

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**



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 It also gives information on projects, replaced and withdrawn publications.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

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.

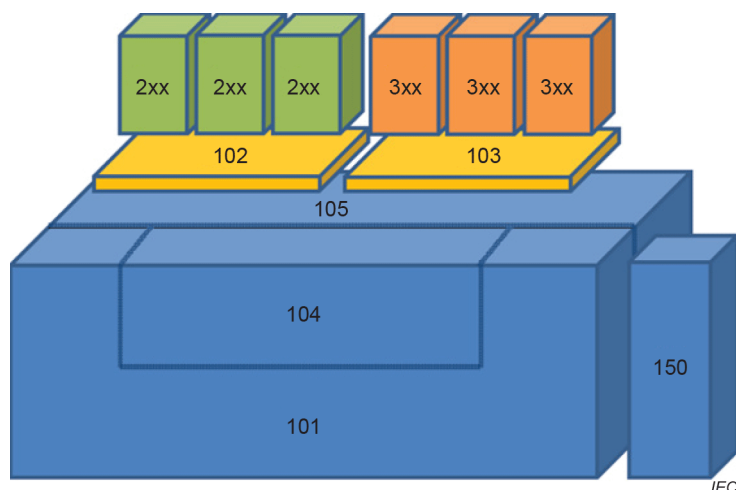


Figure 1 – IEC 62386 graphical overview

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".

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 $[0, "numberOfInstances" - 1]$.

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 through a memory bank.