

# INTERNATIONAL STANDARD

## NORME INTERNATIONALE

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

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



## THIS PUBLICATION IS COPYRIGHT PROTECTED

Copyright © 2014 IEC, Geneva, Switzerland

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

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

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

Tel.: +41 22 919 02 11  
Fax: +41 22 919 03 00  
[info@iec.ch](mailto:info@iec.ch)  
[www.iec.ch](http://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 corrigenda or an amendment might have been published.

#### IEC Catalogue - [webstore.iec.ch/catalogue](http://webstore.iec.ch/catalogue)

The stand-alone application for consulting the entire bibliographical information on IEC International Standards, Technical Specifications, Technical Reports and other documents. Available for PC, Mac OS, Android Tablets and iPad.

#### IEC publications search - [www.iec.ch/searchpub](http://www.iec.ch/searchpub)

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](http://webstore.iec.ch/justpublished)

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

#### Electropedia - [www.electropedia.org](http://www.electropedia.org)

The world's leading online dictionary of electronic and electrical terms containing more than 30 000 terms and definitions in English and French, with equivalent terms in 14 additional languages. Also known as the International Electrotechnical Vocabulary (IEV) online.

#### IEC Glossary - [std.iec.ch/glossary](http://std.iec.ch/glossary)

More than 55 000 electrotechnical terminology entries in English and French extracted from the Terms and Definitions clause of IEC publications issued since 2002. Some entries have been collected from earlier publications of IEC TC 37, 77, 86 and CISPR.

#### IEC Customer Service Centre - [webstore.iec.ch/csc](http://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: [csc@iec.ch](mailto:csc@iec.ch).

### A propos de l'IEC

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

### A propos des publications IEC

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

#### Catalogue IEC - [webstore.iec.ch/catalogue](http://webstore.iec.ch/catalogue)

Application autonome pour consulter tous les renseignements bibliographiques sur les Normes internationales, Spécifications techniques, Rapports techniques et autres documents de l'IEC. Disponible pour PC, Mac OS, tablettes Android et iPad.

#### Recherche de publications IEC - [www.iec.ch/searchpub](http://www.iec.ch/searchpub)

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

#### IEC Just Published - [webstore.iec.ch/justpublished](http://webstore.iec.ch/justpublished)

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

#### Electropedia - [www.electropedia.org](http://www.electropedia.org)

Le premier dictionnaire en ligne de termes électroniques et électriques. Il contient plus de 30 000 termes et définitions en anglais et en français, ainsi que les termes équivalents dans 14 langues additionnelles. Egalement appelé Vocabulaire Electrotechnique International (IEV) en ligne.

#### Glossaire IEC - [std.iec.ch/glossary](http://std.iec.ch/glossary)

Plus de 55 000 entrées terminologiques électrotechniques, en anglais et en français, extraites des articles Termes et Définitions des publications IEC parues depuis 2002. Plus certaines entrées antérieures extraites des publications des CE 37, 77, 86 et CISPR de l'IEC.

#### Service Clients - [webstore.iec.ch/csc](http://webstore.iec.ch/csc)

Si vous désirez nous donner des commentaires sur cette publication ou si vous avez des questions contactez-nous: [csc@iec.ch](mailto:csc@iec.ch).

# INTERNATIONAL STANDARD

## NORME INTERNATIONALE

---

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

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

INTERNATIONAL  
ELECTROTECHNICAL  
COMMISSION

COMMISSION  
ELECTROTECHNIQUE  
INTERNATIONALE

PRICE CODE  
CODE PRIX

**XH**

ICS 29.140, 29.140.50

ISBN 978-2-8322-1905-8

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

## CONTENTS

|                                                      |    |
|------------------------------------------------------|----|
| FOREWORD.....                                        | 11 |
| INTRODUCTION.....                                    | 13 |
| 1 Scope.....                                         | 15 |
| 2 Normative references .....                         | 15 |
| 3 Terms and definitions .....                        | 15 |
| 4 General .....                                      | 18 |
| 4.1 General.....                                     | 18 |
| 4.2 Version number .....                             | 18 |
| 5 Electrical specification .....                     | 18 |
| 6 Interface power supply .....                       | 18 |
| 7 Transmission protocol structure.....               | 18 |
| 7.1 General.....                                     | 18 |
| 7.2 24 bit forward frame encoding.....               | 19 |
| 7.2.1 Frame format for instructions and queries..... | 19 |
| 7.2.2 Frame format for event messages.....           | 20 |
| 8 Timing .....                                       | 21 |
| 9 Method of operation.....                           | 21 |
| 9.1 General.....                                     | 21 |
| 9.2 Application controller .....                     | 21 |
| 9.2.1 General .....                                  | 21 |
| 9.2.2 Single-master application controller .....     | 22 |
| 9.2.3 Multi-master application controller .....      | 22 |
| 9.3 Input device .....                               | 22 |
| 9.4 Instances of input devices.....                  | 23 |
| 9.4.1 General .....                                  | 23 |
| 9.4.2 Instance number.....                           | 23 |
| 9.4.3 Instance type .....                            | 23 |
| 9.4.4 Feature type .....                             | 23 |
| 9.4.5 Instance groups.....                           | 24 |
| 9.5 Commands .....                                   | 24 |
| 9.5.1 General .....                                  | 24 |
| 9.5.2 Device commands .....                          | 24 |
| 9.5.3 Instance commands.....                         | 25 |
| 9.5.4 Feature commands .....                         | 25 |
| 9.6 Event messages .....                             | 25 |
| 9.6.1 Response to event messages .....               | 25 |
| 9.6.2 Device power cycle event .....                 | 25 |
| 9.6.3 Input notification event .....                 | 25 |
| 9.6.4 Event message filter .....                     | 26 |
| 9.7 Input signal and input value .....               | 27 |
| 9.7.1 General .....                                  | 27 |
| 9.7.2 Input resolution.....                          | 27 |
| 9.7.3 Getting the input value.....                   | 27 |
| 9.7.4 Notification of changes .....                  | 28 |
| 9.8 System failure.....                              | 28 |

|        |                                                           |    |
|--------|-----------------------------------------------------------|----|
| 9.9    | Operating a control device .....                          | 29 |
| 9.9.1  | Enable/disable the application controller.....            | 29 |
| 9.9.2  | Enable/disable event messages.....                        | 29 |
| 9.9.3  | Quiescent mode .....                                      | 29 |
| 9.9.4  | Modes of operation .....                                  | 30 |
| 9.10   | Memory banks .....                                        | 30 |
| 9.10.1 | General .....                                             | 30 |
| 9.10.2 | Memory map.....                                           | 31 |
| 9.10.3 | Selecting a memory bank location .....                    | 31 |
| 9.10.4 | Memory bank reading .....                                 | 32 |
| 9.10.5 | Memory bank writing.....                                  | 32 |
| 9.10.6 | Memory bank 0.....                                        | 33 |
| 9.10.7 | Memory bank 1.....                                        | 35 |
| 9.10.8 | Manufacturer specific memory banks.....                   | 37 |
| 9.10.9 | Reserved memory banks .....                               | 37 |
| 9.11   | Reset.....                                                | 37 |
| 9.11.1 | Reset operation .....                                     | 37 |
| 9.11.2 | Reset memory bank operation .....                         | 37 |
| 9.12   | Power on behaviour .....                                  | 37 |
| 9.12.1 | Power on .....                                            | 37 |
| 9.12.2 | Power cycle notification .....                            | 38 |
| 9.13   | Priority use .....                                        | 38 |
| 9.13.1 | General .....                                             | 38 |
| 9.13.2 | Priority of input notifications .....                     | 38 |
| 9.14   | Assigning short addresses .....                           | 39 |
| 9.14.1 | General .....                                             | 39 |
| 9.14.2 | Random address allocation.....                            | 39 |
| 9.14.3 | Identification of a device .....                          | 39 |
| 9.15   | Exception handling .....                                  | 40 |
| 9.16   | Device capabilities and status information .....          | 40 |
| 9.16.1 | Device capabilities.....                                  | 40 |
| 9.16.2 | Device status.....                                        | 40 |
| 9.16.3 | Instance status .....                                     | 41 |
| 9.17   | Non-volatile memory .....                                 | 41 |
| 10     | Declaration of variables .....                            | 42 |
| 11     | Definition of commands .....                              | 43 |
| 11.1   | General.....                                              | 43 |
| 11.2   | Overview sheets .....                                     | 43 |
| 11.3   | Event messages .....                                      | 48 |
| 11.3.1 | INPUT NOTIFICATION ( <i>device/instance, event</i> )..... | 48 |
| 11.3.2 | POWER NOTIFICATION ( <i>device</i> ) .....                | 48 |
| 11.4   | Device control instructions .....                         | 48 |
| 11.4.1 | General .....                                             | 48 |
| 11.4.2 | IDENTIFY DEVICE .....                                     | 48 |
| 11.4.3 | RESET POWER CYCLE SEEN .....                              | 49 |
| 11.5   | Device configuration instructions.....                    | 49 |
| 11.5.1 | General .....                                             | 49 |
| 11.5.2 | RESET .....                                               | 49 |
| 11.5.3 | RESET MEMORY BANK ( <i>DTR0</i> ) .....                   | 49 |

|         |                                                           |    |
|---------|-----------------------------------------------------------|----|
| 11.5.4  | SET SHORT ADDRESS ( <i>DTR0</i> ) .....                   | 49 |
| 11.5.5  | ENABLE WRITE MEMORY .....                                 | 49 |
| 11.5.6  | ENABLE APPLICATION CONTROLLER .....                       | 50 |
| 11.5.7  | DISABLE APPLICATION CONTROLLER .....                      | 50 |
| 11.5.8  | SET OPERATING MODE ( <i>DTR0</i> ) .....                  | 50 |
| 11.5.9  | ADD TO DEVICE GROUPS 0-15 ( <i>DTR2:DTR1</i> ) .....      | 50 |
| 11.5.10 | ADD TO DEVICE GROUPS 16-31 ( <i>DTR2:DTR1</i> ) .....     | 50 |
| 11.5.11 | REMOVE FROM DEVICE GROUPS 0-15 ( <i>DTR2:DTR1</i> ).....  | 50 |
| 11.5.12 | REMOVE FROM DEVICE GROUPS 16-31 ( <i>DTR2:DTR1</i> )..... | 50 |
| 11.5.13 | START QUIESCENT MODE .....                                | 50 |
| 11.5.14 | STOP QUIESCENT MODE .....                                 | 50 |
| 11.5.15 | ENABLE POWER CYCLE NOTIFICATION .....                     | 51 |
| 11.5.16 | DISABLE POWER CYCLE NOTIFICATION .....                    | 51 |
| 11.5.17 | SAVE PERSISTENT VARIABLES .....                           | 51 |
| 11.6    | Device queries .....                                      | 51 |
| 11.6.1  | General .....                                             | 51 |
| 11.6.2  | QUERY DEVICE CAPABILITIES.....                            | 51 |
| 11.6.3  | QUERY DEVICE STATUS .....                                 | 51 |
| 11.6.4  | QUERY APPLICATION CONTROLLER ERROR .....                  | 52 |
| 11.6.5  | QUERY INPUT DEVICE ERROR .....                            | 52 |
| 11.6.6  | QUERY MISSING SHORT ADDRESS.....                          | 52 |
| 11.6.7  | QUERY VERSION NUMBER.....                                 | 52 |
| 11.6.8  | QUERY CONTENT <i>DTR0</i> .....                           | 52 |
| 11.6.9  | QUERY NUMBER OF INSTANCES.....                            | 52 |
| 11.6.10 | QUERY CONTENT <i>DTR1</i> .....                           | 52 |
| 11.6.11 | QUERY CONTENT <i>DTR2</i> .....                           | 52 |
| 11.6.12 | QUERY RANDOM ADDRESS (H) .....                            | 53 |
| 11.6.13 | QUERY RANDOM ADDRESS (M).....                             | 53 |
| 11.6.14 | QUERY RANDOM ADDRESS (L).....                             | 53 |
| 11.6.15 | READ MEMORY LOCATION ( <i>DTR1</i> , <i>DTR0</i> ).....   | 53 |
| 11.6.16 | QUERY APPLICATION CONTROL ENABLED .....                   | 53 |
| 11.6.17 | QUERY OPERATING MODE .....                                | 53 |
| 11.6.18 | QUERY MANUFACTURER SPECIFIC MODE .....                    | 53 |
| 11.6.19 | QUERY QUIESCENT MODE.....                                 | 53 |
| 11.6.20 | QUERY DEVICE GROUPS 0-7 .....                             | 53 |
| 11.6.21 | QUERY DEVICE GROUPS 8-15 .....                            | 54 |
| 11.6.22 | QUERY DEVICE GROUPS 16-23 .....                           | 54 |
| 11.6.23 | QUERY DEVICE GROUPS 24-31 .....                           | 54 |
| 11.6.24 | QUERY POWER CYCLE NOTIFICATION .....                      | 54 |
| 11.6.25 | QUERY EXTENDED VERSION NUMBER( <i>DTR0</i> ) .....        | 54 |
| 11.6.26 | QUERY RESET STATE .....                                   | 54 |
| 11.7    | Instance control instructions .....                       | 54 |
| 11.8    | Instance configuration instructions.....                  | 54 |
| 11.8.1  | General .....                                             | 54 |
| 11.8.2  | ENABLE INSTANCE .....                                     | 55 |
| 11.8.3  | DISABLE INSTANCE .....                                    | 55 |
| 11.8.4  | SET PRIMARY INSTANCE GROUP ( <i>DTR0</i> ) .....          | 55 |
| 11.8.5  | SET INSTANCE GROUP 1 ( <i>DTR0</i> ).....                 | 55 |
| 11.8.6  | SET INSTANCE GROUP 2 ( <i>DTR0</i> ).....                 | 55 |

|          |                                                                                    |    |
|----------|------------------------------------------------------------------------------------|----|
| 11.8.7   | SET EVENT SCHEME ( <i>DTR0</i> ).....                                              | 55 |
| 11.8.8   | SET EVENT PRIORITY ( <i>DTR0</i> ).....                                            | 56 |
| 11.8.9   | SET EVENT FILTER ( <i>DTR2</i> , <i>DTR1</i> , <i>DTR0</i> ) .....                 | 56 |
| 11.9     | Instance queries .....                                                             | 56 |
| 11.9.1   | General .....                                                                      | 56 |
| 11.9.2   | QUERY INSTANCE TYPE .....                                                          | 56 |
| 11.9.3   | QUERY RESOLUTION .....                                                             | 56 |
| 11.9.4   | QUERY INSTANCE ERROR .....                                                         | 56 |
| 11.9.5   | QUERY INSTANCE STATUS.....                                                         | 56 |
| 11.9.6   | QUERY INSTANCE ENABLED .....                                                       | 57 |
| 11.9.7   | QUERY PRIMARY INSTANCE GROUP .....                                                 | 57 |
| 11.9.8   | QUERY INSTANCE GROUP 1.....                                                        | 57 |
| 11.9.9   | QUERY INSTANCE GROUP 2.....                                                        | 57 |
| 11.9.10  | QUERY EVENT SCHEME.....                                                            | 57 |
| 11.9.11  | QUERY INPUT VALUE .....                                                            | 57 |
| 11.9.12  | QUERY INPUT VALUE LATCH.....                                                       | 57 |
| 11.9.13  | QUERY EVENT PRIORITY .....                                                         | 57 |
| 11.9.14  | QUERY FEATURE TYPE.....                                                            | 58 |
| 11.9.15  | QUERY NEXT FEATURE TYPE.....                                                       | 58 |
| 11.9.16  | QUERY EVENT FILTER 0-7 .....                                                       | 58 |
| 11.9.17  | QUERY EVENT FILTER 8-15 .....                                                      | 58 |
| 11.9.18  | QUERY EVENT FILTER 16-23.....                                                      | 58 |
| 11.10    | Special commands.....                                                              | 58 |
| 11.10.1  | General .....                                                                      | 58 |
| 11.10.2  | TERMINATE.....                                                                     | 58 |
| 11.10.3  | INITIALISE ( <i>device</i> ).....                                                  | 59 |
| 11.10.4  | RANDOMISE .....                                                                    | 59 |
| 11.10.5  | COMPARE .....                                                                      | 59 |
| 11.10.6  | WITHDRAW.....                                                                      | 59 |
| 11.10.7  | SEARCHADDRH ( <i>data</i> ).....                                                   | 60 |
| 11.10.8  | SEARCHADDRM ( <i>data</i> ) .....                                                  | 60 |
| 11.10.9  | SEARCHADDRL ( <i>data</i> ) .....                                                  | 60 |
| 11.10.10 | PROGRAM SHORT ADDRESS ( <i>data</i> ) .....                                        | 60 |
| 11.10.11 | VERIFY SHORT ADDRESS ( <i>data</i> ) .....                                         | 60 |
| 11.10.12 | QUERY SHORT ADDRESS .....                                                          | 61 |
| 11.10.13 | WRITE MEMORY LOCATION ( <i>DTR1</i> , <i>DTR0</i> , <i>data</i> ) .....            | 61 |
| 11.10.14 | WRITE MEMORY LOCATION – NO REPLY ( <i>DTR1</i> , <i>DTR0</i> , <i>data</i> ) ..... | 61 |
| 11.10.15 | <i>DTR0</i> ( <i>data</i> ) .....                                                  | 61 |
| 11.10.16 | <i>DTR1</i> ( <i>data</i> ) .....                                                  | 62 |
| 11.10.17 | <i>DTR2</i> ( <i>data</i> ) .....                                                  | 62 |
| 11.10.18 | DIRECT WRITE MEMORY ( <i>DTR1</i> , <i>offset</i> , <i>data</i> ) .....            | 62 |
| 11.10.19 | <i>DTR1</i> : <i>DTR0</i> ( <i>data1</i> , <i>data0</i> ).....                     | 62 |
| 11.10.20 | <i>DTR2</i> : <i>DTR1</i> ( <i>data2</i> , <i>data1</i> ).....                     | 62 |
| 11.10.21 | SEND TESTFRAME ( <i>data</i> ) .....                                               | 62 |
| 12       | Test procedures .....                                                              | 63 |
| 12.1     | General notes on test.....                                                         | 63 |
| 12.1.1   | General .....                                                                      | 63 |
| 12.1.2   | Test execution .....                                                               | 63 |
| 12.1.3   | Data transmission.....                                                             | 64 |

|         |                                                                 |     |
|---------|-----------------------------------------------------------------|-----|
| 12.1.4  | Test setup .....                                                | 64  |
| 12.1.5  | Test output .....                                               | 64  |
| 12.1.6  | Test notation .....                                             | 65  |
| 12.1.7  | Test execution limitations .....                                | 66  |
| 12.1.8  | Test results .....                                              | 66  |
| 12.1.9  | Exception handling .....                                        | 66  |
| 12.1.10 | Unexpected answer .....                                         | 66  |
| 12.2    | Preamble .....                                                  | 68  |
| 12.2.1  | Test preamble .....                                             | 68  |
| 12.3    | Physical operational parameters .....                           | 79  |
| 12.3.1  | Polarity test .....                                             | 79  |
| 12.3.2  | Maximum and minimum system voltage .....                        | 80  |
| 12.3.3  | Overvoltage protection test.....                                | 80  |
| 12.3.4  | Current rating test.....                                        | 81  |
| 12.3.5  | Transmitter voltages .....                                      | 83  |
| 12.3.6  | Transmitter rising and falling edges .....                      | 84  |
| 12.3.7  | Transmitter bit timing.....                                     | 86  |
| 12.3.8  | Transmitter frame timing.....                                   | 88  |
| 12.3.9  | Receiver start-up behavior.....                                 | 89  |
| 12.3.10 | Receiver threshold.....                                         | 90  |
| 12.3.11 | Receiver bit timing.....                                        | 91  |
| 12.3.12 | Extended receiver bit timing .....                              | 95  |
| 12.3.13 | Receiver forward frame violation.....                           | 97  |
| 12.3.14 | Receiver settling timing .....                                  | 97  |
| 12.3.15 | Receiver frame timing FF-FF send twice.....                     | 98  |
| 12.3.16 | Transmitter collision avoidance by priority .....               | 100 |
| 12.3.17 | Transmitter collision detection for truncated idle phase.....   | 101 |
| 12.3.18 | Transmitter collision detection for extended active phase ..... | 104 |
| 12.4    | Device configuration instructions.....                          | 107 |
| 12.4.1  | RESET deviceGroups .....                                        | 107 |
| 12.4.2  | RESET quiescentMode.....                                        | 108 |
| 12.4.3  | RESET instance groups.....                                      | 109 |
| 12.4.4  | RESET event filter .....                                        | 110 |
| 12.4.5  | RESET event scheme .....                                        | 111 |
| 12.4.6  | RESET: timeout / command in-between.....                        | 112 |
| 12.4.7  | Send twice timeout (device).....                                | 114 |
| 12.4.8  | Send twice timeout (instance).....                              | 117 |
| 12.4.9  | Commands in-between (device).....                               | 119 |
| 12.4.10 | Commands in-between (instance).....                             | 122 |
| 12.4.11 | SAVE PERSISTENT VARIABLES .....                                 | 125 |
| 12.4.12 | SET OPERATING MODE.....                                         | 125 |
| 12.4.13 | Device Disable/Enable Application Controller .....              | 126 |
| 12.4.14 | Multi Master Control Device PING .....                          | 127 |
| 12.4.15 | Quiescent Mode .....                                            | 128 |
| 12.4.16 | Device power cycle notification.....                            | 129 |
| 12.4.17 | SET SHORT ADDRESS.....                                          | 130 |
| 12.4.18 | Reset/Power-on values (device) .....                            | 131 |
| 12.4.19 | Reset/Power-on values (instance) .....                          | 133 |
| 12.4.20 | DTR0 / DTR1 / DTR2.....                                         |     |

|         |                                                        |     |
|---------|--------------------------------------------------------|-----|
| 12.4.21 | DTR1:DTR0 and DTR2:DTR1 .....                          | 135 |
| 12.4.22 | Device Groups.....                                     | 136 |
| 12.5    | Device queries.....                                    | 137 |
| 12.5.1  | Device query capabilities .....                        | 137 |
| 12.5.2  | QUERY VERSION NUMBER.....                              | 137 |
| 12.5.3  | Device power cycle seen .....                          | 138 |
| 12.5.4  | Input device error .....                               | 138 |
| 12.6    | Device Memory banks.....                               | 139 |
| 12.6.1  | READ MEMORY LOCATION on Memory Bank 0.....             | 139 |
| 12.6.2  | READ MEMORY LOCATION on Memory Bank 1.....             | 144 |
| 12.6.3  | READ MEMORY LOCATION on other Memory Banks.....        | 146 |
| 12.6.4  | Memory bank writing.....                               | 148 |
| 12.6.5  | ENABLE WRITE MEMORY: writeEnableState.....             | 153 |
| 12.6.6  | ENABLE WRITE MEMORY: timeout / command in-between..... | 155 |
| 12.6.7  | RESET MEMORY BANK: timeout / command in-between .....  | 156 |
| 12.6.8  | RESET MEMORY BANK.....                                 | 159 |
| 12.7    | Device Special commands .....                          | 160 |
| 12.7.1  | INITIALISE – timer.....                                | 160 |
| 12.7.2  | TERMINATE.....                                         | 161 |
| 12.7.3  | INITIALISE - device addressing .....                   | 162 |
| 12.7.4  | RANDOMISE .....                                        | 163 |
| 12.7.5  | COMPARE .....                                          | 163 |
| 12.7.6  | WITHDRAW.....                                          | 165 |
| 12.7.7  | SEARCHADDRH / SEARCHADDRM / SEARCHADDRL .....          | 166 |
| 12.7.8  | PROGRAM SHORT ADDRESS.....                             | 167 |
| 12.7.9  | VERIFY SHORT ADDRESS .....                             | 169 |
| 12.7.10 | QUERY SHORT ADDRESS .....                              | 170 |
| 12.7.11 | IDENTIFY DEVICE .....                                  | 172 |
| 12.8    | Logical unit cross contamination .....                 | 174 |
| 12.8.1  | DTR0.....                                              | 174 |
| 12.8.2  | NVM variables .....                                    | 174 |
| 12.8.3  | Random address generation .....                        | 175 |
| 12.8.4  | Addressing 1 .....                                     | 176 |
| 12.8.5  | Addressing 2 .....                                     | 177 |
| 12.8.6  | Addressing 3 .....                                     | 179 |
| 12.9    | Instance addressing.....                               | 180 |
| 12.9.1  | Instance Type Addressing .....                         | 180 |
| 12.9.2  | Instance Primary Group .....                           | 181 |
| 12.9.3  | Instance Group 2 .....                                 | 182 |
| 12.9.4  | Instance Group 1 .....                                 | 184 |
| 12.9.5  | Instance Group Combinations.....                       | 185 |
| 12.9.6  | Multiple Instances Answer .....                        | 187 |
| 12.10   | Instance configuration instructions.....               | 188 |
| 12.10.1 | Instance Enable/Disable .....                          | 188 |
| 12.10.2 | Event Scheme .....                                     | 190 |
| 12.10.3 | Input Resolution & Input Value .....                   | 195 |
| 12.10.4 | Event Filter .....                                     | 195 |
| 12.11   | Instance queries .....                                 | 196 |
| 12.11.1 | Instance Number and Types .....                        |     |

|          |                                                    |     |
|----------|----------------------------------------------------|-----|
| 12.11.2  | Instance Status.....                               | 197 |
| 12.11.3  | Instance Error.....                                | 197 |
| 12.12    | Instance cross contamination.....                  | 198 |
| 12.12.1  | Instance Event Priority.....                       | 198 |
| 12.13    | Reserved Commands.....                             | 199 |
| 12.13.1  | Reserved standard device commands.....             | 199 |
| 12.13.2  | Reserved instance commands (instance type 0) ..... | 200 |
| 12.13.3  | Reserved special commands .....                    | 200 |
| 12.14    | General subsequences .....                         | 201 |
| 12.14.1  | Reset Device .....                                 | 201 |
| 12.14.2  | EnableApplicationControllerAndAllInstances.....    | 202 |
| 12.14.3  | DisableApplicationControllerAndAllInstances .....  | 202 |
| 12.14.4  | HasApplicationController .....                     | 202 |
| 12.14.5  | GetVersionNumber .....                             | 203 |
| 12.14.6  | AddDeviceGroups.....                               | 203 |
| 12.14.7  | RemoveDeviceGroups .....                           | 203 |
| 12.14.8  | ClearAllDeviceGroups.....                          | 204 |
| 12.14.9  | CheckDeviceGroups .....                            | 204 |
| 12.14.10 | GetDeviceGroups .....                              | 205 |
| 12.14.11 | PowerCycle .....                                   | 205 |
| 12.14.12 | PowerCycleAndWaitForBusPower .....                 | 205 |
| 12.14.13 | PowerCycleAndWaitForDecoder .....                  | 206 |
| 12.14.14 | SetupTestFrame .....                               | 206 |
| 12.14.15 | GetNumberOfInstances .....                         | 207 |
| 12.14.16 | GetEventFilter .....                               | 207 |
| 12.14.17 | SetEventFilter.....                                | 207 |
| 12.14.18 | GetNumberOfLogicalUnits .....                      | 207 |
| 12.14.19 | GetIndexOfLogicalUnit.....                         | 207 |
| 12.14.20 | GetRandomAddress.....                              | 208 |
| 12.14.21 | GetLimitedRandomAddress .....                      | 208 |
| 12.14.22 | SetSearchAddress .....                             | 208 |
| 12.14.23 | SetShortAddress .....                              | 209 |
| 12.14.24 | ReadMemBankMultibyteLocation .....                 | 209 |
| 12.14.25 | FindImplementedMemoryBank.....                     | 210 |
| 12.14.26 | FindAllImplementedMemoryBanks .....                | 210 |
| 12.14.27 | ShortAddress.....                                  | 211 |
| 12.14.28 | GroupAddress .....                                 | 211 |
| 12.14.29 | Broadcast .....                                    | 211 |
| 12.14.30 | BroadcastUnaddressed.....                          | 211 |
| 12.14.31 | InstanceNumber .....                               | 211 |
| 12.14.32 | InstanceGroup.....                                 | 212 |
| 12.14.33 | InstanceType .....                                 | 212 |
| 12.14.34 | InstanceBroadcast.....                             | 212 |
| 12.14.35 | FeatureOfInstanceNumber.....                       | 212 |
| 12.14.36 | FeatureOfInstanceGroup .....                       | 213 |
| 12.14.37 | FeatureOfInstanceType .....                        | 213 |
| 12.14.38 | FeatureOfInstanceBroadcast .....                   | 213 |
| 12.14.39 | FeatureOfDevice .....                              | 213 |
| 12.14.40 | FeatureOfDeviceWithGroupAddress .....              | 213 |

|                                                                                                                 |     |
|-----------------------------------------------------------------------------------------------------------------|-----|
| 12.14.41 FeatureOfDeviceWithBroadcast .....                                                                     | 214 |
| Bibliography.....                                                                                               | 215 |
| Figure 1 - IEC 62386 graphical overview.....                                                                    | 13  |
| Figure 2 – Current rating test.....                                                                             | 82  |
| Table 1 – 24-bit command frame encoding.....                                                                    | 19  |
| Table 2 – Instance byte in a command frame .....                                                                | 19  |
| Table 3 – 24-bit event message frame encoding .....                                                             | 20  |
| Table 4 – Instance types.....                                                                                   | 23  |
| Table 5 – Feature types .....                                                                                   | 23  |
| Table 6 – Instance group variables .....                                                                        | 24  |
| Table 7 – Device address information in power cycle event .....                                                 | 25  |
| Table 8 – Event addressing schemes.....                                                                         | 26  |
| Table 9 – Signal level (~50%) versus resolution and input value .....                                           | 27  |
| Table 10 – Example querying sequence to read a 4-byte input value .....                                         | 28  |
| Table 11 – Basic memory map of memory banks .....                                                               | 31  |
| Table 12 – Memory map of memory bank 0.....                                                                     | 34  |
| Table 13 – Memory map of memory bank 1.....                                                                     | 36  |
| Table 14 – Control device capabilities.....                                                                     | 40  |
| Table 15 – Control device status.....                                                                           | 40  |
| Table 16 – Instance status.....                                                                                 | 41  |
| Table 17 – Declaration of device variables.....                                                                 | 42  |
| Table 18 – Declaration of instance variables.....                                                               | 43  |
| Table 19 – Instance event messages .....                                                                        | 43  |
| Table 20 – Device event messages.....                                                                           | 43  |
| Table 21 – Standard commands.....                                                                               | 44  |
| Table 22 – Special commands (implemented by both application controller and input device).....                  | 47  |
| Table 23 – Device addressing with “INITIALISE ( <i>device</i> )” .....                                          | 59  |
| Table 24 – Unexpected outcome.....                                                                              | 67  |
| Table 25 – Parameters for test sequence Check Factory Default 103.....                                          | 74  |
| Table 26 – Parameters for test sequence CheckFactoryDefault103PerLogicalUnit .....                              | 77  |
| Table 27 – Parameters for test sequence Transmitter bit timing .....                                            | 79  |
| Table 28 – Parameters for test sequence Maximum and minimum system voltage .....                                | 80  |
| Table 29 – Parameters for test sequence Transmitter voltages .....                                              | 84  |
| Table 30 – Parameters for test sequence Transmitter rising and falling edges .....                              | 85  |
| Table 31 – Parameters for test sequence Transmitter bit timing .....                                            | 88  |
| Table 32 – Parameters for test sequence Receiver frame timing.....                                              | 89  |
| Table 33 – Parameters for test sequence Receiver start-up behavior.....                                         | 90  |
| Table 34 – Parameters for test sequence Receiver bit timing .....                                               | 92  |
| Table 35 – Parameters for test sequence extended receiver bit timing .....                                      | 96  |
| Table 36 – Parameters for test sequence Receiver frame violation and recovering after frame size violation..... |     |

|                                                                                                            |     |
|------------------------------------------------------------------------------------------------------------|-----|
| Table 37 – Parameters for test sequence Receiver frame timing .....                                        | 98  |
| Table 38 – Parameters for test sequence transmitter collision avoidance by priority .....                  | 101 |
| Table 39 – Parameters for test sequence transmitter collision detection for truncated<br>idle phase .....  | 104 |
| Table 40 – Parameters for test sequence transmitter collision detection for extended<br>active phase ..... | 107 |
| Table 41 – Parameters for test sequence RESET instance groups .....                                        | 110 |
| Table 42 – Parameters for test sequence Send twice timeout (device) .....                                  | 116 |
| Table 43 – Parameters for test sequence Send twice timeout (instance) .....                                | 118 |
| Table 44 – Parameters for test sequence Commands in-between (device) .....                                 | 121 |
| Table 45 – Parameters for test sequence Commands in-between .....                                          | 124 |
| Table 46 – Parameters for test sequence SET SHORT ADDRESS .....                                            | 131 |
| Table 47 – Parameters for test sequence Reset/Power-on values (device) .....                               | 132 |
| Table 48 – Parameters for test sequence Reset/Power-on values (instance) .....                             | 134 |
| Table 49 – Parameters for test sequence DTR0 / DTR1 / DTR2 .....                                           | 134 |
| Table 50 – Parameters for test sequence DTR1:DTR0 and DTR2:DTR1 .....                                      | 136 |
| Table 51 – Parameters for test sequence READ MEMORY LOCATION on Memory<br>Bank 0 .....                     | 143 |
| Table 52 – Parameters for test sequence READ MEMORY LOCATION on Memory<br>Bank 1 .....                     | 146 |
| Table 53 – Parameters for test sequence Memory bank writing .....                                          | 151 |
| Table 54 – Parameters for test sequence ENABLE WRITE MEMORY: writeEnableState .....                        | 154 |
| Table 55 – Parameters for test sequence ENABLE WRITE MEMORY: timeout /<br>command in-between .....         | 156 |
| Table 56 – Parameters for test sequence RESET MEMORY BANK: timeout / command<br>in-between .....           | 159 |
| Table 57 – Parameters for test sequence RESET MEMORY BANK .....                                            | 160 |
| Table 58 – Parameters for test sequence INITIALISE - device addressing .....                               | 162 |
| Table 59 – Parameters for test sequence COMPARE .....                                                      | 164 |
| Table 60 – Parameters for test sequence WITHDRAW .....                                                     | 166 |
| Table 61 – Parameters for test sequence PROGRAM SHORT ADDRESS .....                                        | 168 |
| Table 62 – Parameters for test sequence VERIFY SHORT ADDRESS .....                                         | 170 |
| Table 63 – Parameters for test sequence QUERY SHORT ADDRESS .....                                          | 171 |
| Table 64 – Parameters for test sequence IDENTIFY DEVICE .....                                              | 173 |
| Table 65 – Parameters for test sequence Addressing 2 .....                                                 | 179 |
| Table 66 – Parameters for test sequence Reserved commands: standard device<br>commands .....               | 199 |
| Table 67 – Parameters for test sequence Reserved instance commands (instance type 0) ..                    | 200 |
| Table 68 – Parameters for test sequence Reserved special commands .....                                    | 201 |

## INTERNATIONAL ELECTROTECHNICAL COMMISSION

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

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

International Standard IEC 62386-103 has been prepared by subcommittee 34C: Auxiliaries for lamps, of IEC technical committee 34: Lamps and related equipment.

The text of this standard is based on the following documents:

| FDIS          | Report on voting |
|---------------|------------------|
| 34C/1100/FDIS | 34C/1113/RVD     |

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

This publication has been drafted in accordance with the ISO/IEC Directives, Part 2.

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

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

The committee has decided that the contents of this publication will remain unchanged until the stability date indicated on the IEC web site under "<http://webstore.iec.ch>" in the data related to the specific publication. At this date, the publication will be

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

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

## INTRODUCTION

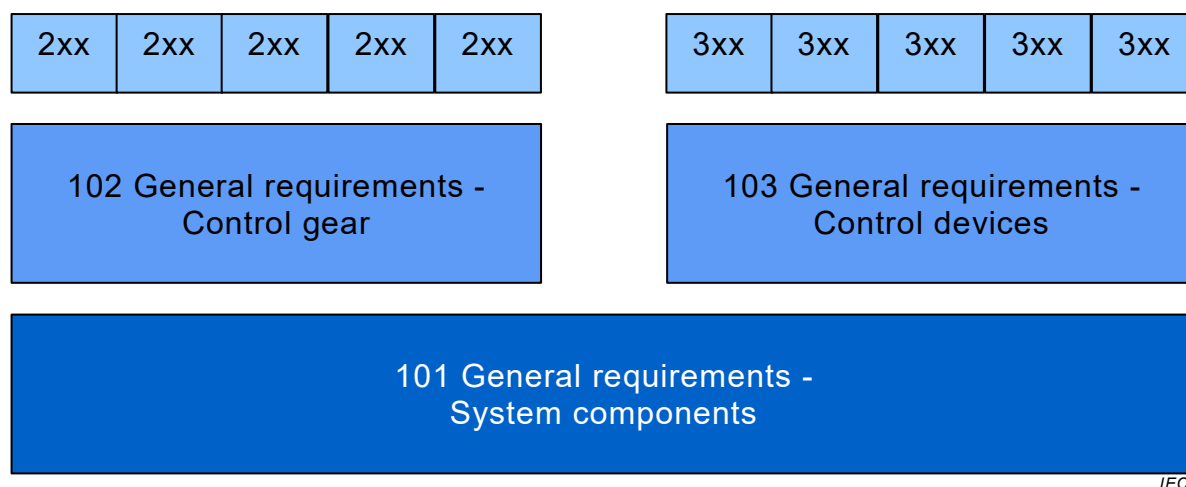
IEC 62386 contains several parts, referred to as series. The 1xx series includes the basic specifications. Part 101 contains general requirements for system components, Part 102 extends this information with general requirements for control gear and Part 103 extends it further with general requirements for control devices.

The 2xx parts extend 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 3xx parts extend 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 edition of IEC 62386-103 is published in conjunction with IEC 62386-101:2014, IEC 62386-102:2014 and with the various parts that make up the IEC 62386-2xx series for control gear, together with the various parts that make up the IEC 62386-3xx series of particular requirements for control devices. The division into separately published parts provides for ease of future amendments and revisions. Additional requirements will be added as and when a need for them is recognised.

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



IEC

**Figure 1 – IEC 62386 graphical overview**

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

All numbers used in this International Standard 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"

## DIGITAL ADDRESSABLE LIGHTING INTERFACE –

### Part 103: General requirements – Control devices

#### 1 Scope

This Part of IEC 62386 is applicable to control devices in a bus system for control by digital signals of electronic lighting equipment. This electronic lighting equipment should be in line with the requirements of IEC 61347, with the addition of d.c. supplies.

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

#### 2 Normative references

The following documents, in whole or in part, are normatively referenced in this document and are indispensable for its application. 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, *Digital addressable lighting interface – Part 101: General requirements – System components*

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

#### 3 Terms and definitions

For the purposes of this document, the terms and definitions given in IEC 62386-101:2014, Clause 3 apply, with the following additional terms and definitions.

##### 3.1

##### **broadcast**

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

##### 3.2

##### **broadcast unaddressed**

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

##### 3.3

##### **device command**

command which addresses the control device and has a value of 0xFE in the instance byte of the command frame

##### 3.4

##### **device group**

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

### 3.5

#### **DTR**

##### **data transfer register**

multipurpose register used to exchange data

### 3.6

#### **event**

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

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

### 3.7

#### **event scheme**

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

### 3.8

#### **feature command**

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

### 3.9

#### **GTIN**

number used for the unique identification of trade items worldwide.

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

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

### 3.10

#### **input signal**

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

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

### 3.11

#### **identification**

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

### 3.12

#### **input value**

encoded data, representing the input signal

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

### 3.13

#### **instance command**

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

### 3.14

#### **MASK**

the value 0xFF

**3.15****NO**

if a query is asked where the answer is NO, there will be no response, such that the sender of the query will conclude “no backward frame” following 8.2.5 of IEC 62386-101:2014

Note 1 to entry: The answer NO could also be triggered by a missed query.

**3.16****NVM**

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

**3.17****opcode****operation code**

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

**3.18****operating mode**

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

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

**3.19****PING**

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

Note 1 to entry: As specified in Part 102 of this standard, PING has no meaning to control gear.

**3.20****quiescent mode**

temporary mode in which the device does not send forward frames

**3.21****RAM**

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

**3.22****random address**

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

**3.23****reset state**

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

**3.24****ROM**

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

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

**3.25****search address**

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

### 3.26

#### **short address**

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

### 3.27

#### **YES**

if a query is asked where the answer is yes, the response will be a backward frame containing the value of *MASK*

## 4 General

### 4.1 General

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

### 4.2 Version number

This subclause replaces IEC 62386-101:2014, Subclause 4.2.

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

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

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

The current version number is "2.0".

NOTE Normally 2 amendments on IEC documents are made before a new edition is created.

## 5 Electrical specification

The requirements of IEC 62386-101:2014, Clause 5 apply.

## 6 Interface power supply

If a bus power supply is integrated into a control device, the requirements of IEC 62386-101:2014, Clause 6 apply.

## 7 Transmission protocol structure

### 7.1 General

The requirements of Clause 7 of IEC 62386-101:2014 apply, with the following additions;.

## 7.2 24 bit forward frame encoding

### 7.2.1 Frame format for instructions and queries

#### 7.2.1.1 General

For Subclause 7.2.1 commands shall be interpreted as instructions and queries. The 24 bit forward frame shall be encoded as shown in Table 1 and Table 2

**Table 1 – 24-bit command frame encoding**

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

**Table 2 – Instance byte in a command frame**

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

#### 7.2.1.2 Address byte

The address byte provides

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

### 7.2.1.3 Instance byte

The instance byte provides

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

### 7.2.1.4 Opcode byte

The opcode byte provides

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

## 7.2.2 Frame format for event messages

### 7.2.2.1 General

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

**Table 3 – 24-bit event message frame encoding**

| Bits                     |                    |                    |    |    |    |    |    |    |                     |    |    |    |            | Event scheme <sup>a</sup> / source |          |       |                 |
|--------------------------|--------------------|--------------------|----|----|----|----|----|----|---------------------|----|----|----|------------|------------------------------------|----------|-------|-----------------|
| Event source information |                    |                    |    |    |    |    |    |    |                     |    |    |    | Event info |                                    |          |       |                 |
| 23                       | 22                 | 21                 | 20 | 19 | 18 | 17 | 16 | 15 | 14                  | 13 | 12 | 11 | 10         |                                    |          | 9...0 |                 |
| 0                        | 64 short addresses |                    |    |    |    |    | 0  | 0  | 32 instance types   |    |    |    |            |                                    | Event    | 1     | Device          |
| 0                        | 64 short addresses |                    |    |    |    |    | 0  | 1  | 32 instance numbers |    |    |    |            |                                    |          | 2     | Device/instance |
| 1                        | 0                  | 32 device groups   |    |    |    |    | 0  | 0  | 32 instance types   |    |    |    |            |                                    |          | 3     | Device group    |
| 1                        | 0                  | 32 instance types  |    |    |    |    | 0  | 1  | 32 instance numbers |    |    |    |            |                                    |          | 0     | Instance        |
| 1                        | 1                  | 32 instance groups |    |    |    |    | 0  | 0  | 32 instance types   |    |    |    |            |                                    |          | 4     | Instance group  |
| 1                        | 1                  | 0                  | x  | x  | x  | x  | 0  | 1  | x                   | x  | x  | x  | x          | Reserved                           | Reserved |       |                 |
| 1                        | 1                  | 1                  | 0  | x  | x  | x  | 0  | 1  | x                   | x  | x  | x  | x          |                                    |          |       |                 |
| 1                        | 1                  | 1                  | 1  | 0  | x  | x  | 0  | 1  | x                   | x  | x  | x  | x          |                                    |          |       |                 |
| 1                        | 1                  | 1                  | 1  | 1  | 0  | x  | 0  | 1  | x                   | x  | x  | x  | x          |                                    |          |       |                 |
| 1                        | 1                  | 1                  | 1  | !  | 1  | 0  | 0  | 1  | x                   | x  | x  | x  | x          |                                    |          |       |                 |
| 1                        | 1                  | 1                  | !  | 1  | 1  | 1  | 0  | 1  | 0                   | x  | x  | x  | x          |                                    |          |       |                 |
| 1                        | 1                  | 1                  | 1  | 1  | 1  | 1  | 0  | 1  | 1                   | 0  | x  | x  | x          |                                    |          |       |                 |

| Bits                                                                                   |    |    |    |    |    |    |    |    |    |    |                                                            |            |    | Event scheme <sup>a</sup> /<br>source |
|----------------------------------------------------------------------------------------|----|----|----|----|----|----|----|----|----|----|------------------------------------------------------------|------------|----|---------------------------------------|
| Event source information                                                               |    |    |    |    |    |    |    |    |    |    |                                                            | Event info |    |                                       |
| 23                                                                                     | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12                                                         | 11         | 10 |                                       |
| 1                                                                                      | 1  | 1  | 1  | 1  | 1  | 1  | 0  | 1  | 1  | 1  | Short address/device group<br>information, refer to 9.6.2. |            |    | Device power cycle                    |
| <sup>a</sup> Refer to 9.6.2 of this standard for further information on event schemes. |    |    |    |    |    |    |    |    |    |    |                                                            |            |    |                                       |

### 7.2.2.2 Event source information

The event source information provides:

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

Events are instance type specific. This means that the event source information must be such that a receiver can derive – either explicitly or implicitly – the instance type of the transmitting instance. The identified event source schemes in Table 3 (and only these) satisfy this condition.

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

### 7.2.2.3 Event information

The event information provides the 10-bit event number and/or event data. Event information is instance type specific and is defined in the applicable Parts 3xx of this standard that describe the instance type.

## 8 Timing

The requirements of IEC 62386-101:2014, Clause 8 apply.

## 9 Method of operation

### 9.1 General

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

### 9.2 Application controller

#### 9.2.1 General

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

- it is an application controller that commissions and configures the system (including available control gear);
- it is an application controller that makes the system react to changes in the environment (based on information coming from input devices);
- it is an application controller that changes the behaviour of control gear in the system (possibly using any command defined in IEC 62386-102).

### 9.2.2 Single-master application controller

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

A single-master application controller may try to configure other control devices on the bus, and/or change the behaviour of control gear in the system, thereby using any command defined in IEC 62386-102 and/or instructions and queries defined in IEC 62386-103.

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

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

For all following subclauses, this standard assumes a control device to be a multi-master control device.

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

### 9.2.3 Multi-master application controller

For all following subclauses, this standard assumes a control device to be a multi-master control device.

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

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

In most cases, a system will have only one application controller active (refer to 9.9.1), but multiple application controllers can be operational in a single system.

An application controller shall accept commands (from other application controllers) according to Table 21 and Table 22. It is part of the system integration to ensure that the application controllers will do this in such a way that a correctly functioning system results.

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

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

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

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

An application controller shall not transmit PING messages.

## 9.3 Input device

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

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

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

## 9.4 Instances of input devices

### 9.4.1 General

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

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

### 9.4.2 Instance number

Each instance shall have a unique “*instanceNumber*” in the range [0, “*numberOfInstances*”–1].

### 9.4.3 Instance type

The instance type for each of the instances of an input device can be different. It can be queried by “QUERY INSTANCE TYPE”. The meaning of event information transmitted by means of “INPUT NOTIFICATION (*device/instance, event*)” depends on the instance type.

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

**Table 4 – Instance types**

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

### 9.4.4 Feature type

This standard allows for the future publication of feature extensions that extend the requirements in this specification, or exempt particular requirements.

The features for each of the instances of an input device can be different. They can be queried by “QUERY FEATURE TYPE” and “QUERY NEXT FEATURE TYPE”

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

**Table 5 – Feature types**

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

### 9.4.5 Instance groups

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

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

**Table 6 – Instance group variables**

| Variable                | Description                                                      |
|-------------------------|------------------------------------------------------------------|
| <i>“instanceGroup0”</i> | Primary instance group number, MASK if no membership defined.    |
| <i>“instanceGroup1”</i> | Additional instance group number, MASK if no membership defined  |
| <i>“instanceGroup2”</i> | Additional instance group number, MASK if no membership defined. |

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

- “SET PRIMARY INSTANCE GROUP (*DTR0*)”, “QUERY PRIMARY INSTANCE GROUP”
- “SET INSTANCE GROUP 1 (*DTR0*)”, “QUERY INSTANCE GROUP 1”
- “SET INSTANCE GROUP 2 (*DTR0*)”, “QUERY INSTANCE GROUP 2”

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

## 9.5 Commands

### 9.5.1 General

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

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

NOTE For instance commands, additional conditions for command acceptance hold. These are given in 9.5.3.

### 9.5.2 Device commands

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

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

### 9.5.3 Instance commands

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

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

### 9.5.4 Feature commands

For feature commands that are accepted by an input device (refer to 9.5), the feature addressing scheme determines the intended (set of) receiving features within that device.

## 9.6 Event messages

### 9.6.1 Response to event messages

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

NOTE If an application controller or input device is disabled, it is not allowed to send any response but can still update its internal state based on messages received.

### 9.6.2 Device power cycle event

Since the power cycle (see 9.12.2) event is a device event, it does not adhere to the default event frame format. Bits 12 through 0 carry device address information as is indicated in Table 7.

**Table 7 – Device address information in power cycle event**

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

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

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

### 9.6.3 Input notification event

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

**Table 8 – Event addressing schemes**

| <b>“eventScheme”</b> | <b>Description</b>                                                   |
|----------------------|----------------------------------------------------------------------|
| 0 (default)          | Instance addressing, using instance type and number.                 |
| 1                    | Device addressing, using short address and instance type.            |
| 2                    | Device/instance addressing, using short address and instance number. |
| 3                    | Device group addressing, using device group and instance type.       |
| 4                    | Instance group addressing, using instance group and type.            |

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

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

In the following situations, the instance shall immediately revert to the default instance addressing scheme:

- “eventScheme” has been set to 1 or 2 whereas the containing device has no short address;
- “eventScheme” has been set to 3 whereas the containing device is not member of a device group;
- “eventScheme” has been set to 4 whereas the instance has no primary instance group membership (see 9.4.5).

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

Once reverted to the default event scheme:

- “QUERY EVENT SCHEME” shall reflect this.
- Only a new “SET EVENT SCHEME (DTR0)” command may change the actual event scheme.

NOTE 3 This implies that the command “SET EVENT SCHEME (DTR0)” can “fail”, rather than that it expresses a preference that may be granted sooner or later. The application controller is recommended to set the desired event scheme only after completing those configuration aspects that influence event scheme operation.

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

- only refer to “instanceNumber” as the instance number;
- only refer to “instanceType” as the instance type;
- only refer to “instanceGroup0” as the instance group;
- only refer to “shortAddress” as the containing device short address;
- only refer to the lowest device group number of membership of the containing device.

#### **9.6.4 Event message filter**

The event message filter can be used to enable and disable specific events. To enable or disable all events see 9.9.2.

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

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

## 9.7 Input signal and input value

### 9.7.1 General

An instance shall process its input signal into an input value and expose this value to the system, as described in the following subclauses.

### 9.7.2 Input resolution

The processing shall be done with a precision which is indicated by “*resolution*”. The actual resolution used for particular instance (type) can be subject to Part 3xx requirements and/or manufacturer choice.

The result of the conversion shall be available in the *N*-byte variable “*inputValue*”, where *N* is the minimum number of bytes needed to contain at least “*resolution*” bits.

NOTE 1 *N* is computed as (*resolution*)/8) rounded up to the nearest integer. With “*resolution*” in the range [1,255], “*inputValue*” can span up to 32 bytes.

The result of the conversion and the “*inputValue*” shall be MSB-aligned. Unused bits in “*inputValue*” shall contain a repeating pattern of the most significant bit(s).

Table 9 provides an example, which shows a signal level of just below 50% latched into a 1-byte “*inputValue*” after being processed with a “*resolution*” of 3, 4 and 5 bits respectively.

**Table 9 – Signal level (~50%) versus resolution and input value**

| Resolution | Signal level  | Bits |   |   |   |   |   |   |   | Input value |
|------------|---------------|------|---|---|---|---|---|---|---|-------------|
|            |               | 7    | 6 | 5 | 4 | 3 | 2 | 1 | 0 |             |
| 3-bits     | 3 of [0, 7]   | 0    | 1 | 1 | 0 | 1 | 1 | 0 | 1 | 109         |
| 4-bits     | 7 of [0, 15]  | 0    | 1 | 1 | 1 | 0 | 1 | 1 | 1 | 119         |
| 5-bits     | 15 of [0, 31] | 0    | 1 | 1 | 1 | 1 | 0 | 1 | 1 | 123         |

NOTE 2 The grey shaded bits are (part of) the (first) repetition of the significant bits.

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

### 9.7.3 Getting the input value

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

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

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

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

The *“inputValue”* that is latched is the *“inputValue”* at the moment “QUERY INPUT VALUE” is received.

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

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

The application controller shall transmit the necessary queries for this scenario within a transaction.

NOTE 2 Using a transaction prevents concurrent access to the latched data.

An application controller may exit the scenario at any point.

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

#### 9.7.4 Notification of changes

A change or a sequence of changes in the input signal of an instance shall result in an event message as required by this document or that Part 3xx of the standard that describes the *“instanceType”* (see 9.4.3) of that instance.

The event message shall be sent using “INPUT NOTIFICATION (*device/instance, event*)”, as described in 11.3.1 of this standard.

NOTE The manufacturer of the input device should ensure that no event is lost. Parts 3xx of this standard may impose additional restrictions, e.g. to avoid event flooding.

#### 9.8 System failure

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

NOTE Bus powered devices may shutdown at a power outage of 40 ms.

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

## 9.9 Operating a control device

### 9.9.1 Enable/disable the application controller

If present, the application controller is either active or not-active, as shall be reflected by “*applicationActive*”. While deactivated, the application controller shall not send any forward frames, except possibly a power cycle notification (see 9.12.2).

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

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

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

### 9.9.2 Enable/disable event messages

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

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

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

To limit the event messages when enabled, filtering is also available, see 9.6.4.

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

### 9.9.3 Quiescent mode

In quiescent mode, the control device shall not produce any forward frames. No commands (see also 9.9.1), and no event messages (see also 9.9.2) shall be transmitted, regardless of “*applicationActive*” or any “*instanceActive*”.

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

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

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

NOTE 1 Quiescent mode can be used by the application controller during initialisation (see 9.14) to ensure that random address comparisons are not frustrated by forward frames from other devices on the bus.

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

## 9.9.4 Modes of operation

### 9.9.4.1 General

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

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

### 9.9.4.2 Operating mode 0x00: standard mode

If a device is in “*operatingMode*” 0x00, its behaviour shall be as is required per this specification, until it is set in an operating mode different from 0x00.

### 9.9.4.3 Operating mode 0x01 to 0x7F: reserved

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

### 9.9.4.4 Operating mode 0x80 to 0xFF: manufacturer specific modes

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

- as far as the control device accesses the bus, it shall adhere to IEC 62386-101:2014.
- the control device shall adhere to this specification at least as far as the following commands are concerned:
  - “SET OPERATING MODE (*DTR0*)”, and “QUERY OPERATING MODE” and “QUERY MANUFACTURER SPECIFIC MODE”.
  - All special commands (see 11.9.14) except WRITE MEMORY LOCATION (*DTR1*, *DTR0*, *data*), WRITE MEMORY LOCATION – NO REPLY (*DTR1*, *DTR0*, *data*) and DIRECT WRITE MEMORY (*DTR1*, *offset*, *data*).

For the above commands the various addressing methods shall apply, see 7.2.1.2.

It is recommended that even in manufacturer specific modes, the commands as specified in this standard be still obeyed.

## 9.10 Memory banks

### 9.10.1 General

Memory banks are freely accessible memory spaces defined for e.g. identification of the control device in a system. Not all consecutive memory banks need to be implemented. Also within a memory bank not all consecutive locations need to be implemented. All implemented memory bank locations of implemented memory banks are readable using memory access commands. Part of the memory is read-only and programmed by the manufacturer of the control device. For all other parts, write access using memory access commands can be enabled by the manufacturer. Write access to a memory bank location can be locked. Memory banks can be implemented using RAM, ROM or NVM.

The addressable memory space is limited to a maximum of almost 64 kBytes, organized in maximum 256 memory banks of maximum 255 bytes each. As this standard prescribes how to

implement memory bank 0 and 1 (if present), and reserves memory banks 200 to 255, this leaves room for 198 memory banks for manufacturer specific purposes in the range of [2,199].

### 9.10.2 Memory map

If a manufacturer specific memory bank in the range of [2,199] is implemented, allocation of its content shall comply with the memory map provided in Table 11.

**Table 11 – Basic memory map of memory banks**

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

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

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

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

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

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

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

### 9.10.3 Selecting a memory bank location

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

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

#### 9.10.4 Memory bank reading

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

If the selected memory bank is not implemented, the command shall be ignored. If the memory bank exists, and selected memory bank location is

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

the answer shall be NO.

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

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

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

#### 9.10.5 Memory bank writing

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

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

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

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

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

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

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

If the selected memory bank location is

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

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

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

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

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

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

#### 9.10.6 Memory bank 0

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

Memory bank 0 shall be implemented using the memory map shown in Table 12, with at least the memory locations up to address 0x7F implemented, excluding reserved locations.

**Table 12 – Memory map of memory bank 0**

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

<sup>a</sup> It is recommended that the product GTIN is not re-used within the expected lifetime of the product after installation.

<sup>b</sup> Format of the version number is defined in IEC62386-101, Clause 4.2. If not implemented, this is indicated by 0xFF.

<sup>c</sup> Purpose and (default) value of these bytes shall be defined by the manufacturer.

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

A bus unit might contain both control gear and control devices. They share various numbers (e.g. GTIN, identification number, etc). To avoid problems when reading, and getting different answers depending on the addressing scheme used, the memory bank layout are the same for control gear and for control devices up to and including location 0x19. The data shall be the same as well. The application controller can use either the 102 or the 103 commands to identify the basic data, provided both are implemented.

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

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

The bytes in locations 0x0B to 0x12 ("identification number byte 0" to "identification number byte 7") shall contain 64 bits of an identification number of the bus unit, preferably the serial number. The identification number shall be stored with least significant byte in "identification number byte 8" and unused bits shall be filled with 0.

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

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

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

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

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

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

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

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

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

### 9.10.7 Memory bank 1

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

If implemented, memory bank 1 shall at least implement the memory locations up to and including address 0x10. A recommended memory map is shown in Table 13.

**Table 13 – Memory map of memory bank 1**

| Address | Description                                                                                                                        | Default value (factory)            | RESET value <sup>b</sup> | Memory type      |
|---------|------------------------------------------------------------------------------------------------------------------------------------|------------------------------------|--------------------------|------------------|
| 0x00    | Address of last accessible memory location                                                                                         | factory burn-in, range [0x10,0xFE] | no change                | ROM              |
| 0x01    | Indicator byte <sup>a</sup>                                                                                                        | <sup>a</sup>                       | <sup>a</sup>             | any <sup>a</sup> |
| 0x02    | Memory bank 1 lock byte. Lockable bytes in the memory bank shall be read-only while the lock byte has a value different from 0x55. | 0xFF                               | 0xFF <sup>c</sup>        | RAM              |
| 0x03    | OEM GTIN byte 0 (MSB)                                                                                                              | 0xFF                               | no change                | NVM (lockable)   |
| 0x04    | OEM GTIN byte 1                                                                                                                    | 0xFF                               | no change                | NVM (lockable)   |
| 0x05    | OEM GTIN byte 2                                                                                                                    | 0xFF                               | no change                | NVM (lockable)   |
| 0x06    | OEM GTIN byte 3                                                                                                                    | 0xFF                               | no change                | NVM (lockable)   |
| 0x07    | OEM GTIN byte 4                                                                                                                    | 0xFF                               | no change                | NVM (lockable)   |
| 0x08    | OEM GTIN byte 5 (LSB)                                                                                                              | 0xFF                               | no change                | NVM (lockable)   |
| 0x09    | OEM identification number byte 0 (MSB)                                                                                             | 0xFF                               | no change                | NVM (lockable)   |
| 0x0A    | OEM identification number byte 1                                                                                                   | 0xFF                               | no change                | NVM (lockable)   |
| 0x0B    | OEM identification number byte 2                                                                                                   | 0xFF                               | no change                | NVM (lockable)   |
| 0x0C    | OEM identification number byte 3                                                                                                   | 0xFF                               | no change                | NVM (lockable)   |
| 0x0D    | OEM identification number byte 4                                                                                                   | 0xFF                               | no change                | NVM (lockable)   |
| 0x0E    | OEM identification number byte 5                                                                                                   | 0xFF                               | no change                | NVM (lockable)   |
| 0x0F    | OEM identification number byte 6                                                                                                   | 0xFF                               | no change                | NVM (lockable)   |
| 0x10    | OEM identification number byte 7 (LSB)                                                                                             | 0xFF                               | no change                | NVM (lockable)   |
| ≥ 0x11  | Additional control device information <sup>a</sup>                                                                                 | <sup>a</sup>                       | <sup>a</sup>             | <sup>a</sup>     |
| 0xFF    | Reserved – not implemented                                                                                                         | answer NO                          | no change                | n.a.             |

<sup>a</sup> Purpose, default/ power on/reset value and memory access of these bytes shall be defined by the manufacturer.

<sup>b</sup> Reset value after "RESET MEMORY BANK"

<sup>c</sup> Also used as power on value

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

The bytes in locations 0x09 to 0x10 ("OEM identification number byte 0" to "OEM identification number byte 7") should contain 64 bits of an identification number of the device.

product. If the bytes are used for the identification number, it shall be stored with the least significant byte in " identification number byte 7" and unused bits shall be filled with 0. These bytes should be programmed by the OEM.

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

#### **9.10.8 Manufacturer specific memory banks**

The manufacturer may use additional memory banks in the range of 2 to 199 to store additional information. The memory map of additional banks shall comply with Table 11.

#### **9.10.9 Reserved memory banks**

Memory banks 200 to 255 are reserved for future use and shall not be implemented.

### **9.11 Reset**

#### **9.11.1 Reset operation**

A control device shall implement a reset operation to set all device variables and instance variables (see Table 17 and Table 18) to their reset values.

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

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

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

#### **9.11.2 Reset memory bank operation**

A control device shall implement a reset operation to set the content of all unlocked memory banks (see 9.10) to their reset values, followed by locking the memory banks.

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

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

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

### **9.12 Power on behaviour**

#### **9.12.1 Power on**

After an external power cycle (see IEC 62386-101:2014, subclause 4.11.1), the device shall maintain its most recent configuration, with the following exceptions:

- the memory bank write enable state shall be disabled for all memory banks and the lock byte shall be set to 0xFF;
- quiescent mode shall be cancelled (see 9.9.3);
- All running timers shall be stopped and cancelled/reset;
- "powerCycleSeen" shall be set to TRUE.

“*powerCycleSeen*” can be observed through “QUERY DEVICE STATUS”.

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

In a system with multiple application controllers all application controllers may need power cycle information of other control devices in the system. Clearing “*powerCycleSeen*” should be done with some consideration.

### 9.12.2 Power cycle notification

After completing its external power cycle, a control device shall generate a power cycle event message if “*powerCycleNotification*” is ENABLED.

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

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

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

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

## 9.13 Priority use

### 9.13.1 General

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

- Priority 1 shall be used for all forward frames within a transaction (see IEC62386-101:2014, subclause 9.3), except for the first forward frame. Priority 1 shall neither be used for forward frames that are not part of a transaction, nor for those that start a transaction.
- Priority 2 should be used to execute user instigated actions for switching or dimming the lights. This implies appropriate event messages and arc power commands. Priority 2 might also be used during commissioning (e.g. addressing).

NOTE 1 Examples are switching or dimming actions triggered via push-button or presence detector.

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

NOTE 2 Examples are writing to memory banks or feedback events.

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

NOTE 3 Examples are switching or dimming actions triggered by a light sensor.

- Priority 5 should be used for periodic query commands.

### 9.13.2 Priority of input notifications

An instance shall use a default “*eventPriority*” equal to Priority 4 when transmitting an event message to produce an “INPUT NOTIFICATION (*device/instance, event*)”. For particular instance types, this default priority is subject to change by Parts 3xx of this standard.

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

## 9.14 Assigning short addresses

### 9.14.1 General

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

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

### 9.14.2 Random address allocation

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

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

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

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

The following (special) commands are initialisation commands:

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

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

### 9.14.3 Identification of a device

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

Identification shall be stopped upon reception of any instruction other than INITIALISE (*device*) or “IDENTIFY DEVICE”.

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

NOTE The actual procedure is manufacturer specific.

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

### 9.15 Exception handling

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

- An application controller shall change “*applicationControllerError*”. This status can be queried through “QUERY DEVICE STATUS” (see 9.16.2). Detailed error information may be available from “QUERY APPLICATION CONTROLLER ERROR” (see 11.6.4).
- A control device that is not an application controller shall have “*applicationControllerError*” set to FALSE.
- An input device shall change “*inputDeviceError*”. This status can be queried through “QUERY DEVICE STATUS” (see 9.16.2). Detailed error information may be available from “QUERY INPUT DEVICE ERROR” (see 11.6.5).
- A control device that is not an input device shall have “*inputDeviceError*” set to FALSE.
- An instance shall change “*instanceError*”. This status can be queried through “QUERY INSTANCE STATUS” (see 9.16.2.1). Detailed error information may be available from “QUERY INSTANCE ERROR” (see 11.9.4).

### 9.16 Device capabilities and status information

#### 9.16.1 Device capabilities

Each control device shall expose its features as a combination of device capabilities as given in Table 14:

**Table 14 – Control device capabilities**

| Bit | Description                                      | Value               | See subclause |
|-----|--------------------------------------------------|---------------------|---------------|
| 0   | “ <i>applicationControllerPresent</i> ” is TRUE? | “1” = “Yes”         | 9.1           |
| 1   | “ <i>numberOfInstances</i> ” is greater than 0?  | “1” = “Yes”         | 9.4.2         |
| 2-7 | unused                                           | “0” = default value |               |

The device capabilities can be queried using “QUERY DEVICE CAPABILITIES”.

#### 9.16.2 Device status

Each control device shall expose its status as a combination of device properties as given in Table 15:

**Table 15 – Control device status**

| Bit | Description                                    | Value               | See subclause |
|-----|------------------------------------------------|---------------------|---------------|
| 0   | “ <i>inputDeviceError</i> ” is TRUE?           | “1” = “Yes”         | 9.14.3        |
| 1   | “ <i>quiescentMode</i> ” is ENABLED?           | “1” = “Yes”         | 9.9.3         |
| 2   | “ <i>shortAddress</i> ” is MASK?               | “1” = “Yes”         | 9.5           |
| 3   | “ <i>applicationActive</i> ” is TRUE?          | “1” = “Yes”         | 9.9.1         |
| 4   | “ <i>applicationControllerError</i> ” is TRUE? | “1” = “Yes”         | 9.14.3        |
| 5   | “ <i>powerCycleSeen</i> ” is TRUE?             | “1” = “Yes”         | 9.12          |
| 6   | “ <i>resetState</i> ” is TRUE?                 | “1” = “Yes”         | 9.16.2.1      |
| 7   | unused                                         | “0” = default value |               |



The device status can be queried using “QUERY DEVICE STATUS”.

#### 9.16.2.1 Bit 6: Reset state

“resetState” shall be set to TRUE if all the NVM variables mentioned in Table 17 and Table 18 are at their reset value. The NVM variables that are marked with ‘no change’ in the reset value column shall not be considered. NVM variables defined in implemented Parts 3xx shall be included.

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

#### 9.16.3 Instance status

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

**Table 16 – Instance status**

| Bit | Description               | Value               | See subclause |
|-----|---------------------------|---------------------|---------------|
| 0   | “instanceError” is TRUE?  | “1” = “Yes”         | 9.14.3        |
| 1   | “instanceActive” is TRUE? | “1” = “Yes”         | 9.9.2         |
| 2-7 | unused                    | “0” = default value |               |

The instance status can be observed using “QUERY INSTANCE STATUS”.

#### 9.17 Non-volatile memory

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

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

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

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

Physically saving the variables in response to the instruction shall take at most 300 ms to complete. While the saving operation is on-going, the control device may or may not respond to any command.

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

## 10 Declaration of variables

Table 17 shows the default values, the reset values, the range of validity and the type of memory of the defined instance independent variables.

Table 18 shows the default values, the reset values, the range of validity and the type of memory of the defined variables of each of the instances.

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

**Table 17 – Declaration of device variables**

| VARIABLE                                                                        | DEFAULT VALUE (factory)      | RESET VALUE | POWER ON VALUE     | RANGE OF VALIDITY              | MEMORY TYPE |
|---------------------------------------------------------------------------------|------------------------------|-------------|--------------------|--------------------------------|-------------|
| "shortAddress"                                                                  | MASK (no address)            | no change   | no change          | [0,63], MASK                   | NVM         |
| "deviceGroups"                                                                  | 0x0000 0000                  | 0x0000 0000 | no change          | [0,0xFFFF FFFF]                | NVM         |
| "searchAddress"                                                                 | <sup>a</sup>                 | 0xFF FF FF  | 0xFF FF FF         | [0,0xFF FF FF]                 | RAM         |
| "randomAddress"                                                                 | 0xFF FF FF                   | 0xFF FF FF  | no change          | [0,0xFF FF FF]                 | NVM         |
| "DTR0"                                                                          | <sup>a</sup>                 | no change   | 0x00               | [0,0xFF]                       | RAM         |
| "DTR1"                                                                          | <sup>a</sup>                 | no change   | 0x00               | [0,0xFF]                       | RAM         |
| "DTR2"                                                                          | <sup>a</sup>                 | no change   | 0x00               | [0,0xFF]                       | RAM         |
| "numberOfInstances"                                                             | factory burn-in              | no change   | no change          | [0,32]                         | ROM         |
| "operatingMode"                                                                 | factory burn-in              | no change   | no change          | 0, [0x80,0xFF]                 | NVM         |
| "quiescentMode"                                                                 | <sup>a</sup>                 | DISABLED    | DISABLED           | [ENABLED, DISABLED]            | RAM         |
| "applicationActive"                                                             | applicationControllerPresent | no change   | no change          | [TRUE, FALSE]                  | NVM         |
| "writeEnableState"                                                              | <sup>a</sup>                 | DISABLED    | DISABLED           | [ENABLED, DISABLED]            | RAM         |
| "applicationControllerPresent"                                                  | factory burn-in              | no change   | no change          | [TRUE, FALSE]                  | ROM         |
| "powerCycleSeen"                                                                | <sup>a</sup>                 | FALSE       | TRUE               | [TRUE, FALSE]                  | RAM         |
| "powerCycleNotification"                                                        | DISABLED                     | no change   | no change          | [ENABLED, DISABLED]            | NVM         |
| "initialisationState"                                                           | <sup>a</sup>                 | DISABLED    | DISABLED           | [ENABLED, DISABLED, WITHDRAWN] | RAM         |
| "applicationControllerError"                                                    | <sup>a</sup>                 | FALSE       | FALSE <sup>b</sup> | [TRUE, FALSE]                  | RAM         |
| "inputDeviceError"                                                              | <sup>a</sup>                 | FALSE       | FALSE <sup>b</sup> | [TRUE, FALSE]                  | RAM         |
| "resetState"                                                                    | TRUE                         | TRUE        | TRUE <sup>b</sup>  | [TRUE, FALSE]                  | RAM         |
| <sup>a</sup> Not applicable.                                                    |                              |             |                    |                                |             |
| <sup>b</sup> The value should reflect the actual situation as soon as possible. |                              |             |                    |                                |             |

**Table 18 – Declaration of instance variables**

| VARIABLE                                                                                                                       | DEFAULT VALUE (factory) | RESET VALUE | POWER ON VALUE         | RANGE OF VALIDITY                   | MEMORY TYPE |
|--------------------------------------------------------------------------------------------------------------------------------|-------------------------|-------------|------------------------|-------------------------------------|-------------|
| "instanceGroup0"                                                                                                               | MASK                    | MASK        | no change              | [0,31], MASK                        | NVM         |
| "instanceGroup1"                                                                                                               | MASK                    | MASK        | no change              | [0,31], MASK                        | NVM         |
| "instanceGroup2"                                                                                                               | MASK                    | MASK        | no change              | [0,31], MASK                        | NVM         |
| "instanceActive"                                                                                                               | TRUE                    | no change   | no change              | [TRUE, FALSE]                       | NVM         |
| "instanceType"                                                                                                                 | factory burn-in         | no change   | no change              | [0,31]                              | ROM         |
| "resolution"                                                                                                                   | factory burn-in         | no change   | no change              | [1,255]                             | ROM         |
| "inputValue"                                                                                                                   | <sup>a</sup>            | no change   | no change <sup>b</sup> | $[0, 2^{N*8}-1]^c$                  | RAM         |
| "instanceNumber"                                                                                                               | factory burn-in         | no change   | no change              | $[0, \text{"numberOfInstances"}-1]$ | ROM         |
| "eventFilter" <sup>d</sup>                                                                                                     | 0xFF FF FF              | 0xFF FF FF  | no change              | [0, 0xFF FF FF]                     | NVM         |
| "eventScheme"                                                                                                                  | 0                       | 0           | no change              | [0,4]                               | NVM         |
| "eventPriority" <sup>d</sup>                                                                                                   | 4                       | no change   | no change              | [2,5]                               | NVM         |
| "instanceError"                                                                                                                | <sup>a</sup>            | FALSE       | FALSE <sup>b</sup>     | [TRUE, FALSE]                       | RAM         |
| <sup>a</sup> Not applicable.                                                                                                   |                         |             |                        |                                     |             |
| <sup>b</sup> The value should reflect the actual situation as soon as possible.                                                |                         |             |                        |                                     |             |
| <sup>c</sup> $N$ computed as ("resolution"/8) rounded up to the nearest integer.                                               |                         |             |                        |                                     |             |
| <sup>d</sup> For particular instance types, the values belonging to this variable can be changed by Part 3xx of this standard. |                         |             |                        |                                     |             |

## 11 Definition of commands

### 11.1 General

Unused opcodes are reserved for future needs.

### 11.2 Overview sheets

Table 19 to Table 22 give an overview of the control device's event messages and commands.

**Table 19 – Instance event messages**

| Event message name                                   | Event source information | Event info   | References      | Command subclause |
|------------------------------------------------------|--------------------------|--------------|-----------------|-------------------|
| INPUT NOTIFICATION ( <i>device/instance, event</i> ) | <i>device/instance</i>   | <i>event</i> | 9.6.2 and 9.7.4 | 11.3.1            |

**Table 20 – Device event messages**

| Event message name                   | Bits [23,13] | Bits [12,0]   | References       | Command subclause |
|--------------------------------------|--------------|---------------|------------------|-------------------|
| POWER NOTIFICATION ( <i>device</i> ) | 0x7F7        | <i>device</i> | 9.6.2 and 9.12.2 | 11.3.2            |

Table 21 – Standard commands

| Command name                                | Address byte | Instance byte | Opcode byte | App Ctrl | Input dev | DTR0 | DTR1 | DTR2 | Answer | Send twice | See subclause | Command subclause |
|---------------------------------------------|--------------|---------------|-------------|----------|-----------|------|------|------|--------|------------|---------------|-------------------|
| IDENTIFY DEVICE                             | Device       | 0xFE          | 0x00        | ✓        | ✓         |      |      |      |        | ✓          | 9.14.3        | 11.4.2            |
| RESET POWER CYCLE SEEN                      | Device       | 0xFE          | 0x01        | ✓        | ✓         |      |      |      |        | ✓          | 9.12.1        | 11.4.3            |
| RESET                                       |              |               |             |          |           |      |      |      |        |            |               |                   |
| RESET MEMORY BANK (DTR0)                    | Device       | 0xFE          | 0x10        | ✓        | ✓         |      |      |      |        | ✓          | 9.11.1        | 11.5.2            |
| SET SHORT ADDRESS (DTR0)                    | Device       | 0xFE          | 0x11        | ✓        | ✓         | ✓    |      |      |        | ✓          | 9.11.2        | 11.5.3            |
| ENABLE WRITE MEMORY                         | Device       | 0xFE          | 0x14        | ✓        | ✓         | ✓    |      |      |        | ✓          |               | 11.5.4            |
| ENABLE APPLICATION CONTROLLER               | Device       | 0xFE          | 0x15        | ✓        | ✓         |      |      |      |        | ✓          | 9.10.5        | 11.5.5            |
| DISABLE APPLICATION CONTROLLER              | Device       | 0xFE          | 0x16        | ✓        |           |      |      |      |        | ✓          | 9.9.1         | 11.5.6            |
| SET OPERATING MODE (DTR0)                   | Device       | 0xFE          | 0x17        | ✓        |           |      |      |      |        | ✓          | 9.9.1         | 11.5.7            |
| ADD TO DEVICE GROUPS 0-15 (DTR2:DTR1)       | Device       | 0xFE          | 0x18        | ✓        | ✓         | ✓    |      |      |        | ✓          | 9.9.4         | 11.5.8            |
| ADD TO DEVICE GROUPS 16-31 ( DTR2:DTR1 )    | Device       | 0xFE          | 0x19        | ✓        | ✓         |      | ✓    | ✓    |        | ✓          |               | 11.5.9            |
| REMOVE FROM DEVICE GROUPS 0-15 (DTR2:DTR1)  | Device       | 0xFE          | 0x1A        | ✓        | ✓         |      | ✓    | ✓    |        | ✓          |               | 11.5.10           |
| REMOVE FROM DEVICE GROUPS 16-31 (DTR2:DTR1) | Device       | 0xFE          | 0x1B        | ✓        | ✓         |      | ✓    | ✓    |        | ✓          |               | 11.5.11           |
| START QUIESCENT MODE                        | Device       | 0xFE          | 0x1C        | ✓        | ✓         |      | ✓    | ✓    |        | ✓          |               | 11.5.12           |
| STOP QUIESCENT MODE                         | Device       | 0xFE          | 0x1D        | ✓        | ✓         |      |      |      |        | ✓          | 9.9.3         | 11.5.13           |
| ENABLE POWER CYCLE NOTIFICATION             | Device       | 0xFE          | 0x1E        | ✓        | ✓         |      |      |      |        | ✓          | 9.9.3         | 11.5.14           |
| DISABLE POWER CYCLE NOTIFICATION            | Device       | 0xFE          | 0x1F        | ✓        | ✓         |      |      |      |        | ✓          | 9.12.2        | 11.5.15           |
| SAVE PERSISTENT VARIABLES                   | Device       | 0xFE          | 0x20        | ✓        | ✓         |      |      |      |        | ✓          | 9.12.2        | 11.5.16           |
|                                             | Device       | 0xFE          | 0x21        | ✓        | ✓         |      |      |      |        | ✓          | 9.17          | 11.5.17           |
| QUERY DEVICE STATUS                         | Device       | 0xFE          | 0x30        | ✓        | ✓         |      |      |      | ✓      |            | 9.16.2        | 11.6.3            |
| QUERY APPLICATION CONTROLLER ERROR          | Device       | 0xFE          | 0x31        | ✓        | ✓         |      |      |      | ✓      |            | 9.14.3        | 11.6.4            |
| QUERY INPUT DEVICE ERROR                    | Device       | 0xFE          | 0x32        | ✓        | ✓         |      |      |      | ✓      |            | 9.14.3        | 11.6.5            |
| QUERY MISSING SHORT ADDRESS                 | Device       | 0xFE          | 0x33        | ✓        | ✓         |      |      |      | ✓      |            |               | 11.6.6            |

| Command name                                       | Address byte | Instance byte | Opcode byte | App Ctrl | Input dev | DTR0 | DTR1 | DTR2 | Answer | Send twice | See subclause | Command subclause |
|----------------------------------------------------|--------------|---------------|-------------|----------|-----------|------|------|------|--------|------------|---------------|-------------------|
| QUERY VERSION NUMBER                               | Device       | 0xFE          | 0x34        | ✓        | ✓         |      |      |      | ✓      |            | 4.2           | 11.6.7            |
| QUERY NUMBER OF INSTANCES                          | Device       | 0xFE          | 0x35        | ✓        | ✓         |      |      |      | ✓      |            | 9.4           | 11.6.9            |
| QUERY CONTENT DTR0                                 | Device       | 0xFE          | 0x36        | ✓        | ✓         | ✓    |      |      | ✓      |            |               | 11.6.8            |
| QUERY CONTENT DTR1                                 | Device       | 0xFE          | 0x37        | ✓        | ✓         |      | ✓    |      | ✓      |            |               | 11.6.10           |
| QUERY CONTENT DTR2                                 | Device       | 0xFE          | 0x38        | ✓        | ✓         |      |      | ✓    | ✓      |            |               | 11.6.11           |
| QUERY RANDOM ADDRESS (H)                           | Device       | 0xFE          | 0x39        | ✓        | ✓         |      |      |      | ✓      |            |               | 11.6.12           |
| QUERY RANDOM ADDRESS (M)                           | Device       | 0xFE          | 0x3A        | ✓        | ✓         |      |      |      | ✓      |            |               | 11.6.13           |
| QUERY RANDOM ADDRESS (L)                           | Device       | 0xFE          | 0x3B        | ✓        | ✓         |      |      |      | ✓      |            |               | 11.6.14           |
| READ MEMORY LOCATION ( <i>DTR1</i> , <i>DTR0</i> ) | Device       | 0xFE          | 0x3C        | ✓        | ✓         | ✓    | ✓    |      | ✓      |            | 9.10.4        | 11.6.15           |
| QUERY APPLICATION CONTROL ENABLED                  | Device       | 0xFE          | 0x3D        | ✓        | ✓         |      |      |      | ✓      |            | 9.9.1         | 11.6.16           |
| QUERY OPERATING MODE                               | Device       | 0xFE          | 0x3E        | ✓        | ✓         |      |      |      | ✓      |            | 9.9.4         | 11.6.17           |
| QUERY MANUFACTURER SPECIFIC MODE                   | Device       | 0xFE          | 0x3F        | ✓        | ✓         |      |      |      | ✓      |            | 9.9.4         | 11.6.18           |
| QUERY QUIESCENT MODE                               | Device       | 0xFE          | 0x40        | ✓        | ✓         |      |      |      | ✓      |            | 9.9.3         | 11.6.19           |
| QUERY DEVICE GROUPS 0-7                            | Device       | 0xFE          | 0x41        | ✓        | ✓         |      |      |      | ✓      |            |               | 11.6.20           |
| QUERY DEVICE GROUPS 8-15                           | Device       | 0xFE          | 0x42        | ✓        | ✓         |      |      |      | ✓      |            |               | 11.6.21           |
| QUERY DEVICE GROUPS 16-23                          | Device       | 0xFE          | 0x43        | ✓        | ✓         |      |      |      | ✓      |            |               | 11.6.22           |
| QUERY DEVICE GROUPS 24-31                          | Device       | 0xFE          | 0x44        | ✓        | ✓         |      |      |      | ✓      |            |               | 11.6.23           |
| QUERY POWER CYCLE NOTIFICATION                     | Device       | 0xFE          | 0x45        | ✓        | ✓         |      |      |      | ✓      |            | 9.12.2        | 11.6.24           |
| QUERY DEVICE CAPABILITIES                          | Device       | 0xFE          | 0x46        | ✓        | ✓         |      |      |      | ✓      |            | 9.16.1        | 11.6.2            |
| QUERY EXTENDED VERSION NUMBER( <i>DTR0</i> )       | Device       | 0xFE          | 0x47        | ✓        | ✓         | ✓    |      |      | ✓      |            |               | 11.6.25           |
| QUERY RESET STATE                                  | Device       | 0xFE          | 0x48        | ✓        | ✓         |      |      |      | ✓      |            | 9.16.2.1      | 11.6.26           |
| SET EVENT PRIORITY ( <i>DTR0</i> )                 | Device       | Instance      | 0x61        |          | ✓         | ✓    |      |      |        | ✓          | 9.13.2        | 11.8.8            |
| ENABLE INSTANCE                                    | Device       | Instance      | 0x62        |          | ✓         |      |      |      |        | ✓          | 9.9.2         | 11.8.2            |
| DISABLE INSTANCE                                   | Device       | Instance      | 0x63        |          | ✓         |      |      |      |        | ✓          | 9.9.2         | 11.8.3            |

| Command name                                 | Address byte | Instance byte | Opcode byte | App Ctrl | Input dev | DTR0 | DTR1 | DTR2 | Answer | Send twice | See subclause | Command subclause |
|----------------------------------------------|--------------|---------------|-------------|----------|-----------|------|------|------|--------|------------|---------------|-------------------|
| SET PRIMARY INSTANCE GROUP ( <i>DTR0</i> )   | Device       | Instance      | 0x64        |          | ✓         | ✓    |      |      |        | ✓          | 9.4.5         | 11.8.4            |
| SET INSTANCE GROUP 1 ( <i>DTR0</i> )         | Device       | Instance      | 0x65        |          | ✓         | ✓    |      |      |        | ✓          | 9.4.5         | 11.8.5            |
| SET INSTANCE GROUP 2 ( <i>DTR0</i> )         | Device       | Instance      | 0x66        |          | ✓         | ✓    |      |      |        | ✓          | 9.4.5         | 11.8.6            |
| SET EVENT SCHEME ( <i>DTR0</i> )             | Device       | Instance      | 0x67        |          | ✓         | ✓    |      |      |        | ✓          | 9.6.2         | 11.8.7            |
| SET EVENT FILTER ( <i>DTR2, DTR1, DTR0</i> ) | Device       | Instance      | 0x68        |          | ✓         | ✓    | ✓    | ✓    |        | ✓          | 9.6.4         | 11.8.9            |
|                                              |              |               |             |          |           |      |      |      |        |            |               |                   |
| QUERY INSTANCE TYPE                          | Device       | Instance      | 0x80        |          | ✓         |      |      |      | ✓      |            | 9.4.3         | 11.9.2            |
| QUERY RESOLUTION                             | Device       | Instance      | 0x81        |          | ✓         |      |      |      | ✓      |            | 9.7.2         | 11.9.3            |
| QUERY INSTANCE ERROR                         | Device       | Instance      | 0x82        |          | ✓         |      |      |      | ✓      |            | 9.14.3        | 11.9.4            |
| QUERY INSTANCE STATUS                        | Device       | Instance      | 0x83        |          | ✓         |      |      |      | ✓      |            | 9.16.2.1      | 11.9.5            |
| QUERY EVENT PRIORITY                         | Device       | Instance      | 0x84        |          | ✓         |      |      |      | ✓      |            | 9.13.2        | 11.9.13           |
| QUERY INSTANCE ENABLED                       | Device       | Instance      | 0x86        |          | ✓         |      |      |      | ✓      |            | 9.9.2         | 11.9.6            |
| QUERY PRIMARY INSTANCE GROUP                 | Device       | Instance      | 0x88        |          | ✓         |      |      |      | ✓      |            | 9.4.5         | 11.9.7            |
| QUERY INSTANCE GROUP 1                       | Device       | Instance      | 0x89        |          | ✓         |      |      |      | ✓      |            | 9.4.5         | 11.9.8            |
| QUERY INSTANCE GROUP 2                       | Device       | Instance      | 0x8A        |          | ✓         |      |      |      | ✓      |            | 9.4.5         | 11.9.9            |
| QUERY EVENT SCHEME                           | Device       | Instance      | 0x8B        |          | ✓         |      |      |      | ✓      |            | 9.6.2         | 11.9.10           |
| QUERY INPUT VALUE                            | Device       | Instance      | 0x8C        |          | ✓         |      |      |      | ✓      |            | 9.7.3         | 11.9.11           |
| QUERY INPUT VALUE LATCH                      | Device       | Instance      | 0x8D        |          | ✓         |      |      |      | ✓      |            | 9.7.3         | 11.9.12           |
| QUERY FEATURE TYPE                           | Device       | Instance      | 0x8E        |          | ✓         |      |      |      | ✓      |            | 0             | 11.9.14           |
| QUERY NEXT FEATURE TYPE                      | Device       | Instance      | 0x8F        |          | ✓         |      |      |      | ✓      |            | 0             | 11.9.15           |
| QUERY EVENT FILTER 0-7                       | Device       | Instance      | 0x90        |          | ✓         |      |      |      | ✓      |            | 9.6.4         | 11.9.16           |
| QUERY EVENT FILTER 8-15                      | Device       | Instance      | 0x91        |          | ✓         |      |      |      | ✓      |            | 9.6.4         | 11.9.17           |
| QUERY EVENT FILTER 16-23                     | Device       | Instance      | 0x92        |          | ✓         |      |      |      | ✓      |            | 9.6.4         | 11.9.18           |

Table 22 – Special commands (implemented by both application controller and input device)

| Command name                                                                 | Address byte | Instance byte | Opcode byte   | DTR0 | DTR1 | DTR2 | Answer | Send twice | See subclause | Command subclause |
|------------------------------------------------------------------------------|--------------|---------------|---------------|------|------|------|--------|------------|---------------|-------------------|
| TERMINATE                                                                    | 0xC1         | 0x00          | 0x00          |      |      |      |        |            |               | 11.10.1           |
| INITIALISE ( <i>device</i> )                                                 | 0xC1         | 0x01          | <i>device</i> |      |      |      |        | ✓          | 9.14          | 11.10.3           |
| RANDOMISE                                                                    | 0xC1         | 0x02          | 0x00          |      |      |      |        | ✓          | 9.14          | 11.10.4           |
| COMPARE                                                                      | 0xC1         | 0x03          | 0x00          |      |      |      | ✓      |            | 9.14          | 11.10.5           |
| WITHDRAW                                                                     | 0xC1         | 0x04          | 0x00          |      |      |      |        |            | 9.14          | 11.10.6           |
| SEARCHADDRH ( <i>data</i> )                                                  | 0xC1         | 0x05          | <i>data</i>   |      |      |      |        |            | 9.14          | 11.10.7           |
| SEARCHADDRM ( <i>data</i> )                                                  | 0xC1         | 0x06          | <i>data</i>   |      |      |      |        |            | 9.14          | 11.10.8           |
| SEARCHADDRL ( <i>data</i> )                                                  | 0xC1         | 0x07          | <i>data</i>   |      |      |      |        |            | 9.14          | 11.10.9           |
| PROGRAM SHORT ADDRESS ( <i>data</i> )                                        | 0xC1         | 0x08          | <i>data</i>   |      |      |      |        |            | 9.14          | 11.10.10          |
| VERIFY SHORT ADDRESS ( <i>data</i> )                                         | 0xC1         | 0x09          | <i>data</i>   |      |      |      | ✓      |            | 9.14          | 11.10.11          |
| QUERY SHORT ADDRESS                                                          | 0xC1         | 0x0A          | 0x00          |      |      |      | ✓      |            | 9.14          | 11.10.12          |
| WRITE MEMORY LOCATION ( <i>DTR1</i> , <i>DTR0</i> , <i>data</i> )            | 0xC1         | 0x20          | <i>data</i>   | ✓    | ✓    |      | ✓      |            | 9.10.5        | 11.10.13          |
| WRITE MEMORY LOCATION – NO REPLY ( <i>DTR1</i> , <i>DTR0</i> , <i>data</i> ) | 0xC1         | 0x21          | <i>data</i>   | ✓    | ✓    |      |        |            | 9.10.5        | 11.10.14          |
| DTR0 ( <i>data</i> )                                                         | 0xC1         | 0x30          | <i>data</i>   | ✓    |      |      |        |            |               | 11.10.15          |
| DTR1 ( <i>data</i> )                                                         | 0xC1         | 0x31          | <i>data</i>   |      | ✓    |      |        |            |               | 11.10.16          |
| DTR2 ( <i>data</i> )                                                         | 0xC1         | 0x32          | <i>data</i>   |      |      | ✓    |        |            |               | 11.10.17          |
| SEND TESTFRAME ( <i>data</i> )                                               | 0xC1         | 0x33          | <i>data</i>   | ✓    | ✓    | ✓    |        |            |               | 11.10.21          |
| DIRECT WRITE MEMORY ( <i>DTR1</i> , <i>offset</i> , <i>data</i> )            | 0xC5         | <i>offset</i> | <i>data</i>   | ✓    | ✓    |      | ✓      |            | 9.10.5        | 11.10.18          |
| DTR1:DTR0 ( <i>data1</i> , <i>data0</i> )                                    | 0xC7         | <i>data1</i>  | <i>data0</i>  | ✓    | ✓    |      |        |            |               | 11.10.19          |
| DTR2:DTR1 ( <i>data2</i> , <i>data1</i> )                                    | 0xC9         | <i>data2</i>  | <i>data1</i>  |      | ✓    | ✓    |        |            |               | 11.10.20          |

### 11.3 Event messages

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

The event message notifies of a change or a series of changes of “*inputValue*” at the instance of an input device as required by this specification or by the Part 3xx of IEC 62386 corresponding to the “*instanceType*” of the instance.

The transmitting instance shall

- generate the event message only while “*instanceActive*” is TRUE
- generate the event message only while it is not in an error condition that prevents operation (see 9.14.3)
- use the currently active “*eventScheme*”.
- use the requested “*eventPriority*”.

Refer to 9.6 and 9.7 for further information.

#### 11.3.2 POWER NOTIFICATION (*device*)

The event notifies of a control device power cycle completion and shall be generated following the requirements as stated in 9.12.2.

### 11.4 Device control instructions

#### 11.4.1 General

Device control instructions are used to modify property values of a control device. For this reason a device control instruction shall not be executed, unless it is received twice according to the requirements as stated in 9.3 of IEC 62386-101:2014.

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

- the instruction shall be ignored if so required by the provisions of 9.5 of this standard;
- the control device shall not reply to the instruction.

#### 11.4.2 IDENTIFY DEVICE

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

The identification shall be aborted immediately upon reception of “TERMINATE” or “WITHDRAW”.

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

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

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

Refer to 9.14 for further information.