IEC 62386-102

Edition 3.0 2022-11
COMMENTED VERSION

INTERNATIONAL STANDARD

colour inside

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

IEC 62386-102:2022-11 CMV(en)

IEC Secretariat
3, rue de Varembé
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 It also gives information on projects, replaced and withdrawn publications.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

colour inside

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

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.

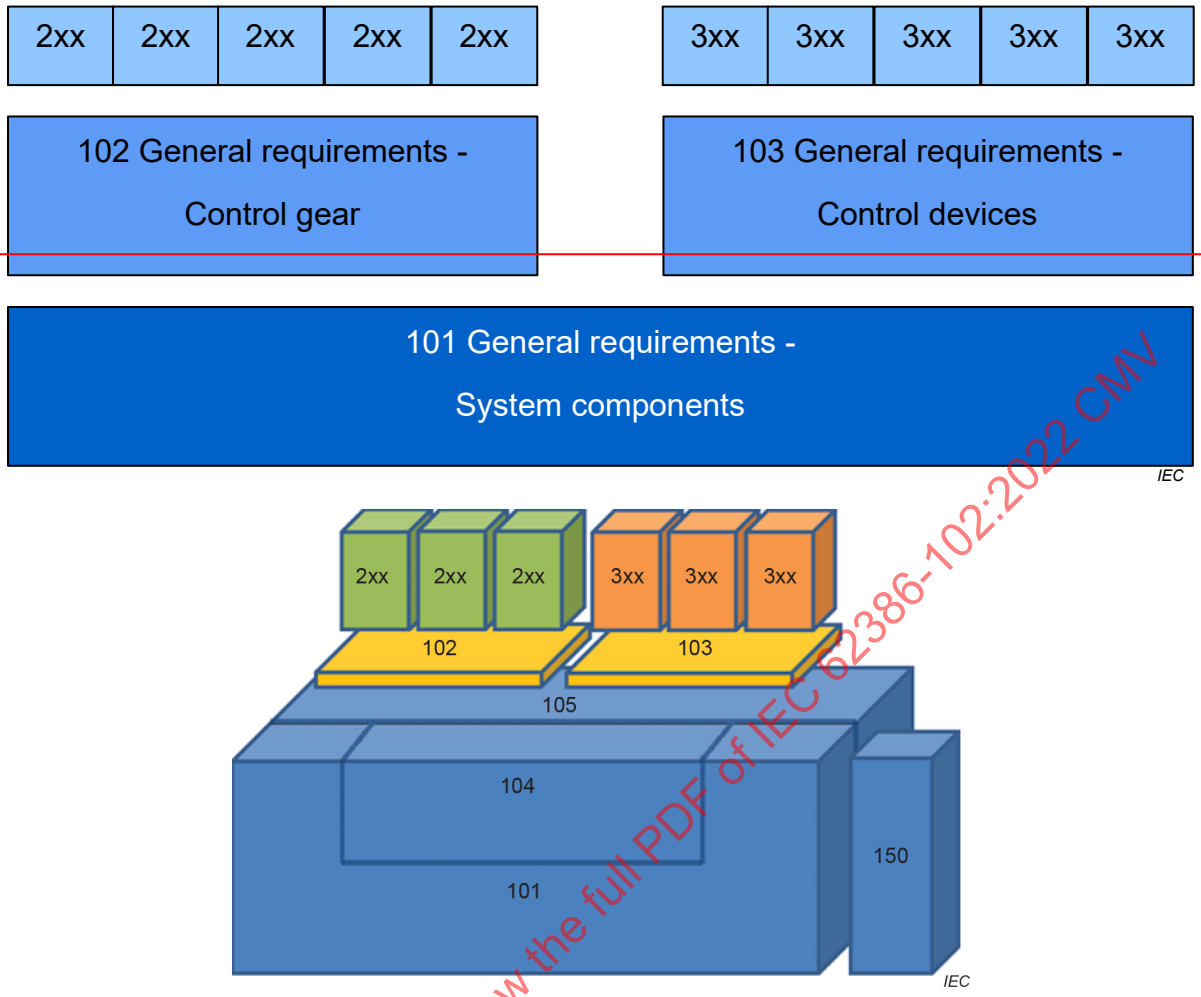


Figure 1 – IEC 62386 graphical overview 1

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:~~2014~~2022, *Digital addressable lighting interface – Part 101: General requirements – System components*
~~IEC 62386-101:2014/AMD1:2018~~

IEC 62386-103:~~2014~~2022, *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 IEC62386-101:2014/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]	7...0	
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/AMD1:2018~~2022, Clause 8 apply.

9 Method of operation

9.1 General

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

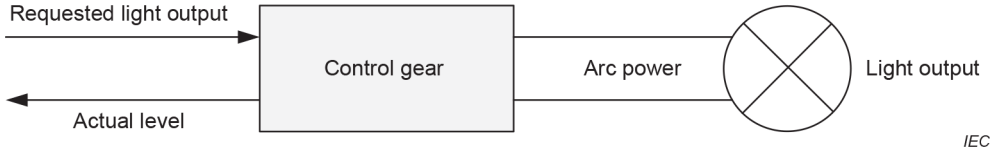


Figure 2 – Control gear directly operating a light source

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

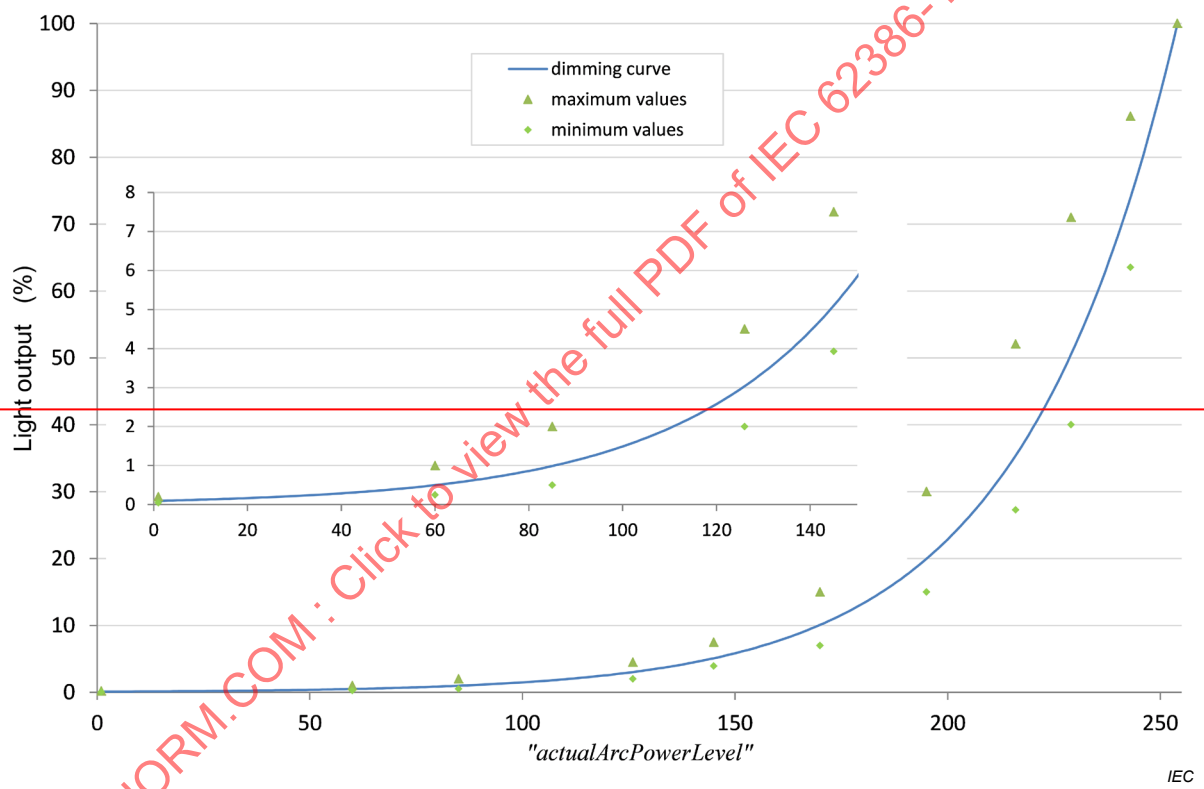
The dimming curve determines how the level shall be translated into light output.

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

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

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 ±½ 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.



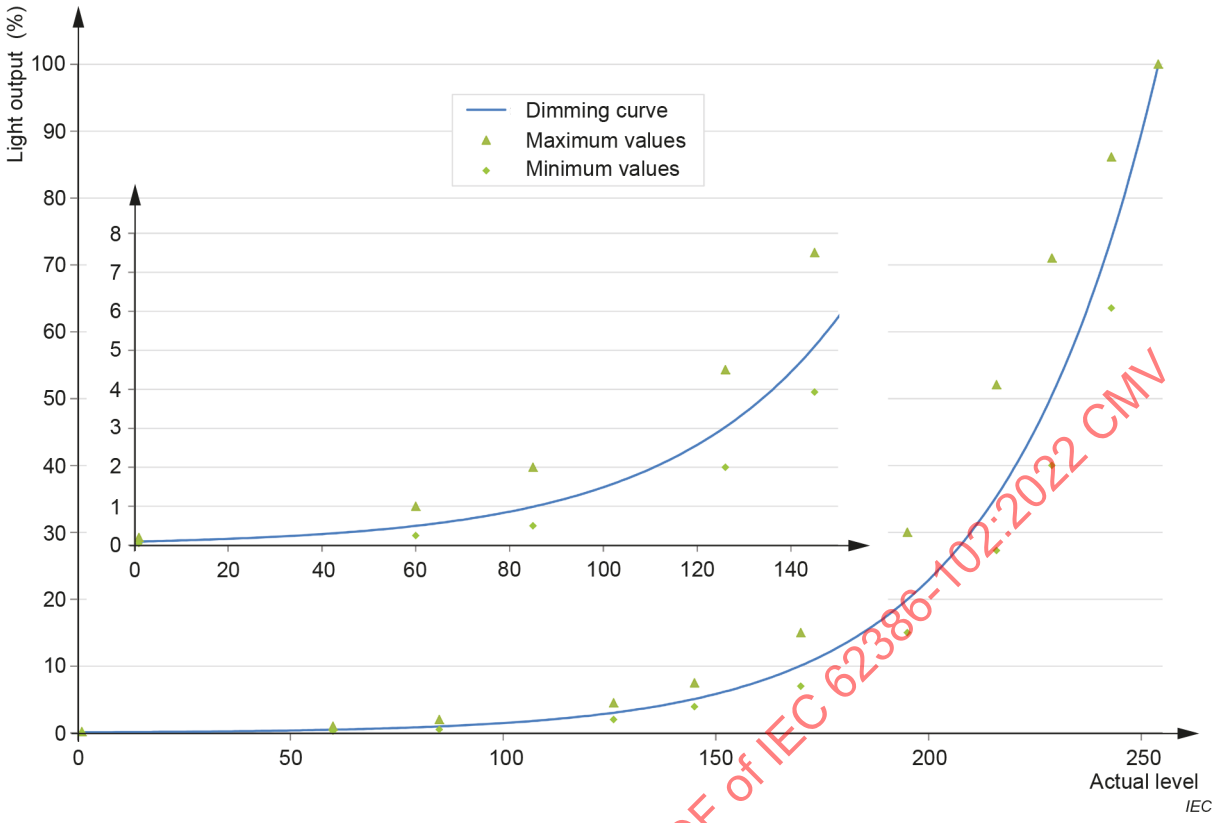


Figure 3 – Dimming curve

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.

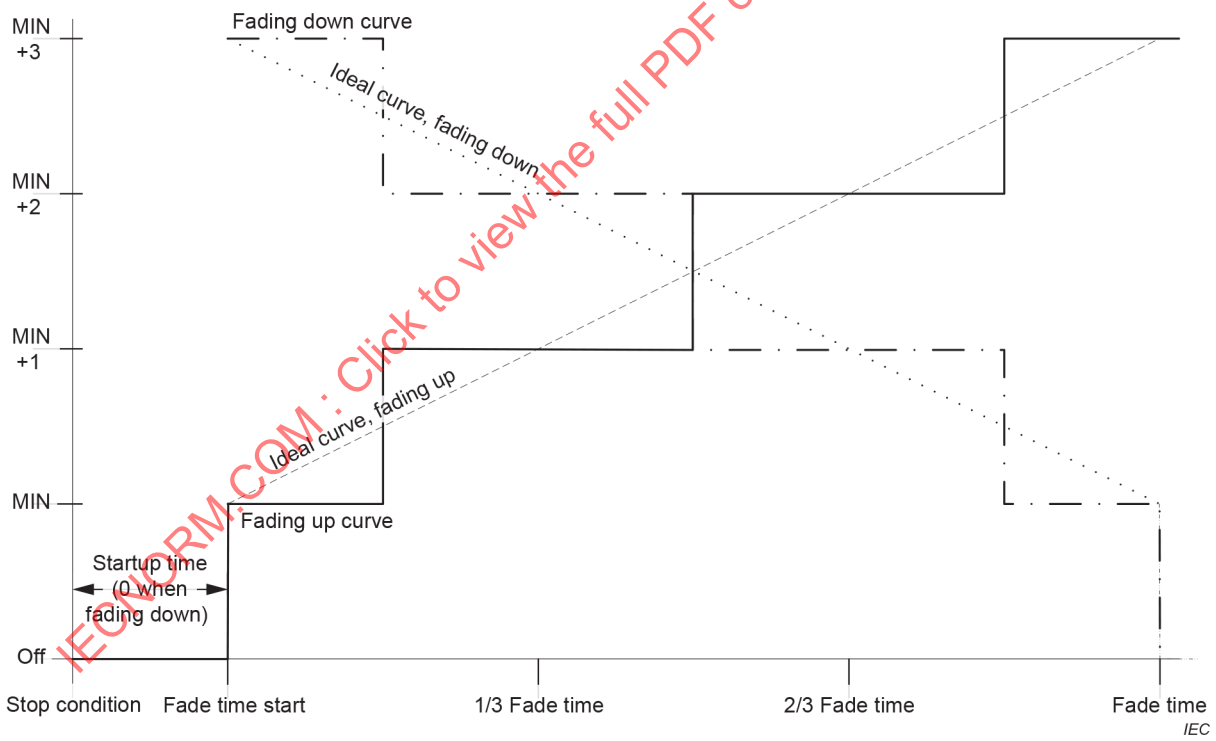


Figure 4 – Level over time, fading up and down

Testing shall be done with "*minLevel*" ≥ 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**.

~~*"fadeRate"*~~	~~**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]:~~ $\frac{1}{2}\cdot\sqrt{2^{"fadeTime"}}\cdot 1\ \text{s}$

Table 4 lists the possible fade time values.

Table 4 – Fade times

"fadeTime"	*Minimum fade time* s	*Nominal fade time* s	*Maximum fade time* 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:

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

Fade rate $= \frac{506}{2^{\left(\frac{"fadeRate"}{2}\right)}}$ steps/s **4**.

Table 5 lists the possible fade rate values.

Table 5 – Fade rates

"fadeRate"	*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

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.

Fade time = $extendedFadeTimeBase \cdot extendedFadeTimeMultiplier$

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 0YYYAAAAb, where YYYb equals the fade time multiplier, and AAAAb the fade time base: The resulting fade time shall be monotically 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 YYYb.

The extended fade time can be queried using "QUERY EXTENDED FADE TIME". The answer shall be 0 YYY AAAAb, where YYYb 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 ≤ *"DTR0"* ≤ PHM: PHM;
- if *"DTR0"* ≥ *"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"* ≥ *"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~~2022, 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*" ± 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 · *"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.

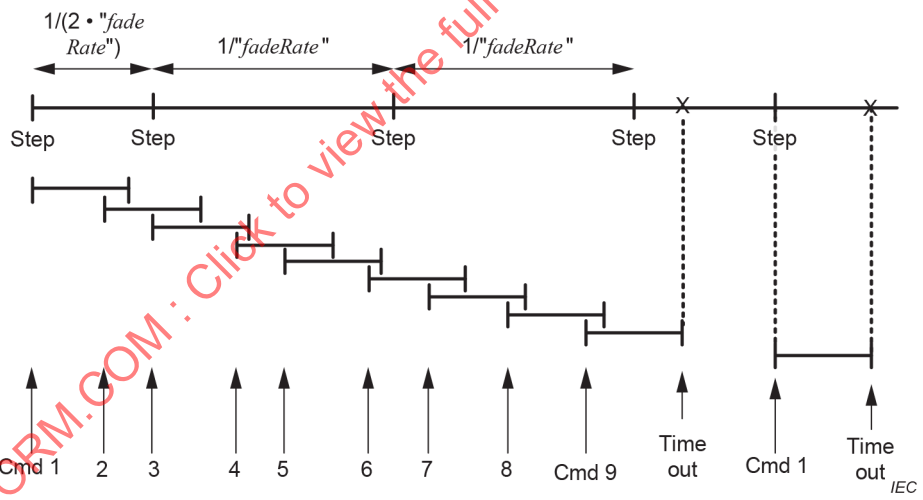


Figure 5 – Timing and response when executing command iteration

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 ms ± 20 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 IEC62386-101:2014/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/AMD1:2018~~2022, 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/AMD1:2018~~2022, 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/AMD1:2018~~2022.

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"* = 1xxxxxxxb 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 ± 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.

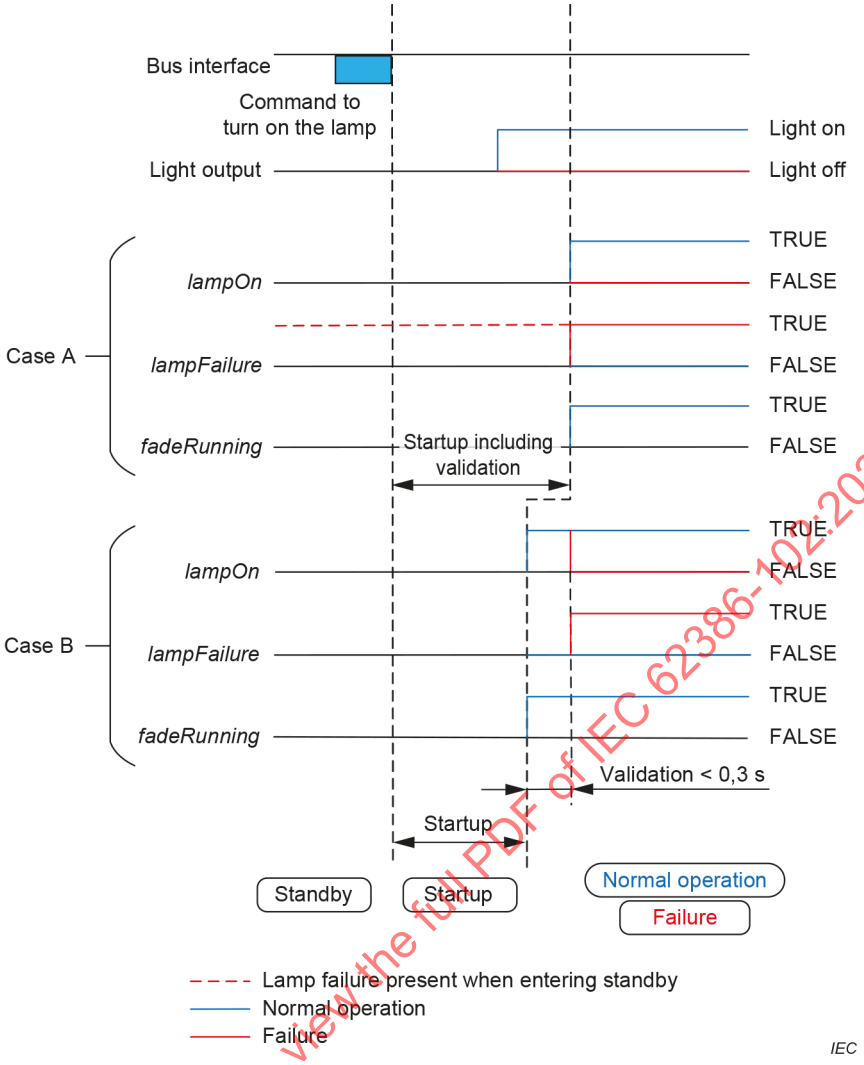


Figure 6 – Correlation between "*lampFailure*", "*lampOn*" and "*fadeRunning*" bits

The startup phase shall include the validation that the lamp is stable and emitting light before continuing with normal operation or failure. This behaviour is illustrated in Figure 6, Case A and applies to the following situations:

- "*lampFailure*" is TRUE before entering standby, or
- the lamp validation takes longer than 0,3 s.

The startup phase may exclude the validation of the lamp in the following situation, illustrated in Figure 6, Case B:

- "*lampFailure*" is FALSE before entering standby and,
- the lamp validation takes less than 0,3 s.

In this situation the validation should be part of the normal operation and shall be executed immediately after startup. This implies that "*lampOn*" ~~might~~ can be incorrect for a maximum of 0,3 s until the validation is finished.

NOTE Figure 6 also illustrates the effect of "*lampFailure*" on "*lampOn*" and "*fadeRunning*" bits for the scenarios described above.

For the light source type "unknown light source type", there shall be support for this bit. If there is no support for the lamp failure bit, this shall be made explicit in the corresponding part of the IEC 62386-2xx series.

For the light source type "no light source"~~, there may be~~ support for this bit is optional. Testing is excluded for this light source type. If there is support for the lamp failure bit, this shall be made explicit in the corresponding part of the IEC 62386-2xx series.

9.16.4 Bit 2: lamp on

"lampOn" shall be set to FALSE when the lamp is off, during startup, and in case of total lamp failure. In all other cases it shall be set to TRUE.

9.16.5 Bit 3: limit error

Except for activation of "*powerOnLevel*" (see 9.13) **22**, if the last requested target level has been modified in accordance with *"minLevel"* or *"maxLevel"* limitations, or *"targetLevel"* has been modified due to a change of *"minLevel"* or *"maxLevel"*, "*limitError*" shall be set to TRUE.

If the last target level requested by "DAPC (*level*)" equals "MASK", "*limitError*" shall not change.

In all other cases "*limitError*" shall be set to FALSE.

9.16.6 Bit 4: fade running

"fadeRunning" shall be set to FALSE except for the time during which the fade timer is running. *"fadeRunning"* shall be set to TRUE from the beginning of the fade (after startup) until the end of the fade time, regardless of whether *"targetLevel"* and *"actualLevel"* reach the same level.

9.16.7 Bit 5: reset state

"resetState" shall be set to TRUE if all the NVM variables mentioned in Table 16, except "*lastLightLevel*", are at their reset value. The NVM variables that are marked with 'no change' in the reset value column shall not be considered. NVM variables defined in implemented parts of the IEC 62386-2xx series shall be included.

In all other cases the bit shall be set to FALSE.

9.16.8 Bit 6: missing short address

This bit indicates whether a short address has been assigned to the gear, by checking *"shortAddress"*. The bit shall be TRUE if *"shortAddress"* = MASK.

In all other cases the bit shall be set to FALSE.

9.16.9 Bit 7: power cycle seen

"powerCycleSeen" shall be set to TRUE after an external power cycle (see IEC 62386-101:~~2014 and IEC62386-101:2014/AMD1:2018~~2022, 4.11) has occurred.

"powerCycleSeen" shall be set to FALSE once one of the following commands has been executed:

"RESET", "DAPC (*level*)", "OFF", "UP", "DOWN", "STEP UP", "STEP DOWN", "RECALL MAX LEVEL", "RECALL MIN LEVEL", "GO TO LAST ACTIVE LEVEL", "STEP DOWN AND OFF", "ON AND STEP UP", "CONTINUOUS UP", "CONTINUOUS DOWN", "GO TO SCENE (*sceneNumber*)".

9.17 Non-volatile memory **23**

~~Physical non-volatile memory typically supports a limited number of write cycles. Since many variables are NVM type, the physical limitations need some attention.~~

~~A control gear should store NVM variables in such a way that their content is never lost and the intended lifetime of the device can be reached. This means that it may not be possible to physically write every change in a variable immediately. There may be situations in which the control gear is not able to physically write the variables to NVM, especially if a particular NVM variable is changed very frequently.~~

~~Since the application controller cannot know the control gear's internal mechanism for physically saving persistent variables, the instruction "SAVE PERSISTENT VARIABLES" is defined to force the control gear to physically write all variables of type NVM to memory. This command is an addition to the normal writing of NVM variables. Its intended use is to ensure that important changes made by an application controller cannot be lost, e.g. after assigning all short addresses or setting other important (and stable) configuration data. Clearly it is not intended to be used after every level change. Typically, this command is used only a handful of times for an entire installation.~~

~~NOTE Typically the command can be used a few thousand times before causing physical damage to the control gear's NVM.~~

~~Physically saving the variables in response to the instruction shall take at most 300 ms to complete. While the saving operation is on-going, the light output may fluctuate and the control gear may or may not respond to any command. However, until the operation is complete, the value of the affected variables may be undefined. Moreover, if the light is off when the instruction is executed, the light shall stay off; in this case, no flicker shall be visible.~~

~~The light output may not fluctuate during saving operations unless these are triggered by this command.~~

~~An application controller can trigger the save operation using the "SAVE PERSISTENT VARIABLES" instruction and should wait at least 350 ms to ensure all gear have finished the operation.~~

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

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

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

9.18 Device types and features

Commands with their opcode in the range 0xE0 to 0xFF are reserved for special device types or features. Each device type~~/~~ or feature re-defines these commands, except for the command with opcode 0xFF ("QUERY EXTENDED VERSION NUMBER").

The device type~~/~~ or feature specific command set can be selected by the instruction "ENABLE DEVICE TYPE (*data*)".

This instruction shall select the device type~~/~~ or feature for which only the next application extended command (refer to 11.6) is valid. Executing this instruction shall cancel any previous selection of a device type.

The enabling of the device type~~/~~ or feature shall be cancelled upon execution of the next command, and that command shall be executed according to its specification, regardless of whether it is an application extended command or not.

A control gear shall not react to commands which belong to the application extended commands if *data* equals MASK, 254 or represents a device type~~/~~ or feature not supported by this control gear.

The device types~~/~~ and features shall be coded as specified in the ~~particular~~ relevant parts of the IEC 62386-2xx series.

An application controller can check which device types~~/~~ and features are supported by the control gear. "QUERY DEVICE TYPE" reports the supported device ~~type/~~types and features. If more than one device type~~/~~ or feature is supported, "QUERY DEVICE TYPE" reports MASK. In that case, the application controller can check all supported device types~~/~~ and features by repeating "QUERY NEXT DEVICE TYPE" until 254 is received as an answer. Issuing "QUERY DEVICE TYPE" automatically ensures that the first supported device type~~/~~ or feature will be reported by "QUERY NEXT DEVICE TYPE".

To check the version number of the supported device types~~/~~ and features, the application controller can send "ENABLE DEVICE TYPE (*data*) followed by "QUERY EXTENDED VERSION NUMBER". This will report the version number of that specific device type~~/~~ or feature implementation.

Application controllers should be able to identify individual control gear and store the relationship between the control gear's individual address, and its device types~~/~~ and features, in persistent memory.

9.19 Using scenes

A control gear shall support the use of 16 scenes. The following commands shall be supported:

"GO TO SCENE (*sceneNumber*)", "REMOVE FROM SCENE (*sceneX*)", "QUERY SCENE LEVEL (*sceneX*)" and "SET SCENE (*DTR0, sceneX*)".

These commands actually comprise 16 commands each, one for each scene. This is accomplished by selecting a block of 16 consecutive opcodes. The number of the scene to be used can thus easily be calculated.

Upon execution of one of the scene commands, *sceneNumber* shall be derived from the opcode: *sceneNumber* = opcode – opcodeBase. This identifies the scene to be used. The opcodeBase can be found in Table 14.

Table 14 – Scenes

Command	opcodeBase	Opcode range
GO TO SCENE (*sceneNumber)*	0x10	[0x10,0x1F]
REMOVE FROM SCENE (*sceneX*)	0x50	[0x50,0x5F]
QUERY SCENE LEVEL (*sceneX*)	0xB0	[0xB0,0xBF]
SET SCENE (*DTR0, sceneX*)	0x40	[0x40,0x4F]

The *"sceneX"* variable also stands for 16 individual variables, where *X* equals *sceneNumber* in the range of [0,15].

On accepting command "GO TO SCENE (*sceneNumber*)" the reaction of the control gear shall depend upon the current value of "*sceneX*", where *X* is derived from *sceneNumber*. If "*sceneX*" equals MASK, "*targetLevel*" shall not be affected. Otherwise, the control gear shall behave exactly as if "DAPC (*level*)" had been accepted with level equal to "*sceneX*".

NOTE Using "DAPC (*level*)" implies the transition is made using the set fade time.

9.20 Current bus unit configuration 24

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

- the number of logical units,
- the device types implemented for each logical unit, and
- particular types, defined for a particular device type.

For the given value of current bus unit configuration, the implementation shall be as shown in Table 15.

Table 15 – Current bus unit configuration

Current bus unit configuration	Number of logical units[a]	Minimum device types present[b]	Description
0	1	1 of [6]	1 logical unit of LED (device type 6)
1	2	2 of [6]	2 logical units of LED (device type 6)
2	3	3 of [6]	3 logical units of LED (device type 6)
3	4	4 of [6]	4 logical units of LED (device type 6)
4	1	1 of [8(Tc)]	1 logical unit of colour type Tc (device type 8)
5	2	2 of [8(Tc)]	2 logical units of colour type Tc (device type 8)
6	1	1 of [8(RGBWAF)]	1 logical unit of colour type RGBWAF (device type 8)
7	1	1 of [8(Tc+RGBWAF+xy)]	1 logical unit with colour types Tc, RGBWAF and xy (device type 8)
8	2	1 of [6], 1 of [8(Tc)]	1 logical unit of LED (device type 6) and 1 logical unit of colour type Tc (device type 8)
9	1	1 of [1(type A)]	1 logical unit of self-contained emergency type A (maintained dimming controllable, device type 1)
10	1	1 of [1(type B)]	1 logical unit of self-contained emergency type B (maintained on/off controllable, device type 1)
11	1	1 of [1(type C)]	1 logical unit of self-contained emergency type C (maintained non-controllable, device type 1)
12	1	1 of [1(type D)]	1 logical unit of self-contained emergency type D (non-maintained non-controllable, device type 1)
13-191	Reserved		
192-255	Manufacturer-specific		

[a] Total number of logical units in the bus unit, of the device type given in brackets.

[b] At least the listed device types shall be implemented in the given configuration.

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

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

10 Declaration of variables

The default values, the reset values, power on values, the range of validity and the type of memory of the defined variables shall be as given in Table 16.

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

Table 16 – Declaration of variables

Variable	Default value (factory)	Reset value	Power on value	Range of validity	Memory type
"actualLevel"	[a]	0xFE	0x00	0, [*"minLevel"*, *"maxLevel"*]	RAM
"targetLevel"	[a]	0xFE	See 9.13 Power on	0, [*"minLevel"*, *"maxLevel"*]	RAM
"lastActiveLevel"	[a]	0xFE	*"maxLevel"*	[*"minLevel"*, *"maxLevel"*]	RAM
"lastLightLevel"	0xFE	0xFE[c]	no change	0, [*"minLevel"*, *"maxLevel"*]	NVM
"powerOnLevel"	0xFE	0xFE	no change	[0,0xFF]	NVM
"systemFailureLevel"	0xFE	0xFE	no change	[0,0xFF]	NVM
"minLevel"	PHM	PHM	no change	[PHM,*"maxLevel"*]	NVM
"maxLevel"	0xFE	0xFE	no change	[*"minLevel"*,0xFE]	NVM
"fadeRate"	7	7	no change	[1,0xF]	NVM
"fadeTime"	0	0	no change	[0,0xF]	NVM
"extendedFadeTimeBase"	0	0	no change	[0,1111b]	NVM
"extendedFadeTimeMultiplier"	0	0	no change	[0,100b]	NVM
"shortAddress"	MASK (no address)	no change	no change	[0,63], MASK	NVM
"searchAddress"	[a]	0xFF FF FF	0xFF FF FF	[0,0xFF FF FF]	RAM
"randomAddress"	0xFF FF FF	0xFF FF FF	no change	[0,0xFF FF FF]	NVM
"operatingMode"	factory burn-in	no change	no change	0,[0x80,0xFF]	NVM
"initialisationState"	[a]	no change	DISABLED	[ENABLED, DISABLED, WITHDRAWN]	RAM
"writeEnableState"	[a]	DISABLED	DISABLED	[ENABLED, DISABLED]	RAM
"controlGearFailure"	[a]	[b]	FALSE[d]	[TRUE, FALSE]	RAM
"lampFailure"	[a]	[b]	FALSE[d]	[TRUE, FALSE]	RAM
"lampOn"	[a]	[b]	FALSE	[TRUE, FALSE]	RAM
"limitError"	[a]	FALSE	FALSE[d]	[TRUE, FALSE]	RAM
"fadeRunning"	[a]	FALSE	FALSE	[TRUE, FALSE]	RAM
"resetState"	TRUE	TRUE	TRUE[d]	[TRUE, FALSE]	RAM
"powerCycleSeen"	[a]	FALSE	TRUE	[TRUE, FALSE]	RAM
"gearGroups"	0x00 00 (no group)	0x00 00 (no group)	no change	[0,0xFF FF]	NVM
"sceneX" [e]	MASK	MASK	no change	[0,0xFF]	NVM

"DTR0"	[a]	no change	0x00	[0,0xFF]	RAM
"DTR1"	[a]	no change	0x00	[0,0xFF]	RAM
"DTR2"	[a]	no change	0x00	[0,0xFF]	RAM
PHM	factory burn-in	no change	no change	[1,0xFE]	ROM
"versionNumber"	~~2.1~~ 3.0	no change	no change	~~00001001b~~ 00001100b	ROM

[a] Not applicable.

[b] The value could change as a consequence of the RESET command execution.

[c] This NVM variable is excluded for "*resetState*".

[d] The value should reflect the actual situation as soon as possible.

[e] *X* is in the range 0x0 to 0xF, effectively there is one variable for each of the 16 scenes.

11 Definition of commands

11.1 General

Unused opcodes are reserved for future needs.

11.2 Overview sheets

Table 17 gives an overview of the standard commands. The special commands overview can be found in Table 18.

Table 17 – Standard commands

Command name	Address byte		Opcode byte	Ed. 1 command number	DTR0	DTR1	DTR2	Answer	Send twice	References	Command reference
	See 7.2.2	Select or bit									
DAPC (*level*)	*Device*	0	*level*	-						9.4, 9.7.3, 9.8	11.3.1
OFF	*Device*	1	0x00	0						9.7.2	11.3.2
UP	*Device*	1	0x01	1						9.7.3	11.3.3
DOWN	*Device*	1	0x02	2						9.7.3	11.3.4
STEP UP	*Device*	1	0x03	3						9.7.2	11.3.5
STEP DOWN	*Device*	1	0x04	4						9.7.2	11.3.6
RECALL MAX LEVEL	*Device*	1	0x05	5						9.7.2, 9.14.3	11.3.7
RECALL MIN LEVEL	*Device*	1	0x06	6						9.7.2, 9.14.3	11.3.8
STEP DOWN AND OFF	*Device*	1	0x07	7						9.7.2	11.3.9
ON AND STEP UP	*Device*	1	0x08	8						9.7.2	11.3.10
ENABLE DAPC SEQUENCE	*Device*	1	0x09	9						9.8	11.3.11
GO TO LAST ACTIVE LEVEL	*Device*	1	0x0A							9.7.3	11.3.12
CONTINUOUS UP	*Device*	1	0x0B							9.7.3	11.3.13

Command name	Address byte		Opcode byte	Ed. 1 command number	DTR0	DTR1	DTR2	Answer	Send twice	References	Command reference
	See 7.2.2	Select or bit									
CONTINUOUS DOWN	*Device*	1	0x0C							9.7.3	11.3.14
GO TO SCENE (*sceneNumber*)[a]	*Device*	1	0x10 + *sceneNumber*	16 to 31						9.7.3, 9.19	11.3.15
RESET	*Device*	1	0x20	32					✓	9.11.1, 10	11.4.2
STORE ACTUAL LEVEL IN DTR0	*Device*	1	0x21	33	✓				✓		11.4.3
~~SAVE PERSISTENT VARIABLES~~	~~*Device*~~	~~1~~	~~0x22~~						~~✓~~	~~9.17, 10~~	~~11.4.4~~
Reserved[b]	*Device*	1	0x22 [b]						✓		
SET OPERATING MODE (*DTR0*)	*Device*	1	0x23		✓				✓	9.9	11.4.4
RESET MEMORY BANK (*DTR0*)	*Device*	1	0x24		✓				✓	9.11.2	11.4.5
IDENTIFY DEVICE	*Device*	1	0x25						✓	9.14.3	11.4.6
SET MAX LEVEL (*DTR0*)	*Device*	1	0x2A	42	✓				✓	9.6	11.4.7
SET MIN LEVEL (*DTR0*)	*Device*	1	0x2B	43	✓				✓	9.6	11.4.8
SET SYSTEM FAILURE LEVEL (*DTR0*)	*Device*	1	0x2C	44	✓				✓	9.12	11.4.9
SET POWER ON LEVEL (*DTR0*)	*Device*	1	0x2D	45	✓				✓	9.13	11.4.10
SET FADE TIME (*DTR0*)	*Device*	1	0x2E	46	✓				✓	9.5.2	11.4.11
SET FADE RATE (*DTR0*)	*Device*	1	0x2F	47	✓				✓	9.5.3	11.4.12
SET EXTENDED FADE TIME (*DTR0*)	*Device*	1	0x30		✓				✓	9.5.4	11.4.13
SET SCENE (*DTR0, sceneX*)[a]	*Device*	1	0x40 + *sceneNumber*	64 to 79	✓				✓	9.19	11.4.14
REMOVE FROM SCENE (*sceneX*)[a]	*Device*	1	0x50 + *sceneNumber*	80 to 95					✓	9.19	11.4.15
ADD TO GROUP (*group*)[a]	*Device*	1	0x60 + *group*	96 to 111					✓		11.4.16
REMOVE FROM GROUP (*group*)[a]	*Device*	1	0x70 + *group*	112 to 127					✓		11.4.17
SET SHORT ADDRESS (*DTR0*)	*Device*	1	0x80	128	✓				✓	9.14.4	11.4.18

Command name	Address byte		Opcode byte	Ed. 1 command number	DTR0	DTR1	DTR2	Answer	Send twice	References	Command reference
	See 7.2.2	Select or bit									
ENABLE WRITE MEMORY	*Device*	1	0x81	129				✓	✓	9.10.6	11.4.19
Reserved for IEC 62386-104 (see [3])	*Device*	1	0x82		✓						
QUERY STATUS	*Device*	1	0x90	144				✓		9.16	11.5.2
QUERY CONTROL GEAR PRESENT	*Device*	1	0x91	145				✓			11.5.3
QUERY LAMP FAILURE	*Device*	1	0x92	146				✓			11.5.4
QUERY LAMP POWER ON	*Device*	1	0x93	147				✓			11.5.6
QUERY LIMIT ERROR	*Device*	1	0x94	148				✓			11.5.7
QUERY RESET STATE	*Device*	1	0x95	149				✓			11.5.8
QUERY MISSING SHORT ADDRESS	*Device*	1	0x96	150				✓		9.14.2	11.5.9
QUERY VERSION NUMBER	*Device*	1	0x97	151				✓			11.5.10
QUERY CONTENT DTR0	*Device*	1	0x98	152	✓			✓			11.5.11
QUERY DEVICE TYPE	*Device*	1	0x99	153				✓		9.18	11.5.12
QUERY PHYSICAL MINIMUM	*Device*	1	0x9A	154				✓			11.5.13
QUERY POWER FAILURE	*Device*	1	0x9B	155				✓			11.5.15
QUERY CONTENT DTR1	*Device*	1	0x9C	156		✓		✓			11.5.16
QUERY CONTENT DTR2	*Device*	1	0x9D	157			✓	✓			11.5.17
QUERY OPERATING MODE	*Device*	1	0x9E					✓		9.9	11.5.18
QUERY LIGHT SOURCE TYPE	*Device*	1	0x9F		✓	✓	✓	✓			11.5.19
QUERY ACTUAL LEVEL	*Device*	1	0xA0	160				✓			11.5.20
QUERY MAX LEVEL	*Device*	1	0xA1	161				✓			11.5.21
QUERY MIN LEVEL	*Device*	1	0xA2	162				✓			11.5.22
QUERY POWER ON LEVEL	*Device*	1	0xA3	163				✓		9.13	11.5.23

Command name	Address byte		Opcode byte	Ed. 1 command number	DTR0	DTR1	DTR2	Answer	Send twice	References	Command reference
	See 7.2.2	Select or bit									
QUERY SYSTEM FAILURE LEVEL	*Device*	1	0xA4	164				✓		9.12	11.5.24
QUERY FADE TIME/FADE RATE	*Device*	1	0xA5	165				✓			11.5.25
QUERY MANUFACTURER SPECIFIC MODE	*Device*	1	0xA6					✓		9.9	11.5.27
QUERY NEXT DEVICE TYPE	Device	1	0xA7					✓		9.18	11.5.13
QUERY EXTENDED FADE TIME	Device	1	0xA8					✓		9.5.4	11.5.26
QUERY CONTROL GEAR FAILURE	Device	1	0xAA					✓		9.16.2	11.5.4
Reserved for IEC 62386-104 (see [3])	Device	1	0xAB					✓			
QUERY SCENE LEVEL (*sceneX*)[a]	*Device*	1	0xB0 + *sceneNumber*	176 to 191				✓		9.19	11.5.28
QUERY GROUPS 0-7	*Device*	1	0xC0	192				✓			11.5.29
QUERY GROUPS 8-15	*Device*	1	0xC1	193				✓			11.5.30
QUERY RANDOM ADDRESS (H)	*Device*	1	0xC2	194				✓			11.5.31
QUERY RANDOM ADDRESS (M)	*Device*	1	0xC3	195				✓			11.5.32
QUERY RANDOM ADDRESS (L)	*Device*	1	0xC4	196				✓			11.5.33
READ MEMORY LOCATION (*DTR1, DTR0*)	*Device*	1	0xC5	197	✓	✓		✓		9.10.5	11.5.34
Application extended commands	*Device*	1	0xE0 to 0xFE	224 to 254	?	?	?	?	?	9.18	11.6
QUERY EXTENDED VERSION NUMBER	*Device*	1	0xFF	255				✓			11.6.2

[a] There is one command per scene, so there are actually 16 commands for scenes 0~~—5~~ to 15. Similarly for the 16 group commands.

[b] Reserved to maintain backward compatibility due to use in Edition 2 of IEC 62386-102:2014 (see [2]).

Table 18 – Special commands

Command name	Address byte	Opcode byte	Ed.1 command number	DTR0	DTR1	DTR2	Answer	Send twice	References	Command reference
TERMINATE	0xA1	0x00	256						9.14.2	11.7.1
DTR0 (*data*)	0xA3	*data*	257	✓						11.7.3
INITIALISE (*device*)	0xA5	*device*	258					✓	9.14.2	11.7.4
RANDOMISE	0xA7	0x00	259					✓	9.14.2	11.7.5
COMPARE	0xA9	0x00	260				✓		9.14.2	11.7.6
WITHDRAW	0xAB	0x00	261						9.14.2	11.7.7
PING	0xAD	0x00								11.7.19
SEARCHADDRH (*data*)	0xB1	*data*	264						9.14.2	11.7.8
SEARCHADDRM (*data*)	0xB3	*data*	265						9.14.2	11.7.9
SEARCHADDRL (*data*)	0xB5	*data*	266						9.14.2	11.7.10
PROGRAM SHORT ADDRESS (*data*)	0xB7	*data*	267						9.14.2	11.7.11
VERIFY SHORT ADDRESS (*data*)	0xB9	*data*	268				✓		9.14.2	11.7.12
QUERY SHORT ADDRESS	0xBB	0x00	269				✓		9.14.2	11.7.13
Reserved for IEC 62386-104 (see [3])	0xBB	0x01		✓	✓		✓			
Reserved for IEC 62386-104 (see [3])	0xBD	*data*								
Reserved for IEC 62386-104 (see [3])	0xBF	*data*								
ENABLE DEVICE TYPE (*data*)	0xC1	*data*	272						9.14.2	11.7.14
DTR1 (*data*)	0xC3	*data*	273		✓					11.7.15
DTR2 (*data*)	0xC5	*data*	274			✓				11.7.16
WRITE MEMORY LOCATION (*DTR1, DTR0, data*)	0xC7	*data*	275	✓	✓		✓		9.10.6	11.7.17
WRITE MEMORY LOCATION – NO REPLY (*DTR1, DTR0, data*)	0xC9	*data*		✓	✓				9.10.6	11.7.18

11.3 Level instructions

11.3.1 DAPC (*level*)

Upon execution of "DAPC (*level*)" (direct arc power control), *"targetLevel"* shall be calculated on the basis of *"level"*.

The transition from *"actualLevel"* to *"targetLevel"* shall start using the applicable fade time.

Refer to 9.4, 9.7.3 and 9.13 for further information.

11.3.2 OFF

"targetLevel" shall be set to 0x00 and the lamp(s) shall switch off.

The transition from *"actualLevel"* to *"targetLevel"* shall be immediate and the light output shall be adjusted as quickly as possible.

Refer to 9.7.2 for further information.

11.3.3 UP

Dim up using a 200 ms fade with the set fade rate. *"targetLevel"* shall be calculated on the basis of *"actualLevel"* and the set fade rate.

To ensure that there is a reaction to the command, at least one step (final "*targetLevel*" = calculated "*targetLevel*" + 1) shall be made upon execution of the first command of an iteration. After that first step, the next steps shall be executed using the specified fade rate while the fading is running. Every "UP" instruction executed as a part of an iteration shall cause the 200 ms fade to be restarted and *"targetLevel"* to be recalculated on the basis of *"actualLevel"* and the set fade rate.

There shall be no change to *"actualLevel"* if *"actualLevel"* is at *"maxLevel"* or 0x00.

Refer to 9.5.1, 9.7.3 and 9.8.2 for further information.

11.3.4 DOWN

Dim down using a 200 ms fade with the set fade rate. *"targetLevel"* shall be calculated on the basis of *"actualLevel"* and the set fade rate.

To ensure that there is a reaction to the command, at least one step (final "*targetLevel*" = calculated "*targetLevel*" − 1) shall be made upon execution of the first command of an iteration. After that first step, the next steps shall be executed using the specified fade rate while the fading is running. Every "DOWN" instruction executed as a part of an iteration shall cause the 200 ms fade to be restarted and *"targetLevel"* to be recalculated on the basis of *"actualLevel"* and the set fade rate.

There shall be no change to *"actualLevel"* if *"actualLevel"* is at *"minLevel"* or 0x00.

Refer to 9.5.1, 9.7.3 and 9.8.2 for further information.

11.3.5 STEP UP

"targetLevel" shall be set to:

- if *"targetLevel"* = 0: 0x00;
- if *"minLevel"* ≤ *"targetLevel"* < *"maxLevel": "targetLevel"* + 1;
- if *"targetLevel"* = *"maxLevel": "maxLevel".*

The transition from *"actualLevel"* to *"targetLevel"* shall be immediate and the light output shall be adjusted as quickly as possible.

Refer to 9.4 and 9.5.9 for further information.

11.3.6 STEP DOWN

"targetLevel" shall be set to:

- if *"targetLevel"* = 0: 0x00;
- if *"minLevel"* < *"targetLevel"* ≤ *"maxLevel": "targetLevel"* − 1;
- if *"targetLevel" = "minLevel": "minLevel".*

The transition from *"actualLevel"* to *"targetLevel"* shall be immediate and the light output shall be adjusted as quickly as possible.

Refer to 9.4 and 9.5.9 for further information.

11.3.7 RECALL MAX LEVEL

When the *"initialisationState"* is DISABLED, *"targetLevel"* and *"actualLevel"* shall be set to *"maxLevel"* immediately and the light output shall be adjusted as quickly as possible.

Refer to 9.7.2 for further information.

When the *"initialisationState"* is not DISABLED, the control gear shall set "*actualLevel*" and "*targetLevel*" to "*maxLevel*", and then adjust the light output as quickly as possible to 100 % temporarily ignoring "*maxLevel*" and "*actualLevel*".

If the device is unable to visually identify itself in this way, 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 RECALL MAX LEVEL and RECALL MIN LEVEL commands being executed in an alternating sequence.

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

Identification shall be stopped immediately when the *"initialisationState"* changes to DISABLED and upon execution of any instruction other than INITIALISE (*device*), RECALL MIN LEVEL, RECALL MAX LEVEL or IDENTIFY DEVICE.

When the *"initialisationState"* changes to DISABLED, the identification shall stop immediately.

Refer to 9.14.3 for further information.

11.3.8 RECALL MIN LEVEL

When the *"initialisationState"* is DISABLED, *"targetLevel"* and *"actualLevel"* shall be set to *"minLevel"* immediately and the light output shall be adjusted as quickly as possible.

Refer to 9.7.2 for further information.

When *"initialisationState"* is not DISABLED, the control gear shall set "*actualLevel*" and "*targetLevel*" to "*minLevel*" and then adjust the light output as quickly as possible to its PHM level temporarily ignoring "*minLevel*" and "*actualLevel*". If, however, the PHM is not visibly significantly different from 100 %, then the lamp shall be temporarily switched off instead.

If the device is unable to visually identify itself in this way, 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 RECALL MAX LEVEL and RECALL MIN LEVEL commands being executed in an alternating sequence.

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

Identification shall be stopped immediately when the *"initialisationState"* changes to DISABLED and upon execution of any instruction other than INITIALISE (*device*), RECALL MIN LEVEL, RECALL MAX LEVEL or IDENTIFY DEVICE.

When the *"initialisationState"* changes to DISABLED, identification shall stop immediately.

Refer to 9.14.3 for further information.

11.3.9 STEP DOWN AND OFF

"targetLevel" shall be set to:

- if *"targetLevel"* = 0: 0x00;
- if *"minLevel"* < *"targetLevel"* ≤ *"maxLevel"*: *"targetLevel"* − 1;
- if *"targetLevel"* = *"minLevel"*: 0x00.

The transition from *"actualLevel"* to *"targetLevel"* shall be immediate and the light output shall be adjusted as quickly as possible.

Refer to 9.4 and 9.5.9 for further information.

11.3.10 ON AND STEP UP

"targetLevel" shall be set to:

- if *"targetLevel"* = 0: *"minLevel"*;
- if *"minLevel"* ≤ *"targetLevel"* < *"maxLevel"*: *"targetLevel"* + 1;
- if *"targetLevel"* ≥ *"maxLevel"*: *"maxLevel"*.

The transition from *"actualLevel"* to *"targetLevel"* shall be immediate and the light output shall be adjusted as quickly as possible.

Refer to 9.4 and 9.5.9 for further information.

11.3.11 ENABLE DAPC SEQUENCE

Indicates the start of a command iteration of "DAPC (*level*)" commands.

Refer to 9.8.3 for further information.

11.3.12 GO TO LAST ACTIVE LEVEL

Upon execution of this command *"targetLevel"* shall be calculated based on *"lastActiveLevel"*.

The transition from *"actualLevel"* to *"targetLevel"* shall start using the set fade time.

Refer to 9.7.3 and 9.4 for further information.

11.3.13 CONTINUOUS UP

Dim up using the set fade rate. "*targetLevel*" shall be set to "*maxLevel*" and a fade shall be started using the set fade rate. The fade shall stop when "*maxLevel*" is reached.

There shall be no change to "*actualLevel*" if "*actualLevel*" is at "*maxLevel*" or 0x00.

Refer to 9.7.3 and 9.5.6.2 for further information.

11.3.14 CONTINUOUS DOWN

Dim down using the set fade rate. "*targetLevel*" shall be set to "*minLevel*" and a fade shall be started using the set fade rate. The fade shall stop when "*minLevel*" is reached.

There shall be no change to "*actualLevel*" if "*actualLevel*" is at "*minLevel*" or 0x00.

Refer to 9.7.3 and 9.5.6.2 for further information.

11.3.15 GO TO SCENE (*sceneNumber*)

The control gear shall react depending on the actual value of *"sceneX"* where *X* is derived from *sceneNumber*:

- if *"sceneX"* = *MASK*: the command shall not affect *"targetLevel";*
- in all other cases: internally "DAPC (*level*)", with *level* equal to *"sceneX"* shall be executed.

NOTE Using "DAPC (*level*)" implies the transition is made using the set fade time.

Refer to 9.19 and 11.3.1 for further information.

11.4 Configuration instructions

11.4.1 General

~~Device~~ Configuration instructions are used to change the configuration and/or the mode of operation of the control gear. For this reason a device configuration instruction shall be discarded, unless it is accepted twice according to the requirements as stated in IEC 62386-101:~~2014 and IEC62386-101:2014/AMD1:2018~~2022, 9.4.

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

- The instruction shall be ignored if so required by the provisions of 9.7.
- The control gear shall not reply to the instruction.

11.4.2 RESET

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

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

Refer to 9.11.1 and Table 16 for further information.

11.4.3 STORE ACTUAL LEVEL IN DTR0

The *"actualLevel"* shall be stored in *"DTR0"*.

~~11.4.4 SAVE PERSISTENT VARIABLES~~ 25

~~The control gear shall physically store all variables identified in Table 14 as non-volatile memory (NVM). This shall include all application extended NVM variables defined in the applicable parts 2xx.~~

~~The control gear might not react to commands after execution of this command. Control gear shall start to react properly to commands no later than 300 ms after the execution of the instruction has been started.~~

~~During processing of this command, the light output may fluctuate. After processing is completed, the light output shall be at the level as expected before the execution of this command, based on *"targetLevel"* and the transition that was active (if any).~~

~~This command is recommended to be used typically after commissioning. Due to the limited number of write-cycles of persistent memory and due to the fact that there might be a visible reaction, the control devices should limit the use of this command.~~

~~As there might be visual artefacts, it is recommended to use this command only during the off state.~~

~~Refer to Table 14 and subclause 9.17 for further information.~~

11.4.4 SET OPERATING MODE (*DTR0*)

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

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

Refer to 9.9 for further information.

11.4.5 RESET MEMORY BANK (*DTR0*)

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

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

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

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

Refer to 9.11.2 for further information.

11.4.6 IDENTIFY DEVICE

The control gear shall start or restart a 10 s ± 1 s timer. While the timer is running, a procedure shall run which enables an observer to distinguish any control gear running this process from any devices (of the same type) which are not running it. If the timer expires, identification shall stop.

During identification no variables shall be affected except when 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 %, MIN, MAX and "*actualLevel*" being in effect temporarily ignored.

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

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.

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).

Identification can be used during commissioning in that it allows the installer to, for example, allocate the particular identified device to a particular device group.

The indication can be done, for example by flashing a LED, by producing a sound or other visual or audible means. The exact process used to identify is manufacturer specific and should be described in the manual.

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

Refer to 9.14.3 for further information.

11.4.7 SET MAX LEVEL (*DTR0*)

"maxLevel" shall be set to:

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

If, as a result of setting a new max level *"actualLevel"* > *"maxLevel"*, *"targetLevel"* shall be calculated on the basis of *"maxLevel"*. The transition from *"actualLevel"* to *"targetLevel"* shall start immediately and the light output shall be adjusted as quickly as possible.

Refer to 9.7.2 for further information.

11.4.8 SET MIN LEVEL (*DTR0*)

"minLevel" shall be set to:

- if 0 ≤ *"DTR0"* ≤ PHM: PHM;
- if *"DTR0"* ≥ *"maxLevel"* or MASK: *"maxLevel"*;
- in all other cases: *"DTR0"*.

If *"actualLevel"* > 0 and as a result of setting a new min level *"actualLevel"* < *"minLevel"*, *"targetLevel"* shall be calculated on the basis of *"minLevel"*. The transition from *"actualLevel"* to *"targetLevel"* shall be immediate and the light output shall be adjusted as quickly as possible. Refer to 9.7.2 for further information.

11.4.9 SET SYSTEM FAILURE LEVEL (*DTR0*)

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

Refer to 9.12 for further information.

11.4.10 SET POWER ON LEVEL (*DTR0*)

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

Refer to 9.13 for further information.

11.4.11 SET FADE TIME (*DTR0*)

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

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

If "*fadeTime"* is not equal to 0, the fade time shall be calculated on the basis of *"fadeTime"*. If "*fadeTime"* is equal to 0, the extended fade time shall be used.

If a new fade time is stored during a running fade process, this process shall be finished first before the new value is used in the following fade.

Refer to 9.5, 9.7.3, 11.4.13 and 11.7.19 for further information.

11.4.12 SET FADE RATE (*DTR0*)

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"*. If a new fade rate is stored during a running fade process, this process shall be finished first before the new value is used in the following fade.

Refer to 9.5 and 9.7.3 for further information.

11.4.13 SET EXTENDED FADE TIME (*DTR0*)

The *"extendedFadeTimeBase"* and "*extendedFadeTimeMultiplier*" shall be set to a value according to the following steps:

- If "*DTR0"* > 0x4F (0100 1111~~b~~):
  - "*extendedFadeTimeBase*" shall be set to 0;
  - "*extendedFadeTimeMultiplier*" shall be set to 0;

effectively selecting a fade as quickly as possible.

- For all other cases:
  - "*extendedFadeTimeBase*" shall be set to AAAAb where *"DTR0"* = xxxxAAAAb;
  - "*extendedFadeTimeMultiplier*" shall be set to YYYb where *"DTR0"* = xYYYxxxxb.
- The fade time shall be calculated by multiplying the base value and the multiplier.

If a new fade time is stored during a running fade process, this process shall be finished first before the new value is used in the following fade.

Refer to 9.5.4 for further information.

11.4.14 SET SCENE (*DTR0, sceneX*)

This command actually comprises 16 commands, one for each scene. This is accomplished by selecting a block of 16 consecutive opcodes.

Upon execution of "SET SCENE (*DTR0, sceneX*)", the scene number shall be derived from the opcode: *sceneNumber* = opcode – 0x40. This identifies the "*sceneX"* to be used.

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

Refer to 9.19 for further information.

11.4.15 REMOVE FROM SCENE (*sceneX*)

This command actually comprises 16 commands, one for each scene. This is accomplished by selecting a block of 16 consecutive opcodes.

Upon execution of "REMOVE FROM SCENE (*sceneX*), the scene number shall be derived from the opcode: *sceneNumber* = opcode – 0x50. This identifies the "*sceneX"* to be used.

"sceneX" shall be set to MASK. This effectively removes the control gear as member from the scene.

Refer to 9.19 for further information.

11.4.16 ADD TO GROUP (*group*)

This command actually comprises 16 commands, one for each group. This is accomplished by selecting a block of 16 consecutive opcodes.

Upon execution of "ADD TO GROUP (*group*)", *group* shall be derived from the opcode: *group* = opcode – 0x60. This identifies the *group* to be used.

bit[*group*] of *"gearGroups"* shall be set to TRUE. This implies that the control gear is a member of this group.

11.4.17 REMOVE FROM GROUP (*group*)

This command actually comprises 16 commands, one for each group. This is accomplished by selecting a block of 16 consecutive opcodes.

Upon execution of "REMOVE FROM GROUP (*group*)", *group* shall be derived from the opcode: *group* = opcode – 0x70. This identifies the *group* to be used.

bit[*group*] of *"gearGroups"* shall be set to FALSE. This implies that the control gear is not a member of this group.

11.4.18 SET SHORT ADDRESS (*DTR0*)

"shortAddress" shall be set to:

- if *"DTR0"* = MASK: MASK (effectively deleting the short address);
- if *"DTR0"* = 1xxxxxxxb or xxxxxxx0b: no change;
- in all other cases (0AAAAAA1b): 00AAAAAAb.

11.4.19 ENABLE WRITE MEMORY

"writeEnableState" shall be set to ENABLED.

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

Refer to 9.10.6 for further information.

11.5 Queries

11.5.1 General

Queries are used to retrieve property values from a control gear. The addressed control gear returns the queried property value in a backward frame.

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

- The query shall be ignored if so required by the provisions of 9.7.

When applicable, the query shall be discarded if any of the parameter values (in *"DTR0"*, *"DTR1"* and *"DTR2"*) are outside the range of validity of the addressed device variables, as given in Table 16.

11.5.2 QUERY STATUS

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

Refer to 9.16 for further information.

11.5.3 QUERY CONTROL GEAR PRESENT

The answer shall be YES.

11.5.4 QUERY CONTROL GEAR FAILURE

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

11.5.5 QUERY LAMP FAILURE

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

11.5.6 QUERY LAMP POWER ON

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

11.5.7 QUERY LIMIT ERROR

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

11.5.8 QUERY RESET STATE

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

11.5.9 QUERY MISSING SHORT ADDRESS

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

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

11.5.10 QUERY VERSION NUMBER

The answer shall be ~~the content of memory bank 0 location 0x16~~ "*versionNumber*".

See 4.2 and Table 16.

11.5.11 QUERY CONTENT DTR0

The answer shall be *"DTR0"*.

11.5.12 QUERY DEVICE TYPE

The answer shall be:

- if no part of the IEC 62386-2xx series is implemented: 254;
- if one device type~~/~~ or feature is ~~supported~~ implemented: the device type~~/~~ or feature number;
- if more than one device type~~/~~ or feature is ~~supported~~ implemented: MASK.

The coding of the device types~~/~~ and features shall be as specified in the relevant parts of the IEC 62386-2xx series.

Refer to 9.18 and 11.5.13 for further information.

11.5.13 QUERY NEXT DEVICE TYPE

The answer shall be:

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

The sequence shall be cancelled if a 16-bit forward frame is received which meets any of the following conditions:

- the address byte differs, or
- the frame contains a command other than QUERY DEVICE TYPE and QUERY NEXT DEVICE TYPE. **26**

The sequence of commands shall only be accepted as long as they use the same address byte. Multi-master transmitters shall send such sequence as a transaction. The coding of the device types~~/~~ and features shall be as specified in the relevant parts of the IEC 62386-2xx series.

Refer to 9.18 and 11.5.12 for further information.

11.5.14 QUERY PHYSICAL MINIMUM

The answer shall be PHM.

11.5.15 QUERY POWER FAILURE

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

11.5.16 QUERY CONTENT DTR1

The answer shall be *"DTR1"*.

11.5.17 QUERY CONTENT DTR2

The answer shall be *"DTR2"*.

11.5.18 QUERY OPERATING MODE

The answer shall be *"operatingMode"*.

Refer to 9.9 for further information.

11.5.19 QUERY LIGHT SOURCE TYPE

The answer shall be the number of the light source type given in Table 19.

Table 19 – Light source type encoding

Type of light source	Encoding	Lamp failure detection	
		Open circuit (lamp disconnected)	Short circuit
Low pressure fluorescent	0	✓	
HID	2	✓	
Low voltage halogen	3	✓	
Incandescent	4	✓	
LED	6	✓[a]	✓[a]
OLED	7	✓[a]	✓[a]
Other than listed above	252	✓	
Unknown light source type[b]	253	✓[d]	[d]
No light source[c]	254	[d]	[d]
Multiple light source types	MASK	✓[e]	✓[e]
Reserved	1, 5, [8,251]		

[a] Testing shall be done with light output of at least 5 %.

[b] Typically used in the case of signal conversion, for example 1 V to 10 V.

[c] Used in cases where no light source is connected, for example a relay.

[d] See 9.16.3.

[e] Depending on the supported light source types.

When MASK is answered the content of DTR0 shall contain a value representing the first light source type, DTR1 shall represent the second light source type, and DTR2 shall represent the third light source type.

When exactly two different light source types are available, DTR2 shall contain 254, indicating "no light source".

When more than three different light source types are available, DTR2 shall contain 255.

11.5.20 QUERY ACTUAL LEVEL

The answer shall be:

- if *"actualLevel"* = 0x00: 0x00 (see also 9.13);
- In all other cases:
  - during startup: MASK;
  - no light output (e.g. due to total lamp failure, control gear failure) while light output is expected: MASK;
  - in all other cases: *"actualLevel"*.

11.5.21 QUERY MAX LEVEL

The answer shall be *"maxLevel"*.

11.5.22 QUERY MIN LEVEL

The answer shall be *"minLevel"*.

11.5.23 QUERY POWER ON LEVEL

The answer shall be *"powerOnLevel"*.

Refer to 9.13 for further information.

11.5.24 QUERY SYSTEM FAILURE LEVEL

The answer shall be *"systemFailureLevel"*.

Refer to 9.12 for further information.

11.5.25 QUERY FADE TIME/FADE RATE

The answer shall be XXXX YYYYb, where XXXXb equals *"fadeTime"* and YYYYb equals *"fadeRate"*.

11.5.26 QUERY EXTENDED FADE TIME

The answer shall be 0 XXX YYYYb, where XXXb equals "*extendedFadeTimeMultiplier*" and YYYYb equals "*extendedFadeTimeBase*".

11.5.27 QUERY MANUFACTURER SPECIFIC MODE

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

11.5.28 QUERY SCENE LEVEL (*sceneX*)

This command actually comprises 16 commands, one for each scene. This is accomplished by selecting a block of 16 consecutive opcodes.

Upon execution of "QUERY SCENE LEVEL (*sceneX*)", the scene number shall be derived from the opcode: *sceneNumber* = opcode – 0xB0. This identifies the "*sceneX"* to be used.

The answer shall be *"sceneX"*.

Refer to 9.19 for further information.

11.5.29 QUERY GROUPS 0-7

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

The membership of groups 0 to 7 shall be represented as an 8-bit value, with one bit for each group. "0" shall be interpreted as not a member, and "1" shall be interpreted as member of the group. Bit[X] shall represent membership of group X, where X is in the range [0,7].

11.5.30 QUERY GROUPS 8-15

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

The membership of groups 8 to 15 shall be represented as an 8-bit value, with one bit for each group. "0" shall be interpreted as not a member, and "1" shall be interpreted as member of the group. Bit[X] shall represent membership of group X + 8, where X is in the range [0,7].

11.5.31 QUERY RANDOM ADDRESS (H)

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

11.5.32 QUERY RANDOM ADDRESS (M)

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

11.5.33 QUERY RANDOM ADDRESS (L)

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

11.5.34 READ MEMORY LOCATION (*DTR1, DTR0*)

The query shall be discarded if the addressed memory bank is not implemented.

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

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

NOTE 1 This allows ~~holes~~ gaps in the memory bank implementation.

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

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

Refer to 9.10.5 for further information.

11.6 Application extended commands

11.6.1 General

"ENABLE DEVICE TYPE (*data*)" shall be executed before an application extended command to enable the correct device type~~/~~ or feature command set. For further requirements, see command "ENABLE DEVICE TYPE (*data*)" and 11.7.14.

If "ENABLE DEVICE TYPE (*data*)" is discarded or not received ~~directly~~ before an application extended command is accepted, the application extended command shall be discarded.

The definitions of the extended commands are in the relevant parts of the IEC 62386-2xx series.

Refer to 9.18 for further information.

11.6.2 QUERY EXTENDED VERSION NUMBER

The answer shall be the version number of the relevant part of the IEC 62386-2xx series for the corresponding device type~~/~~ or feature as an 8-bit number.

The answer shall be:

- if the enabled device type~~/~~ or feature is not implemented: NO;
- if the enabled device type~~/~~ or feature is supported: the version number belonging to the device type~~/~~ or feature ~~number~~.

Refer to 9.18 for further information.

11.7 Special commands

11.7.1 General

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

11.7.2 TERMINATE

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

- Initialisation, "*initialisationState*" shall be set to DISABLED.
- Identification, whether started as part of initialisation (using RECALL MAX LEVEL, RECALL MIN LEVEL) or as a standard operation (IDENTIFY DEVICE) shall be stopped.

The command could also terminate other processes as identified in the relevant parts of the IEC 62386-2xx series.

Refer to 9.14.2 for further information.

11.7.3 DTR0 (*data*)

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

Refer to 9.10 for further information.

11.7.4 INITIALISE (*device*)

This instruction shall be discarded, unless it is accepted twice according to the requirements as stated in IEC 62386-101:~~2014 and IEC62386-101:2014/AMD1:2018~~2022, 9.4.

Only devices matching the given *device* shall respond to the instruction, as specified in Table 20:

Table 20 – Device addressing with "INITIALISE (*device*)"

Device	Responsive device(s)
0AAAAAA1b	Device(s) with *"shortAddress"* equal to 00AAAAAAb
11111111b	Control gear without *"shortAddress"* shall react
00000000b	All control gear shall react
Other	None

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

Refer to 9.14.2 for further information.

11.7.5 RANDOMISE

This instruction shall be discarded, unless it is accepted twice according to the requirements as stated in IEC 62386-101:~~2014 and IEC62386-101:2014/AMD1:2018~~2022, 9.4.

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

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

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

There shall be no reply to this instruction.

Refer to 9.14.2 for further information.

11.7.6 COMPARE

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

If executed, the control gear shall answer:

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

Refer to 9.14.2 for further information.

11.7.7 WITHDRAW

The instruction shall be discarded unless the following conditions hold:

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

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

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

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

Refer to 9.14.2 for further information.

11.7.8 SEARCHADDRH (*data*)

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

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

Refer to 9.14.2 for further information.

11.7.9 SEARCHADDRM (*data*)

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

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

Refer to 9.14.2 for further information.

11.7.10 SEARCHADDRL (*data*)

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

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

Refer to 9.14.2 for further information.

11.7.11 PROGRAM SHORT ADDRESS (*data*)

The instruction shall be discarded unless the following conditions hold:

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

If executed, *"shortAddress"* shall be set as follows:

- if *data* = MASK: MASK (effectively deleting the short address);
- if *data* = 1xxxxxxxb or xxxxxxx0b: no change;
- in all other cases (0AAAAAA1b): 00AAAAAAb.

Refer to 9.14.2 for further information.

11.7.12 VERIFY SHORT ADDRESS (*data*)

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

If executed, the answer shall be YES if *"shortAddress"* is equal to 00AAAAAAb for *data* given by 0AAAAAA1b, and NO otherwise.

Refer to 9.14.2 for further information.

11.7.13 QUERY SHORT ADDRESS

The query shall be discarded if:

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

If executed, the answer shall be 0AAAAAA1b, where "*shortAddress*" is equal to 00AAAAAAb, or MASK, in case "*shortAddress*" equals MASK.

Refer to 9.14.2 for further information.

11.7.14 ENABLE DEVICE TYPE (*data*)

This instruction shall select the device type or feature for which the next application extended command (refer to 11.6) is valid. Executing "ENABLE DEVICE TYPE (*data*)" shall cancel any previous selection of a device type~~/~~ or feature. The selection is only valid for the next application extended command.

If a non-application extended command is accepted after executing "ENABLE DEVICE TYPE (*data*)", the enabling of the device type~~/~~ or feature shall be cancelled and that command shall be executed according to its specification.

A control gear shall not react to commands which belong to the application extended commands if *data* equals MASK, 254 or represents a device type~~/~~ or feature not supported by this control gear.

The device types~~/~~ and features shall be coded as specified in the relevant parts of the IEC 62386-2xx series.

Application controllers should be able to identify individual ~~gears~~ control gear and store the relationship between ~~the~~ each control gear's ~~individual~~ address and the device types~~/~~ and features, in persistent memory.

11.7.15 DTR1 (*data*)

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

Refer to 9.10 for further information.

11.7.16 DTR2 (*data*)

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

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

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

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

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

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

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

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

If the selected memory bank location is

- not implemented, or
- above the last accessible memory location, or
- locked (see 9.10.2), or
- not writeable,

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

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

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

Refer to 9.10 for further information.

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

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

Refer to 9.10 for further information.

11.7.19 PING

The ping command is used by single master application controllers (see IEC 62386-103) to indicate their presence. The ping command shall be ignored by the control gear.

~~12 Test procedures~~

~~Void~~

Annex A
(informative)

Examples of algorithms

A.1 Random address allocation

The control gear are connected to a control device that uses random address allocation for setup of the system.

a) Start the algorithm with "INITIALISE (*device*)" which enables the addressing commands for a time period of 15 min.

b) Send "RANDOMISE"; all control gear choose a *randomAddress* so that 0 ≤ *randomAddress* ≤ $+2^{24}-2$.

c) The control device searches the control gear with the lowest random address by means of an algorithm which uses "SEARCHADDRH (*data*)", "SEARCHADDRM (*data*)", "SEARCHADDRL (*data*)" and "COMPARE". The control gear with the lowest random address is found. At this point, the control device needs to be able to handle different timing in backward frames coming from different control gear. Also, there is a chance that the control gear generated the same *randomAddress* in which case randomisation should be restarted for the remaining gear.

d) The short address is programmed to the control gear found with aid of "PROGRAM SHORT ADDRESS (*data*)".

e) "VERIFY SHORT ADDRESS (*data*)" can be used to verify the correct programming.

f) The found control gear can be identified by using "IDENTIFY DEVICE", or use an alternating sequence of "RECALL MAX LEVEL" and "RECALL MIN LEVEL" with the programmed short address to record the local position of the respective control gear.

g) If needed, the short address can be changed going back to step d).

h) The control gear found shall be removed from the search process by means of "WITHDRAW".

i) Repeat from step c) on until no further control gear can be found. Use "INITIALISE (*device*)" to prolong the 15 min timer if needed.

j) Stop the process with "TERMINATE".

In the event two or more control gear have the same short address, restart the addressing procedure only for these control gear with "INITIALISE (*device*)" (using the short address in the second byte) followed by steps b) to j).

A.2 One single control gear connected to the control device

Only one control gear is connected to a control device that uses the following algorithm to program a short address.

a) Transmit the new short address (0AAA AAA1b) by "DTR0 (*data*)".

b) Verify the content of the DTR0 by "QUERY CONTENT DTR0".

c) Send "SET SHORT ADDRESS (*DTR0*)" twice in accordance with the requirements ~~as stated~~ specified in IEC 62386-101:~~2014 and IEC62386-101:2014/AMD1:2018~~2022, 9.4.

A.3 Using application extended commands

A control device using application extend commands needs to detect which application extended commands are supported by the different control gear. The following algorithm can be used:

a) Initialisation process and address allocation.

b) Query the device type~~/~~ or feature of every control gear in the system. If the received answer is 'MASK' the device supports more than one device type or feature. In this case the following procedure can be used to get a list of the device types the control gear belongs to:

   1) Send the "QUERY EXTENDED VERSION NUMBER" command preceded by "ENABLE DEVICE TYPE (*data*)" with *data* equal to "0". If there is an answer, the control gear belongs to device type 0.

   2) Repeat step a) with all other device types and features supported by the control device.

c) The control device shall send "ENABLE DEVICE TYPE (*data*)" before every application extended command.

Annex B
(normative)

High resolution dimmer

A high resolution dimmer is not mandatory. However, if it is implemented, it shall be implemented according to this Annex B.

The *"actualLevel"* shall report the nearer level, after rounding of an internal value as can be seen in Figure B.1.

Light output shall always match a discrete point on the selected dimming curve, except when:

- a fade is running (via ideal, or high resolution internal curve);
- a fade is stopped before the fade time has elapsed;
- another dimming curve is selected;
- a level was programmed with another dimming curve (typically the scenes, including power on level and system failure level, store the level as well as the corresponding dimming curve, to allow representing in other curves and back without loss).

When light output is "in between" two discrete points on the selected dimming curve:

- *"actualLevel"* is set to the nearest of these two points;
- a new fade starts from the actual light output (which ~~may~~ will possibly not match a discrete point on the selected dimming curve) and ends at the target light output (which ~~may~~ will possibly not match a discrete point on the selected dimming curve, i.e. when a preset is used that was set using another dimming curve);
- the fade shall always follow the ideal or internal high resolution curve without making jumps to a discrete point on the selected dimming curve;
- there are some consequences for testing:
  - after stopping a fade, or switching dimming curve, a first step ~~might~~ can be less than or more than a full step in the active dimming curve;
  - testing with levels from more than one dimming curve at a time is complex, perhaps better avoided.

Behaviour that can be experienced:

- theoretically a minimal fade is possible while *"actualLevel"* is not changed;
- direct arc power command (DAPC) to the actual level ~~might~~ can fade from the point on the high resolution dimmer to the discrete point (less than half a dim step);
- a STEP UP or STEP DOWN can, in the worst case, result in a "half step" delta or "one and a half step"; for example a high resolution dimmer that ended at stepsize + 0,49 stepsize: response to step up is 0,51 stepsize, while the response to step down is 1,49 stepsize to end at a discrete step;
- the "fade running" status is set to "1" at fade time start and lasts until FadeTime has elapsed, even when "Actual Level" is already at the final level;
- actual level always represents the nearer discrete point on the dimming curve.

Special cases:

- When a fade is started from the "Lamp off" condition, the first step from off to "Min Level" ~~must~~ shall be made at fade start time. The step from off to min level is not part of the fade time.
- When a fade is started to the "Lamp off" condition, the last step from "Min Level" to off ~~must~~ shall be made at fade start time + FadeTime. The step from min level to off is not part of the fade time.

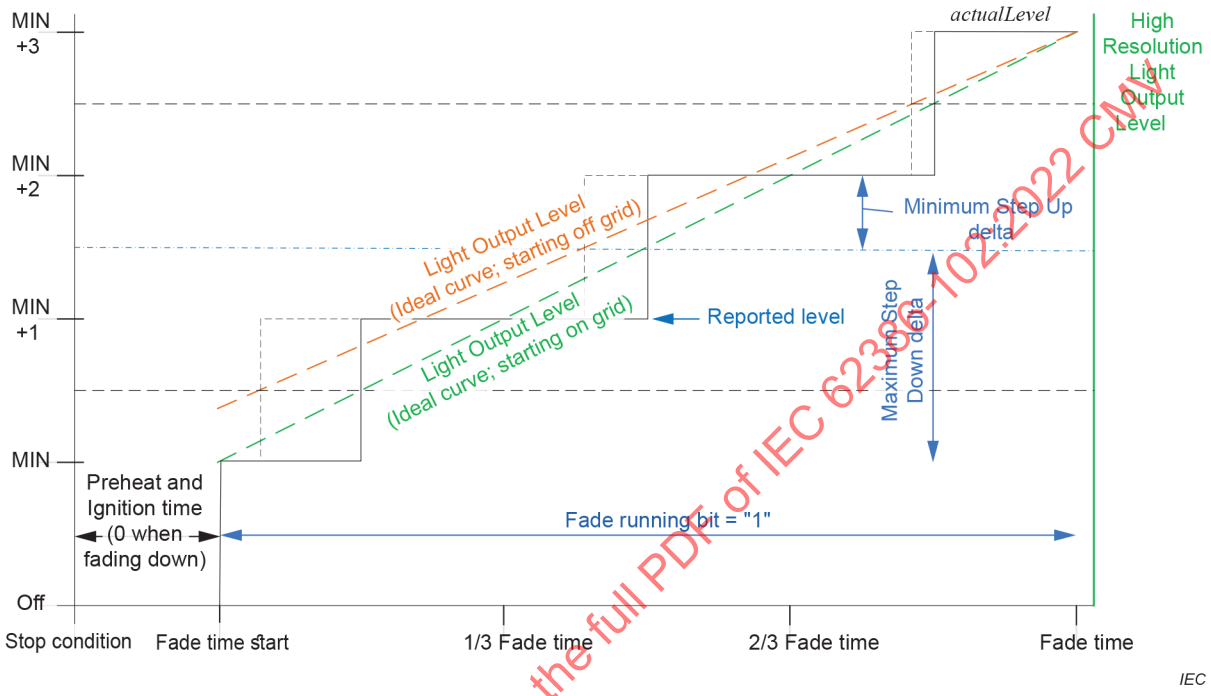


Figure B.1 – Level behaviour in case of off-grid starting points

Bibliography

~~IEC 60598-1, *Luminaires – Part 1: General requirements and tests*~~

~~IEC 60669-2-1, *Switches for household and similar fixed electrical installations – Part 2-2, Particular requirements – Electronic switches*~~

~~IEC 60921, *Ballasts for tubular fluorescent lamps – Performance requirements*~~

~~IEC 60923, *Auxiliaries for lamps – Ballasts for discharge lamps (excluding tubular fluorescent lamps – Performance requirements*~~

~~IEC 60925, *D.C.-supplied electronic ballasts for tubular fluorescent lamps – performance requirements*~~

~~IEC 61347 (all parts), *Lamp controlgear*~~

~~IEC 61547, *Equipment for general lighting purposes – EMC immunity requirements*~~

~~CISPR 15, *Limits and methods of measurement of radio disturbance characteristics of electrical lighting and similar equipment*~~

[1] GS1 *General Specifications*, latest version, available at: www.gs1.org

[2] IEC 62386-102:~~2009~~2014[2], *Digital addressable lighting interface – Part 102: General requirements – Control gear*

[3] IEC 62386-104, *Digital addressable lighting interface – Part 104: General requirements – Wireless and alternative wired system components*

[2] The second edition will be withdrawn and replaced upon publication of this third edition.

List of comments

1 Figure 1 is updated since the IEC 62386 series was extended by new parts.

2 This is primarily used for memory bank content.

3 This is a simpler notation of the formula.

4 This is a simpler notation of the formula.

5 The paragraph is rephrased to make it clearer. There is no technical change.

6 The paragraph is rephrased to make it clearer. There is no technical change.

7 The RAM and NVM memory types are sub-divided into RO and RW types. This distinction is required in IEC 62386 parts 250 to 253, and may be used in other parts in the future.

This table clarifies the volatility and possibility for modification over the interface, for each memory type.

8 Protectable memory locations have been introduced in IEC 62386 parts 25x.

9 These requirements are moved to Subclause 9.10.5.3.

10 This paragraph is now incorporated in Subclause 9.10.5.2.

11 Several multi-byte values have been introduced in IEC 62386 parts 252 and 253. The requirements here allow a consistent method to reliably read multi-byte values. Starting a read operation from the first byte fetches the latest value, and latches the remaining bytes, allowing them to be read without the possibility of their values changing before they can be read out.

12 Unimplemented locations now differ depending on whether MASK is part of the allowed range of validity or not; this is related to protectable memory locations described in Subclause 9.10.4.

13 Temporarily unavailable locations with value TMASK are now used in IEC 62386 parts 252 and 253.

14 This latching mechanism of complete memory banks has been introduced in IEC 62386 parts 252 and 253. It allows for the read-out of a memory bank's contents over a period of time, with all content relating to the same point in time (the time when the bank was latched).

15 This mechanism for writing multi-byte values is used in IEC 62386 part 253.

16 "Current bus unit configuration" is a new and optional feature, described in Subclause 9.20.

17 See IEC 62386-251 for an allowed format of these memory locations.

18 The impact on "*limitError*" was not clearly defined in previous edition 2.0.

19 This table is revised to reflect the expected behaviour more clearly.

20 This note is added due to a common mistake in implementing this standard. The term "instruction" is defined in IEC 62386-101.

21 This is not a technical change from previous edition 2.0, instead it clarifies that "*lampFailure*" cannot be changed to TRUE when a lamp failure is not currently being detected.

22 The impact on "*limitError*" at power-on was not clearly defined in previous edition 2.0.

23 Storing NVM variables is changed in a way that control gear must now automatically save NVM variables to flash (non-volatile memory), ensuring they are saved in case a power-cycle occurs within a defined time of the changed variables. The control gear may no longer rely on a command to force this save to non-volatile memory. Therefore the command SAVE PERSISTENT VARIABLES is removed.

24 This is a new and optional feature. Some control gear can be programmed in the factory to operate in different bus unit configurations, whereas the GTIN remains the same. This edition 3.0 now provides a new byte in memory bank 0 to read out the current configuration.

An example is a 2-channel control gear, which can operate as one logical device type 8 unit or as two logical device type 6 units.

25 Storing NVM variables is changed in a way that control gear must now automatically save NVM variables to flash (non-volatile memory), ensuring they are saved in case a power-cycle occurs within a defined time of the changed variables. The control gear may no longer rely on a command to force this save to non-volatile memory. Therefore the command SAVE PERSISTENT VARIABLES is removed.

26 In previous edition 2.0 it was not clearly defined how to react in these conditions.

IEC 62386-102

Edition 3.0 2022-11

INTERNATIONAL STANDARD

NORME INTERNATIONALE

colour inside

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

Interface d'éclairage adressable numérique – Partie 102: Exigences générales – Appareillages de commande

IEC 62386-102:2022-11(en-fr)

CONTENTS

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.

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 third edition of IEC 62386-102 is intended to be used in conjunction with IEC 62386-101 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 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.

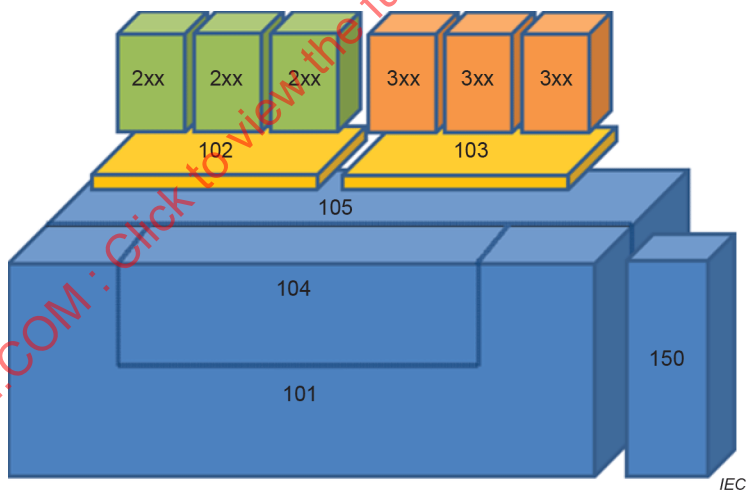


Figure 1 – IEC 62386 graphical overview

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

All numbers used in this document are decimal numbers unless otherwise noted. Hexadecimal numbers are given in the format 0xVV, where VV is the value. Binary numbers are given in the format 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 for control by digital signals of electronic lighting equipment.

2 Normative references

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

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

IEC 62386-103:2022, *Digital addressable lighting interface – Part 103: General requirements – Control devices*

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

3.4
broadcast unaddressed
type of address used to simultaneously address all control gear in the system that have no short address

3.5
DAPC
direct arc power control
method to directly control the light output

3.6
DTR
data transfer register
multipurpose register used to exchange data

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]).

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
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
NO
answer to a query where no backward frame is sent

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

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

3.13
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 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
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
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 to a query where a backward frame of MASK is sent

4 General

4.1 General

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

4.2 Version number

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

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

At each amendment to an edition of IEC 62386-102 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 IEC documents are generally subject to two amendments before a new edition is prepared.

5 Electrical specification

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

6 Bus power supply

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

7 Transmission protocol structure

7.1 General

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

7.2 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]	7...0	
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 62386-101:2022, Clause 8 apply.

9 Method of operation

9.1 General

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

9.2 Control gear

9.2.1 General

Control gear receive commands from application controllers. Application controllers are specified in IEC 62386-103:2022.

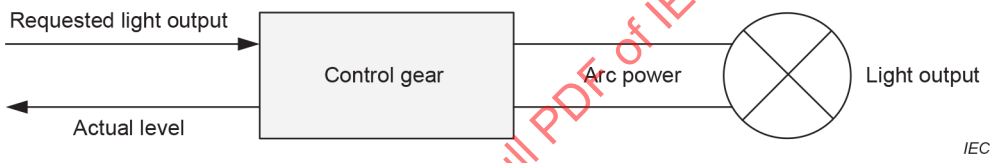


Figure 2 – Control gear directly operating a light source

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

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

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 ±½ 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.

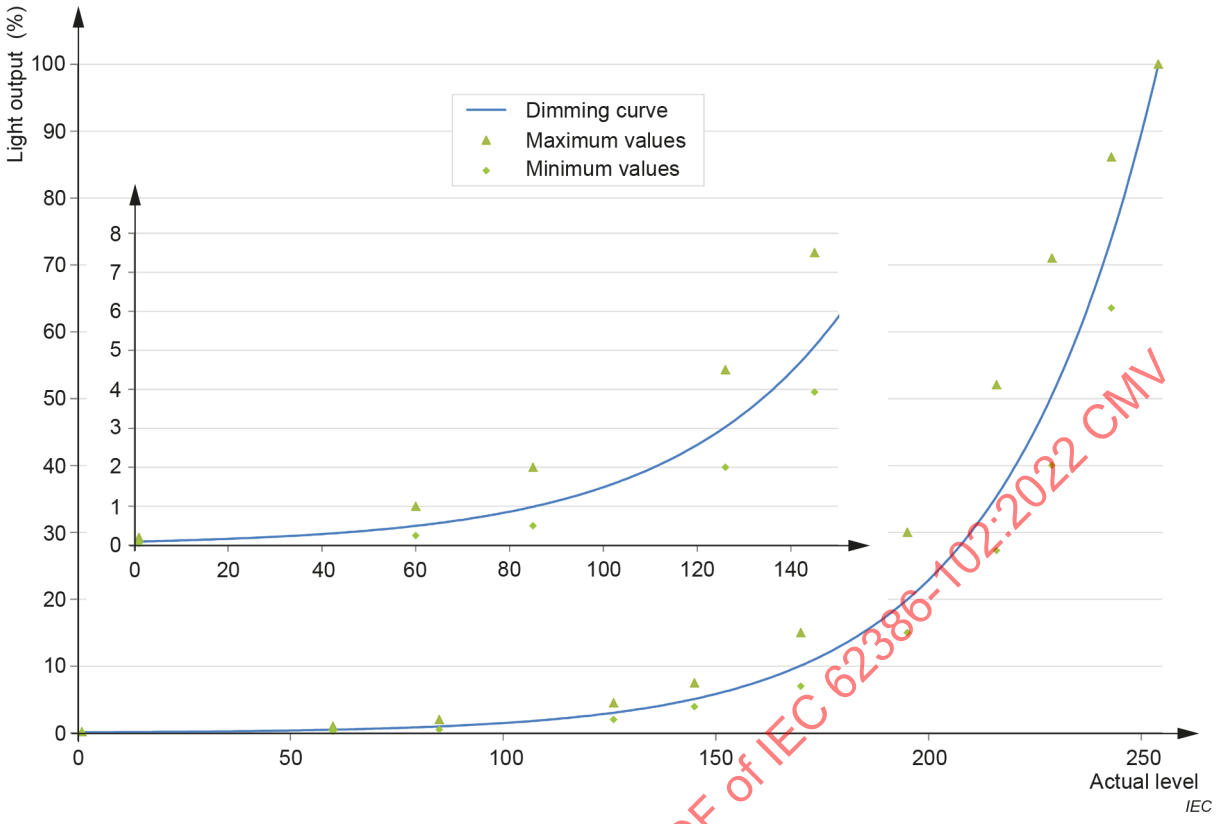


Figure 3 – Dimming curve

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.

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 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.

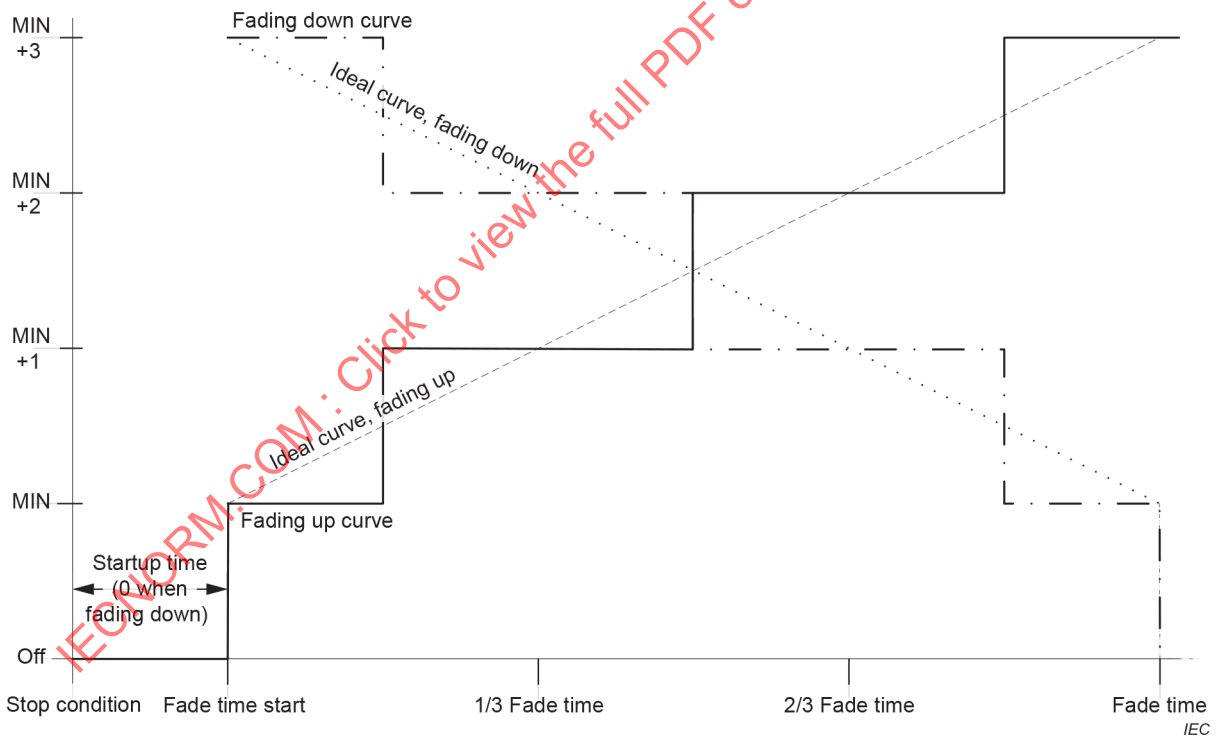


Figure 4 – Level over time, fading up and down

Testing shall be done with "*minLevel*" ≥ 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.

Table 4 lists the possible fade time values.

Table 4 – Fade times

"fadeTime"	*Minimum fade time* s	*Nominal fade time* s	*Maximum fade time* 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:

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

Table 5 lists the possible fade rate values.

Table 5 – Fade rates

"fadeRate"	*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

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.

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

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 0YYYAAAAb, where YYYb equals the fade time multiplier, and AAAAb the fade time base: The resulting fade time shall be monotically 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 YYYb.

The extended fade time can be queried using "QUERY EXTENDED FADE TIME". The answer shall be 0 YYY AAAAb, where YYYb 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.

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.

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 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 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)", "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 ≤ *"DTR0"* ≤ PHM: PHM;
- if *"DTR0"* ≥ *"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"* ≥ *"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 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 62386-101:2022, 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*" ± 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 · *"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.

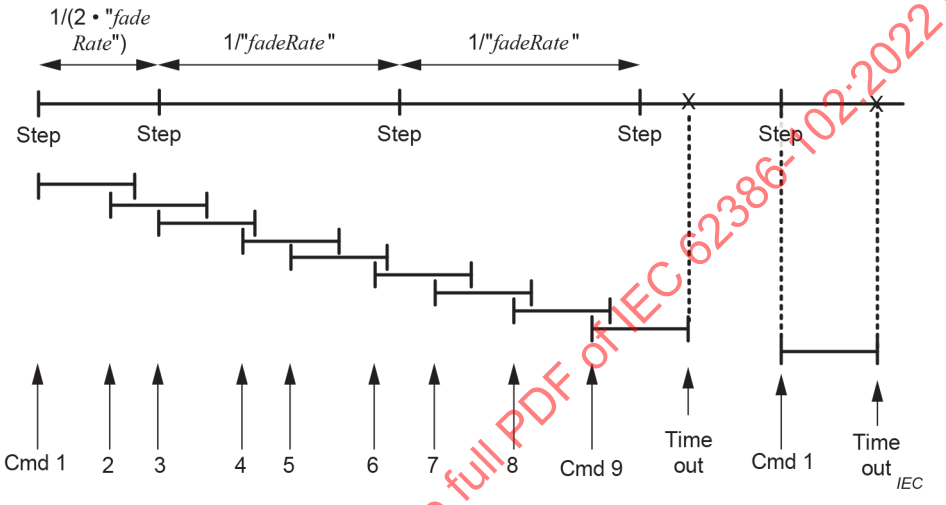


Figure 5 – Timing and response when executing command iteration

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 ms ± 20 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.

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.

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:

- 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 requirements of 7.2 shall apply.

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

9.10 Memory banks

9.10.1 General

Memory banks are freely accessible memory spaces containing information or configuration settings of the control gear. 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. 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

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

9.10.2 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 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

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

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

9.10.5.2 Reading multi-byte values

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

Reading from a multi-byte location shall reply with the stored value, and not values from the write-buffer that is used to buffer values before the complete value is written to memory (see 9.10.6.3).

9.10.5.3 Unimplemented locations

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

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

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

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

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

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

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

9.10.5.4 Temporarily unavailable locations

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

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

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

NOTE 1 The contents of buffers used to store temporary bytes for multi-byte writing can be lost when a write operation is started to any other multi-byte value in the same memory bank, or during a power-cycle, or as a result of execution of RESET MEMORY BANK. 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

Address	Description	Default value (factory)	Memory type
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]	factory burn-in[g]	ROM
[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.

Key

GTIN global trade item number

LSB least significant byte

MSB most significant byte

[a] It is recommended that the product GTIN is not re-used within the expected lifetime of the product after installation.

[b] Format of the version number is defined in IEC 62386-101:2022, 4.2.

[c] Format of the version number is defined in 4.2.

[d] Format of the version number is defined in IEC 62386-103:2022, 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 can contain both control gear and control devices. They share various numbers (e.g. GTIN, unique identification number). To avoid problems when reading, and getting different answers depending on the addressing scheme used, the memory bank layout is the same for control gear and for control devices up to and including location 0x19. The data shall be the same as well. The application controller can use either the commands specified in 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) in binary (see [1][1]). The bytes shall be stored most significant first and filled with leading zeroes.

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

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

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

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

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

The byte in location 0x16 shall contain the implemented IEC 62386-102 version number of the bus unit.

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.

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. 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.

---

1 Numbers in square brackets refer to the Bibliography.

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, 0xFE]	Use of these locations is specified in other parts of the IEC 62386-2xx series.	[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 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, 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 can respond to commands or not. However, until the reset operation is complete, none of the affected variables needs to have a defined value.

An application controller can trigger the reset operation using the "RESET" instruction and should wait at least 350 ms to ensure all control 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 can respond to 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 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 62386-101:2022, 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 62386-101:2022, 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.

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

Power on system response	Minimum time	Maximum time
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 62386-101:2022.

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 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"* = 1xxxxxxxb 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 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.

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 ± 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.

EXAMPLE A control gear 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.

EXAMPLE Mains under voltage, over temperature, unexpected watchdog timers firing.

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. 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.

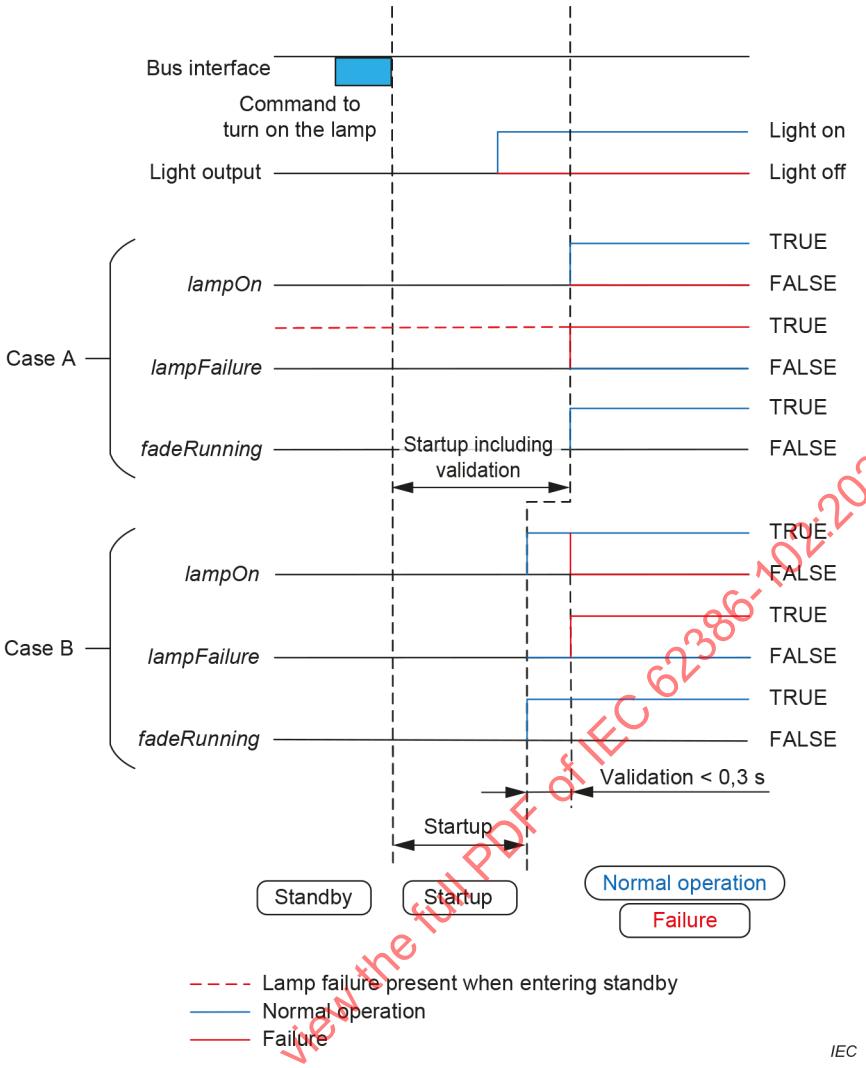


Figure 6 – Correlation between "*lampFailure*", "*lampOn*" and "*fadeRunning*" bits

The startup phase shall include the validation that the lamp is stable and emitting light before continuing with normal operation or failure. This behaviour is illustrated in Figure 6, Case A and applies to the following situations:

- "*lampFailure*" is TRUE before entering standby, or
- the lamp validation takes longer than 0,3 s.

The startup phase may exclude the validation of the lamp in the following situation, illustrated in Figure 6, Case B:

- "*lampFailure*" is FALSE before entering standby and,
- the lamp validation takes less than 0,3 s.

In this situation the validation should be part of the normal operation and shall be executed immediately after startup. This implies that "*lampOn*" can be incorrect for a maximum of 0,3 s until the validation is finished.

NOTE Figure 6 also illustrates the effect of "*lampFailure*" on "*lampOn*" and "*fadeRunning*" bits for the scenarios described above.

For the light source type "unknown light source type", there shall be support for this bit. If there is no support for the lamp failure bit, this shall be made explicit in the corresponding part of the IEC 62386-2xx series.

For the light source type "no light source", support for this bit is optional. Testing is excluded for this light source type. If there is support for the lamp failure bit, this shall be made explicit in the corresponding part of the IEC 62386-2xx series.

9.16.4 Bit 2: lamp on

"lampOn" shall be set to FALSE when the lamp is off, during startup, and in case of total lamp failure. In all other cases it shall be set to TRUE.

9.16.5 Bit 3: limit error

Except for activation of "*powerOnLevel*" (see 9.13), if the last requested target level has been modified in accordance with *"minLevel"* or *"maxLevel"* limitations, or *"targetLevel"* has been modified due to a change of *"minLevel"* or *"maxLevel"*, "*limitError*" shall be set to TRUE.

If the last target level requested by "DAPC (*level*)" equals "MASK", "*limitError*" shall not change.

In all other cases "*limitError*" shall be set to FALSE.

9.16.6 Bit 4: fade running

"fadeRunning" shall be set to FALSE except for the time during which the fade timer is running. *"fadeRunning"* shall be set to TRUE from the beginning of the fade (after startup) until the end of the fade time, regardless of whether *"targetLevel"* and *"actualLevel"* reach the same level.

9.16.7 Bit 5: reset state

"resetState" shall be set to TRUE if all the NVM variables mentioned in Table 16, except "*lastLightLevel*", are at their reset value. The NVM variables that are marked with 'no change' in the reset value column shall not be considered. NVM variables defined in implemented parts of the IEC 62386-2xx series shall be included.

In all other cases the bit shall be set to FALSE.

9.16.8 Bit 6: missing short address

This bit indicates whether a short address has been assigned to the gear, by checking *"shortAddress"*. The bit shall be TRUE if *"shortAddress"* = MASK.

In all other cases the bit shall be set to FALSE.

9.16.9 Bit 7: power cycle seen

"powerCycleSeen" shall be set to TRUE after an external power cycle (see IEC 62386-101:2022, 4.11) has occurred.

"powerCycleSeen" shall be set to FALSE once one of the following commands has been executed:

"RESET", "DAPC (*level*)", "OFF", "UP", "DOWN", "STEP UP", "STEP DOWN", "RECALL MAX LEVEL", "RECALL MIN LEVEL", "GO TO LAST ACTIVE LEVEL", "STEP DOWN AND OFF", "ON AND STEP UP", "CONTINUOUS UP", "CONTINUOUS DOWN", "GO TO SCENE (*sceneNumber*)".

9.17 Non-volatile memory

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

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

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

9.18 Device types and features

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

The device type or feature specific command set can be selected by the instruction "ENABLE DEVICE TYPE (*data*)".

This instruction shall select the device type or feature for which only the next application extended command (refer to 11.6) is valid. Executing this instruction shall cancel any previous selection of a device type.

The enabling of the device type or feature shall be cancelled upon execution of the next command, and that command shall be executed according to its specification, regardless of whether it is an application extended command or not.

A control gear shall not react to commands which belong to the application extended commands if *data* equals MASK, 254 or represents a device type or feature not supported by this control gear.

The device types and features shall be coded as specified in the relevant parts of the IEC 62386-2xx series.

An application controller can check which device types and features are supported by the control gear. "QUERY DEVICE TYPE" reports the supported device types and features. If more than one device type or feature is supported, "QUERY DEVICE TYPE" reports MASK. In that case, the application controller can check all supported device types and features by repeating "QUERY NEXT DEVICE TYPE" until 254 is received as an answer. Issuing "QUERY DEVICE TYPE" automatically ensures that the first supported device type or feature will be reported by "QUERY NEXT DEVICE TYPE".

To check the version number of the supported device types and features, the application controller can send "ENABLE DEVICE TYPE (*data*) followed by "QUERY EXTENDED VERSION NUMBER". This will report the version number of that specific device type or feature implementation.

Application controllers should be able to identify individual control gear and store the relationship between the control gear's individual address, and its device types and features, in persistent memory.

9.19 Using scenes

A control gear shall support the use of 16 scenes. The following commands shall be supported:

"GO TO SCENE (*sceneNumber*)", "REMOVE FROM SCENE (*sceneX*)", "QUERY SCENE LEVEL (*sceneX*)" and "SET SCENE (*DTR0, sceneX*)".

These commands actually comprise 16 commands each, one for each scene. This is accomplished by selecting a block of 16 consecutive opcodes. The number of the scene to be used can thus easily be calculated.

Upon execution of one of the scene commands, *sceneNumber* shall be derived from the opcode: *sceneNumber* = opcode – opcodeBase. This identifies the scene to be used. The opcodeBase can be found in Table 14.

Table 14 – Scenes

Command	opcodeBase	Opcode range
GO TO SCENE (*sceneNumber)*	0x10	[0x10,0x1F]
REMOVE FROM SCENE (*sceneX*)	0x50	[0x50,0x5F]
QUERY SCENE LEVEL (*sceneX*)	0xB0	[0xB0,0xBF]
SET SCENE (*DTR0, sceneX*)	0x40	[0x40,0x4F]

The *"sceneX"* variable also stands for 16 individual variables, where *X* equals *sceneNumber* in the range of [0,15].

On accepting command "GO TO SCENE (*sceneNumber*)" the reaction of the control gear shall depend upon the current value of "*sceneX*", where *X* is derived from *sceneNumber*. If "*sceneX*" equals MASK, "*targetLevel*" shall not be affected. Otherwise, the control gear shall behave exactly as if "DAPC (*level*)" had been accepted with level equal to "*sceneX*".

NOTE Using "DAPC (*level*)" implies the transition is made using the set fade time.

9.20 Current bus unit configuration

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

- the number of logical units,
- the device types implemented for each logical unit, and
- particular types, defined for a particular device type.

For the given value of current bus unit configuration, the implementation shall be as shown in Table 15.

Table 15 – Current bus unit configuration

Current bus unit configuration	Number of logical units[a]	Minimum device types present[b]	Description
0	1	1 of [6]	1 logical unit of LED (device type 6)
1	2	2 of [6]	2 logical units of LED (device type 6)
2	3	3 of [6]	3 logical units of LED (device type 6)
3	4	4 of [6]	4 logical units of LED (device type 6)
4	1	1 of [8(Tc)]	1 logical unit of colour type Tc (device type 8)
5	2	2 of [8(Tc)]	2 logical units of colour type Tc (device type 8)
6	1	1 of [8(RGBWAF)]	1 logical unit of colour type RGBWAF (device type 8)
7	1	1 of [8(Tc+RGBWAF+xy)]	1 logical unit with colour types Tc, RGBWAF and xy (device type 8)
8	2	1 of [6], 1 of [8(Tc)]	1 logical unit of LED (device type 6) and 1 logical unit of colour type Tc (device type 8)
9	1	1 of [1(type A)]	1 logical unit of self-contained emergency type A (maintained dimming controllable, device type 1)
10	1	1 of [1(type B)]	1 logical unit of self-contained emergency type B (maintained on/off controllable, device type 1)
11	1	1 of [1(type C)]	1 logical unit of self-contained emergency type C (maintained non-controllable, device type 1)
12	1	1 of [1(type D)]	1 logical unit of self-contained emergency type D (non-maintained non-controllable, device type 1)
13-191	Reserved		
192-255	Manufacturer-specific		

[a] Total number of logical units in the bus unit of the device type given in brackets.

[b] At least the listed device types shall be implemented in the given configuration.

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

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

10 Declaration of variables

The default values, the reset values, power on values, the range of validity and the type of memory of the defined variables shall be as given in Table 16.

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

Table 16 – Declaration of variables

Variable	Default value (factory)	Reset value	Power on value	Range of validity	Memory type
"actualLevel"	[a]	0xFE	0x00	0, [*"minLevel"*, *"maxLevel"*]	RAM
"targetLevel"	[a]	0xFE	See 9.13 Power on	0, [*"minLevel"*, *"maxLevel"*]	RAM
"lastActiveLevel"	[a]	0xFE	*"maxLevel"*	[*"minLevel"*, *"maxLevel"*]	RAM
"lastLightLevel"	0xFE	0xFE[c]	no change	0, [*"minLevel"*, *"maxLevel"*]	NVM
"powerOnLevel"	0xFE	0xFE	no change	[0,0xFF]	NVM
"systemFailureLevel"	0xFE	0xFE	no change	[0,0xFF]	NVM
"minLevel"	PHM	PHM	no change	[PHM,*"maxLevel"*]	NVM
"maxLevel"	0xFE	0xFE	no change	[*"minLevel"*,0xFE]	NVM
"fadeRate"	7	7	no change	[1,0xF]	NVM
"fadeTime"	0	0	no change	[0,0xF]	NVM
"extendedFadeTimeBase"	0	0	no change	[0,1111b]	NVM
"extendedFadeTimeMultiplier"	0	0	no change	[0,100b]	NVM
"shortAddress"	MASK (no address)	no change	no change	[0,63], MASK	NVM
"searchAddress"	[a]	0xFF FF FF	0xFF FF FF	[0,0xFF FF FF]	RAM
"randomAddress"	0xFF FF FF	0xFF FF FF	no change	[0,0xFF FF FF]	NVM
"operatingMode"	factory burn-in	no change	no change	0,[0x80,0xFF]	NVM
"initialisationState"	[a]	no change	DISABLED	[ENABLED, DISABLED, WITHDRAWN]	RAM
"writeEnableState"	[a]	DISABLED	DISABLED	[ENABLED, DISABLED]	RAM
"controlGearFailure"	[a]	[b]	FALSE[d]	[TRUE, FALSE]	RAM
"lampFailure"	[a]	[b]	FALSE[d]	[TRUE, FALSE]	RAM
"lampOn"	[a]	[b]	FALSE	[TRUE, FALSE]	RAM
"limitError"	[a]	FALSE	FALSE[d]	[TRUE, FALSE]	RAM
"fadeRunning"	[a]	FALSE	FALSE	[TRUE, FALSE]	RAM
"resetState"	TRUE	TRUE	TRUE[d]	[TRUE, FALSE]	RAM
"powerCycleSeen"	[a]	FALSE	TRUE	[TRUE, FALSE]	RAM
"gearGroups"	0x00 00 (no group)	0x00 00 (no group)	no change	[0,0xFF FF]	NVM
"sceneX" [e]	MASK	MASK	no change	[0,0xFF]	NVM
"DTR0"	[a]	no change	0x00	[0,0xFF]	RAM
"DTR1"	[a]	no change	0x00	[0,0xFF]	RAM
"DTR2"	[a]	no change	0x00	[0,0xFF]	RAM
PHM	factory burn-in	no change	no change	[1,0xFE]	ROM
"versionNumber"	3.0	no change	no change	00001100b	ROM

[a] Not applicable.

[b] The value could change as a consequence of the RESET command execution.

[c] This NVM variable is excluded for "*resetState*".

[d] The value should reflect the actual situation as soon as possible.

[e] *X* is in the range 0x0 to 0xF, effectively there is one variable for each of the 16 scenes.

11 Definition of commands

11.1 General

Unused opcodes are reserved for future needs.

11.2 Overview sheets

Table 17 gives an overview of the standard commands. The special commands overview can be found in Table 18.

Table 17 – Standard commands

Command name	Address byte		Opcode byte	Ed. 1 command number	DTR0	DTR1	DTR2	Answer	Send twice	References	Command reference
	See 7.2.2	Select or bit									
DAPC (*level*)	*Device*	0	*level*	-						9.4, 9.7.3, 9.8	11.3.1
OFF	*Device*	1	0x00	0						9.7.2	11.3.2
UP	*Device*	1	0x01	1						9.7.3	11.3.3
DOWN	*Device*	1	0x02	2						9.7.3	11.3.4
STEP UP	*Device*	1	0x03	3						9.7.2	11.3.5
STEP DOWN	*Device*	1	0x04	4						9.7.2	11.3.6
RECALL MAX LEVEL	*Device*	1	0x05	5						9.7.2, 9.14.3	11.3.7
RECALL MIN LEVEL	*Device*	1	0x06	6						9.7.2, 9.14.3	11.3.8
STEP DOWN AND OFF	*Device*	1	0x07	7						9.7.2	11.3.9
ON AND STEP UP	*Device*	1	0x08	8						9.7.2	11.3.10
ENABLE DAPC SEQUENCE	*Device*	1	0x09	9						9.8	11.3.11
GO TO LAST ACTIVE LEVEL	*Device*	1	0x0A							9.7.3	11.3.12
CONTINUOUS UP	*Device*	1	0x0B							9.7.3	11.3.13

Command name	Address byte		Opcode byte	Ed. 1 command number	DTR0	DTR1	DTR2	Answer	Send twice	References	Command reference
	See 7.2.2	Select or bit									
CONTINUOUS DOWN	*Device*	1	0x0C							9.7.3	11.3.14
GO TO SCENE (*sceneNumber*)[a]	*Device*	1	0x10 + *sceneNumber*	16 to 31						9.7.3, 9.19	11.3.15
RESET	*Device*	1	0x20	32					✓	9.11.1, 10	11.4.2
STORE ACTUAL LEVEL IN DTR0	*Device*	1	0x21	33	✓				✓		11.4.3
Reserved[b]	*Device*	1	0x22 [b]						✓		
SET OPERATING MODE (*DTR0*)	*Device*	1	0x23		✓				✓	9.9	11.4.4
RESET MEMORY BANK (*DTR0*)	*Device*	1	0x24		✓				✓	9.11.2	11.4.5
IDENTIFY DEVICE	*Device*	1	0x25						✓	9.14.3	11.4.6
SET MAX LEVEL (*DTR0*)	*Device*	1	0x2A	42	✓				✓	9.6	11.4.7
SET MIN LEVEL (*DTR0*)	*Device*	1	0x2B	43	✓				✓	9.6	11.4.8
SET SYSTEM FAILURE LEVEL (*DTR0*)	*Device*	1	0x2C	44	✓				✓	9.12	11.4.9
SET POWER ON LEVEL (*DTR0*)	*Device*	1	0x2D	45	✓				✓	9.13	11.4.10
SET FADE TIME (*DTR0*)	*Device*	1	0x2E	46	✓				✓	9.5.2	11.4.11
SET FADE RATE (*DTR0*)	*Device*	1	0x2F	47	✓				✓	9.5.3	11.4.12
SET EXTENDED FADE TIME (*DTR0*)	*Device*	1	0x30		✓				✓	9.5.4	11.4.13
SET SCENE (*DTR0, sceneX*)[a]	*Device*	1	0x40 + *sceneNumber*	64 to 79	✓				✓	9.19	11.4.14
REMOVE FROM SCENE (*sceneX*)[a]	*Device*	1	0x50 + *sceneNumber*	80 to 95					✓	9.19	11.4.15
ADD TO GROUP (*group*)[a]	*Device*	1	0x60 + *group*	96 to 111					✓		11.4.16
REMOVE FROM GROUP (*group*)[a]	*Device*	1	0x70 + *group*	112 to 127					✓		11.4.17
SET SHORT ADDRESS (*DTR0*)	*Device*	1	0x80	128	✓				✓	9.14.4	11.4.18
ENABLE WRITE MEMORY	*Device*	1	0x81	129					✓	9.10.6	11.4.19

Command name	Address byte		Opcode byte	Ed. 1 command number	DTR0	DTR1	DTR2	Answer	Send twice	References	Command reference
	See 7.2.2	Select or bit									
Reserved for IEC 62386-104 (see [3])	*Device*	1	0x82		✓						
QUERY STATUS	*Device*	1	0x90	144				✓		9.16	11.5.2
QUERY CONTROL GEAR PRESENT	*Device*	1	0x91	145				✓			11.5.3
QUERY LAMP FAILURE	*Device*	1	0x92	146				✓			11.5.4
QUERY LAMP POWER ON	*Device*	1	0x93	147				✓			11.5.6
QUERY LIMIT ERROR	*Device*	1	0x94	148				✓			11.5.7
QUERY RESET STATE	*Device*	1	0x95	149				✓			11.5.8
QUERY MISSING SHORT ADDRESS	*Device*	1	0x96	150				✓		9.14.2	11.5.9
QUERY VERSION NUMBER	*Device*	1	0x97	151				✓			11.5.10
QUERY CONTENT DTR0	*Device*	1	0x98	152	✓			✓			11.5.11
QUERY DEVICE TYPE	*Device*	1	0x99	153				✓		9.18	11.5.12
QUERY PHYSICAL MINIMUM	*Device*	1	0x9A	154				✓			11.5.13
QUERY POWER FAILURE	*Device*	1	0x9B	155				✓			11.5.15
QUERY CONTENT DTR1	*Device*	1	0x9C	156		✓		✓			11.5.16
QUERY CONTENT DTR2	*Device*	1	0x9D	157			✓	✓			11.5.17
QUERY OPERATING MODE	*Device*	1	0x9E					✓		9.9	11.5.18
QUERY LIGHT SOURCE TYPE	*Device*	1	0x9F		✓	✓	✓	✓			11.5.19
QUERY ACTUAL LEVEL	*Device*	1	0xA0	160				✓			11.5.20
QUERY MAX LEVEL	*Device*	1	0xA1	161				✓			11.5.21
QUERY MIN LEVEL	*Device*	1	0xA2	162				✓			11.5.22
QUERY POWER ON LEVEL	*Device*	1	0xA3	163				✓		9.13	11.5.23
QUERY SYSTEM FAILURE LEVEL	*Device*	1	0xA4	164				✓		9.12	11.5.24

Command name	Address byte		Opcode byte	Ed. 1 command number	DTR0	DTR1	DTR2	Answer	Send twice	References	Command reference
	See 7.2.2	Select or bit									
QUERY FADE TIME/FADE RATE	*Device*	1	0xA5	165				✓			11.5.25
QUERY MANUFACTURER SPECIFIC MODE	*Device*	1	0xA6					✓		9.9	11.5.27
QUERY NEXT DEVICE TYPE	Device	1	0xA7					✓		9.18	11.5.13
QUERY EXTENDED FADE TIME	Device	1	0xA8					✓		9.5.4	11.5.26
QUERY CONTROL GEAR FAILURE	Device	1	0xAA					✓		9.16.2	11.5.4
Reserved for IEC 62386-104 (see [3])	Device	1	0xAB					✓			
QUERY SCENE LEVEL (*sceneX*)[a]	*Device*	1	0xB0 + *sceneNumber*	176 to 191				✓		9.19	11.5.28
QUERY GROUPS 0-7	*Device*	1	0xC0	192				✓			11.5.29
QUERY GROUPS 8-15	*Device*	1	0xC1	193				✓			11.5.30
QUERY RANDOM ADDRESS (H)	*Device*	1	0xC2	194				✓			11.5.31
QUERY RANDOM ADDRESS (M)	*Device*	1	0xC3	195				✓			11.5.32
QUERY RANDOM ADDRESS (L)	*Device*	1	0xC4	196				✓			11.5.33
READ MEMORY LOCATION (*DTR1, DTR0*)	*Device*	1	0xC5	197	✓	✓		✓		9.10.5	11.5.34
Application extended commands	*Device*	1	0xE0 to 0xFE	224 to 254	?	?	?	?	?	9.18	11.6
QUERY EXTENDED VERSION NUMBER	*Device*	1	0xFF	255				✓			11.6.2

[a] There is one command per scene, so there are actually 16 commands for scenes 0 to 15. Similarly for the 16 group commands.

[b] Reserved to maintain backward compatibility due to use in Edition 2 of IEC 62386-102:2014 (see [2]).

Table 18 – Special commands

Command name	Address byte	Opcode byte	Ed.1 command number	DTR0	DTR1	DTR2	Answer	Send twice	References	Command reference
TERMINATE	0xA1	0x00	256						9.14.2	11.7.1
DTR0 (*data*)	0xA3	*data*	257	✓						11.7.3
INITIALISE (*device*)	0xA5	*device*	258					✓	9.14.2	11.7.4
RANDOMISE	0xA7	0x00	259					✓	9.14.2	11.7.5
COMPARE	0xA9	0x00	260				✓		9.14.2	11.7.6
WITHDRAW	0xAB	0x00	261						9.14.2	11.7.7
PING	0xAD	0x00								11.7.19
SEARCHADDRH (*data*)	0xB1	*data*	264						9.14.2	11.7.8
SEARCHADDRM (*data*)	0xB3	*data*	265						9.14.2	11.7.9
SEARCHADDRL (*data*)	0xB5	*data*	266						9.14.2	11.7.10
PROGRAM SHORT ADDRESS (*data*)	0xB7	*data*	267						9.14.2	11.7.11
VERIFY SHORT ADDRESS (*data*)	0xB9	*data*	268				✓		9.14.2	11.7.12
QUERY SHORT ADDRESS	0xBB	0x00	269				✓		9.14.2	11.7.13
Reserved for IEC 62386-104 (see [3])	0xBB	0x01		✓	✓		✓			
Reserved for IEC 62386-104 (see [3])	0xBD	*data*								
Reserved for IEC 62386-104 (see [3])	0xBF	*data*								
ENABLE DEVICE TYPE (*data*)	0xC1	*data*	272						9.14.2	11.7.14
DTR1 (*data*)	0xC3	*data*	273		✓					11.7.15
DTR2 (*data*)	0xC5	*data*	274			✓				11.7.16
WRITE MEMORY LOCATION (*DTR1, DTR0, data*)	0xC7	*data*	275	✓	✓		✓		9.10.6	11.7.17
WRITE MEMORY LOCATION – NO REPLY (*DTR1, DTR0, data*)	0xC9	*data*		✓	✓				9.10.6	11.7.18

11.3 Level instructions

11.3.1 DAPC (*level*)

Upon execution of "DAPC (*level*)" (direct arc power control), *"targetLevel"* shall be calculated on the basis of *"level"*.

The transition from *"actualLevel"* to *"targetLevel"* shall start using the applicable fade time.

Refer to 9.4, 9.7.3 and 9.13 for further information.

11.3.2 OFF

"targetLevel" shall be set to 0x00 and the lamp(s) shall switch off.

The transition from *"actualLevel"* to *"targetLevel"* shall be immediate and the light output shall be adjusted as quickly as possible.

Refer to 9.7.2 for further information.

11.3.3 UP

Dim up using a 200 ms fade with the set fade rate. *"targetLevel"* shall be calculated on the basis of *"actualLevel"* and the set fade rate.

To ensure that there is a reaction to the command, at least one step (final "*targetLevel*" = calculated "*targetLevel*" + 1) shall be made upon execution of the first command of an iteration. After that first step, the next steps shall be executed using the specified fade rate while the fading is running. Every "UP" instruction executed as a part of an iteration shall cause the 200 ms fade to be restarted and *"targetLevel"* to be recalculated on the basis of *"actualLevel"* and the set fade rate.

There shall be no change to *"actualLevel"* if *"actualLevel"* is at *"maxLevel"* or 0x00.

Refer to 9.5.1, 9.7.3 and 9.8.2 for further information.

11.3.4 DOWN

Dim down using a 200 ms fade with the set fade rate. *"targetLevel"* shall be calculated on the basis of *"actualLevel"* and the set fade rate.

To ensure that there is a reaction to the command, at least one step (final "*targetLevel*" = calculated "*targetLevel*" − 1) shall be made upon execution of the first command of an iteration. After that first step, the next steps shall be executed using the specified fade rate while the fading is running. Every "DOWN" instruction executed as a part of an iteration shall cause the 200 ms fade to be restarted and *"targetLevel"* to be recalculated on the basis of *"actualLevel"* and the set fade rate.

There shall be no change to *"actualLevel"* if *"actualLevel"* is at *"minLevel"* or 0x00.

Refer to 9.5.1, 9.7.3 and 9.8.2 for further information.

11.3.5 STEP UP

"targetLevel" shall be set to:

- if *"targetLevel"* = 0: 0x00;
- if *"minLevel"* ≤ *"targetLevel"* < *"maxLevel": "targetLevel"* + 1;
- if *"targetLevel"* = *"maxLevel": "maxLevel".*

The transition from *"actualLevel"* to *"targetLevel"* shall be immediate and the light output shall be adjusted as quickly as possible.

Refer to 9.4 and 9.5.9 for further information.

11.3.6 STEP DOWN

"targetLevel" shall be set to:

- if *"targetLevel"* = 0: 0x00;
- if *"minLevel"* < *"targetLevel"* ≤ *"maxLevel": "targetLevel"* − 1;
- if *"targetLevel" = "minLevel": "minLevel".*

The transition from *"actualLevel"* to *"targetLevel"* shall be immediate and the light output shall be adjusted as quickly as possible.

Refer to 9.4 and 9.5.9 for further information.

11.3.7 RECALL MAX LEVEL

When the *"initialisationState"* is DISABLED, *"targetLevel"* and *"actualLevel"* shall be set to *"maxLevel"* immediately and the light output shall be adjusted as quickly as possible.

Refer to 9.7.2 for further information.

When the *"initialisationState"* is not DISABLED, the control gear shall set "*actualLevel*" and "*targetLevel*" to "*maxLevel*", and then adjust the light output as quickly as possible to 100 % temporarily ignoring "*maxLevel*" and "*actualLevel*".

If the device is unable to visually identify itself in this way, 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 RECALL MAX LEVEL and RECALL MIN LEVEL commands being executed in an alternating sequence.

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

Identification shall be stopped immediately when the *"initialisationState"* changes to DISABLED and upon execution of any instruction other than INITIALISE (*device*), RECALL MIN LEVEL, RECALL MAX LEVEL or IDENTIFY DEVICE.

When the *"initialisationState"* changes to DISABLED, the identification shall stop immediately.

Refer to 9.14.3 for further information.

11.3.8 RECALL MIN LEVEL

When the *"initialisationState"* is DISABLED, *"targetLevel"* and *"actualLevel"* shall be set to *"minLevel"* immediately and the light output shall be adjusted as quickly as possible.

Refer to 9.7.2 for further information.

When *"initialisationState"* is not DISABLED, the control gear shall set "*actualLevel*" and "*targetLevel*" to "*minLevel*" and then adjust the light output as quickly as possible to its PHM level temporarily ignoring "*minLevel*" and "*actualLevel*". If, however, the PHM is not visibly significantly different from 100 %, then the lamp shall be temporarily switched off instead.

If the device is unable to visually identify itself in this way, 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 RECALL MAX LEVEL and RECALL MIN LEVEL commands being executed in an alternating sequence.

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

Identification shall be stopped immediately when the *"initialisationState"* changes to DISABLED and upon execution of any instruction other than INITIALISE (*device*), RECALL MIN LEVEL, RECALL MAX LEVEL or IDENTIFY DEVICE.

When the *"initialisationState"* changes to DISABLED, identification shall stop immediately.

Refer to 9.14.3 for further information.

11.3.9 STEP DOWN AND OFF

"targetLevel" shall be set to:

- if *"targetLevel"* = 0: 0x00;
- if *"minLevel"* < *"targetLevel"* ≤ *"maxLevel"*: *"targetLevel"* − 1;
- if *"targetLevel"* = *"minLevel"*: 0x00.

The transition from *"actualLevel"* to *"targetLevel"* shall be immediate and the light output shall be adjusted as quickly as possible.

Refer to 9.4 and 9.5.9 for further information.

11.3.10 ON AND STEP UP

"targetLevel" shall be set to:

- if *"targetLevel"* = 0: *"minLevel";*
- if *"minLevel"* ≤ *"targetLevel"* < *"maxLevel"*: *"targetLevel"* + 1;
- if *"targetLevel"* ≥ *"maxLevel"*: *"maxLevel"*.

The transition from *"actualLevel"* to *"targetLevel"* shall be immediate and the light output shall be adjusted as quickly as possible.

Refer to 9.4 and 9.5.9 for further information.

11.3.11 ENABLE DAPC SEQUENCE

Indicates the start of a command iteration of "DAPC (*level*)" commands.

Refer to 9.8.3 for further information.

11.3.12 GO TO LAST ACTIVE LEVEL

Upon execution of this command *"targetLevel"* shall be calculated based on *"lastActiveLevel"*.

The transition from *"actualLevel"* to *"targetLevel"* shall start using the set fade time.

Refer to 9.7.3 and 9.4 for further information.

11.3.13 CONTINUOUS UP

Dim up using the set fade rate. "*targetLevel*" shall be set to "*maxLevel*" and a fade shall be started using the set fade rate. The fade shall stop when "*maxLevel*" is reached.

There shall be no change to "*actualLevel*" if "*actualLevel*" is at "*maxLevel*" or 0x00.

Refer to 9.7.3 and 9.5.6.2 for further information.

11.3.14 CONTINUOUS DOWN

Dim down using the set fade rate. "*targetLevel*" shall be set to "*minLevel*" and a fade shall be started using the set fade rate. The fade shall stop when "*minLevel*" is reached.

There shall be no change to "*actualLevel*" if "*actualLevel*" is at "*minLevel*" or 0x00.

Refer to 9.7.3 and 9.5.6.2 for further information.

11.3.15 GO TO SCENE (*sceneNumber*)

The control gear shall react depending on the actual value of *"sceneX"* where *X* is derived from *sceneNumber*:

- if *"sceneX"* = *MASK*: the command shall not affect *"targetLevel";*
- in all other cases: internally "DAPC (*level*)", with *level* equal to *"sceneX"* shall be executed.

NOTE Using "DAPC (*level*)" implies the transition is made using the set fade time.

Refer to 9.19 and 11.3.1 for further information.

11.4 Configuration instructions

11.4.1 General

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

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

- The instruction shall be ignored if so required by the provisions of 9.7.
- The control gear shall not reply to the instruction.

11.4.2 RESET

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

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

Refer to 9.11.1 and Table 16 for further information.

11.4.3 STORE ACTUAL LEVEL IN DTR0

The *"actualLevel"* shall be stored in *"DTR0"*.

11.4.4 SET OPERATING MODE (*DTR0*)

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

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

Refer to 9.9 for further information.

11.4.5 RESET MEMORY BANK (*DTR0*)

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

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

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

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

Refer to 9.11.2 for further information.

11.4.6 IDENTIFY DEVICE

The control gear shall start or restart a 10 s ± 1 s timer. While the timer is running, a procedure shall run which enables an observer to distinguish any control gear running this process from any devices (of the same type) which are not running it. If the timer expires, identification shall stop.

During identification no variables shall be affected except when 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 can be at any level between off and 100 %, MIN, MAX and "*actualLevel*" being in effect temporarily ignored.

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

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.

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).

Identification can be used during commissioning in that it allows the installer to, for example, allocate the particular identified device to a particular device group.

The indication can be done, for example by flashing a LED, by producing a sound or other visual or audible means. The exact process used to identify is manufacturer specific and should be described in the manual.

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

Refer to 9.14.3 for further information.

11.4.7 SET MAX LEVEL (*DTR0*)

"maxLevel" shall be set to:

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

If, as a result of setting a new max level *"actualLevel"* > *"maxLevel"*, *"targetLevel"* shall be calculated on the basis of *"maxLevel"*. The transition from *"actualLevel"* to *"targetLevel"* shall start immediately and the light output shall be adjusted as quickly as possible.

Refer to 9.7.2 for further information.

11.4.8 SET MIN LEVEL (*DTR0*)

"minLevel" shall be set to:

- if 0 ≤ *"DTR0"* ≤ PHM: PHM;
- if *"DTR0"* ≥ *"maxLevel"* or MASK: *"maxLevel"*;
- in all other cases: *"DTR0".*

If *"actualLevel"* > 0 and as a result of setting a new min level *"actualLevel"* < *"minLevel"*, *"targetLevel"* shall be calculated on the basis of *"minLevel"*. The transition from *"actualLevel"* to *"targetLevel"* shall be immediate and the light output shall be adjusted as quickly as possible. Refer to 9.7.2 for further information.

11.4.9 SET SYSTEM FAILURE LEVEL (*DTR0*)

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

Refer to 9.12 for further information.

11.4.10 SET POWER ON LEVEL (*DTR0*)

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

Refer to 9.13 for further information.

11.4.11 SET FADE TIME (*DTR0*)

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

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

If "*fadeTime"* is not equal to 0, the fade time shall be calculated on the basis of *"fadeTime"*. If "*fadeTime"* is equal to 0, the extended fade time shall be used.

If a new fade time is stored during a running fade process, this process shall be finished first before the new value is used in the following fade.

Refer to 9.5, 9.7.3, 11.4.13 and 11.7.19 for further information.

11.4.12 SET FADE RATE (*DTR0*)

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"*. If a new fade rate is stored during a running fade process, this process shall be finished first before the new value is used in the following fade.

Refer to 9.5 and 9.7.3 for further information.

11.4.13 SET EXTENDED FADE TIME (*DTR0*)

The *"extendedFadeTimeBase"* and "*extendedFadeTimeMultiplier*" shall be set to a value according to the following steps:

- If "*DTR0*" > 0x4F (0100 1111):
  - "*extendedFadeTimeBase*" shall be set to 0;
  - "*extendedFadeTimeMultiplier*" shall be set to 0;

effectively selecting a fade as quickly as possible.

- For all other cases:
  - "*extendedFadeTimeBase*" shall be set to AAAAb where *"DTR0"* = xxxxAAAAb;
  - "*extendedFadeTimeMultiplier*" shall be set to YYYb where *"DTR0"* = xYYYxxxxb.
- The fade time shall be calculated by multiplying the base value and the multiplier.

If a new fade time is stored during a running fade process, this process shall be finished first before the new value is used in the following fade.

Refer to 9.5.4 for further information.

11.4.14 SET SCENE (*DTR0, sceneX*)

This command actually comprises 16 commands, one for each scene. This is accomplished by selecting a block of 16 consecutive opcodes.

Upon execution of "SET SCENE (*DTR0, sceneX*)", the scene number shall be derived from the opcode: *sceneNumber* = opcode – 0x40. This identifies the "*sceneX*" to be used.

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

Refer to 9.19 for further information.

11.4.15 REMOVE FROM SCENE (*sceneX*)

This command actually comprises 16 commands, one for each scene. This is accomplished by selecting a block of 16 consecutive opcodes.

Upon execution of "REMOVE FROM SCENE (*sceneX*), the scene number shall be derived from the opcode: *sceneNumber* = opcode – 0x50. This identifies the "*sceneX*" to be used.

"sceneX" shall be set to MASK. This effectively removes the control gear as member from the scene.

Refer to 9.19 for further information.

11.4.16 ADD TO GROUP (*group*)

This command actually comprises 16 commands, one for each group. This is accomplished by selecting a block of 16 consecutive opcodes.

Upon execution of "ADD TO GROUP (*group*)", *group* shall be derived from the opcode: *group* = opcode – 0x60. This identifies the *group* to be used.

bit[*group*] of *"gearGroups"* shall be set to TRUE. This implies that the control gear is a member of this group.

11.4.17 REMOVE FROM GROUP (*group*)

This command actually comprises 16 commands, one for each group. This is accomplished by selecting a block of 16 consecutive opcodes.

Upon execution of "REMOVE FROM GROUP (*group*)", *group* shall be derived from the opcode: *group* = opcode – 0x70. This identifies the *group* to be used.

bit[*group*] of *"gearGroups"* shall be set to FALSE. This implies that the control gear is not a member of this group.

11.4.18 SET SHORT ADDRESS (*DTR0*)

"shortAddress" shall be set to:

- if *"DTR0"* = MASK: MASK (effectively deleting the short address);
- if *"DTR0"* = 1xxxxxxxb or xxxxxxx0b: no change;
- in all other cases (0AAAAAA1b): 00AAAAAAb.

11.4.19 ENABLE WRITE MEMORY

"writeEnableState" shall be set to ENABLED.

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

Refer to 9.10.6 for further information.

11.5 Queries

11.5.1 General

Queries are used to retrieve property values from a control gear. The addressed control gear returns the queried property value in a backward frame.

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

- The query shall be ignored if so required by the provisions of 9.7.

When applicable, the query shall be discarded if any of the parameter values (in *"DTR0"*, *"DTR1"* and *"DTR2"*) are outside the range of validity of the addressed device variables, as given in Table 16.

11.5.2 QUERY STATUS

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

Refer to 9.16 for further information.

11.5.3 QUERY CONTROL GEAR PRESENT

The answer shall be YES.

11.5.4 QUERY CONTROL GEAR FAILURE

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

11.5.5 QUERY LAMP FAILURE

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

11.5.6 QUERY LAMP POWER ON

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

11.5.7 QUERY LIMIT ERROR

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

11.5.8 QUERY RESET STATE

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

11.5.9 QUERY MISSING SHORT ADDRESS

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

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

11.5.10 QUERY VERSION NUMBER

The answer shall be "*versionNumber*".

See 4.2 and Table 16.

11.5.11 QUERY CONTENT DTR0

The answer shall be *"DTR0"*.

11.5.12 QUERY DEVICE TYPE

The answer shall be:

- if no part of the IEC 62386-2xx series is implemented: 254;
- if one device type or feature is implemented: the device type or feature number;
- if more than one device type or feature is implemented: MASK.

The coding of the device types and features shall be as specified in the relevant parts of the IEC 62386-2xx series.

Refer to 9.18 and 11.5.13 for further information.

11.5.13 QUERY NEXT DEVICE TYPE

The answer shall be:

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

The sequence shall be cancelled if a 16-bit forward frame is received which meets any of the following conditions:

- the address byte differs, or
- the frame contains a command other than QUERY DEVICE TYPE and QUERY NEXT DEVICE TYPE.

The sequence of commands shall only be accepted as long as they use the same address byte. Multi-master transmitters shall send such sequence as a transaction. The coding of the device types and features shall be as specified in the relevant parts of the IEC 62386-2xx series.

Refer to 9.18 and 11.5.12 for further information.

11.5.14 QUERY PHYSICAL MINIMUM

The answer shall be PHM.

11.5.15 QUERY POWER FAILURE

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

11.5.16 QUERY CONTENT DTR1

The answer shall be *"DTR1"*.

11.5.17 QUERY CONTENT DTR2

The answer shall be *"DTR2"*.

11.5.18 QUERY OPERATING MODE

The answer shall be *"operatingMode"*.

Refer to 9.9 for further information.

11.5.19 QUERY LIGHT SOURCE TYPE

The answer shall be the number of the light source type given in Table 19.

Table 19 – Light source type encoding

Type of light source	Encoding	Lamp failure detection	
		Open circuit (lamp disconnected)	Short circuit
Low pressure fluorescent	0	✓	
HID	2	✓	
Low voltage halogen	3	✓	
Incandescent	4	✓	
LED	6	✓[a]	✓[a]
OLED	7	✓[a]	✓[a]
Other than listed above	252	✓	
Unknown light source type[b]	253	✓[d]	[d]
No light source[c]	254	[d]	[d]
Multiple light source types	MASK	✓[e]	✓[e]
Reserved	1, 5, [8,251]		

[a] Testing shall be done with light output of at least 5 %.

[b] Typically used in the case of signal conversion, for example 1 V to 10 V.

[c] Used in cases where no light source is connected, for example a relay.

[d] See 9.16.3.

[e] Depending on the supported light source types.

When MASK is answered the content of DTR0 shall contain a value representing the first light source type, DTR1 shall represent the second light source type, and DTR2 shall represent the third light source type.

When exactly two different light source types are available, DTR2 shall contain 254, indicating "no light source".

When more than three different light source types are available, DTR2 shall contain 255.

11.5.20 QUERY ACTUAL LEVEL

The answer shall be:

- if *"actualLevel"* = 0x00: 0x00 (see also 9.13);
- In all other cases:
  - during startup: MASK;
  - no light output (e.g. due to total lamp failure, control gear failure) while light output is expected: MASK;
  - in all other cases: *"actualLevel"*.

11.5.21 QUERY MAX LEVEL

The answer shall be *"maxLevel"*.

11.5.22 QUERY MIN LEVEL

The answer shall be *"minLevel"*.

11.5.23 QUERY POWER ON LEVEL

The answer shall be *"powerOnLevel"*.

Refer to 9.13 for further information.

11.5.24 QUERY SYSTEM FAILURE LEVEL

The answer shall be *"systemFailureLevel"*.

Refer to 9.12 for further information.

11.5.25 QUERY FADE TIME/FADE RATE

The answer shall be XXXX YYYYb, where XXXXb equals *"fadeTime"* and YYYYb equals *"fadeRate"*.

11.5.26 QUERY EXTENDED FADE TIME

The answer shall be 0 XXX YYYYb, where XXXb equals "*extendedFadeTimeMultiplier*" and YYYYb equals "*extendedFadeTimeBase*".

11.5.27 QUERY MANUFACTURER SPECIFIC MODE

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

11.5.28 QUERY SCENE LEVEL (*sceneX*)

This command actually comprises 16 commands, one for each scene. This is accomplished by selecting a block of 16 consecutive opcodes.

Upon execution of "QUERY SCENE LEVEL (*sceneX*)", the scene number shall be derived from the opcode: *sceneNumber* = opcode – 0xB0. This identifies the "*sceneX*" to be used.

The answer shall be *"sceneX"*.

Refer to 9.19 for further information.

11.5.29 QUERY GROUPS 0-7

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

The membership of groups 0 to 7 shall be represented as an 8-bit value, with one bit for each group. "0" shall be interpreted as not a member, and "1" shall be interpreted as member of the group. Bit[X] shall represent membership of group X, where X is in the range [0,7].

11.5.30 QUERY GROUPS 8-15

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

The membership of groups 8 to 15 shall be represented as an 8-bit value, with one bit for each group. "0" shall be interpreted as not a member, and "1" shall be interpreted as member of the group. Bit[X] shall represent membership of group X + 8, where X is in the range [0,7].

11.5.31 QUERY RANDOM ADDRESS (H)

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

11.5.32 QUERY RANDOM ADDRESS (M)

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

11.5.33 QUERY RANDOM ADDRESS (L)

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

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

The query shall be discarded if the addressed memory bank is not implemented.

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

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

NOTE 1 This allows gaps in the memory bank implementation.

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

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

Refer to 9.10.5 for further information.

11.6 Application extended commands

11.6.1 General

"ENABLE DEVICE TYPE (*data*)" shall be executed before an application extended command to enable the correct device type or feature command set. For further requirements, see command "ENABLE DEVICE TYPE (*data*)" and 11.7.14.

If "ENABLE DEVICE TYPE (*data*)" is discarded or not received before an application extended command is accepted, the application extended command shall be discarded.

The definitions of the extended commands are in the relevant parts of the IEC 62386-2xx series.

Refer to 9.18 for further information.

11.6.2 QUERY EXTENDED VERSION NUMBER

The answer shall be the version number of the relevant part of the IEC 62386-2xx series for the corresponding device type or feature as an 8-bit number.

The answer shall be:

- if the enabled device type or feature is not implemented: NO;
- if the enabled device type or feature is supported: the version number belonging to the device type or feature.

Refer to 9.18 for further information.

11.7 Special commands

11.7.1 General

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

11.7.2 TERMINATE

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

- Initialisation, "*initialisationState*" shall be set to DISABLED.
- Identification, whether started as part of initialisation (using RECALL MAX LEVEL, RECALL MIN LEVEL) or as a standard operation (IDENTIFY DEVICE) shall be stopped.

The command could also terminate other processes as identified in the relevant parts of the IEC 62386-2xx series.

Refer to 9.14.2 for further information.

11.7.3 DTR0 (*data*)

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

Refer to 9.10 for further information.

11.7.4 INITIALISE (*device*)

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

Only devices matching the given *device* shall respond to the instruction, as specified in Table 20:

Table 20 – Device addressing with "INITIALISE (*device*)"

Device	Responsive device(s)
0AAAAAA1b	Device(s) with *"shortAddress"* equal to 00AAAAAAb
11111111b	Control gear without *"shortAddress"* shall react
00000000b	All control gear shall react
Other	None

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

Refer to 9.14.2 for further information.

11.7.5 RANDOMISE

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

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

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

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

There shall be no reply to this instruction.

Refer to 9.14.2 for further information.

11.7.6 COMPARE

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

If executed, the control gear shall answer:

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

Refer to 9.14.2 for further information.

11.7.7 WITHDRAW

The instruction shall be discarded unless the following conditions hold:

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

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

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

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

Refer to 9.14.2 for further information.

11.7.8 SEARCHADDRH (*data*)

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

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

Refer to 9.14.2 for further information.

11.7.9 SEARCHADDRM (*data*)

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

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

Refer to 9.14.2 for further information.

11.7.10 SEARCHADDRL (*data*)

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

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

Refer to 9.14.2 for further information.

11.7.11 PROGRAM SHORT ADDRESS (*data*)

The instruction shall be discarded unless the following conditions hold:

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

If executed, *"shortAddress"* shall be set as follows:

- if *data* = MASK: MASK (effectively deleting the short address);
- if *data* = 1xxxxxxxb or xxxxxxx0b: no change;
- in all other cases (0AAAAAA1b): 00AAAAAAb.

Refer to 9.14.2 for further information.

11.7.12 VERIFY SHORT ADDRESS (*data*)

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

If executed, the answer shall be YES if *"shortAddress"* is equal to 00AAAAAAb for *data* given by 0AAAAAA1b, and NO otherwise.

Refer to 9.14.2 for further information.

11.7.13 QUERY SHORT ADDRESS

The query shall be discarded if:

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

If executed, the answer shall be 0AAAAAA1b, where "*shortAddress*" is equal to 00AAAAAAb, or MASK, in case "*shortAddress*" equals MASK.

Refer to 9.14.2 for further information.

11.7.14 ENABLE DEVICE TYPE (*data*)

This instruction shall select the device type or feature for which the next application extended command (refer to 11.6) is valid. Executing "ENABLE DEVICE TYPE (*data*)" shall cancel any previous selection of a device type or feature. The selection is only valid for the next application extended command.

If a non-application extended command is accepted after executing "ENABLE DEVICE TYPE (*data*)", the enabling of the device type or feature shall be cancelled and that command shall be executed according to its specification.

A control gear shall not react to commands which belong to the application extended commands if *data* equals MASK, 254 or represents a device type or feature not supported by this control gear.

The device types and features shall be coded as specified in the relevant parts of the IEC 62386-2xx series.

Application controllers should be able to identify individual control gear and store the relationship between each control gear's address and the device types and features, in persistent memory.

11.7.15 DTR1 (*data*)

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

Refer to 9.10 for further information.

11.7.16 DTR2 (*data*)

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

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

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

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

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

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

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

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

If the selected memory bank location is

- not implemented, or
- above the last accessible memory location, or
- locked (see 9.10.2), or
- not writeable,

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

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

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

Refer to 9.10 for further information.

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

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

Refer to 9.10 for further information.

11.7.19 PING

The ping command is used by single master application controllers (see IEC 62386-103) to indicate their presence. The ping command shall be ignored by the control gear.

Annex A
(informative)

Examples of algorithms

A.1 Random address allocation

The control gear are connected to a control device that uses random address allocation for setup of the system.

a) Start the algorithm with "INITIALISE (*device*)" which enables the addressing commands for a time period of 15 min.

b) Send "RANDOMISE"; all control gear choose a *randomAddress* so that 0 ≤ *randomAddress* ≤ $+2^{24}-2$.

c) The control device searches the control gear with the lowest random address by means of an algorithm which uses "SEARCHADDRH (*data*)", "SEARCHADDRM (*data*)", "SEARCHADDRL (*data*)" and "COMPARE". The control gear with the lowest random address is found. At this point, the control device needs to be able to handle different timing in backward frames coming from different control gear. Also, there is a chance that the control gear generated the same *randomAddress* in which case randomisation should be restarted for the remaining gear.

d) The short address is programmed to the control gear found with aid of "PROGRAM SHORT ADDRESS (*data*)".

e) "VERIFY SHORT ADDRESS (*data*)" can be used to verify the correct programming.

f) The found control gear can be identified by using "IDENTIFY DEVICE", or use an alternating sequence of "RECALL MAX LEVEL" and "RECALL MIN LEVEL" with the programmed short address to record the local position of the respective control gear.

g) If needed, the short address can be changed going back to step d).

h) The control gear found shall be removed from the search process by means of "WITHDRAW".

i) Repeat from step c) on until no further control gear can be found. Use "INITIALISE (*device*)" to prolong the 15 min timer if needed.

j) Stop the process with "TERMINATE".

In the event two or more control gear have the same short address, restart the addressing procedure only for these control gear with "INITIALISE (*device*)" (using the short address in the second byte) followed by steps b) to j).

A.2 One single control gear connected to the control device

Only one control gear is connected to a control device that uses the following algorithm to program a short address.

a) Transmit the new short address (0AAA AAA1b) by "DTR0 (*data*)".

b) Verify the content of the DTR0 by "QUERY CONTENT DTR0".

c) Send "SET SHORT ADDRESS (*DTR0*)" twice in accordance with the requirements specified in IEC 62386-101:2022, 9.4.

A.3 Using application extended commands

A control device using application extend commands needs to detect which application extended commands are supported by the different control gear. The following algorithm can be used:

a) Initialisation process and address allocation.

b) Query the device type or feature of every control gear in the system. If the received answer is 'MASK' the device supports more than one device type or feature. In this case the following procedure can be used to get a list of the device types the control gear belongs to:

   1) Send the "QUERY EXTENDED VERSION NUMBER" command preceded by "ENABLE DEVICE TYPE (*data*)" with *data* equal to "0". If there is an answer, the control gear belongs to device type 0.

   2) Repeat step a) with all other device types and features supported by the control device.

c) The control device shall send "ENABLE DEVICE TYPE (*data*)" before every application extended command.

Annex B
(normative)

High resolution dimmer

A high resolution dimmer is not mandatory. However, if it is implemented, it shall be implemented according to this Annex B.

The *"actualLevel"* shall report the nearer level, after rounding of an internal value as can be seen in Figure B.1.

Light output shall always match a discrete point on the selected dimming curve, except when:

- a fade is running (via ideal, or high resolution internal curve);
- a fade is stopped before the fade time has elapsed;
- another dimming curve is selected;
- a level was programmed with another dimming curve (typically the scenes, including power on level and system failure level, store the level as well as the corresponding dimming curve, to allow representing in other curves and back without loss).

When light output is "in between" two discrete points on the selected dimming curve:

- *"actualLevel"* is set to the nearest of these two points;
- a new fade starts from the actual light output (which will possibly not match a discrete point on the selected dimming curve) and ends at the target light output (which will possibly not match a discrete point on the selected dimming curve, i.e. when a preset is used that was set using another dimming curve);
- the fade shall always follow the ideal or internal high resolution curve without making jumps to a discrete point on the selected dimming curve;
- there are some consequences for testing:
  - after stopping a fade, or switching dimming curve, a first step can be less than or more than a full step in the active dimming curve;
  - testing with levels from more than one dimming curve at a time is complex, perhaps better avoided.

Behaviour that can be experienced:

- theoretically a minimal fade is possible while *"actualLevel"* is not changed;
- direct arc power command (DAPC) to the actual level can fade from the point on the high resolution dimmer to the discrete point (less than half a dim step);
- a STEP UP or STEP DOWN can, in the worst case, result in a "half step" delta or "one and a half step"; for example a high resolution dimmer that ended at stepsize + 0,49 stepsize: response to step up is 0,51 stepsize, while the response to step down is 1,49 stepsize to end at a discrete step;
- the "fade running" status is set to "1" at fade time start and lasts until FadeTime has elapsed, even when "Actual Level" is already at the final level;
- actual level always represents the nearer discrete point on the dimming curve.

Special cases:

- When a fade is started from the "Lamp off" condition, the first step from off to "Min Level" shall be made at fade start time. The step from off to min level is not part of the fade time.
- When a fade is started to the "Lamp off" condition, the last step from "Min Level" to off shall be made at fade start time + FadeTime. The step from min level to off is not part of the fade time.

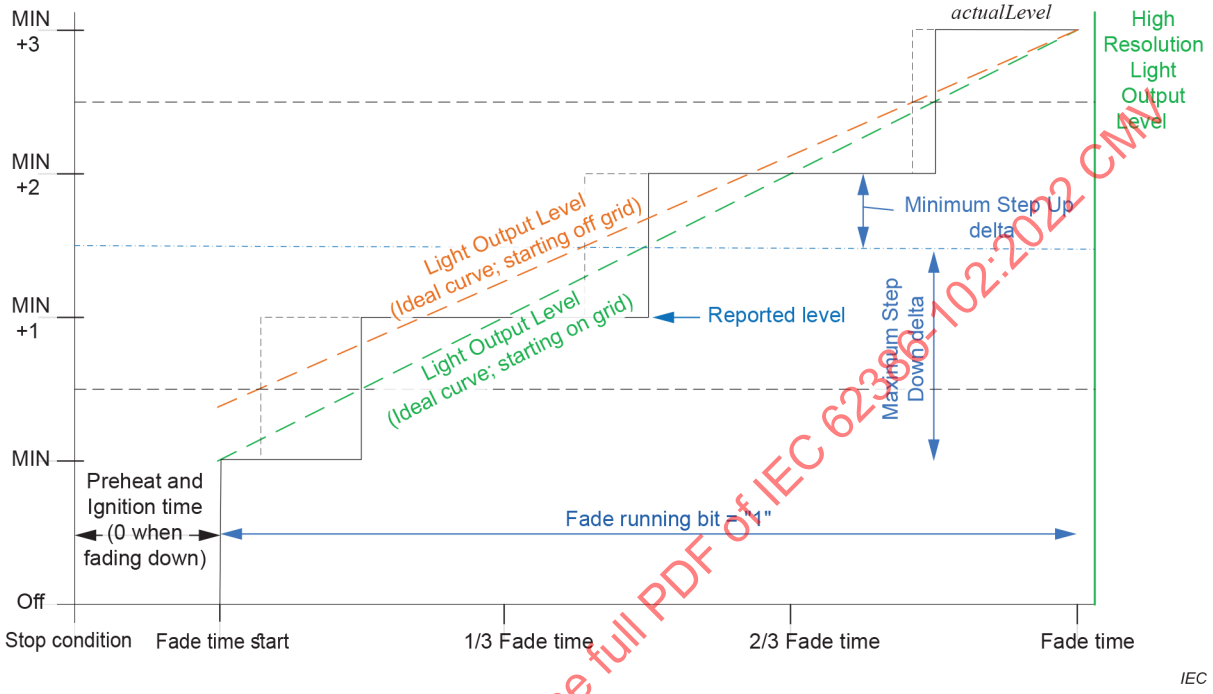


Figure B.1 – Level behaviour in case of off-grid starting points

Bibliography

[1] GS1 *General Specifications*, latest version, available at: www.gs1.org

[2] IEC 62386-102:2014[2], *Digital addressable lighting interface – Part 102: General requirements – Control gear*

[3] IEC 62386-104, *Digital addressable lighting interface – Part 104: General requirements – Wireless and alternative wired system components*

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

SOMMAIRE

COMMISSION ÉLECTROTECHNIQUE INTERNATIONALE

_____

INTERFACE D'ÉCLAIRAGE ADRESSABLE NUMÉRIQUE –

Partie 102: Exigences générales – Appareillages de commande

AVANT-PROPOS

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

L'IEC 62386-102 a été établie par le comité d'études 34 de l'IEC: Eclairage. Il s'agit d'une Norme internationale.

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

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

- le domaine d'application a été mis à jour;
- les références ont été mises à jour;
- la lecture dans le bloc de mémoire de valeurs à plusieurs octets a été ajoutée;

- le bloc de mémoire 0 et les exigences communes pour les blocs de mémoire ont été mis à jour;
- les blocs de mémoire réservés ont été mis à jour;
- la durée de sauvegarde de la mémoire non volatile (NVM) a été ajoutée, et la commande SAVE PERSISTENT VARIABLES a été supprimée;
- le numéro de version a été mis à jour;
- la configuration de l'unité de bus a été ajoutée.

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

Projet	Rapport de vote
34/948/FDIS	34/989/RVD

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

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

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

La présente Partie 102 de l'IEC 62386 est destinée à être utilisée conjointement avec la Partie 101, qui contient les exigences générales pour le type de produit applicable (système), et avec la Partie 2xx appropriée (exigences particulières pour les appareillages de commande) qui comporte les articles qui complètent ou modifient les articles correspondants de la Partie 101 et de la Partie 102, afin de fournir les exigences correspondantes pour chaque type de produit.

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

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

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

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

INTRODUCTION

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

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

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

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

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

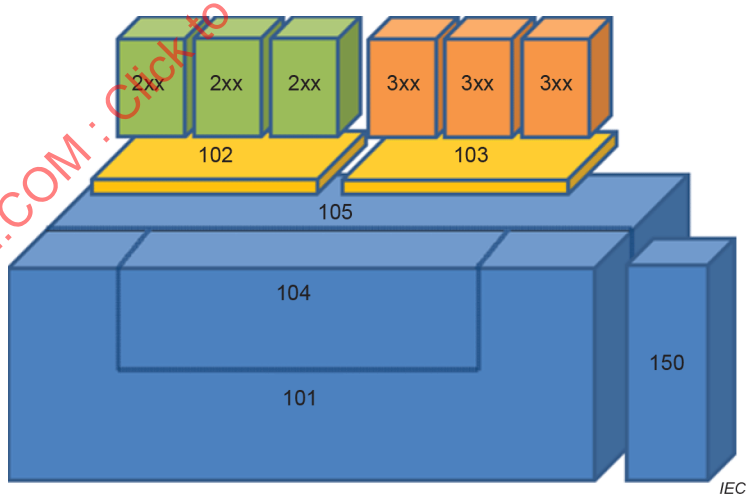


Figure 1 – Représentation graphique générale de l'IEC 62386

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

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

Les expressions typographiques suivantes sont utilisées:

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

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

Commande: "NOM DE LA COMMANDE".

INTERFACE D'ÉCLAIRAGE ADRESSABLE NUMÉRIQUE –

Partie 102: Exigences générales – Appareillages de commande

1 Domaine d'application

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

2 Références normatives

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

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

IEC 62386-103:2022, *Interface d'éclairage adressable numérique – Partie 103: Exigences générales – Dispositifs de commande*

IEC 62386-2xx (toutes les parties), *Interface d'éclairage adressable numérique – Partie 2xx: Exigences particulières pour les appareillages de commande*

3 Termes et définitions

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

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

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

3.1
niveau réel
valeur qui représente le rendement lumineux actuel

3.2
puissance d'arc
puissance fournie aux sources de lumière (lampes)

3.3
diffusion
type d'adresse utilisé pour adresser simultanément l'ensemble des appareillages de commande dans le système

3.4
diffusion non adressée
type d'adresse utilisé pour adresser simultanément l'ensemble des appareillages de commande dans le système qui n'ont pas d'adresse courte

3.5
DAPC
contrôle direct de la puissance d'arc
méthode de contrôle direct du rendement lumineux

Note 1 à l'article: L'abréviation "DAPC" est dérivée du terme anglais développé correspondant "direct arc power control".

3.6
DTR
registre de transfert des données
registre polyvalent utilisé pour l'échange de données

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

3.7
adresse de groupe
type d'adresse utilisé pour adresser simultanément un groupe d'appareillages de commande dans le système

3.8
GTIN
code article international
numéro utilisé pour identifier de façon unique, partout dans le monde, les articles commercialisés

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

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

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

3.9
identification
état provisoire appliqué lors de la mise en service qui permet à l'installateur d'identifier un appareillage de commande spécifique

3.10
niveau
valeur à 8 bits

3.11
MASK
valeur dont tous les chiffres binaires sont définis sur 1

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

3.12
NO
réponse à une requête dans laquelle aucune trame en arrière n'est envoyée

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

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

3.13
NVM
mémoire non volatile
mémoire en lecture/écriture non volatile dont le contenu peut être modifié et n'est pas perdu en raison d'un cycle de mise sous tension

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

3.14
NVM-RO
NVM qui ne peut faire l'objet d'une saisie à l'aide d'aucune commande

3.15
NVM-RW
NVM qui peut être modifiée à l'aide d'une ou de plusieurs commandes

3.16
opcode
code de fonctionnement
partie d'une trame en avant qui identifie la commande à exécuter

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

3.17
mode de fonctionnement
ensemble d'états identifiés par un nombre compris dans la plage [0,255], caractérisé par un groupe de variables et de paramètres de mémoire, et utilisé pour sélectionner un ensemble de fonctionnalités à présenter par un appareillage de commande, y compris sa réaction nécessaire aux commandes

Note 1 à l'article: Un appareillage de commande peut prendre en charge plusieurs modes de fonctionnement.

3.18
PHM
niveau physique minimal
niveau qui correspond au rendement lumineux minimal auquel l'appareillage de commande peut fonctionner

Note 1 à l'article: L'abréviation "PHM" est dérivée du terme anglais développé correspondant "physical minimum level".

3.19
RAM
mémoire en lecture/écriture volatile dont le contenu peut être modifié et est perdu en raison d'un cycle de mise sous tension

3.20
RAM-RO
RAM qui ne peut faire l'objet d'une saisie à l'aide d'aucune commande

3.21
RAM-RW
RAM qui peut être modifiée à l'aide d'une ou de plusieurs commandes

3.22
adresse aléatoire
numéro à 24 bits aléatoire généré par l'appareillage de commande sur demande au cours de l'initialisation du système

Note 1 à l'article: L'Article A.1 fournit un exemple de pratique d'utilisation des adresses de recherche et aléatoires.

3.23
état réinitialisé
état dans lequel toutes les variables NVM de l'appareillage de commande présentent des valeurs réinitialisées, sauf celles qui portent le marquage "pas de modification" ou qui sont explicitement exclues de toute autre manière

3.24
ROM
mémoire en lecture seule non volatile dont le contenu est fixe

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

3.25
scénario
niveau préréglé configurable

3.26
adresse de recherche
nombre à 24 bits utilisé pour identifier un appareillage de commande particulier dans le système au cours de l'initialisation

Note 1 à l'article: L'Article A.1 fournit un exemple de pratique d'utilisation des adresses de recherche et aléatoires.

3.27
adresse courte
type d'adresse utilisé pour adresser un appareillage de commande particulier dans le système

3.28
démarrage
temps nécessaire pour passer de l'état hors tension de la lampe au fonctionnement normal de la lampe ou à l'état de défaillance

Note 1 à l'article: Ce temps inclut le préchauffage et l'amorçage.

3.29
strictement monotonique
entièrement croissant ou entièrement décroissant sans répétition des valeurs

3.30
niveau cible
valeur qui représente le rendement lumineux cible prévu après exécution de la commande de niveau réel

3.31
TMASK
valeur dont le bit de poids faible est défini sur 0 et dont tous les autres chiffres binaires sont définis sur 1

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

3.32
YES
réponse à une requête dans laquelle une trame en arrière de MASK est envoyée

4 Généralités

4.1 Généralités

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

4.2 Numéro de version

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

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

A chaque amendement d'une édition de l'IEC 62386-102, le numéro de version secondaire doit être augmenté de un.

Lors d'une nouvelle édition de l'IEC 62386-102, le numéro de version principale doit être augmenté de un et le numéro de version secondaire doit être égal à 0.

Le numéro de version actuel est "*versionNumber*" comme cela est défini dans le Tableau 16.

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

5 Spécifications électriques

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

6 Alimentation électrique du bus

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

7 Structure du protocole de transmission

7.1 Généralités

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

7.2 Codage de trame en avant à 16 bits

7.2.1 Généralités

Pour les commandes, la trame en avant à 16 bits doit être codée comme cela est indiqué dans le Tableau 1.

Tableau 1 – Codage de la trame de commande à 16 bits

Octets/Bits									Méthode d'adressage des dispositifs
Octet d'adresse								Octet de code de fonctionnement	
15	14	13	12	11	10	9	8[a]	7...0	
0	64 adresses courtes						x		Adressage court
1	0	0	16 adresses de groupe				x		Adressage de groupes
1	1	1	1	1	1	0	x		Diffusion non adressée
1	1	1	1	1	1	1	x		Diffusion
1010 0000 à 1100 1011									Commande spéciale
1100 1100 à 1111 1011									Réservé

[a] Bit sélecteur, voir 7.2.2; 0 indique DAPC, 1 indique une autre commande.

7.2.2 Octet d'adresse

L'octet d'adresse fournit

- la méthode d'adressage des dispositifs utilisée par le contrôleur d'application;
- le type de commande transmis dans l'octet de code de fonctionnement;
  Bit 8 = '1': commande normalisée;
  Bit 8 = '0': commande de contrôle direct de la puissance d'arc (DAPC);
- les espaces d'adressage pour les commandes spéciales;
- les adresses de dispositif réservées.

Il convient que le contrôleur d'application n'utilise pas les adresses réservées.

7.2.3 Octet de code de fonctionnement

L'octet de code de fonctionnement fournit

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

8 Cadencement

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

9 Mode de fonctionnement

9.1 Généralités

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

9.2 Appareillage de commande

9.2.1 Généralités

Les appareillages de commande reçoivent des commandes des contrôleurs d'application. Les contrôleurs d'application sont spécifiés par l'IEC 62386-103:2022.

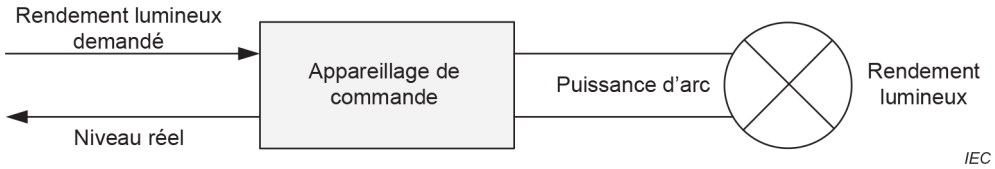


Figure 2 – Appareillages de commande qui font fonctionner directement une source de lumière

La Figure 2 montre comment les différents niveaux génèrent le rendement lumineux. Le niveau de rendement (lumineux) maximal d'un appareillage de commande est défini comme étant égal à 100 %. Tous les niveaux sont spécifiés de manière relative. L'appareillage de commande peut fournir, physiquement, un niveau minimal tout en maintenant un rendement lumineux effectif. Ce niveau est défini comme le niveau minimal physique (PHM).

NOTE Le PHM est spécifique à l'appareillage de commande et il est supérieur à 0.

9.2.2 Phases de l'appareillage de commande

9.2.2.1 Généralités

Selon la source de lumière, différentes phases de fonctionnement peuvent être identifiées au sein d'un appareillage de commande. Voir 9.2.2.2 à 9.2.2.5.

9.2.2.2 Veille

Au cours de cette phase, la lampe est éteinte et, durant cette phase uniquement, "*targetLevel*" et "*actualLevel*" sont à 0.

9.2.2.3 Démarrage

Le démarrage est une phase de transition entre l'état de veille et le fonctionnement normal ou la défaillance. Cette phase est parfois perçue comme un retard. Exemples:

- préchauffage: la lampe est chauffée en vue de son amorçage. Ceci est généralement observé pour les sources de lumière à fluorescence;
- amorçage: la lampe est amorcée. Ceci est généralement observé pour les sources de lumière à décharge à haute intensité (HID, *High Intensity Discharge*) et les sources de lumière à fluorescence après préchauffage;
- préparation de la mise sous tension.

Pour les informations complémentaires et les exceptions, se référer au 9.16.3.

9.2.2.4 Fonctionnement normal

En fonctionnement normal, la lampe émet de la lumière et peut fonctionner comme attendu.

Pour les informations complémentaires et les exceptions, se référer au 9.16.3.

9.2.2.5 Défaillance

Au cours de la phase de défaillance, la lampe ne peut pas fonctionner comme attendu.

Pour les informations complémentaires et les exceptions, se référer au 9.16.3.

9.3 Courbe de gradation

La courbe de gradation détermine de quelle manière le niveau doit être transformé en rendement lumineux.

Un *"actualLevel"* supérieur ou égal à 1 et inférieur ou égal à 254 doit être transformé en rendement lumineux selon

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

Le rendement lumineux est exprimé par rapport au rendement lumineux maximal possible d'une combinaison appareillage de commande-lampe donnée. La courbe de gradation commence à 0,1 % pour *"actualLevel"* égal à 0x01 et finit à 100 % pour *"actualLevel"* égal à 0xFE. La courbe de gradation est strictement monotonique, et la précision relative doit être de ±½ pas. Ceci doit être vérifié par essai utilisant une modification de l'intensité lumineuse, en excluant le PHM.

NOTE 1 La courbe de gradation est destinée à compenser la courbe de sensibilité à la lumière de l'œil humain.

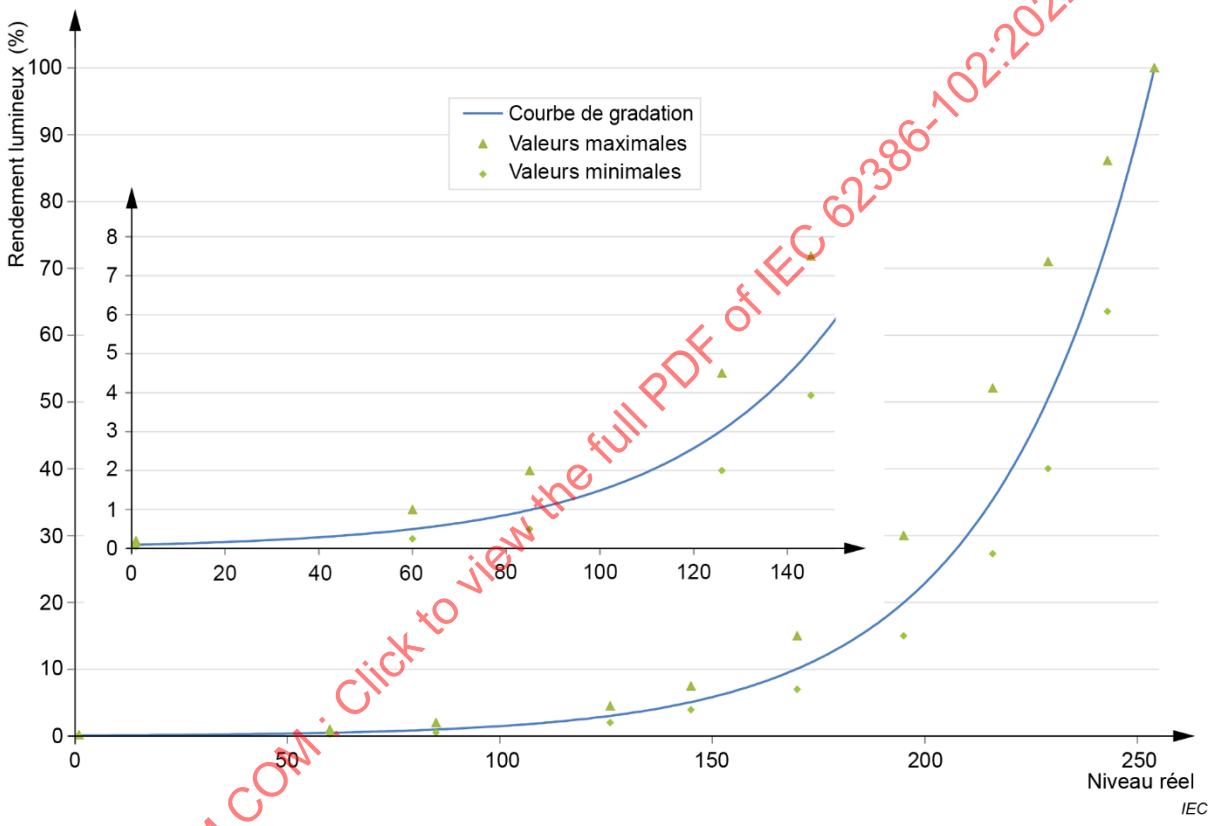


Figure 3 – Courbe de gradation

La précision du rendement lumineux est spécifiée par les points d'essai donnés dans le Tableau 2. Les points d'essai du Tableau 2 et la courbe de gradation sont représentés à la Figure 3, et le Tableau 3 montre le rendement lumineux par rapport au niveau. Le type de lampe ou la charge utilisé(e) lors des essais doit être indiqué(e) à des fins de reproductibilité.

Tableau 2 – Tolérance de la courbe de gradation

Niveau	1	60	85	126	145	170	195	216	229	243	254
Valeur minimale	0,05	0,25	0,50	2,00	3,93	7,00	15,00	27,28	40,00	63,53	
Valeur nominale	0,10	0,50	0,99	3,04	5,10	10,09	19,97	35,43	50,53	74,05	100,00
Valeur maximale	0,20	1,00	2,00	4,50	7,50	15,0	30,00	52,09	71,00	86,14	
Les tolérances sont exprimées en % et arrondies à deux décimales.											

Tableau 3 – Courbe de gradation

Niveau	Rendement lumineux	Niveau	Rendement lumineux	Niveau	Rendement lumineux	Niveau	Rendement lumineux	Niveau	Rendement lumineux
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

Niveau	Rendement lumineux	Niveau	Rendement lumineux	Niveau	Rendement lumineux	Niveau	Rendement lumineux	Niveau	Rendement lumineux
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
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 Calcul de *"targetLevel"*

Un contrôleur d'application indique à l'appareillage de commande le rendement lumineux demandé et le comportement effectif au cours du passage d'*"actualLevel"* à *"targetLevel"* au moyen de codes de fonctionnement appropriés.

Le *"targetLevel"* doit être calculé à partir du rendement lumineux demandé comme suit:

- 0x00 doit être accepté comme *"targetLevel"* et avec la lampe éteinte;
- toute valeur comprise entre 0x01 et *"minLevel"* doit donner *"targetLevel" = "minLevel"*;
- toute valeur comprise entre *"maxLevel"* et 0xFE doit donner *"targetLevel" = "maxLevel"*;
- "MASK" ne doit pas influer sur *"targetLevel"* sauf lorsqu'une modification de l'intensité lumineuse est en cours, voir 9.5.9;
- toutes les autres valeurs doivent être acceptées comme *"targetLevel"*.

Le calcul du niveau cible demandé de *"targetLevel"* doit également être appliqué si la demande est fondée sur une valeur enregistrée, telle qu'un scénario, *"powerOnLevel"* ou *"systemFailureLevel"*.

A chaque modification de *"targetLevel"*, à l'exception de l'initialisation provoquée par un cycle de mise sous tension, l'appareillage de commande doit actualiser *"limitError"* (voir 9.16.5) et doit régler *"lastLightLevel"* sur le nouveau *"targetLevel"*. Si "*targetLevel"* n'est pas égal à 0x00, *"lastActiveLevel"* doit être réglé sur "*targetLevel"*.

9.5 Modification de l'intensité lumineuse

9.5.1 Généralités

Le processus de modification de l'intensité lumineuse correspond à un temps de transition linéaire entre *"actualLevel"* et *"targetLevel"*. *"actualLevel"* et, par conséquent, le rendement lumineux, doivent être strictement monotoniques selon la courbe de gradation applicable.

Une modification de l'intensité lumineuse peut être lancée de deux manières:

- utilisation d'une durée de modification de l'intensité lumineuse: cette méthode définit une durée d'utilisation du processus de modification de l'intensité lumineuse;
- utilisation d'une vitesse de modification de l'intensité lumineuse: cette méthode définit une vitesse d'utilisation du processus de modification de l'intensité lumineuse.

Une modification de l'intensité lumineuse ne doit pas être lancée si le *"targetLevel"* calculé équivaut à *"actualLevel"*.

Lorsqu'une modification de l'intensité lumineuse est lancée, la minuterie associée doit être déclenchée et *"fadeRunning"* doit être réglé sur TRUE (voir 9.16.6).

Au cours de la modification de l'intensité lumineuse, le rendement lumineux doit être maintenu le plus près possible de la courbe de modification de l'intensité lumineuse idéale.

Au cours d'un processus de modification ascendante de l'intensité lumineuse, *"actualLevel"* doit être augmenté selon une durée qui correspond à l'intersection d'une courbe de modification de l'intensité lumineuse idéale avec le point milieu entre *"actualLevel"* et *"actualLevel"* + 1. De même, au cours d'une modification descendante de l'intensité lumineuse, *"actualLevel"* doit être diminué selon une durée qui correspond à l'intersection d'une courbe de modification de l'intensité lumineuse idéale avec le point milieu entre *"actualLevel"* et *"actualLevel"* – 1. La Figure 4 représente cet aspect, et s'applique aux modifications de l'intensité lumineuse lancées avec une durée ou une vitesse de modification de l'intensité lumineuse.

Les mesurages de la durée ou de la vitesse de modification de l'intensité lumineuse doivent commencer après l'arrêt de la commande qui les déclenche. Lorsque la modification de l'intensité lumineuse se produit immédiatement après le démarrage, les mesurages doivent être effectués à partir du moment où "*lampOn*" est TRUE ou, dans le cas d'une défaillance totale des lampes, à partir du moment où "*lampFailure*" est confirmée comme TRUE. Une modification de l'intensité lumineuse doit s'interrompre automatiquement lorsque la minuterie associée a été active pendant la durée correspondante applicable. A ce stade, la minuterie de modification de l'intensité lumineuse doit être interrompue et *"fadeRunning"* doit être réglé sur FALSE (voir 9.16.6).

Cela signifie que l'appareillage de commande modifie l'intensité lumineuse jusqu'au niveau cible même en cas de lampe totalement grillée. Si une lampe doit être éteinte à la fin de la modification de l'intensité lumineuse, la phase de transition de *"minLevel"* à 0x00 ne doit pas contribuer à la durée de ladite modification. La phase de transition de *"minLevel"* à 0x00 doit être suivie immédiatement après écoulement de la durée de modification de l'intensité lumineuse.

Si une lampe doit être allumée au début de la modification de l'intensité lumineuse et si son intensité lumineuse est à modifier pour atteindre une certaine valeur, la phase de transition de 0x00 à *"minLevel"* ne doit pas contribuer à la durée de ladite modification. Cela signifie que la durée de modification de l'intensité lumineuse commence lorsque la phase de démarrage est terminée.

NOTE Le passage de 0x00 à *"minLevel"* comprend le démarrage.

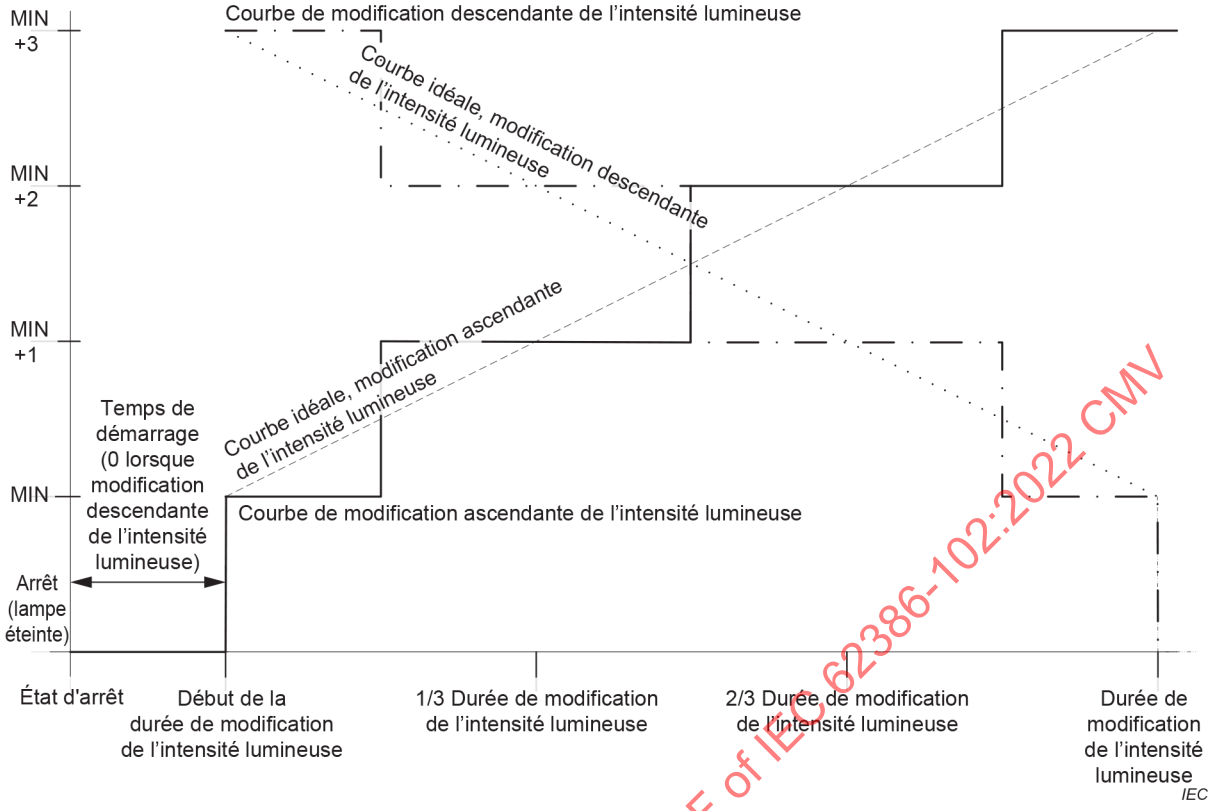


Figure 4 – Niveau par rapport à la durée, modification ascendante et descendante de l'intensité lumineuse

Les essais doivent être réalisés avec "*minLevel*" ≥ PHM + 1. Pour plus d'informations, voir Annexe B.

9.5.2 Durée de modification de l'intensité lumineuse

La durée de modification de l'intensité lumineuse doit être conforme au Tableau 4:

"fadeTime" doit être défini à réception de la commande "SET FADE TIME (*DTR0*)". "*fadeTime*" peut faire l'objet d'une requête en utilisant QUERY FADE TIME/FADE RATE;

"fadeTime" doit être réglé sur une valeur selon les phases suivantes:

- si *"DTR0"* > 15: 15
- dans tous les autres cas: *"DTR0"*

La durée de modification de l'intensité lumineuse doit être calculée à partir de *"fadeTime"* comme suit:

- si *"fadeTime"* = 0: utiliser la durée étendue de modification de l'intensité lumineuse (voir 9.5.4);
- si *"fadeTime"* se situe dans la plage [1,15]: $2^{\left(\frac{"fadeTime"}{2}-1\right)}$ s.

Le Tableau 4 répertorie les valeurs possibles de durée de modification de l'intensité lumineuse.

Tableau 4 – Durées de modification de l'intensité lumineuse

"fadeTime"	***Durée minimale de modification de l'intensité lumineuse*** s	***Durée nominale de modification de l'intensité lumineuse*** s	***Durée maximale de modification de l'intensité lumineuse*** s
0	Modification de l'intensité lumineuse étendue		
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 Vitesse de modification de l'intensité lumineuse

La vitesse de modification de l'intensité lumineuse doit être conforme au Tableau 5.

"fadeRate" doit être définie à réception de la commande "SET FADE RATE (*DTR0*)". "*fadeRate*" peut faire l'objet d'une requête en utilisant QUERY FADE TIME/FADE RATE.

"fadeRate" doit être réglée sur une valeur selon les phases suivantes:

- si *"DTR0"* > 15: 15
- Si *"DTR0"* = 0: 1
- dans tous les autres cas: *"DTR0"*

La vitesse de modification de l'intensité lumineuse doit être calculée à partir de *"fadeRate"* comme suit:

$$\text{Vitesse de modification de l'intensité lumineuse} = \frac{506}{2^{\left(\frac{\text{"fadeRate"}}{2}\right)}} \text{ pas/s.}$$

Le Tableau 5 répertorie les valeurs possibles de vitesse de modification de l'intensité lumineuse.

Tableau 5 – Vitesses de modification de l'intensité lumineuse

"fadeRate"	*Vitesse minimale de modification de l'intensité lumineuse* pas/s	*Vitesse nominale de modification de l'intensité lumineuse* pas/s	*Vitesse maximale de modification de l'intensité lumineuse* pas/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 Durée étendue de modification de l'intensité lumineuse

Si "*fadeTime*" est égale à 0, et si la durée rapide de modification de l'intensité lumineuse telle que définie dans l'IEC 62386-207 est mise en œuvre et est égale à 0, la durée étendue de modification de l'intensité lumineuse doit être utilisée.

La durée étendue de modification de l'intensité lumineuse peut être définie en utilisant une valeur de base et un multiplicateur selon le Tableau 6 et le Tableau 7. La durée étendue de modification de l'intensité lumineuse peut être calculée à partir de la valeur de base et du facteur de multiplication.

Durée de modification de l'intensité lumineuse =
$extendedFadeTimeBase \cdot extendedFadeTimeMultiplier$

Ceci donne une plage de 100 ms pour une durée de 16 min, et une valeur spéciale qui indique aucune modification de l'intensité lumineuse (le plus rapidement possible).

Tableau 6 – Durée étendue de modification de l'intensité lumineuse – Valeur de base

Bits de base	Valeur de base
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

Tableau 7 – Durée étendue de modification de l'intensité lumineuse – Multiplicateur

Bits de multiplicateur	Facteur de multiplication		
	Minimal	Nominal	Maximal
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		Réservé	
110b		Réservé	
111b		Réservé	
[a] Aucune modification de l'intensité lumineuse (le plus rapidement possible).			

À l'exécution de "SET EXTENDED FADE TIME (*DTR0*)", l'appareillage de commande doit définir les valeurs suivantes à partir de *"DTR0"*. Le format utilisé doit être 0YYYAAAAb, où YYYb est égal au multiplicateur de la durée de modification de l'intensité lumineuse, et AAAAb équivaut à la base de calcul de cette même durée: La durée de modification de l'intensité lumineuse résultante doit être augmentée de façon monotonique lorsque la durée de base augmente.

- Si "*DTR0*" > 0100 1111b:
  - "*extendedFadeTimeBase*" doit être réglée sur 0;
  - "*extendedFadeTimeMultiplier*" doit être réglé sur 0 ms, en réglant effectivement la durée de modification de l'intensité lumineuse sur 0 s, ce qui signifie aucune modification de l'intensité lumineuse (le plus rapidement possible). Le passage d'*"actualLevel"* à *"targetLevel"* doit être immédiat et le rendement lumineux doit être ajusté le plus rapidement possible, ce qui signifie moins de 0,8 s, soit la durée maximale de modification de l'intensité lumineuse pour *"fadeTime"* = 1 (voir Tableau 4).

- Dans tous les autres cas:
  - "*extendedFadeTimeBase*" doit être réglée sur AAAAb;
  - "*extendedFadeTimeMultiplier*" doit être réglé sur YYYb.

La durée étendue de modification de l'intensité lumineuse peut faire l'objet d'une requête en utilisant "QUERY EXTENDED FADE TIME". La réponse doit être 0 YYY AAAAb, où YYYb équivaut à "*extendedFadeTimeMultiplier*" et AAAAb équivaut à "*extendedFadeTimeBase*".

9.5.5 Utilisation de la durée de modification de l'intensité lumineuse

Les commandes qui utilisent une durée de modification de l'intensité lumineuse doivent lancer la modification en utilisant la durée applicable. Cette durée peut être déterminée à partir des règles suivantes:

- si "*fadeTime*" > 0: voir Tableau 4;
- si "*fadeTime*" = 0: la durée étendue de modification de l'intensité lumineuse doit être utilisée, voir Tableau 6 et Tableau 7. La durée étendue de modification de l'intensité lumineuse peut être calculée par multiplication de la valeur de base et du multiplicateur;
- si "*extendedFadeTimeMultiplier*" = 0 ms, la durée de modification de l'intensité lumineuse est égale à 0 s, ce qui signifie aucune modification de l'intensité lumineuse (le plus rapidement possible). Le passage d'*"actualLevel"* à *"targetLevel"* doit être immédiat et le rendement lumineux doit être ajusté le plus rapidement possible.

Le niveau cible doit être calculé à partir du paramètre de commande. Le niveau cible calculé doit être atteint après expiration de la durée de modification de l'intensité lumineuse.

La durée étendue de modification de l'intensité lumineuse prend également en charge les durées de modification de l'intensité lumineuse inférieures à 0,7 s. Si une durée de modification de l'intensité lumineuse inférieure à 0,7 s est lancée alors que cette durée ne peut pas être exécutée par la combinaison d'appareillage de commande et de source de lumière, l'appareillage de commande doit ajuster le rendement lumineux le plus rapidement possible. Toutefois, l'appareillage de commande doit répondre comme si la modification de l'intensité lumineuse était achevée dans la durée demandée.

9.5.6 Utilisation de la vitesse de modification de l'intensité lumineuse

9.5.6.1 Modification de l'intensité lumineuse au moyen des commandes "UP" et "DOWN"

Les commandes "UP" et "DOWN" doivent lancer une modification d'une durée de 200 ms ± 20 ms.

"*targetLevel*" doit être calculé à partir d'*"actualLevel"* en utilisant la vitesse de modification de l'intensité lumineuse applicable. Le niveau cible calculé doit être atteint après expiration de la modification de l'intensité lumineuse d'une durée de 200 ms.

NOTE 1 Etant donné que la vitesse de modification de l'intensité lumineuse est utilisée, il est possible d'atteindre *"minLevel"* ou *"maxLevel"* avant la fin de la modification. Ceci n'entraîne pas la suppression du bit *"fadeRunning"*.

NOTE 2 En raison des tolérances relatives à la vitesse de modification de l'intensité lumineuse, différents appareillages de commande peuvent réagir aux commandes qui utilisent des vitesses effectives légèrement différentes. Par conséquent, après le traitement de ces commandes de gradation relative, différents appareillages de commande peuvent avoir des valeurs différentes pour *"targetLevel"* (et par conséquent également pour "*actualLevel*" et *"lastLightLevel"*).

9.5.6.2 Modification de l'intensité lumineuse au moyen des commandes "CONTINUOUS UP" et "CONTINUOUS DOWN"

La commande "CONTINUOUS UP " doit régler *"targetLevel"* à *"maxLevel"* et lancer la modification en utilisant la vitesse applicable. La modification de l'intensité lumineuse doit s'arrêter lorsque "*maxLevel*" est atteint.

La commande "CONTINUOUS DOWN" doit régler *"targetLevel"* à *"minLevel"* et lancer la modification en utilisant la vitesse applicable. La modification de l'intensité lumineuse doit s'arrêter lorsque "*minLevel*" est atteint.

A l'exécution de "CONTINUOUS UP" ou "CONTINUOUS DOWN", au moins un pas doit être exécuté, sauf si cela est exclu par *"minLevel"* ou *"maxLevel"*.

Se reporter au 9.5.9 pour l'interruption d'une modification de l'intensité lumineuse avant d'atteindre *"minLevel"* ou *"maxLevel"*.

NOTE 1 Contrairement aux instructions "UP" et "DOWN", il n'est pas possible d'atteindre *"minLevel"* ou *"maxLevel"* avant la fin de la modification. Par conséquent, le bit *"fadeRunning"* est supprimé à la fin de la modification.

NOTE 2 Comme pour les instructions "UP" et "DOWN", différents appareillages de commande peuvent avoir des valeurs différentes pour *"targetLevel"*, "*actualLevel*" et *"lastLightLevel"* après arrêt préalable de la modification (par exemple par le biais de DAPC(MASK)).

9.5.7 Réponse du système à une modification de l'intensité lumineuse

Si *"fadeTime"*, *"extendedFadeTimeBase"*, *"extendedFadeTimeMultiplier"* et/ou *"fadeRate"* sont modifiées lors d'une modification de l'intensité lumineuse en cours, cette dernière doit alors s'achever sans recalculer la durée et/ou la vitesse de modification de l'intensité lumineuse. La modification de l'intensité lumineuse suivante doit utiliser les valeurs recalculées.

9.5.8 Réponse du système lors d'une modification de la veille et du démarrage

Si une modification de l'intensité lumineuse est lancée au cours de la veille, son processus doit se terminer par "*actualLevel*" égal à "*minLevel*" au cours de la phase de démarrage. La réaction aux commandes de niveau doit être identique à ce qu'elle serait si la ou les lampes fonctionnaient au *"minLevel"*.

Si une modification de l'intensité lumineuse est lancée lors du démarrage, son processus doit se terminer par "*actualLevel*". La réaction aux commandes de niveau doit être identique à ce qu'elle serait si la ou les lampes fonctionnaient à l'*"actualLevel"*.

La modification de l'intensité lumineuse doit commencer:

- dès que *"lampOn"* est TRUE, ou
- dans le cas d'une défaillance totale des lampes, dès que "*lampFailure*" est confirmée comme TRUE.

Pour de plus amples informations concernant *"lampOn"* et "*lampFailure*", voir 9.16.3 et 9.16.4.

9.5.9 Interruption d'une modification de l'intensité lumineuse

Toute commande qui règle une ou plusieurs des variables suivantes

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

ainsi que l'exécution de l'une des commandes suivantes

- "DAPC(MASK)", "IDENTIFY DEVICE"

doit interrompre une modification de l'intensité lumineuse en cours.

NOTE 1 La modification de l'intensité lumineuse s'interrompt même si la valeur de la variable concernée ne change pas.

Lorsqu'une modification de l'intensité lumineuse en cours est interrompue par un contrôleur d'application, la minuterie associée doit être arrêtée immédiatement. Après l'interruption de la minuterie, *"targetLevel"* doit être réglé sur *"actualLevel"* et la commande doit être exécutée (le cas échéant).

Lorsqu'une modification de l'intensité lumineuse en cours est interrompue alors qu'elle était en attente à l'état *"minLevel"* lors du démarrage, l'appareillage de commande doit achever le processus de démarrage.

NOTE 2 Ceci implique que, dans ce type de cas, *"targetLevel"* et *"actualLevel"* équivalent tous deux à *"minLevel"*.

9.6 Niveau min et max

La modification du niveau min ou max doit interrompre toute modification de l'intensité lumineuse en cours, avant la mémorisation du nouveau niveau min ou max.

"SET MIN LEVEL (*DTR0*)" doit définir *"minLevel"* selon la valeur *"DTR0"*:

- si 0 ≤ *"DTR0"* ≤ PHM: PHM;
- si *"DTR0"* ≥ *"maxLevel"* ou MASK: *"maxLevel";*
- dans tous les autres cas: *"DTR0"*.

Si *"actualLevel"* > 0 et *"actualLevel"* < *"minLevel"* à la suite du réglage d'un nouveau niveau min, *"targetLevel"* doit être recalculé à partir du nouveau *"minLevel"*. *"actualLevel"* doit être modifié en *"targetLevel"* immédiatement et le rendement lumineux doit être ajusté le plus rapidement possible. Ainsi, "*limitError*" doit être réglée sur TRUE.

"SET MAX LEVEL (*DTR0*)" doit définir *"maxLevel"* selon la valeur *"DTR0"*, comme suit:

- si *"minLevel"* ≥ *"DTR0"*: *"minLevel";*
- si *"DTR0"* = MASK: 0xFE;
- dans tous les autres cas: *"DTR0"*.

Si *"actualLevel"* > *"maxLevel"* à la suite du réglage d'un nouveau niveau max, *"targetLevel"* doit être recalculé à partir du nouveau *"maxLevel"*. *"actualLevel"* doit être modifié en *"targetLevel"* immédiatement et le rendement lumineux doit être ajusté le plus rapidement possible. Ainsi, "*limitError*" doit être réglée sur TRUE.

NOTE *"minLevel"* et *"maxLevel"* peuvent être utilisés pour compenser les différences de propriétés des appareillages de commande. Par exemple, si les appareillages de commande ont des valeurs différentes pour PHM, ils peuvent être réglés de manière à se comporter de façon similaire en ajustant *"minLevel"*.

9.7 Commandes

9.7.1 Généralités

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

- la commande est envoyée par adressage court, et l'adresse courte attribuée n'est pas égale à *"shortAddress"*;
- la commande est transmise par adressage de groupes, et le groupe attribué ne correspond à aucun des groupes identifiés par *"gearGroups"*;
- la commande est envoyée par adressage réservé;
- la commande est envoyée par adressage de diffusion non adressée, et *"shortAddress"* n'est pas la valeur MASK;
- la commande n'est pas définie (commande réservée, par exemple).

Les groupes de commande suivants peuvent être identifiés:

- instructions de niveau;
  - instructions de niveau sans modification de l'intensité lumineuse;
  - instructions de niveau qui déclenchent une modification de l'intensité lumineuse;
- instructions de configuration;
- requêtes;
- commandes spéciales;
  - instructions;
  - requêtes;
- commandes d'application étendues.

9.7.2 Instructions de niveau sans modification de l'intensité lumineuse

Les instructions de niveau sans modification de l'intensité lumineuse sont des instructions où le *"targetLevel"* doit être calculé; le passage d'*"actualLevel"* à *"targetLevel"* doit être immédiat et le rendement lumineux doit être ajusté le plus rapidement possible.

Ces commandes peuvent être réparties en trois catégories:

- commandes de niveau absolu
  - "OFF", "RECALL MIN LEVEL", "RECALL MAX LEVEL";
- commandes de niveau relatif
  - "STEP UP", "STEP DOWN", "ON AND STEP UP", "STEP DOWN AND OFF";
- commandes de configuration
  - "RESET", "SET MIN LEVEL (*DTR0*)", "SET MAX LEVEL (*DTR0*)".

9.7.3 Instructions de niveau qui déclenchent une modification de l'intensité lumineuse

Les instructions de niveau qui déclenchent une modification de l'intensité lumineuse sont des instructions où le *"targetLevel"* doit être calculé; *"actualLevel"* doit subir une modification de l'intensité lumineuse vers *"targetLevel"* en utilisant la durée ou la vitesse correspondante applicable. Si la durée de modification de l'intensité lumineuse est égale à 0 s, le passage d'*"actualLevel"* à *"targetLevel"* doit être immédiat et le rendement lumineux doit être ajusté le plus rapidement possible.

Ces commandes peuvent être réparties en deux catégories:

- instructions de niveau absolu qui utilisent la durée de modification de l'intensité lumineuse
  - "DAPC (*level*)", "GO TO SCENE (*sceneNumber*)", "GO TO LAST ACTIVE LEVEL";
- instructions de niveau relatif qui utilisent la vitesse de modification de l'intensité lumineuse
  - "UP", "DOWN";
  - "CONTINUOUS UP", "CONTINUOUS DOWN".

9.7.4 Instructions de configuration

Les instructions de configuration peuvent être utilisées pour modifier plusieurs propriétés d'appareillages de commande.

9.7.5 Requêtes

Les requêtes peuvent être utilisées pour demander la valeur de plusieurs propriétés d'appareillages de commande.

9.7.6 Commandes spéciales

Les commandes spéciales constituent un groupe de commandes non adressables. Tous les appareillages de commande doivent interpréter les commandes spéciales.

9.7.7 Commandes d'application étendues

Les commandes dont le code de fonctionnement se situe dans la plage de 0xE0 à 0xFF sont réservées aux types ou aux caractéristiques de dispositifs spéciaux. Chaque type ou caractéristique de dispositif redéfinit ces commandes, sauf la commande avec le code de fonctionnement 0xFF ("QUERY EXTENDED VERSION NUMBER"). Pour plus d'informations, voir 9.18.

9.8 Itérations de commandes

9.8.1 Généralités

Les exigences du 9.4 de l'IEC 62386-101:2022 s'appliquent avec les ajouts suivants.

9.8.2 Itération des commandes "UP" et "DOWN"

Les instructions "UP" et "DOWN" peuvent être transmises en tant qu'itération de commandes. A l'exécution de la première instruction de ce type d'itération, à moins que les valeurs de "*minLevel*" ou "*maxLevel*" ne l'excluent, un pas ("*targetLevel*" final = "*targetLevel*" calculé ± 1) doit être exécuté.

NOTE 1 Ceci assure un impact certain au début d'une itération.

Au terme de ce premier pas, la modification de l'intensité lumineuse d'une durée de 200 ms doit commencer à la vitesse correspondante applicable. Les pas ultérieurs doivent être exécutés à des intervalles déterminés par la vitesse de modification de l'intensité lumineuse applicable, tant que l'itération se poursuit. Chaque instruction "UP" ou "DOWN" exécutée comme partie intégrante de l'itération doit provoquer le redémarrage du processus de modification de l'intensité lumineuse d'une durée de 200 ms et un nouveau calcul de *"targetLevel"* en conséquence. La transition d'"*actualLevel*" doit se produire conformément au 9.5.1 et à la Figure 4, la première transition, en excluant le pas initial, se produisant à $1/(2 \cdot$ *"fadeRate"*$)$ après l'exécution de la première commande "UP" et "DOWN".

NOTE 2 Lorsque la vitesse de modification de l'intensité lumineuse change au cours d'une itération de commande, la nouvelle vitesse n'est pas appliquée lors de l'exécution de cette itération de commande.

La Figure 5 récapitule le comportement nécessaire. Les itérations commencent à Cmd 1 et s'achèvent à Time out.

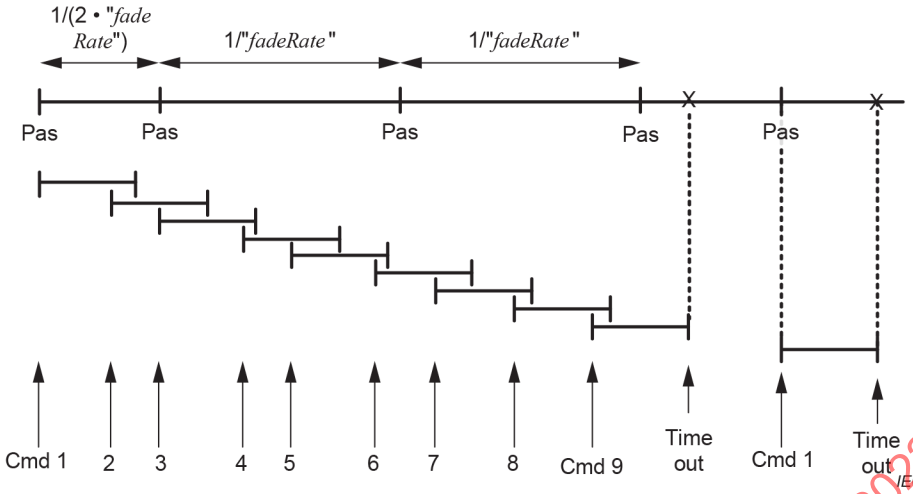


Figure 5 – Cadencement et réponse lors de l'exécution d'une itération de commande

9.8.3 DAPC SEQUENCE (déconseillé)

"ENABLE DAPC SEQUENCE" lance une itération de commande de contrôle direct de la puissance d'arc (DAPC) qui permet un contrôle dynamique du rendement lumineux.

À l'exécution de "ENABLE DAPC SEQUENCE", l'appareillage de commande doit utiliser provisoirement une durée de modification de l'intensité lumineuse de 200 ms ± 20 ms, tandis que l'itération de commande est active indépendamment de la durée de modification de l'intensité lumineuse réelle ou étendue. Lorsque la dernière modification de la séquence est achevée, les valeurs d'origine doivent être utilisées.

NOTE Étant donné que les variables de durée et de vitesse de modification de l'intensité lumineuse ne changent pas, la durée et la vitesse de modification peut être définie comme normale et/ou faire l'objet d'une requête en ce sens.

La séquence DAPC doit se terminer si, après 200 ms, l'appareillage de commande n'a pas accepté de commande "DAPC (*level*)". La séquence DAPC doit être abandonnée à l'exécution d'une commande de contrôle indirect de la puissance d'arc. "ENABLE DAPC SEQUENCE" acceptée lors d'une itération de commande DAPC doit être rejetée.

Lorsque la séquence DAPC est active, chaque exécution d'une commande "DAPC (*level*)" doit lancer une modification de l'intensité lumineuse d'une durée de 200 ms.

Si la combinaison d'appareillage de commande et de source de lumière ne peut pas exécuter une durée de modification de l'intensité lumineuse de 200 ms comme cela est exigé par la séquence DAPC, l'appareillage de commande doit ajuster le rendement lumineux le plus rapidement possible. Toutefois, l'appareillage de commande doit répondre comme si la modification de l'intensité lumineuse était achevée dans la durée demandée.

9.9 Modes de fonctionnement

9.9.1 Généralités

Différents modes de fonctionnement peuvent être sélectionnés au moyen de la commande "SET OPERATING MODE (*DTR0*)". L'*"operatingMode"* actuellement sélectionné peut faire l'objet d'une requête au moyen de "QUERY OPERATING MODE".

Les modes de fonctionnement 0x00 à 0x7F sont définis dans le présent document. Le mode de fonctionnement 0x00 au moins doit être disponible. Les modes de fonctionnement 0x80 à 0xFF sont spécifiques au fabricant. La requête "QUERY MANUFACTURER SPECIFIC MODE" peut être utilisée pour déterminer si le mode de fonctionnement de l'appareillage de commande est un mode normal défini dans l'IEC 62386 ou un mode spécifique au fabricant.

9.9.2 Mode de fonctionnement 0x00: mode normal

Lorsqu'un dispositif est en mode normal (*"operatingMode"*= 0x00), son comportement doit être tel qu'exigé par le présent document, jusqu'à ce qu'il soit réglé sur un mode de fonctionnement différent de 0x00.

9.9.3 Mode de fonctionnement 0x01 à 0x7F: réservé

Les modes de fonctionnement 0x01 à 0x7F sont réservés et ne doivent pas être utilisés.

9.9.4 Mode de fonctionnement 0x80 à 0xFF: modes spécifiques au fabricant

Il convient d'utiliser les modes spécifiques au fabricant uniquement si les caractéristiques exigées par l'application ne sont pas couvertes par la série IEC 62386. Lorsque le mode de fonctionnement d'un appareillage de commande est spécifique au fabricant, le comportement dudit appareillage peut également être spécifique au fabricant, avec les exceptions suivantes:

- en ce qui concerne l'accès au bus, l'appareillage de commande doit être conforme à l'IEC 62386-101:2022;
- l'appareillage de commande doit être conforme au présent document au moins tant que les commandes suivantes sont concernées:
  - "SET OPERATING MODE (*DTR0*)", "QUERY OPERATING MODE", et "QUERY MANUFACTURER SPECIFIC MODE";
  - toutes les commandes spéciales (voir 11.7) sauf WRITE MEMORY LOCATION (*DTR1, DTR0, data*), WRITE MEMORY LOCATION – NO REPLY (*DTR1, DTR0, data*) et PING.

Pour les commandes ci-dessus, les exigences du 7.2 s'appliquent.

Il est recommandé de continuer à suivre les commandes spécifiées dans toutes les parties mises en œuvre de la série IEC 62386 même dans le cas de modes spécifiques au fabricant.

9.10 Blocs de mémoire

9.10.1 Généralités

Les blocs de mémoire sont des espaces mémoire librement accessibles qui contiennent les informations ou les paramètres de configuration de l'appareillage de commande. Il n'est pas nécessaire de mettre en œuvre tous les blocs de mémoire consécutifs. De même, au sein d'un bloc de mémoire, il n'est pas nécessaire de mettre en œuvre tous les emplacements consécutifs. Tous les emplacements de blocs de mémoire mis en œuvre de tous les blocs de mémoire également mis en œuvre peuvent être lus au moyen de commandes d'accès de la mémoire. Une partie de la mémoire est en lecture seule et programmée par le fabricant de l'appareillage de commande ou mise à jour par l'appareillage de commande lui-même. Pour toutes les autres parties, l'accès en écriture au moyen de commandes d'accès de la mémoire peut être activé par le fabricant. A l'exception de l'emplacement 0x02, tous les emplacements inscriptibles d'un bloc de mémoire sont verrouillables, sauf spécification contraire dans la table des blocs de mémoire correspondante. Les emplacements dans les blocs de mémoire doivent être mis en œuvre en utilisant les types de mémoires indiqués dans le Tableau 8.

Tableau 8 – Types de mémoires

Type de mémoire	Accessibilité par le bus[a]	Volatilité[b]	Peut être modifiée de manière autonome par l'appareillage de commande pendant l'exécution	Description
ROM	RO	NV	Non	Concerne l'ensemble des valeurs fixes qui ne varient pas pendant l'exécution de l'appareillage de commande. NOTE La mémoire ROM est par nature en lecture seule, mais peut être modifiée lors de la programmation au cours de la production.
RAM-RO	RO	V	Oui	Pour les valeurs qui ne sont pas retenues sur un cycle de mise sous tension. Le contenu ne peut pas être modifié à partir de l'interface.
RAM-RW	RW	V	Oui	Pour les valeurs qui ne sont pas retenues sur un cycle de mise sous tension. Le contenu peut être modifié à partir de l'interface.
NVM-RO	RO	NV	Oui	Pour les valeurs qui sont retenues sur un cycle de mise sous tension. Le contenu ne peut pas être modifié à partir de l'interface.
NVM-RW	RW	NV	Oui	Pour les valeurs qui sont retenues sur un cycle de mise sous tension. Le contenu peut être modifié à partir de l'interface.

[a] RO: lecture seule (*Read-Only*). RW: lecture-écriture (*Read-Write*).

[b] V: volatile (non retenue sur un cycle de mise sous tension). NV: non volatile (retenue sur un cycle de mise sous tension).

L'espace mémoire adressable est limité à un espace maximal de 256 blocs de mémoire, chacun comportant au maximum 255 octets (environ 64 ko). Dans la mesure où le présent document spécifie comment mettre en œuvre les blocs de mémoire 0 et 1 (lorsqu'ils existent), et réserve les blocs de mémoire 200 à 255, cela laisse de l'espace pour 198 blocs de mémoire pour des besoins spécifiques au fabricant dans la plage de [2,199].

9.10.2 Carte de mémoire

Lorsqu'un bloc de mémoire spécifique au fabricant dans la plage de [2,199] est mis en œuvre, l'affectation de son contenu doit être conforme à la carte de mémoire fournie dans le Tableau 9.

Tableau 9 – Carte de mémoire de base des blocs de mémoire

Adresse	Description	Valeur par défaut (valeur d'origine)	Valeur réinitialisée[b]	Type de mémoire
0x00	Adresse du dernier emplacement de mémoire accessible	Rodage en usine, plage [0x03,0xFE]	Pas de modification	ROM
0x01	Octet indicateur[a]	[a]	[a]	Tout type[a]
0x02	Octet de verrouillage du bloc de mémoire. Les octets verrouillables dans le bloc de mémoire doivent être en lecture seule, tandis que l'octet de verrouillage a une valeur différente de 0x55.	0xFF	0xFF[c]	RAM-RW
[0x03,0xFE]	Contenu de bloc de mémoire[a]	[a]	[a]	Tout type[a]
0xFF	Réservée – non mise en œuvre	Réponse NO	Pas de modification	n.a.

[a] L'objet, les valeurs par défaut, les valeurs de mise sous tension, les valeurs réinitialisées et les types de mémoires de ces octets doivent être définis par le fabricant.

[b] Réinitialisation de la valeur après une commande "RESET MEMORY BANK".

[c] Egalement utilisée comme valeur de mise sous tension, sauf indication contraire explicite.

L'octet qui se trouve dans l'emplacement 0x00 de chaque bloc contient l'adresse du dernier emplacement de mémoire accessible du bloc. La valeur doit être comprise dans la plage [0x03,0xFE].

L'octet dans l'emplacement 0x01 est spécifique au fabricant. Lorsqu'il est mis en œuvre, il convient que le fabricant décrive l'utilisation de cet octet (ainsi que l'ensemble du contenu du bloc de mémoire).

NOTE 1 Il peut être utilisé, par exemple, pour mémoriser une somme de contrôle dans le cas d'un bloc de mémoire à contenu statique. Il n'est pas utile d'utiliser une somme de contrôle dans un bloc de mémoire dont le contenu est modifié par l'appareillage de commande.

L'octet de l'emplacement 0x02 doit être utilisé pour verrouiller l'accès en écriture. L'emplacement de mémoire 0x02 proprement dit ne doit jamais être verrouillé pour écriture. Alors que cet emplacement de mémoire contient toute valeur différente de 0x55, tous les emplacements de mémoire marqués "(verrouillable)" du bloc de mémoire correspondant doivent être en lecture seule. L'appareillage de commande ne doit pas modifier la valeur de l'octet de verrouillage autrement qu'à la suite du cycle de mise sous tension, de "RESET MEMORY BANK (*DTR0*)" ou de toute autre commande qui affecte l'octet de verrouillage.

L'emplacement 0xFF est un emplacement réservé dans chaque bloc de mémoire et n'est pas accessible. Cet emplacement ne doit pas être mis en œuvre comme emplacement de bloc de mémoire normal. Lorsqu'il est adressé, l'appareillage de commande doit répondre comme si cet emplacement n'était pas mis en œuvre, et il ne doit pas augmenter "*DTR0*".

NOTE 2 Cet emplacement est réservé pour interrompre l'augmentation automatique de "*DTR0*".

9.10.3 Sélection d'un emplacement de bloc de mémoire

Pour sélectionner un emplacement de bloc de mémoire, une combinaison de numéro et d'emplacement au sein du bloc de mémoire est exigée.

Le bloc de mémoire doit être sélectionné par un réglage du numéro de bloc de mémoire en *"DTR1"*. L'emplacement dans le bloc de mémoire doit être sélectionné par la valeur en *"DTR0"*.

9.10.4 Emplacements de mémoire protégeables

Les emplacements de blocs de mémoire marqués comme "protégeables" peuvent être protégés contre la lecture ou l'écriture ou être supprimés à l'aide d'une méthode spécifique au fabricant.

Les emplacements protégeables dont la protection contre la lecture est activée doivent répondre MASK à la commande "READ MEMORY LOCATION (*DTR1, DTR0*)". Dans ce cas, la mise en œuvre ou la mise à disposition de la caractéristique associée est facultative.

NOTE 1 La protection contre la lecture peut être activée par défaut. Si la protection contre la lecture est activée, cela signifie que la fonction associée peut ne pas être mise en œuvre et que le fabricant n'est pas obligé de fournir un mécanisme pour désactiver la protection contre la lecture.

Le contenu des emplacements protégeables dont la protection contre l'écriture est activée ne peut pas être écrasé.

NOTE 2 Cela signifie que la commande WRITE MEMORY LOCATION ne renvoie aucune réponse lors d'une tentative d'écriture dans un emplacement protégé contre l'écriture.

9.10.5 Lecture dans le bloc de mémoire

9.10.5.1 Généralités

Un emplacement de bloc de mémoire sélectionné peut être lu au moyen de la commande "READ MEMORY LOCATION (*DTR1, DTR0*)". La réponse doit être la valeur de l'octet à l'emplacement de bloc de mémoire adressé.

Lorsque le bloc de mémoire sélectionné n'est pas mis en œuvre, la commande doit être rejetée.

Lorsque l'emplacement de bloc de mémoire sélectionné se situe au-dessous de l'emplacement 0xFF, *"DTR0"* doit être augmenté de un, même si l'emplacement de mémoire n'est pas mis en œuvre. Dans le cas contraire, *"DTR0"* ne doit pas varier. Ce mécanisme permet une lecture consécutive facile des emplacements de blocs de mémoire.

Après lecture de plusieurs octets issus d'un bloc de mémoire, il convient que le contrôleur d'application vérifie la valeur de "*DTR0*" afin de s'assurer que son emplacement est celui attendu ou souhaité. Tout défaut d'adaptation indique une erreur de lecture.

9.10.5.2 Lecture de valeurs à plusieurs octets

Pour assurer la cohérence des données lors de la lecture d'une valeur à plusieurs octets dans un bloc de mémoire, un mécanisme doit être mis en œuvre afin de verrouiller l'ensemble des octets de la valeur à plusieurs octets lorsque le premier octet de cette valeur est lu, et de les déverrouiller lorsque le premier octet d'une autre valeur à un octet ou à plusieurs octets de tout bloc de mémoire de la même unité logique est lu.

La lecture d'un emplacement à plusieurs octets doit restituer la valeur stockée, mais pas les valeurs qui résident dans le tampon d'écriture utilisé pour la mise en tampon tant que la valeur complète n'a pas été écrite en mémoire (voir 9.10.6.3).

9.10.5.3 Emplacements non mis en œuvre

Lorsqu'un bloc de mémoire existe, que l'emplacement de bloc de mémoire sélectionné

- n'est pas mis en œuvre, et que MASK n'apparaît pas dans la plage de valeurs admise, ou
- se situe au-dessus du dernier emplacement de mémoire accessible,

la réponse à "READ MEMORY LOCATION (*DTR1, DTR0*)" doit être NO.