" ne doit pas varier. Ce mécanisme permet une écriture consécutive facile dans les emplacements de blocs de mémoire.

9.11.6.3 Écriture de valeurs à plusieurs octets

Pour assurer la cohérence des données lors de l'écriture d'une valeur à plusieurs octets dans un bloc de mémoire, un tampon RAM doit être utilisé afin de stocker les octets temporaires en cours d'écriture tant que l'octet de poids faible de la valeur à plusieurs octets n'a pas été écrit. Dès que l'octet de poids faible est écrit, la valeur complète est alors écrite dans les emplacements du bloc de mémoire.

NOTE 1 Le contenu des tampons utilisés pour le stockage des octets temporaires lors de l'écriture de valeurs à plusieurs octets peut être perdu lors d'une opération d'écriture d'une autre valeur à plusieurs octets dans le même bloc de mémoire, ou pendant un cycle de mise sous tension, ou à la suite de l'exécution d'une commande RESET MEMORY BANK (*DTR0*). Cela signifie qu'il suffit d'inclure un seul tampon RAM de ce type par bloc de mémoire pour l'écriture.

Après saisie de plusieurs octets dans un bloc de mémoire, il convient que le contrôleur d'application vérifie la valeur de "*DTR0*" afin de s'assurer que son emplacement est celui attendu ou souhaité. Tout défaut d'adaptation indique une erreur d'écriture.

NOTE 2 "*DTR0*" est également augmenté lorsqu'un emplacement de bloc de mémoire non mis en œuvre est adressé avant d'atteindre 0xFF.

9.11.7 Bloc de mémoire 0

Le bloc de mémoire 0 contient les informations concernant le dispositif de commande. Le bloc de mémoire 0 doit être mis en œuvre dans tous les dispositifs de commande à plusieurs maîtres.

Le bloc de mémoire 0 doit être mis en œuvre en utilisant la carte de mémoire décrite dans le Tableau 13, avec au moins la mise en œuvre des emplacements de mémoire jusqu'à l'adresse 0x7F, en excluant les emplacements réservés.

Tableau 13 – Carte de la mémoire du bloc de mémoire 0

Adresse	Description	Valeur par défaut (valeur d'origine)	Type de mémoire
0x00	Adresse du dernier emplacement de mémoire accessible	rodage en usine	ROM
0x01	Réservée – non mise en œuvre	réponse NO	n.a.
0x02	Numéro du dernier bloc de mémoire accessible	rodage en usine, plage [0,0xFF]	ROM
0x03	Octet GTIN 0 (octet de poids fort) ^a	rodage en usine	ROM
0x04	Octet GTIN 1	rodage en usine	ROM
0x05	Octet GTIN 2	rodage en usine	ROM
0x06	Octet GTIN 3	rodage en usine	ROM
0x07	Octet GTIN 4	rodage en usine	ROM
0x08	Octet GTIN 5 (octet de poids faible)	rodage en usine	ROM
0x09	Version du microprogramme (principale)	rodage en usine	ROM
0x0A	Version du microprogramme (secondaire)	rodage en usine	ROM
0x0B	Octet de numéro d'identification 0 (octet de poids fort)	rodage en usine	ROM
0x0C	Octet de numéro d'identification 1	rodage en usine	ROM
0x0D	Octet de numéro d'identification 2	rodage en usine	ROM
0x0E	Octet de numéro d'identification 3	rodage en usine	ROM
0x0F	Octet de numéro d'identification 4	rodage en usine	ROM
0x10	Octet de numéro d'identification 5	rodage en usine	ROM
0x11	Octet de numéro d'identification 6	rodage en usine	ROM
0x12	Octet de numéro d'identification 7 (octet de poids faible)	rodage en usine	ROM
0x13	Version du matériel (principale)	rodage en usine	ROM
0x14	Version du matériel (secondaire)	rodage en usine	ROM
0x15	Numéro de version 101 ^b	rodage en usine, selon le numéro de version mis en œuvre	ROM
0x16	Numéro de version 102 de tous les appareillages de commande intégrés ^c	rodage en usine, selon le numéro de version mis en œuvre	ROM
0x17	Numéro de version 103 de tous les dispositifs de commande intégrés ^d	rodage en usine, selon le numéro de version mis en œuvre	ROM
0x18	Nombre de dispositifs de commande logiques dans l'unité de bus	rodage en usine, plage [1,64]	ROM
0x19	Nombre d'appareillages de commande logiques dans l'unité de bus	rodage en usine, plage [0,64]	ROM
0x1A	Indice de ce dispositif de commande logique	rodage en usine, plage [0,(emplacement 0x18) – 1]	ROM
0x1B ^f	Configuration actuelle de l'unité de bus ^f	rodage en usine ^g	ROM
[0x1C,0x7F]	Réservée – non mise en œuvre	réponse NO	n.a.
[0x80,0xFE]	Informations supplémentaires concernant les dispositifs de commande ^e	^e	ROM
0xFF	Réservée – non mise en œuvre	réponse NO	n.a.

Légende

GTIN code article international

LSB octet de poids faible

MSB octet de poids fort

- ^a Il est recommandé de ne pas réutiliser le produit GTIN au cours de la durée de vie prévue du produit après installation.
- ^b Le format du numéro de version est défini en 4.2 de l'IEC 62386-101:2022.
- ^c Le format du numéro de version est défini en 4.2 de l'IEC 62386-102:2022. Lorsqu'il n'est pas mis en œuvre, ceci est indiqué par 0xFF.
- ^d Le format du numéro de version est défini en 4.2.
- ^e L'objet et la valeur (par défaut) de ces octets doivent être définis par le fabricant.
- ^f Voir 9.20. Si cet emplacement n'est pas mis en œuvre, la réponse doit être NO.
- ^g La configuration actuelle de l'unité de bus peut être modifiée par une méthode spécifique au fabricant.

Lorsqu'une unité de bus comporte plus d'une unité logique, toutes les unités logiques doivent avoir les mêmes valeurs dans les emplacements de blocs de mémoire 0x03 jusqu'à 0x19 compris. Toutes les unités logiques du dispositif de commande doivent avoir les mêmes valeurs dans l'emplacement 0x1B.

Une unité de bus peut contenir à la fois des appareillages de commande et des dispositifs de commande. Ceux-ci partagent différents numéros (par exemple, GTIN, numéro d'identification). Afin d'éviter les problèmes en lecture et d'obtenir différentes réponses selon le schéma d'adressage utilisé, la présentation du bloc de mémoire est identique pour les appareillages de commande et les dispositifs de commande jusqu'à l'emplacement 0x19 compris. Les données doivent être également identiques. Le contrôleur d'application peut utiliser les commandes spécifiées dans l'IEC 62386-102:2022 ou celles spécifiées dans le présent document pour identifier les données de base, à condition que ces deux types de commandes soient mis en œuvre.

Les octets des emplacements 0x03 à 0x08 ("GTIN 0" à "GTIN 5") doivent contenir le code article international (GTIN) en binaire (voir [1]¹). Les octets doivent être mémorisés en commençant par le bit de poids fort et doivent contenir des zéros non significatifs.

Les octets des emplacements 0x09 et 0x0A ("version du microprogramme") doivent contenir la version du microprogramme de l'unité de bus.

Les octets des emplacements 0x0B à 0x12 ("octet de numéro d'identification 0" à "octet de numéro d'identification 7") doivent contenir 64 bits d'un numéro d'identification de l'unité de bus. Il est recommandé que ce numéro d'identification soit le numéro de série. Le numéro d'identification doit être mémorisé avec l'octet de poids fort dans l'octet de numéro d'identification 7" et les bits non utilisés doivent être remplis de 0.

La combinaison du numéro d'identification et du numéro GTIN doit être unique.

L'octet des emplacements 0x13 et 0x14 ("version du matériel") doit contenir la version du matériel de l'unité de bus.

L'octet de l'emplacement 0x15 doit contenir le numéro de version IEC 62386-101 mis en œuvre de l'unité de bus.

¹ Les chiffres entre crochets renvoient à la Bibliographie.

L'octet de l'emplacement 0x16 doit contenir le numéro de version IEC 62386-102 mis en œuvre de l'unité de bus. En l'absence de mise en œuvre d'un appareillage de commande, le numéro de version doit être 0xFF.

L'octet de l'emplacement 0x17 doit contenir le numéro de version IEC 62386-103 mis en œuvre de l'unité de bus.

L'octet de l'emplacement 0x18 doit contenir le nombre de dispositifs de commande logiques intégrés dans l'unité de bus. Le nombre d'unités logiques doit être compris entre 1 et 64.

L'octet de l'emplacement 0x19 doit contenir le nombre d'appareillages de commande logiques intégrés dans l'unité de bus. Le nombre d'unités logiques doit être compris entre 0 et 64.

L'octet de l'emplacement 0x1A doit représenter l'indice unique du dispositif de commande logique qui met en œuvre ce bloc de mémoire. La plage valide de cet indice est comprise entre 0 et le nombre total de dispositifs de commande logiques intégrés dans l'unité de bus moins un.

EXEMPLE Un produit peut contenir trois dispositifs de commande logiques avec trois adresses courtes différentes. Chacune de ces unités logiques a le même GTIN et le même numéro d'identification, et chacune consigne la valeur 3 pour le nombre de dispositifs de commande logiques, l'indice des trois dispositifs de commande logiques étant pour sa part consigné comme valeur 0, 1 ou 2, respectivement. La lecture de l'emplacement 0x1A par diffusion produit une trame en arrière conformément au 9.6.2 de l'IEC 62386-101:2022 (trame en arrière erronée).

L'octet dans l'emplacement 0x1B, s'il existe, doit contenir la configuration actuelle de l'unité de bus, qui doit indiquer la mise en œuvre actuellement sélectionnée des contrôleurs d'application et des unités logiques. Voir 9.20.

9.11.8 Bloc de mémoire 1 (facultatif)

Le bloc de mémoire 1 est réservé pour être utilisé par un fabricant d'origine (OEM *Original Equipment Manufacturer*, par exemple fabricant de luminaires) pour mémoriser des informations supplémentaires sans aucune influence sur la fonctionnalité du dispositif de commande. La mise en œuvre du bloc de mémoire 1 est facultative.

Lorsqu'il est mis en œuvre, le bloc de mémoire 1 doit au moins mettre en œuvre les emplacements de mémoire jusqu'à l'adresse 0x10 comprise. L'utilisation fixe pour l'emplacement 0x00 à 0x02 et l'utilisation de la carte de mémoire recommandée pour les emplacements 0x03 à 0x10 sont décrites dans le Tableau 14.

Tableau 14 – Carte de la mémoire du bloc de mémoire 1

Adresse	Description	Valeur par défaut (valeur d'origine)	Valeur réinitialisée ^b	Type de mémoire
0x00	Adresse du dernier emplacement de mémoire accessible	rodage en usine, plage [0x10,0xFE]	pas de modification	ROM
0x01	Octet indicateur ^a	a	a	tout type ^a
0x02	Octet de verrouillage du bloc de mémoire 1. Les octets verrouillables dans le bloc de mémoire doivent être en lecture seule, tandis que l'octet de verrouillage a une valeur différente de 0x55.	0xFF	0xFF ^c	RAM-RW
0x03	Octet GTIN fabricant d'origine 0 (octet de poids fort)	0xFF	pas de modification	NVM-RW (verrouillable)
0x04	Octet GTIN fabricant d'origine 1	0xFF	pas de modification	NVM-RW (verrouillable)

Adresse	Description	Valeur par défaut (valeur d'origine)	Valeur réinitialisée ^b	Type de mémoire
0x05	Octet GTIN fabricant d'origine 2	0xFF	pas de modification	NVM-RW (verrouillable)
0x06	Octet GTIN fabricant d'origine 3	0xFF	pas de modification	NVM-RW (verrouillable)
0x07	Octet GTIN fabricant d'origine 4	0xFF	pas de modification	NVM-RW (verrouillable)
0x08	Octet GTIN fabricant d'origine 5 (octet de poids faible)	0xFF	pas de modification	NVM-RW (verrouillable)
0x09	Octet de numéro d'identification fabricant d'origine 0 (octet de poids fort)	0xFF	pas de modification	NVM-RW (verrouillable)
0x0A	Octet de numéro d'identification fabricant d'origine 1	0xFF	pas de modification	NVM-RW (verrouillable)
0x0B	Octet de numéro d'identification fabricant d'origine 2	0xFF	pas de modification	NVM-RW (verrouillable)
0x0C	Octet de numéro d'identification fabricant d'origine 3	0xFF	pas de modification	NVM-RW (verrouillable)
0x0D	Octet de numéro d'identification fabricant d'origine 4	0xFF	pas de modification	NVM-RW (verrouillable)
0x0E	Octet de numéro d'identification fabricant d'origine 5	0xFF	pas de modification	NVM-RW (verrouillable)
0x0F	Octet de numéro d'identification fabricant d'origine 6	0xFF	pas de modification	NVM-RW (verrouillable)
0x10	Octet de numéro d'identification fabricant d'origine 7 (octet de poids faible)	0xFF	pas de modification	NVM-RW (verrouillable)
≥ 0x11	Informations supplémentaires concernant les dispositifs de commande ^a	a	a	a
0xFF	Réservée – non mise en œuvre	réponse NO	pas de modification	n.a.
Légende GTIN code article international LSB octet de poids faible MSB octet de poids fort OEM fabricant d'origine ^a L'objet, la valeur par défaut, la valeur de mise sous tension, la valeur réinitialisée et le type de mémoire de ces octets doivent être définis par le fabricant. ^b Réinitialisation de la valeur après une commande "RESET MEMORY BANK". ^c Également utilisée comme valeur de mise sous tension.				

Il convient d'utiliser les octets des emplacements 0x03 à 0x08 ("GTIN fabricant d'origine 0" à "GTIN fabricant d'origine 5") pour identifier le produit qui contient le dispositif de commande. Lorsque les octets sont utilisés pour le GTIN, ils doivent être mémorisés en commençant par le bit de poids fort et doivent contenir des zéros non significatifs. Il convient que ces octets soient programmés par le fabricant d'origine.

Il convient que les octets des emplacements 0x09 à 0x10 ("Octet de numéro d'identification fabricant d'origine 0" à "Octet de numéro d'identification fabricant d'origine 7") contiennent 64 bits d'un numéro d'identification du produit du fabricant d'origine. Lorsque les octets sont utilisés pour le numéro d'identification, ils doivent être mémorisés avec l'octet de poids faible dans l'octet de numéro d'identification 7" et les bits non utilisés doivent être remplis de 0. Il convient que ces octets soient programmés par le fabricant d'origine.

Il convient que la combinaison du GTIN fabricant d'origine et du numéro d'identification fabricant d'origine soit unique.

9.11.9 Blocs de mémoire spécifiques au fabricant

Le fabricant peut utiliser des blocs de mémoire supplémentaires dans la plage de 2 à 199 afin de mémoriser des informations supplémentaires. La carte de la mémoire des blocs supplémentaires doit être conforme au Tableau 12.

9.11.10 Blocs de mémoire réservés

Les blocs de mémoire 200 à 255 sont réservés pour une utilisation future, ou sont décrits dans la série IEC 62386-3xx. Aucune mise en œuvre ou utilisation autre que celles décrites dans la série IEC 62386-3xx n'est admise.

9.12 Réinitialisation

9.12.1 Opération de réinitialisation

Un dispositif de commande doit mettre en œuvre une opération de réinitialisation afin de régler toutes les variables de dispositif et d'instance (voir Tableau 19 et Tableau 20) sur leurs valeurs réinitialisées.

NOTE Pour certaines variables, cette opération peut n'avoir strictement aucun impact.

L'opération de réinitialisation doit durer 300 ms au maximum. Au cours de l'exécution de l'opération de réinitialisation, le dispositif de commande peut ou non répondre aux commandes. Toutefois, tant que l'opération de réinitialisation n'est pas achevée, il n'est pas nécessaire de définir une valeur pour les variables concernées, quelles qu'elles soient.

Un contrôleur d'application peut déclencher l'opération de réinitialisation au moyen de l'instruction "RESET" et il convient que celui-ci attende au moins 350 ms pour s'assurer que tous les dispositifs de commande ont achevé l'opération de réinitialisation.

9.12.2 Opération de réinitialisation des blocs de mémoire

Un dispositif de commande doit effectuer une opération de réinitialisation afin de régler le contenu de tous les blocs de mémoire non verrouillés sur leurs valeurs réinitialisées (voir 9.11), suivie par le verrouillage des blocs de mémoire.

NOTE Pour certains emplacements de blocs de mémoire, cette opération peut n'avoir strictement aucun impact.

L'opération de réinitialisation des blocs de mémoire doit durer 10 s au maximum. Au cours de l'exécution de cette opération de réinitialisation, le dispositif de commande peut ou non répondre aux commandes. Toutefois, cette opération de réinitialisation des blocs de mémoire n'est pas achevée, aucun des emplacements de mémoire concernés n'a de valeur définie.

Un contrôleur d'application peut déclencher l'opération de réinitialisation pour un bloc de mémoire spécifique, ou pour tous les blocs de mémoire mis en œuvre, au moyen de l'instruction "RESET MEMORY BANK (DTR0)", et il convient que celui-ci attende au moins 10,1 s pour s'assurer que tous les dispositifs de commande disposent d'un délai suffisant pour achever l'opération de réinitialisation des blocs de mémoire.

9.13 Comportement lors de la mise sous tension

9.13.1 Mise sous tension

Après un cycle de mise sous tension externe (voir 4.11.1 et 4.11.5 de l'IEC 62386-101:2022), le dispositif doit conserver sa configuration la plus récente des variables NVM sauvegardées conformément au 9.18, avec les exceptions suivantes:

- toutes les variables mentionnées dans le Tableau 19 et le Tableau 20 doivent être réglées à la valeur indiquée dans la colonne "valeur de mise sous tension". Les variables marquées "pas de modification" dans la colonne "valeur de mise sous tension" ne doivent pas être prises en compte. Les variables définies dans les parties de la série IEC 62386-3xx mises en œuvre doivent être incluses;
- l'état d'autorisation d'écriture des blocs de mémoire doit être désactivé pour tous les blocs de mémoire et l'octet de verrouillage doit être réglé sur 0xFF;
- le mode repos doit être annulé (voir 9.10.4);
- toutes les minuteries actives doivent être arrêtées et annulées (réinitialisées);
- "*powerCycleSeen*" doit être réglé sur TRUE.

"*powerCycleSeen*" peut être observé par "QUERY DEVICE STATUS".

Afin d'observer un cycle de mise sous tension ultérieur, il convient que le contrôleur d'application supprime "*powerCycleSeen*", au moyen de la commande "RESET POWER CYCLE SEEN".

NOTE Dans un système avec plusieurs contrôleurs d'application qui utilisent les informations du cycle de mise sous tension des autres dispositifs de commande dans le système, les contrôleurs d'application peuvent les prendre en compte lors de la suppression de "*powerCycleSeen*".

9.13.2 Notification du cycle de mise sous tension

Après achèvement de son comportement de mise sous tension conformément au 9.13.1, une unité de bus doit générer un message d'événement de cycle de mise sous tension par dispositif si "*powerCycleNotification*" est ENABLED pour au moins une de ses unités logiques.

Un contrôleur d'application peut utiliser "ENABLE POWER CYCLE NOTIFICATION" et "DISABLE POWER CYCLE NOTIFICATION" pour activer/désactiver les événements de cycle de mise sous tension pour des dispositifs de commande spécifiques. "*powerCycleNotification*" doit être DISABLED par défaut.

NOTE 1 La notification du cycle de mise sous tension n'est pas empêchée par "*applicationActive*", ni par "*instanceActive*".

L'événement doit être généré à l'aide du message "POWER NOTIFICATION (*device*)" comme cela est décrit en 11.2. Le message d'événement doit être envoyé une seule fois en appliquant la priorité 2 et avec un retard uniformément réparti entre 1,3 s et 5,0 s après achèvement de la procédure de mise sous tension.

NOTE 2 L'application d'un retard aléatoire permet d'éviter les collisions de notifications de cycle de mise sous tension.

9.14 Utilisation prioritaire

9.14.1 Généralités

Les priorités des trames en avant ont pour objet de faciliter le comportement approprié d'un système à plusieurs maîtres. Les priorités assurent que les transmissions pour une réaction du système prioritaire prévalent sur les transmissions pour un fonctionnement du système non prioritaire.

- La première trame en avant d'une transaction (voir 9.3 de l'IEC 62386-101:2022) doit être envoyée avec une priorité de 2 à 5. Toutes les autres trames en avant d'une transaction doivent être envoyées avec la priorité 1.
- Les trames en avant qui ne font pas partie d'une transaction doivent être envoyées avec une priorité de 2 à 5.
- Il convient d'utiliser la priorité 2 pour exécuter les actions initiées par les utilisateurs pour la commutation et la gradation des lumières. Ceci implique des messages d'événement et

des instructions de niveau appropriés. La priorité 2 peut également être utilisée lors de la mise en service (adressage, par exemple).

EXEMPLE 1 Les actions de commutation ou de gradation déclenchées par l'intermédiaire d'un bouton-poussoir ou d'un détecteur de présence.

- Il convient d'utiliser la priorité 3 pour la configuration d'une unité de bus et pour les messages d'événement non couverts par les priorités 2 et 4.

EXEMPLE 2 L'écriture dans les blocs de mémoire ou les événements rétroactifs.

- Il convient d'utiliser la priorité 4 pour exécuter les actions automatiques pour la commutation et la gradation des lumières. Cela signifie l'envoi de messages d'événement et d'instructions de niveau appropriés.

EXEMPLE 3 Les actions de commutation ou de gradation déclenchées par un capteur de luminosité.

- Il convient d'utiliser la priorité 5 pour les commandes de requête périodiques.

9.14.2 Priorité des notifications d'entrée

Une instance doit utiliser une *"eventPriority"* par défaut égale à la priorité 4 lors de la transmission d'un message d'événement afin de générer une "INPUT NOTIFICATION (*device/instance, event*)". Pour les types d'instances particuliers, cette priorité par défaut est soumise à une modification par les parties applicables de la série IEC 62386-3xx.

Dans un système, l'*"eventPriority"* par défaut peut être annulée par le contrôleur d'application en utilisant "SET EVENT PRIORITY (*DTR0*)". "QUERY EVENT PRIORITY" peut être utilisée pour observer l'*"eventPriority"* actuellement active.

9.15 Attribution d'adresses courtes

9.15.1 Généralités

"shortAddress" doit être déduite de *data* ou *"DTR0"* selon la commande utilisée. Elle doit être définie à l'exécution de "PROGRAM SHORT ADDRESS (*data*)" ou "SET SHORT ADDRESS (*DTR0*)" comme suit:

- si *data* ou *"DTR0"* = MASK: MASK (l'adresse courte est effectivement supprimée);
- si *data* ou *"DTR0"* < 0x40: *data* ou *"DTR0"*;
- dans tous les autres cas: pas de modification.

9.15.2 Affectation d'adresses aléatoires

Un dispositif de commande doit mettre en œuvre un état d'initialisation qui active uniquement, hormis les autres opérations identifiées dans le présent document, un ensemble de commandes qui permettent à un contrôleur d'application de détecter et d'identifier de manière unique les dispositifs de commande disponibles sur le bus et d'attribuer des adresses courtes à ces dispositifs.

L'état d'initialisation est un état provisoire activé au moyen de la commande "INITIALISE (*device*)". Il doit s'interrompre automatiquement dans un délai de 15 min ± 1,5 min après exécution de la dernière commande "INITIALISE (*device*)". De plus, un cycle de mise sous tension ou la commande "TERMINATE" doit conduire le dispositif de commande à quitter immédiatement l'état d'initialisation.

Le dispositif de commande doit comporter trois valeurs possibles pour *"initialisationState"*:

- DISABLED, dans un état autre que l'état d'initialisation;
- ENABLED, dans l'état d'initialisation;
- WITHDRAWN, dans l'état d'initialisation, effectivement identifié et retiré.

Les commandes (spéciales) suivantes sont des commandes d'initialisation:

- "RANDOMISE", "COMPARE" et "WITHDRAW";
- "SEARCHADDRH (*data*)", "SEARCHADDRM (*data*)" et "SEARCHADDRL (*data*)";
- "PROGRAM SHORT ADDRESS (*data*)", "VERIFY SHORT ADDRESS (*data*)" et "QUERY SHORT ADDRESS";
- "IDENTIFY DEVICE".

NOTE "IDENTIFY DEVICE" n'est pas en soi une commande d'initialisation, mais elle est généralement utilisée lors de l'initialisation.

9.15.3 Identification d'un dispositif

Aucune variable ne doit être affectée par la procédure d'identification. Le processus normal doit continuer, c'est-à-dire que les événements peuvent être générés par une modification de la valeur d'entrée. Cela ne doit pas interrompre l'identification. Le cas échéant, les variables peuvent être temporairement ignorées, de sorte que lorsque l'identification est terminée, aucun effet secondaire n'est constaté.

L'identification doit s'interrompre à l'exécution de toute instruction autre que INITIALISE (*device*) ou "IDENTIFY DEVICE".

L'identification peut être lancée par l'envoi de l'instruction "IDENTIFY DEVICE". Cela doit lancer ou relancer une minuterie de 10 s ± 1 s. Alors que la minuterie est en fonctionnement, une procédure qui permet à un observateur d'identifier le dispositif de commande sélectionné doit être exécutée. En cas d'expiration de la minuterie, l'identification doit s'interrompre.

NOTE La procédure réelle est spécifique au fabricant.

Lorsqu'une identification est interrompue par un contrôleur d'application, la minuterie correspondante doit être annulée immédiatement.

9.16 Traitement des exceptions

Les dispositifs de commande et les instances doivent indiquer si une erreur s'est produite par réglage (en cas d'erreur) et par réinitialisation (en l'absence d'erreur) des balises suivantes.

- Un contrôleur d'application doit modifier "*applicationControllerError*". Cet état peut faire l'objet d'une requête par "QUERY DEVICE STATUS" (voir 9.17.2). Des informations d'erreur détaillées peuvent être obtenues par "QUERY APPLICATION CONTROLLER ERROR" (voir 11.6.4).
- Un dispositif de commande qui n'est pas un contrôleur d'application doit avoir "*applicationControllerError*" réglée sur FALSE.
- Un dispositif d'entrée doit modifier "*inputDeviceError*" pour indiquer une erreur dans l'une des instances de dispositif d'entrée de l'unité logique. Cet état peut faire l'objet d'une requête par "QUERY DEVICE STATUS" (voir 9.17.2). Des informations d'erreur détaillées peuvent être obtenues par "QUERY INPUT DEVICE ERROR" (voir 11.6.5).
- Un dispositif de commande qui n'est pas un dispositif d'entrée doit avoir "*inputDeviceError*" réglée sur FALSE.
- Une instance doit modifier "*instanceError*". Cet état peut faire l'objet d'une requête par "QUERY INSTANCE STATUS" (voir 9.17.3). Des informations d'erreur détaillées peuvent être obtenues par "QUERY INSTANCE ERROR" (voir 11.9.4).

9.17 Informations de capacités et d'état du dispositif

9.17.1 Capacités du dispositif

Chaque dispositif de commande doit présenter ses caractéristiques sous la forme d'une combinaison de capacités de dispositif comme cela est indiqué dans le Tableau 15:

Tableau 15 – Capacités du dispositif de commande

Bit	Description	Valeur	Voir
0	"applicationControllerPresent" est TRUE?	"1" = "Yes"	9.3.3
1	"numberOfInstances" est supérieur à 0?	"1" = "Yes"	9.5.2
2	"applicationControllerAlwaysActive" est TRUE?	"1" = "Yes"	9.10.2
3 à 4	Réservé pour l'IEC 62386-104 (voir [2])	"0" = valeur par défaut	
5	Au moins une instance prend en charge la configuration d'"instanceType" ou "instanceConfiguration[]"	"1" = "Yes"	9.19
6 à 7	non utilisé	"0" = valeur par défaut	

Les capacités du dispositif peuvent faire l'objet d'une requête en utilisant "QUERY DEVICE CAPABILITIES".

9.17.2 État du dispositif

Chaque dispositif de commande doit présenter son état sous la forme d'une combinaison de propriétés de dispositif comme cela est indiqué dans le Tableau 16:

Tableau 16 – État du dispositif de commande

Bit	Description	Valeur	Voir
0	"inputDeviceError" est TRUE?	"1" = "Yes"	9.16
1	"quiescentMode" est ENABLED?	"1" = "Yes"	9.10.4
2	"shortAddress" est la valeur MASK?	"1" = "Yes"	9.15.1
3	"applicationActive" est TRUE?	"1" = "Yes"	9.10.1
4	"applicationControllerError" est TRUE?	"1" = "Yes"	9.16
5	"powerCycleSeen" est TRUE?	"1" = "Yes"	9.13
6	"resetState" est TRUE?	"1" = "Yes"	Voir ci-dessous
7	Réservé pour l'IEC 62386-104 (voir [2])	"0" = valeur par défaut	

L'état du dispositif peut faire l'objet d'une requête en utilisant "QUERY DEVICE STATUS".

Bit 6: "resetState" doit être réglé sur TRUE si toutes les variables NVM mentionnées dans le Tableau 19 et le Tableau 20 présentent leur valeur réinitialisée. Les variables NVM qui comportent le marquage "pas de modification" dans la colonne propre à la valeur réinitialisée ne doivent pas être prises en considération. Les variables NVM définies dans les parties de la série IEC 62386-3xx mises en œuvre doivent être incluses. Dans tous les autres cas, le bit doit être réglé sur FALSE.

9.17.3 État d'instance

Chaque instance doit présenter son état sous la forme d'une combinaison de propriétés d'instance comme cela est indiqué dans le Tableau 17:

Tableau 17 – État d'instance

Bit	Description	Valeur	Voir
0	"instanceError" est TRUE?	"1" = "Yes"	9.16
1	"instanceActive" est TRUE?	"1" = "Yes"	9.10.3
2 à 7	non utilisé	"0" = valeur par défaut	

L'état d'instance peut être observé en utilisant "QUERY INSTANCE STATUS".

9.18 Mémoire non volatile

Après toute modification d'une variable NVM, la nouvelle valeur doit être restaurée après un cycle de mise sous tension, à condition qu'une période d'au moins 30 s ait fait suite à la modification de la variable NVM, avant le cycle de mise sous tension.

Si la condition ci-dessus n'est pas remplie, il est possible que la variable NVM modifiée ne soit pas restaurée avec la nouvelle valeur.

NOTE La méthode utilisée pour s'assurer que les variables NVM sont sauvegardées est spécifique au fabricant. Une méthode possible consiste à sauvegarder certaines variables NVM dans les 30 s qui suivent leur modification, les autres étant sauvegardées au moment où la panne d'alimentation externe se produit. La méthode choisie est susceptible d'influencer la durée de vie de la mémoire non volatile.

9.19 Types et configuration d'instances

Les instances peuvent éventuellement prendre en charge une modification d'"instanceType" avec l'instruction "SET INSTANCE TYPE (DTR0)".

Le type d'instance actuel peut être déterminé en utilisant "QUERY INSTANCE TYPE", et tous les types d'instances disponibles peuvent être découverts au moyen de "QUERY AVAILABLE INSTANCE TYPES".

Les instances peuvent éventuellement prendre en charge une modification d'"instanceConfiguration[]" avec l'instruction "SET INSTANCE CONFIGURATION (DTR0, DTR2:DTR1)".

La configuration d'instance actuelle peut être découverte au moyen de "QUERY INSTANCE CONFIGURATION (DTR0)".

Toutes les valeurs normalisées d'"instanceConfiguration[]" sont publiées dans les parties appropriées de la série IEC 62386-3xx. Une plage d'"instanceConfiguration[]" spécifique au fabricant peut être utilisée, mais l'utilisation de telles valeurs ne doit supprimer aucune exigence du présent document.

Si la configuration d'instance est prise en charge, le manuel doit indiquer les emplacements d'"instanceConfiguration[]" pris en charge, doit décrire l'utilisation de ces emplacements dans le cas où ceux-ci se situent dans la plage spécifique au fabricant, et doit indiquer leur type de mémoire (accessible en lecture et/ou en écriture).

EXEMPLE 1 Modification du type d'instance: une instance avec des entrées sans tension peut prendre en charge une modification de type d'instance entre le type 1 (bouton-poussoir) et le type 2 (dispositif d'entrée absolu/de commutation).

EXEMPLE 2 Modification de la configuration d'instance: une instance de détecteur de présence avec plusieurs technologies de capteurs (détecteur infrarouge passif, capteur à hyperfréquences/Doppler et capteur ultrasonore, par exemple) peut permettre la configuration des technologies de capteurs actuellement actives.

9.20 Configuration actuelle de l'unité de bus

Le cas échéant, la valeur de configuration actuelle de l'unité de bus issue du bloc de mémoire 0 doit indiquer la mise en œuvre actuellement active:

- de la valeur d'"*applicationControllerPresent*".

Pour la valeur donnée de configuration actuelle de l'unité de bus, la mise en œuvre doit être conforme au Tableau 18.

Tableau 18 – Configuration actuelle de l'unité de bus

Configuration actuelle de l'unité de bus	" <i>applicationControllerPresent</i> "	Description
0 à 191 ^a		Réservé
192 à 255		Spécifique au fabricant
^a Valeurs réservées pour de futures mises à jour du présent document.		

Une méthode spécifique au fabricant est admise pour la sélection de la configuration actuelle. Après modification, l'unité de bus peut exiger un certain temps, un cycle de mise sous tension ou toute autre exigence avant que la nouvelle configuration devienne active.

Après modification de la configuration, la valeur de "Configuration actuelle de l'unité de bus" dans le bloc de mémoire 0 doit indiquer la configuration active, et le fonctionnement de l'unité de bus doit être mis à jour conformément à la nouvelle configuration.

10 Déclaration des variables

Le Tableau 19 indique les valeurs par défaut, les valeurs réinitialisées, le domaine de validité et le type de mémoire des variables définies indépendantes des instances.

Le Tableau 20 indique les valeurs par défaut, les valeurs réinitialisées, le domaine de validité et le type de mémoire des variables définies de chaque instance.

Les variables déclarées dans le présent Article 10 ne doivent pas pouvoir être saisies dans un bloc de mémoire.

Tableau 19 – Déclaration des variables de dispositif

Variable	Valeur par défaut (valeur d'origine)	Valeur réinitialisée	Valeur de mise sous tension	Plage de validité	Type de mémoire
"shortAddress"	MASK (pas d'adresse)	pas de modification	pas de modification	[0,63], MASK	NVM
"deviceGroups"	0x0000 0000	0x0000 0000	pas de modification	[0,0xFFFF FFFF]	NVM
"searchAddress"	a	0xFF FF FF	0xFF FF FF	[0,0xFF FF FF]	RAM
"randomAddress"	0xFF FF FF	0xFF FF FF	pas de modification	[0,0xFF FF FF]	NVM
"DTR0"	a	pas de modification	0x00	[0,0xFF]	RAM
"DTR1"	a	pas de modification	0x00	[0,0xFF]	RAM
"DTR2"	a	pas de modification	0x00	[0,0xFF]	RAM
"numberOfInstances"	rodage en usine	pas de modification	pas de modification	[0,32]	ROM
"operatingMode"	rodage en usine	pas de modification	pas de modification	0, [0x80,0xFF]	NVM
"quiescentMode"	a	DISABLED	DISABLED	[ENABLED, DISABLED]	RAM
"applicationActive"	Application Controller Present	pas de modification	pas de modification	[TRUE, FALSE]	NVM
"writeEnableState"	a	DISABLED	DISABLED	[ENABLED, DISABLED]	RAM
"applicationControllerPresent"	rodage en usine	pas de modification	pas de modification	[TRUE, FALSE]	ROM
"applicationControllerAlwaysActive"	rodage en usine	pas de modification	pas de modification	[TRUE, FALSE]	ROM
"powerCycleSeen"	a	FALSE	TRUE	[TRUE, FALSE]	RAM
"powerCycleNotification"	DISABLED	pas de modification	pas de modification	[ENABLED, DISABLED]	NVM
"initialisationState"	a	pas de modification	DISABLED	[ENABLED, DISABLED, WITHDRAWN]	RAM
"applicationControllerError"	a	c	FALSE ^b	[TRUE, FALSE]	RAM
"inputDeviceError"	a	c	FALSE ^b	[TRUE, FALSE]	RAM
"resetState"	TRUE	TRUE	TRUE ^b	[TRUE, FALSE]	RAM
"eventPriority" ^d	4	pas de modification	pas de modification	[2,5]	NVM
"versionNumber"	3.0	pas de modification	pas de modification	00001100b	ROM
^a Non applicable. ^b Il convient que la valeur reflète la situation réelle dès que possible. ^c La valeur peut varier à la suite de l'exécution de la commande RESET. ^d Cette variable est indépendante de la ou des variables d'instance "eventPriority".					

Tableau 20 – Déclaration des variables d'instance

Variable	Valeur par défaut (valeur d'origine)	Valeur réinitialisée	Valeur de mise sous tension	Plage de validité	Type de mémoire
"instanceGroup0"	MASK	MASK	pas de modification	[0,31], MASK	NVM
"instanceGroup1"	MASK	MASK	pas de modification	[0,31], MASK	NVM
"instanceGroup2"	MASK	MASK	pas de modification	[0,31], MASK	NVM
"instanceActive"	TRUE	pas de modification	pas de modification	[TRUE, FALSE]	NVM
"instanceType"	rodage en usine	pas de modification	pas de modification	[0,31]	ROM ou NVM
"resolution"	rodage en usine	pas de modification	pas de modification	[1 255]	ROM
"inputValue"	^a	pas de modification	pas de modification ^b	$[0, 2^N - 1]^c$	RAM
"instanceNumber"	rodage en usine	pas de modification	pas de modification	[0, "numberOfInstances" – 1]	ROM
"eventFilter" ^d	0xFF FF FF	0xFF FF FF	pas de modification	[0, 0xFF FF FF]	NVM
"eventScheme"	0	0	pas de modification	[0,4]	NVM
"eventPriority" ^{d f}	4	pas de modification	pas de modification	[2,5]	NVM
"instanceError"	^a	^e	FALSE ^b	[TRUE, FALSE]	RAM
"instanceConfiguration[]" ^g	rodage en usine	pas de modification	pas de modification	[0, 0xFFFF] ^g	NVM ou ROM

^a Non applicable.
^b Il convient que la valeur reflète la situation réelle dès que possible.
^c *N* est calculé comme ("resolution"/8) arrondi à l'entier le plus proche.
^d Pour les types d'instances particuliers, les valeurs qui appartiennent à cette variable peuvent être modifiées par la partie applicable de la série IEC 62386-3xx.
^e La valeur peut varier à la suite de l'exécution de la commande RESET.
^f La ou les variables d'instance "eventPriority" sont indépendantes de la variable de dispositif "eventPriority".
^g La taille de cette matrice est spécifique au fabricant, de 0 à 256 emplacements. La taille maximale des emplacements mis en œuvre est de 16 bits.

11 Définition des commandes

11.1 Généralités

Les codes de fonctionnement non utilisés sont réservés à des besoins futurs.

11.2 Fiches de vue d'ensemble

Les informations du Tableau 21 au Tableau 24 donnent une vue d'ensemble des messages d'événement, des commandes normalisées et des commandes spéciales du dispositif de commande.