IEC 62386-105

Edition 2.0 2024-12
REDLINE VERSION

INTERNATIONAL STANDARD

colour inside

Digital addressable lighting interface –
Part 105: Particular requirements for control gear and control devices – Firmware transfer

IEC 62386-105:2024-12 RLV(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 It also gives information on projects, replaced and withdrawn publications.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, graphical symbols and the glossary. 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 22 500 terminological entries in English and French, with equivalent terms in 25 additional languages. Also known as the International Electrotechnical Vocabulary (IEV) online.

IEC 62386-105

Edition 2.0 2024-12
REDLINE VERSION

INTERNATIONAL STANDARD

colour inside

Digital addressable lighting interface – Part 105: Particular requirements for control gear and control devices – Firmware transfer

INTERNATIONAL ELECTROTECHNICAL COMMISSION

ICS 29.140.50, 29.140.99

ISBN 978-2-8327-0105-8

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

CONTENTS

INTERNATIONAL ELECTROTECHNICAL COMMISSION

_____

DIGITAL ADDRESSABLE LIGHTING INTERFACE –

Part 105: Particular requirements for control gear and control devices – Firmware transfer

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) IEC draws attention to the possibility that the implementation of this document may involve the use of (a) patent(s). IEC takes no position concerning the evidence, validity or applicability of any claimed patent rights in respect thereof. As of the date of publication of this document, IEC had not received notice of (a) patent(s), which may be required to implement this document. However, implementers are cautioned that this may not represent the latest information, which may be obtained from the patent database available at https://patents.iec.ch. IEC shall not be held responsible for identifying any or all such patent rights.

This redline version of the official IEC Standard allows the user to identify the changes made to the previous edition IEC 62386-105:2020. A vertical bar appears in the margin wherever a change has been made. Additions are in green text, deletions are in strikethrough red text.

IEC 62386-105 has been prepared by IEC technical committee 34: Lighting. It is an International Standard.

This second edition cancels and replaces the first edition published in 2020. This edition constitutes a technical revision.

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

a) several commands have been modified, renamed and added;
b) variables have been modified and added;
c) recommendations for implementation within emergency control gear have been added;
d) requirements for block acceptance have been changed;
e) example process-flow diagrams have been added;
f) requirements for restarting and power-on have been changed.

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

Draft	Report on voting
34/1258/FDIS	34/1281/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/publications.

This document is intended to be used in conjunction with:

- IEC 62386-101, which contains general requirements for system components;
- IEC 62386-102, which contains general requirements for the relevant product type (control gear), and with the appropriate parts of the IEC 62386-2xx series (particular requirements for control gear);
- IEC 62386-103, which contains general requirements for the relevant product type (control devices), and with the appropriate parts of the IEC 62386-3xx series (particular requirements for control devices);
- IEC 62386-104, which contains general requirements for wireless and alternative wired system components.

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, or
- revised.

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 a 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. IEC 62386-101 contains general requirements for system components, IEC 62386-102 extends this information with general requirements for control gear and IEC 62386-103 extends it further with general requirements for control devices. IEC 62386-104 and IEC 62386-105 can be applied to control gear or control devices. IEC 62386-104 gives requirements for wireless and alternative wired system components. IEC 62386-105 describes firmware transfer. IEC 62386-150 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 ~~first~~ second edition of IEC 62386-105 is intended to be used in conjunction with IEC 62386-101, IEC 62386-102 and the various parts that make up the IEC 62386-2xx series for control gear, together with IEC 62386-103 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 recognized.

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

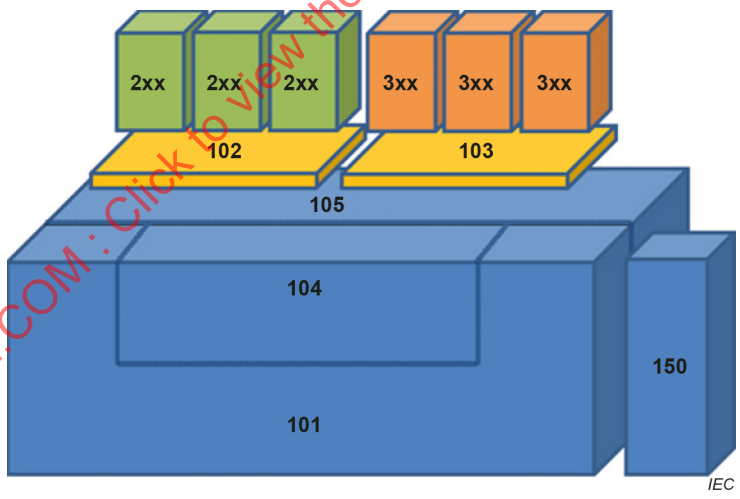


Figure 1 – IEC 62386 graphical overview

When this document refers to any of the clauses 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, "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".

Function or command parameters: *parameter name*.

DIGITAL ADDRESSABLE LIGHTING INTERFACE –

Part 105: Particular requirements for control gear and control devices – Firmware transfer

1 Scope

This part of IEC 62386 applies to control gear and control devices for control by digital signals of electronic lighting equipment.

Typically, a bus unit according to the IEC 62386 series contains firmware. There are circumstances where it ~~might~~ can be necessary to change the firmware after production or shipping of the product, for example if the bus unit does not operate as intended. In such a case, a firmware update of a bus unit via the interface is beneficial.

This firmware update process is primarily designed to be a bug fix process, not a feature extension process. Nevertheless, the firmware update process can be used for feature extensions. But it is important that the risk of negative effects to the complete system be considered in detail.

NOTE Annex D provides a "Firmware update management check sheet" to support risk estimation.

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 62386-102:~~2014~~2022, *Digital addressable lighting interface – Part 102: General requirements – Control gear*
~~IEC 62386-102:2014/AMD1:2018~~

IEC 62386-103:~~2014~~2022, *Digital addressable lighting interface – Part 103: General requirements – Control devices*
~~IEC 62386-103:2014/AMD1:2018~~

3 Terms and definitions

For the purposes of this document, the terms and definitions given in IEC 62386-101, IEC 62386-102, IEC 62386-103 and the following apply.

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

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

3.1
FW
firmware
software programmed into a control gear or control device

Note 1 to entry: Firmware can be changed during an update.

~~Note 1 to entry: This note applies to the French language only.~~

3.2
CRC
cyclic redundancy check
checksum used to prevent data corruption

Note 1 to entry: Annex B provides detailed information about CRC calculation.

3.3
block
unit of data containing information

Note 1 to entry: Information in a firmware update block usually contains firmware content.

3.4
programming
writing firmware transfer data to non-volatile memory (NVM)

3.5
normal operation
operation according to IEC 62386-102 or IEC 62386-103

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.

NOTE Systems with a single-master application controller are unlikely to operate correctly when other master control devices, such as ~~upgrade~~ firmware update tools, are connected.

~~**4.2 Transmitters and receivers in bus units**~~

~~The requirements of IEC 62386-101:2014 and IEC 62386-101:2014/AMD1:2018, 4.6.1 shall be extended as follows.~~

~~Bus units supporting firmware transfer shall be additionally capable of receiving 32 bit forward frames as specified in IEC 62386-101:2014, 7.4.3 (Reserved forward frame).~~

4.2 Logical units in a bus unit

If the firmware update process is started on a bus unit, all logical units inside the bus unit shall be affected. All variables defined in Table 5 shall be shared by all logical units of the bus unit. Commands addressed to one or more logical units within the bus unit shall be accepted by the bus unit according to the requirements of 9.2.

4.3 Updating control gear for emergency lighting

If IEC 62386-105 is implemented in control gear for emergency lighting, the product manual or data sheet should include guidance on the safety implications of a firmware update that should be considered.

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

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~~2022, Clause 7 apply, with the following additions.

7.2 32-bit forward frame encoding

The forward frame format used for firmware update consists of n = 32 data bits as described in IEC 62386-101:~~2014~~2022, 7.4.3 (~~Reserved~~ 32-bit forward frame).

For commands, the 32-bit forward frame ~~for 32 bit frames~~ shall be encoded as shown in Table 1.

Table 1 – 32-bit command frame encoding

Bytes/Bits											Device addressing method
Address byte								Opcode byte			
								1	2	3	
31	30	29	28	27	26	25	24[a]	23…16	15…8	7…0	
0	64 short addresses						x				Short addressing
1	0	1	1	1	1	0	1				Data transfer command[b]
1	1	0	0	1	0	1	1				Data transfer command[b]
1	1	1	1	1	1	0	x				Broadcast unaddressed
1	1	1	1	1	1	1	x				Broadcast
All other address byte values											Reserved

[a] ~~For bit 24~~ Where bit 24 is shown as "x", 0 indicates address space for control gear, 1 indicates address space for control devices.

[b] See Table 7 for data transfer 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, 9.8 (Dealing with frames and commands), 9.2 (Transactions) with the exception that the total duration may exceed 400 ms, 9.4 (Command iteration) and 9.6 (Use of multiple bus power supplies) apply.~~

The requirements of IEC 62386-101:2022, Clause 9 apply, with the exception that the total duration of a transaction may exceed 400 ms (IEC 62386-101:2022, 9.3).

9.2 Commands

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

- The command is sent using short address, broadcast addressing or broadcast unaddressed addressing, and bit 24 of the command frame does not match the type of bus unit (control gear or control device).
- The command is sent using short addressing and the given short address is not equal to its short address.
- The command is sent using reserved addressing.
- The command is sent using broadcast unaddressed addressing and the short address is not MASK.
- The command is not defined (e.g. reserved command).

The following command groups can be identified:

- standard commands;
  - instructions;
  - queries;
- data transfer commands.

9.3 Data transmission

A bus unit receives a new FW block by block. The first block (block 0) contains information about the type of bus unit (see Table 3), which receives a new FW. This avoids transferring the wrong FW to a bus unit if more than one bus unit is updated at a time.

~~NOTE Annex A provides detailed information about the update file.~~

~~The opcode byte 1 shall be 0xFB for 32-bit standard commands (see Table 6). If the opcode byte 1 in a standard command is not equal to 0xFB, the bus unit shall not accept the standard command.~~

The update file shall contain release notes as described in Annex A.

9.4 Duration of firmware update

A data transmission frame consists of a start bit, a 32-bit data ~~bits~~ transfer command and a stop condition, which occupies the bus for around 30 ms. With a settling time of less than 15 ms ~~(maximum frame priority)~~ (priority 1 settling time), the transmission of three bytes takes less than 45 ms. For an update of 64 ~~kByte~~ kB it is expected to take less than 20 min.

9.5 Security

This document specifies the use of CRC checksums to help ensure error-free transfer of data. In addition, it is recommended that the individual manufacturer ensures firmware image integrity and authenticity, for example by making use of the device key.

9.6 Firmware update features

Each bus unit shall expose its firmware update features as a combination of device properties as given in Table 2.

Table 2 – Firmware update features

Bit	Description	Value	See
0	*"fwUpdateCancelSupported"* is TRUE?	"1" = "YES"	~~XXX~~ 9.7.2.3
1	Integrated bus power supply is present and is disabled during firmware update	"1" = "YES" "0" = bus power supply state is unchanged during firmware update, or is not present	
~~1~~ 2 to 7	Reserved – Not implemented	"0" = "NO"	

The bus unit firmware update features can be queried using QUERY FW UPDATE FEATURES.

~~If the bus unit supports cancellation of the firmware update process (see 9.6.2.3), "*fwUpdateCancelSupported*" is set to TRUE.~~

Bit 1: "1" shall indicate that an integrated bus power supply is present and is automatically disabled during the firmware update process. "0" shall indicate that either there is no integrated bus power supply, or if there is an integrated bus power supply then its state shall not be automatically changed when starting, during, or at the end of the firmware update. To help prevent loss of bus power during a firmware update, a lock-out mechanism is described in 11.3.2.

NOTE For bus units with bit 1 = "1", using a firmware update tool that is unable to apply suitable bus power can result in a system with no bus power and so no communications are possible. In such cases, a bus power supply can be temporarily connected to the system to allow the firmware update process to continue.

9.7 Update process

9.7.1 Start firmware update

~~A bus unit shall enable the firmware update process by the acceptance of the command START FW TRANSFER. Several bus units can be addressed in this way to update more than one bus unit at a time.~~

A bus unit shall enable the firmware update process whenever *"fwUpdateProcessEnabled"* is TRUE. This can be due to execution of the command START FW TRANSFER, or after a power cycle where the previous update failed (see Table 5). Several bus units can be addressed in this way to update more than one bus unit at a time.

NOTE 1 Annex C provides an example of the firmware update process.

~~It is recommended not to trigger erasing of the memory before block 0 is verified.~~

The bus unit shall be capable of returning to normal operation until at least a block 0 has been successfully verified.

NOTE 2 This means that bus units with a single flash image of the firmware used for normal operation cannot erase that firmware until at least block 0 has been verified.

Whilst *"fwUpdateProcessEnabled"* is TRUE, the operation of the bus unit is manufacturer-specific except for the requirements given in this document.

NOTE 3 This includes, for example, the reaction to commands of other parts of the IEC 62386 series.

9.7.2 Data transfer

9.7.2.1 Block 0 (information block)

Block 0 contains all data for the bus unit to decide if it will accept the new firmware or not.

The global trade item number (GTIN), the hardware version number and the firmware version number contained in the bus unit are described in IEC 62386-102:~~2014 and IEC 62386-102:2014/AMD1:2018~~2022, 9.10.~~6~~7 (Memory bank 0 for control gear) and IEC 62386-103:~~2014 and IEC 62386-103:2014/AMD1:2018~~2022,~~ 9.10.6~~9.11.7 (Memory bank 0 for control devices). ~~A bus unit shall have a maximum of one family GTIN which shall be shown in the documentation of the bus unit.~~

If *"currentBlock"* is 0, then upon reception of a complete block 0, the following information ~~is~~ shall be checked, where Table 3 shows block 0 content:

- The received block 0 "Size of block" is equal to the value shown in Table 3.
- The received block 0 "Session key" is not MASK or 0.
- The received block 0 "Block number" is equal to the value shown in Table 3.
- The received block 0 "Block 0 version" is equal to the value shown in Table 3.
- The received block 0 "GTIN" matches the GTIN stored in memory bank 0.
- (Received block 0 "FW version min") ≤ (memory bank 0, Firmware version) ≤ (received block 0 "FW version max").
- (Received block 0 "HW version min") ≤ (memory bank 0, Hardware version) ≤ (received block 0 "HW version max").
- (Received block 0 "Identification number min") ≤ (memory bank 0, Identification number) ≤ (received block 0 "Identification number max").
- The received block 0 "Device key" meets manufacturer-specific requirements.
- The received block 0 "CRC" matches the calculated value based on the block 0 content; Annex B is applicable).

If the above check is successful, the following operation shall result:

- *"sessionKey"* shall be set to the received block 0 "Session key",
- ~~"*currentBlock*" shall be set to 1,~~
- ~~"*currentBlockByte*" shall be set to 0,~~
- previously received block data that is unwritten may be discarded.

Otherwise, the following operation shall result:

- ~~"*fwUpdateProcessEnabled*" shall be set to FALSE and resume normal operation if possible.~~
- The block 0 content shall be discarded.

NOTE In the case where block 0 content is discarded, *"fwUpdateProcessEnabled"* remains TRUE, so the bus unit will continue looking for valid block 0 data. This enables multiple devices to be updated, even where these devices use identical firmware but have different GTINs, due to the possibility of an update tool sending multiple block 0s.

~~**Table 3 – Block 0 definitions**~~

~~Address (hex)~~	~~Size (bytes)~~	~~Description~~
~~00~~	~~1~~	~~Size of block (fixed value of 0x3D for block 0)~~
~~01~~	~~1~~	~~Block 0 version (always 0x00)~~
~~02..04~~	~~3~~	~~Total block count (MSB first) [a]~~
~~05..0A~~	~~6~~	~~GTIN (MSB first)~~
~~0B..0C~~	~~2~~	~~HW version min (MSB first)~~
~~0D..0E~~	~~2~~	~~HW version max (MSB first)~~
~~0F..10~~	~~2~~	~~FW version min (MSB first)~~
~~11..12~~	~~2~~	~~FW version max (MSB first)~~
~~13.. 1A~~	~~8~~	~~Identification number min (MSB first)~~
~~1B..22~~	~~8~~	~~Identification number max (MSB first)~~
~~23..2A~~	~~8~~	~~Session key [b]~~
~~2B..3A~~	~~16~~	~~Device key [c]~~
~~3B..3C~~	~~2~~	~~CRC (MSB first)~~

~~[a] This is the amount of blocks being transferred during the firmware update.~~

~~[b] The session key is generated by the bus unit, which transfers the firmware update.~~

~~[c] The device key and its use is manufacturer-specific. It allows the manufacturer to specify different areas/options in his firmware.~~

Offset (hex)	Size (bytes)	Description
00..01	2	Size of block 0 (MSB first, fixed value of 0x0041 for block 0)
02..09	8	Session key[a]
0A..0C	3	Block number (always 0x000000 for block 0)
0D	1	Block 0 version (always 0x01)
0E..10	3	Total block count (MSB first)[b]
11..16	6	GTIN (MSB first)
17..18	2	HW version min (MSB first)
19..1A	2	HW version max (MSB first)
1B..1C	2	FW version min (MSB first)
1D..1E	2	FW version max (MSB first)
1F..26	8	Identification number min (MSB first)
27..2E	8	Identification number max (MSB first)
2F..3E	16	Device key[c]
3F..40	2	CRC (MSB first)

Key

MSB Most significant byte

[a] The session key is generated by the FW update tool which transfers the firmware update.

[b] This is the number of blocks, excluding block 0, being transferred during the firmware update.

[c] The device key and its use is manufacturer-specific. It allows the manufacturer to specify different areas or options in their firmware. The device key shall be independent of the session key, allowing a firmware update tool to change the session key without requiring a change to the device key.

It is recommended to calculate the CRC checksums with the incoming bytes to minimize delay at the end of the block reception.

9.7.2.2 Block 1..*n* (data block)

~~A bus unit shall only accept a block, if the following condition is true:~~

- ~~"*fwUpdateProcessEnabled*" is TRUE, and~~
- ~~The session key matches the received block 0 "Session key".~~

~~After reception of a whole block, the consistency of the firmware data inside the block shall be verified by CRC (address 0D...0E). The data consistency of the whole block shall be verified by a second CRC (address s+0F...s+10). If the verification fails, the bus unit shall discard the block. If a block is valid, but is the same as the last programmed block, it is recommended to discard the block to prevent unnecessary write cycles.~~

Table 4 shows the content of a data block. A bus unit shall only accept a data block if the following conditions are true:

- *"fwUpdateProcessEnabled"* is TRUE, and
- the session key matches *"sessionKey"*, and is not MASK or 0, and
- the block number matches *"currentBlock"*, and
- *"currentBlock"* is not 0.

After reception of a whole block, the consistency of the firmware data inside the block shall be verified by CRC (offset 0x0D...0x0E). The data consistency of the whole block shall be verified by a second CRC (offset s + 0x0F…s + 0x10). The block may optionally be discarded if "size of whole block" is an unexpected size, as determined by the manufacturer. If the verification fails, the bus unit shall discard the block. If a block is valid, but is the same as the last programmed block, it is recommended to discard the block to prevent unnecessary write cycles.

Table 4 – Block 1..*n* definitions

~~Address (hex)~~	~~Size (bytes)~~	~~Description~~
~~00..01~~	~~2~~	~~s = Size of block data bytes (MSB first)~~
~~02..09~~	~~8~~	~~Session key~~
~~0A..0C~~	~~3~~	~~Block number (MSB first)~~
~~0D..0E~~	~~2~~	~~CRC (of block data bytes)~~
~~0F..(s+0E)~~	~~s~~	~~Firmware data (optionally encrypted by manufacturer)~~
~~s+0F..s+10~~	~~2~~	~~CRC (of total block)~~
~~NOTE This allows a theoretical maximum of 65 535 bytes of firmware data per block, resulting in a maximum total of 65 552 bytes in the block.~~		

Offset (hex)	Size (bytes)	Description
00..01	2	Size of whole block (MSB first)[a] [b]
02..09	8	Session key
0A..0C	3	Block number (MSB first)
0D..0E	2	CRC (of block firmware data bytes)
0F..(s + 0E)	s	Firmware data bytes (optionally encrypted by manufacturer)
(s + 0F)..(s + 10)	2	CRC (of whole block)

Key

MSB Most significant byte

[a] This allows a theoretical maximum block size of 65 535 bytes, resulting in a maximum firmware data size of 65 518 bytes.

[b] Whole block size = s + 17.

It is recommended to calculate the CRC checksums with the incoming bytes to minimize delay at the end of the block reception.

9.7.2.3 Cancel firmware update

~~If a bus unit physically supports cancellation of FW updates and can return to the previous FW, the bus unit shall respond to QUERY FW UPDATE FEATURES with 00000001b.~~

~~If a bus unit supports cancellation of the FW updates, the bus unit shall set "*fwUpdateProcessEnabled*" to FALSE by the acceptance of the command CANCEL FW UPDATE.~~

~~If a bus unit physically does not support a cancellation of the FW update process, it shall ignore this command.~~

If the bus unit supports cancellation of the firmware update process even when *"sessionKey"* is not MASK, *"fwUpdateCancelSupported"* shall be TRUE.

In this case, the command CANCEL FW UPDATE can be used to cancel the firmware update at any time and return the bus unit to normal operation.

9.7.3 Persistent variables during firmware update

A firmware update may ~~totally~~ change the internal ~~structure~~ memory arrangement and content of the corresponding bus unit, except for the following cases:

The values for

- the GTIN,
- the identification number,
- the hardware version,

shall not be affected by a firmware update.

~~If a bus unit operates in the standard mode described as operating mode 0x00 in IEC 62386-102:2014 and IEC 62386-102:2014/AMD1:2018, 9.10 for control gear and in IEC 62386-103:2014 and IEC 62386-103:2014/AMD1:2018, 9.10.5 for control devices, each NVM variable shall either remain unchanged or be set to factory default as a result of a firmware update. the short addresses of all logical units shall be maintained at least until the firmware update successfully completes. For a bus unit in an operating mode different from 0x00, the variables do not need to be in factory default state.~~

If firmware update features bit 1 is "0", the enabled or disabled state of an integrated bus power supply shall not be changed by a firmware update (see 9.6).

For the purposes of this Subclause 9.7.3, a "structural change" in the bus unit is when any of the following are changed by the firmware update:

- number of logical units, or
- implemented device types, or
- quantity or types of implemented input device instances, or
- implemented feature types.

NOTE 1 A change in "current bus unit configuration" in memory bank 0 can cause a change of these "structural change" parameters.

The short addresses of all logical units shall not change as a result of a firmware update with a new firmware image until a restart with the new firmware image is carried out. If the restart with the new firmware image results in a structural change of the bus unit, then the short addresses of all logical units shall be set to MASK and it is recommended that all NVM variables are set to their factory defaults. If instead, the restart with a new firmware image results in no structural change of the bus unit, the short addresses of all logical units shall remain unchanged and it is recommended that all NVM variables are unchanged.

The manufacturer shall provide a document stipulating which variables are affected by the update and whether a re-commissioning of the system is necessary after the firmware update process.

After a firmware update, an updated bus unit should first proceed to a power-up sequence to (re-)load RAM variables.

~~It is recommended that all NVM variables remain unchanged as a result of the firmware update.~~

NOTE 2 Owing to the fact that the programme memory of a bus unit is updated, values in other parts of the IEC 62386 series marked as ROM can be changed.

9.7.4 Firmware version number

It is ~~allowed~~ permitted to transfer the same firmware to a bus unit multiple times using the firmware transfer procedure described in this document.

It is strongly recommended that two firmware update files containing different firmware do not contain the same GTIN number and the same firmware version number.

9.7.5 Firmware update in a system

~~If "*fwUpdateProcessEnabled*" is TRUE, it is permitted that the bus unit discards some or all commands not described in this document (see 11.3.1).~~

While *"fwUpdateProcessEnabled"* is TRUE, the bus unit may discard some or all commands not described in this document. Execution or discarding of such commands shall not affect the firmware update process as described in this document.

While *"fwUpdateProcessEnabled"* is TRUE, the bus unit shall not send any forward frames.

It is recommended to run the firmware update under human observation to be able to react to any errors that occur.

It is recommended to avoid other communication on the bus while a firmware update process is running.

NOTE The ~~device executing the firmware~~ FW update tool can use the quiescent mode defined in IEC 62386-103:~~2014 and IEC 62386-103:2014/AMD1:2018~~2022, 9.10.4, to suppress forward frames from other bus units.

9.7.6 Error recovery

~~If a bus unit is prevented from completing the firmware update by a temporary event such as power failure or communications interruption, it shall be possible to re-establish and complete the firmware update process over the bus.~~

The requirements in this document ensure that an interrupted or incomplete firmware update process can be restarted or resumed until successfully completed. Such interruptions can be caused by power failure or communications interruption.

EXAMPLE An example implementation would be a "bootloader". The bootloader is a smaller partitioned piece of firmware that is capable of communicating over the bus and implementing the download and programming of a new firmware. If the bootloader does not complete the firmware update it will retain control until the process is re-established and completed. A common method is to start again from the beginning.

NOTE If a firmware update process is started, some or all ~~other~~ commands not specified in this document can be stopped from execution.

9.8 Power-on

If an external power cycle occurs and *"fwUpdateRestartEnabled"* is TRUE, then normal operation implementing the changes made by the most recently transferred firmware image shall be executed immediately after the power cycle.

If an external power cycle interrupts a firmware update, the operation proceeds as follows:

- If *"fwUpdateCancelSupported"* is TRUE: normal operation shall result after a power-cycle.
- If *"fwUpdateCancelSupported"* is FALSE: the power on values shall be set in accordance with footnote "a" of Table 5.

In all other cases, normal operation shall resume after the power cycle.

10 Declaration of variables

~~Additional~~ Variables are given in Table 5.

The variables in other implemented parts of the IEC 62386 series shall also exist at least whenever *"fwUpdateProcessEnabled"* is FALSE.

Table 5 – Declaration of ~~additional~~ variables

Variable	Default value (factory)	Reset value	Power-on value	Range of validity	Memory type
"fwUpdateProcessEnabled"	FALSE	no change	FALSE[a]	[TRUE, FALSE]	RAM
"currentBlock"	0	no change	0	[0, 0xFFFFFF]	RAM
"currentBlockByte"	0	no change	0	[0, 0xFFFF]	RAM
"blockIncomplete"	FALSE	no change	FALSE	[TRUE, FALSE]	RAM
"sessionKey"	~~0~~ MASK	no change	~~0~~ MASK[a]	[0, 0xFF FF FF FF FF FF FF FF]	RAM
"fwUpdateRestartEnabled"	FALSE	no change	FALSE	[TRUE, FALSE]	RAM
"fwUpdateCancelSupported"	[b]	no change	[b]	[TRUE, FALSE]	ROM

[a] If a previous update process has failed, ~~this value~~ *"fwUpdateProcessEnabled"* can still be set to TRUE after a power cycle, in which case the power-on value of *"sessionKey"* shall be 0, indicating that a previously interrupted update attempt can only be restarted from block 0.

[b] (Default) value of this bit shall be defined by the manufacturer.

11 Definition of commands

11.1 General

Standard commands are either control instructions defined in 11.3, or queries defined in 11.4. Data transfer commands are defined in 11.5.

Unused opcodes are reserved for future needs.

11.2 Overview sheets

Table 6 gives an overview of the standard commands for bus units having implemented this document.

Table 6 – Standard commands ~~for bus units with firmware update capability~~

Command name	Address byte	Opcode byte			Answer	Send twice	References	Command reference
	See 7.2							
		1	2	3				
START FW TRANSFER	*Device*[a]	0xFB	0x00	0x00	✓	✓	9.7.1	11.3.2
RESTART FW	*Device*[a]	0xFB	0x01	0x00	✓	✓		11.3.3
ENABLE RESTART	*Device*[a]	0xFB	0x02	0x00		✓		11.3.4
FINISH FW UPDATE	~~*0xBF*~~*Device*[a]	0xFB	0x03	0x00	✓	✓		11.3.5
CANCEL FW UPDATE	~~*0xBF*~~*Device*[a]	0xFB	0x04	0x00		✓	9.7.2.3	11.3.6
QUERY FW UPDATE FEATURES	*Device*[a]	0xFB	0x05	0x00	✓		9.6	11.4.1
QUERY FW RESTART ENABLED	*Device*[a]	0xFB	0x06	0x00	✓			11.4.2
~~QUERY FW UPDATE RUNNING~~	~~*0xBF*~~	~~0xFB~~	~~0x07~~	~~0x00~~	~~✓~~			~~11.3.2.4~~
~~QUERY BLOCK FAULT~~	~~*0xBF*~~	~~0xFB~~	~~0x08~~	~~0x00~~	~~✓~~			~~11.3.2.5~~
QUERY FW UPDATE RECEIVER READY	*Device*[a]	0xFB	0x07	0x00	✓			11.4.3
QUERY BLOCK INCOMPLETE OR FAULT	*Device*[a]	0xFB	0x08	0x00	✓			11.4.4
QUERY FW TRANSFER VERSION	*Device*[a]	0xFB	0x09	0x00	✓		9.7.4	11.4.5
QUERY BLOCK 0 ACCEPTED	*Device*[a]	0xFB	0x0A	0x00	✓		9.7.2.1	11.4.6
[a] *Device* means Short addressing, Broadcast or Broadcast unaddressed from Table 1.								

Table 7 gives an overview of the data transfer commands for bus units having implemented this document.

Table 7 – Data transfer commands ~~for bus units with firmware update capability~~

Command name	Address byte	Opcode byte			Answer	Send twice	References	Command reference
		1	2	3				
BEGIN BLOCK (*data h, data m, data l*)	0xCB	data h	data m	data l		✓	9.7.2	11.5.1
TRANSFER BLOCK DATA (*data h, data m, data l*)	0xBD	data h	data m	data l			9.7.2	11.5.3

~~**11.3 Commands**~~

~~**11.3.1 General**~~

~~The following five commands shall be discarded while "*fwUpdateProcessEnabled*" is TRUE:~~

- ~~START FW TRANSFER~~
- ~~ENABLE RESTART~~
- ~~QUERY FW RESTART ENABLED~~
- ~~RESTART FW~~
- ~~QUERY FW UPDATE FEATURES~~

All other commands in 11.3 shall be discarded if "*fwUpdateProcessEnabled*" is FALSE.

11.3.2 Standard commands

11.3.2.1 START FW TRANSFER

The bus unit shall set "*fwUpdateProcessEnabled*" to TRUE and respond with "YES".

The bus unit is ready to receive the new firmware after a waiting time of 500 ms.

11.3.2.2 FINISH FW UPDATE

If the following condition is true:

- all data blocks (see Table 4) for the firmware update have been successfully received, and
- a manufacturer specific condition for completing the firmware update is TRUE.

then "*fwUpdateProcessEnabled*" shall be set to FALSE and "*fwUpdateRestartEnabled*" shall be set to TRUE, the reply shall be "NO" and the bus unit may optionally continue the operation described for RESTART FW (11.3.2.8) except for the reply specified for that command.

NOTE An example of a case when FINISH FW UPDATE will also cause the device to re-start, is for devices that only have space for a single firmware image that is destroyed when the block data transfer starts. In such cases, the device is executing a bootloader until the firmware update finishes and the device is restarted.

Blocks do not need to be received consecutively. Gaps are permitted between blocks. Use of total block count from block 0 is manufacturer-specific.

If the above condition is FALSE, then the answer shall be "YES" and "*fwUpdateProcessEnabled*" shall be kept TRUE.

11.3.2.3 QUERY FW UPDATE FEATURES

The bus unit shall answer with the firmware update feature set, which is formed by a combination of bus unit features.

Refer to 9.5 for further information.

11.3.2.4 QUERY FW UPDATE RUNNING

The bus unit shall respond with "YES", if the variable "*fwUpdateProcessEnabled*" is TRUE.

11.3.2.5 QUERY BLOCK FAULT

The bus unit shall respond with "NO", if the following condition is true:

- *"currentBlockByte"* matches the block size in address 0 of this block, and
- both CRC checksums of the block addressed by *"currentBlock"* match the corresponding CRC checksums of this block.

If the above condition is FALSE, then the answer shall be "YES".

11.3.2.6 CANCEL FW UPDATE

If the bus unit supports cancellation of the firmware update, the bus unit executing the command CANCEL FW UPDATE shall set the variable "*fwUpdateProcessEnabled*" to FALSE and discard all transferred blocks. Then the bus unit shall run through a power up sequence, executing the previous firmware.

~~**11.3.2.7 ENABLE RESTART**~~

~~The bus unit shall set "*fwUpdateRestartEnabled*" to TRUE.~~

~~**11.3.2.8 RESTART FW**~~

~~The bus unit shall restart firmware execution.~~

~~If "*fwUpdateRestartEnabled*" is FALSE, this command shall be discarded.~~

~~After execution of the command RESTART FW, "*fwUpdateRestartEnabled*" shall be set to FALSE.~~

~~NOTE 1 Depending on the state of the firmware image, this could mean that the device restarts the same firmware previously executed, or a different firmware than was previously executed, or a firmware image that can Implement only the commands described in this document (bootloader).~~

~~If the firmware that will be executed by this command does not meet the requirements specified in IEC 62386-102 or IEC 62386-103, then the reply shall be "YES", otherwise the reply shall be "NO".~~

~~NOTE 2 An example where a "YES" reply is sent is when a "bootloader" image is restarted. This could occur if the device does not contain an alternative valid firmware image.~~

~~The bus unit shall be ready to receive frames latest 30 s after the execution of this command.~~

~~**11.3.2.9 QUERY FW RESTART ENABLED**~~

~~The bus unit shall respond with "YES", if the variable "*fwUpdateRestartEnabled*" is TRUE.~~

11.3 Control instructions

11.3.1 General

Control instructions are used to change the configuration or the mode of operation of the bus unit. A control instruction marked as "send twice" in Table 6 shall be discarded unless it is accepted twice according to the requirements stated in IEC 62386-101:2022, 9.4. Some control instructions provide an answer, as shown in Table 6.

11.3.2 START FW TRANSFER

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

If bit 1 of the firmware update features described in 9.6 is "1", and the integrated bus power supply is enabled, then this command shall be discarded.

NOTE 1 This provides a lock-out mechanism to help prevent the loss of bus power when performing a firmware update on a bus unit that contains a switchable integrated bus power supply. When updating bus units that do not implement IEC 62386-250, but do have bit 1 of the firmware update features described in 9.6 equal to "1", a firmware update tool can prompt the user to apply the manufacturer-specific method to disable the integrated bus power supply before continuing with the firmware update.

In some manufacturer-specific conditions, this command can be discarded. Such conditions shall be stated in the product manual or datasheet.

NOTE 2 Some examples of manufacturer-specific conditions causing this command to be discarded are as follows: a self-contained emergency control gear whilst operating from the battery; a centrally-supplied emergency control gear whilst the emergency condition is true; control gear failure; application controller failure; or input device failure.

If the command is not discarded, execution shall continue as follows:

The bus unit shall set *"fwUpdateProcessEnabled"* to TRUE, set *"sessionKey"* to MASK and answer with "YES".

The bus unit shall be ready to receive the new firmware after a maximum time of 500 ms.

11.3.3 RESTART FW

If *"fwUpdateProcessEnabled"* is TRUE, or if *"fwUpdateRestartEnabled"* is FALSE, then this command shall be discarded, otherwise execution shall continue as follows:

"fwUpdateRestartEnabled" shall be set to FALSE.

This instruction is intended to restart firmware execution.

If the firmware that will be executed by this command does not meet the requirements specified in IEC 62386-102 or IEC 62386-103, then the reply shall be "YES", otherwise the reply shall be "NO".

NOTE An example where a "YES" reply is sent is when a "bootloader" image is restarted. This can occur if the device does not contain an alternative valid firmware image. Leaving the device in this state without completing a firmware update results in the device not having functionality according to IEC 62386-102 or IEC 62386-103.

The bus unit shall then restart firmware execution and shall be ready to receive frames within 120 s from the execution of this command.

Depending on the state of the firmware image, the device can restart the same firmware as previously executed, or a different firmware than was previously executed (for example, the newly transferred image), or a firmware that implements only the commands described in this document (bootloader).

11.3.4 ENABLE RESTART

If *"fwUpdateProcessEnabled"* is TRUE, this command shall be discarded, otherwise execution shall continue as follows:

The bus unit shall set *"fwUpdateRestartEnabled"* to TRUE.

11.3.5 FINISH FW UPDATE

If *"fwUpdateProcessEnabled"* is FALSE, this command shall be discarded, otherwise execution shall continue as follows:

If the following conditions are all true:

- *"currentBlock"* is equal to the total block count from block 0 (see Table 3), and
- *"currentBlockByte"* matches the block size in address 0 and 1 of this last received block, and
- an optional manufacturer-specific condition for completing the firmware update is TRUE,

then *"fwUpdateProcessEnabled"* shall be set to FALSE and *"fwUpdateRestartEnabled"* shall be set to TRUE, the reply shall be "NO" and the bus unit may optionally continue the operation described for RESTART FW except for the reply specified for that command.

NOTE An example of a case when FINISH FW UPDATE will also cause the device to restart, is for devices that only have space for a single firmware image that is destroyed when the block data transfer starts. In such cases, the device executes a bootloader until the firmware update finishes and the device is restarted.

If any of the conditions in the above bulleted list are FALSE, then the answer shall be "YES".

11.3.6 CANCEL FW UPDATE

If *"fwUpdateProcessEnabled"* is FALSE, this command shall be discarded, otherwise execution shall continue as follows:

"currentBlock" and *"currentBlockByte"* shall both be set to 0.

If *"sessionKey"* is MASK, or *"fwUpdateCancelSupported"* is TRUE, the bus unit shall:

- set the variable *"fwUpdateProcessEnabled"* to FALSE, and
- discard all transferred blocks, and
- then return to normal operation, which can be achieved by running through a power-up sequence and executing the previous firmware.

NOTE If *"fwUpdateProcessEnabled"* remains TRUE, the firmware update process can only proceed if block 0 is then successfully resent.

11.4 Queries

11.4.1 QUERY FW UPDATE FEATURES

If *"fwUpdateProcessEnabled"* is TRUE, this command shall be discarded, otherwise execution shall continue as follows:

The bus unit shall answer with the firmware update feature set, which is formed by a combination of bus unit features.

Refer to 9.6 for further information.

11.4.2 QUERY FW RESTART ENABLED

If *"fwUpdateProcessEnabled"* is TRUE, this command shall be discarded, otherwise execution shall continue as follows:

The bus unit shall answer with "YES", if the variable "*fwUpdateRestartEnabled*" is TRUE.

11.4.3 QUERY FW UPDATE RECEIVER READY

If "*fwUpdateProcessEnabled*" is FALSE, this command shall be discarded, otherwise execution shall continue as follows:

The bus unit shall answer with "YES", if the variable "*fwUpdateProcessEnabled*" is TRUE and the device is ready to receive data, otherwise the reply shall be "NO". The status "ready to receive data" includes the case when the last block has been transferred and written to memory.

NOTE Bus units will possibly need some time to erase flash or write firmware data after receiving each complete block. An FW update tool can use this query to determine when a bus unit is ready to continue. The query can also be used to determine if the firmware update process has been started and not completed in any bus units.

11.4.4 QUERY BLOCK INCOMPLETE OR FAULT

If *"fwUpdateProcessEnabled"* is FALSE, this command shall be discarded, otherwise execution shall continue as follows:

The bus unit shall answer with "NO", if *"blockIncomplete"* is FALSE, otherwise the answer shall be:

- if *"shortAddress"* of all logical units is MASK: "YES",

- in all other cases: the *"shortAddress"* of the first addressed logical unit (starting from logical unit index 0), 0AAAAAA1b.

NOTE Sending this query when an incomplete block has been transferred, will cause a reply, which does not necessarily indicate a fault, but indicates that the block transfer is incomplete. To check for successful block transfer, update tools can send this query at the end of each complete block transfer, for blocks 1 or greater.

11.4.5 QUERY FW TRANSFER VERSION

The answer to this command shall be 1.

NOTE A firmware update tool can use this version number to determine an acceptable format for block data. This version number will be increased in a future update to this document if the block data format is changed and can be increased if other changes are made.

11.4.6 QUERY BLOCK 0 ACCEPTED

If *"fwUpdateProcessEnabled"* is FALSE, this command shall be discarded, otherwise execution shall continue as follows:

If *"sessionKey"* is not MASK or 0, the reply shall be "YES", otherwise the reply shall be "NO".

NOTE If the device is busy, for example erasing flash, it is possible that there will be no answer even if block 0 has been accepted. A reply of "YES" indicates that block 0 has been accepted, and the device is ready to receive further block data.

11.5 Data transfer commands

11.5.1 General

A data transfer command marked as "send twice" in Table 7 shall be discarded unless it is accepted twice according to the requirements stated in IEC 62386-101:2022, 9.4.

All data transfer commands shall be interpreted as instructions unless explicitly stated otherwise in the description of the command.

11.5.2 BEGIN BLOCK (*data h, data m, data l*)

If the value of "*data h*", "*data m*" and "*data l*" is equal to any of the following:

- 0, or
- *"currentBlock"*, or
- *"currentBlock"* + 1,

and provided that all of the following conditions are false:

- the value of "*data h*", "*data m*" and "*data l*" is equal to *"currentBlock"* + 1 and *"currentBlockByte"* is not equal to the size of the current block, or
- the value of "*data h*", "*data m*" and "*data l*" is equal to *"currentBlock"* + 1 and *"blockIncomplete"* is TRUE, or
- the value of "*data h*", "*data m*" and "*data l*" is greater than the total block count from block 0,

then the following shall apply:

- *"blockIncomplete"* shall be set to TRUE.
- The bus unit shall set *"currentBlockByte"* to 0.
- The bus unit shall set *"currentBlock"* to the value transferred by the command itself. The parameters "*data h*", "*data m*" and "*data l*" ~~are used to~~ shall set the MSB, the middle byte and the LSB of *"currentBlock"* respectively.

Otherwise, the command shall be discarded.

NOTE ~~This allows to address 2^24 blocks in total.~~ This enables the addressing of 2^{24} blocks in total.

~~The programming of a block shall not take more than 300 ms.~~

11.5.3 TRANSFER BLOCK DATA (*data h, data m, data l*)

~~If the bus unit receives this command, 3 bytes of data are added to the current block. The bus unit shall increment "*currentBlockByte*" after byte is added to the current block. If "*currentBlockByte*" is the same or bigger than the current block data size, the data shall be discarded.~~

~~The byte "data h" in-command transferred byte shall be the first byte added to the block, the byte "data m" in-command transferred byte shall be the second byte and the byte "data l" in-command transferred byte shall be the third byte.~~

On execution of this command, if *"currentBlockByte"* is already equal to the current block data size, then *"blockIncomplete"* shall be set to TRUE to indicate unexpected block data and the three bytes of data shall be discarded, otherwise execution shall proceed as follows:

Up to three bytes of data shall be added to the current block. The bus unit shall increment *"currentBlockByte"* by 1 after each byte is added to the current block. If *"currentBlockByte"* reaches the current block data size, any remaining data shall be discarded. The byte "*data h*" shall be the first byte added to the block, "*data m*" shall be the second byte added and "*data l*" shall be the third byte added to the block.

Upon *"currentBlockByte"* reaching the current block data size, *"blockIncomplete"* shall be set to FALSE if all of the following conditions are met:

- *"currentBlock"* is 0 or *"sessionKey"* matches the session key of the block data, and
- *"currentBlock"* matches the block number of the block data, and
- the CRC (in the case of block 0) or both CRCs (in the case of block n) match the values calculated from the transferred block data.

See 9.7.2.1 for checks and operation upon completion of a block 0 transfer. See 9.7.2.2 for checks and operation upon completion of a block 1.. n transfer.

Whilst programming, it is possible that the bus unit will be unable to process receptions. This period during which the receiver is unresponsive shall be no more than 300 ms in the case where *"currentBlock"* is greater than 0, and no more than 120 s for the case where *"currentBlock"* is 0.

NOTE In the case where *"currentBlockByte"* reaches the current block data size, the command QUERY FW UPDATE RECEIVER READY can be used to determine when the bus unit has completed programming. In the case where block 0 has just been accepted, the bus unit can take longer before it becomes responsive, due to the time needed to erase flash memory.

Annex A
(normative)

Update file description

Besides update data, the manufacturer shall provide release notes in human readable form. The update file shall contain:

- date of release in ISO 8601 format: yyyy-mm-dd;
- affected devices;
- change log;
- description of the update.

It is recommended that the "description of the update" includes a statement on the possible impact of the firmware update on the number of logical units, device types and instances.

The human readable part of the file shall be separated from the block data by a line containing only 20 hyphen characters ('-' decimal code 45) and an end of line character '\n'.

Each block is then a line in the file starting with the block number (hexadecimal coded, 6 hex-characters) followed by a space character followed by the block data (hexadecimal coded, 2 characters per byte). Each line shall end with '\n'.

EXAMPLE (only one block, CRC not valid)

1970-01-01 Release 1.2\n

All LED drivers from version 1.0 up to version 1.1 from demo-company\n

Problem xx Fixed\n

--------------------\n

~~000000000000 3D0000010000000000000000000000000000000000000000000000000000000000000000000000000 000000000000000000000000000002222\n~~

000000 0041010203040506070800000001BBBBBB011F71FB268D0000FFFF0000FFFF0000000000000000FFFFFF FFFFFFFFFFABABABABABABABABABABABABABABABAB2222\n

It is recommended that the firmware update file extension is ".d2fw".

Annex B
(normative)

CRC16 calculation

This document specifies the use of CRC checksums to help ensure error-free transfer of data. The parameters of the CRC calculation shall be implemented according to this Annex B.

Detailed parameters are:

- polynomial is $P(x) = x^{16} + x^{15} + x^2 + 1$ (0x8005 in big-endian hex representation);
- initial CRC value is 0xFFFF;
- final XOR value is zero;
- input is reflected;
- result is reflected.

"Input reflected" means that each input byte is reflected before being used in the calculation. "Reflected" means that the bits of the input byte are used in reverse order.

"Result reflected" means that the final CRC value is reflected before being returned.

These two test vectors shall support testing of an implementation:

0x1A2B3C4D = 0x01A6

0x12345678 = 0x107B

Annex C
(informative)

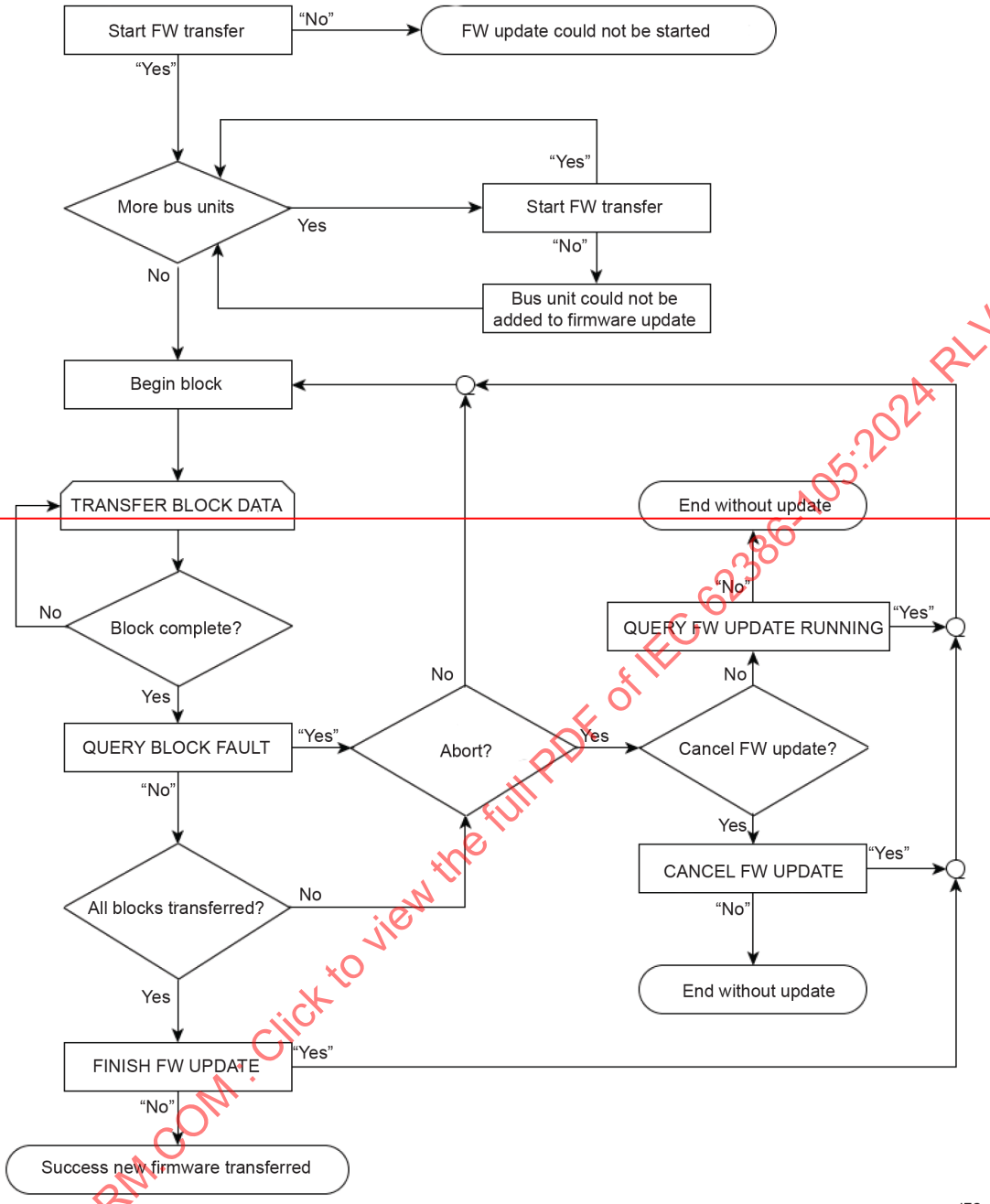
Firmware update process example

This document provides an example of a firmware update process and recommendations for a ~~programming device~~ FW update tool.

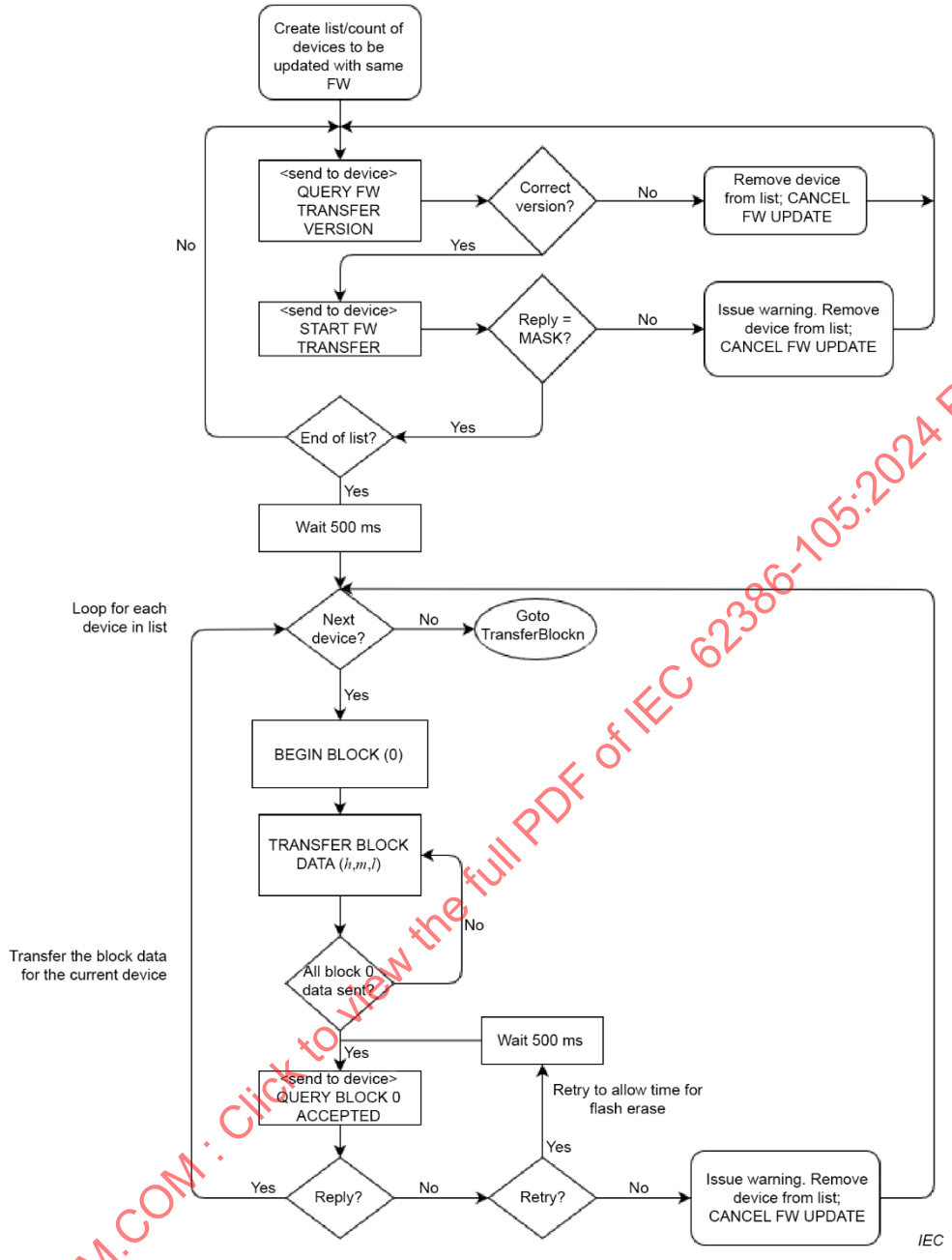
Figure C.1 gives an example of a firmware update process. This example uses broadcast commands, unless prefixed with "send to device". Most of these can be converted to broadcast commands.

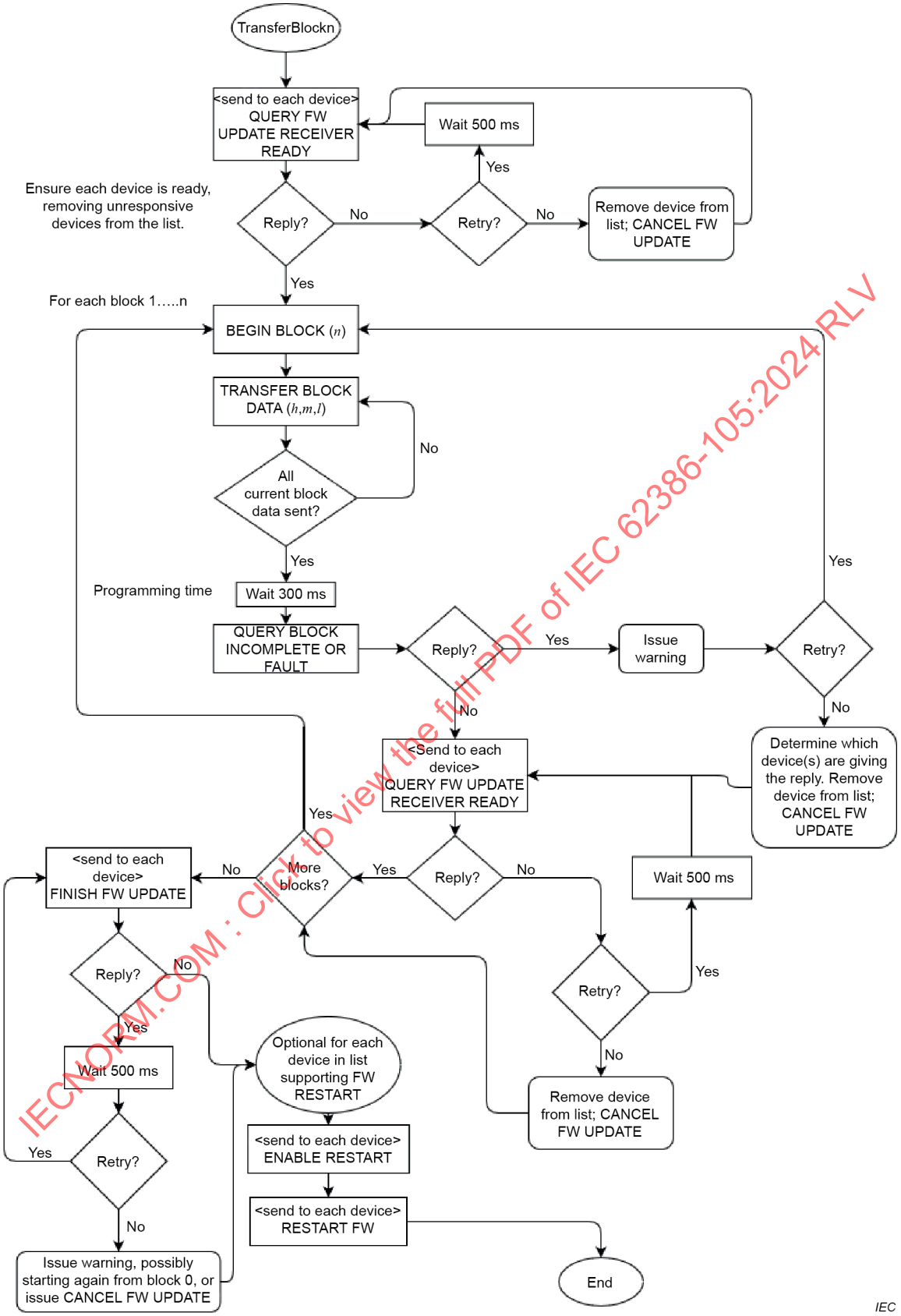
The example given in Figure C.1 does not consider that some bus units may have integrated bus power supplies that are disabled during a firmware update (see 9.6). To allow for this, QUERY FW UPDATE FEATURES can be sent to each bus unit to be updated, to determine if the bus unit has an integrated bus power supply that will be disabled during the firmware update. If this is the case, the tool can ensure that the bus unit's integrated bus power supply is disabled, either by sending commands or by prompting the user, and can then provide additional bus power whilst the bus unit's integrated bus power supply is disabled.

NOTE Where "Correct version?" is shown in Figure C.1, the decision as to which versions to accept is manufacturer specific.



NOTE 1 Answers on the bus are marked with quotation marks.





NOTE ~~2~~ After the "FINISH FW UPDATE"-block, some devices can restart automatically, whilst other devices cannot restart until execution of "RESTART FW".

Figure C.1 – Example of a firmware update process

After a firmware update data block (see 9.7.2.2) is transmitted, it is recommended that the bus unit checks the data consistency with the command ~~QUERY BLOCK FAULT~~ QUERY BLOCK INCOMPLETE OR FAULT (see 11.4.4). If the bus unit responds to this command with "YES", it is recommended that the ~~programming device~~ firmware update tool retransmits the block. If several block faults occur on the same bus unit, although the transmitted block data matches the bus unit and the block data is consistent, it is recommended that the site engineer checks the wiring.

It is recommended to transmit the firmware update data blocks with Priority 1 according to IEC 62386-101~~:2014, Table 22~~.

Annex D
(informative)

Firmware update management check sheet

Table D.1 gives an example of a check sheet that can be used for the firmware update of control gear. This can be used to manage the risk of updating control gear firmware in a live site. A similar checklist can be used when updating control devices.

Table D.1 – Example check sheet for firmware update of control gear

Firmware update name		
Check sheet version		Date:
Reason why firmware update is required *(bug fix, feature upgrade, consequences if update does not occur)*		
Control gear details	Manufacturer	
	Description (name)	
	Manufacture part ref.	
	GTIN	
	Identification/serial number range	
	Firmware range	
Location of gear *(in stock; installed; if installed, details of other gear on bus, type of installation)*		
Quantity of gear to be firmware updated		
Required conditions for firmware update *(Example: can other gear or other application controllers (multi-master bus) be connected to bus?)*		
Reference of new firmware (to be used in update)		
~~Reference of firmware update tool to be used~~		
Details of suitable firmware update tools		
Behaviour of gear during update	Fully functional	
	No functionality – light levels unchanged (mask)	
	No functionality – lights Off	
	No functionality – lights On	
	Other – give details	
If emergency ~~inverter~~ control gear (device type 1) firmware update, specifics to be detailed. (Example pre-, during and post-update: emergency test results, battery charge, failure status, mode status, post-update behaviour, example battery conditioning)		
Post firmware update: will commissioning activities be required? *(short and group addressing, scene settings)* If yes, give details.		
Detail actions and requirements concerning:-	Operating mode (if manufacturer-specific modes are affected, give details)	
	Bus configuration	
	Built-in bus power supplies (activation status)	

Brief description of firmware update process (method) *(including preparation, individual gear or by bus, by short address, group or broadcast, number of times firmware update will need to be run, serially or in parallel, hand-back details)*		
Length of time that update will require	Per single gear	
	For all gear	
Has a trial of this firmware update been successfully performed?		
If firmware update is at an installation, did firmware update trial simulate installed conditions? *(Example: other gear on bus, bus wiring)*		
If firmware update is at an installation and an emergency ~~inverter~~ control gear (device type 1) is connected to the bus or the firmware update is on an emergency ~~inverter~~ control gear (device type 1), detail emergency lighting system status relating to building/area occupancy during and 24 h after firmware update.		
If firmware update is for emergency ~~inverter~~ control gear (device type 1) at an installation, is a recertification of the emergency lighting installation by the appropriate fire authority required?		
Where is the firmware update being performed?		
Who is to perform the firmware update?		
When is the firmware update to be performed? *(date + time window, if no functionality during update, fully functional working hand back deadline time)*		
If at an installation, what are the client's hand back criteria post firmware updates?		
If at an installation, have permits been obtained? Provide details.		
What are the contingency plans including:- *(likely occurrence and severity of the event to be considered, the risk)*	If first firmware update fails (power outage, etc.)	
	Firmware update is not successful and control gear subsequently has no bus control functionality.	
	Consideration of other gear on bus being affected by firmware update process	
Do any liability disclaimers need generating? *(if yes, give details)*		
Organization(s) responsible for firmware update costs *(provide details as appropriate)*		
Agreement approval for firmware update activity to proceed		
Gear manufacturer	End client	Organization performing the firmware update
Signature:	Signature:	Signature:
Name:	Name:	Name:
Organization:	Organization:	Organization:
Date:	Date:	Date:

Bibliography

IEC 62386-250, *Digital addressable lighting interface – Part 250: Particular requirements – Integrated power supply (device type 49)*

_____

IEC 62386-105

Edition 2.0 2024-12

INTERNATIONAL STANDARD

NORME INTERNATIONALE

colour inside

Digital addressable lighting interface – Part 105: Particular requirements for control gear and control devices – Firmware transfer

Interface d'éclairage adressable numérique – Partie 105: Exigences particulières pour appareillages et dispositifs de commande – Transfert du microprogramme

IEC 62386-105:2024-12(en-fr)

CONTENTS

INTERNATIONAL ELECTROTECHNICAL COMMISSION

_____

DIGITAL ADDRESSABLE LIGHTING INTERFACE –

Part 105: Particular requirements for control gear and control devices – Firmware transfer

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) IEC draws attention to the possibility that the implementation of this document may involve the use of (a) patent(s). IEC takes no position concerning the evidence, validity or applicability of any claimed patent rights in respect thereof. As of the date of publication of this document, IEC had not received notice of (a) patent(s), which may be required to implement this document. However, implementers are cautioned that this may not represent the latest information, which may be obtained from the patent database available at https://patents.iec.ch. IEC shall not be held responsible for identifying any or all such patent rights.

IEC 62386-105 has been prepared by IEC technical committee 34: Lighting. It is an International Standard.

This second edition cancels and replaces the first edition published in 2020. This edition constitutes a technical revision.

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

a) several commands have been modified, renamed and added;
b) variables have been modified and added;
c) recommendations for implementation within emergency control gear have been added;
d) requirements for block acceptance have been changed;

e) example process-flow diagrams have been added;

f) requirements for restarting and power-on have been changed.

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

Draft	Report on voting
34/1258/FDIS	34/1281/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/publications.

This document is intended to be used in conjunction with:

- IEC 62386-101, which contains general requirements for system components;
- IEC 62386-102, which contains general requirements for the relevant product type (control gear), and with the appropriate parts of the IEC 62386-2xx series (particular requirements for control gear);
- IEC 62386-103, which contains general requirements for the relevant product type (control devices), and with the appropriate parts of the IEC 62386-3xx series (particular requirements for control devices);
- IEC 62386-104, which contains general requirements for wireless and alternative wired system components.

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, or
- revised.

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 a 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. IEC 62386-101 contains general requirements for system components, IEC 62386-102 extends this information with general requirements for control gear and IEC 62386-103 extends it further with general requirements for control devices. IEC 62386-104 and IEC 62386-105 can be applied to control gear or control devices. IEC 62386-104 gives requirements for wireless and alternative wired system components. IEC 62386-105 describes firmware transfer. IEC 62386-150 gives requirements for an auxiliary power supply which can be stand-alone, or built into control gear or control devices.

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

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

This second edition of IEC 62386-105 is intended to be used in conjunction with IEC 62386-101, IEC 62386-102 and the various parts that make up the IEC 62386-2xx series for control gear, together with IEC 62386-103 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 recognized.

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

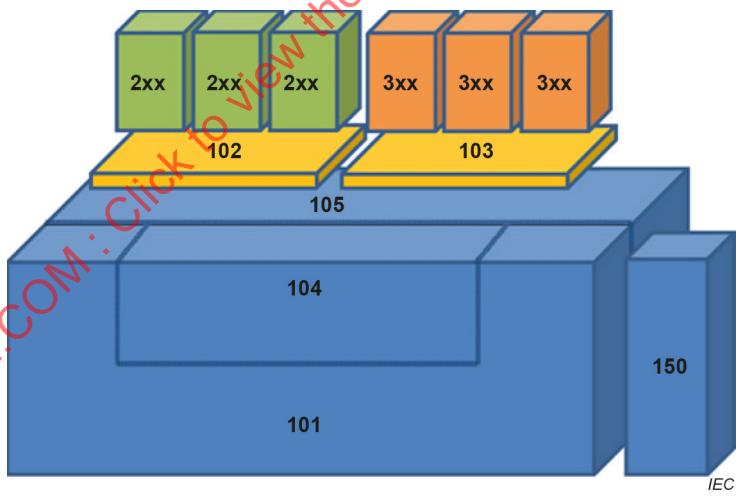


Figure 1 – IEC 62386 graphical overview

When this document refers to any of the clauses 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, "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".

Function or command parameters: *parameter name*.

DIGITAL ADDRESSABLE LIGHTING INTERFACE –

Part 105: Particular requirements for control gear and control devices – Firmware transfer

1 Scope

This part of IEC 62386 applies to control gear and control devices for control by digital signals of electronic lighting equipment.

Typically, a bus unit according to the IEC 62386 series contains firmware. There are circumstances where it can be necessary to change the firmware after production or shipping of the product, for example if the bus unit does not operate as intended. In such a case, a firmware update of a bus unit via the interface is beneficial.

This firmware update process is primarily designed to be a bug fix process, not a feature extension process. Nevertheless, the firmware update process can be used for feature extensions. But it is important that the risk of negative effects to the complete system be considered in detail.

NOTE Annex D provides a "Firmware update management check sheet" to support risk estimation.

2 Normative references

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

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

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

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

3 Terms and definitions

For the purposes of this document, the terms and definitions given in IEC 62386-101, IEC 62386-102, IEC 62386-103 and the following apply.

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

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

3.1
FW
firmware
software programmed into a control gear or control device

Note 1 to entry: Firmware can be changed during an update.

3.2
CRC
cyclic redundancy check
checksum used to prevent data corruption

Note 1 to entry: Annex B provides detailed information about CRC calculation.

3.3
block
unit of data containing information

Note 1 to entry: Information in a firmware update block usually contains firmware content.

3.4
programming
writing firmware transfer data to non-volatile memory (NVM)

3.5
normal operation
operation according to IEC 62386-102 or IEC 62386-103

4 General

4.1 General

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

NOTE Systems with a single-master application controller are unlikely to operate correctly when other master control devices, such as firmware update tools, are connected.

4.2 Logical units in a bus unit

If the firmware update process is started on a bus unit, all logical units inside the bus unit shall be affected. All variables defined in Table 5 shall be shared by all logical units of the bus unit. Commands addressed to one or more logical units within the bus unit shall be accepted by the bus unit according to the requirements of 9.2.

4.3 Updating control gear for emergency lighting

If IEC 62386-105 is implemented in control gear for emergency lighting, the product manual or data sheet should include guidance on the safety implications of a firmware update that should be considered.

5 Electrical specification

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

6 Bus power supply

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 32-bit forward frame encoding

The forward frame format used for firmware update consists of n = 32 data bits as described in IEC 62386-101:2022, 7.4.3 (32-bit forward frame).

For commands, the 32-bit forward frame shall be encoded as shown in Table 1.

Table 1 – 32-bit command frame encoding

Bytes/Bits											Device addressing method
Address byte								Opcode byte			
								1	2	3	
31	30	29	28	27	26	25	24[a]	23…16	15…8	7…0	
0	64 short addresses						x				Short addressing
1	0	1	1	1	1	0	1				Data transfer command[b]
1	1	0	0	1	0	1	1				Data transfer command[b]
1	1	1	1	1	1	0	x				Broadcast unaddressed
1	1	1	1	1	1	1	x				Broadcast
All other address byte values											Reserved

[a] Where bit 24 is shown as "x", 0 indicates address space for control gear, 1 indicates address space for control devices.

[b] See Table 7 for data transfer 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 exception that the total duration of a transaction may exceed 400 ms (IEC 62386-101:2022, 9.3).

9.2 Commands

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

- The command is sent using short address, broadcast addressing or broadcast unaddressed addressing, and bit 24 of the command frame does not match the type of bus unit (control gear or control device).
- The command is sent using short addressing and the given short address is not equal to its short address.
- The command is sent using reserved addressing.

- The command is sent using broadcast unaddressed addressing and the short address is not MASK.
- The command is not defined (e.g. reserved command).

The following command groups can be identified:

- standard commands;
  - instructions;
  - queries;
- data transfer commands.

9.3 Data transmission

A bus unit receives a new FW block by block. The first block (block 0) contains information about the type of bus unit (see Table 3), which receives a new FW. This avoids transferring the wrong FW to a bus unit if more than one bus unit is updated at a time.

The update file shall contain release notes as described in Annex A.

9.4 Duration of firmware update

A data transmission frame consists of a start bit, a 32-bit data transfer command and a stop condition, which occupies the bus for around 30 ms. With a settling time of less than 15 ms (priority 1 settling time), the transmission of three bytes takes less than 45 ms. For an update of 64 kB it is expected to take less than 20 min.

9.5 Security

This document specifies the use of CRC checksums to help ensure error-free transfer of data. In addition, it is recommended that the individual manufacturer ensures firmware image integrity and authenticity, for example by making use of the device key.

9.6 Firmware update features

Each bus unit shall expose its firmware update features as a combination of device properties as given in Table 2.

Table 2 – Firmware update features

Bit	Description	Value	See
0	*"fwUpdateCancelSupported"* is TRUE?	"1" = "YES"	9.7.2.3
1	Integrated bus power supply is present and is disabled during firmware update	"1" = "YES" "0" = bus power supply state is unchanged during firmware update, or is not present	
2 to 7	Reserved – Not implemented	"0" = "NO"	

The bus unit firmware update features can be queried using QUERY FW UPDATE FEATURES.

Bit 1: "1" shall indicate that an integrated bus power supply is present and is automatically disabled during the firmware update process. "0" shall indicate that either there is no integrated bus power supply, or if there is an integrated bus power supply then its state shall not be automatically changed when starting, during, or at the end of the firmware update. To help prevent loss of bus power during a firmware update, a lock-out mechanism is described in 11.3.2.

NOTE For bus units with bit 1 = "1", using a firmware update tool that is unable to apply suitable bus power can result in a system with no bus power and so no communications are possible. In such cases, a bus power supply can be temporarily connected to the system to allow the firmware update process to continue.

9.7 Update process

9.7.1 Start firmware update

A bus unit shall enable the firmware update process whenever *"fwUpdateProcessEnabled"* is TRUE. This can be due to execution of the command START FW TRANSFER, or after a power cycle where the previous update failed (see Table 5). Several bus units can be addressed in this way to update more than one bus unit at a time.

NOTE 1 Annex C provides an example of the firmware update process.

The bus unit shall be capable of returning to normal operation until at least a block 0 has been successfully verified.

NOTE 2 This means that bus units with a single flash image of the firmware used for normal operation cannot erase that firmware until at least block 0 has been verified.

Whilst *"fwUpdateProcessEnabled"* is TRUE, the operation of the bus unit is manufacturer-specific except for the requirements given in this document.

NOTE 3 This includes, for example, the reaction to commands of other parts of the IEC 62386 series.

9.7.2 Data transfer

9.7.2.1 Block 0 (information block)

Block 0 contains all data for the bus unit to decide if it will accept the new firmware or not.

The global trade item number (GTIN), the hardware version number and the firmware version number contained in the bus unit are described in IEC 62386-102:2022, 9.10.7 (Memory bank 0 for control gear) and IEC 62386-103:2022, 9.11.7 (Memory bank 0 for control devices).

If *"currentBlock"* is 0, then upon reception of a complete block 0, the following information shall be checked, where Table 3 shows block 0 content:

- The received block 0 "Size of block" is equal to the value shown in Table 3.
- The received block 0 "Session key" is not MASK or 0.
- The received block 0 "Block number" is equal to the value shown in Table 3.
- The received block 0 "Block 0 version" is equal to the value shown in Table 3.
- The received block 0 "GTIN" matches the GTIN stored in memory bank 0.
- (Received block 0 "FW version min") ≤ (memory bank 0, Firmware version) ≤ (received block 0 "FW version max").
- (Received block 0 "HW version min") ≤ (memory bank 0, Hardware version) ≤ (received block 0 "HW version max").
- (Received block 0 "Identification number min") ≤ (memory bank 0, Identification number) ≤ (received block 0 "Identification number max").
- The received block 0 "Device key" meets manufacturer-specific requirements.
- The received block 0 "CRC" matches the calculated value based on the block 0 content; Annex B is applicable).

If the above check is successful, the following operation shall result:

- *"sessionKey"* shall be set to the received block 0 "Session key",
- previously received block data that is unwritten may be discarded.

Otherwise, the following operation shall result:

- The block 0 content shall be discarded.

NOTE In the case where block 0 content is discarded, *"fwUpdateProcessEnabled"* remains TRUE, so the bus unit will continue looking for valid block 0 data. This enables multiple devices to be updated, even where these devices use identical firmware but have different GTINs, due to the possibility of an update tool sending multiple block 0s.

Table 3 – Block 0 definitions

Offset (hex)	Size (bytes)	Description
00..01	2	Size of block 0 (MSB first, fixed value of 0x0041 for block 0)
02..09	8	Session key[a]
0A..0C	3	Block number (always 0x000000 for block 0)
0D	1	Block 0 version (always 0x01)
0E..10	3	Total block count (MSB first)[b]
11..16	6	GTIN (MSB first)
17..18	2	HW version min (MSB first)
19..1A	2	HW version max (MSB first)
1B..1C	2	FW version min (MSB first)
1D..1E	2	FW version max (MSB first)
1F..26	8	Identification number min (MSB first)
27..2E	8	Identification number max (MSB first)
2F..3E	16	Device key[c]
3F..40	2	CRC (MSB first)

Key

MSB Most significant byte

[a] The session key is generated by the FW update tool which transfers the firmware update.

[b] This is the number of blocks, excluding block 0, being transferred during the firmware update.

[c] The device key and its use is manufacturer-specific. It allows the manufacturer to specify different areas or options in their firmware. The device key shall be independent of the session key, allowing a firmware update tool to change the session key without requiring a change to the device key.

It is recommended to calculate the CRC checksums with the incoming bytes to minimize delay at the end of the block reception.

9.7.2.2 Block 1..*n* (data block)

Table 4 shows the content of a data block. A bus unit shall only accept a data block if the following conditions are true:

- *"fwUpdateProcessEnabled"* is TRUE, and
- the session key matches *"sessionKey"*, and is not MASK or 0, and
- the block number matches *"currentBlock"*, and
- *"currentBlock"* is not 0.

After reception of a whole block, the consistency of the firmware data inside the block shall be verified by CRC (offset 0x0D...0x0E). The data consistency of the whole block shall be verified by a second CRC (offset s + 0x0F…s + 0x10). The block may optionally be discarded if "size of whole block" is an unexpected size, as determined by the manufacturer. If the verification fails, the bus unit shall discard the block. If a block is valid, but is the same as the last programmed block, it is recommended to discard the block to prevent unnecessary write cycles.

Table 4 – Block 1..*n* definitions

Offset (hex)	Size (bytes)	Description
00..01	2	Size of whole block (MSB first)[a] [b]
02..09	8	Session key
0A..0C	3	Block number (MSB first)
0D..0E	2	CRC (of block firmware data bytes)
0F..(s + 0E)	s	Firmware data bytes (optionally encrypted by manufacturer)
(s + 0F)..(s + 10)	2	CRC (of whole block)

Key

MSB Most significant byte

[a] This allows a theoretical maximum block size of 65 535 bytes, resulting in a maximum firmware data size of 65 518 bytes.

[b] Whole block size = s + 17.

It is recommended to calculate the CRC checksums with the incoming bytes to minimize delay at the end of the block reception.

9.7.2.3 Cancel firmware update

If the bus unit supports cancellation of the firmware update process even when *"sessionKey"* is not MASK, *"fwUpdateCancelSupported"* shall be TRUE.

In this case, the command CANCEL FW UPDATE can be used to cancel the firmware update at any time and return the bus unit to normal operation.

9.7.3 Persistent variables during firmware update

A firmware update may change the internal memory arrangement and content of the corresponding bus unit, except for the following cases:

The values for

- the GTIN,
- the identification number,
- the hardware version,

shall not be affected by a firmware update.

If firmware update features bit 1 is "0", the enabled or disabled state of an integrated bus power supply shall not be changed by a firmware update (see 9.6).

For the purposes of this Subclause 9.7.3, a "structural change" in the bus unit is when any of the following are changed by the firmware update:

- number of logical units, or
- implemented device types, or
- quantity or types of implemented input device instances, or
- implemented feature types.

NOTE 1 A change in "current bus unit configuration" in memory bank 0 can cause a change of these "structural change" parameters.

The short addresses of all logical units shall not change as a result of a firmware update with a new firmware image until a restart with the new firmware image is carried out. If the restart with the new firmware image results in a structural change of the bus unit, then the short addresses of all logical units shall be set to MASK and it is recommended that all NVM variables are set to their factory defaults. If instead, the restart with a new firmware image results in no structural change of the bus unit, the short addresses of all logical units shall remain unchanged and it is recommended that all NVM variables are unchanged.

The manufacturer shall provide a document stipulating which variables are affected by the update and whether a re-commissioning of the system is necessary after the firmware update process.

After a firmware update, an updated bus unit should first proceed to a power-up sequence to (re-)load RAM variables.

NOTE 2 Owing to the fact that the programme memory of a bus unit is updated, values in other parts of the IEC 62386 series marked as ROM can be changed.

9.7.4 Firmware version number

It is permitted to transfer the same firmware to a bus unit multiple times using the firmware transfer procedure described in this document.

It is strongly recommended that two firmware update files containing different firmware do not contain the same GTIN number and the same firmware version number.

9.7.5 Firmware update in a system

While *"fwUpdateProcessEnabled"* is TRUE, the bus unit may discard some or all commands not described in this document. Execution or discarding of such commands shall not affect the firmware update process as described in this document.

While *"fwUpdateProcessEnabled"* is TRUE, the bus unit shall not send any forward frames.

It is recommended to run the firmware update under human observation to be able to react to any errors that occur.

It is recommended to avoid other communication on the bus while a firmware update process is running.

NOTE The FW update tool can use the quiescent mode defined in IEC 62386-103:2022, 9.10.4, to suppress forward frames from other bus units.

9.7.6 Error recovery

The requirements in this document ensure that an interrupted or incomplete firmware update process can be restarted or resumed until successfully completed. Such interruptions can be caused by power failure or communications interruption.

EXAMPLE An example implementation would be a "bootloader". The bootloader is a smaller partitioned piece of firmware that is capable of communicating over the bus and implementing the download and programming of a new firmware. If the bootloader does not complete the firmware update it will retain control until the process is re-established and completed. A common method is to start again from the beginning.

NOTE If a firmware update process is started, some or all commands not specified in this document can be stopped from execution.

9.8 Power-on

If an external power cycle occurs and *"fwUpdateRestartEnabled"* is TRUE, then normal operation implementing the changes made by the most recently transferred firmware image shall be executed immediately after the power cycle.

If an external power cycle interrupts a firmware update, the operation proceeds as follows:

- If *"fwUpdateCancelSupported"* is TRUE: normal operation shall result after a power-cycle.
- If *"fwUpdateCancelSupported"* is FALSE: the power on values shall be set in accordance with footnote "a" of Table 5.

In all other cases, normal operation shall resume after the power cycle.

10 Declaration of variables

Variables are given in Table 5. The variables in other implemented parts of the IEC 62386 series shall also exist at least whenever *"fwUpdateProcessEnabled"* is FALSE.

Table 5 – Declaration of variables

Variable	Default value (factory)	Reset value	Power-on value	Range of validity	Memory type
"fwUpdateProcessEnabled"	FALSE	no change	FALSE[a]	[TRUE, FALSE]	RAM
"currentBlock"	0	no change	0	[0, 0xFFFFFF]	RAM
"currentBlockByte"	0	no change	0	[0, 0xFFFF]	RAM
"blockIncomplete"	FALSE	no change	FALSE	[TRUE, FALSE]	RAM
"sessionKey"	MASK	no change	MASK[a]	[0, 0xFF FF FF FF FF FF FF FF]	RAM
"fwUpdateRestartEnabled"	FALSE	no change	FALSE	[TRUE, FALSE]	RAM
"fwUpdateCancelSupported"	[b]	no change	[b]	[TRUE, FALSE]	ROM

[a] If a previous update process has failed, *"fwUpdateProcessEnabled"* can still be set to TRUE after a power cycle, in which case the power-on value of *"sessionKey"* shall be 0, indicating that a previously interrupted update attempt can only be restarted from block 0.

[b] (Default) value of this bit shall be defined by the manufacturer.

11 Definition of commands

11.1 General

Standard commands are either control instructions defined in 11.3, or queries defined in 11.4. Data transfer commands are defined in 11.5.

Unused opcodes are reserved for future needs.

11.2 Overview sheets

Table 6 gives an overview of the standard commands for bus units having implemented this document.

Table 6 – Standard commands

Command name	Address byte	Opcode byte			Answer	Send twice	References	Command reference
	See 7.2							
		1	2	3				
START FW TRANSFER	*Device*[a]	0xFB	0x00	0x00	✓	✓	9.7.1	11.3.2
RESTART FW	*Device*[a]	0xFB	0x01	0x00	✓	✓		11.3.3
ENABLE RESTART	*Device*[a]	0xFB	0x02	0x00		✓		11.3.4
FINISH FW UPDATE	*Device*[a]	0xFB	0x03	0x00	✓	✓		11.3.5
CANCEL FW UPDATE	*Device*[a]	0xFB	0x04	0x00		✓	9.7.2.3	11.3.6
QUERY FW UPDATE FEATURES	*Device*[a]	0xFB	0x05	0x00	✓		9.6	11.4.1
QUERY FW RESTART ENABLED	*Device*[a]	0xFB	0x06	0x00	✓			11.4.2
QUERY FW UPDATE RECEIVER READY	*Device*[a]	0xFB	0x07	0x00	✓			11.4.3
QUERY BLOCK INCOMPLETE OR FAULT	*Device*[a]	0xFB	0x08	0x00	✓			11.4.4
QUERY FW TRANSFER VERSION	*Device*[a]	0xFB	0x09	0x00	✓		9.7.4	11.4.5
QUERY BLOCK 0 ACCEPTED	*Device*[a]	0xFB	0x0A	0x00	✓		9.7.2.1	11.4.6

[a] *Device* means Short addressing, Broadcast or Broadcast unaddressed from Table 1.

Table 7 gives an overview of the data transfer commands for bus units having implemented this document.

Table 7 – Data transfer commands

Command name	Address byte	Opcode byte			Answer	Send twice	References	Command reference
		1	2	3				
BEGIN BLOCK (*data h, data m, data l*)	0xCB	data h	data m	data l		✓	9.7.2	11.5.1
TRANSFER BLOCK DATA (*data h, data m, data l*)	0xBD	data h	data m	data l			9.7.2	11.5.3

11.3 Control instructions

11.3.1 General

Control instructions are used to change the configuration or the mode of operation of the bus unit. A control instruction marked as "send twice" in Table 6 shall be discarded unless it is accepted twice according to the requirements stated in IEC 62386-101:2022, 9.4. Some control instructions provide an answer, as shown in Table 6.

11.3.2 START FW TRANSFER

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

If bit 1 of the firmware update features described in 9.6 is "1", and the integrated bus power supply is enabled, then this command shall be discarded.

NOTE 1 This provides a lock-out mechanism to help prevent the loss of bus power when performing a firmware update on a bus unit that contains a switchable integrated bus power supply. When updating bus units that do not implement IEC 62386-250, but do have bit 1 of the firmware update features described in 9.6 equal to "1", a firmware update tool can prompt the user to apply the manufacturer-specific method to disable the integrated bus power supply before continuing with the firmware update.

In some manufacturer-specific conditions, this command can be discarded. Such conditions shall be stated in the product manual or datasheet.

NOTE 2 Some examples of manufacturer-specific conditions causing this command to be discarded are as follows: a self-contained emergency control gear whilst operating from the battery; a centrally-supplied emergency control gear whilst the emergency condition is true; control gear failure; application controller failure; or input device failure.

If the command is not discarded, execution shall continue as follows:

The bus unit shall set *"fwUpdateProcessEnabled"* to TRUE, set *"sessionKey"* to MASK and answer with "YES".

The bus unit shall be ready to receive the new firmware after a maximum time of 500 ms.

11.3.3 RESTART FW

If *"fwUpdateProcessEnabled"* is TRUE, or if *"fwUpdateRestartEnabled"* is FALSE, then this command shall be discarded, otherwise execution shall continue as follows:

"fwUpdateRestartEnabled" shall be set to FALSE.

This instruction is intended to restart firmware execution.

If the firmware that will be executed by this command does not meet the requirements specified in IEC 62386-102 or IEC 62386-103, then the reply shall be "YES", otherwise the reply shall be "NO".

NOTE An example where a "YES" reply is sent is when a "bootloader" image is restarted. This can occur if the device does not contain an alternative valid firmware image. Leaving the device in this state without completing a firmware update results in the device not having functionality according to IEC 62386-102 or IEC 62386-103.

The bus unit shall then restart firmware execution and shall be ready to receive frames within 120 s from the execution of this command.

Depending on the state of the firmware image, the device can restart the same firmware as previously executed, or a different firmware than was previously executed (for example, the newly transferred image), or a firmware that implements only the commands described in this document (bootloader).

11.3.4 ENABLE RESTART

If *"fwUpdateProcessEnabled"* is TRUE, this command shall be discarded, otherwise execution shall continue as follows:

The bus unit shall set *"fwUpdateRestartEnabled"* to TRUE.

11.3.5 FINISH FW UPDATE

If *"fwUpdateProcessEnabled"* is FALSE, this command shall be discarded, otherwise execution shall continue as follows:

If the following conditions are all true:

- *"currentBlock"* is equal to the total block count from block 0 (see Table 3), and

- *"currentBlockByte"* matches the block size in address 0 and 1 of this last received block, and
- an optional manufacturer-specific condition for completing the firmware update is TRUE,

then *"fwUpdateProcessEnabled"* shall be set to FALSE and *"fwUpdateRestartEnabled"* shall be set to TRUE, the reply shall be "NO" and the bus unit may optionally continue the operation described for RESTART FW except for the reply specified for that command.

NOTE An example of a case when FINISH FW UPDATE will also cause the device to restart, is for devices that only have space for a single firmware image that is destroyed when the block data transfer starts. In such cases, the device executes a bootloader until the firmware update finishes and the device is restarted.

If any of the conditions in the above bulleted list are FALSE, then the answer shall be "YES".

11.3.6 CANCEL FW UPDATE

If *"fwUpdateProcessEnabled"* is FALSE, this command shall be discarded, otherwise execution shall continue as follows:

"currentBlock" and *"currentBlockByte"* shall both be set to 0.

If *"sessionKey"* is MASK, or *"fwUpdateCancelSupported"* is TRUE, the bus unit shall:

- set the variable *"fwUpdateProcessEnabled"* to FALSE, and
- discard all transferred blocks, and
- then return to normal operation, which can be achieved by running through a power-up sequence and executing the previous firmware.

NOTE If *"fwUpdateProcessEnabled"* remains TRUE, the firmware update process can only proceed if block 0 is then successfully resent.

11.4 Queries

11.4.1 QUERY FW UPDATE FEATURES

If *"fwUpdateProcessEnabled"* is TRUE, this command shall be discarded, otherwise execution shall continue as follows:

The bus unit shall answer with the firmware update feature set, which is formed by a combination of bus unit features.

Refer to 9.6 for further information.

11.4.2 QUERY FW RESTART ENABLED

If *"fwUpdateProcessEnabled"* is TRUE, this command shall be discarded, otherwise execution shall continue as follows:

The bus unit shall answer with "YES", if the variable "*fwUpdateRestartEnabled*" is TRUE.

11.4.3 QUERY FW UPDATE RECEIVER READY

If "*fwUpdateProcessEnabled*" is FALSE, this command shall be discarded, otherwise execution shall continue as follows:

The bus unit shall answer with "YES", if the variable "*fwUpdateProcessEnabled*" is TRUE and the device is ready to receive data, otherwise the reply shall be "NO". The status "ready to receive data" includes the case when the last block has been transferred and written to memory.

NOTE Bus units will possibly need some time to erase flash or write firmware data after receiving each complete block. An FW update tool can use this query to determine when a bus unit is ready to continue. The query can also be used to determine if the firmware update process has been started and not completed in any bus units.

11.4.4 QUERY BLOCK INCOMPLETE OR FAULT

If *"fwUpdateProcessEnabled"* is FALSE, this command shall be discarded, otherwise execution shall continue as follows:

The bus unit shall answer with "NO", if *"blockIncomplete"* is FALSE, otherwise the answer shall be:

- if *"shortAddress"* of all logical units is MASK: "YES",
- in all other cases: the *"shortAddress"* of the first addressed logical unit (starting from logical unit index 0), 0AAAAAA1b.

NOTE Sending this query when an incomplete block has been transferred, will cause a reply, which does not necessarily indicate a fault, but indicates that the block transfer is incomplete. To check for successful block transfer, update tools can send this query at the end of each complete block transfer, for blocks 1 or greater.

11.4.5 QUERY FW TRANSFER VERSION

The answer to this command shall be 1.

NOTE A firmware update tool can use this version number to determine an acceptable format for block data. This version number will be increased in a future update to this document if the block data format is changed and can be increased if other changes are made.

11.4.6 QUERY BLOCK 0 ACCEPTED

If *"fwUpdateProcessEnabled"* is FALSE, this command shall be discarded, otherwise execution shall continue as follows:

If *"sessionKey"* is not MASK or 0, the reply shall be "YES", otherwise the reply shall be "NO".

NOTE If the device is busy, for example erasing flash, it is possible that there will be no answer even if block 0 has been accepted. A reply of "YES" indicates that block 0 has been accepted, and the device is ready to receive further block data.

11.5 Data transfer commands

11.5.1 General

A data transfer command marked as "send twice" in Table 7 shall be discarded unless it is accepted twice according to the requirements stated in IEC 62386-101:2022, 9.4.

All data transfer commands shall be interpreted as instructions unless explicitly stated otherwise in the description of the command.

11.5.2 BEGIN BLOCK (*data h, data m, data l*)

If the value of "*data h*", "*data m*" and "*data l*" is equal to any of the following:

- 0, or
- *"currentBlock"*, or
- *"currentBlock"* + 1,

and provided that all of the following conditions are false:

- the value of "*data h*", "*data m*" and "*data l*" is equal to *"currentBlock"* + 1 and *"currentBlockByte"* is not equal to the size of the current block, or
- the value of "*data h*", "*data m*" and "*data l*" is equal to *"currentBlock"* + 1 and *"blockIncomplete"* is TRUE, or
- the value of "*data h*", "*data m*" and "*data l*" is greater than the total block count from block 0,

then the following shall apply:

- *"blockIncomplete"* shall be set to TRUE.
- The bus unit shall set *"currentBlockByte"* to 0.
- The bus unit shall set *"currentBlock"* to the value transferred by the command itself. The parameters "*data h*", "*data m*" and "*data l*" shall set the MSB, the middle byte and the LSB of *"currentBlock"* respectively.

Otherwise, the command shall be discarded.

NOTE This enables the addressing of 2^{24} blocks in total.

11.5.3 TRANSFER BLOCK DATA (*data h, data m, data l*)

On execution of this command, if *"currentBlockByte"* is already equal to the current block data size, then *"blockIncomplete"* shall be set to TRUE to indicate unexpected block data and the three bytes of data shall be discarded, otherwise execution shall proceed as follows:

Up to three bytes of data shall be added to the current block. The bus unit shall increment *"currentBlockByte"* by 1 after each byte is added to the current block. If *"currentBlockByte"* reaches the current block data size, any remaining data shall be discarded. The byte "*data h*" shall be the first byte added to the block, "*data m*" shall be the second byte added and "*data l*" shall be the third byte added to the block.

Upon *"currentBlockByte"* reaching the current block data size, *"blockIncomplete*" shall be set to FALSE if all of the following conditions are met:

- *"currentBlock"* is 0 or *"sessionKey"* matches the session key of the block data, and
- *"currentBlock"* matches the block number of the block data, and
- the CRC (in the case of block 0) or both CRCs (in the case of block n) match the values calculated from the transferred block data.

See 9.7.2.1 for checks and operation upon completion of a block 0 transfer. See 9.7.2.2 for checks and operation upon completion of a block 1.. n transfer.

Whilst programming, it is possible that the bus unit will be unable to process receptions. This period during which the receiver is unresponsive shall be no more than 300 ms in the case where *"currentBlock"* is greater than 0, and no more than 120 s for the case where *"currentBlock"* is 0.

NOTE In the case where *"currentBlockByte"* reaches the current block data size, the command QUERY FW UPDATE RECEIVER READY can be used to determine when the bus unit has completed programming. In the case where block 0 has just been accepted, the bus unit can take longer before it becomes responsive, due to the time needed to erase flash memory.

Annex A
(normative)

Update file description

Besides update data, the manufacturer shall provide release notes in human readable form. The update file shall contain:

- date of release in ISO 8601 format: yyyy-mm-dd;
- affected devices;
- change log;
- description of the update.

It is recommended that the "description of the update" includes a statement on the possible impact of the firmware update on the number of logical units, device types and instances.

The human readable part of the file shall be separated from the block data by a line containing only 20 hyphen characters ('-' decimal code 45) and an end of line character '\n'.

Each block is then a line in the file starting with the block number (hexadecimal coded, 6 hex-characters) followed by a space character followed by the block data (hexadecimal coded, 2 characters per byte). Each line shall end with '\n'.

EXAMPLE (only one block, CRC not valid)

1970-01-01 Release 1.2\n

All LED drivers from version 1.0 up to version 1.1 from demo-company\n

Problem xx Fixed\n

----------\n

000000 00410102030405060708000000010BBBBBB011F71FB268D0000FFFF0000FFFF00000000000000000FFFFFF
FFFFFFFFFFABABABABABABABABABABABABABABAB2222\n

It is recommended that the firmware update file extension is ".d2fw".

Annex B
(normative)

CRC16 calculation

This document specifies the use of CRC checksums to help ensure error-free transfer of data. The parameters of the CRC calculation shall be implemented according to this Annex B.

Detailed parameters are:

- polynomial is $P(x) = x^{16} + x^{15} + x^2 + 1$ (0x8005 in big-endian hex representation);
- initial CRC value is 0xFFFF;
- final XOR value is zero;
- input is reflected;
- result is reflected.

"Input reflected" means that each input byte is reflected before being used in the calculation. "Reflected" means that the bits of the input byte are used in reverse order.

"Result reflected" means that the final CRC value is reflected before being returned.

These two test vectors shall support testing of an implementation:

0x1A2B3C4D = 0x01A6

0x12345678 = 0x107B

Annex C
(informative)

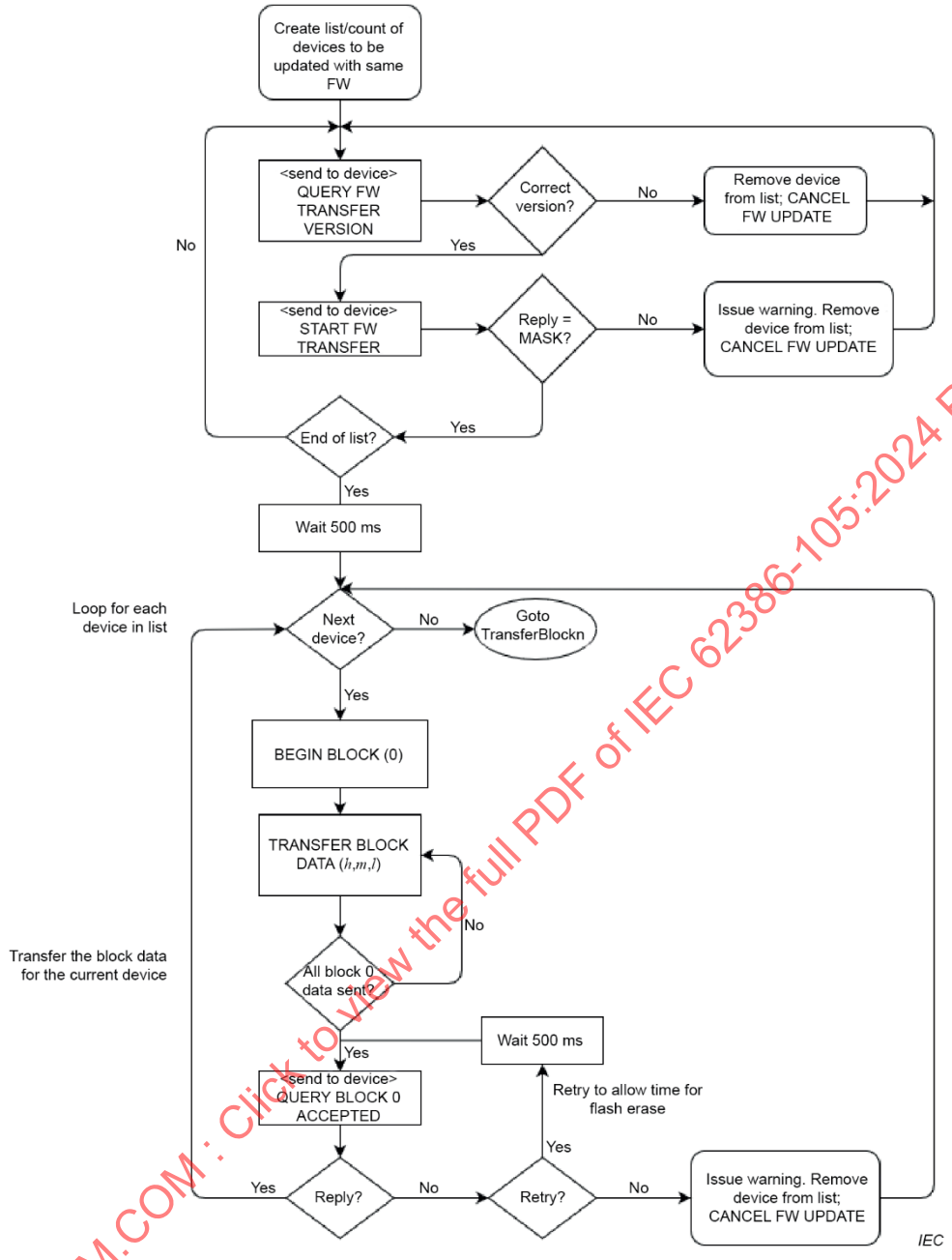
Firmware update process example

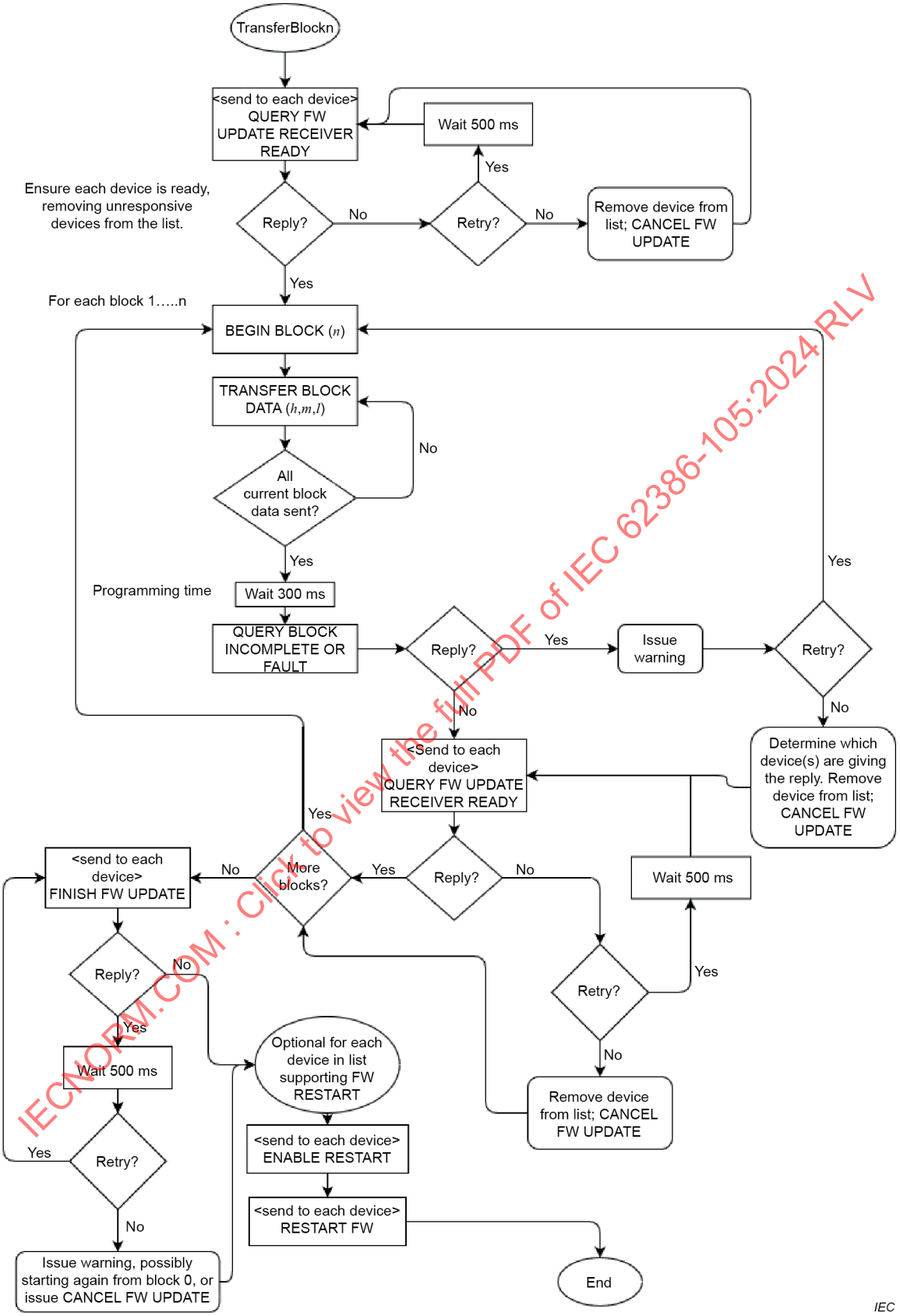
This document provides an example of a firmware update process and recommendations for a FW update tool.

Figure C.1 gives an example of a firmware update process. This example uses broadcast commands, unless prefixed with "send to device". Most of these can be converted to broadcast commands.

The example given in Figure C.1 does not consider that some bus units may have integrated bus power supplies that are disabled during a firmware update (see 9.6). To allow for this, QUERY FW UPDATE FEATURES can be sent to each bus unit to be updated, to determine if the bus unit has an integrated bus power supply that will be disabled during the firmware update. If this is the case, the tool can ensure that the bus unit's integrated bus power supply is disabled, either by sending commands or by prompting the user, and can then provide additional bus power whilst the bus unit's integrated bus power supply is disabled.

NOTE Where "Correct version?" is shown in Figure C.1, the decision as to which versions to accept is manufacturer specific.





NOTE After the "FINISH FW UPDATE"-block, some devices can restart automatically, whilst other devices cannot restart until execution of "RESTART FW".

Figure C.1 – Example of a firmware update process

After a firmware update data block (see 9.7.2.2) is transmitted, it is recommended that the bus unit checks the data consistency with the command QUERY BLOCK INCOMPLETE OR FAULT (see 11.4.4). If the bus unit responds to this command with "YES", it is recommended that the firmware update tool retransmits the block. If several block faults occur on the same bus unit, although the transmitted block data matches the bus unit and the block data is consistent, it is recommended that the site engineer checks the wiring.

It is recommended to transmit the firmware update data blocks with Priority 1 according to IEC 62386-101.

Annex D
(informative)

Firmware update management check sheet

Table D.1 gives an example of a check sheet that can be used for the firmware update of control gear. This can be used to manage the risk of updating control gear firmware in a live site. A similar checklist can be used when updating control devices.

Table D.1 – Example check sheet for firmware update of control gear

Firmware update name		
Check sheet version		Date:
Reason why firmware update is required *(bug fix, feature upgrade, consequences if update does not occur)*		
Control gear details	Manufacturer	
	Description (name)	
	Manufacture part ref.	
	GTIN	
	Identification/serial number range	
	Firmware range	
Location of gear *(in stock; installed; if installed, details of other gear on bus, type of installation)*		
Quantity of gear to be firmware updated		
Required conditions for firmware update *(Example: can other gear or other application controllers (multi-master bus) be connected to bus?)*		
Reference of new firmware (to be used in update)		
Details of suitable firmware update tools		
Behaviour of gear during update	Fully functional	
	No functionality – light levels unchanged (mask)	
	No functionality – lights Off	
	No functionality – lights On	
	Other – give details	
If emergency control gear (device type 1) firmware update, specifics to be detailed. (Example pre-, during and post-update: emergency test results, battery charge, failure status, mode status, post-update behaviour, example battery conditioning)		
Post firmware update: will commissioning activities be required? *(short and group addressing, scene settings)* If yes, give details.		
Detail actions and requirements concerning:-	Operating mode (if manufacturer-specific modes are affected, give details)	
	Bus configuration	
	Built-in bus power supplies (activation status)	

Brief description of firmware update process (method) *(including preparation, individual gear or by bus, by short address, group or broadcast, number of times firmware update will need to be run, serially or in parallel, hand-back details)*		
Length of time that update will require	Per single gear	
	For all gear	
Has a trial of this firmware update been successfully performed?		
If firmware update is at an installation, did firmware update trial simulate installed conditions? *(Example: other gear on bus, bus wiring)*		
If firmware update is at an installation and an emergency control gear (device type 1) is connected to the bus or the firmware update is on an emergency control gear (device type 1), detail emergency lighting system status relating to building/area occupancy during and 24 h after firmware update.		
If firmware update is for emergency control gear (device type 1) at an installation, is a recertification of the emergency lighting installation by the appropriate fire authority required?		
Where is the firmware update being performed?		
Who is to perform the firmware update?		
When is the firmware update to be performed? *(date + time window, if no functionality during update, fully functional working hand back deadline time)*		
If at an installation, what are the client's hand back criteria post firmware updates?		
If at an installation, have permits been obtained? Provide details.		
What are the contingency plans including:- *(likely occurrence and severity of the event to be considered, the risk)*	If first firmware update fails (power outage, etc.)	
	Firmware update is not successful and control gear subsequently has no bus control functionality.	
	Consideration of other gear on bus being affected by firmware update process	
Do any liability disclaimers need generating? *(if yes, give details)*		
Organization(s) responsible for firmware update costs *(provide details as appropriate)*		
Agreement approval for firmware update activity to proceed		
Gear manufacturer	End client	Organization performing the firmware update
Signature:	Signature:	Signature:
Name:	Name:	Name:
Organization:	Organization:	Organization:
Date:	Date:	Date:

Bibliography

IEC 62386-250, *Digital addressable lighting interface – Part 250: Particular requirements – Integrated power supply (device type 49)*

_____

SOMMAIRE

COMMISSION ÉLECTROTECHNIQUE INTERNATIONALE

_____

INTERFACE D'ÉCLAIRAGE ADRESSABLE NUMÉRIQUE –

Partie 105: Exigences particulières pour appareillages et dispositifs de commande – Transfert du microprogramme

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. À 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'IEC attire l'attention sur le fait que la mise en application du présent document peut entraîner l'utilisation d'un ou de plusieurs brevets. L'IEC ne prend pas position quant à la preuve, à la validité et à la portée de ces droits de propriété. À la date de publication du présent document, l'IEC n'a reçu aucune déclaration relative à des droits de brevets, qui pourraient être exigés pour la mise en œuvre du présent document. Toutefois, il est rappelé aux responsables de cette mise en œuvre qu'il ne s'agit peut-être pas des informations les plus récentes, qui peuvent être obtenues dans la base de données disponible à l'adresse https://patents.iec.ch. L'IEC ne saurait être tenue pour responsable de ne pas avoir identifié de tels droits de brevets.

L'IEC 62386-105 a été établie par le comité d'études 34 de l'IEC: Éclairage. Il s'agit d'une Norme internationale.

Cette deuxième édition annule et remplace la première édition parue en 2020. Cette édition constitue une révision technique.

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

a) plusieurs commandes ont été modifiées, renommées et ajoutées;

b) des variables ont été modifiées et ajoutées;

c) des recommandations de mise en œuvre dans les appareillages de commande de secours ont été ajoutées;

d) les exigences relatives à l'acceptation des blocs ont été modifiées;

e) des exemples de schémas de flux de processus ont été ajoutés;

f) les exigences relatives au redémarrage et à la mise sous tension ont été modifiées.

Le texte de la présente Norme internationale est issu des documents suivants:

Projet	Rapport de vote
34/1258/FDIS	34/1281/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.

Le présent document a été rédigé selon les Directives ISO/IEC, Partie 2, et 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/publications.

Le présent document est destiné à être utilisé conjointement avec:

- l'IEC 62386-101, qui contient des exigences générales relatives aux composants de système;
- l'IEC 62386-102, qui contient des exigences générales relatives au type de produit pertinent (appareillages de commande), et les parties appropriées de la série IEC 62386-2xx (exigences particulières relatives aux appareillages de commande);
- l'IEC 62386-103, qui contient des exigences générales relatives au type de produit pertinent (dispositifs de commande), et les parties appropriées de la série IEC 62386-3xx (exigences particulières relatives aux dispositifs de commande);
- IEC 62386-104, qui contient des exigences générales relatives aux composants de système à connexion alternative ou sans fil.

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 http://webstore.iec.ch dans les données relatives au document recherché. À cette date, le document sera

- reconduit,
- supprimé, ou
- révisé

IMPORTANT – Le logo "colour inside" qui se trouve sur la page de couverture de ce document 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 collectivement désignées comme une série. La série IEC 62386 spécifie un système à bus pour la commande par signaux numériques des équipements d'éclairage électroniques. La série IEC 62386-1xx inclut les spécifications de base. L'IEC 62386-101 contient les exigences générales relatives aux composants de système, l'IEC 62386-102 complète ces informations avec les exigences générales relatives aux appareillages de commande et l'IEC 62386-103 complète ces informations avec les exigences générales relatives aux dispositifs de commande. L'IEC 62386-104 et l'IEC 62386-105 peuvent être appliquées aux appareillages ou aux dispositifs de commande. L'IEC 62386-104 définit les exigences relatives aux composants de système à connexion alternative ou sans fil. L'IEC 62386-105 décrit le transfert du microprogramme. L'IEC 62386-150 définit les exigences relatives à une source d'alimentation auxiliaire qui peut être autonome, ou intégrée à des appareillages ou des 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'Édition 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 seconde édition de l'IEC 62386-105 est destinée à être utilisée conjointement avec l'IEC 62386-101, l'IEC 62386-102 et les différentes parties qui composent la série IEC 62386-2xx relatives aux appareillages de commande, ainsi qu'avec l'IEC 62386-103 et les différentes parties qui composent la série IEC 62386-3xx qui spécifient les exigences particulières relatives aux dispositifs de commande. La division en parties publiées séparément facilite 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.

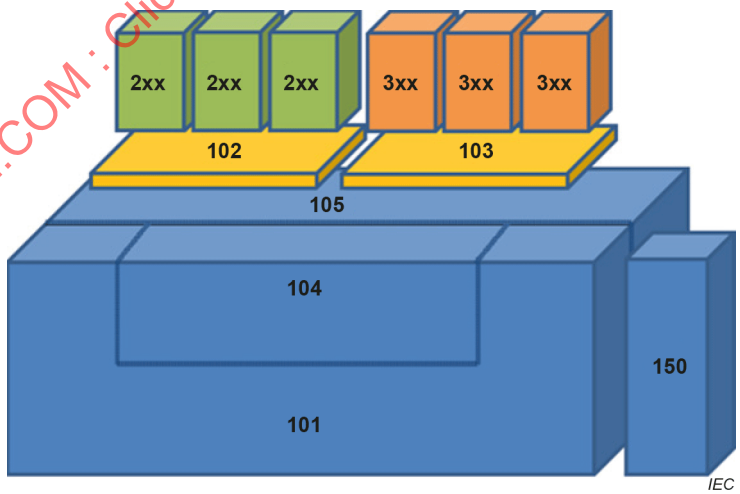


Figure 1 – Vue d'ensemble graphique de l'IEC 62386

Le présent document, tout en faisant référence à un article quelconque de la série IEC 62386-1xx, spécifie la mesure dans laquelle un article s'applique. Les autres parties contiennent également des exigences supplémentaires, si nécessaire.

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, 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 définit uniquement les bits 3 à 0 de *variableName;*

plage de valeurs: [lowest, highest] ([minimale, maximale]);

commande: "COMMAND NAME" (Nom de la commande);

les paramètres de fonction ou de commande: *parameter name*.

INTERFACE D'ÉCLAIRAGE ADRESSABLE NUMÉRIQUE –

Partie 105: Exigences particulières pour appareillages et dispositifs de commande – Transfert du microprogramme

1 Domaine d'application

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

Un appareillage de bus conforme à la série IEC 62386 contient généralement un microprogramme. Il peut être nécessaire, dans certaines circonstances, de modifier ce microprogramme après la production ou l'expédition du produit, par exemple lorsque l'appareillage de bus ne fonctionne pas comme prévu. Dans ce cas, il est bénéfique de mettre à jour le microprogramme de l'appareillage de bus par l'intermédiaire de l'interface.

Ce processus de mise à jour du microprogramme est principalement conçu comme un processus de correction des bogues et non comme un processus d'extension de caractéristiques. Le processus de mise à jour du microprogramme peut néanmoins être utilisé pour l'extension des caractéristiques. Il est important cependant d'étudier en détail le risque d'effets négatifs sur le système entier.

NOTE L'Annexe D fournit une "fiche de vérification de gestion de mise à jour du microprogramme" afin de prendre en charge l'estimation des risques.

2 Références normatives

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

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

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

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

3 Termes et définitions

Pour les besoins du présent document, les termes et définitions de l'IEC 62386-101, de l'IEC 62386-102, de l'IEC 62386-103, 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 https://www.electropedia.org/
- ISO Online browsing platform: disponible à l'adresse https://www.iso.org/obp

3.1
FW
microprogramme
logiciel programmé dans un appareillage ou un dispositif de commande

Note 1 à l'article: Le microprogramme peut être modifié lors d'une mise à jour.

Note 2 à l'article: L'abréviation "FW" est dérivée du terme anglais développé correspondant "*firmware*".

3.2
CRC
contrôle de redondance cyclique
somme de contrôle utilisée pour empêcher la corruption des données

Note 1 à l'article: L'Annexe B fournit des informations détaillées relatives au calcul du CRC.

Note 2 à l'article: L'abréviation "CRC" est dérivée du terme anglais développé correspondant "*cyclic redundancy check*".

3.3
bloc
unité de données contenant des informations

Note 1 à l'article: les informations contenues dans un bloc de mise à jour d'un microprogramme comprennent généralement des contenus relatifs au microprogramme.

3.4
programmation
écriture des données de transfert du microprogramme en mémoire non volatile, (NVM)

3.5
fonctionnement normal
fonctionnement conforme à l'IEC 62386-102 ou l'IEC 62386-103

4 Généralités

4.1 Généralités

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

NOTE Il est peu probable que les systèmes avec un contrôleur d'application à un seul maître fonctionnent correctement lorsque d'autres dispositifs de commande avec dispositif maître, comme des outils de mise à jour du microprogramme, sont connectés.

4.2 Unités logiques dans un appareillage de bus

Le démarrage du processus de mise à jour du microprogramme doit concerner toutes les unités logiques internes à l'appareillage de bus. Toutes les variables définies dans le Tableau 5 doivent être partagées par toutes les unités logiques de l'appareillage de bus. Les commandes adressées à une ou plusieurs unités logiques dans l'appareillage de bus doivent être acceptées par l'appareillage de bus conformément aux exigences du 9.2.

4.3 Mise à jour de l'appareillage de commande pour éclairage de secours

Si l'IEC 62386-105 est mise en œuvre dans l'appareillage de commande pour éclairage de secours, il convient que le manuel du produit ou la fiche technique contienne des recommandations sur les implications en termes de sécurité d'une mise à jour du microprogramme qu'il convient de prendre en compte.

5 Spécifications électriques

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

6 Alimentation électrique du bus

Les exigences de l'IEC 62386-101:2022, Article 6 s'appliquent.

7 Structure du protocole de transmission

7.1 Généralités

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

7.2 Codage des trames en avant de 32 bits

Le format de trame en avant utilisé pour la mise à jour du microprogramme consiste en n = 32 bits de données comme cela est décrit dans l'IEC 62386-101:2022, 7.4.3 (trame en avant de 32 bits).

Pour les commandes, la trame en avant de 32 bits doit être codée comme cela est présenté dans le Tableau 1.

Tableau 1 – Codage de la trame de commande de 32 bits

Octets/bits											Méthode d'adressage du dispositif
Octet d'adresse								Octet de code de fonctionnement			
								1	2	3	
31	30	29	28	27	26	25	24[a]	23…16	15…8	7…0	
0	64 adresses courtes						x				Adressage court
1	0	1	1	1	1	0	1				Commande de transfert de données[b]
1	1	0	0	1	0	1	1				Commande de transfert de données[b]
1	1	1	1	1	1	0	x				Diffusion non adressée
1	1	1	1	1	1	1	x				Diffusion
Toutes les autres valeurs d'octets d'adresses											Réservé

[a] Lorsque le bit 24 est représenté par un "x", 0 indique l'espace d'adressage des appareillages de commande, 1 indique l'espace d'adressage des dispositifs de commande.

[b] Voir le Tableau 7 pour les commandes de transfert de données.

8 Cadencement

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

9 Mode de fonctionnement

9.1 Généralités

Les exigences de l'IEC 62386-101:2022, Article 9 s'appliquent, à l'exception du fait que la durée totale d'une transaction peut dépasser 400 ms (IEC 62386-101:2022,9.3).

9.2 Commandes

Un appareillage de bus doit vérifier le schéma d'adressage du dispositif pour déterminer s'il est adressé par une commande. L'appareillage de bus doit accepter la commande, sauf si l'une des conditions suivantes s'applique:

- la commande est envoyée au moyen d'une adresse courte, par adressage par diffusion ou adressage par diffusion non adressée, et le bit 24 de la trame de commande ne correspond pas au type d'appareillage de bus (appareillage de commande ou dispositif de commande);
- la commande est envoyée par adressage court et l'adresse courte donnée n'est pas identique à son adresse courte;
- la commande est envoyée par adressage réservé;
- la commande est envoyée par adressage par diffusion non adressée et l'adresse courte n'est pas la valeur MASK;
- la commande n'est pas définie (commande réservée, par exemple).

Les groupes de commandes suivants peuvent être identifiés:

- commandes normalisées;
  - instructions;
  - requêtes;
- commandes de transfert de données.

9.3 Transmission de données

Un appareillage de bus reçoit un nouveau microprogramme bloc par bloc. Le premier bloc (bloc 0) contient des informations relatives au type d'appareillage de bus (voir le Tableau 3), qui reçoit un nouveau microprogramme. Cette disposition permet d'éviter d'envoyer le mauvais microprogramme à un appareillage de bus si plusieurs appareillages de bus sont mis à jour simultanément.

Le fichier de mise à jour doit contenir des notes de version telles qu'elles sont décrites à l'Annexe A.

9.4 Durée de la mise à jour du microprogramme

Une trame de transmission de données consiste en un bit de départ, une commande de transfert de données de 32 bits et un état d'arrêt, qui occupe le bus pendant environ 30 ms. Avec une durée d'établissement inférieure à 15 ms (durée d'établissement de priorité 1), la transmission de trois octets dure moins de 45 ms. Il est prévu qu'une mise à jour de 64 Ko dure moins de 20 min.

9.5 Sécurité

Le présent document spécifie l'utilisation des sommes CRC afin d'assurer un transfert de données sans erreur. En outre, il est recommandé que chaque fabricant veille à l'intégrité et à l'authenticité de l'image du microprogramme, par exemple, en utilisant la clé du dispositif.

9.6 Caractéristiques de mise à jour du microprogramme

Chaque appareillage de bus doit présenter ses caractéristiques de mise à jour du microprogramme en tant que combinaison des propriétés de dispositifs indiquées dans le Tableau 2.

Tableau 2 – Caractéristiques de mise à jour du microprogramme

Bit	Description	Valeur	Voir
0	*"fwUpdateCancelSupported"* est TRUE?	"1" = "YES"	9.7.2.3
1	L'alimentation de bus intégrée est présente et elle est désactivée pendant la mise à jour du microprogramme	"1" = "YES" "0" = l'état de l'alimentation de bus n'est pas modifié pendant la mise à jour du microprogramme, ou n'est pas présent	
2 à 7	Réservés – Pas mis en œuvre	"0" = "NO"	

Les caractéristiques de mise à jour du microprogramme de l'appareillage de bus peuvent faire l'objet d'une requête par le biais de la commande QUERY FW UPDATE FEATURES.

Bit 1: "1" doit indiquer qu'une alimentation de bus intégrée est présente et qu'elle est automatiquement désactivée pendant le processus de mise à jour du microprogramme. "0" doit indiquer que, soit il n'y a pas d'alimentation de bus intégrée, soit si une telle alimentation existe, que son état ne doit pas être automatiquement modifié au démarrage, au cours ou à l'issue de la mise à jour du microprogramme. Un mécanisme de verrouillage qui vise à empêcher la perte d'alimentation du bus au cours de la mise à jour du microprogramme est décrit au 11.3.2.

NOTE ¨Les appareillages de commande dont le bit 1 = "1", qui utilisent un outil de mise à jour du microprogramme qui n'est pas capable d'appliquer une alimentation adaptée du bus peuvent donner lieu à un système sans alimentation du bus et rendre ainsi impossibles les communications. Dans de tels cas, une alimentation de bus peut être temporairement connectée au système pour permettre au processus de mise à jour du microprogramme de continuer.

9.7 Processus de mise à jour

9.7.1 Début de la mise à jour du microprogramme

Un appareillage de bus doit autoriser le processus de mise à jour du microprogramme à chaque fois que "*fwUpdateProcessEnabled*" est TRUE. Cette disposition peut être due à l'exécution de la commande START FW TRANSFER, ou à un cycle de mise sous tension au cours duquel la mise à jour précédente a échoué (voir le Tableau 5). Plusieurs appareillages de bus peuvent être adressés de cette façon afin de mettre à jour plusieurs appareillages de bus simultanément.

NOTE 1 Un exemple de processus de mise à jour du microprogramme est donné à l'Annexe C.

L'appareillage de bus doit pouvoir revenir à un fonctionnement normal jusqu'à ce qu'au moins un bloc 0 ait été vérifié avec succès.

NOTE 2 Cela signifie que les appareillages de bus qui disposent d'une seule image flash du microprogramme utilisé pour le fonctionnement normal ne peuvent pas effacer ce microprogramme jusqu'à ce qu'au moins le bloc 0 ait été vérifié.

Alors que "*fwUpdateProcessEnabled*" est TRUE, le fonctionnement de l'appareillage de bus est spécifique au fabricant, à l'exception des exigences spécifiées dans le présent document.

NOTE 3 Cette disposition inclut, par exemple, la réaction aux commandes définies dans les autres parties de la série IEC 62386.

9.7.2 Transfert de données

9.7.2.1 Bloc 0 (bloc d'information)

Le bloc 0 contient toutes les données permettant à l'appareillage de bus de décider de l'acceptation ou non du nouveau microprogramme.

Le code article international GTIN (*Global Trade Item Number*), le numéro de version du matériel et le numéro de version du microprogramme contenus dans l'appareillage de bus sont décrits dans l'IEC 62386-102:2022, 9.10.7 (Bloc de mémoire 0 pour les appareillages de commande) et l'IEC 62386-103:2022, 9.11.7 (Bloc de mémoire 0 pour les dispositifs de commande).

Si "*currentBlock*" est 0, à réception d'un bloc 0 complet, les informations suivantes doivent être vérifiées, le Tableau 3 présentant le contenu du bloc 0:

- le bloc 0 reçu "Taille du bloc" est égal à la valeur présentée dans le Tableau 3;
- le bloc 0 reçu "Clé de session" n'est pas MASK ou 0;
- le bloc 0 reçu "Numéro du bloc" est égal à la valeur présentée dans le Tableau 3;
- le bloc 0 reçu "Version du bloc 0" est égal à la valeur présentée dans le Tableau 3;
- le bloc 0 reçu "GTIN" correspond au GTIN archivé dans le bloc de mémoire 0;
- (bloc 0 reçu "version minimale du microprogramme") ≤ (bloc de mémoire 0, version du microprogramme) ≤ (bloc reçu 0 "version maximale du microprogramme");
- (bloc 0 reçu "version minimale du matériel") ≤ (bloc de mémoire 0, version du matériel) ≤ (bloc reçu 0 "version maximale du matériel");
- (bloc 0 reçu "Numéro d'identification minimal") ≤ (bloc de mémoire 0, numéro d'identification) ≤ (bloc reçu 0 "Numéro d'identification maximal");
- le bloc 0 reçu "Clé du dispositif" satisfait aux exigences spécifiques au fabricant;
- le bloc 0 reçu "CRC" correspond à la valeur calculée fondée sur le contenu du bloc 0; l'Annexe B s'applique).

Si le contrôle ci-dessus est satisfaisant, l'opération suivante doit se produire:

- "*sessionKey*" doit être réglée sur le bloc 0 reçu "Clé de session",
- les données de bloc reçues précédemment, mais non écrites peuvent être rejetées.

Dans le cas contraire, l'opération suivante doit se produire:

- Le contenu du bloc 0 doit être rejeté.

NOTE Si le contenu du bloc 0 est rejeté, "*fwUpdateProcessEnabled*" reste TRUE, et l'appareillage de bus continue par conséquent à chercher des données valides pour le bloc 0. Cela permet la mise à jour de plusieurs dispositifs, même si ces dispositifs utilisent des microprogrammes identiques mais qui ont des GTIN différents, parce qu'il est possible qu'un outil de mise à jour envoie de multiples blocs 0.