

INTERNATIONAL STANDARD



Digital addressable lighting interface – Part 102: General requirements – Control gear



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.

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 It also gives information on projects, replaced and withdrawn publications.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



IEC 62386-102

Edition 3.0 2022-11
COMMENTED VERSION

INTERNATIONAL STANDARD



**Digital addressable lighting interface –
Part 102: General requirements – Control gear**

INTERNATIONAL
ELECTROTECHNICAL
COMMISSION

ICS 29.140.50; 29.140.99

ISBN 978-2-8322-6103-3

Warning! Make sure that you obtained this publication from an authorized distributor.

CONTENTS

FOREWORD	7
INTRODUCTION	10
1 Scope	12
2 Normative references	12
3 Terms and definitions	12
4 General	15
4.1 General	15
4.2 Version number	15
5 Electrical specification	16
6 Interface Bus power supply	16
7 Transmission protocol structure	16
7.1 General	16
7.2 16-bit forward frame encoding	16
7.2.1 General	16
7.2.2 Address byte	16
7.2.3 Opcode byte	17
8 Timing	17
9 Method of operation	17
9.1 General	17
9.2 Control gear	17
9.2.1 General	17
9.2.2 Control gear phases	17
9.3 Dimming curve	18
9.4 Calculating " <i>targetLevel</i> "	22
9.5 Fading	22
9.5.1 General	22
9.5.2 Fade time	23
9.5.3 Fade rate	25
9.5.4 Extended fade time	26
9.5.5 Using the fade time	28
9.5.6 Using the fade rate	28
9.5.7 System response to changes during a fade	29
9.5.8 System response to changes during standby and startup	29
9.5.9 Stopping a fade	29
9.6 Min and max level	29
9.7 Commands	30
9.7.1 General	30
9.7.2 Level instructions without fade	31
9.7.3 Level instructions initiating a fade	31
9.7.4 Configuration instructions	31
9.7.5 Queries	31
9.7.6 Special commands	31
9.7.7 Application extended commands	31

9.8	Command iterations.....	32
9.8.1	General.....	32
9.8.2	Command iteration of "UP" and "DOWN" commands.....	32
9.8.3	DAPC SEQUENCE (deprecated).....	32
9.9	Modes of operation.....	33
9.9.1	General.....	33
9.9.2	Operating mode 0x00: standard mode.....	33
9.9.3	Operating mode 0x01 to 0x7F: reserved.....	33
9.9.4	Operating mode 0x80 to 0xFF: manufacturer-specific modes.....	33
9.10	Memory banks.....	34
9.10.1	General.....	34
9.10.2	Memory map.....	34
9.10.3	Selecting a memory bank location.....	35
9.10.4	Protectable memory locations.....	35
9.10.5	Memory bank reading.....	36
9.10.6	Memory bank writing.....	37
9.10.7	Memory bank 0.....	38
9.10.8	Memory bank 1 (optional).....	41
9.10.9	Manufacturer-specific memory banks.....	43
9.10.10	Reserved memory banks.....	43
9.11	Reset.....	43
9.11.1	Reset operation.....	43
9.11.2	Reset memory bank operation.....	43
9.12	System failure.....	44
9.13	Power on.....	44
9.14	Assigning short addresses.....	45
9.14.1	General.....	45
9.14.2	Random address allocation.....	45
9.14.3	Identification of a device.....	46
9.14.4	Direct address allocation.....	47
9.15	Failure state behaviour.....	47
9.16	Status information.....	47
9.16.1	General.....	47
9.16.2	Bit 0: Control gear failure.....	48
9.16.3	Bit 1: lamp failure.....	48
9.16.4	Bit 2: lamp on.....	50
9.16.5	Bit 3: limit error.....	50
9.16.6	Bit 4: fade running.....	50
9.16.7	Bit 5: reset state.....	50
9.16.8	Bit 6: missing short address.....	50
9.16.9	Bit 7: power cycle seen.....	50
9.17	Non-volatile memory.....	50
9.18	Device types and features.....	51
9.19	Using scenes.....	51
9.20	Current bus unit configuration.....	52
10	Declaration of variables.....	53

11	Definition of commands	56
11.1	General	56
11.2	Overview sheets	56
11.3	Level instructions	61
11.3.1	DAPC (<i>level</i>)	61
11.3.2	OFF	61
11.3.3	UP	61
11.3.4	DOWN	61
11.3.5	STEP UP	62
11.3.6	STEP DOWN	62
11.3.7	RECALL MAX LEVEL	62
11.3.8	RECALL MIN LEVEL	63
11.3.9	STEP DOWN AND OFF	63
11.3.10	ON AND STEP UP	63
11.3.11	ENABLE DAPC SEQUENCE	64
11.3.12	GO TO LAST ACTIVE LEVEL	64
11.3.13	CONTINUOUS UP	64
11.3.14	CONTINUOUS DOWN	64
11.3.15	GO TO SCENE (<i>sceneNumber</i>)	64
11.4	Configuration instructions	64
11.4.1	General	64
11.4.2	RESET	65
11.4.3	STORE ACTUAL LEVEL IN DTR0	65
~~11.4.4~~	~~SAVE PERSISTENT VARIABLES~~	~~65~~
11.4.4	SET OPERATING MODE (<i>DTR0</i>)	65
11.4.5	RESET MEMORY BANK (<i>DTR0</i>)	65
11.4.6	IDENTIFY DEVICE	65
11.4.7	SET MAX LEVEL (<i>DTR0</i>)	66
11.4.8	SET MIN LEVEL (<i>DTR0</i>)	66
11.4.9	SET SYSTEM FAILURE LEVEL (<i>DTR0</i>)	66
11.4.10	SET POWER ON LEVEL (<i>DTR0</i>)	67
11.4.11	SET FADE TIME (<i>DTR0</i>)	67
11.4.12	SET FADE RATE (<i>DTR0</i>)	67
11.4.13	SET EXTENDED FADE TIME (<i>DTR0</i>)	67
11.4.14	SET SCENE (<i>DTR0</i> , <i>sceneX</i>)	68
11.4.15	REMOVE FROM SCENE (<i>sceneX</i>)	68
11.4.16	ADD TO GROUP (<i>group</i>)	68
11.4.17	REMOVE FROM GROUP (<i>group</i>)	68
11.4.18	SET SHORT ADDRESS (<i>DTR0</i>)	68
11.4.19	ENABLE WRITE MEMORY	69
11.5	Queries	69
11.5.1	General	69
11.5.2	QUERY STATUS	69
11.5.3	QUERY CONTROL GEAR PRESENT	69
11.5.4	QUERY CONTROL GEAR FAILURE	69
11.5.5	QUERY LAMP FAILURE	69
11.5.6	QUERY LAMP POWER ON	69

11.5.7	QUERY LIMIT ERROR.....	69
11.5.8	QUERY RESET STATE	69
11.5.9	QUERY MISSING SHORT ADDRESS	69
11.5.10	QUERY VERSION NUMBER	70
11.5.11	QUERY CONTENT DTR0	70
11.5.12	QUERY DEVICE TYPE	70
11.5.13	QUERY NEXT DEVICE TYPE	70
11.5.14	QUERY PHYSICAL MINIMUM.....	70
11.5.15	QUERY POWER FAILURE.....	71
11.5.16	QUERY CONTENT DTR1	71
11.5.17	QUERY CONTENT DTR2	71
11.5.18	QUERY OPERATING MODE.....	71
11.5.19	QUERY LIGHT SOURCE TYPE	71
11.5.20	QUERY ACTUAL LEVEL.....	72
11.5.21	QUERY MAX LEVEL.....	72
11.5.22	QUERY MIN LEVEL.....	72
11.5.23	QUERY POWER ON LEVEL	72
11.5.24	QUERY SYSTEM FAILURE LEVEL	72
11.5.25	QUERY FADE TIME/FADE RATE.....	72
11.5.26	QUERY EXTENDED FADE TIME	72
11.5.27	QUERY MANUFACTURER SPECIFIC MODE	72
11.5.28	QUERY SCENE LEVEL (<i>sceneX</i>)	72
11.5.29	QUERY GROUPS 0-7	73
11.5.30	QUERY GROUPS 8-15	73
11.5.31	QUERY RANDOM ADDRESS (H).....	73
11.5.32	QUERY RANDOM ADDRESS (M)	73
11.5.33	QUERY RANDOM ADDRESS (L)	73
11.5.34	READ MEMORY LOCATION (<i>DTR1</i> , <i>DTR0</i>).....	73
11.6	Application extended commands	73
11.6.1	General.....	73
11.6.2	QUERY EXTENDED VERSION NUMBER.....	74
11.7	Special commands.....	74
11.7.1	General.....	74
11.7.2	TERMINATE.....	74
11.7.3	DTR0 (<i>data</i>).....	74
11.7.4	INITIALISE (<i>device</i>).....	74
11.7.5	RANDOMISE	75
11.7.6	COMPARE	75
11.7.7	WITHDRAW	75
11.7.8	SEARCHADDRH (<i>data</i>).....	75
11.7.9	SEARCHADDRM (<i>data</i>).....	76
11.7.10	SEARCHADDRL (<i>data</i>)	76
11.7.11	PROGRAM SHORT ADDRESS (<i>data</i>).....	76
11.7.12	VERIFY SHORT ADDRESS (<i>data</i>).....	76
11.7.13	QUERY SHORT ADDRESS.....	76
11.7.14	ENABLE DEVICE TYPE (<i>data</i>)	77
11.7.15	DTR1 (<i>data</i>).....	77
11.7.16	DTR2 (<i>data</i>).....	77

11.7.17	WRITE MEMORY LOCATION (<i>DTR1</i> , <i>DTR0</i> , <i>data</i>)	77
11.7.18	WRITE MEMORY LOCATION – NO REPLY (<i>DTR1</i> , <i>DTR0</i> , <i>data</i>).....	78
11.7.19	PING.....	78
~~12~~	~~Test procedures.....~~	~~78~~
Annex A	(informative) Examples of algorithms.....	79
A.1	Random address allocation.....	79
A.2	One single control gear connected to the control device.....	79
A.3	Using application extended commands.....	80
Annex B	(normative) High resolution dimmer.....	81
Bibliography		83
List of comments		84
Figure 1	– IEC 62386 graphical overview	11
Figure 2	– Control gear directly operating a light source	17
Figure 3	– Dimming curve.....	19
Figure 4	– Level over time, fading up and down	23
Figure 5	– Timing and response when executing command iteration	32
Figure 6	– Correlation between " <i>lampFailure</i> ", " <i>lampOn</i> " and " <i>fadeRunning</i> " bits	49
Figure B.1	– Level behaviour in case of off-grid starting points	82
Table 1	– 16-bit command frame encoding.....	16
Table 2	– Dimming curve tolerance.....	20
Table 3	– Dimming curve.....	21
Table 4	– Fade times.....	25
Table 5	– Fade rates	25
Table 6	– Extended fade time – Base value.....	27
Table 7	– Extended fade time – Multiplier	27
Table 8	– Memory types	34
Table 9	– Basic memory map of memory banks.....	35
Table 10	– Memory map of memory bank 0.....	38
Table 11	– Memory map of memory bank 1	42
Table 12	– Power on timing	45
Table 13	– Control gear status.....	47
Table 14	– Scenes	52
Table 15	– Current bus unit configuration.....	53
Table 16	– Declaration of variables.....	53
Table 17	– Standard commands	56
Table 18	– Special commands	60
Table 19	– Light source type encoding.....	71
Table 20	– Device addressing with "INITIALISE (<i>device</i>)"	74

INTERNATIONAL ELECTROTECHNICAL COMMISSION

DIGITAL ADDRESSABLE LIGHTING INTERFACE –

Part 102: General requirements – Control gear

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.

This commented version (CMV) of the official standard IEC 62386-102:2022 edition 3.0 allows the user to identify the changes made to the previous IEC 62386-102:2014+AMD1:2018 CSV edition 2.1. Furthermore, comments from IEC TC 34 experts are provided to explain the reasons of the most relevant changes, or to clarify any part of the content.

A vertical bar appears in the margin wherever a change has been made. Additions are in green text, deletions are in strikethrough red text. Experts' comments are identified by a blue-background number. Mouse over a number to display a pop-up note with the comment.

This publication contains the CMV and the official standard. The full list of comments is available at the end of the CMV.

IEC 62386-102 has been prepared by IEC technical committee 34: Lighting. It is an International Standard.

This third edition cancels and replaces the second 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:

- the scope has been updated;
- references have been updated;
- memory bank reading of multi-byte values has been added;
- memory bank 0 and common memory bank requirements have been updated;
- reserved memory banks have been updated;
- non-volatile memory (NVM) save time has been added, and SAVE PERSISTENT VARIABLES removed;
- version number has been updated;
- bus unit configuration has been added.

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

Draft	Report on voting
34/948/FDIS	34/989/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 102 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 Part 2xx (particular requirements for control gear) containing clauses to supplement or modify the corresponding clauses in Part 101 and Part 102 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~~ third edition of IEC 62386-102 is intended to be used in conjunction with IEC 62386-101:2014 and IEC 62386-101:2014/AMD1:2018 and with the various parts that make up the IEC 62386-2xx series for control gear, and can be used together with IEC 62386-103:2014 and IEC 62386-103:2014/AMD1:2018 and the various parts that make up the IEC 62386-3xx series of particular requirements for control devices. 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.

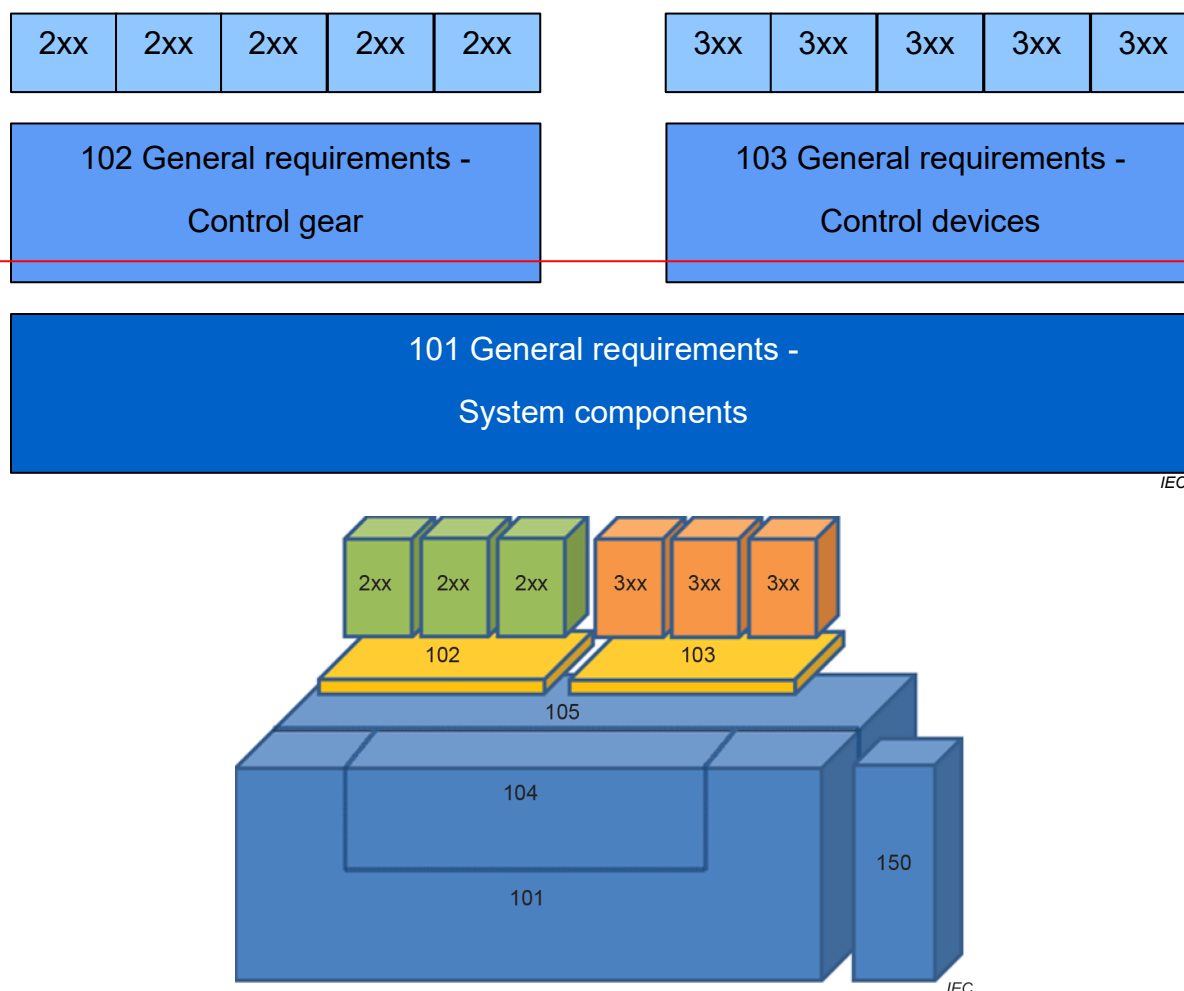


Figure 1 – IEC 62386 graphical overview 1

When this part of IEC 62386 refers to any of the clauses of the other ~~two~~ parts of the IEC 62386-1xx series, the extent to which such a clause is applicable ~~and the order in which the tests are to be performed are~~ 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 XXXXXXXXb or in the format XXXX XXXX, where X is 0 or 1 and "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 102: General requirements – Control gear

1 Scope

This part of IEC 62386 is applicable to control gear ~~in a bus system~~ for control by digital signals of electronic lighting equipment ~~which is in line with the requirements of IEC 61347 (all parts), with the addition of DC supplies.~~

~~NOTE Tests in this standard are type tests. Requirements for testing individual control gear during production are not included.~~

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 62386-101:20142022, *Digital addressable lighting interface – Part 101: General requirements – System components*
~~IEC 62386-101:2014/AMD1:2018~~

IEC 62386-103:20142022, *Digital addressable lighting interface – Part 103: General requirements – Control devices*
~~IEC 62386-103:2014/AMD1:2018~~

IEC 62386-2xx (all parts), *Digital addressable lighting interface – Part 2xx: Particular requirements for control gear*

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

actual level

value representing the current light output

3.2

arc power

power supplied to the light sources (lamps)

3.3

broadcast

type of address used to ~~simultaneously~~ address all control gear in the system ~~at once~~

3.4**broadcast unaddressed**

type of address used to **simultaneously** address all control ~~devices~~ gear in the system that have no short address ~~at once~~

3.5**DAPC****direct arc power control**

method to directly control the light output

~~Note 1 to entry: The note to entry in French concerns the French text only.~~

3.6**DTR****data transfer register**

multipurpose register used to exchange data

~~Note 1 to entry: The note to entry in French concerns the French text only.~~

3.7**group address**

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

3.8**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]).

~~Note 3 to entry: The note 3 to entry in French concerns the French text only.~~

3.9**identification**

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

3.10**level**

8-bit value

3.11**MASK**

~~the value 0xFF~~

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 0xFFFFFF.

3.12**monotonic**

~~a function f defined on a subset of the real numbers with real values is called monotonically non-decreasing, if for all x and y such that $x \leq y$ one has $f(x) \leq f(y)$, so f preserves the order. Likewise, a function is called monotonically non-increasing if, whenever $x \leq y$, then $f(x) \geq f(y)$, so it reverses the order. For this standard monotonic is defined as either monotonically non-decreasing or monotonically non-increasing~~

3.12

NO

answer ~~used to deny or refuse~~ 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 62386-101:2014 ~~and IEC 62386-101:2014/AMD1:2018~~ 2022, 8.2.5.

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

3.13

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

3.14

NVM-RO

NVM that cannot be written using any command

3.15

NVM-RW

NVM that can be modified using one or more commands

3.16

opcode

operation code

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

3.17

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 gear, including its required reaction to commands

Note 1 to entry: Control gear ~~may~~ can support more than one operating mode.

3.18

PHM

physical minimum level

level corresponding to the minimum light output the control gear can operate at

3.19

RAM

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

3.20

RAM-RO

RAM that cannot be written using any command

3.21

RAM-RW

RAM that can be modified using one or more commands

3.22

random address

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

Note 1 to entry: Clause A.1 provides an example of how the search and random addresses are used

3.23**reset state**

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

3.24**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.

3.25**scene**

configurable preset level

3.26**search address**

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

Note 1 to entry: Clause A.1 provides an example of how the search and random addresses are used.

3.27**short address**

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

3.28**startup**

time needed to change from lamp off to normal operation of the lamp or failure state

Note 1 to entry: This time includes preheat and ignition.

3.29**strictly monotonic**

~~a function f defined on a subset of the real numbers with real values is called monotonically increasing, if for all x and y such that $x < y$ one has $f(x) < f(y)$, so f preserves the order. Likewise, a function is called monotonically decreasing if, whenever $x < y$, then $f(x) > f(y)$, so it reverses the order. For this standard strictly monotonic is defined as either monotonically increasing or monotonically decreasing~~

either entirely increasing or decreasing without repeating values

3.30**target level**

value representing the target light output expected after completion of the current level command

3.31**TMASK 2**

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.

3.32**YES**

answer ~~used to accept or affirm~~ to a query where a backward frame of MASK is sent

~~Note 1 to entry: If a query is asked where the answer is YES, the response will be a backward frame containing the value of MASK.~~

4 General

4.1 General

The requirements of IEC 62386-101:~~2014 and IEC 62386-101:2014/AMD1:2018~~2022, Clause 4 apply, with the restrictions, changes and additions identified below.

4.2 Version number

This subclause replaces IEC 62386-101:~~2014 and IEC 62386-101:2014/AMD1:2018~~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 62386-102 the minor version number shall be incremented by one.

At a new edition of IEC 62386-102 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 16.

NOTE ~~Normally 2 amendments on IEC documents are made before a new edition is created.~~ IEC documents are generally subject to two amendments before a new edition is prepared.

5 Electrical specification

The requirements of IEC 62386-101:~~2014 and IEC 62386-101:2014/AMD1:2018~~2022, Clause 5 apply.

6 ~~Interface~~ Bus power supply

If a bus power supply is integrated into a control gear, the requirements of IEC 62386-101:~~2014 and IEC 62386-101:2014/AMD1:2018~~2022, Clause 6 apply.

7 Transmission protocol structure

7.1 General

The requirements of IEC 62386-101:~~2014 and IEC 62386-101:2014/AMD1:2018~~2022, Clause 7 apply, with the following additions.

7.2 16-bit forward frame encoding

7.2.1 General

For commands, the 16-bit forward frame shall be encoded as depicted in Table 1.

Table 1 – 16-bit command frame encoding

Bytes/Bits								Device addressing method	
Address byte							Opcode byte		
15	14	13	12	11	10	9	8 ^a		
0	64 short addresses						x		Short addressing
1	0	0	16 group addresses				x		Group addressing
1	1	1	1	1	1	0	x		Broadcast unaddressed
1	1	1	1	1	1	1	x		Broadcast
1010 0000 to 1100 1011								Special command	
1100 1100 to 1111 1011								Reserved	
^a Selector bit, see 7.2.2; 0 indicates DAPC, 1 indicates other command.									

7.2.2 Address byte

The address byte provides

- the method of device addressing used by the application controller;
- the type of command transmitted in the opcode byte;
 - Bit 8 = '1': standard command;
 - Bit 8 = '0': direct arc power control (DAPC) command;
- address spaces for special commands;
- reserved device addresses.

Reserved addresses should not be used by the application controller.

7.2.3 Opcode byte

The opcode byte provides

- for DAPC commands, the requested light output;
- for standard commands, the opcode;
- command-specific information for special commands;
- reserved information for reserved commands.

8 Timing

The requirements of IEC 62386-101:2014 and IEC 62386-101:2014/AMD1:2018²⁰²², Clause 8 apply.

9 Method of operation

9.1 General

The requirements of IEC 62386-101:2014 and IEC 62386-101:2014/AMD1:2018²⁰²², Clause 9 apply with the following additions.

9.2 Control gear

9.2.1 General

Control gear ~~may~~ receive commands from application controllers. Application controllers are specified in IEC 62386-103:2014 and IEC 62386-103:2014/AMD1:2018 ~~2022~~.

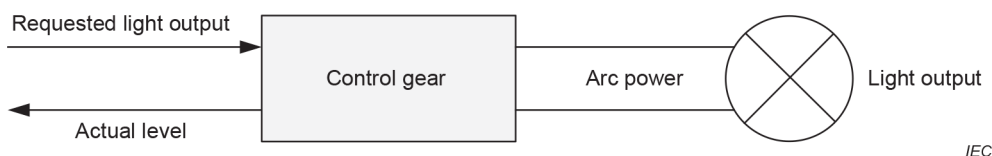


Figure 2 – Control gear directly operating a light source

Figure 2 shows how the various levels lead to light output. The maximum (light) output level of a control gear is referred to as 100 %. All levels are specified in a relative way. Physically there is a minimum that the control gear can supply whilst there is still light output. This is known as the physical minimum level (PHM).

NOTE PHM is control gear specific, and is greater than 0.

9.2.2 Control gear phases

9.2.2.1 General

Depending on the light source, various phases of operation can be identified within a control gear. See 9.2.2.2 through 9.2.2.5.

9.2.2.2 Standby

During this phase, the lamp is off and only in this phase both "*targetLevel*" and "*actualLevel*" are 0.

9.2.2.3 Startup

Startup is a transitional phase changing from standby to normal operation or failure. This phase is sometimes noticeable as a delay. Examples are:

- preheat: the lamp is heated to prepare for ignition. This is typically seen for fluorescent light sources;
- ignition: the lamp is ignited. This is typically seen for high intensity discharge (HID) light sources and fluorescent light sources after preheat;
- power stage preparation.

For further information and exceptions refer to 9.16.3.

9.2.2.4 Normal operation

While in normal operation the lamp is emitting light and can be operated as expected.

For further information and exceptions refer to 9.16.3.

9.2.2.5 Failure

During the failure phase the lamp cannot be operated as expected.

For further information and exceptions refer to 9.16.3.

9.3 Dimming curve

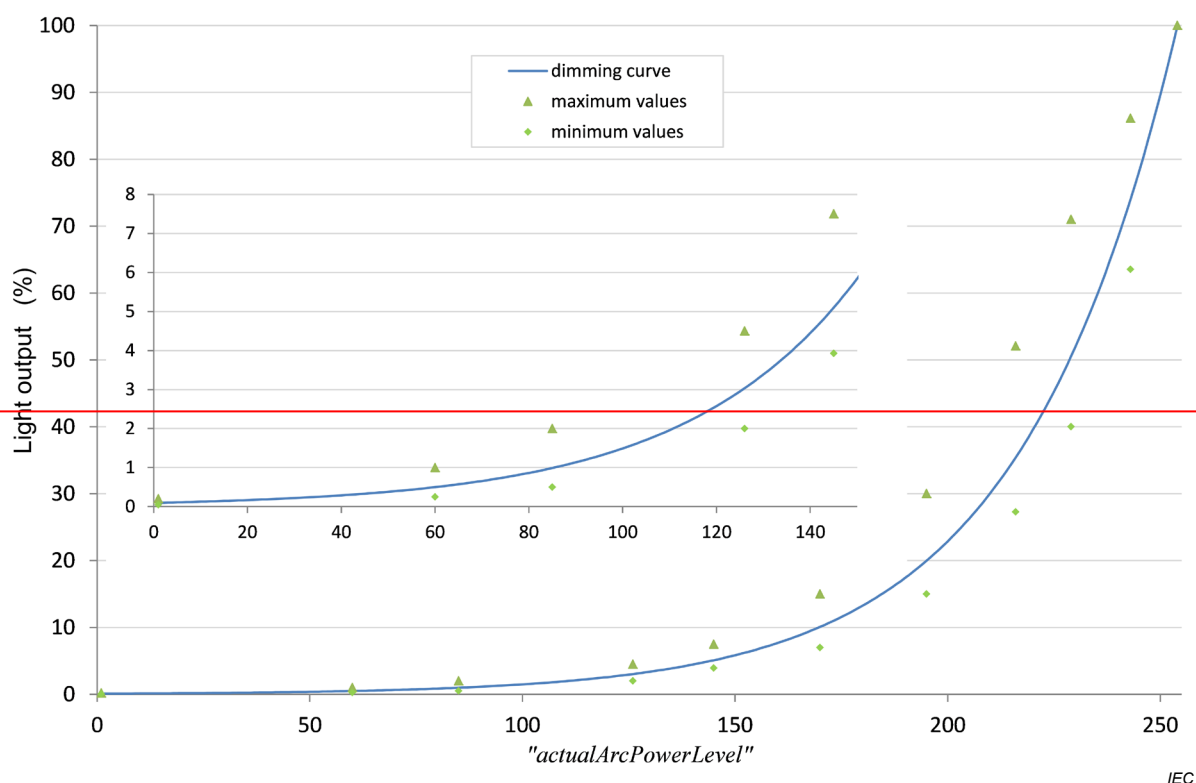
The dimming curve determines how the level shall be translated into light output.

An "actualLevel" greater than or equal to 1 and less than or equal to 254 shall be translated into light output according to

$$\text{Light output (actualLevel)} = 10^{\frac{\text{actualLevel} - 1}{253/3} - 1} \% .$$

Light output is expressed relative to the maximum possible light output of a given control-gear-lamp combination. The dimming curve starts at 0,1 % for "actualLevel" equal to 0x01 and ends at 100 % for "actualLevel" equal to 0xFE. The dimming curve is strictly monotonic, and the relative accuracy shall be $\pm\frac{1}{2}$ step. This shall be tested using a fade, excluding PHM.

NOTE 1 The dimming curve is intended to compensate the light sensitivity curve of the human eye.



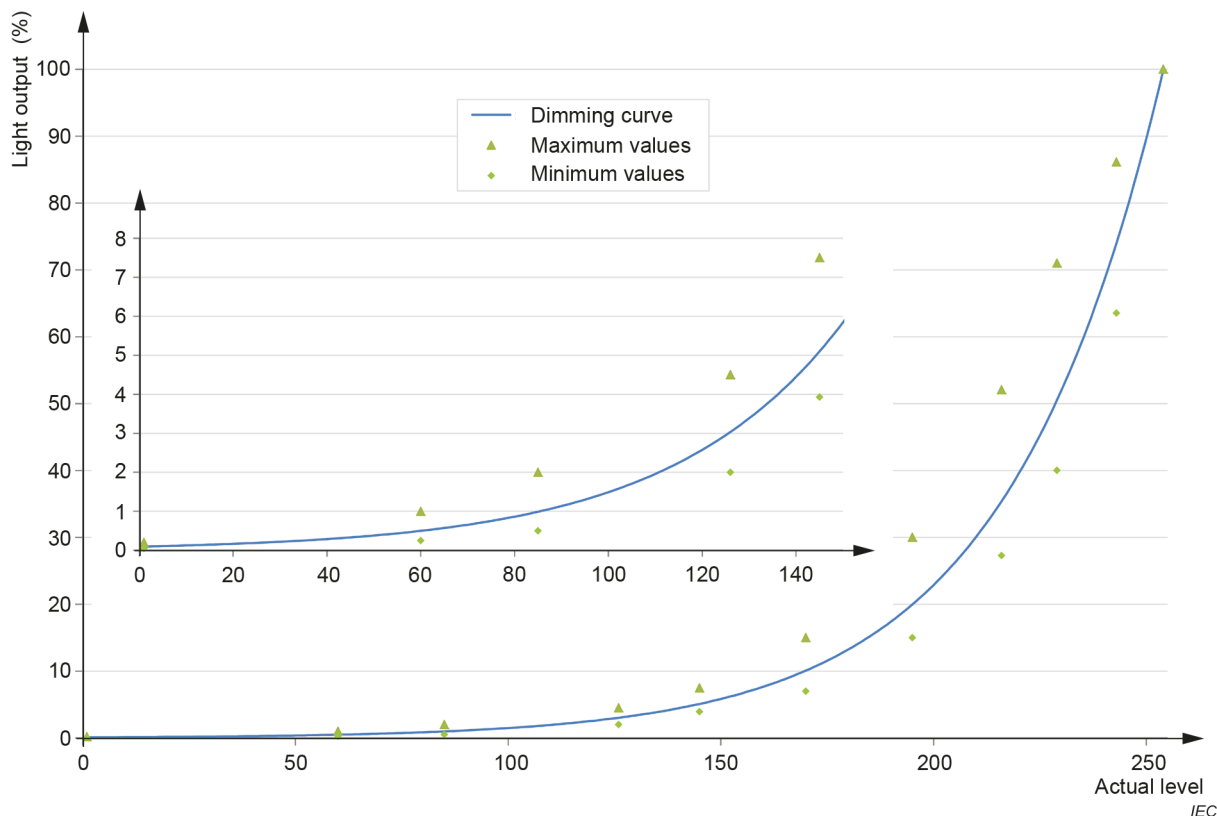


Figure 3 – Dimming curve

The accuracy of light output is specified by the test points given in Table 2. The test points of Table 2 and the dimming curve are depicted in Figure 3, and Table 3 shows the light output versus the level. The lamp type or load used during testing shall be stated for reproducibility.

~~NOTE 2 The minimum and maximum values are based on the test values that can be found in IEC 62386-102:2009.~~

Table 2 – Dimming curve tolerance

Level	1	60	85	126	145	170	195	216	229	243	254
Minimum value	0,05	0,25	0,50	2,00	3,93	7,00	15,00	27,28	40,00	63,53	
Nominal value	0,10	0,50	0,99	3,04	5,10	10,09	19,97	35,43	50,53	74,05	100,00
Maximum value	0,20	1,00	2,00	4,50	7,50	15,0	30,00	52,09	71,00	86,14	
Tolerances are given in % and rounded to two decimal places.											

Table 3 – Dimming curve

Level	Light output	Level	Light output	Level	Light output	Level	Light output	Level	Light output
1	0,100	52	0,402	103	1,620	154	6,520	205	26,241
2	0,103	53	0,414	104	1,665	155	6,700	206	26,967
3	0,106	54	0,425	105	1,711	156	6,886	207	27,713
4	0,109	55	0,437	106	1,758	157	7,076	208	28,480
5	0,112	56	0,449	107	1,807	158	7,272	209	29,269
6	0,115	57	0,461	108	1,857	159	7,473	210	30,079
7	0,118	58	0,474	109	1,908	160	7,680	211	30,911
8	0,121	59	0,487	110	1,961	161	7,893	212	31,767
9	0,124	60	0,501	111	2,015	162	8,111	213	32,646
10	0,128	61	0,515	112	2,071	163	8,336	214	33,550
11	0,131	62	0,529	113	2,128	164	8,567	215	34,479
12	0,135	63	0,543	114	2,187	165	8,804	216	35,433
13	0,139	64	0,559	115	2,248	166	9,047	217	36,414
14	0,143	65	0,574	116	2,310	167	9,298	218	37,422
15	0,147	66	0,590	117	2,374	168	9,555	219	38,457
16	0,151	67	0,606	118	2,440	169	9,820	220	39,522
17	0,155	68	0,623	119	2,507	170	10,091	221	40,616
18	0,159	69	0,640	120	2,577	171	10,371	222	41,740
19	0,163	70	0,658	121	2,648	172	10,658	223	42,895
20	0,168	71	0,676	122	2,721	173	10,953	224	44,083
21	0,173	72	0,695	123	2,797	174	11,256	225	45,303
22	0,177	73	0,714	124	2,874	175	11,568	226	46,557
23	0,182	74	0,734	125	2,954	176	11,888	227	47,846
24	0,187	75	0,754	126	3,035	177	12,217	228	49,170
25	0,193	76	0,775	127	3,119	178	12,555	229	50,531
26	0,198	77	0,796	128	3,206	179	12,902	230	51,930
27	0,203	78	0,819	129	3,294	180	13,260	231	53,367
28	0,209	79	0,841	130	3,386	181	13,627	232	54,844
29	0,215	80	0,864	131	3,479	182	14,004	233	56,362
30	0,221	81	0,888	132	3,576	183	14,391	234	57,922
31	0,227	82	0,913	133	3,675	184	14,790	235	59,526
32	0,233	83	0,938	134	3,776	185	15,199	236	61,173
33	0,240	84	0,964	135	3,881	186	15,620	237	62,866
34	0,246	85	0,991	136	3,988	187	16,052	238	64,607
35	0,253	86	1,018	137	4,099	188	16,496	239	66,395
36	0,260	87	1,047	138	4,212	189	16,953	240	68,233
37	0,267	88	1,076	139	4,329	190	17,422	241	70,121
38	0,275	89	1,105	140	4,449	191	17,905	242	72,062
39	0,282	90	1,136	141	4,572	192	18,400	243	74,057
40	0,290	91	1,167	142	4,698	193	18,909	244	76,107
41	0,298	92	1,200	143	4,828	194	19,433	245	78,213

Level	Light output	Level	Light output	Level	Light output	Level	Light output	Level	Light output
42	0,306	93	1,233	144	4,962	195	19,971	246	80,378
43	0,315	94	1,267	145	5,099	196	20,524	247	82,603
44	0,324	95	1,302	146	5,240	197	21,092	248	84,889
45	0,332	96	1,338	147	5,385	198	21,675	249	87,239
46	0,342	97	1,375	148	5,535	199	22,275	250	89,654
47	0,351	98	1,413	149	5,688	200	22,892	251	92,135
48	0,361	99	1,452	150	5,845	201	23,526	252	94,686
49	0,371	100	1,492	151	6,007	202	24,177	253	97,307
50	0,381	101	1,534	152	6,173	203	24,846	254	100,000
51	0,392	102	1,576	153	6,344	204	25,534		

9.4 Calculating "targetLevel"

An application controller instructs the control gear on the requested light output and on the behaviour during the transition from the "actualLevel" to the "targetLevel" by means of appropriate opcodes.

The "targetLevel" shall be calculated on the basis of the requested light output as follows:

- 0x00 shall be accepted as "targetLevel" and turn off the light.
- Any value between 0x01 and "minLevel" shall result in "targetLevel" = "minLevel".
- Any value between "maxLevel" and 0xFE shall result in "targetLevel" = "maxLevel".
- "MASK" shall have no effect on "targetLevel" except when a fade is running, see 9.5.9.
- All other values shall be accepted as "targetLevel".

The requested target level calculation of "targetLevel" shall also be applied if the request is based on an internally stored value, such as a scene, "powerOnLevel", or "systemFailureLevel".

On every change of "targetLevel", with the exception of the initialisation caused by a power cycle, the control gear shall update "limitError" (see 9.16.5) and shall set "lastLightLevel" to the new "targetLevel". If "targetLevel" is not 0x00, "lastActiveLevel" shall be set to "targetLevel".

9.5 Fading

9.5.1 General

Fading is a linear transition in time from "actualLevel" to "targetLevel". The "actualLevel", and thus the light output, shall be strictly monotonic according to the applicable dimming curve.

A fade can be started in two ways:

- using a fade time: this sets a time to use for the fade process;
- using a fade rate: this sets a speed to use for the fade process.

A fade shall not be started if the calculated "targetLevel" is equal to "actualLevel".

When a fade starts, the fade timer shall be started and "fadeRunning" shall be set to TRUE (see 9.16.6).

During the fade, the light output shall be maintained as close to the ideal fading curve as possible.

During a process of fading up, "*actualLevel*" shall be incremented at a time corresponding to the intersection of an ideal fading curve with the mid-point between "*actualLevel*" and "*actualLevel*" + 1. Likewise, when fading down, "*actualLevel*" shall be decremented at a time corresponding to the intersection of an ideal fading curve with the mid-point between "*actualLevel*" and "*actualLevel*" – 1. Figure 4 illustrates this, and applies for fades started using either fade time or fade rate.

Measurements of fade time or fade rate shall start after the stop condition of the command that triggers it. If the fade takes place immediately after startup, measurement shall be done from the moment "*lampOn*" is TRUE or, in case of total lamp failure, from the moment "*lampFailure*" is confirmed TRUE. A fade shall automatically end when the fade timer has been active for the applicable fade time. At this point the fade timer shall be stopped and "*fadeRunning*" shall be set to FALSE (see 9.16.6).

This means that the control gear fades to the target level even in the case of a total lamp ~~error~~ failure. If a lamp is to be switched off at the end of the fade, the step from "*minLevel*" to 0x00 shall not contribute to the fade time. The step from "*minLevel*" to 0x00 shall be taken immediately after the fade time has elapsed.

If a lamp is to be lit at the beginning of the fade and dimmed to a certain value, the step from 0x00 to "*minLevel*" shall not contribute to the fade time. This means that the fade time starts when the startup phase has finished.

NOTE The transition from 0x00 to "*minLevel*" incorporates startup.

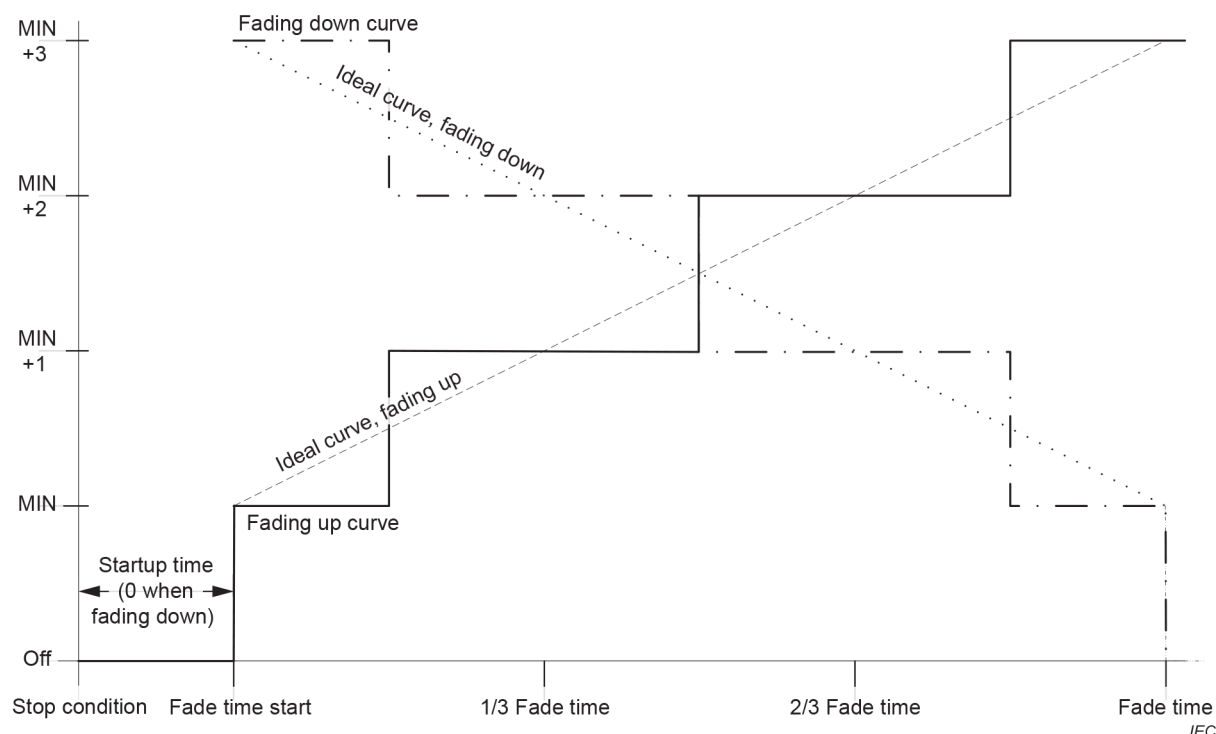


Figure 4 – Level over time, fading up and down

Testing shall be done with "*minLevel*" $\geq$ PHM + 1. For further information, see Annex B.

9.5.2 Fade time

The fade time shall be in accordance with Table 4:

"*fadeTime*" shall be set on receipt of the command "SET FADE TIME (DTR0)". "*fadeTime*" can be queried using QUERY FADE TIME/FADE RATE.

The "*fadeTime*" shall be set to a value according to the following steps:

- if "*DTR0*" > 15: 15
- in all other cases: "*DTR0*"

The fade time shall be calculated on the basis of "*fadeTime*" as follows:

- if "*fadeTime*" = 0: use the extended fade time (see 9.5.4);
- if "*fadeTime*" is in the range [1,15]: $2^{\left(\frac{"fadeTime"}{2}-1\right)}$ s **3**.

<i>"fadeRate"</i>	Minimum fade rate steps/s	Nominal fade rate steps/s	Maximum fade rate steps/s
1	322	358	394
2	228	253	278
3	161	179	197
4	114	127	139
5	80,5	89,4	98,4
6	56,9	63,3	69,6
7	40,3	44,7	49,2
8	28,5	31,6	34,8
9	20,1	22,4	24,6
10	14,2	15,8	17,4
11	10,1	11,2	12,3
12	7,1	7,9	8,7
13	5,0	5,6	6,1
14	3,6	4,0	4,3
15	2,5	2,8	3,1

• ~~Extended fade time~~

• ~~if "*fadeTime*" is in the range [1,15]: $\frac{1}{2} \cdot \sqrt{2^{"fadeTime"}} \cdot 1$ s~~

Table 4 lists the possible fade time values.

Table 4 – Fade times

<i>"fadeTime"</i>	<i>Minimum fade time</i> s	<i>Nominal fade time</i> s	<i>Maximum fade time</i> s
0	Extended fade		
1	0,6	0,7	0,8
2	0,9	1,0	1,1
3	1,3	1,4	1,6
4	1,8	2,0	2,2
5	2,5	2,8	3,1
6	3,6	4,0	4,4
7	5,1	5,7	6,2
8	7,2	8,0	8,8
9	10,2	11,3	12,4
10	14,4	16,0	17,6
11	20,4	22,6	24,9
12	28,8	32,0	35,2
13	40,7	45,3	49,8
14	57,6	64,0	70,4
15	81,5	90,5	99,6

9.5.3 Fade rate

The fade rate shall be in accordance with Table 5.

"fadeRate" shall be set on receipt of the command "SET FADE RATE (DTR0)". *"fadeRate"* can be queried using QUERY FADE TIME/FADE RATE.

The *"fadeRate"* shall be set to a value according to the following steps:

- if *"DTR0"* > 15: 15
- if *"DTR0"* = 0: 1
- in all other cases: *"DTR0"*

The fade rate shall be calculated on the basis of *"fadeRate"* as follows:

$$\text{Fade-rate} = \frac{506}{\sqrt{2^{\text{"fadeRate"}}}} \text{ steps/s.}$$

$$\text{Fade rate} = \frac{506}{2^{\left(\frac{\text{"fadeRate"}}{2}\right)}} \text{ steps/s.}$$

Table 5 lists the possible fade rate values.

Table 5 – Fade rates

<i>"fadeRate"</i>	<i>Minimum fade rate</i> steps/s	<i>Nominal fade rate</i> steps/s	<i>Maximum fade rate</i> steps/s
1	322	358	394
2	228	253	278
3	161	179	197
4	114	127	139
5	80,5	89,4	98,4
6	56,9	63,3	69,6
7	40,3	44,7	49,2
8	28,5	31,6	34,8
9	20,1	22,4	24,6
10	14,2	15,8	17,4
11	10,1	11,2	12,3
12	7,1	7,9	8,7
13	5,0	5,6	6,1
14	3,6	4,0	4,3
15	2,5	2,8	3,1

9.5.4 Extended fade time

If *"fadeTime"* equals 0, and the fast fade time as defined in IEC 62386-207 is implemented and equals 0, the extended fade time shall be used.

The extended fade time can be set using a base value and a multiplier according to Table 6 and Table 7. The extended fade time can be calculated based on the base value and the multiplication factor.

$$\text{Fade time} = \text{extendedFadeTimeBase} \cdot \text{extendedFadeTimeMultiplier}$$

This yields a range of 100 ms to 16 min, and a special value indicating no fade (as quickly as possible).

Table 6 – Extended fade time – Base value

Base bits	Base value
0000b	1
0001b	2
0010b	3
0011b	4
0100b	5
0101b	6
0110b	7
0111b	8
1000b	9
1001b	10
1011b	11
1011b	12
1100b	13
1101b	14
1110b	15
1111b	16

Table 7 – Extended fade time – Multiplier

Multiplier bits	Multiplication factor		
	Minimum	Nominal	Maximum
000b	0 ms ^a	0 ms ^a	0 ms ^a
001b	95 ms	100 ms	105 ms
010b	0,95 s	1 s	1,05 s
011b	9,5 s	10 s	10,5 s
100b	0,95 min	1 min	1,05 min
101b		Reserved	
110b		Reserved	
111b		Reserved	
^a No fade (as quickly as possible).			

On execution of "SET EXTENDED FADE TIME (*DTR0*)" the control gear shall set the following values based on "*DTR0*". The format used shall be 0YYYYAAAAb, where YYYb equals the fade time multiplier, and AAAAb the fade time base: The resulting fade time shall be monotonically increasing when the base time increases.

- If "*DTR0*" > 0100 1111b:
 - "*extendedFadeTimeBase*" shall be set to 0;
 - "*extendedFadeTimeMultiplier*" shall be set to 0 ms, effectively setting the fade time to 0 s meaning no fade (as quickly as possible). The transition from "*actualLevel*" to "*targetLevel*" shall take place immediately and the light output shall be adjusted as quickly as possible, meaning less than 0,8 s which represents the maximum fade time for "*fadeTime*" = 1 (see Table 4).

- In all other cases:
 - "*extendedFadeTimeBase*" shall be set to AAAAb;
 - "*extendedFadeTimeMultiplier*" shall be set to YYYYb.

The extended fade time can be queried using "QUERY EXTENDED FADE TIME". The answer shall be 0 YYY AAAAb, where YYYYb equals "*extendedFadeTimeMultiplier*" and AAAAb equals "*extendedFadeTimeBase*".

9.5.5 Using the fade time

Commands that use a fade time shall start a fade using the applicable fade time. This time can be determined based on the following rules:

- If "*fadeTime*" > 0: see Table 4.
- If "*fadeTime*" = 0: The extended fade time shall be used, see Table 6 and Table 7. The extended fade time can be calculated by multiplying the base value and the multiplier.
- If "*extendedFadeTimeMultiplier*" = 0 ms, the fade time equals 0 s, meaning no fade (as quickly as possible). The transition from "*actualLevel*" to "*targetLevel*" shall take place immediately and the light output shall be adjusted as quickly as possible.

The target level shall be calculated on the basis of the command parameter. After the fade time has expired, the calculated target level shall be reached.

~~Since the extended fade time also supports fade times below 0,7 s that might not be realised by all control gear and light source combinations, such control gear may simply adjust the light output as quickly as possible when an extended fade time is requested that it physically cannot support. However, it should respond as if the fade has finished within the requested time.~~

The extended fade time also supports fade times below 0,7 s. If a fade time faster than 0,7 s is started, and that fade time cannot be realised by the control gear and light source combination, such control gear shall adjust the light output as quickly as possible. However, the control gear shall respond as if the fade has finished within the requested time. **5**

9.5.6 Using the fade rate

9.5.6.1 Fading with "UP" and "DOWN" commands

Commands "UP" and "DOWN" shall start a 200 ms ± 20 ms fade.

"*targetLevel*" shall be calculated on the basis of the "*actualLevel*" using the applicable fade rate. After the 200 ms fade has expired, the calculated target level shall be reached.

NOTE 1 Since the fade rate is used, it is possible to reach "*minLevel*" or "*maxLevel*" before the end of the fade. This does not result in the "*fadeRunning*" bit being cleared.

NOTE 2 Because there are fade rate tolerances, different control gear can react to commands that use the fade rate at slightly different effective rates. Consequently, after the processing of these relative dimming commands, different gear ~~might~~ can have different values for "*targetLevel*" (and therefore also for "*actualLevel*" and "*lastLightLevel*").

9.5.6.2 Fading with "CONTINUOUS UP" and "CONTINUOUS DOWN" commands

Command "CONTINUOUS UP" shall set "*targetLevel*" to "*maxLevel*" and start a fade using the applicable fade rate. The fade shall stop when "*maxLevel*" is reached.

Command "CONTINUOUS DOWN" shall set "*targetLevel*" to "*minLevel*" and start a fade using the applicable fade rate. The fade shall stop when "*minLevel*" is reached.

Upon execution of either a "CONTINUOUS UP" or "CONTINUOUS DOWN" instruction at least one step shall be made, unless this is precluded by the values of "*minLevel*" or "*maxLevel*".

Refer to 9.5.9 for stopping a fade before reaching "*minLevel*" or "*maxLevel*".

NOTE 1 In contrast to the "UP" and "DOWN" instructions, it is not possible to reach "*minLevel*" or "*maxLevel*" before the end of the fade. Therefore, "*fadeRunning*" bit is going to be cleared at the end of a fade.

NOTE 2 Similar to the "UP" and "DOWN" instructions, different control gear ~~might~~ can end up with different values for "*targetLevel*", "*actualLevel*" and "*lastLightLevel*" after the fade has been stopped ahead of time (e.g. via DAPC(MASK)).

9.5.7 System response to changes during a fade

If "*fadeTime*", "*extendedFadeTimeBase*", "*extendedFadeTimeMultiplier*" and/or "*fadeRate*" is changed during a running fade, then the running fade shall finish without the fade time and/or fade rate being recalculated. The next fade shall use the recalculated values.

9.5.8 System response to changes during standby and startup

If a fade is initiated during standby, the fade process shall be pended with "*actualLevel*" equal to "*minLevel*" during the startup phase. The reaction to level commands shall be the same as if the lamp(s) were operating at "*minLevel*".

If a fade is initiated during startup, the fade process shall be pended at "*actualLevel*". The reaction to level commands shall be the same as if the lamp(s) were operating at "*actualLevel*".

The fade shall start:

- as soon as "*lampOn*" is TRUE, or
- in the case of total lamp failure, as soon as "*lampFailure*" is confirmed TRUE.

For further information on "*lampOn*" and "*lampFailure*" see 9.16.3 and 9.16.4.

9.5.9 Stopping a fade

Any command setting one or more of the following variables

- "*targetLevel*", "*minLevel*", "*maxLevel*"

as well as the execution of one of the following commands

- "DAPC(MASK)", ~~"SAVE PERSISTENT VARIABLES"~~, "IDENTIFY DEVICE"

shall stop a running fade.

NOTE 1 The fade stops even if the value of the affected variable does not change.

When a running fade is stopped by an application controller, the fade timer shall be stopped immediately. After the fade timer has been stopped, "*targetLevel*" shall be set to "*actualLevel*" and the command shall be executed (if applicable).

If a running fade is stopped whilst it was pending at "*minLevel*" during startup, the control gear shall finish the startup process.

NOTE 2 This implies that in such a case both "*targetLevel*" and "*actualLevel*" are equal to "*minLevel*".

9.6 Min and max level

Changing the min or max level shall stop any running fade, before the storage of the new min or max level.

"SET MIN LEVEL (*DTR0*)" shall set "*minLevel*" depending on the "*DTR0*" value:

- if $0 \leq "DTR0" \leq \text{PHM}$: PHM;
- if "*DTR0*" $\geq$ "*maxLevel*" or MASK: "*maxLevel*";
- in all other cases: "*DTR0*".

If "*actualLevel*" > 0 and "*actualLevel*" $<$ "*minLevel*" as a result of setting a new min level, "*targetLevel*" shall be re-calculated on the basis of the new "*minLevel*". "*actualLevel*" shall be changed to "*targetLevel*" immediately and the light output shall be adjusted as quickly as possible. As a consequence, "*limitError*" shall be set to TRUE.

"SET MAX LEVEL (*DTR0*)" shall set "*maxLevel*" depending on the "*DTR0*" value, as follows:

- if "*minLevel*" $\geq$ "*DTR0*": "*minLevel*";
- if "*DTR0*" = MASK: 0xFE;
- in all other cases: "*DTR0*".

If "*actualLevel*" $>$ "*maxLevel*" as a result of setting a new max level, "*targetLevel*" shall be re-calculated on the basis of the new "*maxLevel*". "*actualLevel*" shall be changed to "*targetLevel*" immediately and the light output shall be adjusted as quickly as possible. As a consequence, "*limitError*" shall be set to TRUE.

NOTE "*minLevel*" and "*maxLevel*" can be used to compensate for differences in control gear properties. For example, if control gear have different values for PHM, they can be made to behave in a similar way by adjusting "*minLevel*".

9.7 Commands

9.7.1 General

A control gear shall check the device addressing scheme to see if it is addressed by a command. The control gear shall ~~execute~~ 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 group addressing and the given group does not match any of the groups identified by "*gearGroups*".
- The command is sent using reserved addressing.
- The command is sent using broadcast unaddressed addressing and "*shortAddress*" is not MASK.
- The command is not defined (e.g. reserved command).

The following command groups can be identified:

- level instructions;
 - level instructions without fade;
 - level instructions initiating a fade;
- configuration instructions;
- queries;

- special commands;
 - instructions;
 - queries;
- application extended commands.

9.7.2 Level instructions without fade

Level instructions without fade are instructions where the *"targetLevel"* shall be calculated; the transition from *"actualLevel"* to *"targetLevel"* shall take place immediately and the light output shall be adjusted as quickly as possible.

These commands can be divided into three categories:

- absolute level commands
 - "OFF", "RECALL MIN LEVEL", "RECALL MAX LEVEL";
- relative level commands
 - "STEP UP", "STEP DOWN", "ON AND STEP UP", "STEP DOWN AND OFF";
- configuration commands
 - "RESET", "SET MIN LEVEL (*DTR0*)", "SET MAX LEVEL (*DTR0*)".

9.7.3 Level instructions initiating a fade

Level instructions initiating a fade are instructions where the *"targetLevel"* shall be calculated; *"actualLevel"* shall fade to the *"targetLevel"* using the applicable fade time or *fade rate*. If the fade time equals 0 s, the transition from *"actualLevel"* to *"targetLevel"* shall take place immediately and the light output shall be adjusted as quickly as possible.

These commands can be divided into two categories:

- absolute level instructions using the fade time
 - "DAPC (*level*)", "GO TO SCENE (*sceneNumber*)", "GO TO LAST ACTIVE LEVEL";
- relative level instructions using the fade rate
 - "UP", "DOWN";
 - "CONTINUOUS UP", "CONTINUOUS DOWN".

9.7.4 Configuration instructions

Configuration instructions can be used to modify several control gear properties.

9.7.5 Queries

Queries can be used to request the value of several control gear properties.

9.7.6 Special commands

The special commands are a group of commands that are not addressable. All control gear shall interpret the special commands.

9.7.7 Application extended commands

Commands with their opcode in the range 0xE0 to 0xFF are reserved for special device types or features. Each device type or feature redefines these commands, except for the command with opcode 0xFF ("QUERY EXTENDED VERSION NUMBER"). See 9.18 for further information.

9.8 Command iterations

9.8.1 General

The requirements of IEC 62386-101:2014 and IEC 62386-101:2014/AMD1:2018, 9.4 apply with the following additions.

9.8.2 Command iteration of "UP" and "DOWN" commands

"UP" and "DOWN" instructions can be sent as a command iteration. Upon execution of the first instruction of such an iteration, unless this is precluded by the values of "*minLevel*" or "*maxLevel*", one step (final "*targetLevel*" = calculated "*targetLevel*" $\pm$ 1) shall be made.

NOTE 1 This ensures that there is an effect at the start of an iteration.

After that first step, the 200 ms fade shall start using the applicable fade rate. Subsequent steps shall be executed at intervals determined by the applicable fade rate, as long as the iteration continues. Every "UP" or "DOWN" instruction executed as a part of the iteration shall cause the 200 ms fade time to be restarted and "*targetLevel*" to be recalculated accordingly. The transition of "*actualLevel*" shall occur according to 9.5.1 and Figure 4, with the first such transition, excluding the initial step, occurring at a time of $1/(2 \cdot \text{"fadeRate"})$ after execution of the first "UP" and "DOWN" command.

NOTE 2 If the fade rate changes during a command iteration, the new fade rate is not used during the execution of this command iteration.

Figure 5 summarizes the required behaviour. The iterations start at Cmd 1, and end at Time out.

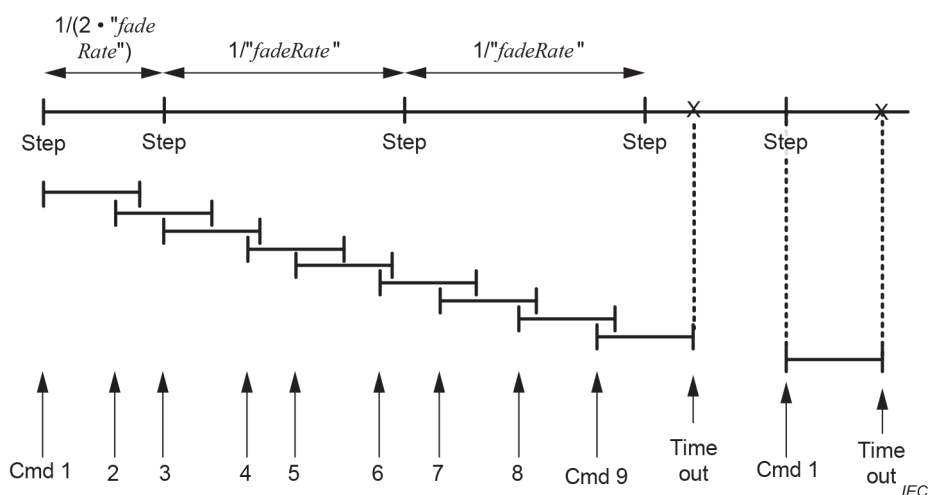


Figure 5 – Timing and response when executing command iteration

9.8.3 DAPC SEQUENCE (deprecated)

"ENABLE DAPC SEQUENCE" starts a direct arc power control (DAPC) command iteration that allows dynamic control of the light output.

Upon execution of "ENABLE DAPC SEQUENCE" the control gear shall temporarily use a fade time of $200 \text{ ms} \pm 20 \text{ ms}$ while the command iteration is active independent of the actual fade time or extended fade time. After the last fade of the sequence has finished, the original values shall be used.

NOTE As the fade time and fade rate variables do not change, the fade time and fade rate can be set and/or queried as normal.

The DAPC sequence shall end if 200 ms elapse without the control gear accepting a "DAPC (*level*)" command. The DAPC sequence shall be aborted on execution of an indirect arc power control command. "ENABLE DAPC SEQUENCE" accepted during DAPC command iteration, shall be discarded.

While the DAPC sequence is active, each execution of a "DAPC (*level*)" command shall start a 200 ms fade.

~~Since the DAPC sequence uses a fade time of 200 ms that might not be realised by all control gear and light source combinations, such gear may simply adjust the light output as quickly as possible. However, it should respond as if the fade has finished within the requested time.~~

If the control gear and light source combination cannot realise a fade time of 200 ms as required by the DAPC sequence, such control gear shall adjust the light output as quickly as possible. However, the control gear shall respond as if the fade has finished within the requested time. **6**

9.9 Modes of operation

9.9.1 General

Different operating modes can be selected 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 gear is in an IEC 62386 standard operating mode or in a manufacturer-specific mode.

9.9.2 Operating mode 0x00: standard mode

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

9.9.3 Operating mode 0x01 to 0x7F: reserved

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

9.9.4 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 gear is in a manufacturer-specific operating mode, the behaviour of the control gear may be manufacturer specific as well, with the following exceptions:

- ~~• as far as the control gear accesses the bus, it shall adhere to IEC 62386-101:2014 and IEC 62386-101:2014/AMD1:2018;~~
- with regard to bus access, the control gear shall adhere to IEC 62386-101:2022;
- the control gear shall adhere to this document at least as far as the following commands are concerned:
 - "SET OPERATING MODE (*DTR0*)", "QUERY OPERATING MODE", and "QUERY MANUFACTURER SPECIFIC MODE".
 - all special commands (see 11.7) except
WRITE MEMORY LOCATION (*DTR1*, *DTR0*, *data*),
WRITE MEMORY LOCATION – NO REPLY (*DTR1*, *DTR0*, *data*) and PING.

For the above commands ~~the various addressing methods shall apply, see 7.2.2~~ the requirements of 7.2 shall apply.

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

9.10 Memory banks

9.10.1 General

Memory banks are freely accessible memory spaces ~~defined for e.g. identification~~ containing information or configuration settings of the control gear ~~in a system~~. 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 gear or updated by the control gear itself. For all other parts, write access using memory access commands can be enabled by the manufacturer. ~~Write access to a memory bank location can be locked. memory banks can be implemented using RAM, ROM or NVM.~~ 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 8.

Table 8 – Memory types 7

Memory type	Accessibility via the bus ^a	Volatility ^b	May be changed autonomously by the control gear during run time	Description
ROM	RO	NV	No	For all fixed values that will not change during run time of control gear. 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 ~~almost 64 kBytes, organized in maximum~~ 256 memory banks of maximum 255 bytes each (approximately 64 kBytes). As this document ~~prescribes~~ 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].

9.10.2 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 9.

Table 9 – 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 access 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 gear 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 gear shall not change the value of the lock byte other than as a consequence of power cycle or a "RESET MEMORY BANK (DTR0)" 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 gear 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".

9.10.3 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".

9.10.4 Protectable memory locations 8

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.

9.10.5 Memory bank reading

9.10.5.1 General

A selected memory bank location can be read by means of 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 memory bank exists, and selected memory bank location is~~

- ~~• not implemented, or~~
- ~~• above the last accessible memory location,~~

~~the answer shall be NO.~~ **9**

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.

~~To ensure consistent data when reading a multi-byte value from a memory bank, it is recommended that a mechanism be implemented that latches all bytes of the multi-byte value when the first byte of the multi-byte value is read and that unlatches the bytes at any other command than "READ MEMORY LOCATION (*DTR1*, *DTR0*)".~~ **10**

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.

9.10.5.2 Reading multi-byte values **11**

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 9.10.6.3).

9.10.5.3 Unimplemented locations **12**

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.

9.10.5.4 Temporarily unavailable locations 13

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 gear 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 gear.

9.10.5.5 Latching a complete memory bank for reading 14

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 were 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.

9.10.6 Memory bank writing

9.10.6.1 General

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

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

- "WRITE MEMORY LOCATION (*DTR1*, *DTR0*, *data*)": The control gear 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 gear to reply.

A control gear 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*)";
- "DTR0 (*data*)", "DTR1 (*data*)", "DTR2 (*data*)";
- "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.

9.10.6.2 Write failure

If the selected memory bank location is

- not implemented, or
- above the last accessible memory location, or
- locked (see 9.10.2), or
- not writeable, or
- is a protectable location that is currently write protected (see 9.10.4), or
- there is an attempt to write a value outside of the permitted range,

the answer to "WRITE MEMORY LOCATION (*DTR1*, *DTR0*, *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 location 0xFF, "*DTR0*" shall be incremented by one. Otherwise, "*DTR0*" shall not change. This mechanism allows for easy consecutive writing to memory bank locations.

9.10.6.3 Writing multi-byte values 15

To ensure consistent data when writing a multi-byte value into a memory bank, ~~it is recommended that a mechanism be implemented that only accepts the new multi-byte value for writing after all bytes of the multi-byte value have been accepted~~ 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. 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.

9.10.7 Memory bank 0

Memory bank 0 contains information about the control gear. Memory bank 0 shall be implemented in all control gear.

Memory bank 0 shall be implemented using the memory map shown in Table 10, with at least the memory locations up to address 0x7F implemented, excluding reserved locations.

Table 10 – 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 [0,64]	ROM
0x19	Number of logical control gear units in the bus unit	factory burn-in, range [1,64]	ROM
0x1A	Index number of this logical control gear unit	factory burn-in, range [0,(location 0x19) – 1]	ROM
0x1B ^f	Current bus unit configuration ^f 16	factory burn-in ^g	ROM
[0x1B,0x1C,0x7F]	Reserved – not implemented	answer NO	n.a.
[0x80,0xFE]	Additional control gear information ^e	^e	ROM
0xFF	Reserved – not implemented	answer NO	n.a.

Address	Description	Default value (factory)	Memory type
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 62386-101:2014 and IEC 62386-101:2014/AMD1:20182022, 4.2. ^c Format of the version number is defined in 4.2. ^d Format of the version number is defined in IEC 62386-103:2014 and IEC 62386-103:2014/AMD1:20182022, 4.2. If not implemented, this is indicated by 0xFF. ^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 gear logical units shall have the same values in location 0x1B.

A bus unit ~~might~~ can contain both control gear and control devices. They share various numbers (e.g. GTIN, ~~unique identification number...~~) in binary (see [1]¹). 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 ~~102 or the 103~~ commands specified in this document or in IEC 62386-103:2022 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), e.g. the EAN, in binary. 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, ~~preferably~~. 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.

¹ Numbers in square brackets refer to the Bibliography.

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.~~

The byte in location 0x17 shall contain the implemented IEC 62386-103 version number of the bus unit. If no control device is implemented, the version number shall be 0xFF.

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 0 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 1 to 64.

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

~~NOTE As an example there might be a product containing three logical devices with three different short addresses. Each of these control gear has the same GTIN and identification number, each reports as number of devices the value 3 and the index of the three control gear is reported as 0, 1 or 2 respectively. Reading location 0x1A using broadcast yields a backward frame according to IEC 62386-101:2014 and IEC62386-101:2014/AMD1:2018, 9.5.2 (overlapping backward frame).~~

EXAMPLE A product can contain three control gear 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 gear units with the value 3 and the index of the three control gear logical units is reported as 0, 1 or 2 respectively. Reading location 0x1A using broadcast yields a backward frame according to IEC 62386-101:2022, 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 device types and logical units. See 9.20.

9.10.8 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 gear. ~~The control gear manufacturer may implement~~ 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 11.

Table 11 – 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)
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 gear information^a~~	a	a	a
[0x11, 0xFE]	Use of these locations is specified in other parts of the IEC 62386-2xx series. 17	d	d	d
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-access type of these bytes shall be defined by the manufacturer. ^b Reset value after "RESET MEMORY BANK". ^c Also used as power on value. ^d See other parts of the IEC 62386-2xx series.				

The bytes in locations 0x03 to 0x08 ("OEM GTIN 0" to "OEM GTIN 5") should be used to identify the product containing the control gear. 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.

9.10.9 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 9.

9.10.10 Reserved memory banks

Memory banks 200 to 255 are reserved for future use ~~and shall not be implemented~~, or are described in the IEC 62386-2xx series. Implementation or use other than described in the IEC 62386-2xx series is not permitted.

9.11 Reset

9.11.1 Reset operation

A control gear shall implement a reset operation to set all variables to their reset values (see Table 16).

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 gear ~~may or may not~~ can respond to ~~any~~ 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~~ gear have finished the reset operation.

9.11.2 Reset memory bank operation

A control gear shall implement a reset operation to set the content of all unlocked memory banks to their reset values (see 9.10), 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 gear ~~may or may not~~ can respond to ~~any~~ commands ~~or not~~. However, until this reset ~~memory bank~~ operation is complete, none of the affected memory bank 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 ~~allow~~ ensure all ~~control~~ gear ~~have~~ enough time to finish the reset memory bank operation.

9.12 System failure

If the control gear detects a system failure (see IEC 62386-101:2014~~—and IEC62386-101:2014/AMD1:2018~~2022, 4.11) and "*systemFailureLevel*" is not MASK, "*targetLevel*" shall be calculated on the basis of "*systemFailureLevel*". The transition from "*actualLevel*" to "*targetLevel*" shall take place immediately and the light output shall be adjusted as quickly as possible.

If "*systemFailureLevel*" is MASK, the control gear shall not react to a system failure.

On restoration of the bus idle voltage the control gear shall not react.

"*systemFailureLevel*" can be set and queried with "SET SYSTEM FAILURE LEVEL (*DTR0*)" and "QUERY SYSTEM FAILURE LEVEL" respectively.

When the bus power is restored after a system failure, bus-powered control gear shall follow the power-on procedure defined in 9.13 below. Consequently, the variable "*systemFailureLevel*" is not used. Nevertheless, all control gear, including bus-powered control gear, shall maintain "*systemFailureLevel*" and conform to the requirements of the specifications of all the commands relating to it.

NOTE Implementing "*systemFailureLevel*", although this variable is normally not applicable for bus-powered devices, is done to avoid separate test conditions of control gear.

9.13 Power on

After an external power cycle (see IEC 62386-101~~—and IEC62386-101:2014/AMD1:2018~~2022, 4.11.1), the device shall retain its most recent configuration, with the following exceptions:

- the memory bank write enable state shall be disabled for all memory banks and the lock byte shall be set to 0xFF;
- all running timers shall be stopped and cancelled~~+~~ (reset);
- "*powerCycleSeen*" shall be set to TRUE;
- "*actualLevel*" shall be set to 0x00 keeping the lamp off;
- "*lampOn*" shall be set to FALSE;
- "*limitError*" shall be set to FALSE;
- "*targetLevel*" shall be set to 0x00;
- the control gear may start preheating the lamp but the lamp shall not ignite. While preheating, "*actualLevel*" shall be kept at 0x00 contrary to normal startup activity;
- all variables mentioned in Table 16 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 62386-2xx series shall be included.

Bus-powered devices shall activate the power on level immediately. For externally-powered devices the following holds:

If a level control command other than "GO TO SCENE (*sceneNumber*)" where the value of the scene equals MASK and other than DAPC(MASK) is accepted, it shall be executed.

If "GO TO SCENE (*sceneNumber*)" where the value of the scene equals MASK is accepted, the control gear shall discard the command and continue as if no level control command has been accepted.

If DAPC(MASK) is executed, the control gear shall stop any startup activity.

NOTE 1 Since *"actualLevel"* = 0, this effectively keeps the lamp off.

The control gear shall activate the power on level according to Table 12 by calculating the *"targetLevel"* on the basis of *"powerOnLevel"*. If *"powerOnLevel"* equals MASK, *"targetLevel"* shall be set to *"lastLightLevel"*. *"actualLevel"* shall be set to *"targetLevel"* immediately and the light output shall be adjusted as quickly as possible. Power on level activation shall not affect *"limitError"*, effectively leaving it as FALSE. 18

If a level control command is accepted before the power on level is activated, this command shall be executed immediately and the control gear shall not activate the power on level.

Table 12 – Power on timing 19

Power on system response	Minimum time	Maximum time
Lamp off		540 ms
Grey area	> 540 ms	< 660 ms
power on level	660 ms	
Lamp off ends and power on level is activated	540 ms	660 ms

NOTE 2 Thus, there is an interval during which a control device can send a level control command which will be obeyed immediately, so DAPC(0x00) or DAPC(MASK) can be used to prevent from going automatically to *"powerOnLevel"*.

It is possible that a system failure is detected before the power on level has been reached. If *"systemFailureLevel"* is not MASK, the *"targetLevel"* is recalculated on the basis of *"systemFailureLevel"* and the power on level shall not be activated.

"powerOnLevel" can be set and queried with "SET POWER ON LEVEL (*DTR0*)" and "QUERY POWER ON LEVEL" respectively.

After receiving the first 16-bit forward frame on the interface after power on, the control gear shall only respond to frames described in IEC 62386-101:2014 and IEC62386-101:2014/AMD1:20182022.

9.14 Assigning short addresses

9.14.1 General

"shortAddress" shall be derived from *data* or *"DTR0"* depending on the command used. It shall be set on ~~receipt~~ 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"* = 1xxxxxxb or xxxxxxx0b: no change;
- in all other cases (0AAAAAA1b): 00AAAAAAb.

9.14.2 Random address allocation

A control gear shall implement an initialisation state, only in which, in addition to the other operations identified in this document, a set of commands are enabled that allow an application controller to detect and uniquely identify control gear 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 gear to leave the initialisation state immediately.

The control gear 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.

9.14.3 Identification of a device

9.14.3.1 General

During identification no variables shall be affected unless explicitly stated otherwise. Where appropriate, variables can be temporarily ignored, so that after the identification has ended, there are no side effects.

When identification is active, the light output ~~may~~ can be at any level between off and 100 %, "*minLevel*" and "*maxLevel*" as well as "*actualLevel*" being in effect temporarily ignored.

Identification shall be stopped upon execution of any instruction other than INITIALISE (*device*), RECALL MIN LEVEL, RECALL MAX LEVEL or IDENTIFY DEVICE.

NOTE Query commands are not instructions. 20

After identification has stopped, the light output shall be adjusted as quickly as possible to reflect "*actualLevel*" and the command shall be executed (if applicable).

9.14.3.2 Method one: single instruction

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

NOTE The actual procedure is manufacturer specific.

While identification is active, the control gear shall, without interrupting the identification procedure:

- on RECALL MIN LEVEL: set "*actualLevel*" and "*targetLevel*" to "*minLevel*";
- on RECALL MAX LEVEL: set "*actualLevel*" and "*targetLevel*" to "*maxLevel*".

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

For examples of how to use the commands, see Annex A.

9.14.3.3 Method two: using "RECALL MAX LEVEL" and/or "RECALL MIN LEVEL" (deprecated)

While "*initialisationState*" is not DISABLED, the control gear shall:

- on RECALL MIN LEVEL: set "*actualLevel*" and "*targetLevel*" to "*minLevel*", and then adjust the light output as quickly as possible to its PHM level. If, however, the PHM is not visibly significantly different from 100 %, then the lamp shall be temporarily switched off instead;
- on RECALL MAX LEVEL: set "*actualLevel*" and "*targetLevel*" to "*maxLevel*", and then adjust the light output as quickly as possible to 100 %.

Alternatively, the control gear shall execute "IDENTIFY DEVICE", starting or re-triggering the identification procedure.

NOTE It is acceptable for the process of identifying individual control gear to depend upon both commands being executed in an alternating sequence.

Identification shall be stopped immediately when one of the following conditions hold:

- the "*initialisationState*" changes to DISABLED;
- upon execution of any instruction other than INITIALISE (*device*), RECALL MIN LEVEL, RECALL MAX LEVEL or IDENTIFY DEVICE.

For examples of how to use the commands, see Annex A.

9.14.4 Direct address allocation

"SET SHORT ADDRESS (*DTR0*)" can be used to directly program a short address to the addressed gear.

9.15 Failure state behaviour

If the control gear is in a failure state, in which operation of the lamp(s) is not possible as intended (lamp failure and/or control gear failure) it shall react to level commands in the following way.

The control gear shall calculate "*targetLevel*" in accordance with the commands executed, and control the lamp insofar as that is practicable. As a consequence of the fault, the normal relationship between "*actualLevel*" and light output could temporarily change.

NOTE EXAMPLE A control gear ~~might~~ can, on detecting an excessively high temperature, protect itself from the risk of thermal damage by limiting the light output.

If the failure state is resolved, the control gear shall re-establish the normal relationship between "*actualLevel*" and light output.

9.16 Status information

9.16.1 General

Each control gear shall expose its status as a combination of device properties as given in Table 13.

Table 13 – Control gear status

Bit	Description	Value	See
0	"controlGearFailure" is TRUE?	"1" = "YES"	9.16.2
1	"lampFailure" is TRUE?	"1" = "YES"	9.16.3
2	"lampOn" is TRUE?	"1" = "YES"	9.16.4
3	"limitError" is TRUE?	"1" = "YES"	9.16.5
4	"fadeRunning" is TRUE?	"1" = "YES"	9.16.6
5	"resetState" is TRUE?	"1" = "YES"	9.16.7
6	"shortAddress" is MASK?	"1" = "YES"	9.16.8
7	"powerCycleSeen" is TRUE?	"1" = "YES"	9.16.9

The device status can be queried using "QUERY STATUS". The bits shall reflect the actual situation without delay unless explicitly stated otherwise.

9.16.2 Bit 0: Control gear failure

A control gear failure according to this document is a situation in which the control gear cannot operate as intended.

NOTE—EXAMPLE Mains under voltage, over temperature, unexpected watchdog timers firing ~~etc.~~

If a control gear failure is detected, "controlGearFailure" shall be set to TRUE.

If the failure is no longer detected, and normal operation has been resumed, "controlGearFailure" shall be set to FALSE.

A control gear failure shall be detected and indicated latest after 30 s.

9.16.3 Bit 1: lamp failure

A lamp failure according to this document is a situation in which the lamp cannot be operated as intended due to, for example, incorrect lamp connection or lamp defects. The minimum detection method for lamp failure is the lamp disconnection, unless explicitly stated otherwise depending on the light source type (see 11.5.19).

If a lamp failure is detected, "lampFailure" shall be set to TRUE, otherwise "lampFailure" shall not be changed from FALSE to TRUE **21**. Lamp failure shall be detected and indicated latest after 30 s when the control gear is not in standby (see 9.2). In case the startup phase takes longer than 30 s, for example for HID lamps, "lampFailure" shall be set at the end of the startup phase to the correct value.

A total lamp failure is a lamp failure with no light output. A partial lamp failure is a lamp failure with still some light output.

NOTE If the connected lamp is unsuitable for the control gear, there is a possibility that lamp failure is detected.

If "lampFailure" is TRUE, the control gear shall periodically check to determine whether the lamp situation has improved. This check shall be executed at least whenever "targetLevel" changes from 0x00 to a greater value. After a successful startup, "lampFailure" shall be set to FALSE.